Abstract

In this paper, the theoretical modeling of interactive performance on a distributed virtual environment (DVE) is carried out. To study the degradation of the information quality caused by the existence of network latency on a simple virtual task model, Markov model is employed. Compared with the numerical simulation, the proposed model can be a good approximation of the simulation results. The results of subjective experiments show that, in average, the accuracy of human foreseeing behavior marked between the score of the first and the second order Markov models.

1. Introduction

Recently, much research has come to be focused on virtual space over a network, i.e., distributed virtual environments (DVE) [1-4]. In this paper, we construct and study a theoretical model of information quality in a DVE having latency. We study, in particular, the use of Markov models to estimate interactive and user behavior on the basis of system parameters such as lag and data-reception interval (update interval) for a generalized task model.

2. Synchronized DVE

This paper considers a server-client model as the most general type of DVE. The server manages and updates terminal entities (e.g. physical information such as position etc.) so that the terminals participating in the DVE can share the virtual space. Time in the DVE is synchronized beforehand both the server and clients. Here, a DVE is defined as an imitation of the space having physical features, and the entities of each avatar (virtual user in the DVE) undergo change continuously over time due to manipulation performed by the user in question. The server manages the entities of each avatar in a uniform fashion and updates those entities at server state-update time based on each user’s avatar manipulation information. After updating, the server immediately sends entity information to each client. The client, meanwhile, obtains avatar manipulation information input by the user and sends it to the server as needed. At frame update time, it presents the latest entity information received from the server. In the above way, a model in which the server’s and each client’s frame update timing is completely synchronized on a fixed time interval is defined as a “synchronized DVE,” the subject of this paper.

In general, it is desirable that concurrent entity information among all clients be presented continuously over time to facilitate smooth task execution. Here, “concurrent entity information” means the consistent entities information, referring the entities information of each avatar on the server at any point in time. “Latesest entities information,” on the other hand, means the most recently received entities information at a client. In a synchronized DVE, the frequency of sending entity information from the server to clients is limited due to limitations in network bandwidth [1][2]. It is therefore impossible for each client to present the entities information of other avatars continuously over time, and a user has to perform the task by foreseeing the entities of the next frame.

3. Network latency and information quality

3.1. Estimation by Markov models

We focus on one client terminal in a DVE and examine the circumstances under which the user of that client foresees the entity of another avatar in the system. We assume here that delays caused by frame presentation and user operations, for example, are negligible compared to the factors described above. Let A(t) denote the vector representing the entity of avatar A at time t, let t denote the sending time of the ith set of data sent from the server, and let u and d denote the DVE frame update interval and the delay incurred in receiving the data sent from the server, respectively. At the client, entity A(t) sent from the server at time t is presented at time t+nu (where n≥1).

Now, as a model of common prediction method, we introduce Markov models that use the last several frames of entities as parameters. In this paper, we consider Markov models of orders 1 and 2 (a Markov model of order m is denoted below as an "m-th order model") [1][2].

A common prediction method based on the 1st order Markov model assumes that the entity of the next frame (time t+1) is the same as the latest entity A(t). Accordingly, denoting predicted entity of the next frame as Â(t+1), we get Â(t+1) = A(t).

Next, a prediction method based on the 2nd order model assumes that the entity of the next frame (time t+1) changes as a result of uniform motion. It changes, in other words, while maintaining the velocity observed over the last two frames and neglecting vi(t) = (A(t)-A(t-1))/u(t-1)

We therefore get the expression for predicted entity of the next frame denoted as Â(t+1).

\[ Â(t+1) = A(t) + (A(t) - A(t-1)) \] (1)

3.2. Approximation model for information quality degradation

We first take a look at the error generated by next-frame prediction. In the following discussion, \( L \) denotes the total number of elapsed frames in task execution.

The error generated between entity A(t) in the ith frame predicted by the 1st-order model and entity A(t) actually received is equivalent to the motion A(t)̂ - A(t). This error is considered to be bounded between frames.

Since A(t) can be assumed to be a continuous and differentiable function of a variable t, we can consider the Taylor expansion of A(t) around t=t_n. The magnitude of that error E_u(t) = |A(t) - A(t)̂| can therefore be calculated as \[ |A(t+1) - A(t)_n| = |A(t) - A(t-1) + \frac{d^2A}{dt^2}(t, t_{n+1})|/2 + \cdots \] (2)

The second term of the right side above can be neglected. In the first term, (t_n, u) and dA/dt_n(t=ti-1) can be replaced by the update interval u and the velocity at t=t_n, respectively, the right side is written as, \[ E_u(t) = |v(t)|u \] (3)

From this, average \( E_u(t) \) per frame can be expressed as follows denoting the average of \( |v(t)| \) as \( \|\| \).

\[ E_u = \frac{1}{L} \frac{1}{u} \sum_{t=1}^{t} |v(t)|u = \|\|u \] (4)

Now, the error generated as a result of prediction using the 2nd-order model is equivalent to divergence from uniform motion, i.e., to acceleration or higher temporal differential terms that occurs between frames. So, using Eq.(2) the magnitude of this error can be approximated by taking the integral of absolute acceleration \( |a(t)| \) generated at time t=t_n. The magnitude of that error \( E_a(t) = |a(t)|u^2/2 \) can be calculated as follows.

\[ E_a = \frac{1}{L} \frac{1}{u^2} \sum_{t=1}^{t} |a(t)|u^2/2 = \|a\|^2/2 \] (5)

From this, the average of \( E_a(t) \) can be given as follows denoting the average of \( |a(t)| \) as \( \|a\| \).

\[ E_a = \frac{1}{L} \frac{1}{u^2} \sum_{t=1}^{t} |a(t)|u^2/2 = \|a\|^2/2 \] (6)

We next take a look at error caused by the lag. The number of frames needed to request entity information sent by the server at the client can be computed as follows regardless of Markov order.

\[ n = \lceil \frac{\delta}{u} \rceil + 1 \] (7)

The square brackets \( \lceil x \rceil \) denotes Gauss symbol whereby \( \lceil x \rceil \) means the greatest integer that is less than or equal to x. Thus, at time t_n, lag-induced error is defined as \( E_l(t_n) = (A(t_n) - A(t_n+1)) \), difference between concurrent entity \( A(t) \) and latest entity \( A(t+1) \). By substituting Eq. (2) into the right side of \( E_a \) above and neglecting the term higher than the second order, it can be calculated as follows.

\[ E_l = \|v(t)\|u \] (8)

Average \( E_l(t) \) can therefore be expressed as follows.

\[ E_l = \|v(t)\|u \] (9)

In the above, \( E_a \) is error caused by prediction, referred to as prediction error, whereas \( E_l \) is temporal ambiguity unrelated to prediction where no information can be given regarding the amount of lag. It can therefore be assumed that these two factors are statistically independent. Average of total error \( E \) per frame can therefore be given as a summation of \( E_a \) and \( E_l \) as follows.

1. (1st-order model) \[ E = \|v(t)\|u + \|a\|^2/2 \] (10)

2. (2nd-order model) \[ E = \|v(t)\|u + \|a\|^2/2 + \|v(t)\|u \] (11)
4. Numerical simulations

4.1. Role of numerical experiment
We performed simulations to evaluate the validity of the proposed theoretical models. In these simulations, specific entity-change patterns are given and the average error per frame is calculated when predicting it by system on the basis of a Markov model. The error is referred to as "total error" in units of [pixel/frame]. It is used to evaluate degradation in information quality between the concurrent entity values and the predicted entity values. In the following discussion, "prediction" is defined as the estimation of the entity values by system and "foreseeing" is that by the human. In order to generate the patterns, we employ two different formulae (Eq. (12) and Eq. (13)). The former is a trigonometric polynomial, and the latter is a Gabor wavelet polynomial.

\[
A(t) = 100 \sum_{i=1}^{2} \sin^2 \left( \sqrt{\pi} \alpha \pi (t+25 + \alpha_i) \right)
\]

(Type I)

\[
A(t) = 1600 \sum_{i=1}^{2} \exp \left( \frac{(t+5)}{2} \right) \sin^2 \left( \frac{2\pi}{25} \right) / 2\sqrt{0.5}\pi + \exp \left( \frac{t+5-20\beta^2}{8} \right) \sin \left( \frac{t}{2} \right) / 2\sqrt{0.5}\pi
\]

(Type II)

Here, in both formulas, \(A(t)\) indicates a generated entity value where the variable \(t\) indicates time. In Eq. (12) and Eq. (13), a random entity-change pattern is generated by assigning random number greater than 0 and less than 1, to variables \(\alpha_i\), \(\beta_i\), and \(\beta_i\) respectively at the beginning of each trial. The variable \(c\) is a parameter for adjusting entity velocity and acceleration. In the experiment, \(c\) is set to 1 or 0.5. When changing from 1 to 0.5, velocity and acceleration change by \(\sqrt{0.5}\) times and 1/2 times respectively. From a characteristic viewpoint of each measurements, 10,000 patterns are generated per case and an average is measured from \(t=0\) to 25 for each simulation case. In the experiments, specific entity-change patterns are taken for each.

5. Subjective experiment

5.1. Experimental method
We also performed a subjective experiment to examine errors in foreseeing made by human subjects and to compare its results with those of the proposed theoretical model in the previous section (Eq. (10) and Eq. (11)). To begin with, we constructed a DVE consisting of two terminals on a LAN. In this DVE, the server changes, at a fixed update interval, the entity of a "remote avatar" that can be observed by the subject and sends the entity information to the client. The remote avatar on the client is drawn as a circle in one-dimensional task space, and the task to be performed by the subject is to foresee the entity value of this avatar to be updated in the next frame by the client. In addition, a "local avatar" is displayed on the client screen to indicate avatar entities foreseen by the subject. The position of the local avatar obtained by the client at the time of frame update is taken to be the subject's foreseen value of the other avatar's entity (position). Type I was employed as the model for entity change. Lag and update interval in this subjective experiment were equally set as in the numerical simulation.

5.2. Results

The experiment was performed using eight subjects. Based on the results we see that, in average, foreseeing error by human caused by lag and update interval was between the results obtained by the 1st-order and the 2nd-order models in the proposed theoretical model (Table 2).

![Figure 1. Approximation by the theoretical model (2nd-order model, \(c_i=0.5\))](image)

4.2. Results

In any case, percentage discrepancy between the numerical simulations and the theoretical models was about 5% in average (Table 1). Figure 1 shows an example of the approximation by the theoretical model. In most cases, less difference could be seen between Type I and Type II for lag greater than 0.75 [sec] and update interval under 200 [msec] regardless of Markov order or severity of entity change. On the basis of these results, we can say that prediction error can be estimated by only using the average of absolute value of velocity and acceleration independently of the pattern of the entity change. All in the above results reveal that the proposed theoretical model approximates well numerical simulations of degradation in information quality due to lag and update interval regardless of case.

<table>
<thead>
<tr>
<th>lag [sec]</th>
<th>0.25</th>
<th>0.5</th>
<th>0.75</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>update interval [msec]</td>
<td>50</td>
<td>100</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>Deviation [%]</td>
<td>4.9</td>
<td>9.9</td>
<td>19.6</td>
<td>46.9</td>
</tr>
<tr>
<td>Maximum value [%]</td>
<td>2.5</td>
<td>6.4</td>
<td>10.5</td>
<td>26.5</td>
</tr>
</tbody>
</table>

Table 1. Numerical simulation vs theoretical model

<table>
<thead>
<tr>
<th>Type</th>
<th>Order</th>
<th>Average [%]</th>
<th>Standard deviation [%]</th>
<th>Maximum value [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>2.9</td>
<td>2.9</td>
<td>9.4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.8</td>
<td>3.6</td>
<td>16.1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.9</td>
<td>3.8</td>
<td>17.1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.1</td>
<td>3.1</td>
<td>12.5</td>
</tr>
<tr>
<td>II</td>
<td>1</td>
<td>2.9</td>
<td>3.1</td>
<td>11.3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.8</td>
<td>3.7</td>
<td>13.6</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.6</td>
<td>3.6</td>
<td>13.8</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.3</td>
<td>3.3</td>
<td>13.3</td>
</tr>
</tbody>
</table>