

(52) CPC특허분류
G06T 15/08 (2013.01)

명세서

청구범위

청구항 1

랜더링을 수행하기 위한 콘텐츠를 분석하는 단계;

상기 콘텐츠의 분석 결과에 기반하여 해상도, 컬러스페이스 및 프레임 오프셋에 대한 적어도 하나의 프로그램 모듈의 호환성 읽기 노드를 설정하는 문자열을 생성하는 단계;

외부로부터 수신한, 상기 콘텐츠를 상기 적어도 하나의 프로그램 모듈에 적용하기 위한 적어도 하나의 기즈모(gizmo)를 포함하여 구성되는 노드 에셋에 따른 문자열을 생성하는 단계;

상기 생성된 문자열에 따라서 상기 랜더링을 수행하도록 처리하는 랜더링 데이터를 생성하는 단계;

상기 랜더링 데이터를 지정된 랜더팜(render farm)으로 전송하는 단계; 및

상기 랜더링 데이터 및 상기 콘텐츠에 기반하여 상기 적어도 하나의 프로그램으로 랜더링된 콘텐츠를 상기 지정된 랜더팜으로부터 수신하는 단계;를 포함하되,

상기 랜더링을 수행하기 위한 콘텐츠를 분석하는 단계는, 상기 랜더링에 사용되는 다수의 명령어를 호출 및 상기 콘텐츠와 관련된 프로젝트, 팀 및 샷 정보 중 적어도 일부를 확인하는 것을 특징으로 하는, 이미지 처리 방법.

청구항 2

제1항에 있어서,

상기 랜더링을 수행하기 위한 콘텐츠를 분석하는 단계는, 상기 문자열을 기록하는 파이프라인을 생성하는 것을 특징으로 하는, 이미지 처리 방법.

청구항 3

제1항에 있어서,

상기 랜더링을 수행하기 위한 콘텐츠를 분석하는 단계는, 상기 랜더링에 사용되는 입력된 다수의 명령어를 호출하는 것을 특징으로 하는, 이미지 처리 방법.

청구항 4

제1항에 있어서,

상기 콘텐츠의 영상 처리를 위한 적어도 하나의 문자열을 생성하는 단계는, 입력되는 다수의 명령어 각각에 대하여 지정된 영상 처리를 수행하도록 처리하는 문자열을 생성하는 것을 특징으로 하는, 이미지 처리 방법.

청구항 5

제1항에 있어서,

상기 외부로부터 수신한 랜더링 형식에 따른 문자열은, 기 생성된 상기 콘텐츠의 영상 처리를 위한 적어도 하나의 문자열 중 적어도 일부를 변경하는 것을 특징으로 하는, 이미지 처리 방법.

청구항 6

제1항에 있어서,

상기 랜더링 데이터에 따라서 랜더링된 콘텐츠를 획득하는 단계;를 더 포함하는, 이미지 처리 방법.

청구항 7

삭제

청구항 8

적어도 하나의 명령어를 입력하는 입력부;

상기 입력된 명령어 각각에 따라서, 랜더링을 수행하기 위한 콘텐츠를 분석하고, 상기 콘텐츠의 분석 결과에 기반하여 해상도, 컬러스페이스 및 프레임 오프셋에 대한 적어도 하나의 프로그램 모듈의 호환성 읽기 노드를 설정하는 문자열을 생성하고, 상기 콘텐츠를 상기 적어도 하나의 프로그램 모듈에 적용하기 위하여 외부로부터 수신한 적어도 하나의 기즈모(gizmo)를 포함하여 구성되는 노드 에셋에 따른 문자열을 생성하는 로직생성부;

상기 생성된 문자열에 따라서 상기 랜더링을 수행하도록 처리하는 랜더링 데이터를 생성하는 데이터생성부; 및

상기 랜더링 데이터를 지정된 랜더팜(render farm)으로 전송하고, 상기 랜더링 데이터 및 상기 콘텐츠에 기반하여 상기 적어도 하나의 프로그램으로 랜더링된 콘텐츠를 상기 지정된 랜더팜으로부터 수신하는 통신부를 포함하되,

상기 분석 결과는, 상기 랜더링에 사용되는 다수의 명령어를 호출 및 상기 콘텐츠와 관련된 프로젝트, 팀 및 샷 정보 중 적어도 일부를 확인 결과를 포함하는, 이미지 처리 장치.

청구항 9

제8항에 있어서,

상기 로직생성부는, 상기 문자열을 기록하는 파이프라인을 생성하는, 이미지 처리 장치.

청구항 10

제8항에 있어서,

상기 로직생성부는, 상기 랜더링에 사용되는 입력된 다수의 명령어를 호출하는, 이미지 처리 장치.

청구항 11

제8항에 있어서,

상기 로직생성부는, 상기 입력부를 통해서 입력되는 입력되는 다수의 명령어 각각에 대하여 지정된 영상 처리를 수행하도록 처리하는 문자열을 생성하는, 이미지 처리 장치.

청구항 12

제8항에 있어서,

상기 로직생성부는, 상기 외부로부터 수신한 랜더링 형식에 따른 문자열에 기반하여, 기 생성된 상기 콘텐츠의 영상 처리를 위한 적어도 하나의 문자열 중 적어도 일부를 변경하는, 이미지 처리 장치.

청구항 13

제8항에 있어서,

상기 렌더링 데이터에 따라서 렌더링된 콘텐츠를 획득하는 렌더링부;를 더 포함하는, 이미지 처리 장치.

청구항 14

삭제

발명의 설명

기술 분야

[0001] 본 발명은 이미지 처리 방법 및 그 장치에 관한 것으로, 상세하게는 렌더링을 수행하기 위한 이미지 처리 방법 및 그 장치에 관한 것이다.

배경 기술

[0003] 3차원 영상 처리 기술이 발전함으로써, 실제와 유사한 수준의 영상을 제작할 수 있게 되었다. 이에 따라서, 영상 처리에 수행되는 작업량은 방대해지고, 공정 또한 복잡해 짐으로써 여러 명의 작업자가 협업하여 하나의 영상을 완성하게 되는 일이 발생한다. 3차원 영상 처리에 있어서, 다양한 작업자가 동일한 콘텐츠에 대한 영상 처리를 수행하는 경우, 적용되는 파이프라인이 다르거나, 명령어 체계가 호환되지 않아 정상적으로 작업을 수행할 수 없는 문제가 발생한다.

[0004] 특허문헌 제10-2011-0020411호는 기존의 그래픽 처리 장치의 파이프라인 구조의 변경을 최소화함으로써, 기존의 그래픽 구동 응용 프로그램 인터페이스(API)와 호환이 가능한 그래픽 처리 장치 및 이를 이용한 그래픽 처리 방법을 개시하고 있다. 하지만, 이러한 방법에 있어서, 콘텐츠에 처리되는 작업의 파이프라인을 최소화 하기 위한 방법을 개시하고 있을 뿐, 다양한 작업자를 통해서 수행되는 그래픽 작업에 있어서, 동일한 콘텐츠에 대하여 처리되는 작업의 파이프라인이 일관되지 못하여 발생하는 문제의 처리가 용이하지 않고, 작업자들의 협업에 있어 효율적인 업무 처리가 제한되고 있다.

[0005] 더하여, 이러한 문제를 해결하기 위한 방법으로, 다양한 고객사 또는 작업자는, 체계화된 영상 처리 및 렌더링을 수행할 수 있는 툴을 사용하고 있다. 하지만, 이러한 툴을 사용하기 위하여는 고비용의 라이선스를 필요로 하며, 작업장의 개별 컴퓨터마다 라이선스를 요구하는 경우, 일부 기능만을 사용하는 경우에도 높은 라이선스 비용을 지출하게 되어 비용의 낭비가 발생하게 된다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 제10-2011-0020411호(공개특허공보)

발명의 내용

해결하려는 과제

[0008] 따라서, 본 발명의 다양한 실시예에 따르면, 고비용의 라이선스를 비용을 지출하는 프로그램 모듈을 사용하지 않고, 영상 처리를 위한 렌더링 데이터를 생성하기 위한 이미지 처리 방법 및 그 장치를 제공할 수 있다.

[0009] 본 발명의 다양한 실시예에 따르면, 영상 처리 공정에서 다양한 작업자가 영상 처리에 대한 파이프라인의 큰 변경 없이 콘텐츠를 처리하기 위한 렌더링 데이터를 제공하는 이미지 처리 방법 및 그 장치를 제공할 수 있다.

[0010] 본 발명의 다양한 실시예에 따르면, 콘텐츠에 대한 영상 처리 연산을 수행함에 있어서, 시스템적으로 의존되지 않는 단일 명령어로 구성되는 이미지 처리 방법 및 그 장치를 제공할 수 있다.

과제의 해결 수단

- [0012] 본 발명의 일 실시예에 따르면, 이미지 처리 방법은, 렌더링을 수행하기 위한 콘텐츠를 분석하는 단계; 상기 콘텐츠의 영상 처리를 위한 적어도 하나의 문자열을 생성하는 단계; 외부로부터 수신한 렌더링 형식에 따른 문자열을 생성하는 단계; 및 상기 생성된 문자열에 따라서 상기 렌더링을 수행하도록 처리하는 렌더링 데이터를 생성한다.
- [0013] 다양한 실시예에 따르면, 상기 렌더링을 수행하기 위한 콘텐츠를 분석하는 단계는, 상기 영상 처리를 위하여 생성되는 문자열들을 기록하는 파이프라인을 생성할 수 있다.
- [0014] 다양한 실시예에 따르면, 상기 렌더링을 수행하기 위한 콘텐츠를 분석하는 단계는, 상기 렌더링에 사용되는 다수의 명령어를 호출할 수 있다.
- [0015] 다양한 실시예에 따르면, 상기 콘텐츠의 영상 처리를 위한 적어도 하나의 문자열을 생성하는 단계는, 입력되는 다수의 명령어 각각에 대하여 지정된 영상 처리를 수행하도록 처리하는 문자열을 생성할 수 있다.
- [0016] 다양한 실시예에 따르면, 상기 외부로부터 수신한 렌더링 형식에 따른 문자열은, 기 생성된 상기 콘텐츠의 영상 처리를 위한 적어도 하나의 문자열 중 적어도 일부를 변경할 수 있다.
- [0017] 다양한 실시예에 따르면, 상기 이미지 처리 방법은, 상기 렌더링 데이터에 따라서 렌더링된 콘텐츠를 획득하는 단계;를 더 포함할 수 있다.
- [0018] 다양한 실시예에 따르면, 상기 이미지 처리 방법은, 상기 렌더링 데이터를 적어도 하나의 렌더팜으로 전송하는 단계; 및 상기 렌더링 데이터에 기반하여 렌더링된 콘텐츠를 상기 적어도 하나의 렌더팜으로부터 수신하는 단계;를 더 포함할 수 있다.
- [0019] 본 발명의 일 실시예에 따르면, 이미지 처리 장치는, 적어도 하나의 명령어를 입력하는 입력부; 상기 입력된 명령어 각각에 따라서, 렌더링을 수행하기 위한 콘텐츠를 분석하고, 상기 콘텐츠의 영상 처리를 위한 적어도 하나의 문자열을 생성하고, 외부로부터 수신한 렌더링 형식에 따른 문자열을 생성하는 로직생성부; 및 상기 생성된 문자열에 따라서 상기 렌더링을 수행하도록 처리하는 렌더링 데이터를 생성하는 데이터생성부;를 포함한다.
- [0020] 다양한 실시예에 따르면, 상기 로직생성부는, 상기 영상 처리를 위하여 생성되는 문자열들을 기록하는 파이프라인을 생성할 수 있다.
- [0021] 다양한 실시예에 따르면, 상기 로직생성부는, 상기 렌더링에 사용되는 다수의 명령어를 호출할 수 있다.
- [0022] 다양한 실시예에 따르면, 상기 로직생성부는, 상기 입력부를 통해서 입력되는 입력되는 다수의 명령어 각각에 대하여 지정된 영상 처리를 수행하도록 처리하는 문자열을 생성할 수 있다.
- [0023] 다양한 실시예에 따르면, 상기 로직생성부는, 상기 외부로부터 수신한 렌더링 형식에 따른 문자열에 기반하여, 기 생성된 상기 콘텐츠의 영상 처리를 위한 적어도 하나의 문자열 중 적어도 일부를 변경할 수 있다.
- [0024] 다양한 실시예에 따르면, 상기 렌더링 데이터에 따라서 렌더링된 콘텐츠를 획득하는 렌더링부;를 더 포함할 수 있다.
- [0025] 다양한 실시예에 따르면, 상기 렌더링 데이터를 적어도 하나의 렌더팜으로 전송하고, 상기 렌더링 데이터에 기반하여 렌더링된 콘텐츠를 상기 적어도 하나의 렌더팜으로부터 수신하는 렌더링부;를 더 포함할 수 있다.

발명의 효과

- [0027] 본 발명의 다양한 실시예에 따르면, 지정된 명령어를 이용하여 렌더링을 위한 파이프라인을 생성함으로써, 렌더링을 수행하는 프로젝트 종류에 관계없이 동일한 파이프 라인을 형성할 수 있다.
- [0028] 본 발명의 다양한 실시예에 따르면, 다양한 장치와의 호환성을 가지는 명령어 및 문자열을 이용하여 콘텐츠를 렌더링하기 위한 렌더링 데이터를 생성함으로써, 다양하게 요구되는 이미지 처리에 대하여 효과적으로 대처할 수 있다.
- [0029] 상술한 바와 같이, 높은 라이선스 비용을 지출하는 프로그램 모듈을 사용하지 않고 렌더링 데이터를 생성하는 방법 및 장치를 제공함으로써, 프로그래밍에 있어 자유도를 확보하고, 또한, 과도하게 지출되는 비용을 절약할 수 있다.

도면의 간단한 설명

- [0031] 도 1은 본 발명의 일 실시예에 따른 이미지 처리 장치 및 이미지 처리 장치를 포함하는 이미지 처리 시스템의 개략적인 구성을 도시한다.
- 도 2는 본 발명의 일 실시예에 따른 이미지 처리 장치에서 로컬 입력을 통해서 확인되는 명령어에 기반하여 렌더링 데이터를 생성하는 흐름도이다.
- 도 3은 본 발명의 일 실시예에 따른 이미지 처리 장치에서 외부 렌더링 형식을 적용하여 파이프라인을 형성하는 동작의 흐름도이다.
- 도 4는 본 발명의 일 실시예에 따른 이미지 처리 장치에서 콘텐츠 렌더링을 수행하는 동작의 흐름도이다.
- 도 5는 본 발명의 일 실시예에 따른 이미지 처리 장치에서 외부 렌더링 형식의 적용 결과를 출력하는 도면이다.
- 도 6 내지 도 8은 본 발명의 일 실시예에 따른 이미지 처리 장치에서 영상 처리에 대한 문자열 생성에 사용되는 다양한 명령어를 도시한다.
- 도 9는 본 발명의 일 실시예에 따른 이미지 처리 장치에서 렌더링 데이터에 포함되는 파이프라인의 전체적인 흐름을 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0032] 이하, 첨부된 도면을 참조하여 본 발명의 다양한 실시예를 설명한다.
- [0033] 본 발명의 실시예는 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예가 도면에 예시되고 관련된 상세한 설명이 기재되어 있다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경 및/또는 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 도면의 설명과 관련하여, 유사한 구성요소에 대해서는 유사한 참조 부호를 사용할 수 있다.
- [0034] 본 발명의 다양한 실시예 가운데 사용될 수 있는 “포함한다” 또는 “포함할 수 있다” 등의 표현은 개시된 해당 기능, 동작 또는 구성요소 등의 존재를 가리키며, 추가적인 하나 이상의 기능, 동작 또는 구성요소 등을 제한하지 않는다.
- [0035] 또한, 본 발명의 다양한 실시예에서, “포함하다” 또는 “가지다” 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0036] 본 발명의 다양한 실시예에서 '또는' 등의 표현은 함께 나열된 단어들의 어떠한, 그리고 모든 조합을 포함한다. 예를 들어, 'A 또는 B'는, A를 포함할 수도, B를 포함할 수도, 또는 A 와 B 모두를 포함할 수도 있다.
- [0037] 본 발명의 다양한 실시예에서 '제1', '제2', '첫째' 또는 '둘째' 등의 표현들이 본 발명의 다양한 구성 요소들을 수식할 수 있지만, 해당 구성요소들의 순서 및/또는 중요도 등을 한정하지 않는다. 또한, 상기 표현들은 한 구성요소를 다른 구성요소와 구분 짓기 위해 사용될 수 있다.
- [0038] 본 발명의 다양한 실시예에서 어떤 구성요소가 다른 구성요소에 '연결되어' 있다거나 '접속되어' 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 '직접 연결되어' 있다거나 '직접 접속되어' 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해될 수 있어야 할 것이다.
- [0039] 본 발명의 다양한 실시예에서 '실질적으로', '예를 들어'와 같은 표현에 따라 설명은 인용된 특성, 변수, 또는 값과 같이 제시한 정보들이 정확하게 일치하지 않을 수 있고, 허용 오차, 측정 오차, 측정 정확도의 한계와 통상적으로 알려진 기타 요인을 비롯한 변형과 같은 효과로 본 발명의 다양한 실시예에 따른 발명의 실시 형태를 한정하지 않아야 할 것이다.
- [0040] 본 발명의 다양한 실시예에서 사용한 용어는 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 용어에 있어서, 단수의 표현은 문맥상 명백하게 다른 것으로 명시하지 않는 한 복수의 표

현을 포함할 수 있다.

- [0041] 또한 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지는 것으로 해석되어야 하며, 본 발명의 실시예에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석하거나 또는 축소 해석하지 않아야 한다.
- [0042] 본 발명의 일 실시예에 따르면, 이미지는, 영화 제작을 위한 시각적 특수효과(visual effect, visual FX, VFX)가 처리되기 위한 하나의 스틸 이미지(still image, 또는 이미지 프레임) 및/또는 다수의 이미지로 구성된 무빙 이미지(moving image)를 포함한다. 이 때, 하나의 스틸 이미지로 제공되는 컷(cut 또는, frame) 및/또는 둘 이상의 연결(또는 나열)되는 스틸 이미지로 제공되는 샷(shot 또는 씬, scene) 중 적어도 일부를 포함하는 콘텐츠(contents)를 정의할 수 있다. 여기에서, 콘텐츠의 샷 또는 씬을 구성하는 이미지 중 적어도 일부는 스틸 이미지에 한정하지 않고, 무빙 이미지로 구현될 수 있다.
- [0043] 도 1은 본 발명의 일 실시예에 따른 이미지 처리 장치(101) 및 이미지 처리 장치(101)를 포함하는 이미지 처리 시스템의 개략적인 구성을 도시한다. 도 1을 참조하면, 이미지 처리 시스템(100)은 이미지 처리 장치(101)를 포함하고, 추가로 외부 장치(랜더팜, 103) 및/또는 저장부(105)를 포함할 수 있다. 여기에서, 저장부(105)는 이미지 처리 장치(101)의 외부에서 이미지 처리 장치(101)와 연결된 것으로 도시되어 있지만, 이에 한정하지 않고, 이미지 처리 장치(101)의 내부에 포함되어 구성될 수도 있다.
- [0044] 이미지 처리 장치(101)는 입력부(110), 로직생성부(120), 데이터생성부(130) 및 렌더링부(140) 중 하나 이상을 포함한다. 여기에서, 각각의 구성은 작업 및 기능으로 구분되는 모듈로 구성될 수 있고, 이 때, 모듈은 소프트웨어 모듈 또는 하드웨어 모듈로 구성될 수 있다.
- [0045] 입력부(110)는 이미지 처리 장치(101)의 외부(예: 사용자, 및/또는 외부 장치)로부터의 입력을 수신할 수 있다. 예를 들어, 입력부(110)는 키보드, 키패드, 트랙패드, 터치 패널, 카메라 등의 입력 인터페이스 중 적어도 하나를 포함하여 구성될 수 있다.
- [0046] 일 실시예에 따르면, 입력부(110)는 키보드 또는 키 패드로 제공될 수 있고, 선택되는 키와 매칭되는 문자를 입력할 수 있다. 여기에서, 입력부를 통해서 입력되는 문자는, 언어, 특수문자, 기호, 숫자, 방향키, 기능키 등의 다양한 형태의 약속들 중 적어도 하나로 정의될 수 있다.
- [0047] 또한, 문자들은 둘 이상의 결합에 기반하여 약속된 문자열을 형성할 수 있다. 그리고, 문자열과 매칭되는 동작(또는 기능을) 수행하도록 처리하는 명령어를 형성할 수 있다.
- [0048] 다양한 실시예에 따르면, 입력부(110)는 터치 패널로 제공될 수 있고, 터치 및 드래그에 기반하여 문자, 문자열, 명령어 등을 입력 받을 수 있다. 다양한 실시예에 따르면, 입력부(110)는 카메라로 제공될 수 있고, 카메라를 통해서 촬영되는 이미지에 포함되는 문자, 문자열 및/또는 명령어를 추출하거나, 촬영된 객체의 형태 및 움직임에 기반하여 문자, 문자열 및/또는 명령어를 확인할 수 있다.
- [0049] 이하 설명에서, 입력부(110)을 통해서 입력되는 문자, 문자열 및/또는 명령어를 로컬입력(local input)으로 표현할 수 있다. 입력부(110)는 수신한 로컬입력을 로직생성부(120)로 전송할 수 있다.
- [0050] 로직생성부(120)는 입력부(110)로부터 수신하는 로컬입력에 따라서 콘텐츠를 처리하기 위한 문자열을 생성할 수 있다. 예를 들어, 로직생성부(120)는 로컬입력에 따라서 콘텐츠의 형식을 설정(예: 입력, 수정 및/또는 변경)하기 위한 다양한 문자열을 생성할 수 있다.
- [0051] 여기에서, 콘텐츠의 형식은, 호환성, 해상도, 컬러스페이스(color space), 렌더링, 저장, 썸네일(thumbnail), 전송 중 적어도 하나에 대한 설정값을 포함하여 구성된다. 이 때, 콘텐츠의 형식에 대한 설정값은 문자열을 통해서 처리될 수 있다.
- [0052] 더하여, 로직생성부(120)는 콘텐츠의 형식을 설정하기 위한 문자열을 생성함에 있어서, 입력부(110)로부터 수신하는 로컬입력에 따라서 처리하는 것에 한정하지 않고, 기 저장된 설정 값을 호출하여 콘텐츠의 형식을 설정하기 위한 문자열을 생성할 수 있다.
- [0053] 예를 들어, 로직생성부(120)는, 콘텐츠 형식에 대한 설정 값이 기 저장된 데이터(이하, 외부형식)를 확인할 수 있다. 여기에서, 외부형식은, 이미지 처리 장치(101)의 외부로부터 수신할 수 있고, 또는 입력부(110)를 통해서 기 입력된 로컬입력에 기반하여 생성된 데이터일 수 있다.

- [0054] 로직생성부(120)는 외부형식이 호출된 것을 확인하는 경우, 외부형식에 포함된 설정 값에 기반하여 콘텐츠의 형식을 설정하기 위한 문자열을 생성할 수 있다.
- [0055] 데이터생성부(130)는 로직생성부(120)를 통해서 생성된 문자열에 따라서 콘텐츠를 처리하기 위한 랜더링 데이터를 생성한다. 이 때, 데이터생성부(130)는, 랜더링된 콘텐츠가 저장될 위치를 확인하고, 랜더링된 콘텐츠를 해당 위치에 저장하도록 처리하는 랜더링 데이터를 생성할 수 있다.
- [0056] 여기에서, 콘텐츠가 저장되는 위치는 데이터베이스(또는 저장부)의 특정 주소, 예를 들어, 디렉터리(directory)일 수 있고, 윈도우(windows)의 폴더 형태로 제공될 수도 있다.
- [0057] 이하 설명에서, 콘텐츠가 저장되는 위치에 한정하지 않고, 이미지 처리 장치(101)의 내부 및/또는 외부의 저장부에서 데이터가 저장되는 위치에 디렉터리(또는 폴더)의 개념을 적용할 수 있다.
- [0058] 데이터생성부(130)는 로컬입력에 포함된 명령어에 기반하여 콘텐츠의 랜더링을 수행할 장치를 확인한다. 예를 들어, 데이터생성부(130)는 이미지 처리 장치(101)에서 랜더링을 수행하도록 처리하는 로컬입력을 확인하는 경우 랜더링 데이터를 랜더링부(140)로 전송하고, 랜더팜(render farm, 103)에서 랜더링을 수행하도록 처리하는 로컬입력을 확인하는 경우 랜더링 데이터를 랜더팜(103)으로 전송한다.
- [0059] 상술한 바와 같이, 이미지 처리 장치(101)의 입력부(110)는 로컬입력을 수신하고, 로직생성부(120)는 수신한 로컬입력에 따라서 문자열을 생성하고, 데이터생성부(130)는 생성된 문자열을 이용하여 콘텐츠를 처리하기 위한 랜더링 데이터를 생성하고, 랜더링부(140) 및/또는 랜더팜(103)을 통해서 생성된 랜더링 데이터에 기반하여 콘텐츠를 생성하도록 처리하는 동작을 수행한다.
- [0060] 상술한 바와 같이, 이미지 처리 장치(101)는 콘텐츠를 처리하기 위한 랜더링 데이터를 생성함에 있어서, 입력부(110)를 통해서 입력되는 로컬 입력에 기반하여 지정된 트리(tree) 또는 파이프(pipe, |) 구조에 따른 연속된 문자열을 포함할 수 있다.
- [0061] 랜더팜(103)은, 콘텐츠의 2차원 그래픽 처리 또는 3차원 그래픽 처리를 수행하기 위한 외부 장치로 제공될 수 있다. 이 때, 랜더팜(103)은 이미지 처리 장치(101)로부터 콘텐츠를 랜더링하기 위한 적어도 하나의 문자열을 포함하는 랜더링 데이터를 수신할 수 있고, 수신한 랜더링 데이터에 포함된 파이프라인에 따라서 콘텐츠를 랜더링할 수 있다.
- [0062] 저장부(130)는 이미지 처리 장치(101)를 통해서 랜더링을 수행하기 위하여 처리되는 명령어 및 명령어에 대응하는 문자열 등을 포함할 수 있다. 또한, 입력되는 명령어에 따라서 형성되는 파이프라인은, 명령어에 따라서 호출되는 룩업테이블(lookup table, lut)을 포함할 수 있다. 그리고, 생성된 파이프라인은 랜더링 데이터로 저장부(130)에 저장될 수 있다. 더하여, 저장부(130)는 랜더링된 콘텐츠가 저장될 수 있다.
- [0063] 도 2를 참조하면, 이미지 처리 장치(101)는 로컬 입력을 통해서 확인되는 명령어에 기반하여 콘텐츠의 영상 처리를 수행하기 위한 랜더링 데이터를 생성할 수 있다.
- [0064] 예를 들어, 이미지 처리 장치(101)는 입력되는 명령어 “dils”를 확인하는 경우, 지정된 콘텐츠의 시퀀스 데이터를 분석하고, 분석된 시퀀스 데이터에 대하여 호환성 읽기의 노드 형성에 필요한 정보를 호출하는 문자열을 생성한다(S201). 여기에서, 호환성 읽기는, 특정 프로그램 모듈(예: 뉴크, nuke)과의 호환성 읽기(nuke read, nkread)로 표현될 수 있다.
- [0065] 이하 설명에서, 호환성을 언급하는 경우 상술한 바와 같이, 뉴크(nuke) 프로그램 모듈을 위한 호환성 처리를 표현할 수 있다. 하지만, 이에 한정하지 않고, 다양한 프로그램에 대한 호환성 처리를 수행할 수 있음은 자명하다.
- [0066] 일 실시예에 따르면, 명령어 “dils”는 “\$ dils /project/path/source”와 같이 처리하고자 하는 시퀀스 데이터가 포함된 주소와 함께 입력될 수 있다.
- [0067] 여기에서, 시퀀스 데이터는, 콘텐츠를 구성하는 복수의 연속된 이미지(예: 무빙 이미지)에 대응되는 데이터일 수 있다. 이미지 처리 장치(101)는, 명령어 “dils”의 입력이 확인되는 디렉터리에 포함된 시퀀스 데이터를 분석하고, 각각의 시퀀스 데이터에 대한 경로(path), 시퀀스 데이터의 시작프레임(start frame), 종료프레임(end frame), 시퀀스 데이터를 포함하는 콘텐츠의 길이 등의 정보 중 적어도 일부를 분석 결과로 출력할 수 있다. 여기에서, 데이터(예: 각각의 시퀀스 데이터)의 경로는, 저장부에서 데이터를 포함하는 주소 전체로 정의될 수 있다.

- [0068] 이미지 처리 장치(101)는, 콘텐츠의 분석 결과를 출력함에 있어서, 동일한 데이터(예: 시퀀스 데이터) 이름에 대하여 다양한 형태의 확장자를 포함하는 경우, 데이터량(예: 비트)가 높은 우선 순위에 따라서 나열할 수 있다.
- [0069] 또한, 디렉터리 내에 동일한 데이터 이름의 dpx, exr, jpg 데이터를 포함하는 경우, exr, dpx, jpg의 순서로 정렬할 수 있다.
- [0070] 이미지 처리 장치(101)는 이미지 처리를 위한 연산을 수행함에 있어서, 상술한 바와 같이 나열된 출력에 기반하여, 가장 정보가 많은(비트가 큰) 이미지를 이용하여 연산을 수행할 수 있다.
- [0071] 또한, 명령어 “dils”는 “\$ dils | sort | xargs dils”와 같이 재귀적으로 사용될 수도 있다.
- [0072] 이미지 처리 장치(101)는 입력된 명령어 “dils”를 확인하는 경우, 렌더링 과정에서 수행되는 영상 처리에 대한 문자열을 기록하는 파이프라인을 생성하고, 도 6에 도시된 바와 같이, 콘텐츠 편집(또는 영상 처리)를 위한 문자열 생성에 사용되는 다양한 명령어들을 활성화 및 호출하기 위한 문자열을 생성한다.
- [0073] 예를 들어, 이미지 처리 장치(101)는 명령어 “nkread”의 입력을 확인하는 경우, 명령어 “dils(601)”에 기반하여 출력된 분석 결과를 이용하여 호환성 읽기 노드(예: nuke read node)의 문자열을 생성할 수 있다.
- [0074] 일 실시예에 따르면, 명령어 “nkread(603)”는 “\$ dils | nkread [colorspace] [offset]”와 같이 처리하고자 하는 컬러스페이스(color space) 및/또는 프레임 오프셋(offset)과 함께 입력될 수 있다.
- [0075] 예를 들어, 프레임 오프셋의 코드는 “101, 1001 / 연산 시작 프레임 수”와 같이 입력될 수 있고, 컬러스페이스의 코드는 “lin, sRGB, rec709, cin, gamma1.8 및/또는 gamma2.2”와 같이 입력될 수 있다.
- [0076] 이미지 처리 장치(101)는 명령어 “nkread(603)”에 대하여 컬러스페이스 코드(또는 예약어)를 확인하는 경우, 소스를 호출하는 시점에 확인된 컬러스페이스를 적용하여 소스를 로딩할 수 있다.
- [0077] 또한, 이미지 처리 장치(101)는, 입력된 명령어 “nkread(603)”에 대하여 문자열을 생성하는 시점에 명령어 “pathinfo(611)”를 이용하여 해당 콘텐츠의 정보, 예를 들어, 프로젝트 정보, 팀 정보 및/또는 샷 정보를 수집하고, 수집된 정보를 문자열에 함께 포함할 수 있다.
- [0078] 여기에서, 명령어 “nkread(603)”에 따라서 시퀀스 데이터에 처리되는 컬러스페이스는 코드에 따라서 시퀀스 데이터의 확장자를 구분하여 처리되도록 설정할 수 있다.
- [0079] 이미지 처리 장치(101)는 명령어 “version(605)”의 입력을 확인하는 경우, 해당 디렉터리에서 콘텐츠의 버전을 확인한다. 이 때, 콘텐츠에 둘 이상의 버전이 확인되는 경우, 해당 버전 중 적어도 하나를 출력할 수 있다. 예를 들어, 해당 콘텐츠에 포함된 시퀀스 데이터에서 2개의 버전을 확인하는 경우, 해당 시퀀스 데이터에 대한 각각의 버전을 출력할 수 있다.
- [0080] 일 실시예에 따르면, 명령어 “version(605)”는 “\$ dils | version”과 같이 입력될 수 있다.
- [0081] 이미지 처리 장치(101)는 입력된 명령어 “timecode(607)”를 확인하는 경우, 명령어 “dils(601)”를 통해서 생성되는 stdout 정보를 통해서 타임 코드를 확인한다.
- [0082] 일 실시예에 따르면, 명령어 “time”는 “\$ dils | timecode”와 같이 입력될 수 있다.
- [0083] 이미지 처리 장치(101)는 입력된 명령어 “caminfo(609)”를 확인하는 경우, 콘텐츠의 시각적 특수효과를 처리하기 위하여 필요한 카메라 스펙(specification) 및/또는 이미지 처리 장치(101)에 구비된(또는 연결된) 카메라의 정보를 출력한다.
- [0084] 일 실시예에 따르면, 명령어 “caminfo(609)”는 “\$ caminfo”와 같이 입력될 수 있다.
- [0085] 이미지 처리 장치(101)는 입력된 명령어 “pathinfo(611)”를 확인하는 경우, 명령어 입력이 확인된 디렉터리의 경로(또는 지정된 경로)에 기반하여 포함된 지정된 콘텐츠(예: 시퀀스 데이터)의 프로젝트 정보, 팀 정보 및/또는 샷 정보를 연동한다.
- [0086] 일 실시예에 따르면, 명령어 “pathinfo(611)”는 “\$ pathinfo”와 같이 독립적으로 입력될 수 있고, 또는 “\$ pathinfo targetpath”와 같이 대상 경로를 함께 입력할 수 있다.
- [0087] 이미지 처리 장치(101)는 상술한 바와 같이 호출된 명령어에 기반하여 렌더링 데이터를 생성하기 위한 콘텐츠 분석을 수행할 수 있다. 여기에서, 문자열 생성을 위한 명령어, 예를 들어, 명령어 “dils” 및 “nkread”를

제외한 나머지 명령어들, 예를 들어, 명령어 “version(605)”, “timecode(607)”, “caminfo(609)” 및 “pathinfo(611)” 중 적어도 하나의 명령어는 렌더링 데이터를 생성하기 위한 동작 중 입력이 확인되는 시점에 해당 기능을 수행할 수 있다.

- [0088] 이미지 처리 장치(101)는 명령어 “dils(601)” 를 확인하는 경우, 상술한 명령어에 한정하지 않고, 콘텐츠(예: 시퀀스 데이터)의 렌더링을 위한 다양한 명령어, 예를 들어, 도 7 및 도 8에 도시된 바와 같이, “nkreformat(701)”, “nk2aces(703)”, “nklut(705)”, “nkgizmo(707)”, “nkffmpeg(801)”, “nkwrite(803)”, “nkmov(805)”, “nksave(807)”, “nkthumb(809)”, “nkrender(811)”, “makealf(813)”, “tractor(815)”, “echo(821)” 및 “>(823)” 중 적어도 하나의 명령어를 함께 호출할 수 있다.
- [0089] 다시 도 2를 참조하면, 이미지 처리 장치(101)는 입력되는 명령어에 기반하여 시퀀스 데이터의 렌더링에 대한 문자열을 생성한다(S203). 일 실시예에 따르면, 이미지 처리 장치(101)는 로컬입력에 포함된 명령어를 확인할 수 있고, 입력된 명령어에 대응되는 콘텐츠 처리에 대한 문자열을 생성할 수 있다.
- [0090] 예를 들어, 이미지 처리 장치(101)는, 명령어 “nkreformat(701)” 의 입력을 확인하는 경우, 호환성을 위한 해상도 리포맷(reformat)의 문자열을 생성한다.
- [0091] 일 실시예에 따르면, 명령어 “nkreformat(701)” 는 “\$ dils | nkread | nkreformat 1920x1080” 와 같이 설정하고자 하는 해상도와 함께 입력될 수 있고, “\$ dils | nkread | nkreformat 1920” 와 같이 설정하고자 하는 해상도의 가로(또는 세로) 방향의 사이즈와 함께 입력될 수 있다.
- [0092] 이 때, 이미지 처리 장치(101)는 입력된 방향(가로 또는 세로)의 해상도 및 해상도에 대하여 기 저장된 룩업 테이블(lookup table)에 따라서 콘텐츠의 해상도를 설정하는 문자열을 생성할 수 있다.
- [0093] 이미지 처리 장치(101)는, 명령어 “nk2aces(703)” 의 입력을 확인하는 경우, 컬러스페이스 변경을 위한 문자열을 생성한다. 예를 들어, 이미지 처리 장치(101)는, 명령어 “nk2aces(703)” 에 대하여, 렌더링을 위한 파이프라인의 표준 입력 데이터(예: stdin)에 ACES 컬러스페이스를 sRGB로 변경하도록 처리하는 문자열을 생성할 수 있다.
- [0094] 일 실시예에 따르면, 일 실시예에 따르면, 명령어 “nk2aces(703)” 는 “\$ dils | nkread | nk2aces” 와 같이 입력될 수 있다.
- [0095] 이미지 처리 장치(101)는, 명령어 “nklut(705)” 의 입력을 확인하는 경우, 콘텐츠의 렌더링에 참조되는 룩업 테이블의 노드를 추가하기 위한 문자열을 생성한다.
- [0096] 일 실시예에 따르면, 명령어 “nklut(705)” 는 “\$ dils | nkread | nklut [lutpath] [incolorspace] [outcolorspace]” 와 같이 룩업테이블의 주소(lutpath)를 포함하고, 추가로 인풋컬러스페이스(incolorspace) 아웃풋컬러스페이스(outcolorspace)와 함께 입력될 수 있다.
- [0097] 이 때, 이미지 처리 장치(101)는, 명령어 “nklut(705)” 에 대하여, 렌더링을 위한 파이프라인의 표준 입력 데이터(예: stdin)에 룩업테이블의 경로를 설정하도록 처리하는 문자열 및/또는 인풋컬러스페이스, 아웃풋컬러스페이스를 설정하도록 처리하는 문자열을 생성할 수 있다.
- [0098] 여기에서, 단계(S203)을 통해서 입력되는 명령어, 예를 들어, “nkreformat(701)”, “nk2aces(703)” 또는 “nklut(705)” 는 상술한 콘텐츠 분석(S201)에 입력된 명령어 “dils(601)” 및 “nkread(603)” 의 입력 이후에 선택적으로 사용될 수 있고, 입력된 순서에 따라서 해당 명령어에 대한 기능이 처리되도록 문자열을 생성할 수 있다.
- [0099] 다시 도 2를 참조하면, 이미지 처리 장치(101)는, 렌더링을 위한 파이프라인을 생성함에 있어서, 외부로부터 수신한 렌더링 형식에 기반하여 콘텐츠를 처리하기 위한 문자열을 생성할 수 있다(S205). 여기에서, 외부로부터 수신한 렌더링 형식(이하, 외부 렌더링 형식)은, 예를 들어, 뉴크 프로그램 모듈에 적용하기 위한 노드 예셋(기즈모, gizmo)를 포함할 수 있다.
- [0100] 일 실시예에 따르면, 명령어 “nkgizmo(707)” 는 “\$ dils | nkread | nkgizmo” 와 같이 “\$ dils | nkread” 와 함께 입력될 수 있고, 명령어 “| nkwrite | nksave” 를 포함하면, 호출된 기즈모를 콘텐츠에 적용하도록 처리할 수 있다.
- [0101] 이하, 도 3의 다양한 실시예를 참고하여, 도 2의 단계(S205)를 통하여 외부 렌더링 형식을 적용하여 파이프라인을 형성하는 동작을 상세하게 설명할 수 있다.

- [0102] 이미지 처리 장치(101)는 명령어 “nkgizmo(707)”의 입력을 확인하는 경우, 디렉터리에 포함된 또는 명령어를 통하여 연결된 외부 렌더링 형식을 확인한다(S301)
- [0103] 이 때, 이미지 처리 장치(101)는 해당 콘텐츠에 대한 외부 렌더링 형식(예: gizmo)을 확인하지 못하는 경우, 기 생성된 파이프라인 및/또는 이후 입력되는 명령어에 대한 문자열 생성을 수행한다(S307).
- [0104] 반면, 이미지 처리 장치(101)는 해당 콘텐츠에 대한 외부 렌더링 형식을 확인하는 경우, 외부 렌더링 형식의 구조 및/또는 포함된 명령어를 분석(S303)하고, 분석 결과에 기반하여 기 생성된 파이프라인을 수정하거나, 또는 외부 렌더링 형식에 포함된 명령어에 해당되는 파이프라인을 생성(S305)한다.
- [0105] 예를 들어, 상술한 바와 같이 명령어 “nkgizmo(707)”가 단독으로 입력된 경우, 외부 렌더링 형식에 포함된 명령어를 확인하고, 적어도 하나의 다른 명령어와 함께 입력되는 경우, 해당 명령어와 기 생성된 파이프라인에 포함된 명령어를 비교할 수 있다.
- [0106] 이 때, 이미지 처리 장치(101)는 외부 렌더링 형식에 포함된 명령어를 기 생성된 파이프라인에 적용함에 있어서, 오류가 발생하는지 여부를 확인할 수 있고, 오류가 발생하는 경우, 도 5에 도시된 바와 같이 해당 오류와 관련된 정보를 출력(500)할 수 있다.
- [0107] 예를 들어, 이미지 처리 장치(101)는 외부 렌더링 형식에 포함된 명령어에 기반하여 생성하는 문자열이 기 생성된 파이프라인에 포함된 문자열과 중복되는 경우, 중복되는 문자열 또는 해당 명령어와 관련된 정보를 출력할 수 있다.
- [0108] 여기에서, 명령어 “nkgizmo(707)”의 입력은 단계(S203)를 통해서 시퀀스 데이터의 처리에 대한 명령어(710)가 문자열로 생성된 이후에 입력(및 생성)되는 것으로 설명하고 있지만, 이에 한정하지 않고, 명령어 “dils(601)” 및 “nkread(603)”의 입력 이후 언제든지 입력될 수 있다.
- [0109] 다시 도 2를 참조하면, 이미지 처리 장치(101)는 입력된 명령어에 기반하여 파이프라인에 기반하는 렌더링 데이터를 생성한다(S207). 이 때, 이미지 처리 장치(101)는 입력되는 명령어에 따른 지정된 렌더링 방식을 수행하도록 문자열을 생성할 수 있다.
- [0110] 예를 들어, 이미지 처리 장치(101)는 명령어 “nkffmpeg(801)”의 입력을 확인하는 경우, ffmpeg를 이용하는 mov 렌더링을 수행하기 위하여 호환성 쓰기 노드(nuke write node)를 생성하도록 처리하는 문자열을 생성할 수 있다.
- [0111] 일 실시예에 따르면, 명령어 “nkffmpeg(801)”는 “\$ dils | nkread | nkffmpeg”와 같이 입력될 수 있다. 여기에서, 이미지 처리 장치(101)는 ffmpeg를 이용하여 mov 렌더링을 수행하는 경우, 렌더링된 콘텐츠의 기본 값은, 23.98fps, sRGB 컬러스페이스로 코딩될 수 있다.
- [0112] 명령어 “nkffmpeg(801)”는 지원되는 fps, 예를 들어, “23.98, 24, 25, 2.97, 30fps” 및/또는 컬러스페이스, 예를 들어, “lin, sRGB, rec709, log, gamma1.8 및/또는 gamma2.2”의 값 중 적어도 일부 인수를 옵션으로 함께 입력될 수 있다.
- [0113] 이미지 처리 장치(101)는 기 생성된 파이프라인에서 표준 입력 데이터에 기록된 정보를 확인하는 경우 해당 정보를 포함할 수 있다. 일 실시예에 따르면, 명령어 “nkffmpeg(801)”는 리눅스 시스템을 위한 문자열을 생성할 수 있다.
- [0114] 이미지 처리 장치(101)는 명령어 “nkwrite(803)”의 입력을 확인하는 경우, 시퀀셜 렌더링을 수행하기 위한 문자열을 생성할 수 있다.
- [0115] 일 실시예에 따르면, 명령어 “nkwrite(803)”는 “\$ dils | nkread | nkwrite ../test/test.####.tga”, “\$ dils | nkread | nkwrite ./folder/file.####.jpg” 또는 “\$ dils | nkread | nkwrite tga”와 같이 확장자 및/또는 컬러스페이스의 인수를 옵션으로 함께 입력할 수 있다.
- [0116] 이미지 처리 장치(101)는 명령어 “nkwrite(803)”와 함께 입력되는 인수에 따라서 렌더링되는 콘텐츠의 셋팅 값을 변경할 수 있다. 이 때, 인수가 함께 입력되지 않는 경우, 기본 값은 jpeg 및 sRGB로 코딩될 수 있다.
- [0117] 여기에서, 명령어 “nkwrite(803)”의 확장자는, 예를 들어, “dpx, jpg, exr, png, tga, tif”가 제공되고, 컬러스페이스는, 예를 들어, “lin, srgb, rec709, log, gamma1.8, gamma2.2”가 제공될 수 있다.
- [0118] 이미지 처리 장치(101)는 기 생성된 파이프라인에서 표준 입력 데이터에 기록된 정보를 확인하는 경우 해당 정

보를 포함할 수 있다.

- [0119] 이미지 처리 장치(101)는 명령어 “nkmov(805)”의 입력을 확인하는 경우, mov 렌더링을 수행하기 위한 문자열을 생성할 수 있다.
- [0120] 일 실시예에 따르면, 명령어 “nkmov(805)”는 “\$ dils | nkread | nkmov”와 같이 입력될 수 있다. 여기에서, 이미지 처리 장치(101)는 mov 렌더링을 수행하는 경우, 렌더링된 콘텐츠의 기본 값은, 23.98fps, photojpeg, sRGB 컬러스페이스로 코딩될 수 있다.
- [0121] 명령어 “nkmov(805)”는 지원되는 fps, 예를 들어, “23.98, 24, 25, 2.97, 30, 60fps”, 컬러스페이스, 예를 들어, “lin, srgb, rec709, cin, gamma1.8, gamma2.2, logc” 및/또는 gamma2.2 및/또는 코덱(codec), 예를 들어, “photojpeg, h264, prores422HQ, proresLT, prores4444, proresProxy, dnx36”의 값 중 적어도 일부 인수가 옵션으로 함께 입력될 수 있다.
- [0122] 일 실시예에 따르면, 렌더링된 콘텐츠의 주소를 변경하려는 경우, “\$ dils | nkread | nkmov ../test/test.mov”와 같이 mov 확장자를 함께 입력한다.
- [0123] 이미지 처리 장치(101)는 기 생성된 파이프라인에서 표준 입력 데이터에 기록된 정보를 확인하는 경우 해당 정보를 포함할 수 있다. 일 실시예에 따르면, 명령어 “nkmov(805)”는 OSX 시스템을 위한 문자열을 생성할 수 있다.
- [0124] 더하여, 이미지 처리 장치(101)는 생성된 파이프라인을 저장함으로써, 렌더링 데이터를 생성할 수 있다. 일 실시예에 따르면, 이미지 처리 장치(101)는 명령어 “nksave”의 입력을 확인하는 경우, 생성된 파이프라인을 저장할 수 있다. 이 때, 생성되는 렌더링 데이터는 호환성 파일로 저장될 수 있다.
- [0125] 일 실시예에 따르면, 명령어 “nksave(807)”는 “\$ dils | nkread | nkmov | nksave”와 같이 콘텐츠가 포함된 디렉터리에 렌더링 데이터를 생성하도록 입력될 수 있고 또는, “\$ dils | nkread | nkmov | nksave /project/paths/save.nk”와 같이 렌더링 데이터를 생성하기 위한 경로(예: /project/paths/save.nk)와 함께 입력될 수 있다.
- [0126] 이 때, 이미지 처리 장치(101)는, 렌더링 데이터를 저장함에 있어서, 해당 디렉터리가 존재하지 않는 경우 디렉터리를 생성할 수 있다.
- [0127] 이미지 처리 장치(101)는, 명령어 “nkthmb(809)”의 입력을 확인하는 경우, 디렉터리의 시퀀스 데이터에 포함되는 적어도 하나의 이미지를 썸네일로 설정하는 문자열을 생성할 수 있다.
- [0128] 일 실시예에 따르면, 명령어 “nkthmb(809)”는, “\$ dils | nkread | nkreformat 410x264 | nkwrite | nksave | nkthmb”과 같이 입력될 수 있다.
- [0129] 이 때, 이미지 처리 장치(101)는 시퀀스 데이터에 포함되는 연속된 이미지들 중 지정된 한 장의 이미지(예: 가운데 위치한 이미지)를 썸네일로 생성하도록 처리하는 문자열을 포함할 수 있다.
- [0130] 또한, 이미지 처리 장치(101)는 명령어 “nkthmb(809)”의 입력을 확인하는 경우, 썸네일 설정에 대한 문자열을 생성하는 것에 한정하지 않고, 확인된 명령어에 기반하여 썸네일을 생성하고, 렌더링 데이터에 포함하도록 처리할 수 있다.
- [0131] 상술한 바와 같이, 이미지 처리 장치(101)는 입력되는 명령어에 따라서, 콘텐츠(예: 시퀀스 데이터)를 렌더링하도록 처리하는 문자열 구조를 포함하는 렌더링 데이터를 생성할 수 있다.
- [0132] 다양한 실시 예에 따르면, 파이프라인에 포함되는 문자열 구조를 생성하기 위한 다양한 명령어 중 적어도 일부는 선택적으로 입력될 수 있다. 예를 들어, 영상 처리를 위한 명령어셋(700)에 포함되는 명령어 “nkreformat(701)”, “nk2aces(703)” 및 “nk1lut(705)” 각각은 해당 명령어가 입력되는지 여부에 따라서 선택적으로 사용될 수 있다. 마찬가지로, 렌더링을 위한 명령어 “nkffmpeg(801)”, “nkwrite(803)” 및 “nkmov(805)” 또한 해당 명령어가 입력되는지 여부에 따라서 선택적으로 사용될 수 있다.
- [0133] 도 4를 참조하면, 이미지 처리 장치(101)에서 렌더링을 수행하기 위한 동작을 설명할 수 있다. 도 4를 통하여 설명하는 다양한 실시예는, 도 2의 단계(S207)을 통해서 추가로 수행되는 동작일 수 있다.
- [0134] 일 실시예에 따르면, 도 4에 도시된 바와 같이, 이미지 처리 장치(101)는 렌더링을 수행하는 명령어 또는 렌더링 데이터를 렌더팜으로 전송하는 명령어의 입력을 확인한다(S401).

- [0135] 여기서, 이미지 처리 장치(101)는 “\$ dils | nkread | nkwrite | nksave | nkrender” 와 같이 입력되는 명령어 “nkrender(811)” 의 입력을 확인하는 경우, 생성된 렌더링 데이터에 포함된 파이프라인에 기반하여 콘텐츠의 렌더링을 수행한다(S407).
- [0136] 이미지 처리 장치(101)는 콘텐츠의 렌더링을 수행한 결과물(예: 렌더링된 콘텐츠)를 표준 출력 데이터(예: stdout)으로 출력할 수 있다.
- [0137] 다양한 실시예에 따르면 이미지 처리 장치(101)입력되는 명령어에 따라서, 적어도 하나의 외부 장치(렌더팜)와 호환되는 렌더링 데이터를 생성할 수 있고, 및/또는 생성된 렌더링 데이터를 해당 외부 장치(예: 렌더팜)을 통해서 렌더링하도록 전송할 수 있다.
- [0138] 예를 들어, 이미지 처리 장치(101)는 명령어 “makealf(813)” 를 확인하는 경우, 시퀀스 데이터에 포함되는 각각의 이미지에 대하여 alf 파일을 생성하도록 처리한다(S403). 이 때, 이미지 처리 장치(101)는 생성된 alf 파일을 표준 출력 데이터(예: stdout)으로 출력하고, alf 파일을 포함하는 렌더링 데이터를 생성할 수 있다.
- [0139] 일 실시예에 따르면, 명령어 “makealf(813)” 는 “\$ dils | nkread | nkwrite | nksave | makealf” 와 같이 입력될 수 있고, 또는 “\$ echo “/bin/touch /tmp.test.txt” | makealf” 와 같이 외부 명령어(echo)를 포함하여 입력될 수 있다.
- [0140] 그리고, 이미지 처리 장치(101)는 명령어 “tractor(815)” 를 확인하는 경우, 생성된 alf 파일을 렌더팜으로 전송한다(S405). 렌더팜 데이터를 수신하는 외부 장치(예: 렌더팜)은 수신한 렌더팜 데이터의 파이프라인에 기반하여 해당 콘텐츠(예: 시퀀스 데이터)를 렌더링할 수 있다.
- [0141] 일 실시예에 따르면, 명령어 “tractor(815)” 는 \$ dils | nkread | nkwrite | nksave | makealf | tractor rfarm(렌더팜 예약어)” 와 같이 입력될 수 있다.
- [0142] 이 때, 이미지 처리 장치(101)는 외부 명령어(예: echo)가 포함되었는지 여부를 확인할 수 있고, 외부 명령어가 포함된 경우, 해당 정보를 함께 전송할 수 있다. 또한, 명령어 “tractor(815)” 는 “| tractor rfarm” 과 같이 렌더링 데이터가 전송될 렌더팜을 지정하는 렌더팜 예약어(rfarm)를 포함할 수 있다. 렌더팜을 통해서 렌더링된 콘텐츠는 표준 출력 데이터(예: stdout)을 통해서 출력될 수 있고, log 서버 등에 저장될 수 있다.
- [0143] 다양한 실시예에 따르면, 명령어 “nksave(807)” 의 입력 없이, 명령어 “nkrender(811)” 를 확인할 수 도 있다. 이 때, 이미지 처리 장치(101)는 기 생성된 파이프라인에 기반하여 해당 콘텐츠(예: 시퀀스 데이터)의 렌더링을 수행할 수 있다.
- [0144] 도 9에 도시된 바와 같이, 이미지 처리 장치(101)는 명령어 “dils” 를 이용하여 렌더링에 사용되는 다양한 명령어들을 호출하고, 입력되는 명령어에 따라서 콘텐츠(예: 시퀀스 데이터)에 처리되는 효과를 선택적으로 결정하며, 명령어 “nkrender” 또는 명령어 “makealf” 및 “tractor” 를 통하여 렌더링 결과를 획득함으로써 연산 과정을 단순화할 수 있고, 특정 규칙들로부터 렌더링 호환성에 대한 자유도를 확보할 수 있다.
- [0145] 본 발명의 다양한 실시예에 따르면, 지정된 명령어를 이용하여 렌더링을 위한 파이프라인을 생성함으로써, 렌더링을 수행하는 프로젝트 종류에 관계없이 동일한 파이프 라인을 형성할 수 있다.
- [0146] 본 발명의 다양한 실시예에 따르면, 다양한 장치와의 호환성을 가지는 명령어 및 문자열을 이용하여 콘텐츠를 렌더링하기 위한 렌더링 데이터를 생성함으로써, 다양하게 요구되는 이미지 처리에 대하여 효과적으로 대처할 수 있다.
- [0147] 상술한 바와 같이, 높은 라이선스 비용을 지출하는 프로그램 모듈을 사용하지 않고 렌더링 데이터를 생성하는 방법 및 장치를 제공함으로써, 프로그래밍에 있어 자유도를 확보하고, 또한, 과도하게 지출되는 비용을 절약할 수 있다.
- [0148] 더하여, 도시하고 있지는, 않지만, 이미지 처리 장치(101)에서 수행되는 외부 장치와의 연결 및/또는 데이터 송수신은 이미지 처리 장치(101)에 포함되는 적어도 하나의 통신부를 통하여 수행될 수 있다.
- [0149] 여기서, 통신부는 이미지 처리 장치(101)를 구성하는 적어도 하나의 구성 요소들, 적어도 하나의 외부 장치 및 외부 프로그램 모듈 간의 통신을 연결할 수 있다. 예를 들면, 통신부는 무선 통신 또는 유선 통신을 통해서 네트워크에 연결되어 장치들과 통신할 수 있다.
- [0150] 통신부는, 와이파이(wireless fidelity, Wifi) 통신, 와이맥스(worldwide interoperability formicrowave

access, WMax) 통신, 블루투스(Bluetooth, BT) 통신, 지그비(Zigbee) 통신 중 적어도 하나의 무선 통신을 수행하기 위한 통신 모듈을 포함할 수 있다.

- [0151] 또한, 통신부는, USB(universal serial bus), HDMI(high definition multimedia interface), RS-232(recommended standard 232) 또는 POTS(plain old telephone service) 중 적어도 하나의 유선 통신을 수행하기 위한 통신 모듈을 포함할 수도 있다. 이 때, 통신부(807)는 하나의 integrated chip(IC) 또는 IC 패키지 내에 포함될 수 있다.
- [0152] 상술한 본 발명의 다양한 실시예들은 소프트웨어로 구현될 수 있다. 소프트웨어로 구현되는 경우, 저장부는 하나 이상의 프로그램(또는 프로그래밍 모듈, 어플리케이션)을 저장하는 컴퓨터 판독 가능 저장 매체(또는 컴퓨터로 읽을 수 있는 저장 매체)로 제공될 수 있다. 예를 들어, 저장부는 프로그래밍 모듈의 형태로 컴퓨터로 읽을 수 있는 저장매체(computer-readable storage media)이며, 소프트웨어는, 저장부에 저장된 명령어로 구현될 수 있다.
- [0153] 하나 이상의 프로그램은, 이미지 처리 장치(101)로 하여금, 본 발명의 청구항 및/또는 명세서에 기재된 실시예에 따른 방법들을 실행하게 하는 명령어(instructions)를 포함할 수 있다. 상기 명령어는, 하나 이상의 제어부에 의해 실행될 경우, 상기 하나 이상의 제어부가 상기 명령어에 해당하는 기능을 수행할 수 있다. 컴퓨터로 읽을 수 있는 저장매체는, 예를 들면, 메모리가 될 수 있다. 이 때, 데이터베이스 및 로거서버 중 적어도 일부는 메모리에 포함되어 구성될 수 있다.
- [0154] 상기 프로그래밍 모듈의 적어도 일부는, 예를 들면, 상기 제어부에 의해 구현(implement)(예: 실행)될 수 있다. 상기 프로그래밍 모듈의 적어도 일부는 하나 이상의 기능을 수행하기 위한, 예를 들면, 모듈, 프로그램, 루틴, 명령어 세트 (sets of instructions) 또는 프로세스 등을 포함할 수 있다.
- [0155] 예를 들어, 본 발명의 일 실시예에 따르면, 이미지 처리 장치(101)은 데이터 오브젝트를 전달하기 위해 개방형 표준 포맷으로 사용되는 javascript object notation(json) 형태의 포맷, 그리고, json 형태로 저장되는 NoSQL 중 하나인 몽고DB(mongodb)를 이용하여 구현될 수 있다.
- [0156] 또한, 본 발명의 일 실시예에 따르면, 이미지 처리 장치(101)은 연결되는 다양한 사용자 장치를 관리하기 위하여 lightweight directory access protocol(LDAP) 프로토콜을 이용하여 구현될 수 있다.
- [0157] 상기 컴퓨터로 판독 가능한 기록 매체에는 하드디스크, 플로피디스크 및 자기 테이프와 같은 마그네틱 매체(Magnetic Media)와, CD-ROM(Compact Disc Read Only Memory), DVD(Digital Versatile Disc)와 같은 광기록 매체(Optical Media)와, 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-Optical Media)와, 그리고 ROM(Read Only Memory), RAM(Random Access Memory), 플래시(flash) 메모리 등과 같은 프로그램 명령(예: 프로그래밍 모듈, 또는 프로그램)을 저장하고 수행하도록 특별히 구성된 하드웨어 장치, 삭제가능 프로그램가능(EEPROM, Electrically Erasable Programmable Read Only Memory), 자기 디스크 저장 장치(magnetic disc storage device) 또는 다른 형태의 광학 저장 장치, 마그네틱 카세트(magnetic cassette)가 포함될 수 있다. 또는, 이들의 일부 또는 전부의 조합으로 구성된 메모리에 저장될 수 있다. 또한, 각각의 구성 메모리는 다수 개 포함될 수도 있다.
- [0158] 더하여, 상기 소프트웨어는, 입력 장치에 인터넷(Internet), 인트라넷(Intranet), LAN(Local Area Network), WLAN(Wide LAN), 또는 SAN(Storage Area Network)과 같은 통신 네트워크, 또는 이들의 조합으로 구성된 통신 네트워크를 통하여 접근(access)할 수 있는 부착 가능한(attachable) 저장 장치(storage device)에 저장될 수 있다. 이러한 저장 장치는 외부 포트를 통하여 이미지 처리 장치(101)에 접속할 수 있다. 또한, 통신 네트워크 상의 별도의 저장장치가 이미지 처리 장치(101)에 접속할 수도 있다.
- [0159] 상술한 하드웨어 장치는 본 발명의 다양한 실시예에 대한 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지다.
- [0160] 본 발명의 다양한 실시예에 따른 모듈 또는 프로그래밍 모듈은 전술한 구성요소들 중 적어도 하나 이상을 포함하거나, 일부가 생략되거나, 또는 추가적인 다른 구성요소를 더 포함할 수 있다. 본 발명의 다양한 실시예에 따른 모듈, 프로그래밍 모듈 또는 다른 구성요소에 의해 수행되는 동작들은 순차적, 병렬적, 반복적 또는 휴리스틱(heuristic)한 방법으로 실행될 수 있다. 또한, 일부 동작은 다른 순서로 실행되거나, 생략되거나, 또는 다른 동작이 추가될 수 있다.
- [0161] 본 명세서와 도면에 개시된 본 발명의 실시예들은 본 발명의 기술 내용을 쉽게 설명하고 본 발명의 이해를 돕기

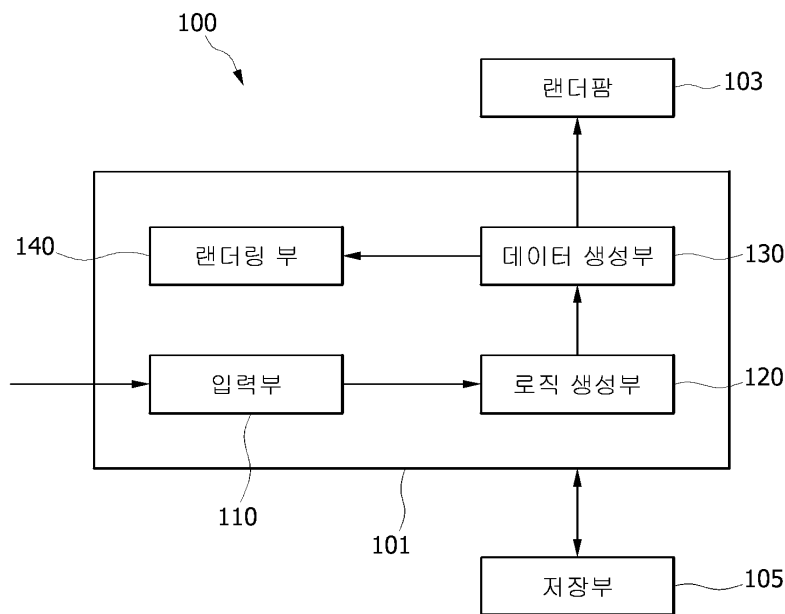
위해 특정 예를 제시한 것일 뿐이며, 본 발명의 범위를 한정하고자 하는 것은 아니다. 따라서 본 발명의 범위는 여기에 개시된 실시예들 이외에도 본 발명의 기술적 사상을 바탕으로 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

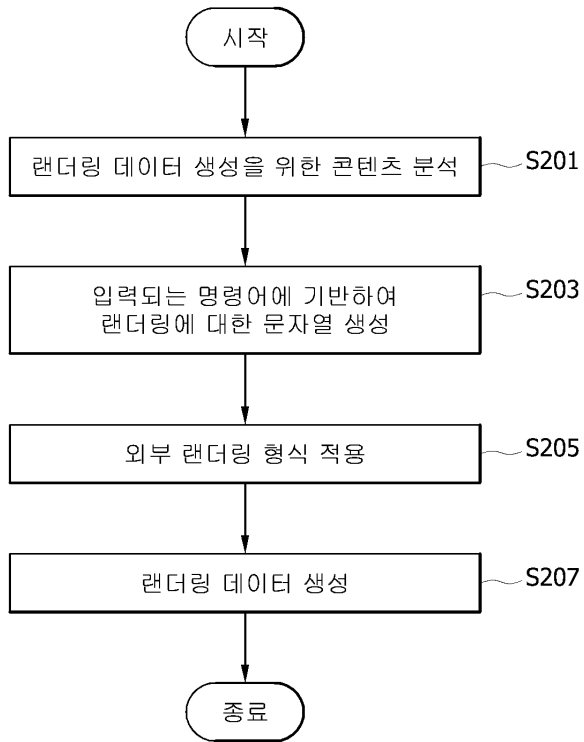
- [0163]
- | | |
|-----------------|----------------|
| 100: 이미지 처리 시스템 | 101: 이미지 처리 장치 |
| 110: 입력부 | 120: 로직 생성부 |
| 130: 데이터 생성부 | 140: 렌더링부 |
| 103: 렌더팜 | 105: 저장부 |

도면

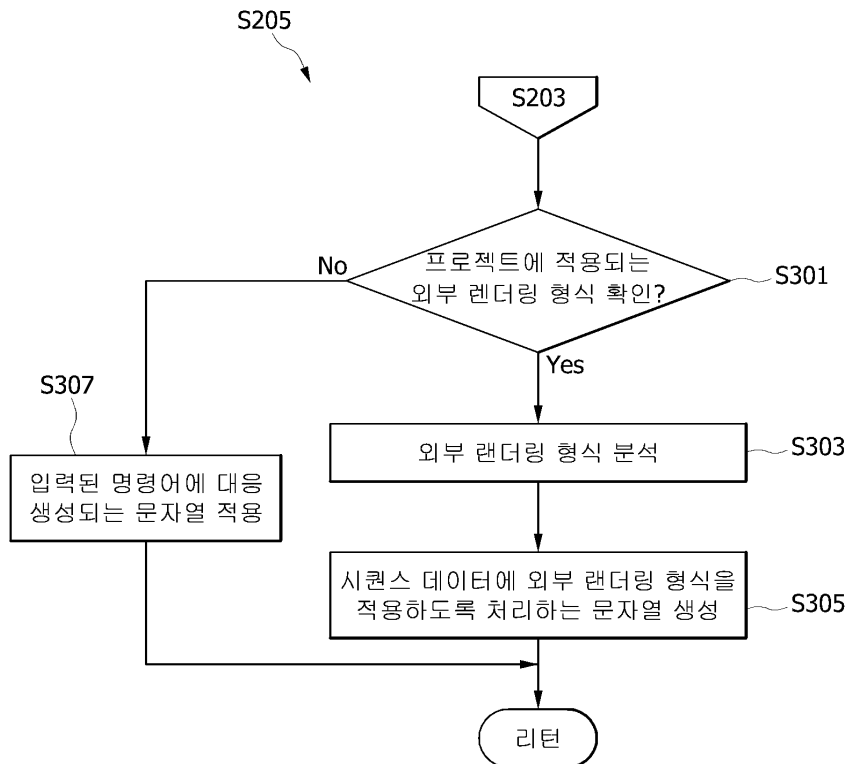
도면1



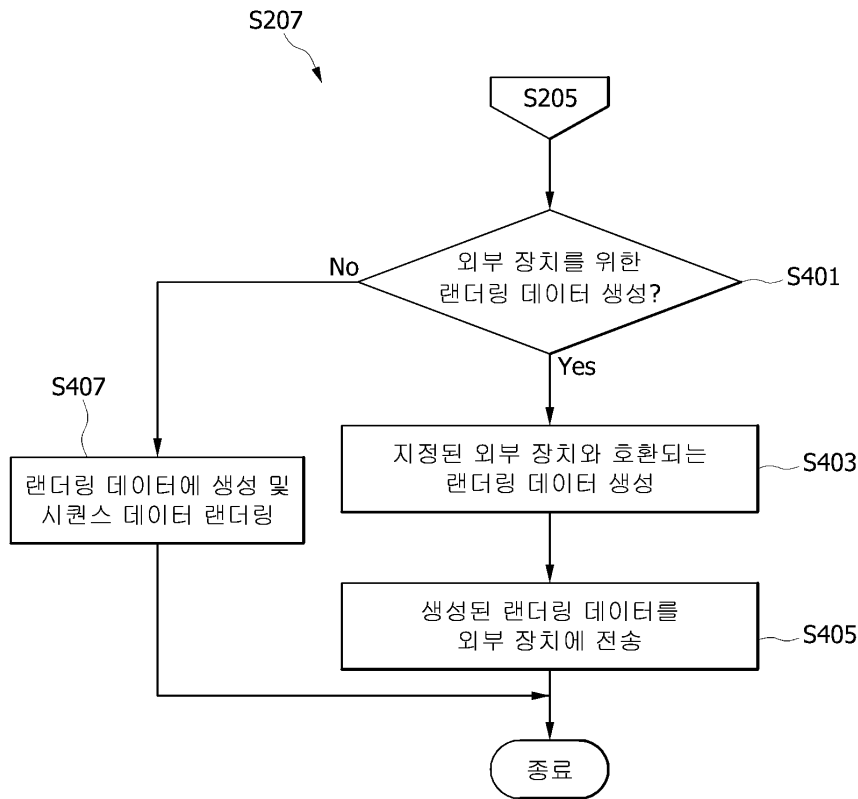
도면2



도면3



도면4

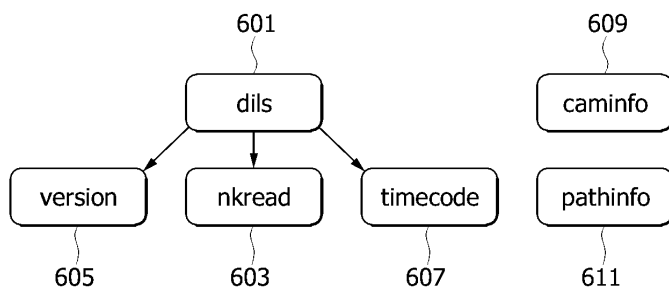


도면5

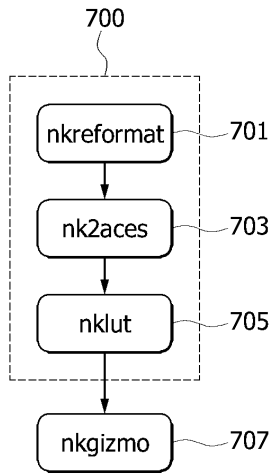
```

[ DIGITALIDEA\dl0191@di90193 dicmd]$ nkgizmo idea_slate
== Check Source ==
/lustre/Digitalidea_source/2d_team_source/font/CI_whiteText.png
/lustre/Digitalidea_source/2d_team_source/font/mono.ttf
/lustre/Digitalidea_source/2d_team_source/font/han01.ttf
/lustre/Digitalidea_source/2d_team_source/font/FreeSans.ttf
== Gizmo Option : idea_slate ==
* plate_artist : .
* plate_shotname : .
* plate_version : .
* plate_jobname : .
* memo : .
* scale : .
    
```

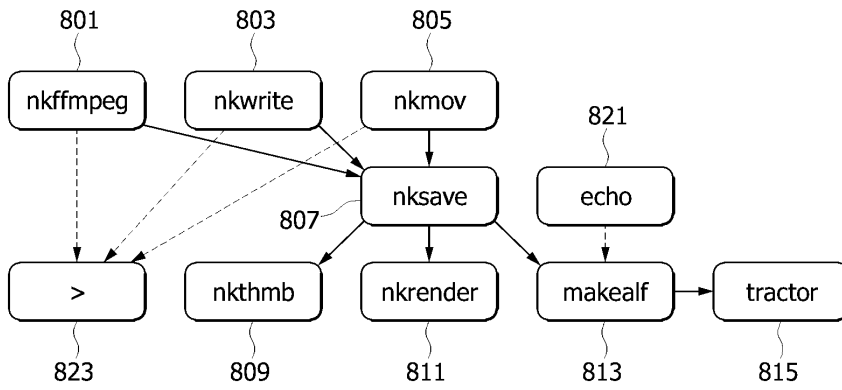
도면6



도면7



도면8



도면9

