네트워크 가상환경에서 경로예측에 의한 동적 데이터 공유

정회원 송 선 희*, 나 상 동**

Path Prediction-based Dynamic Data Sharing in Network Virtual Environment

Sun-Hee Song*, Sang-Dong Ra** *Regular Members*

요 약

네트워크 가상 환경에서 다중 참여자 일관성 유지와 3D 장면의 동적데이터 공유를 연구한다. 클라이언트-서버 구조의 분산 가상환경에서 일관성은 상태정보 교환으로 유지되며, 교환되는 동적데이터의 갱신메시지가 자주 브로드캐스트 되면 패킷 지연에 의한 jerk를 일으키므로 데드레커닝 알고리즘을 이용한 이동경로를 예측하여 네트워크 병목을 줄인다. 동적데이터 경로예측은 데드레커닝 수렴간격의 타당성을 실험하여 공유객체 위치를 근거로 예측값과 실제 상태값과의 오차가 한계값 이상일 때 이전 위치를 보간하고, VRML EAI를 이용하여 3D 가상공간의 동적 데이터 공유를 구현한다.

Key Words : Net-VE, Dead-reckoning, Consistency, VRTP, Multicast

ABSTRACT

This research studies multi participant consistency 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 frequence, the network bottleneck is reduced by predicting the movement path by using the Dead-reckoning algorithm. In Dynamic data path prediction, the tests the location prediction error between Dead-reckoning convergence interval and error of prediction and actual condition one time above threshold it interpolates a previously location. The shared dynamic data of the 3D virtual environment is implementation using the VRML EAI.

Ⅰ. 서 론

네트워크 가상환경(net-VE/Network Virtual Environment)[1][2]은 분산된 다중 사용자들이 실시간 네트워크를 통하여 상호작용 할 수 있도록 가상현실 기술에 분산 네트워크를 접목한 시스템으로 3차원 공간을 제공한다. Net-VE 시스템은 네트워크 대역폭, 공정성, 분산 상호작용, 동기화, 자원관리, 확장성 등 분산네트워크 기술과 여러 기술이 하나의 시스템 내에서 유기적이다. 또한 시간지연에 따른 동기화 작업으로 자연스런 현실감에서 충돌체크, 일치성, 해상도 등의 기술이 필요하다.

클라이언트-서버 구조의 분산 가상환경[3]에서 일관성은 분산된 클라이언트들과 상태정보를 끊임없이 교환함으로써 유지되며, 주기적인 상태정보 전송은 네트워크의 트래픽 오버헤드를 가져온다. 네트워크 참여자들이 서로의 상태를 정확하게 아는 방법은 한 프레임마다 핸드쉐이킹으로 패킷을 전달하는 방

* 동신대학교 디지털콘텐츠협동연구센터 (shsong@dsu.ac.kr), ** 조선대학교 전자정보공과대학 컴퓨터공학과
논문번호 : KICS2006-06-273, 접수일자 : 2006년 6월 15일, 최종논문접수일자 : 2006년 10월 13일

법이나 동기화 부담이 크고 속도가 떨어진다. 패킷 지연이 없어도 현재의 상태를 전송하는 x_0의 값은 t_0시간에 도착하지 않으므로 동기화의 일정 주기동안 오차가 허용될 수 있는 범위설정과 업데이트를 위한 대역폭과의 절충을 해야 한다. 동적 데이터의 수신 트래픽을 최소화 하고 가상공간에서의 전체 대역폭을 줄이기 위한 방법으로는 공간분할 기법과 데드레커닝(Dead-reckoning)[4] 기법으로, 공간분할 기법은 가상환경에서 여러 개의 작은 부분 영역으로 분할하여 상태메시지가 전파되어야 하는 영역을 제한하여 전체적인 메시지트래픽을 줄이는 방법이고, 데드레커닝 기법은 동적데이터 이동경로의 변화정도에서 예측값과 실제값의 차이가 임계값을 넘어설 때만 정보를 전송한다. 하지만, 동기화의 일정 주기동안 오차가 허용될 수 있는 범위설정 문제와 실시간 렌더링을 하기 위해서 매 프레임마다 상태 정보를 갱신해야 하므로 트래픽 오버헤드를 일으킨다.

본 논문에서 네트워크 시스템은 클라이언트-서버와 개인-개인 서버를 경유하는 멀티캐스트 통신으로 분산된 다중 참여 동적 데이터를 처리하는 역할을 기준으로 메시지 서버와 응용서버로 구성하여 실시간 데이터는 동적데이터 서버로, 비실시간 데이터는 상태 데이터 서버로 구성해 서비스 처리 부하를 나눈다. 또한, 네트워크 가상공간의 3D 장면에 새로운 클라이언트가 연결했을 때 일관성을 유지하기 위해 DIS(Distributed Interactive Simulation)의 데드레커닝 경로예측 알고리즘으로 이전 위치를 보간하여 3D 가상공간의 동적 데이터 공유 장면을 보인다.

Ⅱ. 네트워크 가상환경

네트워크 가상환경에 참여한 클라이언트는 오브젝트를 자유로이 움직이고 사용할 수 있어야 하며, 변경된 정보를 다른 클라이언트들에게 즉시 알려 일관성을 유지한다. 일관성과 끊김 없는 클라이언트 장면 공유를 위해서는 초당 30프레임 이상 상태정보를 업데이트 하여야 하지만 서버의 성능에 따라 동시 접속자 인원이 증가하면 기하급수적으로 성능이 저하된다. 클라이언트-서버 구조는 상태정보를 서버에서 관리하므로 일관성 유지가 용이하다는 장점이 있으나 네트워크에 과중한 부담을 주므로 클라이언트-서버, 개인-개인 구조의 복합적 멀티캐스트 통신모델을 설계하여 중앙서버의 트래픽을 줄인다. 또한 네트워크 대역폭, 서버에서 수용 가능한 사용자 수 제한의 적절한 전송율 유지 절충을 위해 위치예측 에러 한계값으로 데드레커닝 수렴간격을 구하여 비동기화 cycleInterval을 결정한다.

2.1 동적 데이터 시스템

일관성 유지를 위한 동적데이터 시스템은 역할을 기준으로 메시지 서버와 응용 서버 로 나누어 클라이언트-서버 시스템으로 그림 2.1과 같이 구성한다. 응용 서버 데이터 종류를 실시간 변경되지 않는 상태 데이터와 실시간 변경되어 즉시 정보를 공유하는 동적 데이터, 그리고 두 가지 특성을 동시에 갖는 데이터로 나눈다. 클라이언트 인증, 3D 엔진, 동적 데이터 같은 신뢰도가 중요한 패킷은 TCP(Transfer Control Protocol) 연결을 하고 속도가 중요한 그래픽 리소스, 상태 데이터 등은 UDP(User Datagram Protocol), 일관성 유지에 필요한 공유데이터는 UDP-ACK(UDP Acknowledge)로 구성한다. 서버 시스템의 메시지 서버는 새로운 상태정보를 지원하여 ConnectionServer 시스템에 연결된 클라이언트 인식을 위한 연결과 인증 후 새로운 상태정보를 다른 서버들에게 통지한다. InitServer는 초기화와 가상 세계의 마지막 상태정보를 유지하여 새롭게 추가된 클라이언트에게 현재 상태를 전송하며, VRMLServer는 공유를 위한 이동객체 엔티티 위치와 방향의 갱신 메시지들을 보낸다. 또한 응용 서버는 각 시스템으로부터 connected/disconnected 상태정보를 인식하기 위해 ConnectionServer와 연결한다. 실시간 지원 데이터를 응용서버의 동적 데이터 서버에서 관리하고, 맵의 논리적 구성 정보들과 그래픽 리소스들은 상태 데이터 서버에서 관리하여 중앙 서버의 서비스 처리 부하를 배분한다.

클라이언트에서는 공유 상태를 유지하기 위한 동적데이터와 상태데이터를 클라이언트 리소스로 할당하여 이벤트 순서에 따른 정보테이블 참조로 다중 사용자 정보를 갱신한다.

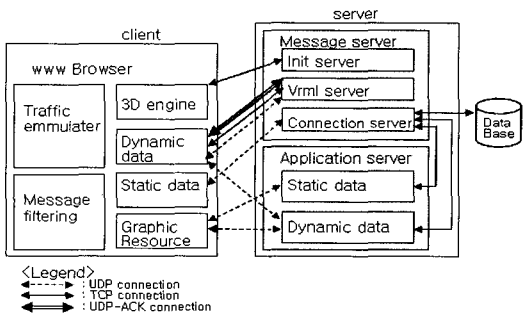


그림 2.1 동적 데이터 시스템 동작

2.2 동적 데이터 통신

3차원 가상환경의 동적데이터 실시간 전송을 위하여 일정 타임간 전송프로토콜인 VRTP(Virtual Reality Transport Protocol)[5]를 이용하여 DIS의 ESPDU 프로토콜에 의해 서버와 동적 데이터 변화의 가변적 결합을 한다.

그림 2.2는 클라이언트-서버, 개인-개인 구조의 복합적 멀티캐스트 통신 모델로 가상공간에 대한 정보는 서버에서 관리하고 클라이언트는 서버와의 통신을 통해 ESPDU 프로토콜로 공유된 가상환경에 접근하여 통신한다. 개인-개인 서버는 정보를 가상공간 참여자에게 전달하는 멀티캐스트 통신 방식을 도입하여 중앙서버의 트래픽을 줄인다. 또한 공유-캐시 http 서버와 이벤트의 엔티티 상호작용은 클라이언트-서버를 통해 개인-개인 서버를 경유한 다중 위치에서 발생하며, 이러한 중간자적 솔루션은 데스크탑 vrtp 구조에 의해 대규모 웹기반의 실시간 다중 참여 3D 그래픽을 지원한다.

네트워크 가상 환경의 동작 프로토콜은 DIS 표준으로 27 프로토콜 데이터 단위(PDU)의 패킷 형식이다. IEEE DIS 프로토콜은 다중참여 환경에서 강건하고 효과적인 객체-이벤트 구조, 자율적 분산 시뮬레이션 노드, 데드레커닝을 위한 예측 알고리즘의 구성요소를 가진다. 객체들의 위치, 방향, 속도 변화를 나타내기 위해 ESPDU(Entity State Protocol Data Unit)로 확장되어 다수의 클라이언트들 사이에 물리적으로 형성된 모델정보를 멀티캐스트 한다.

표 1.1은 DIS 필드 정보로 헤더와 행위정보, 네트워크 구성 정보가 포함되며, 동적데이터 노드의 위치, 방향, 속도 정보가 바뀌면 ESPDU를 생성하고 월드 좌표계(x, y, z)를 (x, -a, y) DIS 좌표계로 변환하여 다른 노드들에게 PDU를 전송한다. 위치 계산에 필요한 선형 속도 linearVelocity와 선형 가속도 linearAcceleration 필드 정보에 의해 경로 예측 알고리즘 Dead-reckoning 필드를 적용한다. Entity ID는 캐쉬에서 새로운 상태정보를 검색하여 갱신

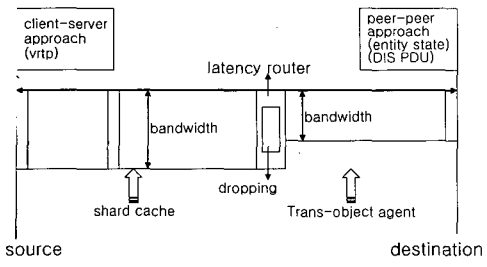


그림 2.2 복합적 멀티캐스트 통신

표 1.1 DIS entity state PDU 필드

Field Size (Bytes)	Entity state PDU field	Contents
12	PDU header	protocol version, PDU type Padding, time stamp, length
6	Entity ID	site, application, entity
1	Force ID	-
1	Number of articulation	N
8	Parameters	entity kind, domain,
	Entity type	country, category
8		same as above
12	Alternative entity	X, Y, Z(32-bit components)
24	type	X, Y, Z(64-bit components)
12	Linear velocity	H, P, R(32-bit components)
4	Location	-
40	Orientation	algorithm, other parameters,
	Appearance	linear acceleration, angular
	Dead-reckoning	velocity
12	Parameters	-
4		32 boolean fields
n×16	Entity markings	change, ID, parameter type,
	Capabilities	value
	Articulation parameters	

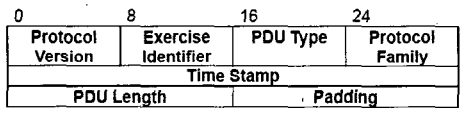


그림 2.3 DIS entity state PDU 헤더

정보를 화면을 재구성하며, 각 클라이언트에게 새로운 정보를 전송한다.

그림 2.3은 DIS entity state PDU[6] 헤더 정보이며, 그림 2.1에서 제시한 것처럼 클라이언트 동적데이터는 다중 대역이 가능한 UDP 소켓 통신으로 3D 장면 내 객체 위치와 방향을 바꾸고, 데드레커닝에 의한 새로운 객체가 등록되면 EspduTransformPDU Type으로 정보를 갱신하여 이벤트 종류에 따라 Collision, Detonate, Entity State, Fire, Receiver, Signal, Transmitter 프로토콜 메시지 유형을 식별하여, 지역시스템 시간 참조로 timestamp 메시지 도착시간(SFTime)을 인식한다.

분산된 다중 참여 동적데이터를 실시간 처리하기 위하여 클라이언트에서 데드레커닝 알고리즘으로 경로 예측하여 Dead-reckoning 필드 정보를 캐쉬 리소스로 할당한다. 경로 예측된 Dead-reckoning 필드 정보는 DIS ESPDU로 멀티캐스트하여 중앙서버의 트래픽을 줄인다.

Ⅲ. 데드레커닝에 의한 동적데이터 공유

3.1 데드레커닝 알고리즘의 경로예측 성능평가

데드레커닝 경로예측 알고리즘은 각 클라이언트에서 상태정보를 급속하게 이동하면 클라이언트 디스플레이에 jerk가 발생하므로 엔티티 실제 상태값과 예측 계산된 상태 값을 비교하여 오차가 미리 설정된 한계값을 초과하면 이전 위치를 보정한다. 데드레커닝을 구현하기 위해서는 사용자 입력 인터페이스에 따라 알고리즘 구성이 달라지게 되는데, 마우스로 예정 지점을 미리 포인트하여 길찾기 알고리즘으로 찾아가는 방식과 장애물 피하는 패턴과 사용자가 한번 앞으로 가면 계속 앞으로 갈 확률이 높은 경우의 입력예측 방식이다.

이동패킷 경로예측은 입력예측 방식을 가정하여 그림 3.1과 같이 주기 동안 진행 궤적을 부분적 직선 구간으로 나누어 이전 위치와 속도 정보로부터 현재 위치 변화를[7] 계산한다.

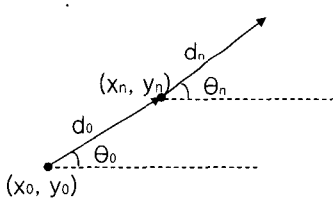


그림 3.1 데드레커닝 알고리즘

초기위치 (x_0, y_0)에서 임의의 위치(x_n, y_n)에 도달했을 때 엔티티 방향(θ_n) 정보, 변화된 위치 예측좌표를 구하는 식은 (1)과 같다.

$$x_n = x_0 + \sum_{i=0}^{n-1} d_i \cos\theta_i$$
$$y_n = y_0 + \sum_{i=0}^{n-1} d_i \sin\theta_i \qquad (1)$$
$$\theta_n = \sum_{i=0}^{n-1} \omega_i$$

현재위치 $x(t)$를 알고 있는 경우 시간 t에서 일정 주기동안 평균속도로 변화된 시간 $t+\Delta t$ 위치는 식 (2)로 구한다. 식 (2)의 공유객체 위치를 근거로 현재 시간의 객체위치를 예측하며, 예측된 상태값과 실제 상태 값과의 오차를 검사하여 오차가 미리 정해진 한계값 이상이면 이전 위치를 보간한다.

$$x(t+\Delta t) = x(t) + \dot{x}\Delta t \qquad \dot{x} = V\cos\Psi \qquad (2)$$
$$y(t+\Delta t) = y(t) + \dot{y}\Delta t \qquad \dot{y} = V\sin\Psi$$

V : 시간 $[t, t+\Delta t]$에서의 평균 속도
Ψ : 시간 $[t, t+\Delta t]$에서의 평균 방향각

대역폭과 적절한 전송율을 유지하기 위하여 그림 3.2는 데드레커닝 수렴과정으로 식 (3)의 예측함수 항을 늘릴수록 보다 정확한 예측이 가능하지만 계산량이 많아지므로 1차 미분한 함수나 2차 미분한 2차 함수를 사용하여 클라이언트의 불연속적인 동적데이터 상태정보를 공유상태 위치 수렴식 (3)으로 연속적 공유상태로 만든다.

데드레커닝 위치예측 에러값에 따른 적절한 수렴 횟수의 한계값을 조정하기 위해서 데드레커닝 간격에 의한 실제위치와 수렴 위치를 표 1.2와 같이 실험하였다. 데드레커닝 수렴 간격에 의한 동적데이터 초기위치 값 (x, y, θ)=(1.5, 1.8, 70.0°)에서 (x_n, y_n, θ_n)=(4.62, 5.64, 70.0°)까지의 경로를 속도 2.4, 가속도 0, timestamp 2.0, DR Interval 0.75, 0.50, 0.10, 0.05로 했을 때 실제위치와 수렴 위치, 예측위치 에러를 측정한 것이다.

표 1.2의 실제위치와 수렴간격에 의한 위치예측 에러를 (x_n, y_n, θ_n)=(4.62, 5.64, 70.0°) 지점에서 측정한 결과 그림 3.3과 같이 데드레커닝 수렴 간격을 0.5로 주었을 때보다 0.05로 주었을 때 위치 예측

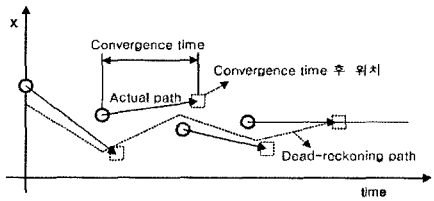
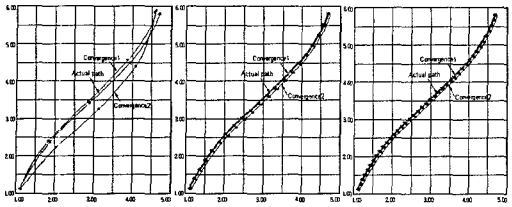


그림 3.2 데드레커닝 수렴

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

표 1.2 데드레커닝 간격에 의한 실제위치와 수렴 위치

Position DR Interval	Actual		Conversion.1	ERR	Conversion.2	ERR
0.75	x	4.62	4.49	-0.13	4.41	-0.21
	y	5.68	5.72	+0.04	5.90	+0.22
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



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

그림 3.3 데드레커닝 수렴간격에 의한 위치예측

에러율이 작고, 실제경로에 근접한 위치예측이 가능하다. (b) DR Interval = 0.10일 때와 (c) DR Interval = 0.05일 때 위치예측 보간 에러는 0이거나 -0.01이므로 클라이언트 렌더링에서 동적데이터 움직임을 느끼지 못한다. 일관성이 높을수록 실시간 렌더링이 가능하지만 서버 성능과 잦은 업데이트는 네트워크 대역폭 지연을 일으키므로 DR Interval=0.10일 때부터 위치예측 에러는 0로 나타나지만 잦은 업데이트는 네트워크 지연을 가져오므로 DR Interval=0.10일 때 대역폭과 업데이트간 절충관계를 충족시키므로 공유 객체 위치변화 상태정보를 다른 클라이언트들에게 보내어 적절한 전송율을 유지한다.

위치 보간은 예측 값과 실제 값과의 오차를 검사하여 오차가 미리 정해진 데드레커닝 수렴 한계값 이상이면 이전 위치를 보간하며, 한계값이 크면 공유상태 오차는 커지지만 상태정보 평균 전송률은 작고, 한계값이 작으면 공유상태 오차는 작아지지만 평균 전송률과 대역폭이 크다. P_{t0}와 V^{r}는 ESPDU 위치와 속도이며, 직선구간 d_n에서 지정시간(timestamp) 속도에 의한 t_o 초기 위치 값과 t_1 위치 예측으로 엔티티 이전 위치를 보간하는 식은 (4)와 같다.

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

식 (4)의 위치 보간 (P_{t1}) 추상노드는 Interpolation으로 개별 노드 $(-\infty,+\infty)$ 범위를 부분적 구간으로 나누면, 선형 보간 함수 $f(t)$ fraction 필드인 보간자 노드는 각각 임의의 값 t에 대해 세부영역 n개 key $t_0,t_1,t_2,....,t_{n-1}$은 $(-\infty,t_0),(t_0,t_1),(t_1,t_2),....,(t_{n-1},+\infty)$이고, n개 value는 $v_0,v_1,v_2,....,v_{n-1}$이므로 보간 함수 $f(t)$는 다음 식 (5)와 같이 정의한다.

<정의1>

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

여기서, $linterp(t,x,y)$는 보간자, i는 $\{0,1,...,n-2\}$이다.

동적데이터의 Entity ID는 캐쉬에서 새로운 상태데이터 정보를 서버에 요청하여 화면을 재구성하기 위해 데드레커닝 한계값으로 이전 위치 보간하며, 각 클라이언트에게 새로운 상태정보를 전송한다. 보간 노드 InterpolatorNode는 그림 3.4와 같으며 set_fraction inputOnly 필드는 key 필드는 키 시간값에 의해 SFFloat 이벤트를 수신하고 보간자 노드의 함수를 식 (3)에 의해 계산하여 set_fraction 이벤트와 동일한 타임스탬프를 가지고 설정된 유형의 value_changed 출력 이벤트를 생성하여 화면에 디스플레이 된다.

```
InterpolatorNode : ChildNode {
  SFFloat   [in]      set_fraction          (-∞,∞)
  MFFloat   [in,out]  key          [] (-∞,∞)
  MF<type>  [in,out]  keyValue     []
  [S|M]F<type>  [out]  value_changed  }
```

그림 3.4 위치보간

3.2 데드레커닝 패킷전송

다중참여자 네트워크 가상환경을 디스플레이 하기 위해서는 현재 접속되어 있는 다른 참여자 모습을 나타내야 하므로 그림 3.5와 같이 식 (4)를 사용해 DIS 노드의 다중대역이 가능한 UDP 소켓을 통하여 정보데이터 오차가 미리 설정된 한계값을 초과하면 이전 위치를 보간하고 ESPDU로 멀티캐스트 한다.

```
main(){
  Event event1;
  DataGramSocket socket1;
  EntityStatePDU espdu1;
  agent Entity_01(initPosition);
  int i;
espdu1.initializeWithPosition(myAgentPosition());
  socket1.send(espdu1.convertToRawPacket());
  lastStateSent = Entity_01;
  lastTimeSent = ctime();
  while(1){//Update my agent based on event
  Entity_01.calcNewPosition(Event.read());
   //Calculate Dead Reckoned Position

Entity_01.setDRposition(lastStateSent.position() +
  lastStateSent.velocity() * (ctime() - lastTimeSent);
   // Only send an update if DR threshold exceeded
  if (abs(Entity_01.position()-lastStateSent.position() >
   thresh){socket1.send(espdu1.convertToRawPacket());
   lastStateSent = Entity_01; lastTimeSent = ctime();
}}}
```

그림 3.5 데드레커닝 정보전송

그림 3.5는 서버에서 데드레커닝 알고리즘 정보를 수신하기 위해 동적 데이터 패킷 ESPDU는 socket1로부터 새로운 패킷을 검사한다. 경로예측 알고리즘에 의한 관측 정보를 갱신하기 위해 convertFromRawPacket 파싱되며, 서버는 ESPDU 패킷에 개체 위치, 속도, 식별자 정보를 vehiche 테이블에 저장하고, 클라이언트는 마지막 상태정보 한계값에 따른 새로운 PDU를 전송한다. 데드레커닝으로 업데이트된 후 이전 위치 값이 누적되면 예측 오차 값이 발생하므로 initializeWithPosition 함수로 초기화 하여 동기화 ctime 동안 마지막 위치 값을 멀티캐스트 한다. 또한 while문은 이동경로 예측 값과 실제 값 차이 임계값으로 데드레커닝 위치 값을 계산한다.

3.3 동적 데이터 공유

ESPDU 멀티캐스트 정보를 네트워크 가상공간에 접속되어 있는 다른 참여자들이 공유하기 위하여 그림 3.6 SVE file의 eventIn, eventOut, ROUTE 공유 노드로 새로운 클라이언트 오브젝트 상태 정보를 인식한다. SVE file은 모든 sNode와 ROUTE의 world를 포함하며, sNode 이벤트와 상태 공유 및 갱신을 위해 클라이언트에게 모든 SharedNode를 전송한다. 클라이언트가 3D 장면에 있는 VRML 노드 값을 변경하면, VRML sNode (EventIn or EventOut) 값들을 포함하여 각각 서버로부터 도착하는 동적 이벤트 메시지가 SharedNode에 적용된다. SharedNode는 매개변수 LastModified를 포함하고 적용되어진 마지막 이벤트 시퀀스 번호를 유지하여 새로운 참여자들에게 EventIn과 EventOut 이벤트를 알려 공유객체를 인식한다.

3D 가상공간의 동적데이터 장면 렌더링은 그림 3.7과 같이 메시지 전달, 객체 움직임, 공간 회전, 크기조정, 좌우 이동을 하여 카메라 시점이나 아바타 시점으로 인터렉티브하게 볼 수 있다. 다른 클라이언트들과 동적데이터 패킷 정보 공유를 위하여 표 1.2의 DR Interval에 따라 일정주기 'cycleInterval'을 1.0초로 하여 멀티캐스트하고 각 클라이언트는 1.0초에서 호스트까지의 지연을 뺀 시간을 기다린 후 위치를 보간 한다. 각 클라이언트가 갖는 객체 정보를 가상환경에 연결된 클라이언트들에게 멀티캐스트하여 위치 이동의 관측점을 유지한다. 시작 시간은 클라이언트 공통시간이고 이동이 시작되면 0으로 초기화한다. 또한 각 참여자들의 입력에 의한 객체 움직임, 정보전달은 공통시간(timestamp)으로 충돌감지와 일관성을 유지한다. 상태 데이터와 동적 데이터는 'cycleInterval' 동안 클라이언트에 적재되며, 동적 데이터는 적재된 정보데이터를 참조하여 'DeadReckoningComparisons' 노드로 최근 변화된 상태값을 비교하여 적용한다.

데드레커닝 알고리즘으로 경로예측한 동적데이터는 보간자 노드 Interpolation으로 이전위치를 보간하여 공유객체 상태정보를 전송하여 Agent_A 객체 상태로 다른 클라이언트들과 일관성을 유지한다. 그림 3.8은 네트워크 가상공간에서 클라이언트 그래픽 렌더링 엔진에 의해 화면에 출력되는 3D 장면이다. 동적데이터 실제경로는 Actual Path이고, 데드레커닝 예측위치 경로는 DR path로 공유상태를 받은 클라이언트가 새롭게 받은 정보로 바로 갱신하는 경우 동적데이터가 순간적으로 이동하게 되면 3D

```
DEF  sharedROUTEsCreator Script {
     eventIn        SFString
     sharedRouteCreateString
     eventIn        SFString
     sharedNodeRouteDeleteString
     eventOut       SFString    routeCrParam
     eventOut       SFString    routeNodeDelParam
     url   "javascript:
     function sharedRouteCreateString(value) {
     routeCrParam = value; }
     function sharedNodeRouteDeleteString(value) {
     routeNodeDelParam = value;          }" }
```

그림 3.6 공유된 객체 인식

```
<TimeSensor DEF='Clock' cycleInterval='1.0'
loop='true' pauseTime='0' isPaused='' resumeTime='0'
fraction_changed=''/>
<Script DEF='Movement'
url='"DeadReckoningComparisons/Project4.class"
<field name='set_time' type='SFFloat'
accessType='inputOnly'/>
<field name='position_changed' type='SFVec3f'
accessType='outputOnly'/>
<field name='text_changed' type='MFString'
accessType='outputOnly'/> </Script> <Transform
translation='-4.0 0.0 0.0'> <Shape> <Appearance>
</Appearance> <Text DEF='Agent_A'
string='"어서오세요" '/> </Shape> </Transform>
<ROUTE fromNode='Clock'
fromField='fraction_changed' toNode='Movement'
toField='set_time'/>
<ROUTE fromNode='Movement'
fromField='position_changed' toNode='Agent_A'
toField='translation'/>
<ROUTE fromNode='Movement'
fromField='text_changed' toNode='Agent_A'
toField='string'/>
```

그림 3.7 가상공간에서의 동적데이터

그림 3.8 3D 그래픽 장면의 데드레커닝 적용

장면은 끊김 현상을 보이므로 Agent_A는 클라이언트 캐쉬 값을 받아 바로 변경하지 않고 수렴간격에 의해 Iterpolation path 경로로 이동함으로써 자연스런 렌더링을 한다.

Ⅳ. 결 론

네트워크 가상 환경에서 일관성 유지와 사실감 있는 3D 그래픽을 제공하기 위해 데드레커닝 경로 예측 알고리즘으로 동적데이터 공유를 구현하였다.

네트워크 3D 가상공간에서 상호작용 및 상태정보 업데이트로 인해 빈번하게 브로드캐스트되면 네트워크 지연과 jerk를 일으키므로 각 클라이언트 버퍼에서 데드레커닝 알고리즘으로 이동 경로를 예측하였다. 데드레커닝 위치예측은 DR Interval=0.10일 때 위치예측 에러가 0으로 나타나 대역폭과 업데이트간 절충관계를 충족시키므로 공유객체 위치를 근거로 예측값과 실제 상태값과의 오차가 한계값 이상일 때 이전 위치를 위치보간(P_{t1}: InterpolatorNode)하여 ESPDU 패킷을 가상공간 참여자에게 멀티캐스트 통신하였다. 네트워크 3D 가상공간은 VRML EAI를 이용하여 SVE file, SharedNode, SFCreate, SFDelete Event 노드로 다중 참여자 동적데이터 공유를 구현하였다.

향후 연구로는 동적 데이터의 사전 설계된 행위에 진화연산을 적용해 새로운 행동양식을 창조해내면 더욱 사실감 있는 상호작용이 가능할 것이다.

참 고 문 헌

[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 simulation, in: Proceedings of the Thirteenth Workshop on Parallel and Distributed Simulation, 1999, pp.82-89.

[5] 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.

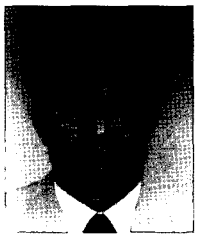
[6] "IEEE standard for Information Technology-Protocols for Distributed Simulation Application : Entity Information and Interaction." IEEE standard 1278~1993, New York: Ieee Computer Society, 1993.

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

송 선 희(Sun-Hee Song) 정회원

1988년 전남대학교 공학학사
2002년 호남대학교 공학석사
2004년 조선대학교 컴퓨터공학과 박사수료
현 재 동신대학교 디지털콘텐츠 협동연구센터
<관심분야> 실시간통신, net-VR, VR, 멀티미디어 등

나 상 동(Sang-Dong Ra) 정회원



1968년 조선대학교 전기공학과 졸업(공학사)

1980년 건국대학교 대학원 졸업 (공학석사)

1995년 원광대학교 대학원 졸업 (공학박사)

1995년~1996년, 2001년~2002년 Dept. of Electrical &Computer Eng. Univ. of California Irvine 연구교수.

1998년 조선대학교 전자계산소 소장 역임

1973년~현재 조선대학교 컴퓨터공학과 교수