



(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.

G06T 15/00 (2006.01)

G06T 11/40 (2006.01)

(11) 공개번호 10-2007-0034412

(43) 공개일자 2007년03월28일

(21) 출원번호 10-2006-0002372

(22) 출원일자 2006년01월09일

심사청구일자 2006년01월09일

(30) 우선권주장 60/719,586 2005년09월23일 미국(US)

(71) 출원인 삼성전자주식회사
경기도 수원시 영통구 매탄동 416

(72) 발명자 이기창
경기 용인시 풍덕천2동 삼성5차아파트 503-1101
김도균
경기 성남시 분당구 분당동 131-3번지 3층
안정환
경기 수원시 영통구 영통동 황골마을2단지아파트 201-208
탁세운
경기 용인시 구성읍 보정리 포스홈타운아파트 210-1202

(74) 대리인 리엔목특허법인

전체 청구항 수 : 총 19 항

(54) 3차원 객체의 효율적 렌더링 방법 및 장치

(57) 요약

본 발명은 3차원 디스플레이를 위해 3차원 객체들을 렌더링(rendering)하는 방법 및 장치에 관한 것이다. 그 방법은 3차원 객체들 각각에 대해 로컬 좌표 연산을 수행하여 움직임 여부를 검출하는 단계; 3차원 객체들 중 움직임이 있는 객체들에 대해서만 글로벌 좌표 연산을 수행하는 단계; 및 좌표 연산이 수행된 3차원 객체들에 대해 렌더링을 수행하는 단계를 포함하는 것을 특징으로 한다.

본 발명에 의하면, 3차원 객체를 렌더링하여 3차원 디스플레이를 하고자하는 경우, 3차원 객체들 중 움직임이 있는 객체들, 더욱 구체적으로는 3차원 객체를 구성하는 트랜스폼(transform)들 중 움직임이 있는 트랜스폼에 대해서만 글로벌 좌표 연산을 수행함으로써, 3차원 객체의 렌더링 시 계산량을 줄일 수 있으며, 그로 인해 등장하는 3차원 객체들의 수 및 움직임이 많은 3차원 게임 등에 있어 3차원 디스플레이의 성능을 크게 증가시킬 수 있다.

대표도

도 4

특허청구의 범위

청구항 1.

3차원 객체를 렌더링하는 방법에 있어서,

3차원 객체들 각각에 대해 로컬 좌표 연산을 수행하여 움직임 여부를 검출하는 단계;

상기 3차원 객체들 중 움직임이 있는 객체들에 대해서만 글로벌 좌표 연산을 수행하는 단계; 및

상기 좌표 연산이 수행된 3차원 객체들에 대해 렌더링을 수행하는 단계를 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 2.

제1항에 있어서, 상기 움직임 여부를 검출하는 단계는

상기 3차원 객체가 동적 객체인지 여부를 판단하는 단계;

동적 객체인 경우, 상기 3차원 객체를 구성하는 트랜스폼(transform)들 중 적어도 하나가 움직임이 있는지 여부를 판단하는 단계; 및

상기 트랜스폼들 중 적어도 하나가 움직임이 있는 경우, 상기 3차원 객체를 움직임이 있는 객체로 검출하는 단계를 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 3.

제2항에 있어서, 상기 움직임 여부를 검출하는 단계는

상기 동적 객체로 판단된 3차원 객체들을 업데이트 리스트(update list)에 포함시키는 단계; 및

상기 업데이트 리스트에 포함된 3차원 객체들 중 움직임을 트랜스폼을 적어도 하나 포함하는 3차원 객체를 리싱크 리스트(resynch list)에 포함시키는 단계를 더 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 4.

제2항에 있어서, 상기 트랜스폼들은

차위(next) 트랜스폼과 자식(child) 트랜스폼으로 연결된 트리 구조를 가지고 상기 3차원 객체를 구성하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 5.

제2항에 있어서, 상기 글로벌 좌표 연산을 수행하는 단계는

상기 움직임이 있는 트랜스폼 및 그의 자식 트랜스폼에 대해서만 글로벌 좌표 연산을 수행하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 6.

제2항에 있어서, 상기 움직임 여부를 검출하는 단계는

상기 움직임이 있는 트랜스폼에 대해서는, 플래그(flag)를 온(on)으로 설정하는 단계를 더 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 7.

제6항에 있어서, 상기 글로벌 좌표 연산을 수행하는 단계는

상기 3차원 객체를 구성하는 트랜스폼들의 플래그를 참조하는 단계; 및

상기 플래그가 온으로 설정되어 있는 트랜스폼 및 그의 자식 트랜스폼에 대해 글로벌 좌표 연산을 수행하는 단계를 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 8.

제1항에 있어서, 상기 3차원 객체들은

일반 객체, 동적 객체 및 관절 운동 객체로 나뉘어 지고,

상기 동적 객체는 상기 일반 객체의 속성을 상속받아 표현되며, 상기 관절 운동 객체는 상기 동적 객체의 속성을 상속받아 표현되는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 9.

3차원 디스플레이를 위해 3차원 객체를 렌더링하는 장치에 있어서,

3차원 객체들 각각에 대해 로컬 좌표 연산을 수행하는 로컬좌표연산부;

상기 로컬 좌표 연산 결과에 따라, 상기 3차원 객체들의 움직임 여부를 검출하는 움직임검출부;

상기 3차원 객체들 중 움직임이 있는 객체들에 대해서만 글로벌 좌표 연산을 수행하는 글로벌좌표연산부; 및

상기 좌표 연산이 수행된 3차원 객체들에 대해 렌더링을 수행하는 렌더링부를 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 장치.

청구항 10.

제9항에 있어서, 상기 움직임검출부는

상기 3차원 객체가 동적 객체인지 여부를 객체판단부; 및

상기 3차원 객체가 동적 객체인 경우, 상기 3차원 객체를 구성하는 트랜스폼(transform)들 중 적어도 하나가 움직임이 있으면 상기 3차원 객체를 움직임이 있는 객체로 검출하는 트랜스폼판단부를 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 장치.

청구항 11.

제10항에 있어서, 상기 움직임검출부는

상기 동적 객체로 판단된 3차원 객체들을 포함하는 업데이트 리스트(update list) 및 상기 업데이트 리스트에 포함된 3차원 객체들 중 움직임이 있는 트랜스폼을 적어도 하나 포함하는 3차원 객체를 포함하는 리싱크 리스트(resynch list)를 저장하는 저장부를 더 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 장치.

청구항 12.

제10항에 있어서, 상기 트랜스폼들은

차위(next) 트랜스폼과 자식(child) 트랜스폼으로 연결된 트리 구조를 가지고 상기 3차원 객체를 구성하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 장치.

청구항 13.

제10항에 있어서, 상기 글로벌좌표연산부는

상기 움직임이 있는 트랜스폼 및 그의 자식 트랜스폼에 대해서만 글로벌 좌표 연산을 수행하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 장치.

청구항 14.

제10항에 있어서, 상기 트랜스폼판단부는

상기 움직임이 있는 트랜스폼에 대해서는, 플래그(flag)를 온(on)으로 설정하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 방법.

청구항 15.

제14항에 있어서, 상기 글로벌좌표연산부는

상기 3차원 객체를 구성하는 트랜스폼들의 플래그를 참조하는 플래그참조부; 및

상기 플래그가 온으로 설정되어 있는 트랜스폼 및 그의 자식 트랜스폼에 대해 글로벌 좌표 연산을 수행하는 연산수행부를 포함하는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 장치.

청구항 16.

제9항에 있어서, 상기 3차원 객체들은

일반 객체, 동적 객체 및 관절 운동 객체로 나뉘어 지고,

상기 동적 객체는 상기 일반 객체의 속성을 상속받아 표현되며, 상기 관절 운동 객체는 상기 동적 객체의 속성을 상속받아 표현되는 것을 특징으로 하는 3차원 객체의 효율적 렌더링 장치.

청구항 17.

3차원 객체를 구성하는 복수의 트랜스폼들 각각을 표현하기 위한 데이터 구조에 있어서,

글로벌 좌표계에서 상기 트랜스폼의 위치에 대한 정보를 가지는 글로벌 좌표 필드;

로컬 좌표계에서 상기 트랜스폼의 위치에 대한 정보를 가지는 로컬 좌표 필드; 및

상기 글로벌 좌표계 또는 로컬 좌표계 상에서, 상기 트랜스폼의 위치가 변경되는지 여부에 대한 정보를 가지는 플래그(flag)를 포함하는 것을 특징으로 하는 3차원 객체를 구성하는 트랜스폼의 데이터 구조.

청구항 18.

3차원 객체를 구성하는 복수의 트랜스폼들 각각을 표현하기 위한 데이터를 기록한 기록매체에 있어서,

글로벌 좌표계에서 상기 트랜스폼의 위치에 대한 정보를 가지는 글로벌 좌표 필드;

로컬 좌표계에서 상기 트랜스폼의 위치에 대한 정보를 가지는 로컬 좌표 필드; 및

상기 글로벌 좌표계 또는 로컬 좌표계 상에서, 상기 트랜스폼의 위치가 변경되는지 여부에 대한 정보를 가지는 플래그(flag) 필드를 가지는 데이터가 기록된 컴퓨터로 읽을 수 있는 기록매체.

청구항 19.

제1항 내지 제8항 중 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램으로 기록한 컴퓨터로 읽을 수 있는 기록매체.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 3차원 디스플레이에 관한 것으로, 보다 상세하게는 3차원 객체들에 대해 로컬 좌표(local coordinate) 연산, 글로벌 좌표(global coordinate) 연산 및 렌더링 작업을 보다 적은 연산량으로 수행하는 방법 및 장치에 관한 것이다.

3차원 디스플레이 기술의 발전에 따라, 게임(game)에 있어서도 3차원 디스플레이 기술이 적용되어 최근에는 3D(Dimension) 게임이 일반화되고 있다. 3D 게임의 경우, 3차원 공간 상에서 움직임을 갖는 많은 수의 3차원 객체들이 존재하고, 각각의 객체들이 특징있는 움직임들을 가진다. 어떠한 객체는 긴 시간 동안 한번의 움직임도 없이 고정된 채로 존재하고, 어떠한 객체는 연속적인 움직임을 가지고 3차원 공간 상에서 존재한다.

상기와 같은 여러 종류의 특징있는 움직임을 가지는 3차원 객체들을 디스플레이하기 위해서는, 각각의 객체들의 움직임에 따른 로컬 좌표(local coordinate) 및 글로벌 좌표(global coordinate) 상의 좌표 변환을 위한 연산과 렌더링 작업이 수행되어야 한다. 상기 로컬 좌표는 각각의 3차원 객체를 중심으로 하는 각 객체만의 좌표이며, 상기 글로벌 좌표는 모든 3차원 객체들에 공통되는 좌표이다.

3차원 공간상에 존재하는 3차원 객체들을 디스플레이하기 위해서는, 모든 3차원 객체들에 대해 상기 로컬 좌표상에서의 객체의 위치 변화를 표현하기 위한 로컬 좌표 연산 및 상기 로컬 좌표 상에서의 위치 변화를 상기 글로벌 좌표상에 반영하는 글로벌 좌표 연산을 수행한 후, 렌더링을 수행하여야한다.

따라서, 매우 많은 3차원 객체들을 포함하는 3D 게임의 경우 상기 로컬 및 글로벌 좌표 연산에 매우 많은 연산량을 필요로 하여 3D 게임의 성능이 저하되는 문제가 있었다.

발명이 이루고자 하는 기술적 과제

본 발명이 이루고자 하는 기술적 과제는, 3차원 디스플레이를 위해 3차원 객체들에 대한 렌더링 작업을 수행함에 있어 상기와 같은 문제점을 해결하기 위해, 글로벌 좌표 연산에 필요한 연산량을 줄이므로써 효율적으로 렌더링을 수행할 수 있도록 하는 3차원 객체의 표현 방법을 제공하고, 그를 이용한 효율적인 3차원 객체의 렌더링 방법 및 장치를 제공하는 것이다.

발명의 구성

상술한 기술적 과제를 해결하기 위한 본 발명에 의한 3차원 객체의 효율적 렌더링 방법은, 3차원 객체들 각각에 대해 로컬 좌표 연산을 수행하여 움직임 여부를 검출하는 단계; 상기 3차원 객체들 중 움직임이 있는 객체들에 대해서만 글로벌 좌표 연산을 수행하는 단계; 및 상기 좌표 연산이 수행된 3차원 객체들에 대해 렌더링을 수행하는 단계를 포함하는 것을 특징으로 한다.

바람직하게는, 상기 움직임 여부를 검출하는 단계는 상기 3차원 객체가 동적 객체인지 여부를 판단하는 단계; 동적 객체인 경우, 상기 3차원 객체를 구성하는 트랜스폼(transform)들 중 적어도 하나가 움직임이 있는지 여부를 판단하는 단계; 및 상기 트랜스폼들 중 적어도 하나가 움직임이 있는 경우, 상기 2차원 객체를 움직임이 있는 객체로 검출하는 단계를 포함한다.

상기 움직임 여부를 검출하는 단계는 상기 동적 객체로 판단된 3차원 객체들을 업데이트 리스트(update list)에 포함시키는 단계; 및 상기 업데이트 리스트에 포함된 3차원 객체들 중 움직임이 있는 트랜스폼을 적어도 하나 포함하는 3차원 객체를 리싱크 리스트(resynch list)에 포함시키는 단계를 더 포함하는 것이 바람직하다.

바람직하게는, 상기 트랜스폼들은 차위(next) 트랜스폼과 자식(child) 트랜스폼으로 연결된 트리 구조를 가지고 상기 3차원 객체를 구성한다.

상기 글로벌 좌표 연산을 수행하는 단계는 상기 움직임이 있는 트랜스폼 및 그의 자식 트랜스폼에 대해서만 글로벌 좌표 연산을 수행하는 것이 바람직하다.

바람직하게는, 상기 움직임 여부를 검출하는 단계는 상기 움직임이 있는 트랜스폼에 대해서는 플래그(flag)를 온(on)으로 설정하는 단계를 더 포함한다.

상기 글로벌 좌표 연산을 수행하는 단계는 상기 3차원 객체를 구성하는 트랜스폼들의 플래그를 참조하는 단계; 및 상기 플래그가 온으로 설정되어 있는 트랜스폼 및 그의 자식 트랜스폼에 대해 글로벌 좌표 연산을 수행하는 단계를 포함하는 것이 바람직하다.

바람직하게는, 상기 3차원 객체들은 일반 객체, 동적 객체 및 관절 운동 객체로 나뉘어 지고, 상기 동적 객체는 상기 일반 객체의 속성을 상속받아 표현되며, 상기 관절 운동 객체는 상기 동적 객체의 속성을 상속받아 표현된다.

상술한 기술적 과제를 해결하기 위한 본 발명에 의한 3차원 객체의 효율적 렌더링 장치는, 3차원 객체들 각각에 대해 로컬 좌표 연산을 수행하는 로컬좌표연산부; 상기 로컬 좌표 연산 결과에 따라, 상기 3차원 객체들의 움직임 여부를 검출하는 움직임검출부; 상기 3차원 객체들 중 움직임이 있는 객체들에 대해서만 글로벌 좌표 연산을 수행하는 글로벌좌표연산부; 및 상기 좌표 연산이 수행된 3차원 객체들에 대해 렌더링을 수행하는 렌더링부를 포함하는 것을 특징으로 한다.

바람직하게는, 상기 움직임검출부는 상기 3차원 객체가 동적 객체인지 여부를 판단하는 객체판단부; 및 상기 3차원 객체가 동적 객체인 경우, 상기 3차원 객체를 구성하는 트랜스폼(transform)들 중 적어도 하나가 움직임이 있으면 상기 3차원 객체를 움직임이 있는 객체로 검출하는 트랜스폼판단부를 포함한다.

상기 움직임검출부는 상기 동적 객체로 판단된 3차원 객체들을 포함하는 업데이트 리스트(update list) 및 상기 업데이트 리스트에 포함된 3차원 객체들 중 움직임이 있는 트랜스폼을 적어도 하나 포함하는 3차원 객체를 포함하는 리싱크 리스트(resynch list)를 저장하는 저장부를 더 포함하는 것이 바람직하다.

바람직하게는, 상기 트랜스폼들은 차위(next) 트랜스폼과 자식(child) 트랜스폼으로 연결된 트리 구조를 가지고 상기 3차원 객체를 구성한다. 상기 글로벌좌표연산부는 상기 움직임이 있는 트랜스폼 및 그의 자식 트랜스폼에 대해서만 글로벌 좌표 연산을 수행하는 것이 바람직하다.

상기 트랜스폼판단부는 상기 움직임이 있는 트랜스폼에 대해서는, 플래그(flag)를 온(on)으로 설정하는 것이 바람직하다. 바람직하게는, 상기 글로벌좌표연산부는 상기 3차원 객체를 구성하는 트랜스폼들의 플래그를 참조하는 플래그참조부; 및 상기 플래그가 온으로 설정되어 있는 트랜스폼 및 그의 자식 트랜스폼에 대해 글로벌 좌표 연산을 수행하는 연산수행부를 포함한다.

바람직하게는, 상기 3차원 객체들은 일반 객체, 동적 객체 및 관절 운동 객체로 나뉘어 지고, 상기 동적 객체는 상기 일반 객체의 속성을 상속받아 표현되며, 상기 관절 운동 객체는 상기 동적 객체의 속성을 상속받아 표현된다.

상술한 기술적 과제를 해결하기 위한 본 발명에 의한 3차원 객체를 구성하는 복수의 트랜스폼들 각각을 표현하기 위한 데이터 구조는, 글로벌 좌표계에서 상기 트랜스폼의 위치에 대한 정보를 가지는 글로벌 좌표 필드; 로컬 좌표계에서 상기 트랜스폼의 위치에 대한 정보를 가지는 로컬 좌표 필드; 및 상기 글로벌 좌표계 또는 로컬 좌표계 상에서, 상기 트랜스폼의 위치가 변경되는지 여부에 대한 정보를 가지는 플래그(flag)를 포함하는 것을 특징으로 한다.

상기 3차원 객체의 효율적 렌더링 방법 및 트랜스폼의 데이터 구조는 바람직하게는 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체로 구현할 수 있다.

이하, 첨부된 도면을 참조하면서 본 발명에 따른 3차원 객체의 효율적 렌더링 방법 및 장치에 대해 상세히 설명한다. 도 1은 3차원 객체(object)를 복수 트랜스폼(transform)들로 표현하는 방법에 대한 실시예를 도시한 것이다.

하나의 3차원 객체는 세부 단위 원소인 트랜스폼의 합으로 구성된다. 도 1에 도시된 바와 같이, 하나의 객체인 사람을 표현하기 위해서는 인체의 각 부분, 즉 몸통, 머리, 오른팔, 왼팔, 오른다리, 왼다리 각각이 트랜스폼이 된다. 3차원 객체 및 각 트랜스폼들의 동적인 움직임을 다양하게 표현하기 위해, 상기 트랜스폼들은 트리(tree) 구조를 가지는 것이 바람직하다.

상기 트랜스폼은 그 특성에 따라, 형태를 표현하는 ShapeTransform, 현재 위치를 외부에 알려주기 위한 HelperTransform 및 관절값 정보를 표현하는 BoneTransform으로 구분하여 사용하는 것이 바람직하다.

다음은 3차원 객체를 표현하는 방법에 대한 일실시예로서, 3차원 객체를 클래스(class) 구조의 코드(code)로 표현한 것이다.

```
class Object
{
    AABB m_bv;
    Transform *m_tree;
    seBool m_dynamic;
    Matrix m_tm;
}
```

상기 m_bv는 상기 객체를 둘러싸는 바운딩 박스(bounding box)를 정의하는 필드(field)이며, 상기 *m_tree는 상기 객체를 구성하는 트리 구조의 트랜스폼들을 정의하는 필드이다. 또한, 상기 m_dynamic은 상기 객체가 움직임이 있는 동적 객체인지, 움직임이 없는 정적 객체인지를 정의하는 필드이다. 상기 m_tm은 글로벌 좌표 상에서 상기 객체의 위치를 나타내는 필드이다.

도 2는 3차원 객체를 구성하는 트랜스폼들의 트리(tree) 구조에 대한 실시예를 도시한 것으로, 트랜스폼들은 서로 차위(next)-자식(child)로 연결되어 있다. 도 2에 도시된 3차원 객체(오브젝트)는 A, B, C, D, E 및 F의 6개의 트랜스폼으로 구성되어 있으며, 서로 차위(next) 트랜스폼 또는 자식(child) 트랜스폼으로 연결되어 있다. 도 2에 도시된 B는 A의 차위(next) 트랜스폼이며, D는 C의 차위(next) 트랜스폼인데, 상기 차위(next) 트랜스폼은 연결된 트랜스폼과 동등한 위치를 가진다.

도 2에 도시된 C는 A의 자식(child) 트랜스폼이며, E는 C의 자식(child) 트랜스폼이고, F는 D의 자식(child) 트랜스폼인데, 상기 자식(child) 트랜스폼은 연결된 트랜스폼에 종속된 위치를 가져, 연결된 트랜스폼에 움직임에 따라 자식(child) 트랜스폼도 움직임을 반드시 가지게 된다.

상기와 같은 차위(next)-자식(child) 구조의 트리로 객체를 구성하면 트랜스폼에 단일 차위 포인터(next pointer)와 자식 포인터(child pointer) 만을 갖도록 구성할 수 있다. 다음은 상기 트랜스폼을 표현하는 방법에 대한 실시예로서, 상기 트랜스폼을 클래스 구조의 코드로 표현한 것이다.

```
class Transform
{
    Tid m_id;
    seUInt32 m_flags;
    Transform *m_next;
    Transform *m_child;

    seMatrix m_global;
    seMatrix m_local;

    seUInt16 m_transformID;
    seUInt16 m_morphID;
}
```

상기 m_id는 상기 트랜스폼의 아이디(IDentification)를 나타내는 필드, m_flags는 계산량 절약을 위해 사용하는 체크 비트(bit)를 나타내는 필드인데 상기 m_flags에 대해서는 이하에서 더욱 자세히 설명하기로 한다. 상기 *m_next는 차위(next) 트랜스폼에 대한 연결 포인터를 나타내는 필드이며, 상기 *m_child는 자식(child) 트랜스폼에 대한 연결 포인터를 나타내는 필드이다. 상기 m_global은 상기 트랜스폼의 글로벌 좌표상의 위치를 나타내는 필드이며, 상기 m_local은 상기 트랜스폼의 로컬 좌표상의 위치를 나타내는 필드이다. 상기 m_transformID는 애니메이션(animation)시 사용되는 상기 트랜스폼의 아이디이며, 상기 m_morphID는 모핑(morphing)시 사용되는 상기 트랜스폼의 아이디이다.

본 발명에서는, 객체의 움직임 특성을 고려하는 객체 지향 방법을 사용하여 3차원객체를 일반 객체(object), 동적 객체(animation object) 및 관절 운동 객체(skinned animation object)로 나누어 표현하는 것이 바람직하다.

상기 일반 객체는 움직임을 갖지 않는 객체이고, 상기 동적 객체는 키프레임 애니메이션(keyframe animation)이나 모핑(morphing)과 같은 단순한 형태의 움직임을 갖는 객체이며, 상기 관절 운동 객체는 관절 운동을 하면서 표면 처리 과정을 포함하는 복잡한 형태의 움직임을 가지는 객체이다. 상기 각 객체는 독립적으로 정의하는 것이 아니라 일반 객체를 정의하고, 상기 정의된 일반 객체를 기반으로 상기 동적 객체를 정의하며, 상기 동적 객체를 기반으로 상기 관절 운동 객체를 정의하는 것이 바람직하다. 상기와 같은 객체 지향 방법을 이용하여 객체 정의 과정에서 중복을 피하고 효율을 높일 수 있다.

다음은 일반 객체를 클래스 구조로 표현한 실시예이다.

```

class Object
{
    AABB m_bv;
    Transform *m_tree;
    seBool m_dynamic;
    Matrix m_tm;
}

```

다음은 일반 객체를 기반으로 상기 일반 객체의 속성을 상속 받고 애니메이션에 필요한 정보를 추가하여 동적 객체를 표현한 실시예이다.

```

class AnimatedObject : public Object
{
    struct Animation
    {
        seReal m_mixCoeff;
        seReal m_animationIndex;
        seReal m_virtualParam;
    };

    anim::FrameManager const *m_frameManager;
    seUInt32 m_maxNumAnimations;
    seUInt32 m_numAnimations;
    Animation *m_animations;
}

```

다음은 동적 객체를 기반으로 상기 동적 객체의 속성을 상속 받고 스킨닝(skinning)에 필요한 정보를 추가하여 관절 운동 객체를 표현한 실시예이다. 상기 스킨닝은 관절체 객체에 피부를 입히는 과정으로서, 관절 운동의 결과 각 관절의 위치 정보가 결정난 후에 수행된다.

```

class SkinnedObject : public AnimatedObject
{
    struct SkinLink
    {
        seBaseSkin *m_skin;
        ShapeTransform *m_shapeTransform;
    };

    seInt32 m_numSkins;
    SkinLink *m_skinLinks;
}

```

도 3a 및 도 3b는 복수의 3차원 객체들을 그룹화하는 방법에 대한 실시예들을 도시한 것으로, 동일한 움직임을 갖거나, 동일한 관리가 필요한 객체들을 그룹화함으로써 객체들에 대한 작업을 효율적으로 수행할 수 있다.

도 3a는 상기 객체들을 그룹화하는 방법에 대한 제1 실시예를 도시한 것으로, 일반 객체가 다른 일반 객체를 포함하도록 하여 객체들을 그룹화하는 방법이 있다. 즉, 상기 제1 실시예는 일반 객체에 또 다른 일반 객체에 대한 정보를 갖도록 하는 것을 허용하여 그룹화를 가능하게 한다. 다음은 상기 객체의 그룹화 방법에 대한 제1 실시예에 따라, 하나의 객체가 다른 객체에 대한 정보를 가지는 경우의 3차원 객체 표현 방법에 대한 실시예를 나타낸 것이다.

```
class Object
{
    AABB m_bv;
    seUInt16 m_num_obj;
    Object *m_obj;
    Transform *m_tree;
    seBool m_dynamic;
    Matrix m_tm;
}
```

상기 m_num_obj가 상기 객체가 속한 객체 그룹 내의 객체들의 수를 나타내는 필드이며, 상기 *m_obj가 상기 그룹을 표현하기 위한 포인터를 나타내는 필드이다. 상기 객체의 그룹화 방법에 대한 제1 실시예는 새로운 표현 정의 없이 기존 객체에 대한 정의에 새로운 필드를 정의하는 것만으로 객체 그룹화가 가능하다는 장점이 있다.

도 3b는 상기 객체들을 그룹화하는 방법에 대한 제2 실시예를 도시한 것으로, 객체들을 포함하는 그룹을 새롭게 정의하고, 상기 그룹을 위한 표현을 새로이 정의해서 사용하는 것이다. 다음은 상기 객체의 그룹화 방법에 대한 제2 실시예에 따라, 새롭게 정의된 그룹을 표현하는 방법에 대한 실시예를 나타낸 것이다.

```
class Group
{
    AABB m_bv;
    seUInt16 m_num_obj;
    Object *m_obj;
    seUInt16 m_num_group;
    Group *m_group;
    seBool m_dynamic;
    Matrix m_tm;
}
```

상기 m_bv는 상기 그룹을 둘러싸는 바운딩 박스를 정의하는 필드이며, 상기 m_num_obj는 상기 그룹에 속하는 객체들의 수를 정의하는 필드이다. 상기 *m_obj는 상기 그룹을 표현하기 위한 포인터 필드이다. 상기 m_num_group는 상기 그룹 내의 하위 그룹의 수를 나타내는 필드이며, 상기 m_dynamic는 정적 또는 동적 그룹의 표현을 위한 필드이고, 상기 m_tm은 상기 그룹의 위치를 나타내는 필드이다.

상기 객체의 그룹화 방법에 대한 제2 실시예는 그룹의 사용을 일반 객체와는 구분지어 새로운 구조를 제시하여 사용하여, 그룹 사용시 혼돈을 막을 수 있고 외형적인 트리 구조상에서 그룹의 사용을 명확히 할 수 있다.

도 4는 본 발명에 따른 3차원 객체를 효율적으로 렌더링하는 장치의 구성에 대한 실시예를 블록도로 도시한 것으로, 도시된 렌더링 장치는 로컬좌표연산부(400), 움직임검출부(410), 글로벌좌표연산부(420) 및 렌더링부(430)를 포함하여 이루어진다. 도 4에 도시된 렌더링 장치의 동작을 도 6에 도시된 본 발명에 따른 3차원 객체의 효율적 렌더링 방법을 나타내는 흐름도와 결부시켜 설명하기로 한다.

로컬좌표연산부(400)는 모든 3차원 객체들 각각에 대해, 움직임에 따른 로컬 좌표 상에서의 위치 변화를 표현하기 위한 로컬 좌표 연산을 수행한다(400단계). 상기 3차원 객체의 움직임은 인공 지능(Artificial Intelligence) 또는 사용자 입력 등에 따라 결정되며, 상기 로컬 좌표는 각각의 3차원 객체를 중심으로 하는 각 객체만의 좌표이다. 로컬좌표연산부(400)의 로컬 좌표 연산에 의해, 모든 객체들 각각을 구성하는 트랜스폼들 중 움직임이 있는 트랜스폼의 로컬 좌표 위치가 변화된다.

움직임검출부(410)는 상기 로컬 좌표 연산의 결과에 따라 상기 모든 3차원 객체들 중 움직임이 있는 객체들, 즉 포함하는 트랜스폼들 중 적어도 하나가 움직임이 있는 3차원 객체를 검출한다(610단계).

도 5는 움직임검출부(410)의 구성에 대한 실시예를 블록도로 도시한 것으로, 도시된 움직임검출부(410)는 객체판단부(500), 저장부(510) 및 트랜스폼판단부(510)를 포함하여 이루어진다. 도 5에 도시된 움직임검출부(410)의 동작을 도 7에 도시된 움직임이 있는 객체를 검출하는 방법에 대한 실시예를 나타내는 흐름도와 결부시켜 설명하기로 한다.

객체판단부(500)는 3차원 객체가 동적 객체인지 정적 객체인지 여부를 판단한다(700단계). 상기 동적 객체 여부를 판단은, 상기의 객체를 표현하는 방법에 대한 실시예에 나타나는 필드들 중 움직임이 있는 동적 객체인지, 움직임이 없는 정적 객체인지를 정의하는 m_dynamic을 참조하여 판단하는 것이 바람직하다.

상기 판단 결과 상기 3차원 객체가 정적 객체인 경우, 상기 3차원 객체 및 상기 3차원 객체를 구성하는 트랜스폼들은 모두 움직임이 없는 것으로 판단된다.

상기 3차원 객체가 동적 객체인 경우, 상기 3차원 객체는 저장부(510)에 저장된 업데이트 리스트(update list)에 포함된다(710단계). 트랜스폼판단부(520)는 상기 업데이트 리스트에 포함된 3차원 객체들에 대해서만, 상기 3차원 객체를 구성하는 트랜스폼들 각각이 움직임이 있는지 여부를 판단한다(720단계).

다음은 상기 트랜스폼을 표현하는 방법에 대한 일실시예이다.

```
class Transform
{
    Tid m_id;
    seUInt32 m_flags;
    Transform *m_next;
    Transform *m_child;

    seMatrix m_global;
    seMatrix m_local;

    seUInt16 m_transformID;
    seUInt16 m_morphID;
}
```

상기 트랜스폼이 움직임이 있는 경우, 상기 로컬 좌표 연산에 의해 로컬 좌표 상의 트랜스폼의 위치를 행렬로 표시하는 m_local이 변경되고, 그와 동시에 상기 트랜스폼을 포함하는 3차원 객체가 움직임이 있는 것으로 판단되어 저장부(510)에 저장된 리싱크 리스트(resynch list)에 포함된다(730단계).

상기와 같은 과정에 의해, 모든 객체들 중 움직임이 있는 객체들, 즉 적어도 하나의 움직임이 있는 트랜스폼을 포함하는 3차원 객체들은 상기 리싱크 리스트에 포함되게 된다.

또한, 트랜스폼판단부(520)는 움직임이 있어 상기 m_local이 변경되는 트랜스폼에 대해서는 상기 m-flags를 온(ON)으로 체크하는 것이 바람직하다. 그에 따라, 움직임이 있는 트랜스폼들은 상기 m-flags가 온(ON)으로 체크되어 있게 된다.

글로벌좌표연산부(420)는 3차원 객체가 움직임이 있는 객체인지 여부를 확인하여(620단계), 모든 3차원 객체들 중 움직임이 있는 객체에 대해서만 글로벌 좌표 연산을 수행한다(630단계). 상기 글로벌 좌표 연산은 상기 로컬 좌표 상에서의 객체 또는 트랜스폼의 움직임을 디스플레이 화면 상에 반영하기 위한 연산이다. 상기 글로벌 좌표 연산에 의해, 움직임이 있는 트랜스폼에 대해서는 글로벌 좌표 상의 상기 트랜스폼의 위치를 행렬로 표시하는 m-global이 변경되게 된다.

글로벌좌표연산부(420)는 움직임이 있는 객체들 각각에 대해, 그를 구성하는 트랜스폼들 중 움직임이 있는 트랜스폼들에 대해서만 글로벌 좌표 연산을 수행하는 것이 바람직하다. 글로벌좌표연산부(420)는 상기 m-flags를 참조하여 트랜스폼이 움직임이 있는 것인지 여부를 확인하는 것이 바람직하다. 즉, 글로벌좌표연산부(420)가 상기 m-flags를 참조하여, 상기 m-flags가 온(ON)으로 체크되어 있는 트랜스폼에 대해서만 글로벌 좌표 연산을 수행하는 것이 바람직하다. 또한, 어느 하나의 트랜스폼이 움직임이 있으면, 그에 종속되는 자식 트랜스폼도 움직임이 있으므로, 글로벌좌표연산부(420)는 상기 m-flags가 온(ON)으로 체크되어 있는 트랜스폼 및 그의 자식 트랜스폼에 대해 글로벌 좌표 연산을 수행하는 것이 바람직하다.

도 2에 도시된 트리 구조의 객체 표현 방법을 참조하여 상기 글로벌좌표연산부(420)의 동작에 대한 실시예를 설명하기로 한다. 글로벌좌표연산부(420)는 B, A, D, F, C, E와 같은 순서로 상기 m-flags의 참조 및 글로벌 좌표 연산을 수행하는 것이 바람직하다.

도 2에 도시된 트랜스폼들 중 D 트랜스폼의 m-flags가 온(ON)으로 체크되어 있다면, 글로벌좌표연산부(420)는 B, A 트랜스폼에 대해 m-flags가 오프(OFF)인 것을 확인하고 글로벌 좌표 연산을 수행하지 않으며, D 트랜스폼에 대해 m-flags가 온(ON)인 것을 확인하고 글로벌 좌표 연산을 수행하고, 상기 D의 자식 트랜스폼인 F 트랜스폼에 대해서도 글로벌 좌표 연산을 수행한다. 글로벌좌표연산부(420)는 C, E 트랜스폼에 대해 m-flags가 오프(OFF)인 것을 확인하고 글로벌 좌표 연산을 수행하지 않는다. 즉, 글로벌좌표연산부(420)는 움직임이 있는 객체(오브젝트)를 구성하는 6개의 트랜스폼들 중 2개의 D와 F 트랜스폼에 대해서만 글로벌 좌표 연산을 수행한다.

상기 움직임이 있는 객체들에 대해서, 바람직하게는 움직임이 있는 트랜스폼들에 대해 글로벌 좌표 연산이 완료되면, 렌더링부(430)는 3차원 객체들을 디스플레이 화면에 디스플레이하기 위해 광원, 상기 연산된 글로벌 좌표상의 객체의 위치 및 색상등을 반영하는 렌더링을 수행한다(640단계).

정적 객체 또는 상기 로컬 좌표 상에서 움직임이 없는 3차원 객체는 글로벌 좌표 상에서도 위치 변화가 없으므로, 상기 움직임이 없는 객체들에 대해서는 글로벌 좌표 연산을 수행하지 하니하고 바로 렌더링을 수행한다(640단계).

본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다.

이상 본 발명의 바람직한 실시예에 대해 상세히 기술하였지만, 본 발명이 속하는 기술분야에 있어서 통상의 지식을 가진 사람이라면, 첨부된 청구범위에 정의된 본 발명의 정신 및 범위에 벗어나지 않으면서 본 발명을 여러 가지로 변형 또는 변경하여 실시할 수 있음을 알 수 있을 것이다. 따라서, 본 발명의 앞으로의 실시예들의 변경은 본 발명의 기술을 벗어날 수 없을 것이다.

발명의 효과

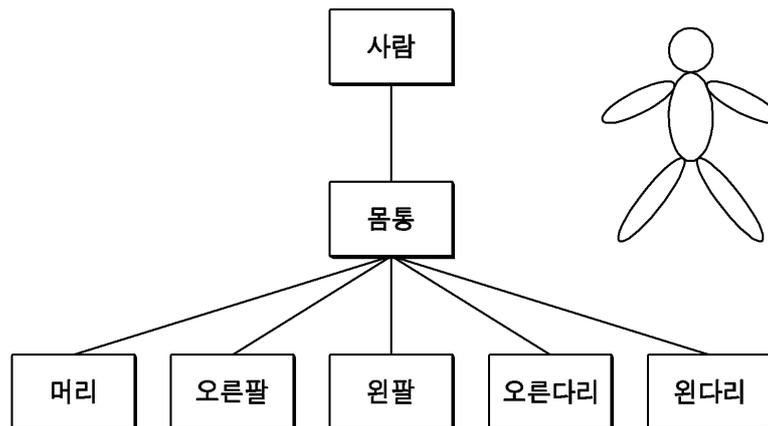
상술한 바와 같이, 본 발명에 의한 3차원 객체의 효율적 렌더링 방법 및 장치에 의하면, 3차원 객체들 중 움직임이 있는 객체들, 더욱 구체적으로는 3차원 객체를 구성하는 트랜스폼(transform)들 중 움직임이 있는 트랜스폼에 대해서만 글로벌 좌표 연산을 수행함으로써, 3차원 객체의 렌더링 시 계산량을 줄일 수 있으며, 그로 인해 등장하는 3차원 객체들의 수 및 움직임이 많은 3차원 게임 등에 있어 3차원 디스플레이의 성능을 크게 증가시킬 수 있다.

도면의 간단한 설명

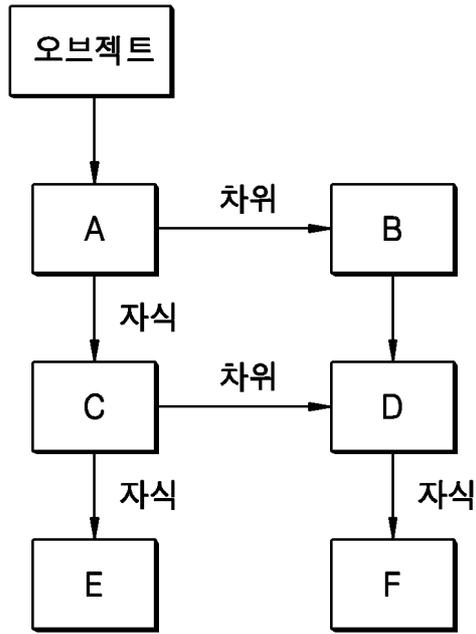
- 도 1은 3차원 객체(object)를 복수 트랜스폼(transform)들로 표현하는 방법에 대한 실시예를 도시한 도면이다.
- 도 2는 3차원 객체를 구성하는 트랜스폼들의 트리(tree) 구조에 대한 실시예를 도시한 도면이다.
- 도 3a 및 도 3b는 복수의 3차원 객체들을 그룹화하는 방법에 대한 실시예들을 도시한 도면이다.
- 도 4는 본 발명에 따른 3차원 객체를 효율적으로 렌더링하는 장치의 구성에 대한 실시예를 나타내는 블록도이다.
- 도 5는 도 4의 움직임검출부에 대한 실시예를 나타내는 블록도이다.
- 도 6은 본 발명에 따른 3차원 객체의 효율적 렌더링 방법을 나타내는 흐름도이다.
- 도 7은 도 6의 움직임이 있는 객체를 검출하는 단계에 대한 실시예를 나타내는 흐름도이다.

도면

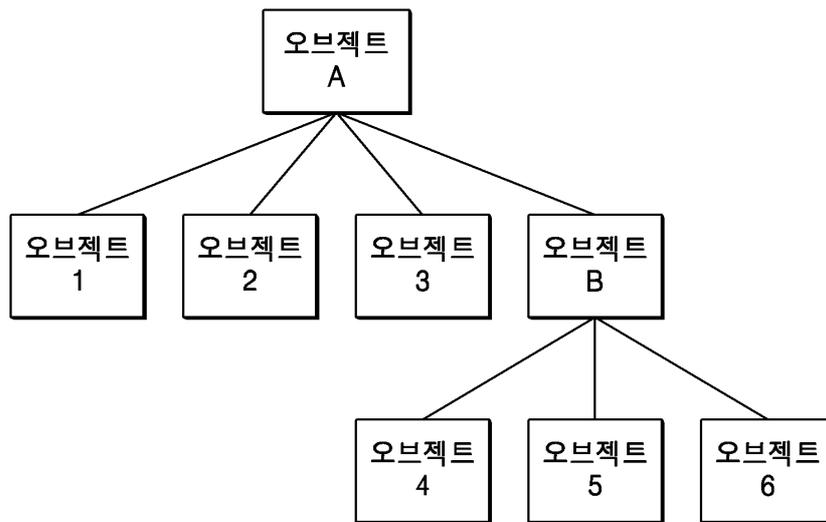
도면1



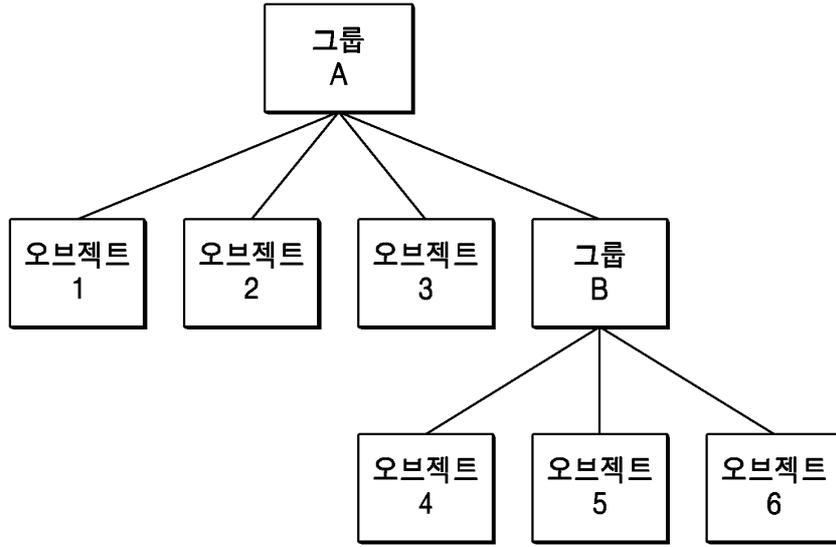
도면2



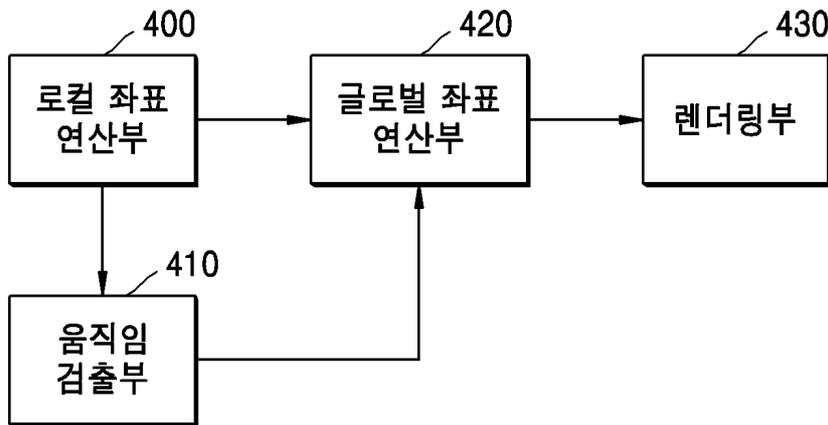
도면3a



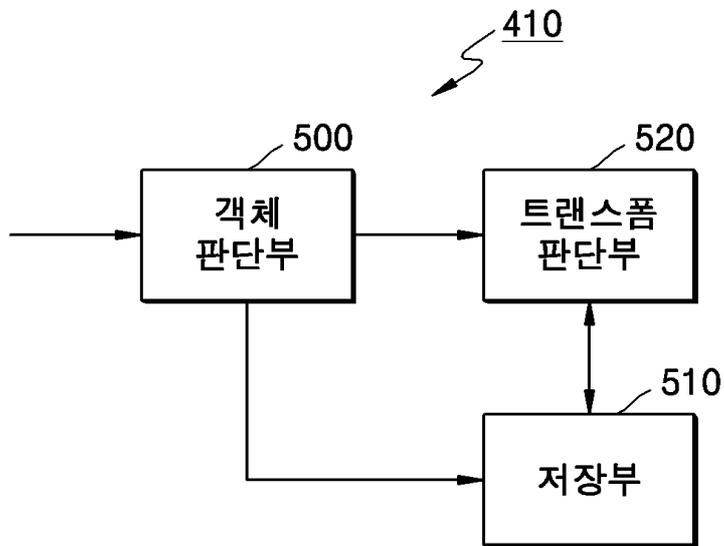
도면3b



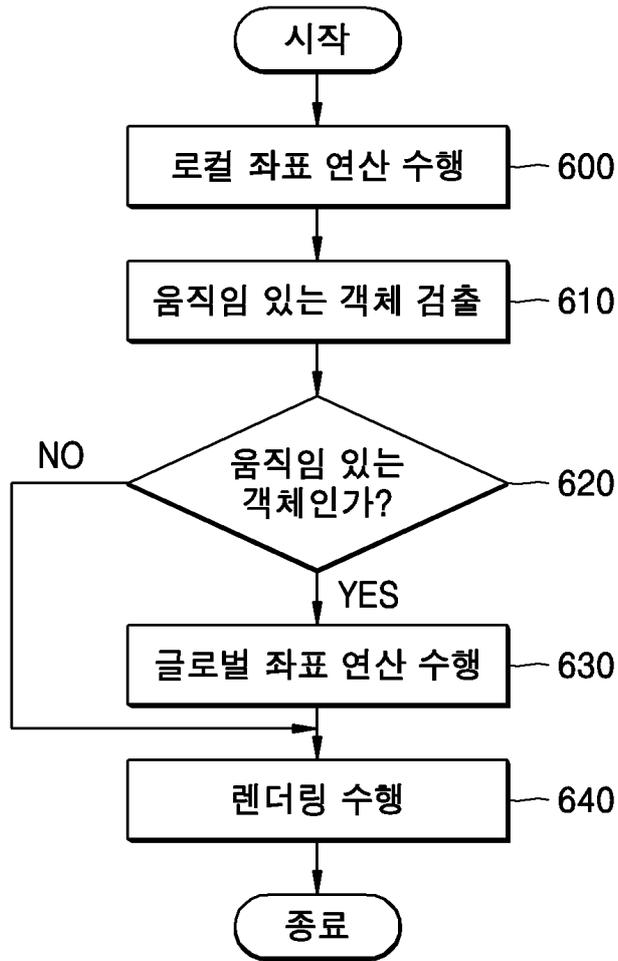
도면4



도면5



도면6



도면7

