



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년01월10일
(11) 등록번호 10-1221169
(24) 등록일자 2013년01월04일

- (51) 국제특허분류(Int. Cl.)
A63F 13/10 (2006.01) A63F 13/12 (2006.01)
- (21) 출원번호 10-2010-7004437
- (22) 출원일자(국제) 2008년09월16일
심사청구일자 2010년04월15일
- (85) 번역문제출일자 2010년02월26일
- (65) 공개번호 10-2010-0063703
- (43) 공개일자 2010년06월11일
- (86) 국제출원번호 PCT/EP2008/062307
- (87) 국제공개번호 WO 2009/037257
국제공개일자 2009년03월26일
- (30) 우선권주장
11/856,526 2007년09월17일 미국(US)
- (56) 선행기술조사문헌
US20030190951 A1
US20060094501 A1
US20080207327 A1

- (73) 특허권자
인터내셔널 비지네스 머신즈 코퍼레이션
미국 10504 뉴욕주 아몬크 뉴오차드 로드
- (72) 발명자
해밀턴 투, 릭, 알렌
미국 버지니아 22903 샬러츠빌 대리 로드 1532
모스코위츠, 폴
미국 뉴욕 10598 요크타운 하이츠 헌터브룩크 로
드 2015
(뒷면에 계속)
- (74) 대리인
허정훈

전체 청구항 수 : 총 8 항

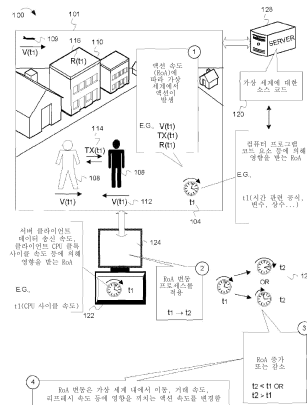
심사관 : 양태환

(54) 발명의 명칭 가상 세계에서의 액션 속도 수정

(57) 요약

가상 세계에서의 액션 발생에 따라 제1 속도를 규정하는 단계와, 가상 세계의 제1 영역에 대하여, 액션이 발생하는 속도를 제1 속도와 다른 제2 속도로 수정하는 단계를 포함하고, 이러한 수정 단계는 제1 영역에서의 액션들이 제2 속도에 따라 발생하게 하는, 방법을 제공한다.

대표도 - 도1



(72) 발명자

오'코넬, 브라이언, 마샬

미국 노스캐롤라이나 27519 캐리 민트 힐 드라이브
226

피코버, 클리포드, 아란

미국 뉴욕 10598 요크타운 하이츠 요크셔 레인 37

씨맨, 제임스, 웨슬리

미국 버지니아 22042 폴스 처치 마샬 스트리트
2759

워커, 키스, 레이몬드

미국 텍사스 78727 오스틴 킨더 패스 13412

특허청구의 범위

청구항 1

가상 세계의 영역을 선택 - 상기 영역에서 시간은 제1 시간 진행 속도(a first time progression rate)로 진행됨 - 하는 단계; 및

상기 영역에 대한 상기 제1 시간 진행 속도를 상기 제1 시간 진행 속도와는 상이한 제2 시간 진행 속도(a second time progression rate)로 수정하는 단계;를 포함하고,

상기 수정하는 단계는 상기 영역 내의 오브젝트들(objects)이 상기 제2 시간 진행 속도에 따라 동작하도록 상기 영역에서의 물리적 규칙들의 효과를 변경하는 단계를 포함하는,

가상 세계에서의 액션 속도를 제어하는 방법.

청구항 2

제 1 항에 있어서,

상기 가상 세계의 안정성에 영향을 미치는 이벤트를 검출하는 단계를 더 포함하고, 상기 수정하는 단계는 상기 검출하는 단계에 응답하는 것인,

가상 세계에서의 액션 속도를 제어하는 방법.

청구항 3

가상 세계 내의 영역 내에서 아바타가 제1 액티비티(activity)를 제1 동작 속도(a first rate of movement)로 수행함을 결정하는 단계;

상기 아바타와 관련된 클라이언트를 검출 - 상기 클라이언트는 데이터 처리 속도에 따라 상기 제1 동작속도로 상기 아바타의 출현(appearance)을 렌더링함 - 하는 단계;

상기 클라이언트에게 상기 데이터 처리 속도를 수정하게 하는 단계 - 상기 수정은 상기 아바타가 상기 영역 내에서 상기 제1 액티비티를 수행하는 속도를 상기 제1 동작속도로부터 제2 동작속도(a second rate of movement)로 늦추게 함 - ; 및

상기 아바타가 상기 제1 액티비티를 상기 제2 동작속도로 수행하는 동안 제2 액티비티를 상기 제1 동작속도로 수행하는 단계를 포함하는,

가상 세계에서의 액션 속도를 제어하는 방법.

청구항 4

통신 네트워크에 접속되어 가상 세계의 데이터를 처리하도록 구성된 기계;

상기 기계와 상기 통신 네트워크 중 임의의 하나에 접속되도록 구성된 시간 변동 장치(a temporal variation device);

상기 시간 변동 장치는

상기 가상 세계 내의 제1 영역을 결정 - 상기 가상 세계에서 시간은 제1 시간 진행 속도로 진행됨 - 하고, 상기 제1 영역에서의 상기 제1 시간 진행 속도를 상기 제1 시간 진행 속도와는 상이한 제2 시간 진행 속도로 수정 - 상기 수정은 액션들이 상기 가상 세계 내의 제2 영역에서 상기 제1 시간 진행 속도에 따라 발생하는 동안 상기 제1 영역에서의 액션들이 상기 제2 시간 진행 속도에 따라 발생하게 함 - 하도록 구성된 시간 변동 컨트롤러(a temporal variation controller)를 포함하고,

상기 제1 시간 진행 속도의 수정의 효과들을 분석하도록 구성된 시간 변동 효과 감시 장치(a temporal variation effects monitoring device); 및

상기 제1 시간 진행 속도의 상기 수정의 지시기(indicator)를 표시 - 상기 지시기는 상기 기계를 통해 볼 수 있

음 - 하도록 구성된 시간 변동 표시 컨트롤러(a temporal variation display controller);를 포함하는, 액션 속도 변동 시스템.

청구항 5

제 4 항에 있어서,
상기 수정을 트리거(trigger)하는 이벤트를 검출하도록 구성된 이벤트 검출기를 더 포함하는, 액션 속도 변동 시스템.

청구항 6

삭제

청구항 7

제 4 항에 있어서,
상기 기계는 데이터 처리 속도로 상기 가상 세계를 렌더링하도록 구성되고,
상기 시간 변동 장치는 상기 데이터 처리 속도를 수정함으로써 상기 제1 시간 진행 속도를 수정하도록 구성되는,
액션 속도 변동 시스템.

청구항 8

제 7 항에 있어서,
상기 데이터 처리 속도는 데이터 전송 속도, 데이터 조절 속도(data throttling rate) 및 컴퓨터 클럭 사이클 속도 중 임의의 하나를 포함하는,
액션 속도 변동 시스템.

청구항 9

삭제

청구항 10

컴퓨터 시스템 내에 로딩되어 실행될 때, 컴퓨터로 하여금 제 1 항 내지 제 3 항 중 어느 한 항에 기재된 방법의 단계들을 수행하게 하는 컴퓨터 프로그램 코드를 포함하는,
컴퓨터 판독가능 매체.

명세서

기술분야

[0001] 본 발명의 실시예들은 일반적으로 가상 세계, 시스템 및 네트워크에 관한 것으로서, 보다 상세하게는, 가상 세계에서의 액션 속도의 수정에 관한 것이다.

배경기술

[0002] 가상 세계 애플리케이션은 사람들이 가상 세계에서 어울리고 상호작용할 수 있게 한다. 가상 세계(VU)는 가상 세계의 거주자(resident)들이 아바타를 이용함으로써 횡단하고, 거주하고, 상호작용하기 위한 컴퓨터 기반 시뮬레이션 환경이다. 많은 VU는 3-D 그래픽과 랜드스케이프를 이용하여 표현되며, 거주자로 알려진 수천명의 사용자로 채워진다. VU에 대한 다른 용어들로는 "메타버스"(metaverse) 및 "3D 인터넷"이 있다.

발명의 내용

해결하려는 과제

[0003] 본 발명은 가상 세계에서 액션 속도를 의도적으로 수정하는 프로세스 및 장치를 제공하고자 한다.

과제의 해결 수단

[0004] 본 명세서에서는 가상 세계에서 액션 속도를 의도적으로 수정하는 프로세스 및 장치를 설명한다. 설명하는 장치들 중 하나는 액션 속도 변동 장치(rate of action variation device)이다. RoA 변동 장치는 가상 세계에서 특정한 속도에서 액션이 발생하는 영역을 결정할 수 있다. RoA 변동 장치는 그 영역에서 액션 속도를 빠르게 또는 느리게 수정할 수 있다. 그 영역에서 RoA를 수정함으로써, 액션이 발생하는 속도가 변경된다. RoA 변동 장치는 수정되지 않는 액션 속도로 그 영역의 경계를 넘는 영역을 유지할 수 있다. 따라서, RoA 변동 장치는 결정된 영역 및 결정된 영역 외부의 가상 세계에서 다른 영역들 사이에 액션 속도의 상대적 차이를 생성할 수 있다.

발명의 효과

[0005] 본 발명에 따른 프로세스 및 장치를 통해, 가상 세계에서 액션 속도를 의도적으로 수정할 수 있다.

도면의 간단한 설명

- [0006] 도 1은 본 발명의 바람직한 일 실시예에 따라 가상 세계 내에서 액션 속도의 변동의 일례를 도시한다.
- 도 2는 본 발명의 바람직한 일 실시예에 따라 운영 환경(200)에서의 액션 속도 변동 장치(202)의 일례를 도시한다.
- 도 3은 본 발명의 바람직한 일 실시예에 따라 액션 속도 변동 장치 아키텍처(300)의 일례를 도시한다.
- 도 4는 본 발명의 바람직한 일 실시예에 따라 운영 환경(400)에서의 액션 속도 변동 장치(402)의 일례를 도시한다.
- 도 5는 본 발명의 바람직한 일 실시예에 따라 가상 세계에서 상대적 액션 속도를 제어하는 것을 도시하는 흐름도(500)의 일례이다.
- 도 6은 본 발명의 바람직한 일 실시예에 따라 데이터 처리 속도를 수정함으로써 가상 세계에서 액션 속도를 수정하는 것을 도시하는 흐름도(600)의 일례이다.
- 도 7은 본 발명의 바람직한 일 실시예에 따라 프로그램 코드 요소를 수정함으로써 액션 속도를 수정하는 것을 예시하는 흐름도(700)의 일례이다.
- 도 8은 본 발명의 바람직한 일 실시예에 따라 가상 세계에서 액션 속도에 변동을 적용하는 것을 예시하는 흐름도(800)의 일례이다.
- 도 9는 본 발명의 바람직한 일 실시예에 따라 가상 세계에서 액션 속도에 변동을 적용하는 것을 예시하는 흐름도(900)의 일례이다.
- 도 10은 본 발명의 바람직한 일 실시예에 따라 네트워크(1000) 상의 액션 속도 변동 장치(1002)의 일례를 도시한다.
- 도 11은 본 발명의 바람직한 일 실시예에 따라 액션 속도 변동 장치 컴퓨터 시스템(1100)의 일례를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0007] 이하의 설명에서는, 본 발명의 실시예들의 기술들을 구체화하는 예시적인 시스템, 방법, 기술, 명령어 시퀀스 및 컴퓨터 프로그램 제품이 포함된다. 그러나, 본 발명의 실시예들은 이러한 특정 상세 없이 실시되어도 된다는 점을 이해할 것이다. 일부 경우에는, 본 발명이 모호해지지 않도록, 잘 알려져 있는 명령어 인스턴스, 프로토콜, 구조 및 기술을 상세히 도시하지 않았다.

[0008] 도입부

[0009] 가상 세계에서, 사용자들은 물리적 규칙들의 소정의 세트에 따라 아바타 및 기타 오브젝트를 제어할 수 있다. 이러한 규칙들은 가상 세계에서 아바타 및 오브젝트의 이동, 스크립트, 액션 및 기타 액티비티(activity)를 제

어하는 물리적 공식 및 변수를 적용한다. (시간 경과와 유사하게) 액션이 취해지는 속도는 그러한 소정의 규칙들 중 하나이다. 일반적으로, 액션은 가상 세계의 다른 부분들과 유사한 가상 세계의 일부분에서 속도로 취해지며 이에 따라 모든 오브젝트와 캐릭터가 일관된 속도로 액션을 경험하게 된다.

[0010] 그러나, 가상 세계의 영역들을 서로 다른 액션 속도에서 기능하게 함으로써, 다양한 사항들(예를 들어, 네트워크 및 사용 사항)이 다루어지도록 발생할 수 있다. 액션들이 수행되는 속도들의 이러한 차이는 본 명세서에서 더 간결하게 액션 속도(rate of action; RoA) 변동이라 칭한다. 가상 세계에서의 영역과 오브젝트 사이의 RoA 변동은 가상 세계에서 흔히 발생하는 유해 사용자, 불안정 및 기타 사항에 관리자가 응답하는 것을 도울 수 있다. 또한, RoA 변동은 사용자, 특히, 미숙한 사용자가 다른 상황에선 매우 빨리 제어할 수 없는 액션을 느리게 할 수 있도록 도울 수 있다. 도 1은 일부 실시예들에 따라 일부 장치들이 가상 세계에서 어떻게 RoA 변동을 야기하도록 동작할 수 있는지를 도시한다.

[0011] 도 1은 본 발명의 일부 실시예들에 따라 가상 세계 내에서의 RoA 변동의 일례를 도시한다. 도 1에서, 운영 환경(100)은 서버 장치("서버"; 128) 및 클라이언트 장치("클라이언트"; 122)를 포함한다. 서버(128)는 가상 세계(101)를 제시하고 관리하는 데 사용되는 코드(120)(예를 들어, 소스 코드, 오브젝트 코드, 기계 코드, 실행가능 파일, 라이브러리, 스크립트 등)를 포함한다. 클라이언트(122)는 가상 세계(101)를 보고 가상 세계(101) 내의 아바타(108)를 제어하도록 활용될 수 있다. 가상 세계(101)는 아바타(108), 건물(110), 차량(109) 등을 포함하는 다양한 오브젝트들을 포함한다. 가상 세계(101)에서, 스테이지 "1"에서는, 오브젝트들이 제1 RoA(104)(t1)에 따라 액션을 수행할 수 있다. 제1 RoA(104)는 가상 세계(101)에서 액션 속도에 영향을 끼친다. 액션 속도는, 오브젝트의 이동 속도(112)(예를 들어, $V(t1)$), 오브젝트들 간의 트랜잭션 속도(114)(TX(t1)), 오브젝트 출현의 리프레시 속도(116)(R(t1)) 등을 포함할 수 있다. 제1 RoA(104)는 시간 관련 공식, 변수 및 상수와 같은 컴퓨터 프로그램 코드 요소들에 의해 결정되고 영향을 받을 수 있다. 또한, 제1 RoA(104)는 데이터 송신 속도, 데이터 조절(data throttling) 및 클록 사이클 속도와 같은 서버 및 클라이언트 처리에 의해 영향을 받을 수 있다.

[0012] 스테이지 "2"에서, 일부 실시예들에 따라 구성된 시스템 장치(예를 들어, 클라이언트(122), 서버(128) 등)는 RoA 변동 프로세스를 적용한다. RoA 변동 프로세스는 제1 RoA가 명백히 다른 제2 RoA로 변경될 수 있게 한다. 스테이지 "3"은 제1 RoA(104)를 제2 RoA(125)로 변경함으로써 이러한 프로세스를 예시한다. 제2 RoA(125)는 증가되거나 감소된다. 스테이지 "4"에서, 제2 RoA(125)는 가상 세계(101)에 적용되고, 이는 가상 세계에서의 액션 속도가 비례적으로 변경되게 한다. 예를 들어, 이동 속도(112)는 RoA에서의 변경에 응답하여 증가되거나 감소된다. 트랜잭션 속도, 그래픽 리프레시 속도 등도 RoA에서의 변경에 응답하여 증가되거나 감소될 수 있다.

[0013] 운영 환경의 예

[0014] 이 섹션은 운영 환경과 네트워크의 예를 설명하며 일부 실시예들의 구조적 양태를 제시한다. 더 구체적으로, 이 섹션은 RoA 변동 장치 운영 환경, RoA 변동 장치 아키텍처 및 RoA 변동 장치 운영 환경에 대하여 설명한다.

[0015] 가상 세계에서의 액션 속도를 수정하고 제어하는 예

[0016] 도 2는 운영 환경(200)에서의 RoA 변동 장치(202)의 일례를 도시한다. 도 2에서, RoA 변동 장치(202)는 운영 환경(200)의 통신 네트워크(222)에 접속된다. 서버(228)와 클라이언트(224)도 통신 네트워크(222)에 접속된다. RoA 변동 장치(202)는, 일부 실시예들에서, 서버(228)와 클라이언트(224, 225, 226)에 개별적으로 또는 별도로 포함될 수 있다. 다른 방안으로, 일부 실시예에서, RoA 변동 장치(202)는 알려져 있는 임의의 장치와는 별도의 것일 수 있다.

[0017] 클라이언트(224)는 가상 세계(201)를 표시한다. 가상 세계에서의 액션들은 제1 RoA(204)에 상관된 속도로 발생할 수 있다. 예를 들어, 오브젝트(209)는 제1 RoA(204)의 함수인 이동 속도($V(t1)$)로 이동한다. 일부 실시예에 따른 RoA 변동 장치(202)는 가상 세계(201) 내의 영역(210)을 선택할 수 있고 제2 RoA(206)를 적용할 수 있다. 아바타(208)와 같은 영역(210)의 오브젝트들은 제2 RoA(206)의 영향을 받는다. 따라서, 이동 속도(212; $V(t2)$)는 제2 RoA의 함수로 된다. 결국, 아바타(208)의 이동 속도(212)는 영역(210)에 대한 제2 RoA(206)의 증가나 감소에 대하여 비례적인 방식과 같은 상관 방식으로 증가되거나 감소된다. 또한, 영역(210) 내의 트랜잭션 속도(214)와 같은 다른 액션 속도도 제2 RoA(206)의 증가나 감소에 대하여 비례적인 방식과 같이 변경될 수 있다. RoA 변동 장치(202)는 영역(218)과 같은 다른 영역들을 선택할 수도 있다. 영역(218)의 경계는, 특정 오브젝트의 경계, 이 경우, RoA 변동 장치(202)가 제1 RoA(204)와도 다른 제3 RoA(216)를 적용하는 건물의 경계를 따른다. 제3 RoA(216)는 제2 RoA(206)와도 다를 수 있다. 따라서, 제2 영역(218) 내의 액션 속도는 예

를 들어 제3 RoA(216)에 비례적으로 변경된다. RoA 변동 장치(202)는, RoA 변동을 트리거(trigger)하기 위한 사용자에게 의한 수동 요청과 같은 트리거 이벤트 또는 가상 세계(201) 내에서 발생하는 이벤트에 따라 가상 세계(210)에서의 RoA 변동을 가능하게 할 수 있다.

- [0018] 액션 속도 변동 장치 아키텍처의 예
- [0019] 도 3은 RoA 변동 장치 아키텍처(300)의 일례를 도시한다. 도 3에서, RoA 변동 장치 아키텍처(300)는 외부 통신 인터페이스(304)를 통해 외부 시스템 및 네트워크(322)와 인터페이싱할 수 있는 RoA 변동 장치(302)를 포함한다. RoA 변동 장치 아키텍처(300)는 가상 세계에서 RoA 변동을 트리거할 수 있는 이벤트들을 검출하도록 구성된 이벤트 검출기(310)를 포함한다.
- [0020] RoA 변동 장치 아키텍처(300)는 가상 세계에서 RoA 변동을 제어하도록 구성된 RoA 변동 컨트롤러(312)도 포함한다. RoA 변동 장치 아키텍처(300)는 가상 세계에 대한 RoA 변동의 효과를 감시하도록 구성된 RoA 변동 효과 감시 장치(314)도 포함한다.
- [0021] 또한, RoA 변동 장치 아키텍처(300)는 가상 세계에 대한 RoA 변동의 적용에 관한 규칙들을 저장하도록 구성된 RoA 변동 규칙 저장소(316)도 포함한다. RoA 변동 장치 아키텍처(300)는 RoA 변동을 경험하는 가상 세계의 영역들을 가리키도록 구성된 RoA 변동 표시 컨트롤러(318)도 포함한다. RoA 변동 장치 아키텍처(300)는 패스워드, 사용자 식별 계정명 등뿐만 아니라 사용자 계정에도 접속하는 데 사용될 수 있는 정보를 저장하도록 구성된 RoA 변동 인가 정보 저장소(320)도 포함한다.
- [0022] 마지막으로, RoA 변동 장치 아키텍처(300)는 RoA 변동 장치(302)의 부품들 간의 통신을 용이하게 하도록 구성된 통신 인터페이스(321)도 포함한다.
- [0023] 가상 세계의 복수의 영역에서의 복수의 액션 속도를 수정하고 제어하는 예
- [0024] 도 4는 운영 환경(400)에서의 RoA 변동 장치(402)의 일례를 도시한다. 운영 환경(400)은 서버(428) 및 복수의 클라이언트(424, 425, 426)도 포함한다. 서버(428), 클라이언트들(424, 425, 426) 및 RoA 변동 장치(402)는 통신 네트워크(422)에 접속된다. RoA 변동 장치(402)는, 일부 실시예들에서, 서버(428) 및 클라이언트들(424, 425, 426)에 개별적으로 또는 별도로 포함될 수 있다. 다른 방안으로, 일부 실시예들에서, RoA 변동 장치(402)는 알려져 있는 임의의 장치와는 별도의 것일 수 있다.
- [0025] 클라이언트들(424, 425, 426)은 가상 세계(401)에 액세스한다. 제1 클라이언트(424)는 제1 아바타(407)를 제어하고, 제2 클라이언트(425)는 제2 아바타(408)를 제어하는 한편, 제3 클라이언트(426)는 관리 역할에 있어서 가상 세계(401)의 다양한 양태들을 감시하고 제어하도록 구성된다. RoA 변동 장치(402)는 가상 세계(401)의 복수의 영역(410, 411, 412)을 선택하고, RoA 변동을 이 영역들(410, 411, 412)에 서로 다른 방식으로 적용한다. 예를 들어, RoA 변동 장치는 제1 RoA(404)를 제1 영역(410)에 적용한다. 제1 영역(410)의 경계는 제2 영역(411) 내로 연장되고 트랜잭션(412)에 관련된 오브젝트 또는 아이템을 둘러싸며, 이에 따라 트랜잭션(412)을 제1 RoA(404)에 상관된 액션 속도로 발생시킨다. RoA 변동 장치(402)는 제2 RoA(413)를 제2 영역(411)에 적용하여, 제2 영역(411) 내의 오브젝트가 제2 RoA(413)에 상관된 액션 속도를 경험하게 한다. 예를 들어, 제2 아바타(408) 및 이동가능 오브젝트(409) 둘 다는 제2 RoA(412)에 상관되는 이동 속도(415)로 이동한다. 그러나, 제1 아바타(407)가 제2 영역(411)의 경계 내에 포함되어 있지만, RoA 변동 장치(402)는 제3 RoA(413)를 제2 영역(411) 내에 포함된 제3 영역(413)에 적용한다. 제3 영역(413)의 경계는 제1 아바타(407) 주위로 연장된다. 따라서, 제1 아바타(407)는 제3 RoA(414)에 상관된 액션 속도를 경험하게 된다. 다시 말하면, 제1 아바타(407)는 제1 영역(410) 또는 제2 영역(411)에서 제1 아바타(408) 또는 다른 임의의 오브젝트와는 다른 속도로 이동한다. 제3 클라이언트(426)는, 예를 들어, 관리 역할에 있어서 RoA 변동 장치(402)를 활용하여, 가상 세계(401) 내의 임의의 근처 오브젝트에 관하여 제1 아바타(407)처럼 아바타들을 느리게 할 수 있다. 제1 아바타(407)가 다른 오브젝트들에 관하여 느리게 되면, 관리자 및 기타 플레이어는 가상 세계에서 문제를 야기할 수 있는 아바타에 비하여 이점을 갖는다. 예를 들어, 시스템에서 해를 끼치는 사용자(rogue user)가 가상 세계에서 하나 이상의 아바타 또는 기타 오브젝트로 하여금 피해를 야기시키고, 불안정을 생성하게 하고, 경제적 테러리즘을 행하게 하거나 부적절하게 행동하게 하면, 관리자는 문제점을 이해하고, 분리하고, 조정(troubleshoot)하는 시간을 필요로 할 수 있다. 따라서, 관리자는 해를 끼치는 사용자의 제어를 받는 아바타 및 오브젝트의 액션을 느리게 하길 원할 수 있다. 따라서, 가상 세계의 내부와 외부 모두에서의 관리자 액티비티는, 가상 세계에서 해를 끼치는 사용자 및 오브젝트의 속도와 비교하여 상대적으로 높은 액션 속도를 가질 수 있다. 보다 높은 액션 속도를 가짐으로써, 관리자는 해를 끼치는 액티비티 및 기타 문제점을 차단하고, 되돌리고(reverse),

정정하는 데 더 많은 시간을 가질 수 있다.

- [0026] 동작의 예
- [0027] 이 섹션은 본 발명의 일부 실시예들에 연관된 동작을 설명한다. 이하의 설명에서는, 전술한 블록도를 참조하여 흐름도를 설명한다. 그러나, 일부 실시예들에서, 이 동작은 블록도에서 설명하지 않은 로직에 의해 수행될 수 있다.
- [0028] 소정의 실시예들에서, 이 동작은 기계 판독가능 매체(예를 들어, 소프트웨어) 상에 상주하는 명령어를 실행함으로써 수행될 수 있는 한편, 다른 실시예들에서, 이 동작은 하드웨어 및/또는 기타 로직(예를 들어, 펌웨어)에 의해 수행될 수 있다. 또한, 일부 실시예들은 임의의 흐름도에서 도시한 모든 동작들보다 적은 동작들을 수행할 수 있다.
- [0029] 도 5는 가상 세계에서 상대적 액션 속도를 제어하는 것을 예시하는 흐름도의 일례이다. 도 5에서, 흐름(500)은 RoA 변동 장치가 제1 RoA에 대하여 액션이 취해지는 가상 세계의 영역을 결정하는 처리 블록(502)으로 시작한다. 제1 RoA는 액션들이 제1 RoA에 대하여 소정의 속도로 영역에서 발생하게 한다. 제1 RoA에 대한 이러한 속도는 그 영역에서 대한 디폴트(default) 속도로 발생할 수 있다. 예를 들어, 이러한 한 속도는 가상 세계의 영역 내에서 아바타가 얼마나 빨리 이동할 수 있는 것일 수 있다. 다시 말하면, 하나의 속도는 아바타의 이동 속도일 수 있다. 기본적으로, 아바타의 이동 속도는 디폴트 속도로 설정될 수 있다. 그러나, 디폴트 속도는 가상 세계에서 액션이 발생하는 속도에 의해 영향을 받는다. 가상 세계의 영역에서 액션이 발생하는 속도가 변경되면, 아바타의 이동 속도도 변경된다. RoA에 의해 영향을 받는 속도의 예로는, 속력 또는 이동 속도, 그래픽 픽셀 리프레시 속도, 트랜잭션 속도 등이 있다.
- [0030] 가상 세계에서 RoA를 변동하는 액션은 가상 세계의 서로 다른 관찰자들 또는 액터들에 관한 가상 세계의 시간 참조(reference of time)를 변경할 수 있다. 이는 일부 관찰자들과 액터들에게는 RoA를 변경하는 것이 유익할 수 있으며 다른 이들에게는 유익하지 않을 수 있다. 이러한 이유로 인해, 일부 실시예들에서, RoA 변동 장치는 시간 진행이 변경될 수 있는 가상 세계 내의 "영역" 또는 소정의 공간을 결정한다. RoA 변동 장치가 하나의 영역만의 RoA를 변경한다면, 다른 영역들을 점유하거나 보고 있는 다른 액터들 또는 관찰자들은 계속해서 비변경 속도로 기능할 수 있다. 다시 말하면, 일부 실시예들에서, 액션 속도들은 결정된 영역에서만 영향을 받는다.
- [0031] 영역 경계는 건물, 방, 또는 도시와 같이 가상 세계 내의 지리적 경계에 특정하게 묶일 수 있다. 또한, 영역 경계는 아바타에 묶이거나 아바타 주위의 소정의 반경 내에 묶일 수 있다. 또한, 영역 경계는 트랜잭션이나 이벤트가 발생하는 오브젝트나 위치에 묶일 수 있다. 예를 들어, 영역은 일상적인 사업 트랜잭션들을 빠르게 진행시키도록 사업 중인 두 개의 아바타에만 묶일 수 있다. 다른 예에서, 영역 경계들은 해를 끼치는 아바타에 의해 야기되는 악성 액티비티를 상당히 느리게 하도록 그 해를 끼치는 아바타 또는 이 아바타가 제어하는 다른 임의의 요소에 묶일 수 있다. 또한, 영역들은 연속적이지 않은 서로 다른 지리적 경계들로 포켓 또는 버블로서 연장될 수 있다. 예를 들어, 두 개의 악성 아바타가 있다면, RoA 변동 장치는, 그 두 개의 아바타가 서로 매우 근접해 있지 않더라도, 그 두 개의 아바타를 포함하지만 그 두 개의 아바타 사이에 있는 오브젝트나 아이টে임을 여전히 포함하지 않도록 경계 영역들을 설정할 수 있다. 다른 예로, RoA 변동에 대한 영역 경계는 가상 세계 내의 좌표에 의해 규정될 수 있다. RoA 변동에 대한 영역 경계는 하드웨어에 의해 규정되어도 된다(예를 들어, 특정한 클라이언트에서 렌더링되거나 특정한 하나 이상의 서버에 의해 렌더링되는 영역(들)에 RoA 변동을 적용한다).
- [0032] 흐름(500)은 RoA 변동 장치가 영역에 대한 제1 RoA를 제1 속도와 다른 제2 RoA로 수정하는 처리 블록(504)에서 계속된다. RoA 변동 장치는 영역에 대한 제1 RoA를 수정할 수 있고, 다시 말하면, RoA를 빠르게 하거나 느리게 할 수 있으며, 이에 따라 그 영역에만 적용되는 제2 RoA를 생성할 수 있다. 제2 RoA는 그 영역에서의 액션들을 상관 방식으로 빠르게 하거나 느리게 할 수 있다. 동시에, RoA 변동 장치는 그 영역을 구분하고 영역 내에 제2 RoA를 적용하는 한편 그 영역 밖의 가상 세계에서 제1 RoA를 적용할 수 있다. 가상 세계의 오브젝트들이 입력되고, 또는 그 영역에 포함되거나 노출되면, 그 오브젝트들은 제2 RoA를 적용받는다. 마찬가지로, 오브젝트들이 가상 세계에서 사라지면, 또는 그 영역으로부터 제거되거나 그 영역에 노출되지 않게 되면, 그 오브젝트들은 제1 RoA를 적용받는다.
- [0033] 흐름(500)은 RoA 변동 장치가 영역에 대한 제1 RoA를 복구할지를 결정하는 처리 블록(506)에서 계속된다. 복구하지 않는다면, 프로세스가 종료된다. 복구한다면, 프로세스는 블록(508)에서 계속 진행된다.
- [0034] 흐름(500)은 RoA 변동 장치가 영역에 대한 제1 RoA를 복구하는 블록(508)에서 진행된다. RoA 변동 장치는 RoA

를 복구할 수 있고, 또는 다시 말하면 RoA의 수정을 되돌려 제2 RoA가 제1 RoA로 복구될 수 있게 한다. 제1 RoA를 복구함으로써, RoA 변동 장치는 그 영역에서의 하나 이상의 액션 속도가 원래의 액션 속도로 복구되게 한다.

- [0035] 도 6은 데이터 처리 속도를 수정함으로써 가상 세계에서의 액션 속도를 수정하는 것을 예시하는 흐름도의 일례이다. 도 6에서, 흐름(600)은 RoA 변동 장치가 가상 세계에 대하여 RoA 변동을 적용하는 것을 결정하는 처리 블록(602)에서 시작한다.
- [0036] 흐름(600)은 가상 세계의 데이터가 클라이언트 장치 상에서 렌더링되는 속도를 증가시키거나 감소시키도록 RoA 변동 장치가 데이터 처리 속도를 수정하는 처리 블록(604)에서 계속된다. 클라이언트 장치는 데이터를 렌더링하여 가상 세계의 액션을 표시한다. 따라서, 가상 세계의 데이터가 클라이언트 상에서 렌더링되는 속도는 가상 세계에서의 시간 진행의 지각을 효과적으로 수정할 수 있다.
- [0037] 흐름(600)은 클라이언트 장치, 또는 다른 일부 장치가 데이터 처리 속도를 수정하는지를 RoA 변동 장치가 결정하는 처리 블록(606)에서 계속된다. 클라이언트 장치는 데이터를 렌더링하여 가상 세계의 액션을 표시하지만, 클라이언트 및 다른 장치는 클라이언트가 그 데이터를 렌더링하는 속도에 영향을 끼치는 처리 속도로 그 데이터를 준비하고 제시할 수 있다. 클라이언트 장치가 데이터 처리 속도를 수정하면, 프로세스는 블록(608)에서 계속된다. 수정하지 않으면, 프로세스는 블록(614)에서 계속된다.
- [0038] 흐름(600)은 RoA 변동 장치가 클라이언트 장치에서의 데이터 처리 속도를 수정하는 처리 블록(608)에서 계속된다. RoA 변동 장치는 클라이언트에서 데이터 처리 속도를 수정할 수 있다. 예를 들어, RoA 변동 장치는 데이터를 데이터 렌더링 부품(예를 들어, 비디오 카드, 프로세서(들), 비디오 카드, 등)에 전달하기 전에 클라이언트에서 그 데이터를 조절할 수 있다. 또한, RoA 변동 장치는 데이터가 보다 느린 컴퓨터 처리 속도나 보다 빠른 컴퓨터 처리 속도에 따라 처리되도록 클럭 사이클 속도를 수정할 수 있다. 따라서, 그 영향은 데이터 렌더링 부품이 그 데이터를 보다 느린 속도나 보다 빠른 속도로 렌더링하게 한다.
- [0039] 흐름(600)은 데이터 처리 속도를 복구해야 하는지를 RoA 변동 장치가 결정하는 처리 블록(610)에서 계속된다. 데이터 처리 속도가 복구되지 않는다면, 프로세스는 종료될 수 있다. 데이터 처리 속도가 복구되어야 한다면, 프로세스는 처리 블록(612)에서 계속될 수 있다.
- [0040] 흐름(600)은 RoA 변동 장치가 클라이언트에서의 데이터 처리 속도를 복구하는 처리 블록(612)에서 계속될 수 있다. RoA 변동 장치는 클라이언트로 하여금 처리 블록(608)에서 설명한 수정을 되돌리게 할 수 있다. 예를 들어, 클라이언트는 데이터를 조절하는 것을 중단하거나 클라이언트 클럭 사이클을 원래의 사이클 속도로 복구시킬 수 있다. 처리 블록(606)으로 복구하여, 처리 블록(606)에서 클라이언트가 데이터 처리 속도를 수정하지 않으면, 프로세스는 처리 블록(614)에서 계속된다.
- [0041] 흐름(600)은 RoA 변동 장치가 서버 장치로부터 데이터 처리 속도를 수정하는 처리 블록(614)에서 시작된다. RoA 변동 장치는 서버 또는 기타 네트워크 장치로 하여금 데이터 처리 속도를 수정하게 할 수 있다. 예를 들어, RoA 변동 장치는 서버로 하여금 클라이언트로의 데이터 공급 속도를 수정하게 할 수 있다. 데이터 공급 속도를 수정함으로써, RoA 변동 장치는 데이터를 클라이언트에게 정상 상태보다 빠르게 또는 느리게 전달할 수 있다. 따라서, 그 효과는 클라이언트로 하여금 데이터를 느리게 또는 빠르게 렌더링하게 할 수 있다.
- [0042] 흐름(600)은 데이터 처리 속도를 복구해야 하는지를 RoA 변동 장치가 결정하는 처리 블록(616)에서 계속된다. 데이터 처리 속도가 복구되지 않으면, 프로세스는 종료될 수 있다. 데이터 처리 속도가 복구되어야 한다면, 프로세스는 처리 블록(618)에서 계속될 수 있다.
- [0043] 흐름(600)은 RoA 변동 장치가 클라이언트에서의 데이터 처리 속도를 복구하는 처리 블록(618)에서 계속된다. RoA 변동 장치는 서버로 하여금 처리 블록(608)에서 설명한 수정을 되돌리게 할 수 있다. 예를 들어, 서버는 데이터 조절을 중단하거나 데이터 전송 속도를 원래의 속도로 복구시킬 수 있다.
- [0044] 도 7은 프로그램 코드 요소들을 수정함으로써 액션의 속도를 수정하는 흐름도의 일례이다. 도 7에서, 흐름(700)은 가상 세계에 대하여 RoA 변동이 적용되는 것을 RoA 변동 장치가 결정하는 처리 블록(702)에서 시작된다.
- [0045] 흐름(700)은 하나 이상의 컴퓨터 프로그램 코드 요소에 따라 RoA가 발생하는 가상 세계의 영역을 RoA 변동 장치가 선택하는 처리 블록(704)에서 계속된다. 컴퓨터 프로그램 코드 요소는, 가상 세계에서의 시간 진행을 결정하는 알고리즘에 영향을 끼칠 수 있는 변수, 공식, 또는 다른 임의의 종류의 컴퓨터 프로그래밍을 포함할 수 있

다.

- [0046] 흐름(700)은 RoA 변동 장치가 그 하나 이상의 컴퓨터 프로그램 코드 요소를 수정하여 선택된 영역에서 RoA가 빨라지거나 느리게 할 수 있는 처리 블록(706)에서 계속된다. 이러한 프로그램 코드 요소는, 예를 들어, 변수로서 시간 계산 또는 시간 자체에 직접적으로 연관되는 프로그래밍 요소를 포함할 수 있다. 이러한 일례로는 그 영역 내의 오브젝트들의 속력 또는 속도에 대한 공식이 있다. 속도를 결정하는 공식은 등식에서 시간을 변수로서 포함한다(예를 들어, 속도 = 거리/시간). RoA 변동 장치가 액션이 발생하는 속도를 수정하면, 이는, 예를 들어, 가상 세계 내에서 시간이 진행되고 속도의 계산에 영향을 끼치는 속도 변경으로 고려할 수 있다. 변수로서 시간에 직접적으로 연관되는 일부 추가 프로그래밍 코드 요소들은 주파수, 가속도 및 중력을 결정하기 위한 공식을 포함한다.
- [0047] 반면에, RoA 변동 장치는, 시간에 직접적으로 연관되지는 않지만 영역에서 액션 속도가 효과적으로 변경될 수 있게 하는, 가상 세계에서의 공기의 점도를 계산하기 위한 공식과 같은 프로그래밍 요소들을 수정할 수도 있다. RoA 변동 장치는, 가상 세계의 영역에서의 공기의 점도를 결정하는 공식, 함수, 상수 등을 수정할 수 있으며, 이에 따라 아이템들이 그 영역에서 보다 빠르게 또는 보다 느리게 이동하게 할 수 있다. 따라서, RoA 변동 장치는 시간 진행의 지각을 제공하는 액션을 수정함으로써 RoA를 수정할 수 있다. 시간에 직접적으로 연관되지는 않지만 시간 진행의 지각을 제공할 수 있는 다른 프로그래밍 요소들은 질량, 마찰, 밀도에 대한 공식, 함수, 상수 등을 포함할 수 있다.
- [0048] 흐름(700)은 RoA 변동 장치가 RoA를 복구할지를 결정하는 처리 블록(708)에서 계속된다. RoA 변동 장치가 RoA를 복구하지 않는다면, 프로세스가 종료된다. 반면에, RoA 변동 장치가 RoA를 복구하기로 결정하면, 프로세스는 처리 블록(708)에서 계속된다.
- [0049] 흐름(700)은 RoA 변동 장치가 하나 이상의 컴퓨터 프로그램 코드 요소를 복구하여 RoA가 복구되게 하는 처리 블록(710)에서 계속된다. 다시 말하면, RoA 변동 장치는 수정된 프로그램 코드 요소를 적용하는 것을 중단하고 수정 전의 프로그램 코드 요소를 적용한다. 이는 액션 속도가 블록(704)에서 설명한 수정 전의 값으로 복구하는 것을 보장한다.
- [0050] 도 8은 가상 세계에서 RoA 변동을 적용하는 것을 예시하는 흐름도의 일례이다. 도 8에서, 흐름(800)은 RoA 변동을 의도적으로 트리거하는 가상 세계의 이벤트를 RoA 변동 장치가 검출하는 처리 블록(802)에서 시작된다. 이벤트는, RoA 변동 규칙 또는 설정에 따라 RoA 변동이 필요함을 가리키는, 수동적으로 개시되거나 자동적으로 발생하는 가상 세계에서의 임의의 액티비티일 수 있다. 예를 들어, 이벤트는 가상 세계에 대한 악성 공격일 수 있다. RoA 변동 장치는 악성 공격을 검출하고 특정 영역을 RoA 변동 과정을 거치게 하여 그 공격의 영향을 약하게 할 수 있다. 반면에, 이벤트는 가상 세계의 사용자에게 의해 행해지는 돈 거래와 같이 바람직한 것일 수 있다. 미숙한 플레이어가 거래를 행하는 경우, RoA 변동 장치는, 거래가 적절히 완료되도록 미숙한 플레이어가 거래를 행하는 데 일부 추가 시간을 필요로 한다는 점을 결정할 수 있다. 따라서, RoA 변동 장치는 재정 거래가 발생하는 영역에 RoA 변동이 적용되어야 함을 검출할 수 있다. 미숙한 플레이어는 RoA 변동을 수동으로 개시할 수도 있고, 이에 따라 RoA 변동을 거래가 발생하는 영역에 적용할 필요가 있음을 수동으로 가리킬 수 있다.
- [0051] 흐름(800)은 RoA 변동 장치가 이벤트의 통지 메시지를 관리 계정에 송신하는 처리 블록(804)에서 계속된다. RoA 변동 장치는 관리자에게 이벤트를 통지하는 메시지를 송신할 수 있다. 관리자는 이벤트 또는 그 이벤트의 특정 효과를 다루도록 특정 액션을 수행할 필요가 있을 수 있다.
- [0052] 흐름(800)은 RoA 변동이 점진적인 순서로 적용될 수 있는 가상 세계의 한 영역 내의 복수의 부영역(sub-area)을 RoA 변동 장치가 결정하는 처리 블록(806)에서 계속된다. 부영역들은, 가상 세계의 전체 영역으로부터 보다 작은 영역, 특정 오브젝트, 또는 거래가 발생하고 있는 위치에 걸쳐 가상 세계 내의 어떠한 것도 둘러싸는 부영역 경계들을 포함할 수 있다. RoA 변동 장치는 점진적인 순서에 따라 부영역들을 결정할 수 있다. 다시 말하면, RoA 변동 장치는 어느 부영역들이 이벤트에 의해 가장 영향을 많이 받을 수 있는지, 어느 부영역들이 RoA 변동에 의해 인터럽트될 수 있는 가장 큰 액티비티를 가질 수 있는지, 또는 어느 영역들이 가상 세계의 거주자들에게 의해 가장 많이 채워지는 영역들일 수 있는지를 결정할 수 있고, 이에 따라 RoA 변동을 그러한 부영역들에 처음에, 마지막에, 또는 거주자들에게는 최소한으로 거슬리거나 시스템의 건강을 위해 최대한 효과적인 다른 논리적 순서로 적용하는 것을 결정할 수 있다. 예를 들어, 이벤트가 시스템의 성능에 영향을 끼칠 수 있는 유해한 이벤트이라면, RoA 변동 장치는, 최적의 점진적 순서는 RoA 변동을 그때 가장 많이 채워진 영역들에 적용하는 것이라고 판단할 수 있다. RoA 변동은 이러한 점진적인 순서에 적용될 수 있으며 그 이유는 이벤트 효과가 가장

많은 사용자들에게 영향을 끼치기 때문이다. 그러나, 덜 채워진 다른 영역들은 많이 채워진 영역들 후에 다루어지게 된다.

- [0053] 흐름(800)은, 이벤트에 의해 영향을 받는, 복수의 부영역이나 오브젝트 중 적어도 하나를, RoA 변동 장치가 점진적으로 선택하는 처리 블록(808)에서 계속된다.
- [0054] 흐름(800)은 RoA 변동 장치가 RoA 변동을 선택된 부영역에 적용하는 처리 블록(810)에서 계속된다. 다시 말하면, RoA 변동 장치는 선택된 부영역 내에서의 액션 속도를 수정한다.
- [0055] 흐름(800)은 RoA 변동 장치가 RoA 변동의 시각적 지시기(indicator)를 가상 세계의 선택된 부영역에 표시하는 처리 블록(812)에서 계속된다. RoA 변동 장치는 안개(haze), 색, 또는 일부 다른 시각적 지시와 같은 지시기를 일시적으로 변동되는 부영역에 표시할 수 있다.
- [0056] 흐름(800)은 RoA 변동 장치가 이벤트 효과를 선택된 부영역에서 다루는 처리 블록(814)에서 계속된다. 예를 들어, 이벤트 효과가 시스템 문제나 불안정성과 같이 유해한 것이라면, RoA 변동 장치는 그 문제를 해결할 수 있으며, 또는 네트워크 사용자나 관리자에 의해 문제가 해결되었음을 검출할 수 있다. 반면에, 이벤트 효과가 부영역에서 재정 거래를 행하는 것과 같이 유익한 것이라면, RoA 변동 장치는 거래를 감시하여 이벤트가 완료되는 때를 결정할 수 있다.
- [0057] 흐름(800)은 RoA 변동 장치가 선택된 부영역을 분석하여 이벤트 효과의 상태를 결정하는 처리 블록(816)에서 계속된다. RoA 변동 장치는 선택된 부영역에서 발생하는 액션을 분석하여 이벤트가 종료되었는지 또는 이벤트 효과가 부영역에 여전히 영향을 끼치고 있는지를 결정할 수 있다. 예를 들어, RoA 변동 장치는 부영역의 현재 액티비티의 메트릭을 이벤트 발생 전의 부영역의 액티비티의 메트릭과 비교할 수 있다.
- [0058] 흐름(800)은 부영역이 이벤트에 의해 여전히 영향을 받고 있는지를 RoA 변동 장치가 결정하는 처리 블록(818)에서 계속된다. 부영역이 이벤트에 의해 여전히 영향을 받고 있다면, 프로세스는 처리 블록(814)으로 복귀하여 부영역이 이벤트에 의해 더 이상 영향을 받지 않을 때까지 블록들(814, 816)을 반복할 수 있다. 프로세스는 처리 블록(820)에서 계속된다.
- [0059] 흐름(800)은 RoA 변동 장치가 부영역에서 RoA 변동을 되돌리는 것을 결정하는 처리 블록(820)에서 계속된다. RoA 변동 장치는 액션 속도를 원래의 속도로 복귀시킬 수 있다. 또한, RoA 변동 장치는 부영역에서 RoA 지시기의 임의의 표시 지시기를 되돌리거나 턴오프할 수 있다.
- [0060] 흐름(800)은 가상 세계의 전체 영역에 걸쳐 이벤트 효과가 충분히 다루어졌는지를 RoA 변동 장치가 결정하는 처리 블록(822)에서 계속된다. 이벤트와 이벤트의 효과가 전체 영역에 걸쳐 종료되었다면, 프로세스가 종료될 수 있다. 그러나, 이벤트 또는 이벤트의 효과를 여전히 경험하고 있는 영역 내의 다른 부영역들이 존재한다면, 프로세스는 처리 블록(808)으로 복귀하여 이벤트 또는 이벤트의 효과가 종료될 때까지 후속 블록들에서의 처리를 반복한다.
- [0061] 부영역들에 RoA 변동을 점진적으로 적용하는 것은 액션 속도의 변동을 점진적으로 적용하는 것의 일례이다. 다른 예에서, RoA 변동은 어느 정도 점진적으로 적용된다. 예를 들어, 영역에 대한 액션 속도는 액션 속도의 세 배로 증분 방식으로 증가될 수 있다. 다른 예로, 오브젝트들은 RoA 변동을 개별적으로 종료할 수 있다. 두 개의 오브젝트를 포함하는 가상 세계의 영역에 대하여 액션 속도가 느려진다고 가정하면, 제1 오브젝트에 대한 액션 속도는 일단 RoA 변동이 그 영역에 대하여 종료되어야 한다고 결정되면 디폴트 속도로 즉시 복구될 수 있다. 제2 오브젝트는 디폴트 액션 속도로 점진적으로 복구될 수 있다.
- [0062] 도 9는 가상 세계에서 RoA 변동을 적용하는 것을 예시하는 흐름도의 일례이다. 도 9에서, 흐름(900)은 RoA 변동 장치가 RoA 변동을 트리거하는 가상 세계의 이벤트를 검출하는 처리 블록(902)에서 시작된다.
- [0063] 흐름(900)은 RoA 변동 장치가 이벤트의 통지를 관리 계정에 송신하는 처리 블록(904)에서 계속된다. RoA 변동 장치는 관리자에게 이벤트를 통지하는 메시지를 송신할 수 있다. 관리자는 이벤트 또는 이벤트의 특정 영향을 다루도록 특정한 액션을 수행할 필요가 있을 수 있다.
- [0064] 흐름(900)은 RoA 변동 장치가 RoA 변동을 이벤트에 의해 영향을 받은 가상 세계의 영역에 적용하는 처리 블록(906)에서 계속된다.
- [0065] 흐름(900)은 RoA 변동 장치가 RoA 변동이 적용된 시각적 지시기를 영향을 받은 영역에 표시하는 처리 블록(908)에서 계속된다. RoA 변동 장치는 안개, 색, 또는 일부 다른 시각적 지시와 같은 지시기를 변동된 RoA의 영향

을 받는 부영역에 표시할 수 있다.

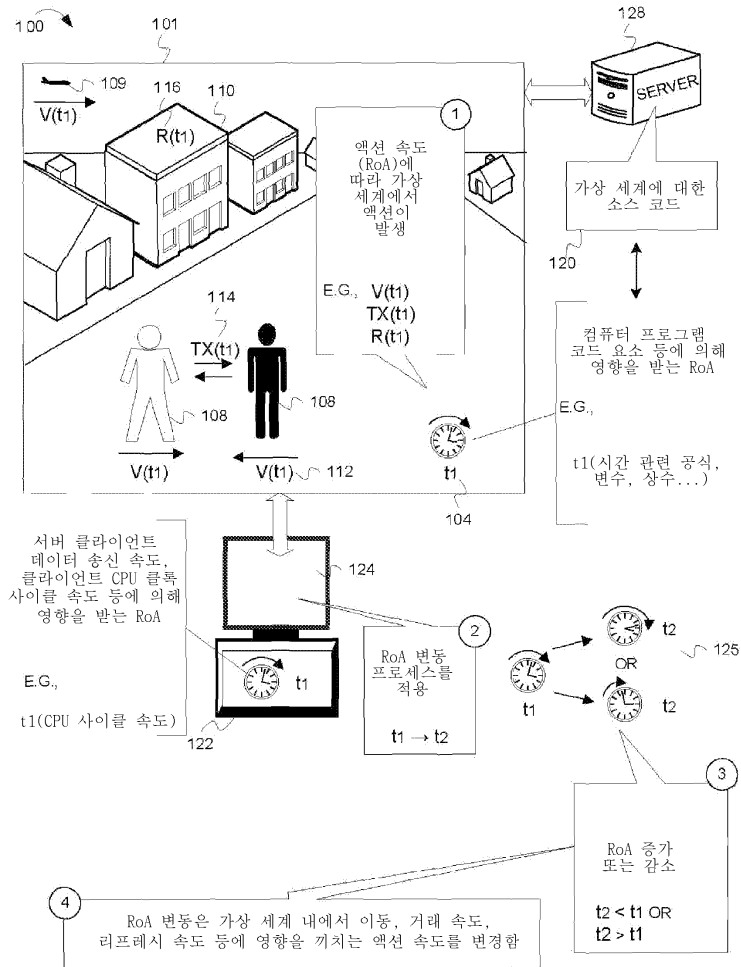
- [0066] 흐름(900)은 RoA 변동을 증분적 순서로 또는 점진적 순서로 되돌릴 수 있는 영향을 받은 영역 내의 복수의 부영역을 RoA 변동 장치가 결정하는 처리 블록(910)에서 계속된다. RoA 변동 장치는 도 8에서 설명한 바와 같이 증분적 순서 또는 점진적 순서에 따라 부영역들을 결정할 수 있다.
- [0067] 흐름(900)은 RoA 변동 장치가 이벤트에 영향을 받은 부영역을 선택하는 처리 블록(912)에서 계속된다.
- [0068] 흐름(900)은 RoA 변동 장치가 선택된 부영역에서 이벤트 효과를 다루는 처리 블록(914)에서 계속된다.
- [0069] 흐름(900)은 RoA 변동 장치가 부영역에서 RoA 변동을 되돌리는 처리 블록(916)에서 계속된다. RoA 변동 장치는, 이벤트 효과가 다루어졌으므로 영역으로부터 RoA 변동을 제거한다. RoA 변동을 제거함으로써 액티비티들이 각자의 원래 속도로 가능한 빠르게 복귀될 수 있다.
- [0070] 흐름(900)은 RoA 변동 장치가 부영역으로부터 RoA 변동의 표시 지시기를 제거하는 처리 블록(918)에서 계속된다.
- [0071] 흐름(900)은 RoA 변동 장치가 선택된 부영역을 분석하여 이벤트 효과의 상태를 결정하는 처리 블록(920)에서 계속된다. RoA 변동 장치는, 선택된 부영역에서 발생하는 액션들을 분석하여 이벤트가 종료되었는지 또는 이벤트 효과가 부영역에 여전히 영향을 끼치고 있는지를 결정할 수 있다. 예를 들어, RoA 변동 장치는 부영역의 현재 액티비티의 메트릭들을 이벤트가 발생하기 전의 부영역의 액티비티의 메트릭과 비교한다.
- [0072] 흐름(900)은 부영역이 이벤트에 의해 여전히 영향을 받는지를 RoA 변동 장치가 결정하는 처리 블록(922)에서 계속된다. 부영역이 이벤트에 의해 여전히 영향을 받고 있다면, 프로세스는 처리 블록(924)으로 진행될 수 있다. 그러나, 부영역이 이벤트에 의해 영향을 받지 않는다면, 프로세스는 처리 블록(926)으로 계속될 수 있다.
- [0073] 흐름(900)은 RoA 변동 장치가 RoA 변동을 다시 적용하여 시각적 지시기를 영향을 받은 부영역에 다시 표시하는 처리 블록(924)에서 계속된다. 이어서, 프로세스는 처리 블록(914)으로 복귀하여 블록(922)을 통한 후속 처리 블록들을 부영역이 이벤트에 의해 더 이상 영향을 받지 않을 때까지 반복할 수 있다.
- [0074] 흐름(900)은 그 영역에서 이벤트의 효과가 충분히 다루어졌는지를 RoA 변동 장치가 결정하는 처리 블록(926)에서 계속된다. 그 영역이 여전히 이벤트 효과를 경험하고 있다면, 프로세스는 블록(912)으로 복귀하여 다른 부영역을 선택하고 후속 처리 블록을 새롭게 선택된 부영역에 적용할 수 있다.
- [0075] 흐름(900)은 RoA 변동 장치가 전체 영역에 걸쳐 RoA 변동을 되돌리는 처리 블록(928)에서 계속된다. 일단 한 영역의 부영역들 중 어떠한 부영역에서도 이벤트 효과가 더 이상 분명하지 않다고 RoA 변동 장치가 결정하면, RoA 변동 장치는 전체 영역에 걸쳐 RoA 변동이 되돌려졌거나 턴오프되어 영역을 원래의 액션 속도로 복귀시켰음을 보장한다.
- [0076] 일부 실시예들에서는, 추가로 전술한 동작들이 연속해서 수행될 수 있는 한편, 다른 실시예에서는, 그 동작들 중 하나 이상이 병행 수행될 수 있다.
- [0077] RoA 변동 장치 네트워크의 예
- [0078] 도 10은 네트워크(1000) 상의 RoA 변동 장치(1002)의 일례를 도시한다. 도 10에서, RoA 변동 장치 네트워크(1000)라고도 칭하는 네트워크(1000)는 RoA 변동 장치(1002)를 이용할 수 있는 네트워크 장치들(1004, 1008)을 포함하는 제1 로컬 네트워크(1012)를 포함한다. 네트워크 장치들(1004, 1008)의 예로는, PC, PDA, 이동 전화, 메인프레임, 미니컴퓨터, 랩탑, 서버 등이 있다. 도 10에서, 일부 네트워크 장치들(1004)은 서버 장치(1008) ("서버")와 함께 동작할 수 있는 클라이언트 장치들("클라이언트")일 수 있다. 서버(1008)와 네트워크 장치들(1004) 중 임의의 하나는 도 11에 도시한 컴퓨터 시스템으로서 구체화될 수 있다. 통신 네트워크(1022)는 제2 로컬 네트워크(1018)를 제1 로컬 네트워크(1012)에 접속한다. 제2 로컬 네트워크(1018)도 RoA 변동 장치(1006)를 이용할 수 있는 클라이언트(1024)와 서버(1028)를 포함한다.
- [0079] 다시 도 10을 참조해 보면, 통신 네트워크(1012)는 LAN 또는 WAN일 수 있다. 통신 네트워크(1012)는 공중 전화망(PSTN), 이더넷, 802.11g, SONET 등과 같은 임의의 적절한 기술을 포함할 수 있다. 편의상, RoA 변동 장치 네트워크(1000)는 통신 네트워크(1022)에 접속된 여섯 개의 클라이언트(1004, 1024) 및 두 개의 서버(1008, 1028)만을 도시하고 있다. 실제로는, 클라이언트와 서버의 개수가 다를 수 있다. 또한, 일부 경우에, 장치는 클라이언트와 서버 둘 다의 기능을 수행할 수 있다. 또한, 클라이언트들(1004, 1024)은 통신 네트워크(1022)에 접속하여 각자의 네트워크(1012, 1028) 또는 다른 네트워크(도시하지 않음) 내의 다른 장치들과 데이터를 교환

할 수 있다.

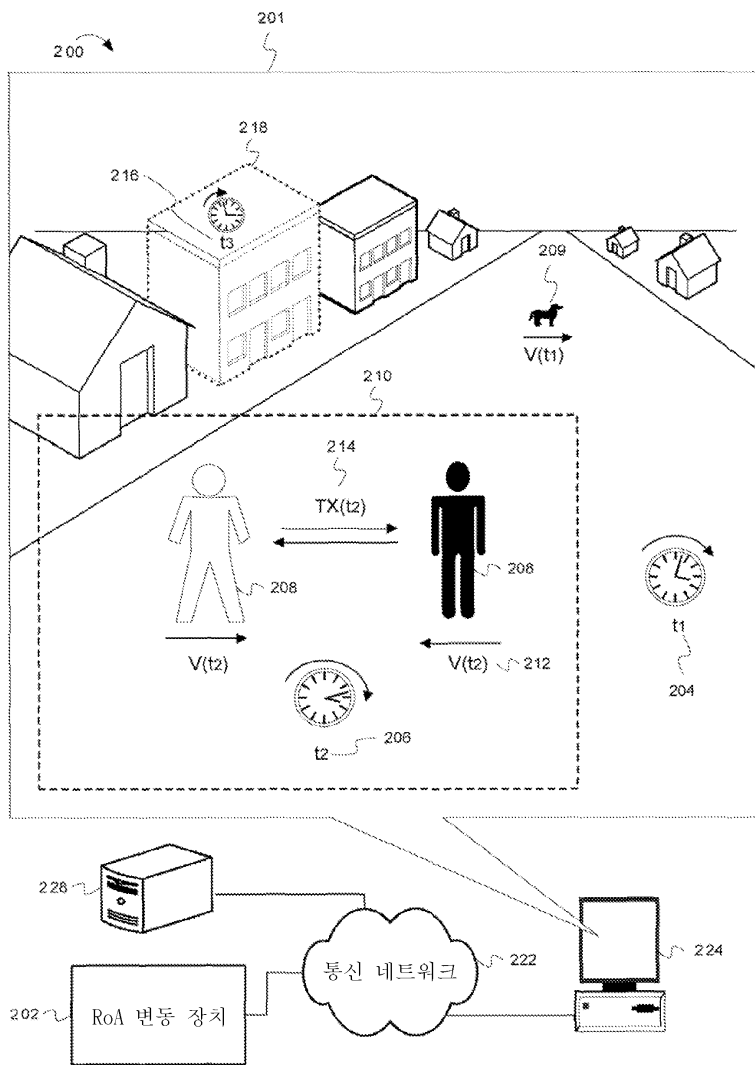
- [0080] RoA 변동 장치 컴퓨터 시스템의 예
- [0081] 도 11은 RoA 변동 장치 컴퓨터 시스템(1100)의 일례를 도시한다. 도 11에서, RoA 변동 장치(1100)("컴퓨터 시스템")는 시스템 버스(1104)에 접속된 CPU(1102)를 포함한다. 시스템 버스(1104)는 (노스 브리지라고도 칭하는) 메모리 컨트롤러(1106)에 접속되고, 이 메모리 컨트롤러는 메인 메모리 유닛(1108), AGP 버스(1110) 및 AGP 비디오 카드(1112)에 접속된다. 메인 메모리 유닛(1108)은 다이내믹 RAM, 연장 데이터 출력(EDO) RAM 등과 같은 임의의 적절한 RAM을 포함할 수 있다.
- [0082] 일 실시예에서, 컴퓨터 시스템(1100)은 RoA 변동 장치(1137)를 포함한다. RoA 변동 장치(1137)는 통신, 커맨드, 또는 기타 정보를 처리할 수 있고, 여기서 처리는 가상 세계에서의 상대적 액션 속도를 제어할 수 있다. RoA 변동 장치(1137)는 시스템 버스(1104)에 접속된 것으로 도시되어 있지만, RoA 변동 장치(1137)는 컴퓨터 시스템(1100) 내의 다른 버스 또는 장치에 접속될 수 있다. RoA 변동 장치(1137)는 메인 메모리(1108)를 활용하는 소프트웨어 모듈을 포함할 수 있다.
- [0083] 확장 버스(1114)는 메모리 컨트롤러(1106)를 (사우스 브리지라고도 칭하는) 입/출력(I/O) 컨트롤러(1116)에 접속할 수 있다. 실시예들에 따르면, 확장 버스(1114)는 주변 부품 상호접속(peripheral component interconnect; PCI) 버스, PCI-X 버스, PC 카드 버스, CardBus 버스, InfiniBand 버스, 또는 산업 표준 아키텍처(ISA) 버스 등을 포함할 수 있다.
- [0084] I/O 컨트롤러(1116)는 하드 디스크 드라이브(HDD; 1118), 디지털 버스타일 디스크(DVD; 1120), 입력 장치 포트(1124)(예를 들어, 키보드 포트, 마우스 포트 및 조이스틱 포트), 병렬 포트(1138) 및 유니버설 시리얼 버스(USB; 1122)에 접속된다. USB(1122)는 USB 포트(1140)에 접속된다. I/O 컨트롤러(1116)는 XD 버스(1126)와 ISA 버스(1128)에도 접속된다. ISA 버스(1128)는 오디오 장치 포트(1136)에 접속되는 한편, XD 버스(1126)는 BIOS ROM(1130)에 접속된다.
- [0085] 일부 실시예들에서, 컴퓨터 시스템(1100)은 도 11에 도시한 각 부품을 하나보다 많게 그리고 추가 주변 장치들을 포함할 수 있다. 예를 들어, 일부 실시예들에서, 컴퓨터 시스템(1100)은 다수의 외부 다중 CPU(1102)를 포함할 수 있다. 일부 실시예들에서, 부품들 중 임의의 것은 통합되거나 세분화될 수 있다.
- [0086] 컴퓨터 시스템(1100)의 임의의 부품은 본 명세서에서 설명하는 동작들을 수행하기 위한 명령어들을 포함하는 하드웨어, 펌웨어, 및/또는 기계 판독가능 매체로서 구현될 수 있다.
- [0087] 진술한 실시예들은, 모든 가능한 변동을 본 명세서에서 열거하지 않으므로, 현재 설명되어 있는지에 상관없이, (본 발명의 실시예들에 따른 프로세스를 수행하도록) 컴퓨터 시스템(또는 다른 전자 장치(들))을 프로그래밍하는 데 사용될 수 있는 명령어들이 저장된 기계 판독가능 매체를 포함할 수 있는 컴퓨터 프로그램 제품 또는 소프트웨어로서 제공될 수 있다. 기계 판독가능 매체는, 기계(예를 들어, 컴퓨터)에 의해 판독가능한 형태(예를 들어, 소프트웨어, 처리 애플리케이션)로 정보를 저장하거나 송신하기 위한 임의의 메카니즘을 포함할 수 있다. 기계 판독가능 매체는, 자기 저장 매체(예를 들어, 플로피 디스켓), 광학 저장 매체(예를 들어, CD-ROM), 자기 광학 저장 매체, ROM, RAM, 소거가능 프로그래밍가능 메모리(예를 들어, EPROM 및 EEPROM), 플래시 메모리, 또는 전자 명령어를 저장하는 데 적절한 다른 유형의 매체를 포함할 수 있지만, 이러한 예로 한정되지는 않는다. 또한, 실시예들은 전달 신호(예를 들어, 반송파, 적외선 신호, 디지털 신호 등) 또는 배선, 무선, 또는 기타 통신 매체의 전기적 형태, 광학적 형태, 음향적 형태 또는 다른 형태로 구체화될 수 있다.
- [0088] 일반 사항
- [0089] 진술한 본 발명의 상세한 설명은 예시한 도면의 특정 예들에 관한 것이다. 이러한 예들은 당업자가 본 발명의 요지를 실시할 수 있도록 충분히 설명되어 있다. 또한, 이러한 예들은 본 발명의 요지가 다양한 목적이나 실시예에 어떻게 적용될 수 있는지를 예시하도록 기능한다. 본 명세서에서 설명한 실시예들에 논리적 변경, 기계적 변경, 전기적 변경 및 기타 변경을 행할 수 있듯이, 다른 실시예들도 본 발명의 요지에 포함된다. 그러나, 본 명세서에서 설명하는 다양한 실시예들의 특징들은, 이러한 특징들이 포함되는 예들에 필수적인 것이지만, 전체적으로 볼 때 본 발명의 요지를 한정하지 않으며, 본 발명에 대한 임의의 참조, 요소, 동작, 및 응용은 전체적으로 볼 때 한정적이지 않으며 이러한 예들을 규정할 뿐이다. 따라서, 이러한 상세한 설명은 본 발명의 실시예들을 한정하지 않으며, 첨부된 청구범위에 의해서만 규정된다. 본 명세서에서 설명하는 실시예들의 각각은 이하의 청구범위에서 설명하는 본 발명의 요지 내에 속하는 것으로 고려된다.

도면

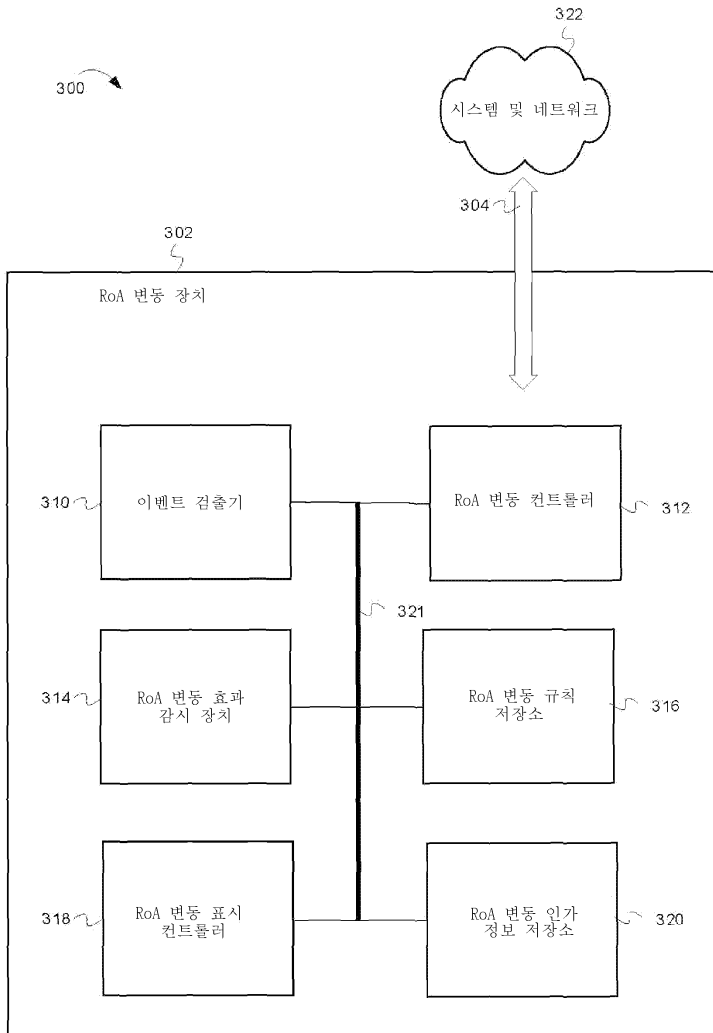
도면1



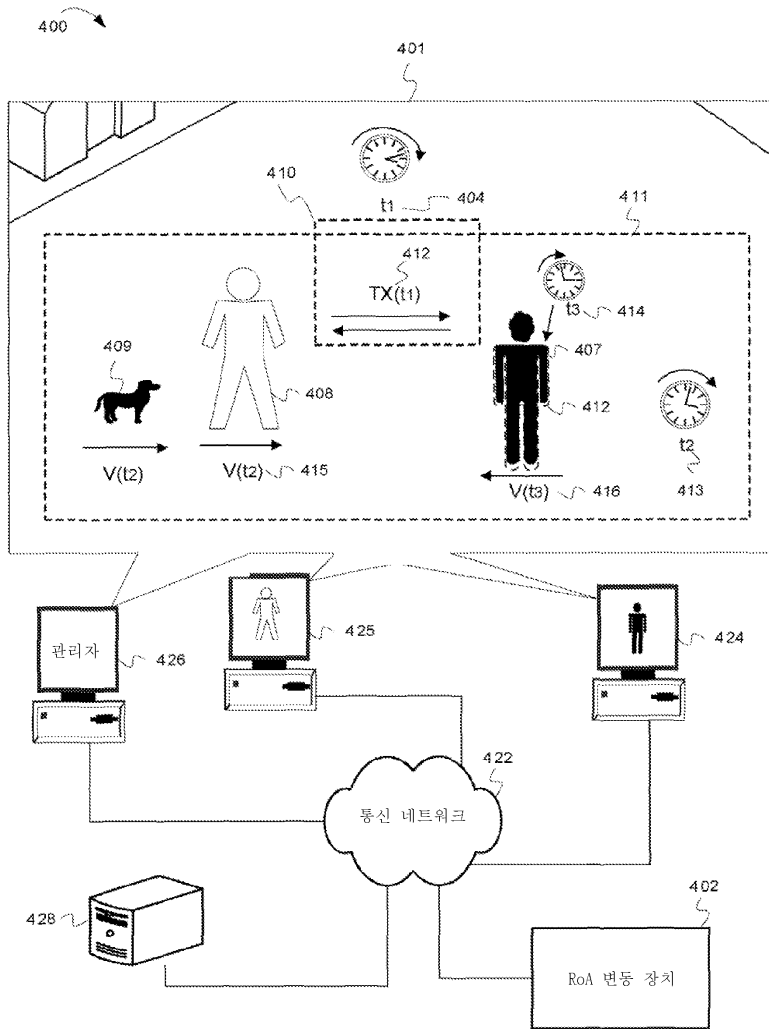
도면2



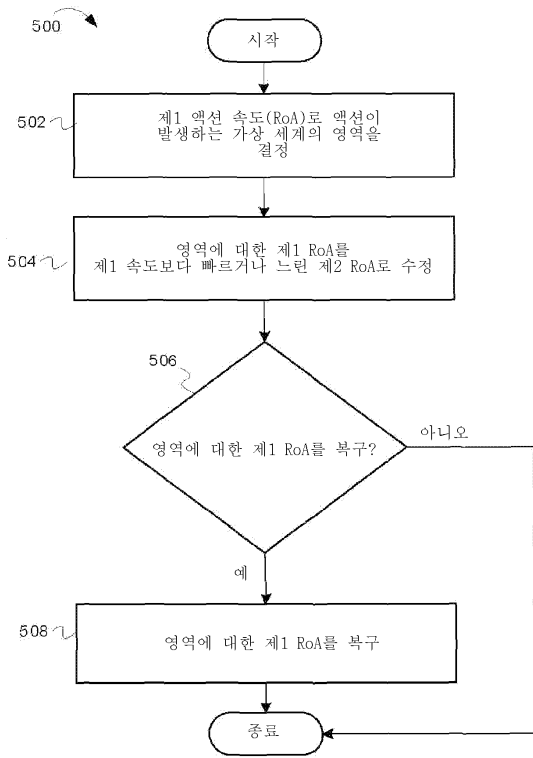
도면3



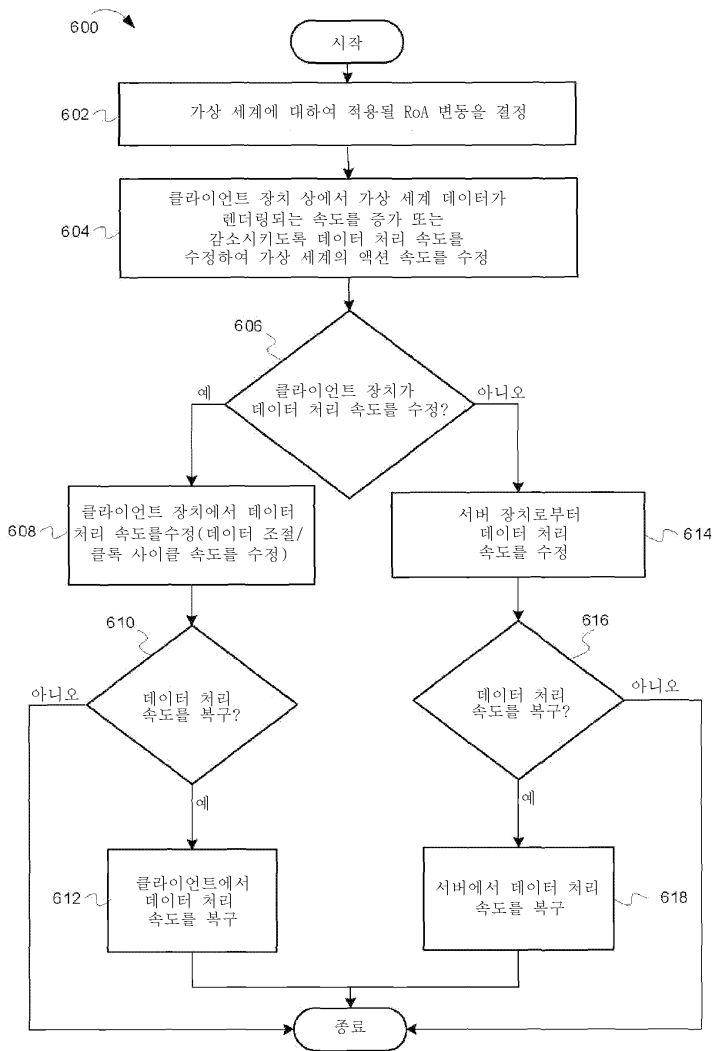
도면4



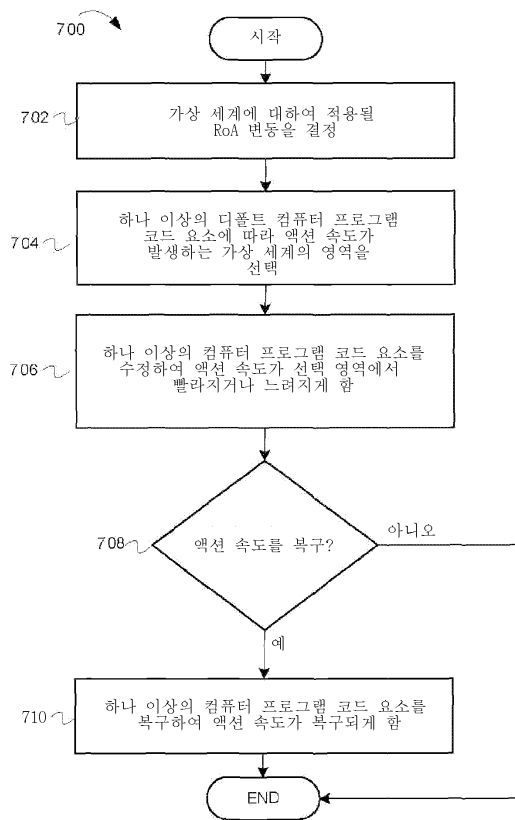
도면5



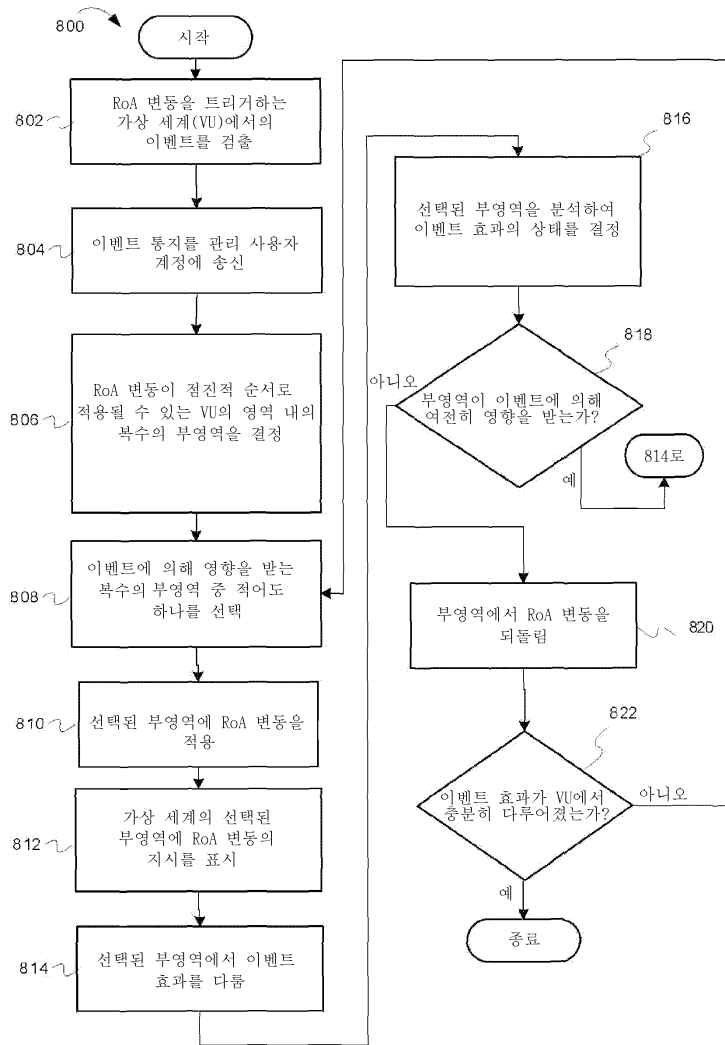
도면6



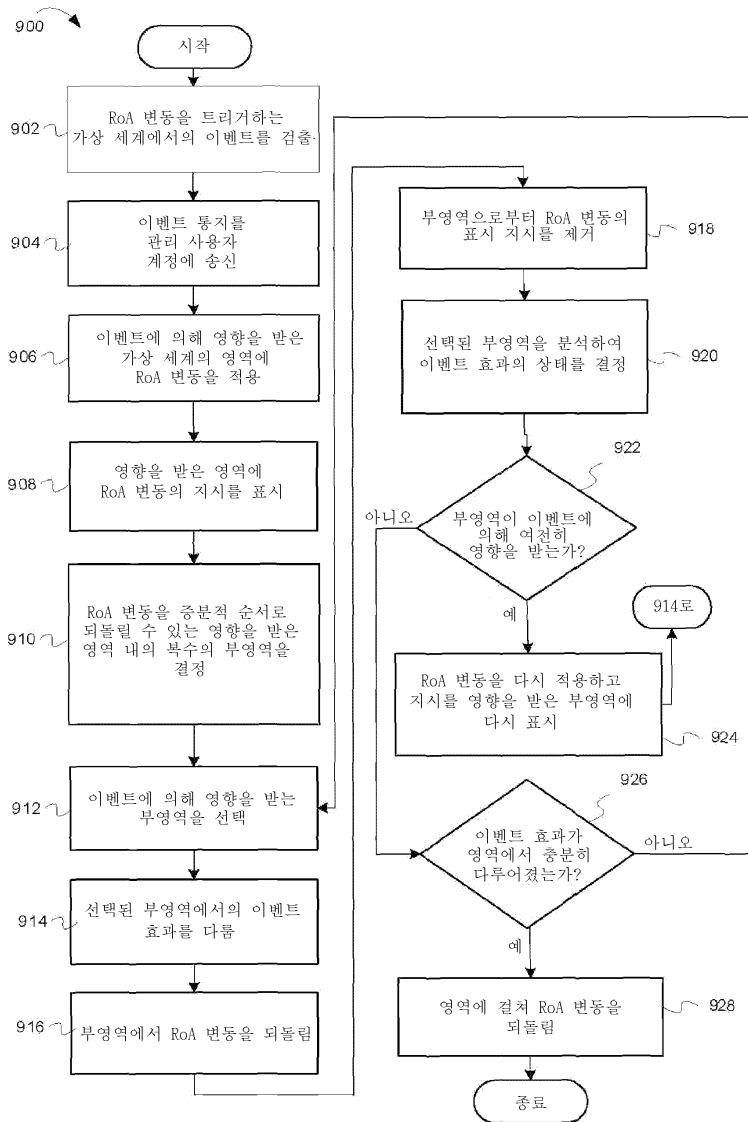
도면7



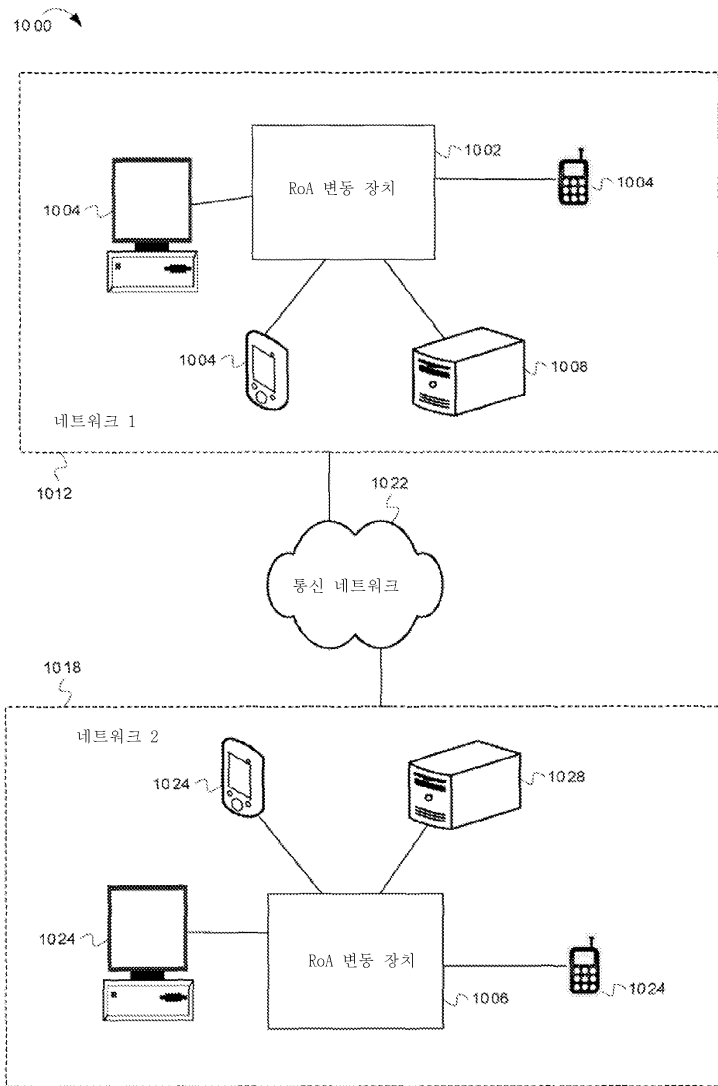
도면8



도면9



도면10



도면11

