Dynamic data Path Prediction in Network Virtual Environment

You-Sun Jeoung, Sang-Dong Rǎ, *Member, KIMICS*

This study was supported (in part) by research funds from Chosun University, 2007

***Abstract*—**This research studies real time interaction and dynamic data shared through 3D scenes in virtual network environments. In a distributed virtual environment of client-server structure, consistency is maintained by the static information exchange; as jerks occur by packet delay when updating messages of dynamic data exchanges are broadcasted disorderly, the network bottleneck is reduced by predicting the movement path by using the Dead-reckoning algorithm. In Dead-reckoning path prediction, the error between the estimated and the actual static values which is over the threshold based on the shared object location requires interpolation and multicasting of the previous location by the ESPDU of DIS. The shared dynamic data of the 3D virtual environment is implementation using the VRML.

***Index Terms*—**net-VE, Dead-reckoning, Consistency, VRTP, Multicast.

I. INTRODUCTION

Net-VE(Network Virtual Environment)[1][2] is a system that connection the distributed network to the virtual reality technology and offers 3D space to cooperate distributed multi-users interaction through real-time networking. In the Net-VE system, different technologies become organic within a single system. Additionally, synchronization resulting from time delay requires technologies regarding collision check at a natural reality, consist and resolution.

Consistency at a distributed virtual environment[3] of the client-server structure is continued by continuous exchange of static information among distributed clients. The cycles transfer of static information brings traffic overhead of network. As the current static transferring value, does not arrive at the time, without packet delay time, a tradeoff between the scope setting permitting errors and broadband for updates is needed during a cycle interval of the synchronization. There are a space partition method and a Dead-reckoning method[4] to minimize the receiving traffic of the dynamic data and reduce the overall broadband of the virtual space. The space partition method classifies the virtual environment into several small region and reduces the overall message traffic by limiting the region that transfers the static message. However, during a cycle interval of the synchronization, it needs to update static information in each frame for the real-time rendering and the scope setting to which limit error is acceptable, which brings traffic overhead.

When a new client is connecter to a 3D scene of the network virtual space, it interpolates the prior location with the Dead-reckoning path prediction algorithm of DIS(Distributed Interactive Simulation) to continue consistency and presentation the dynamic data sharing scene of the 3D virtual space.

II. DYNAMIC DATA SYSTEM OF NET-VE

Although an static information requires an update of more than 30 frames per second for consistency and uninterrupted client scene sharing, there originates a tradeoff by continuing an proper transfer rate because of network delay and limit to the broadband and the number of users acceptable for the server capacity.

To transfer the real-time dynamic data in 3D virtual environments, a variable combination between the server and dynamic data is applicable by the ESPDU(Entity State Protocol Data Unit)[5] protocol of DIS using VRTP(Virtual Reality Transport Protocol)[6] which is the transfer protocol during a periodic time.

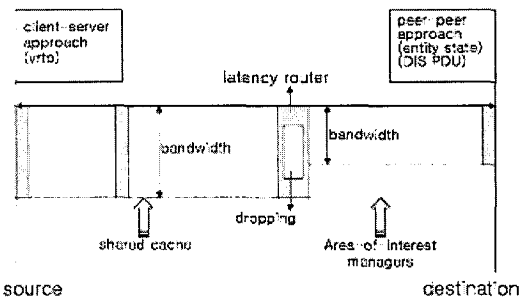


Fig. 1 Composite Multicasting Communication

Fig.1 shows the composite multicasting communication model of the client-server and the peer-peer server which limits virtual space information to within the server. Through communication with the server, the client accesses and shares information about the ESPDU shared protocol virtual environment. The peer-peer server reduces the traffic of the central server by

Manuscript received March 27,2007.

Sang-Dong Rǎ is Professor in Computer Engineering and Chosun University, Seoseok-dong, Dong-gu, Gwangju, 501-759,Republic of Korea,(e-mail :sdna@chosun.ac.kr)

You-Sun Jeoung, is Ph.D. course student in School of Computer Engineering,Chosun University, Seoseok-dong, Dong-gu, Gwangju, 501-759, Republic of Korea,

applying the multicast communication, which transfers information to participants in the virtual space. The entity interaction between the event and shared-cached http server originates at multiple locations via the peer-peer server by the client-server. Such intermediary solution supports the large web-based real-time multi participation 3D graphic by the VRTP structure of desktop computers.

III. DYNAMCI DATA PATH PREDICTION

A. Path Prediction using Dead-reckoning Algorithm

As a delay occurs in the network when a renewed message is frequently broad-casted[7] for events exchanged in real-time at Dead-reckoning, it predicts the location through a Dead-reckoning algorithm during a certain time period and adjusts the previous location and multi casts it with an ESPDU when the error goes over the limit value. Fig.2 shows the Dead-reckoning process that calculates the previous location, movement direction during the given time and current location and direction from the speed information and also calculates the location change assuming the progression locus during the cycle is composed of straight lines.

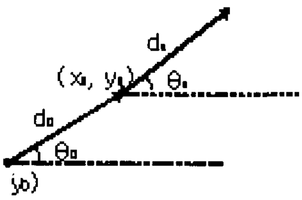


Fig. 2 Dead-reckoning Algorithm

Calculate the predicted changed location coordinate using the coordinate that arrived at a certain location (x_n, y_n) departing from the primary location (x_o, y_o) as the direction (θ_n) information of the entity and the location value $x_n > y_n > \theta_n$ that interpolates the previous location using the calculated value as in formula (1).

$$x_n = x_o + \sum_{i=0}^{n-1} d_i \cos\theta_i$$
$$y_n = y_o + \sum_{i=0}^{n-1} d_i \sin\theta_i$$
$$\theta_n = \sum_{i=0}^{n-1} w_i \qquad (1)$$

When you know the current location $x(t)$,the location at the time change $t+\Delta t$ after movement for a cycle interval from the time t at an average velocity can be calculated as in expression (2). Based on the location of the shared object in expression (2), the object location at the current time can be estimated. The previous location is interpolated if the error is over the predetermined threshold after reviewing the error between the estimated and the actual static values

$$x(t+\Delta t) = x(t) + \overset{\circ}{x}\Delta t \qquad \overset{\circ}{x} = V\cos\psi$$
$$y(t+\Delta t) = y(t) + \overset{\circ}{y}\Delta t \qquad \overset{\circ}{y} = V\sin\psi \qquad (2)$$

V : average velocity at the time $[t, t+\Delta t]$

ψ : average direction angle at the time $[t, t+\Delta t]$

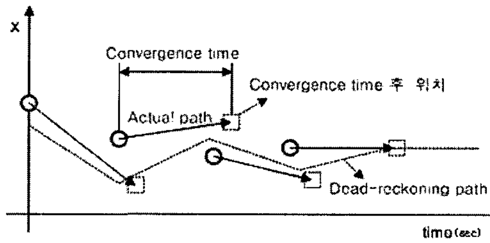


Fig. 3 Dead-reckoning Convergence

Fig. 3 shows the Dead-reckoning convergence process. We can get more precise estimates by increasing the estimate function interval of expression (3), but it results in more composite calculation. Therefore, we use 2[nd] level functions such as 1[st] differentials or 2[nd] differentials. We adjusts the threshold of the Dead-reckoning convergence number and control the static information transfer rate. The client that receives the discreteness dynamic data static information creates a continually shared static using the shared static location convergence expression (3).

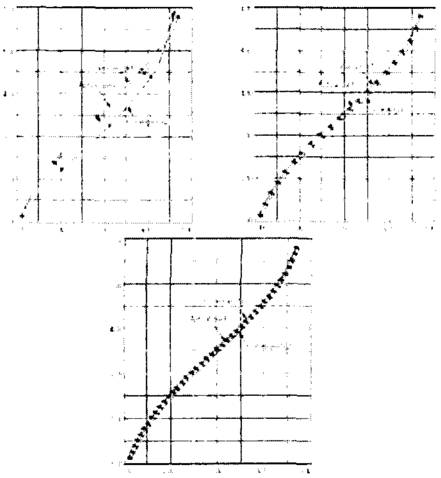
$$x(t) = x(t_o) + (t - t_o)\frac{dx(t)}{dt}\Big|_{t=t_0} + \frac{(t-t_0)^2}{2}\frac{d^2x(t)}{dt^2}\Big|_{t=t_0} + \cdots \qquad (3)$$

Table 1 Actual and Conversion Position to be Dead-reckoning Interval

Position DR Interval		Actual	Conversion.1	ERR	Conversion.2	ERR
0.50	x	4.62	4.52	-0.10	4.55	-0.07
	y	5.64	5.68	+0.04	5.71	+0.08
0.10	x	4.62	4.61	-0.01	4.62	-0.00
	y	5.64	5.64	+0.00	5.64	+0.00
0.05	x	4.62	4.63	-0.00	4.62	-0.00
	y	5.64	5.64	+0.00	5.64	+0.00

Table 1. shows the measurement of the actual location, the convergence location and the estimated location error when the path from the initial value of the dynamic data by the Dead-reckoning convergence width, (x, y, θ) = (1.5, 1.8, 70.0°), to the value, (x_n, y_n, θ_n) = (4.62, 5.64, 70.0°), is set to the velocity 2.4, the acceleration 0, the time stamp 2.0, and the DR Interval 0.50, 0.10, 0.05. When the actual location in Table 2 and the location prediction error by convergence width are measured at the point (x_n, y_n, θ_n)=(4.62, 5.64, 70.0°), the estimated error rate gets smaller and it becomes possible to predict the location which is closer to the actual path when the

Dead-reckoning convergence width is adjusted between 0.05 and 0.5, as shown in Fig. 4. As the location prediction interpolation error is 0 or -0.01 in (b) DR Interval = 0.10 and(c) DR Interval = 0.05, the dynamic data movement is not sensed at the client rendering. Although real-time rendering is more possible as the consistency is higher, it is possible to send the location change information of the shared object to the other clients and continue an proper transfer rate when in 0.10, because the server function and frequent updates cause network broadband width delays.



(a) DR Interval=0.50 (b) DR Interval=0.10
(c) DR Interval=0.05

Fig. 4 Position Prediction of Dead-reckoning Convergence

The location interpolation includes check of the error between the estimated and the real values and interpolation of the previous location when the error is over the predetermined threshold of the Dead-reckoning convergence. If the threshold is small, the average transfer rate and broadband width get heighten even though the error of the shared static gets smaller. P_{t0} And V^r mean the ESPDU location and velocity, respectively. Expression(4) is for interpolating the previous location of the entity using the initial location value t_0 by the time stamp velocity at the linear block d_n and the location estimate t_1.

$$P_{t1} + P_{t0} + V^r(t_1 - t_0) \tag{4}$$

The abstract node of the location interpolation P_{t1} in expression (4) is Interpolation. When the scope of the individual node $(-\infty,+\infty)$ is classified into small blocks, the interpolation node, which is the linear interpolation function $f(t)$ fraction field, is that detailed region n key one, key $t_0, t_1, t_2, \cdots, t_{n-1}$ it interpolation node is $(-\infty, t_n), (t_n, t_1), (t_1, t_2), \cdots (t_{n-1}, +\infty)$ for each value, and as each value is $v_0, v_1, v_2, \cdots, v_{n-1}$, the interpolation function $f(t)$ is defined as in expression (5).

$$\begin{aligned} f(t) &= v_0, \quad if \quad t \le t_0 \\ &= v_{n-1}, \quad if \quad t \le t_{n-1} \\ &= Iinterf(t, v_i, v_{i+1}) \quad if \quad t_i \le t_{i+1} \end{aligned} \tag{5}$$

where linterp (t, x, y) means the interpolator and i represents $\{0, 1, \cdots, n-2\}$.

The Interpolation Node is as shown in Fig.5. For the set fraction input Only field, the key field accepts the SFFloat event by the key time value, calculates the interpolation node function using expression (3), and creates the value changed output event set by the same timestamp as that of set fraction event, and displays it on the screen.

```
InterpolatorNode : ChildNode {
  SFFloat (∞,∞)
  MFFloat (-∞,∞)
  MF<type>
  [S|M]F<type>
}
```

Fig. 5 Position Interpolation

B. Dead-reckoning Packet Transfer

To display the multi participant network virtual environment, it is required to show the other participants that are currently connected in. Therefore, the DIS node interpolates and multicasts the previous location to ESPDU through the UDP socket where multi bandwidth is possible, when the information data error goes over the preset threshold.

```
main(){
  Event event1;
  dataGramSocket socket1;
  EntityStatePDU espdu1;
  agent agent_A(initPosition);
  int i;
  espdu1.initializeWithPosition(myAgentPosition())
;
  socket1.send(espdu1.convertToRawPacket());
  lastStateSent = agent_A
  lastTimeSent = ctime();
  while(1){ //Update my agent based on event
  agent_A.calcNewPosition(Event.read());//Calculat
  e Dead Reckoned Position
  agent_A.setDRposition(lastStateSent.position()+la
  stStateSent.velocity() * (ctime() - lastTimeSent);
  //Only send an update if DR threshold exceeded
  if(abs(agent_A.position()-
   lastStateSent.position()>
   thresh){socket1.send(espdu1.convertToRawPack
   et());
   lastStateSent = agent_A lastTimeSent = ctime();
   }}}
```

Fig. 6 Dead-reckoning Information Sending

As in Fig. 6, the dynamic data packet ESPDU examines the new packet from socket 1 to receive the Dead-reckoning algorithm information at the server. In order to update the measurement information by the path prediction algorithm, the convert FromRawPacket is passed and the server stores the location, the velocity, and the differentiator information of the object on the ESPDU packet in the vehicle table. The client then transfers the new PDU resulting from the last static information threshold. As the threshold of the error between the mobile path prediction value and the actual value, the sentence 'While' calculates the Dead-reckoning location value.

Fig. 7 is the 3D scene with the output through the client rendering engine at the network virtual space. The actual path of the dynamic data agent A is 'Actual Path', and as the Dead-reckoning estimate location path is a 'DR path' and the dynamic data moves suddenly when the user who received the shared static updates the information, it does not change right away to a client cache value, but moves to the 'Interpolation path' by the convergence interval.

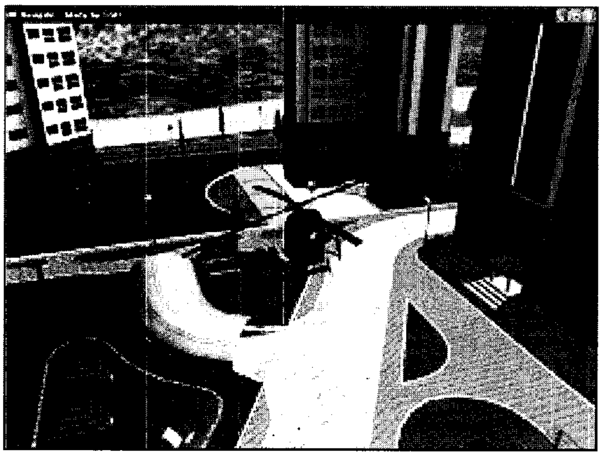


Fig. 7 Dead-reckoning Apply of 3D Graphics Scene

In future research, we can apply the evolution operation to dynamic data with specifically designed behavior and create a new behavioral norm, which will enable more realistic interaction.

IV. CONCLUSION

The dynamic data whose path was predicted by a Dead-reckoning algorithm interpolates the previous location with an interpolation node(Interpolation), transfers the shared object static information, and continues consistency with other clients.

At the network 3D virtual space, the movement path was predicted using the Dead-reckoning algorithm at the client buffer because the · congested broadcast by interaction and static information caused network delay and jerks. The error between the estimated and the actual static values, which is more than the threshold based on the shared object location, required interpolation() of the prior location using the Dead-reckoning estimate function and multicasting of the ESPDU packet of the DIS.

REFERENCES

[1] Singhal, S. and Zyda, M., 1999. Networked Virtual Environments: Design and Implementation, ACM Press [ISBN 0-201-32557-8].

[2] Bouras, C., Triantafillou, V. and Tsiatsos, T., 2001. Aspects of collaborative learning environment using distributed virtual environments In Proceedings of ED-MEDIA, Tampere, Finland, June25-30 pp.173-178.

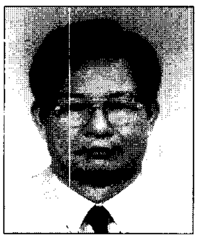
[3] Bouras,C., Psaltoulis, D., Psaroudis, C. and Tsiatsos, T., 2003. Multi user layer in the EVE distributed virtual reality platform In Proceedings of Fifth International Workshop on Multimedia Networks Systems and Applications (MNSA 2003) Providence, Rhode Island, USA, May 19-22 pp.602-607.

[4] W. Cai, F.B.S. Lee, L. Chen, An auto-adaptive Dead-reckoning algorithm for distributed interactive simula-tion, in: Proceedings of the Thirteenth Workshop on Parallel and Distributed Simulation, 1999, pp.82-89.

[5] "IEEE standard for Information Technology-Protocols for Distributed Simulation Application : Entity Information and Interaction." IEEE standard 1278~1993, New York: Ieee Computer Society, 1993.

[6] D. Brutzman, M. zyda, k. watsen, and M. Macedonia. "virtual reality transfer protocol(VRTP) Design Rotionale." Proceedings of the IEEE Sixth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '97), Distributed System Aspects of Sharing a Virtual Reality Workshop, IEEE Computer Society, Cambridge, Massachusetts, 179-186, June 1977.

[7] B. Hofman Wellenhof, H. Lichtenegger, and J. Collins, "GPS Theory and Practice", 4th, springer Wien New York, 1997.



Sang-Dong Rǎ
was born in Naju. South Korea, on February 9, 1945. He received the B.E degree in electrical engineering from the University of Chosun in 1968, the M.E degree in electrical engineering from the University of Gunkook in 1980, and the Ph. D. degree in electrical engineering from the University of Wonkwang in 1995.

He joined the Department of Computer Engineering, Chosun university, in 1983 and became an Assistant Professor, Associate Professor in 1985 and 1990, respectively. Since 1994, he has been a Professor in the Division of Computer Engineering at Chosun University. Currently, he is a chairman of Department of Computer Engineering at Chosun university. During 1995-1996,

2001-2002 He was a Visiting Scientist in the Department of Electrical and Computer Engineering at the University of California, Irvine. He is a member of System H/W Development. Ministry of Trade and Industry. His current research interests lie in the areas of adaptive signal processing, information theory. TCP IP Network, spread spectrum systems and various kinds of communication systems. He is a member of ISS of Korea, ICS of Korea. AS of Korea and ITE Korea. He is also a former member of the IEEE communication Society Board of Governors.

You- Sun Jeoung
was born in Jeonju, south Korea, November 28, 1976. She received the B.E degree in Fashion Design from the University of Wongwang in 1999, the M.E. degree in computer engineering from the University of Chosun in 2005. Since Febuary 2007, she has been Ph.D. student in Computer Communication Lab, Chosun University, Gwangju, Republic of Korea.