



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2009-0031408  
(43) 공개일자 2009년03월25일

(51) Int. Cl.  
*G06T 3/00* (2006.01) *G06T 3/40* (2006.01)  
(21) 출원번호 10-2009-7000316  
(22) 출원일자 2009년01월07일  
심사청구일자 없음  
번역문제출일자 2009년01월07일  
(86) 국제출원번호 PCT/US2007/012641  
국제출원일자 2007년05월29일  
(87) 국제공개번호 WO 2008/008131  
국제공개일자 2008년01월17일  
(30) 우선권주장  
11/482,558 2006년07월07일 미국(US)

(71) 출원인  
**마이크로소프트 코포레이션**  
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
마이크로소프트 웨이  
(72) 발명자  
**고엘, 라자**  
미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
소프트 웨이 마이크로소프트 코포레이션 내  
**굿윈, 마가레트 엘.**  
미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
소프트 웨이 마이크로소프트 코포레이션 내  
(뒷면에 계속)  
(74) 대리인  
**양영준, 백만기**

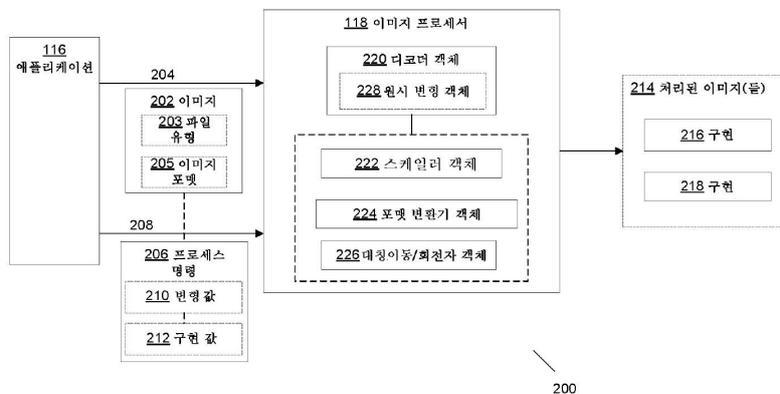
전체 청구항 수 : 총 20 항

**(54) 이미지의 다중 및 원시 표현을 제공하기 위한 방법**

**(57) 요약**

이미지의 다중 및 원시 표현을 제공하는 것에 관하여 각종 실시예들이 개시된다. 예시적인 실시예에 따르면, 예를 들면, 이미지의 단일한 구현만 제공하는 것 대신에, 이미지의 다중 구현이 생성되어 제공될 수 있다. 또한, 다른 실시예에서는, 다중 구현의 생성 및 출력은 하나 이상의 원시 객체를 사용하여 이미지 또는 구현을 제공하기 위한 변형 또는 이미지 처리를 원시적으로 수행할 수 있다.

**대표도**



(72) 발명자

**마가린트, 라두 씨.**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

**올로다르크지크, 로버트 에이.**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

**올슨, 토마스 더블유.**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

**왕, 웨이칭 존스**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

**특허청구의 범위**

**청구항 1**

이미지를 수신하는 단계;

수신된 상기 이미지에 대한 이미지 처리 명령을 수신하는 단계; 및

상기 이미지의 하나 이상의 구현을 제공하기 위해 상기 이미지 처리 명령에 기초하여 상기 이미지에 대하여 복수의 변형(transform)을 원시적으로(natively) 수행하는 단계

를 포함하는 방법.

**청구항 2**

제1항에 있어서, 상기 이미지를 수신하는 단계는,

래스터 그래픽 이미지, 디지털 이미지, 및/또는 비트맵 이미지를 포함하는 파일을 수신하는 단계를 포함하는 방법.

**청구항 3**

제1항에 있어서, 상기 이미지 처리 명령을 수신하는 단계는,

상기 이미지를 디코딩 또는 압축풀기, 상기 이미지를 크기조정하기, 상기 이미지의 포맷을 상이한 포맷으로 변환하기, 및/또는 상기 이미지를 대칭이동(flipping) 또는 회전시키기 중 하나 이상의 변형을 상기 이미지에 수행하라는 이미지 처리 명령을 수신하는 단계를 포함하는 방법.

**청구항 4**

제1항에 있어서, 상기 이미지에 대하여 복수의 변형을 원시적으로 수행하는 단계는,

상기 이미지 처리 명령에 기초하여 상기 이미지에 대하여 디코더에 의해 제1 변형을 원시적으로 수행하는 단계; 및

상기 이미지 처리 명령에 기초하여 상기 이미지에 대하여 제2 변형을 디코더에 의해 원시적으로 수행하는 단계를 포함하는 방법.

**청구항 5**

제1항에 있어서, 상기 이미지에 대하여 복수의 변형을 원시적으로 수행하는 단계는,

상기 이미지를 재-샘플링하지 않고, 상기 이미지의 하나 이상의 구현을 제공하기 위해 상기 이미지 처리 명령에 기초하여 상기 이미지에 대하여 복수의 변형을 수행하는 단계를 포함하는 방법.

**청구항 6**

제1항에 있어서, 상기 이미지에 대하여 복수의 변형을 원시적으로 수행하는 단계는,

디코더 객체를 인스턴스화하는 단계;

하나 이상의 변형을 위한 객체를 인스턴스화하는 단계; 및

상기 하나 이상의 변형을 위한 객체를 상기 디코더 객체와 연관시키는 단계

를 포함하는 방법.

**청구항 7**

제6항에 있어서, 상기 하나 이상의 변형을 위한 객체를 인스턴스화하는 단계는,

스케일러 객체를 인스턴스화하는 단계;

포맷 변환기 객체를 인스턴스화하는 단계; 또는  
대칭이동/회전자 객체를 인스턴스화하는 단계  
를 포함하는 방법.

**청구항 8**

제1항에 있어서, 상기 이미지에 대하여 복수의 변형을 원시적으로 수행하는 단계는,  
처리중인 상기 이미지의 상태를 식별하는 상태 정보를 저장하는 단계를 더 포함하는 방법.

**청구항 9**

제1항에 있어서, 상기 이미지에 대하여 복수의 변형을 원시적으로 수행하는 단계는,  
상기 이미지에 대하여 상기 하나 이상의 변형이 수행된 후의 하나 이상의 중간 상태 이미지 구현을 저장하는 단계를 더 포함하는 방법.

**청구항 10**

제1항에 있어서, 상기 이미지에 대하여 복수의 변형을 원시적으로 수행하는 단계는,  
상기 이미지의 복수의 구현을 제공하기 위해 상기 이미지 처리 명령에 기초하여 상기 이미지에 대해 복수의 변형을 원시적으로 수행하는 단계를 더 포함하는 방법.

**청구항 11**

이미지를 수신하는 단계;  
상기 이미지에 대한 이미지 처리 명령을 수신하는 단계; 및  
상기 이미지의 복수의 구현을 제공하기 위해 상기 이미지 처리 명령에 기초하여 상기 이미지에 대하여 복수의 변형을 수행하는 단계- 상기 복수의 구현 각각은 상기 이미지에 대하여 원시적으로 수행되는 적어도 하나의 변형을 포함함 -  
를 포함하는 방법.

**청구항 12**

제11항에 있어서, 상기 이미지 처리 명령을 수신하는 단계는,  
상기 이미지를 디코딩 또는 압축풀기, 상기 이미지를 크기조정하기, 상기 이미지의 포맷을 상이한 포맷으로 변환하기, 및/또는 상기 이미지를 대칭이동 또는 회전시키기 중 하나 이상의 변형을 상기 이미지에 수행하라는 이미지 처리 명령을 수신하는 단계를 포함하는 방법.

**청구항 13**

제11항에 있어서, 상기 이미지에 대하여 복수의 변형을 수행하는 단계는,  
디코더 객체를 인스턴스화하는 단계;  
하나 이상의 변형을 위한 객체를 인스턴스화하는 단계; 및  
상기 하나 이상의 변형을 위한 객체를 상기 디코더 객체와 연관시키는 단계를 포함하는 방법.

**청구항 14**

제11항에 있어서, 상기 이미지에 대하여 복수의 변형을 수행하는 단계는,  
상기 이미지의 복수의 구현을 제공하기 위해 상기 이미지에 대하여 상기 하나 이상의 변형이 수행된 후의 하나 이상의 중간 상태 이미지 구현을 저장하는 단계를 더 포함하는 방법.

**청구항 15**

제11항에 있어서, 상기 이미지에 대하여 복수의 변형을 수행하는 단계는,  
 복수의 변형 각각을 수행하기 위한 객체를 인스턴스화하는 단계를 포함하고,  
 상기 인스턴스화하는 단계는,  
 디코더 객체를 인스턴스화하는 단계;  
 스케일러 객체를 인스턴스화하는 단계;  
 포맷 변환기 객체를 인스턴스화하는 단계; 또는  
 대칭이동/회전자 객체를 인스턴스화하는 단계  
 중 하나 이상을 포함하는 방법.

**청구항 16**

제11항에 있어서, 상기 이미지에 대하여 복수의 변형을 수행하는 단계는,  
 상기 이미지의 복수의 구현을 제공하기 위해 상기 이미지 처리 명령어에 기초하여 상기 이미지에 대하여 복수의 변형을 수행하는 단계를 포함하고,  
 상기 복수의 구현은 원본(original) 또는 미처리된 이미지 및 하나 이상의 처리되거나 변형된 이미지 구현을 포함하는 방법.

**청구항 17**

제11항에 있어서, 상기 이미지에 대하여 복수의 변형을 수행하는 단계는,  
 상기 이미지의 복수의 구현을 출력하는 단계를 포함하는 방법.

**청구항 18**

이미지를 수신하는 단계;  
 이미지 처리 명령을 수신하는 단계; 및  
 상기 이미지 처리 명령에 기초하여, 상기 이미지 처리 명령에 따라 상기 이미지를 처리하여 상기 이미지의 다중 구현을 제공하기 위해 하나 이상의 변형 및 상기 변형을 위한 배열(arrangement)을 결정하는 단계를 포함하는 방법.

**청구항 19**

제18항에 있어서, 상기 결정하는 단계는,  
 상기 이미지 처리 명령에 기초하여, 상기 이미지 처리 명령에 따라 상기 이미지를 처리하여 상기 이미지의 다중 구현을 제공하기 위해 하나 이상의 변형 및 상기 변형의 직렬 및/또는 병렬 배열 또는 순서를 결정하는 단계를 포함하는 방법.

**청구항 20**

제18항에 있어서,  
 상기 이미지 처리 명령에 따라 상기 이미지의 상기 다중 구현을 제공하기 위한 결정에 기초한 순서 및/또는 배열로 상기 이미지에 대하여 상기 하나 이상의 변형을 수행하는 단계를 더 포함하는 방법.

**명세서**

**배경 기술**

<1> 이미지들은 다양한 포맷 및 해상도로 제공될 수 있다. 몇몇 이미지 처리 애플리케이션은 한 이미지가 처리되거나

나 다른 포맷으로 변환되는 등을 행하게끔 할 수 있다. 이 처리는 통상적으로 이미지에 대하여 수행되는 하나 이상의 변형(transform)을 통해 수행될 수 있다. 몇몇 경우에, 이러한 이미지 처리는 원하는 이미지를 생성할 수는 있지만, 원본 이미지가 처리 중에 통상적으로 폐기될 수 있기 때문에 해상도의 저감 없이 또 다른 처리를 하는 것은 불가능하게 될 수 있다. 또한, 이러한 이미지 처리 기술들은 통상적으로 상당히 제한적이다.

**발명의 상세한 설명**

- <2> 이미지의 다중 및 원시 표현(multiple and native representations)을 제공하는 것에 관련하여 각종 실시예들이 개시된다. 예시적인 실시예에 따르면, 예를 들어, 이미지의 단일한 구현 보다는, 이미지의 다중 구현이 생성되어 제공될 수 있다. 또한, 다중 구현의 생성 및 출력은 변환 또는 이미지 처리를 원시적으로(natively) 수행하는데 원시 객체(native objects)를 사용하여 이미지 또는 구현을 제공할 수 있다.
- <3> 예시적인 실시예에 따라, 한 방법이 제공된다. 이 방법은 이미지를 수신하는 단계, 수신된 이미지에 대한 이미지 처리 명령을 수신하는 단계, 및 이미지의 하나 이상의 구현을 제공하기 위해 이미지 처리 명령에 기초하여 이미지에 대해 복수의 변형을 원시적으로 수행하는 단계를 포함할 수 있다.
- <4> 다른 예시적인 실시예에 따라, 한 방법이 제공된다. 그 방법은 이미지를 수신하는 단계, 그 이미지에 대한 이미지 처리 명령을 수신하는 단계, 및 이미지의 복수의 구현을 제공하기 위해 이미지 처리 명령에 기초하여 이미지에 대하여 복수의 변형을 수행하는 단계를 포함할 수 있다. 복수의 구현 각각은 이미지에 대하여 원시적으로 수행되는 적어도 하나의 변형을 포함할 수 있다.
- <5> 또 다른 예시적인 실시예에 따라, 한 방법이 제공된다. 그 방법은 이미지를 수신하는 단계, 이미지 처리 명령을 수신하는 단계, 및 이미지 처리 명령에 기초하여, 그 이미지 처리 명령에 따라 이미지를 처리하여 이미지의 다중 구현을 제공하기 위해 하나 이상의 변형 및 변형을 위한 배열을 결정하는 단계를 포함할 수 있다.
- <6> 본 설명은 이하의 실시예에서 더 설명되는 개념의 선택을 단순화된 형태로 소개하기 위해 제공된다. 본 설명은 청구되는 요지의 핵심 특징 또는 주요 특징들을 규정하기 위하거나, 청구되는 요지의 범위를 결정하기 위한 목적으로 사용되기 위한 것은 아니다.

**실시예**

- <14> 도면을 참조하면, 동일한 참조부호는 동일한 요소를 지칭하고 있으며, 도 1은 예시적인 실시예에 따른 컴퓨팅 시스템(100)의 블록도이다. 각종 실시예들은 예시적인 실시예에 따라, 컴퓨팅 시스템(100) 또는 기타 컴퓨팅 시스템 내에서 실행되거나 구현될 수 있다. 컴퓨팅 시스템(100)은 예를 들면, 랩탑 컴퓨터, 데스크탑 컴퓨터, 퍼스널 컴퓨터(PC), 서버, 네트워크 디바이스(예를 들면, 스위치, 라우터), 이동 전화기, PDA(personal digital assistant), 게임 디바이스, 핸드헬드 통신 디바이스, 또는 기타 컴퓨팅 디바이스를 포함할 수 있다. 이것들은 단지 일례에 불과하며, 각종 실시예들은 그것에 제한되지 않는다.
- <15> 시스템(100)은 CPU(central processing unit; 102), 및 RAM(random access memory; 106)과 ROM(read-only memory; 108)을 비롯한 시스템 메모리(104)를 포함할 수 있다. CPU(102)는 컴퓨터 애플리케이션 내의 명령어를 해석하고 데이터를 처리할 수 있다. 예를 들면, CPU(102)는 마이크로프로세서일 수 있고 다수의 집적 회로를 포함할 수 있으며, 소프트웨어 또는 명령어를 실행하고 컴퓨터 시스템(100)에 전체적인 제어를 제공할 수 있다. RAM(106)은 RAM에 저장된 데이터가 임의의 순서로 액세스 되게 할 수 있고, 시스템(100)은 또한 예를 들면 RAM(106)에 기입되거나 그로부터 관독될 수 있다. 예를 들면, 데이터 파일은 시각 X에 RAM(106)에 저장될 수 있고, 시각 Y에 RAM(106)으로부터 관독될 수 있다. ROM(108)은 시스템(100)에 의해 ROM(108)으로부터 관독될 수 있는 데이터 또는 시스템을 포함할 수 있다(일반적으로 시스템(100)에 의해 기입 또는 관독될 수 있는 RAM(106)과는 대조적임). 예를 들면, ROM(108)은 시스템(100)이 처음 켜졌을 때 시스템(100)에 의해 실행될 기본 입/출력 시스템(BIOS) 소프트웨어 코드를 포함할 수 있다. ROM(108)은 또한 예를 들면 시계(110)에 결합될 수 있다. 예를 들면, 시계(110)는 ROM(108)을 통해 시스템(100)에 시간을 제공할 수 있다.
- <16> 시스템(100)은 또한 입/출력 제어기(128) 및 디스플레이 디바이스(130)를 포함할 수 있다. 입/출력(I/O) 제어기(128)는 디스플레이 디바이스(130)와 같은, 하나 이상의 I/O 디바이스와 인터페이스할 수 있다. 예를 들면, 입/출력 제어기(128)는 디스플레이 디바이스(130)에 출력을 제공할 수 있다. 디스플레이 디바이스(130)는 입/출력 제어기(128)로부터 데이터를 수신하여 그 데이터를 사용자에게 제공하거나 디스플레이하도록 구성된 디바이스일 수 있다. 도시되지는 않았지만, 입/출력 제어기(128)는 키보드, 마우스, 조이스틱, 또는 시스템(100)에 입력을 제공하도록 구성된 임의의 기타 입력 디바이스와 같은 각종 입력 디바이스들에 또한 결합될 수 있다.

- <17> 시스템(100)은 또한 데이터, 명령어 또는 기타 정보를 저장하기 위한 데이터 저장 디바이스(114)를 포함할 수 있다. 데이터 저장 디바이스(114)는, 예를 들면, 자기 표면에 데이터를 저장하는 하드 디스크 드라이브, CD-ROM, 또는 시스템(100)에 의해 액세스가능한 임의의 기타 이용가능한 미디어 저장 디바이스일 수 있다.
- <18> 제한이 아닌 예로서, 데이터 저장 디바이스(114)는 임의의 컴퓨터-관독가능 매체를 포함할 수 있다. 컴퓨터 관독가능 매체는, 예를 들면, 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터-관독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 매체를 포함할 수 있다. 컴퓨터 저장 매체는 RAM, ROM, EPROM, EEPROM, 플래시 메모리 또는 기타 고상 메모리 기술, CD-ROM, DVD, 또는 기타 광학 저장소, 자기 카세트, 자기 테이프, 자기 디스크 저장소 또는 기타 자기 저장 디바이스, 또는 원하는 정보를 저장하는 데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 기타 매체를 포함할 수 있지만 이에 제한되는 것은 아니다.
- <19> 애플리케이션(116), 이미지 프로세서(118), 운영 체제(120), 및 스택(122)과 같은 하나 이상의 소프트웨어 프로그램 또는 블록이 데이터 저장 디바이스(114)에 저장될 수 있다. 애플리케이션(116)은 웹 브라우저, 워드 프로세서, 멀티미디어 애플리케이션, 또는 기타 애플리케이션일 수 있다. 예를 들어, 애플리케이션(116)은 예를 들면, 이미지 및 이미지 처리(또는 프로세스) 명령을 송신하고 하나 이상의 구현을 비롯한 처리된 이미지들을 수신하도록 구성된 이미징 애플리케이션일 수 있다. 애플리케이션(116)은 예를 들면, 이미지, 프로세스 명령, 처리된 이미지, 및/또는 사용자의 입력에 기초한 구현을 송수신할 수 있다. 예를 들어, 애플리케이션(116)은 디지털 카메라로부터 이미지를 수신할 수 있다. 그런 후에 예를 들면, 애플리케이션(116)은 이미지 프로세서(118)에 이미지를 송신하여 대칭이동/회전 변형(flip/rotate transform)과 같은 변형을 수행하고, 이 변형 후에 이미지 프로세서(118)가 구현(216)을 포함하는 처리된 이미지(214)를 애플리케이션(116)에 반환할 수 있다.
- <20> 이미지 프로세서(118)는 이미지 처리를 수행하는 프로그램일 수 있다. 예를 들면, 이미지 프로세서(118)는 이미지 및 이미지 처리 명령을 수신하고, 이미지의 하나 이상의 구현을 생성하라는 명령에 기초하여 수신된 이미지에 대하여 하나 이상의 변형을 수행하도록 구성된 애플리케이션일 수 있다. 예를 들면, 이미지 프로세서(118)는 이미지 및 그 이미지를 회전시키라는 명령을 수신할 수 있다. 이미지 프로세서(118)는 그 후 회전을 행하여 회전된 이미지 구현을 포함한 처리된 이미지를 출력할 수 있다.
- <21> 운영 체제(120)는 시스템(100)의 하드웨어 및 소프트웨어를 관리하는 소프트웨어 프로그램일 수 있고, 임의 유형의 운영 체제일 수 있다. 예를 들어, 운영 체제는 마이크로소프트 윈도우즈(Microsoft Windows), 리눅스(Linux) 등일 수 있다. 스택(122)은 데이터 저장 디바이스(114)로부터의 데이터를 저장하는 데 사용되는 임시 데이터 구조일 수 있다. 예를 들면, 스택은 이미지 및 대응하는 처리 명령을 애플리케이션(116)으로부터 이미지 프로세서(118)에 저장할 수 있고, 이미지 프로세서(118)는 그 후 스택(122)에 저장된 이미지에 액세스할 수 있다.
- <22> 예시적인 실시예에 따라, 시스템(100)은 예로서, TCP/IP 네트워크(124)와 같은 네트워크를 통해 원격 컴퓨터와 통신하기 위한 논리적 접속을 이용하여 네트워크된 환경에서 동작할 수 있다. 예를 들면, TCP/IP 네트워크(124)는 인터넷, 기업 인트라넷, LAN(local area network), 무선 네트워크, WAN(wide area network) 및/또는 임의의 기타 네트워크일 수 있다. 시스템(100)은 네트워크 인터페이스 유닛(126)을 통해 TCP/IP 네트워크(124)와 통신할 수 있다. 네트워크 인터페이스 유닛(126)은 시스템(100)으로 하여금 TCP/IP 네트워크(124)와 통신하게 하도록 구성된 디바이스 및/또는 소프트웨어 프로그램일 수 있다. 예를 들면, 네트워크 인터페이스 유닛(126)은 네트워크 카드, 네트워크 인터페이스 제어기(NIC), 모뎀, 또는 임의의 기타 통신 허용 디바이스일 수 있다.
- <23> 시스템 버스(112)는 CPU(102), 시스템 메모리(104), 데이터 저장 디바이스(114), 네트워크 인터페이스 유닛(126), 및 입/출력 제어기(128)에 결합될 수 있다. 시스템 버스(112)는 둘 이상의 시스템 컴포넌트 간에 데이터의 전달을 허용할 수 있다. 예를 들면 시스템 버스(112)는 시스템 메모리(104) 및 CPU(102)와의 데이터 전달을 허용할 수 있다. 시스템 버스(112)는 또한 예를 들면, 데이터 저장 디바이스(114)를 입/출력 제어기(128)에 결합하는 것과 같이, 시스템(100)의 다른 부분들을 서로 연결시킬 수 있다.
- <24> 도 2는 다른 예시적인 실시예에 따른 시스템(200)의 블록도이다. 시스템(200)은 상술한 바와 같이, 애플리케이션(116) 및 이미지 프로세서(118)를 포함할 수 있다. 예를 들면, 애플리케이션(116)은 이미지 프로세서(118)에게, 라인(204)을 통해 이미지(202)를 제공하고, 라인(208)을 통해 이미지 프로세스(또는 이미지 처리) 명령(206)을 제공하도록 구성될 수 있다. 이미지(202) 및 프로세스 명령(206)은 예를 들면, 명령 및 하나 이상의

파라미터를 이미지 프로세서(118)에 전달(pass)하는 애플리케이션(116)에 의해 제공될 수 있다.

- <25> 이미지 프로세서(118)는 소프트웨어 프로그램 및/또는 이미지(202) 및 프로세스 명령(206)을 수신하도록 구성된 디바이스일 수 있다. 이미지 프로세서(118)는, 예를 들면, 디코더 객체(220), 및 변형 객체(222, 224, 226, 228)와 같은 하나 이상의 변형 객체를 인스턴스화함으로써 이미지(202)에서 프로세스 명령(206)을 수행할 수 있고, 이미지의 하나 이상의 구현(216, 218)을 포함하는 하나 이상의 처리된 이미지(들)(214)을 제공할 수 있다. 이 방식으로, 이미지 프로세서(118)는 예로서, 프로세스 명령(206)에 기초하여 이미지(202)를 처리하여 구현(216, 218)과 같은 이미지의 하나 이상의 구현을 제공할 수 있다.
- <26> 이미지(202)는 특정한 파일 유형(203)의 파일 내에 있을 수 있다. 파일 유형(203)은 파일 포맷 또는 이미지 데이터를 압축하는 방법일 수 있고 래스터 그래픽 파일 유형, 디지털 파일 유형, 또는 기타 파일 유형과 같은 임의의 수락가능한 파일 유형일 수 있다. 예를 들면, 파일 유형(203)은 JPEG, GIF(Graphics Interchange Format), PNG(Portable Network Graphics), RAW(즉 코딩되지 않은), 또는 임의의 기타 파일 유형일 수 있다. 이미지(202)는 임의의 수락가능한 파일 타입(203)을 포함하는 파일 내에 있을 수 있다. 이미지(202)는 또한 하나 또는 다중 이미지를 포함할 수 있다. 각 이미지는 또한 이미지(또는 픽셀) 포맷(205)을 포함할 수 있다. 예를 들면, 이미지 포맷(205)은 RGB, CMYK, YCbCr, 또는 임의의 색공간이나 픽셀 포맷일 수 있다.
- <27> 프로세스 명령(206)은 단일한 프로세스 명령(206) 또는 다중 프로세스 명령일 수 있다. 프로세스 명령(206)은 예를 들면, 변형 값(210), 구현 값(212), 또는 양쪽 모두를 포함할 수 있다. 프로세스 명령(206)은 이미지(202)와 연관될 수 있다. 변형 값(202)은 예를 들면, 이미지에서 어떤 변형(들)을 수행할지를 나타낼 수 있고, 또한 변형(들)의 애플리케이션의 특정 순서를 나타낼 수 있다. 변형은, 예를 들면, 스케일러 변형(scaler transforms), 포맷 변환기 변형, 및/또는 대칭이동/회전자 변형을 비롯한, 이미지(202)에서 수행될 수 있는 임의의 잠재적인 변경 또는 처리일 수 있지만 이에 제한되는 것은 아니다. 추가적인 및/또는 상이한 변형이 또한 사용될 수 있다.
- <28> 구현 값(212)은 구현의 개수를 식별할 수 있고/있거나 이미지 프로세서(118)에 의해 출력될 구현 및/또는 처리된 이미지(들)을 식별할 수 있다. 구현(216, 218)은 각각 이미지(202)에 대한 하나 이상의 변형의 애플리케이션 전 또는 후의 이미지(202)의 버전일 수 있다. 구현 값(212)은, 예를 들면, 또한 어떤 변형이 어떤 구현(216, 218)을 위해 수행할 지를 나타낼 수 있고, 또한 변형을 수행할 순서를 식별할 수 있다. 처리된 이미지(들)(214)은, 이미지(202)에 대한 하나 이상의 변형의 애플리케이션 전 또는 후에, 이미지 프로세서(118)로부터의 하나 이상의 이미지 출력일 수 있다. 각 처리된 이미지(들)(214)은 하나 이상의 구현(216, 218)을 포함할 수 있거나, 구현으로 지칭될 수 있다.
- <29> 예를 들면, 애플리케이션(116)은 YCbCr의 포맷(205)을 갖는 이미지(202)를 이미지 프로세서(118)에 송신할 수 있다. 애플리케이션(116)은 또한 프로세스 명령(206)을 이미지 프로세서(118)에 송신할 수 있다. 프로세스 명령(206)은 예를 들면, 변형 값(210) 및 구현 값(212)을 포함할 수 있는데, 이는 단지 예시에 불과할 뿐, 실시예들은 이에 제한되지 않는다. 변형 값(212)은 스케일러 변형, 포맷 변환기 변형 및 대칭이동/회전자 변형, 및/또는 이미지에 대하여 수행될 기타 변형을 식별할 수 있다. 구현 값(212)은 예를 들면, 구현(216)이 수신된 이미지(202)를 크기조정(scale)(예를 들면 30%만큼)함으로써 생성되고, 구현(218)이 수신된 이미지(202)를 포맷 변환(예를 들면 YCbCr에서 RGB로)한 후 회전(예를 들면, 90도만큼)시킴으로써 생성되어야 한다는 것을 예로서 나타낼 수 있다. 이들은 단지 예시에 불과하며, 각종 실시예들이 이에 제한되지 않는다. 이미지 프로세서(118)는, 그 후 예를 들면, 스케일러 객체(222), 포맷 변환기 객체(224) 및/또는 대칭이동/회전자 객체(226)를 인스턴스화함으로써 지정된 변형을 수행하여 이미지를 처리하고, 두 개의 구현(216, 218)을 포함하는 처리된 이미지(들)(214)을 출력할 수 있다.
- <30> 이미지 프로세서(118)는 예를 들면, 스케일러 객체(222), 포맷 변환기 객체(224) 및/또는 대칭이동/회전자 객체(226)와 같은, 다수의 변형 객체와 하나의 디코더 객체(220)를 포함할 수 있다.
- <31> 디코더 객체(220)는 이미지(202)를 현재 파일 유형(203) 상태에서부터 디코딩 및/또는 압축해제(decompress)하여, 이미지 프로세서(118) 또는 하나 이상의 변형 객체가 예를 들면, 이미지(202)에 대하여 변형을 수행할 수 있게 한다.
- <32> 스케일러 객체(222)는 이미지(202)에 대하여 스케일러 변형을 수행하도록 구성될 수 있다. 스케일러 객체(222)(또는 스케일러 변형)는 이미지를 크기조정할 수 있는데, 이 크기조정에는 이미지(202)의 크기를 전체적으로 또는 부분적으로 늘이거나 줄이는 것이 포함된다. 스케일러 객체(222)에 의해 수행되는 크기조정(scaling)

은 이미지의 해상도를 바꾸지 않고 수행될 수 있거나, 이미지의 해상도를 바꾸거나 변경하는 것을 포함할 수도 있다. 예를 들어 변형 값(210)은 이미지(202)의 크기를 50%만큼 늘이는 스케일러 변형을 수행하는 것을 나타낼 수 있다. 이러한 이미지 처리 명령에 응답하여, 이미지 프로세서(118) 및/또는 스케일러 객체(222)는 통상적으로 이미지를 50%만큼 크기조정하여 50% 크기조정된 구현을 생성할 수 있다.

<33> 포맷 변환기 객체(224)는 수신된 이미지(202)의 포맷(205)을 바꾸거나 이미지(202)의 포맷(205)을 다른 포맷으로 변환할 수 있다. 예를 들면, 이미지(202)는 RGB 포맷으로 이미지 프로세서(118)에 제공될 수 있고, 변형 값(210)은 이미지를 CMYK로 변환하기 위한 포맷 변환기 변형을 식별할 수 있다. 이러한 프로세스 명령에 응답하여, 이미지 프로세서(118) 및/또는 이미지 프로세서(118)의 포맷 변환기 객체(224)는 수신된 이미지(202)의 포맷(205)을 예를 들면, RGB에서 CMYK로 변환할 수 있다.

<34> 대칭이동/회전자 객체(226)는 이미지(202)를 대칭이동하거나 회전할 수 있다. 대칭이동/회전자 객체(226)(또는 변형)는 이미지(202)를 상-하로 또는 좌-우로 대칭이동시킬 수 있거나 이미지(202)를 소정의 각도로 회전시킬 수 있다. 예를 들어 변형 값(210)은 이미지(202)를 상하로 대칭이동하거나 이미지를 반시계방향으로 30도 회전하기 위한 대칭이동/회전자 변형을 식별할 수 있다. 이러한 이미지 처리 명령에 응답하여, 이미지 프로세서(118) 및/또는 대칭이동/회전자 객체(226)는 그 후 요청받은 대로 이미지(202)를 대칭이동 또는 회전시킬 수 있다.

<35> 상술한 바와 같이, 디코더 객체(220)는 수신된 이미지를 디코딩할 수 있다. 디코더 객체(220)는 이미지(202)를 디코딩하도록 구성되는 것 외에도, 원시 변형 객체(228)를 포함할 수 있다. 원시 변형 객체(228)는 예로서, 원시 스케일러 객체, 원시 포맷 변환기 객체, 및/또는 원시 대칭이동/회전자 객체와 같은, 원시적으로 수행될 수 있는 임의의 변형 객체(또는 이미지 처리 객체)를 포함할 수 있다. 원시 변형 객체(228)는, 원시 변형 객체(228)가 변형 능력을 포함하는 경우에, 디코더 객체(220) 내에서 원시적으로 임의의 변형 또는 변형의 조합을 수행할 수 있다. 예시적인 실시예에서, 디코더 객체(220)의 원시 변형 객체(228)는, 디코더 객체(220)가 수신된 이미지(202)를 디코딩하는 것과 동시에 또는 거의 동시에 이미지에 대하여 하나 이상의 원시 변형을 수행할 수 있다. 이는 디코더 객체(220) 또는 원시 변형 객체(228) 내에서 원시적으로 변형(들)을 수행함으로써, 예를 들면, 수신된 이미지를 재 샘플링(resampling)하지 않은 채 요청받은 변형을 이미지에 대하여 수행하는 것을 통상적으로 포함할 수 있다. 이는 몇몇 경우에서, 예를 들면, 이미지에 대하여 재 샘플링 또는 다운-샘플링이 일어날 수 있는 비-원시 변형(non-native transform)에 비해 향상된 이미지 품질을 제공할 수 있다.

<36> 원시 변형 객체(228)는 예를 들면, 이미지를 재 샘플링하거나 다운 샘플링할 필요 없이, 이미지에 하나 이상의 변형이 원시적으로 적용되게 해준다. 따라서, 몇몇 경우에는, 원시 변형은 예를 들면, 향상된 이미지 품질을 유리하게 제공하거나 이미지 품질의 저하를 회피하는 이점을 제공할 수 있다. 원시 변형 객체(228)는 임의의 개수의 변형 객체를 포함할 수 있다. 예를 들면, 원시 변형 객체(228)는 원시 스케일러 객체, 원시 포맷 변환기 객체 및/또는 원시 대칭이동/회전자 객체를 포함할 수 있어서 모든 요청받은 변환들이 이미지(202)에서 원시적으로 수행되게 한다.

<37> 몇몇 경우에서, 애플리케이션(116)으로부터 프로세스 명령(206)을 수신한 후에, 디코더 객체(220)는 하나 이상의 변형 객체를 인스턴스화하여 이미지(202)에 대하여 요청받은 처리를 수행할 수 있다. 원시 변형 객체(228)가, 요청받은 변형 또는 처리를 수행하는 것이 가능하면, 이미지는 원시 객체를 이용하여 원시적으로 처리될 수 있다. 그렇지 않고, 요청된 변형이 원시 변형으로서 이용가능하지 않다면, 인스턴스화된 변형 객체, 예를 들면, 객체(222, 224 또는 226)가 이미지에 대하여 (예를 들면, 비-원시(non-native)) 처리를 수행할 수 있는데, 이는 몇몇 경우에 이미지를 재 샘플링하는 것을 포함할 수 있다. 따라서, 요청된 변형의 일부(또는 심지어 전부)가 이미지에 대하여 원시적으로 수행될 수 있는 반면에, 나머지(또는 심지어는 전체) 변형은 예를 들면, 다른 변형 객체에 의해 비 원시적으로 후속하여 수행될 수 있다.

<38> 예시적인 실시예에서, 수신된 프로세스 명령(206)은 25%만큼 크기조정하기 및 이미지 포맷 YCbCr에서 RGB로 변환하기와 같은, 이미지(202)에서 수행될 두 개의 변형을 요청할 수 있다. 이 예시에서는, 디코더 객체(220)가 이미지 프로세서(118)에 의해 인스턴스화될 수 있다. 또한, 프로세스 명령(206)에 기초하여, 스케일러 객체 및 포맷 변환기 객체가 인스턴스화되고 디코더 객체(220)와 연관될 수 있다. 이 예에서는, 크기조정은 원시 변형 객체(228)에 의해 원시적으로 수행될 수 있는 반면에, 요청받은 YCbCr로부터 RGB로의 포맷 변환은 원시적으로 가능하지 않을 수 있다. 이는 단지 예시에 불과하다. 따라서, 이미지를 수신하면, 디코더 객체(220)는 이미지를 디코딩할 수 있고 원시 변형 객체(228)는 요청받은 대로 이미지를 25%만큼 원시적으로 크기조정할 수 있다. 다음으로, 디코딩되고 크기조정된 이미지는 예를 들면 이미지가 RGB 포맷으로 변환될 수 있는 포맷 변환기 객체

(224)에 패스되어, (디코딩되고, 크기조정되고 포맷 변환된) 이미지의 구현으로서 출력될 수 있다.

- <39> 다른 예시적인 실시예에서는, 이미지 프로세서(118)가 이미지의 다중 구현을 출력할 수 있다. 각 구현은 상이한 버전의 이미지, 예를 들면 구현 마다 이미지에 대하여 상이한 변형이 수행되었을 수 있는 이미지일 수 있다. 예를 들면, 이미지 프로세서(118)는 원본 이미지(202)일 수 있는 제1 이미지 구현, 디코딩된 후의 이미지(202)일 수 있는 제2 이미지 구현, 및 25%만큼 크기조정되고 RGB로 변환된, 디코딩된 후의 이미지일 수 있는 제3 이미지 구현을 출력할 수 있다. 예시적인 실시예에서는, 이 예시에서의 하나 이상의(또는 모든) 변형이 원시적으로 수행되었을 수 있다. 이는 단지 예시에 불과하다. 예를 들어, 프로세스 명령(206)은 예를 들면, 구현 값(들)(212)을 통해 제공되거나 생성될 하나 이상의 구현을 식별할 수 있다.
- <40> 도 3A는 예시적인 실시예에 따라 다중 구현을 제공하도록 이미지가 처리되는 시스템(300A)의 도면이다. 시스템(300A)은 디코더 객체(202)에 의해 수신되는 이미지(202) 및 프로세스 명령(206)을 포함할 수 있다. 이미지(202)는 과일일 수 있고 상술한 바와 같이, 임의의 이미지 포맷(205)으로 된 임의의 이미지 또는 여러개의 이미지를 포함할 수 있다. 상술한 바와 같이 프로세스 명령(206)은 예를 들면, 변형 값 및/또는 구현 값, 또는 제공되거나 생성될 하나 이상의 이미지 구현을 식별하는 기타 정보를 포함할 수 있다. 프로세스 명령(206)은 예를 들면 얼마나 많은 이미지(202) 구현들을 출력할지, 각 구현을 위해 어떤 변형(들)을 수행할지, 그리고 구현을 생성하기 위해 어떤 순서로 변형(들)을 수행할지를 예를 들어 나타낼 수 있다.
- <41> 디코더 객체(220)는 이미지 메모리(302)를 포함할 수 있다. 디코더 객체(220)는 스케일러 객체(222), 포맷 변환기 객체(224), 및/또는 대칭이동/회전자 객체(226)에 결합되거나 그것들과 연관될 수 있다. 상술한 바와 같이, 디코더 객체(220)는 수신된 이미지(202)를 상술한 것 처럼 디코딩하거나 압축해제할 수 있다. 또한, 스케일러 객체(222)는 이미지를 크기조정할 수 있다. 포맷 변환기 객체(224)는 이미지의 포맷(205)을 다른 포맷으로 변환할 수 있고, 대칭이동/회전자 객체(226)는 이미지를 대칭이동시키거나 회전시킬 수 있다. 다른 변형 객체가 제공될 수도 있다.
- <42> 예시적인 실시예에 따르면, 이미지 프로세서(118)는 예를 들어 구현(304A, 304B, 304C, 및 304D)과 같은 하나 이상의 이미지 구현을 출력할 수 있다. 각 구현은 이미지에 대하여 수행된 변형이 없었거나 하나 이상의 변형이 있는 후의 이미지(202)를 포함할 수 있다. 변형이 없었던 후에 출력되는 경우에, 이는 원본 수신 이미지(202)를 출력할 수 있다. 디코더 객체(220)의 이미지 메모리(302)는 원본 이미지(202)에 대하여 변형이 수행된 이후라도, 그 원본 이미지(202)가 이용가능하도록 수신된 이미지(202)를 저장할 수 있다. 이미지 메모리(302)는, 예를 들면, 하나 이상의 변형이 이미지에 대하여 수행된 후의 하나 이상의 중간 단계 구현을 저장할 수 있다. 이 중간 단계 구현은 구현으로서 출력될 수 있고/있거나 다른 구현으로서 출력하기 위해 다른 변형에 의해 더 처리될 수 있다. 예시적인 실시예에서, 이미지 메모리(302)는 디코더 객체(220)의 일부일 수 있거나, 시스템 메모리(104) 내에(예를 들면, 도 1의 RAM(106)내에) 제공될 수 있다.
- <43> 예를 들면, 원본 이미지(202)는 개(dog)의 이미지 또는 다른 객체의 이미지일 수 있고, 프로세스 명령은 원본 크기의 150%까지 이미지의 크기를 조정하는 제1 구현, 및 이미지의 상하를 대칭이동시키는 제2 구현을 포함할 수 있다. 이미지 메모리(302)는 원본 개 이미지(dog image; 202)를 저장하여, 이미지가 제1 구현을 위해 150%로 크기조정된 후에도, 원본 이미지(202)를 상하 대칭이동시키기 위해 제2 이미지 구현이 이미지 메모리(302)에 저장된 원본 이미지(202)를 계속 사용할 수 있다. 이미지 저장소(302)에 저장된 원본 이미지는 원본 이미지의 구현을 출력하는 데 사용될 수 있다. 예시적인 실시예에서는, 이미지 메모리(302)는 또한 처리되고 있는 이미지(202)의 상태 및/또는 이미지에 대하여 하나 이상의 변형이 수행된 후의 중간 상태 구현을 식별하는 상태 정보를 저장할 수 있다. 예를 들면, 제2 구현(304B)을 제공하는 중에, 이미지 메모리(302)는 스케일러 객체(222)가 스케일러 변형을 수행한 후의 이미지 또는 이미지 상태 정보를 저장할 수 있다. 그 후, 예를 들면, 포맷 변환기 객체(224)가 이미지 메모리(302)로부터 이미지 상태 정보를 검색(retrieve)하여 크기조정된 이미지에 대하여 포맷 변환기 변형을 수행할 수 있다.
- <44> 도 3A에서 시스템(300A)에 의해 수행되는 이미지 처리가 이제 간략하게 기술될 것이다. 예시적인 실시예에서, 디코더 객체(220)는 통상 우선적으로 제공될 수 있다. 디코더 객체(220) 다음에, 시스템(330A)의 스케일러 객체(222), 포맷 변환기 객체(224), 및 대칭이동/회전자 객체(226)와 같은 하나 이상의 변형 객체가 프로세스 명령(206)의 구현 값(212)에 기초하는 등과 같은, 프로세스 명령(206)에 기초하여 직렬 또는 병렬로 조직(organized)되거나, 양쪽 모두의 조합으로 조직될 수 있다. 예를 들면, 도 3A의 예에서 도시된 바와 같이, 디코더 객체(220)는 RGB 포맷의 개의 이미지(202)를 수신할 수 있다. 디코더 객체(220)는 또한 이미지(202)와 연관된 프로세스 명령(206)을 수신할 수 있는데, 여기서 프로세스 명령(206)은 예를 들면, 이미지의 네 개의 구현

을 생성하기 위한 명령을 포함한다. 제1 요청받은 구현(304A)은 디코더 객체에 의해 디코딩되고 스케일러 객체에 의해 (예를 들면, 50%만큼) 크기조정된 이미지(202)를 포함할 수 있다. 제2 요청받은 구현(304B)은 디코더 객체에 의해 디코딩되고, 스케일러 객체에 의해 (예를 들면, 50%만큼) 크기조정되고 나서, 포맷 변환기 객체에 의해 (예를 들면, RGB에서 CMYK로) 포맷 변환된 이미지를 포함할 수 있다. 제3 요청받은 구현(304C)은 디코더 객체에 의해 디코딩되고 나서 대칭이동/회전자 객체에 의해 (예를 들면, 90도 만큼) 회전된 이미지를 포함할 수 있다. 또한 제4 구현(304D)은 임의의 순서로, 디코더 객체에 의해 디코딩되고, 대칭이동/회전자 객체에 의해 (예를 들면, 30도 만큼) 회전되고, 포맷 변환기 객체에 의해 (예를 들면, RGB에서 CMYK로) 포맷 변환된 이미지(202)를 포함할 수 있다.

- <45> 예를 들어, 이미지(202)를 수신하면, 디코더 객체(220)는 추후의 처리 및/또는 미처리된 구현으로서 가능한 출력을 위해, 원본 이미지(202)의 사본을 이미지 메모리(302)에 저장할 수 있다. 도 3A에 도시된 예시적인 실시예에서는, 시스템(300A)의 디코더 객체(220)는 예를 들면, 프로세스 명령이 요청한 스케일러 객체, 포맷 변환기 객체, 또는 대칭이동/회전자 객체를 위한 원시 변형 객체(228)를 포함하지 않는다. 따라서, 이 예시에서는, 요청받은 변형은 비-원시 변형 객체에 의해 수행될 수 있다.
- <46> 이미지(202)를 수신하면, 디코더 객체(220)는 수신된 이미지의 디코딩을 수행하도록 인스턴스화될 수 있다. 또한 하나 이상의 추가적인 객체들이 인스턴스화되어서 프로세스 명령(206)에 기초하여 디코더 객체(220)에 연결되거나 그와 연관될 수 있다. 이는 하나 이상의 객체를 결정하는 것 및 프로세스 명령에 따른 요청받은 이미지 구현을 생성하기 위해 이미지를 처리할 객체에 대한, 배열(arrangement), 예를 들면, 직렬 및/또는 병렬 배열 및/또는 순서를 결정하는 것을 포함할 수 있다. 예를 들면, 도 3A에 도시된 바와 같이, 이는 예로서, 제2 구현(304B)을 생성하기 위해 스케일러 객체(222) 및 포맷 변환기 객체(224)는 직렬로 제공되어야 하는 반면에, 제3 구현(304C)을 생성하기 위해 대칭이동/회전자 객체(226)는 (제2 구현을 생성하기 위해 이용되는 객체들에 대해) 병렬로 위치시킬 것을 결정하는 것을 포함할 수 있다. 디코더 객체(220)가 이미지(202)를 원시적으로 디코딩한 후에, 네 개의 예시적인 구현이 생성될 수 있다.
- <47> 언급한 바와 같이, 도 3A의 제2 구현(304B)의 경우, 디코딩된 이미지는, 스케일러 객체(222)에 의해 크기조정되고 포맷 변환기 객체(224)에 의해 포맷 변환되어 제2 구현(304B)을 생성할 수 있다. 객체들(222 및 224)은 예를 들면, 비-원시 변형일 수 있다. 예를 들면, 비-원시 변형은 디코딩된 이미지를, 예를 들어 재 샘플링하거나 심지어는 다운 샘플링하는 것을 포함할 수 있다. 다른 예시적인 실시예에서는, 스케일러 객체(222) 및 포맷 변환기 객체(224) 양쪽 모두가 원시적으로 이용가능할 수 있어서, 디코더 객체(220) 내에서 원시적으로 수행될 수 있다. 디코딩된 이미지가 스케일러 객체(222)에 의해 크기조정된 후에 중간 단계 구현이 저장될 수 있음을 유념해야 한다. 스케일러 객체(222)로부터 출력된 이 중간 단계 구현은 그 후 제1 구현(304A)으로서 출력될 수 있다.
- <48> 예를 들어 제3 구현(304C)을 생성하기 위해, 이미지 메모리(302)에 저장된 디코딩된 이미지는, 대칭이동/회전자 객체(226)에 의해 회전되어 제3 구현(304C)을 생성할 수 있다. 마지막으로, 제4 구현(308D)을 생성하기 위해, 이미지 메모리(302)에 저장된 디코딩된 이미지는 포맷 변환기 객체(224)에 의해 먼저 포맷 변환되고 나서(예를 들면, 제2 구현(304B)에서 나중이었던 것과는 반대임) 대칭이동/회전자 객체(226)에 의해 회전되어 제4 구현(308D)을 생성할 수 있다. 다른 예시적인 실시예에서는, 포맷 변환기 객체(224), 대칭이동/회전자 객체(226)를 비롯한 제4 구현(304D)을 생성하는 데 사용된 모든 객체들이 원시적으로 제공될 수 있는데, 이는, 예시적인 실시예에서 디코더 객체(220)에 의해 원시적 처리가 수행되어 이미지의 제4 구현(304D)을 생성할 수 있음을 의미한다.
- <49> 시스템(300A)에서, 스케일러 객체(222), 포맷 변환기 객체(224) 및 대칭이동/회전자 객체(226)는 예를 들면, 상술한 바와 같이, 적어도 몇몇 경우에는 이들 각각의 변형을 이미지에서 비-원시적으로 (즉, 그 객체들은 재 샘플링, 다운 샘플링 및/또는 이미지 품질을 떨어뜨릴 수 있음) 수행할 수 있다. 또한 상술한 바와 같이 시스템(300A)에 대하여, 변형, 변형의 순서, 및 적용되는 변형의 수는, 단지 예시적인 목적을 위한 것이다. 변형의 예시적인 실시예에서는, 변형의 순서 및 시스템(300A)에 적용되는 구현의 개수는 다양할 수 있다. 또한, 위에서 단지 예시적으로 설명된 것 처럼, 각 구현이 개별적으로 처리될 필요는 없고, 다중 구현이 동시에 처리될 수 있다. 예시적인 실시예에서, 다중 이미지(202) 및 다중 프로세스 명령(206)이 또한 수신될 수 있다.
- <50> 도 3B는 다른 예시적인 실시예에 따라 다중 구현을 제공하도록 이미지가 처리되는 시스템(300B)의 블록도이다. 시스템(300B)은 디코더 이미지(220)에 의해 수신되는 이미지(202) 및 프로세스 명령(206)을 포함할 수 있다. 시스템(300B)(도 3B)은 시스템(300A)(도 3A)과 매우 유사할 수 있으며, 두 도면 간의 차이점이 이하에서 설명될

것이다. 상술한 바와 같이, 이미지(202)는 수락가능한 파일 유형(203)의 파일일 수 있고/있거나 임의의 호환가능한 이미지 포맷(205)인 임의의 이미지 또는 여러개의 이미지일 수 있다. 상술한 바와 같이, 프로세스 명령(206)은 변형 값(210) 및 구현 값(212)을 포함할 수 있다. 프로세스 명령(206)은 예를 들면 이미지(202)의 얼마나 많은 구현이 출력될지, 각 구현을 위해 어떤 변형이 수행될지, 및 변형(들)이 어떤 순서로 수행될지를 나타낼 수 있다.

<51> 디코더 객체(220)는 예를 들면, 이미지 메모리(302), 원시 대칭이동/회전자 객체(306), 및 원시 포맷 변환기 객체(308)를 포함할 수 있다. 그러나 시스템(300B)에서, 각의 변형을 수행하는 대칭이동/회전자 객체(226) 및 포맷 변환기 객체(224) 대신에, 원시 대칭이동/회전자 객체(306) 및 원시 포맷 변환기 객체(308)가, 이미지(202)에서, 대칭이동/회전자 변형 및 포맷 변환기 변형을 각각 수행할 수 있다. 따라서, 이 예시에서는, 원시 대칭이동/회전자 객체(306) 및 원시 포맷 변환기 객체(308)가 이용가능하여, 이미지를 처리하는 데, 비-원시 객체(226 및 224) 대신에 각각 사용될 수 있다. 시스템(300B)에서, 원시 객체가 이미지(202)에서 변형을 수행하도록 이용가능한 경우에는, 비-원시 객체보다는 원시 객체가 바람직하게 사용될 수 있다. 그러면, 예를 들어, 처리된 이미지는 원시 객체가 이용가능한 비-원시 객체를 통해(또는 바이패스하여) 패스될 수 있다. 따라서, 프로세스 명령에 따라 다중 구현을 생성하기 위해, 네 개의 구현(304E, 304F, 304G 및 304H)은 원시 객체들, 비-원시 객체들 또는 원시 및 비-원시 객체들의 조합에 기초하여 생성될 수 있다.

<52> 시스템(300A)에서와 마찬가지로, 시스템(300B)의 스케일러 객체(222), 포맷 변환기 객체(224), 및 대칭이동/회전자 객체(226), 및/또는 기타 객체들은 프로세스 명령(206)에 기초하여 직렬 또는 병렬로 조직되거나, 직·병렬 조합으로 조직될 수 있다. 예를 들면, 프로세스 명령(206)은 네 개의 구현(304E 내지 304H)을 생성하기 위한 명령을 포함한다. 구현(304E 내지 304H)을 위한 요청 또는 프로세스 명령은, 예를 들면, 구현(304A 내지 304D)(도 3A)을 위한 것과 동일한 프로세스 명령일 수 있지만, 시스템(300B)은 이미지 처리를 수행하여 다중의 요청받은 구현을 생성하기 위해 하나 이상의 원시 객체를 유리하게 이용할 수 있다. (비록 다른실시예에서는 이용가능하더라도) 이용가능한 원시 스케일러 객체가 없을 수 있으므로, 제1 구현(304E)은 원시 디코더(220), 비-원시 스케일러 객체(222)를 이용하여 생성될 수 있다. 그러나, 제2 구현(304F)은 원시 디코더(220), 비-원시 스케일러 객체(222)(예를 들면, 원시 스케일러 객체가 이용가능하지 않으므로), 및 원시 포맷 변환기 객체(308)를 이용하여 생성될 수 있다. 원시 포맷 변환기 객체(308)가 요청받은 포맷 변환을 수행하는 데 이용가능할 수 있기 때문에, 인스턴스화된 비-원시 포맷 변환기 객체(224) 대신에 원시 포맷 변환기(308)를 사용하여 제2 이미지 구현(304F)을 생성할 수 있다. 그러나 예시적인 실시예에서, 비-원시 변형 전에 어떠한 원시 변형도 통상적으로 수행될 수 있다. 예를 들면, 구현(304F)을 생성하기 위해, 비-원시 스케일러 객체(222) 이전에 원시 포맷 변환기 객체(308)를 수행할 수 있다.

<53> 이와 비슷하게, 제3 구현(304G)이 원시 디코더 객체(220) 및 원시 대칭이동/회전자 객체(306)를 이용하여 생성될 수 있다. 이 예시에서는 비-원시 대칭이동/회전자 객체(226) 대신에 원시 대칭이동/회전자 객체(306)가 사용될 수 있다. 제4 이미지 구현(404H) 또한 원시 디코더 객체(220), 원시 포맷 변환기 객체(308), 및 원시 대칭이동/회전자 객체(306)를 비롯한 모든 원시 객체들을 이용하여 생성될 수 있다. 원시 객체들(308 및 306)은, 현존하거나 이용가능한 경우에, 대응하는 비-원시 객체들(224 및 226) 각각 대신에 사용될 수 있다. 예를 들면, 이미지가 (원시 객체들(306 및 308)에 의해 처리되는 것을 포함하여) 디코더 객체(220)에 의해 처리된 후에, 처리된 이미지는 (포맷 변환 및 회전이 원시 포맷 변환기(308) 및 원시 대칭이동/회전자 객체(306)에 의해 이미 수행되었기 때문에) 처리 없이도, 비-원시 객체들(224 및 226)을 통과할 수 있다. 언급한 바와 같이, 원시 객체들이 이용가능한 경우, 이 원시 객체들을 사용하면, 몇몇 경우에 요청받은 처리를 원시적으로, 예를 들면, 이미지를 재 샘플링하지 않고/않거나 품질의 저하 없이, 제공할 수 있다.

<54> 이 방식으로, 예시적인 실시예에 따라, 이미지의 단일한 구현만을 제공하는 대신에, 이미지의 다중 구현들이 생성되어 제공될 수 있다. 또한, 다중 구현의 생성 및 출력은 원시 객체들을 사용하여 출력 이미지 또는 구현의 품질을 향상하기 위한 이미지 처리 또는 변형을 수행할 수 있다.

<55> 대안적인 실시예에서 시스템(300B)에 적용되는 변형, 변형의 순서, 및 변형의 개수는 다를 수 있다. 또한, 위에서 단지 예시적으로 논의된 것처럼, 각 구현이 개별적으로 처리될 필요는 없고, 다중 구현이 동시에 처리될 수 있다. 대안적인 실시예에서, 다중 이미지(202) 및 다중 프로세스 명령(206)이 또한 사용될 수 있다.

<56> 도 4 내지 도 6은 도 1, 2 및 3의 상술된 예시들 및/또는 기타 예시들 및 맥락과 관련하여, 동작적 흐름 또는 흐름도의 각종 예시들을 포함한다. 그러나, 동작적 흐름은 수많은 기타 실시예들 및 맥락에서, 및/또는 도 1, 2 및 3의 변경된 버전에서 실행될 수도 있음이 이해되어야 한다. 또한, 비록 각종 동작적 흐름들을 시퀀스로

나타냈지만, 각종 동작들은 도시된 순서와는 다른 순서로 수행될 수 있거나, 동시적으로도 수행될 수 있음이 이해되어야 한다.

- <57> 도 4는 예시적인 실시예에 따라, 하나 이상의 구현을 생성하도록 이미지에 대하여 복수의 변형을 수행하는 프로세스를 도시하는 흐름도이다. 시작 동작 후에, 동작적 흐름(400)은 이미지가 수신될 수 있는 수신 동작(410)으로 이동한다. 예를 들면, 이미지 프로세서(118)는 애플리케이션(116)으로부터 이미지(202)를 수신할 수 있다. 다른 예시적인 실시예에서는, 애플리케이션(116) 및 이미지 프로세서(118)는, 요구되는 것은 아니지만, 동일한 전체 시스템(200), 애플리케이션, 또는 디바이스의 일부일 수 있다.
- <58> 그 다음에 수신 동작(420)에서, 이미지에 대한 이미지 처리 명령(또는 프로세스 명령)이 수신될 수 있다. 예를 들면, 도 2에 도시된 바와 같이, 이미지 프로세서(118)는 애플리케이션(116)으로부터 프로세스 명령(206)을 수신할 수 있다. 프로세스 명령(206)은 이미지(202)와 연관되어 이미지(202)를 어떻게 처리할지를 나타낼 수 있다. 프로세스 명령(206)은 예를 들면, 변형 값(210) 및/또는 구현 값(212)을 포함할 수 있는데, 이는 단지 일 예시적인 실시예에 지나지 않는다.
- <59> 그 다음 수행 동작(430)에서, 이미지의 하나 이상의 구현을 제공하기 위해 이미지 처리 명령 또는 프로세스 명령에 기초하여 이미지에 대하여 복수의 변형이 원시적으로 수행될 수 있다. 예를 들면, 도 2에 도시된 바와 같이, 프로세스 명령(206)에 기초하여, 이미지 프로세서(118)가 이미지(202)에 대하여 변형을 수행하여, 처리된 이미지(들)(214)의 구현(들)(216, 218)을 제공할 수 있다. 또는, 도 3B에 도시된 바와 같이, 이미지(202) 및 프로세스 명령(206)이 디코더(220)에 의해 수신될 수 있고, 원시 대칭이동/회전자 객체(306) 및 원시 포맷 변환기 객체(308)를 이용하여 원시적으로, 복수의 변형을 수행한 후에, 예를 들어, 구현(304G 및 304H)을 비롯한 하나 이상의 구현이 제공될 수 있다.
- <60> 도 5는 예시적인 실시예에 따른 복수의 변형을 수행하기 위한 프로세스를 도시하는 흐름도이다. 시작 동작 후에, 동작적 흐름(500)은 이미지가 수신될 수 있는 수신 동작(510)으로 이동한다. 예를 들면, 이미지 프로세서(118)는 애플리케이션(116)으로부터 이미지(202)를 수신할 수 있다.
- <61> 그 다음에 수신 동작(520)에서, 이미지에 대한 이미지 처리 명령(또는 프로세스 명령)이 수신될 수 있다. 예를 들면, 도 2에 도시된 바와 같이, 프로세스 명령(206)이 이미지 프로세서(118)에 의해 애플리케이션(116)으로부터 수신될 수 있다. 프로세스 명령(206)은 이미지(202)와 연관되어 이미지(202)를 어떻게 처리할지를 나타낼 수 있고, 제공되거나 생성될 이미지의 하나 이상의 구현을 식별할 수 있다.
- <62> 그 다음에 수행 동작(530)에서, 이미지 처리 명령에 기초하여 이미지에 대하여 복수의 변형이 수행되어 복수의 구현을 제공할 수 있다. 복수의 구현 각각은 이미지에 대하여 원시적으로 수행되는 적어도 하나의 변형을 포함할 수 있다. 예를 들면, 도 3B에 도시된 바와 같이, 이미지(202) 및 프로세스 명령(206)은 디코더(220)에 의해 수신될 수 있다. 디코더 객체(220)가 이미지를 디코딩한 후에, 원시 포맷 변환기 객체(308)에 의해 포맷 변환기 변형이 수행되고 나서, 비-원시 스케일러(222)에 의해 비-원시 스케일러 변형이 수행되어 제2 구현(304F)을 생성할 수 있다. 다른 방법으로는, 원시 포맷 변환기 객체(308)에 의해 수행되는 원시 포맷 변환은 디코더(220)에 의해 이미지가 디코딩되는 때와 동시에 수행될 수도 있다. 상술한 바와 같이, 원시 변형은 비-원시 변형보다 먼저 수행될 수 있다. 예시적인 실시예에서, 원시 변형은 통상적으로 어떠한 순서로도 수행될 수 있지만, 통상적으로는 비-원시 변형보다 먼저 수행된다. 비-원시 변형 또한 통상적으로 어떠한 순서로도 수행될 수 있다. 또한, 제3 구현(304G)을 생성하기 위해, 이미지는 디코더 객체(220)에 의해 디코딩될 수 있고, 그 후에, 또는 그와 동시에, 예를 들면, 원시 대칭이동/회전자 객체(306)에 의해 회전 변형이 이미지에 대하여 수행될 수 있다. 따라서, 이 방식으로, 예로서, 다중 구현이 제공될 수 있는데, 구현들 각각은 해당 구현을 생성하기 위해 이미지에 대하여 수행되는 적어도 하나의 원시 변형을 포함하고 있다.
- <63> 도 6은 또 다른 실시예에 따른 하나 이상의 구현을 생성하기 위해 이미지에 대하여 하나 이상의 변형을 수행하기 위한 프로세스를 도시하는 흐름도이다. 시작 동작 후에, 동작적 흐름(600)은 이미지가 수신될 수 있는 수신 동작(610)으로 이동한다. 그 다음에 수신 동작(620)에서, 이미지 처리 명령이 수신될 수 있다.
- <64> 그 다음에 결정 동작(630)에서, 이미지 처리 명령(또는 프로세스 명령)에 기초하여, 이미지 처리 명령에 따라 이미지를 처리하고 이미지의 다중 구현을 제공하기 위해 하나 이상의 변형 및 변형에 대한 배열이 결정될 수 있다. 예를 들면, 이미지 프로세스 명령에 기초하여, 하나 이상의 변형(또는 객체)이 결정될 수 있다. 또한, 이미지 처리 명령에 따라, 이미지를 처리하여 이미지의 구현을 생성하기 위해, 이미지를 처리하기 위한 객체 또는 변형의 직렬 및/또는 병렬 배열과 같은 배열이 결정될 수 있다. 예를 들면, 도 3A에서는, 수신된 이미지(202)

의 요청받은 구현(304B)을 생성하기 위해, 이미지를 처리하기 위한 스케일러 객체(222) 및 포맷 변환기 객체(224)의 직렬 배열(디코더 객체(220) 다음)을 결정하는 것을 포함할 수 있다. 유사하게, 이 결정은 수신된 이미지를 처리하여 예를 들면, 304A, 304C 및 304D와 같은 다른 요청받은 구현을 생성하기 위해 변형 또는 객체의 순서 및 병렬 배열을 결정하는 것을 포함할 수 있다.

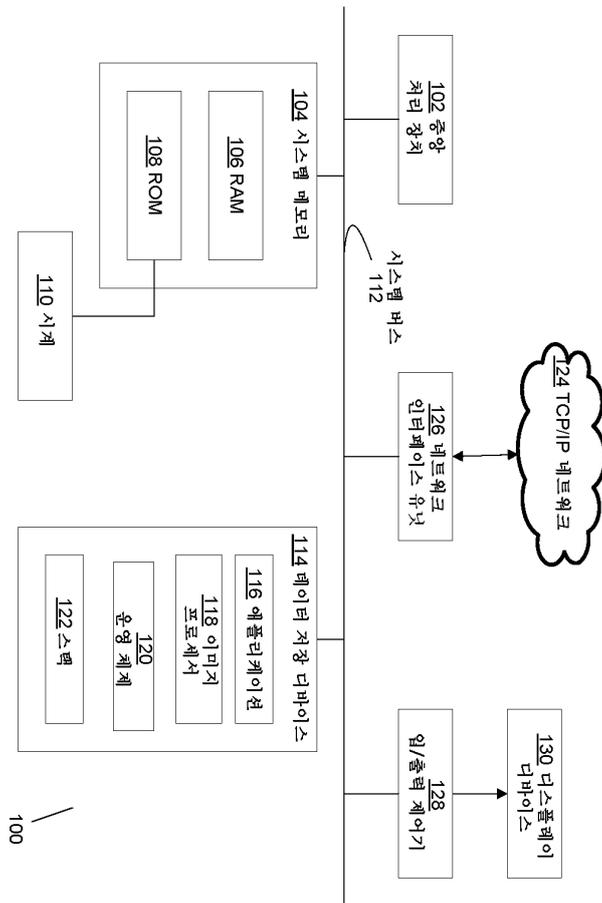
<65> 본원에 설명된 바와 같이 개시된 구현예들의 특정한 특징들이 예시되었지만, 이제 당업자에게는 많은 변경, 대체, 변화 및 등가물이 생겨날 수 있다. 따라서, 첨부된 청구항들이 그러한 변경 및 변화들을, 각종 실시예들의 사상 내에 들어가는 것으로서 포괄하고자 하는 것이 이해되어야 한다.

### 도면의 간단한 설명

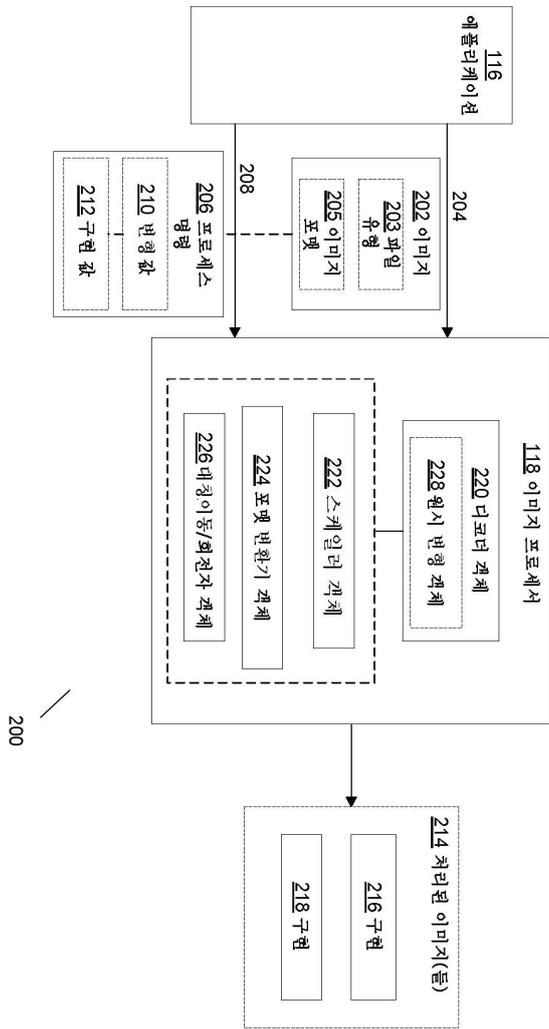
- <7> 도 1은 예시적인 실시예에 따른 컴퓨팅 시스템의 블록도.
- <8> 도 2는 다른 예시적인 실시예에 따른 시스템의 블록도.
- <9> 도 3A는 예시적인 실시예에 따른 다중 구현을 제공하도록 이미지가 처리되는 시스템(300A)의 도면.
- <10> 도 3B는 다른 예시적인 실시예에 따른 다중 구현을 제공하도록 이미지가 처리되는 시스템(300B)의 블록도.
- <11> 도 4는 예시적인 실시예에 따라, 하나 이상의 구현을 생성하도록 이미지에 대하여 복수의 변형을 수행하기 위한 처리를 도시하는 흐름도.
- <12> 도 5는 다른 예시적인 실시예에 따라, 하나 이상의 구현을 생성하도록 복수의 변형을 수행하기 위한 처리를 도시하는 흐름도.
- <13> 도 6은 또 다른 예시적인 실시예에 따라, 하나 이상의 구현을 생성하도록 이미지에 대하여 하나 이상의 변형을 수행하기 위한 처리를 도시하는 흐름도.

도면

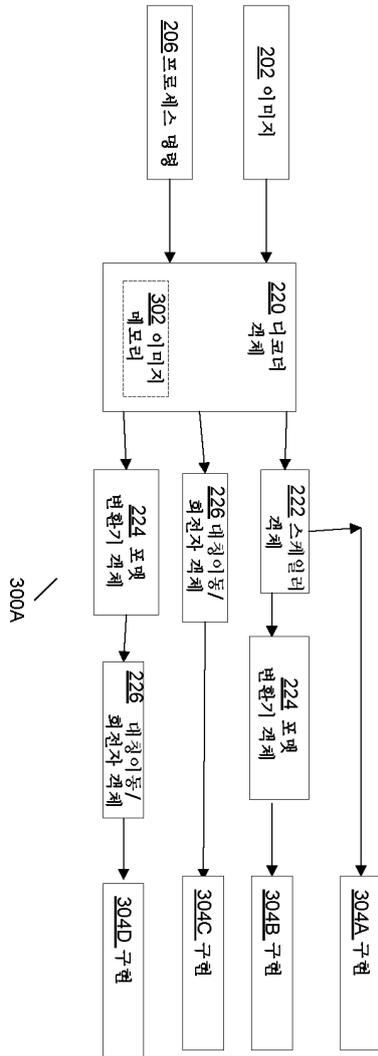
도면1



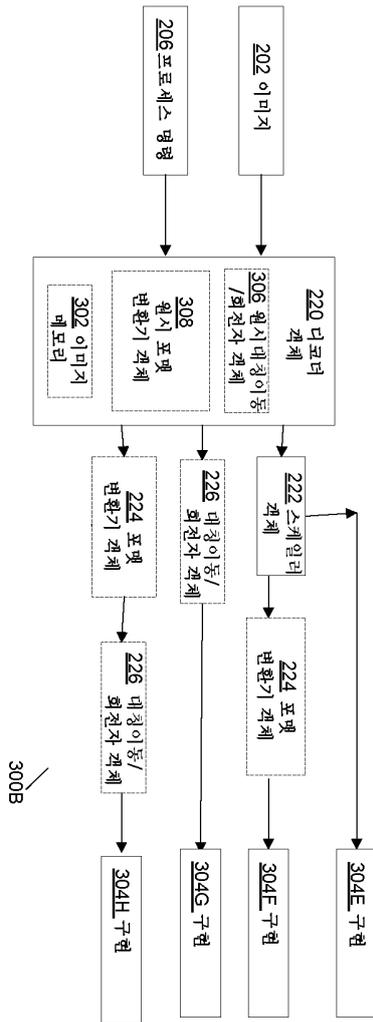
도면2



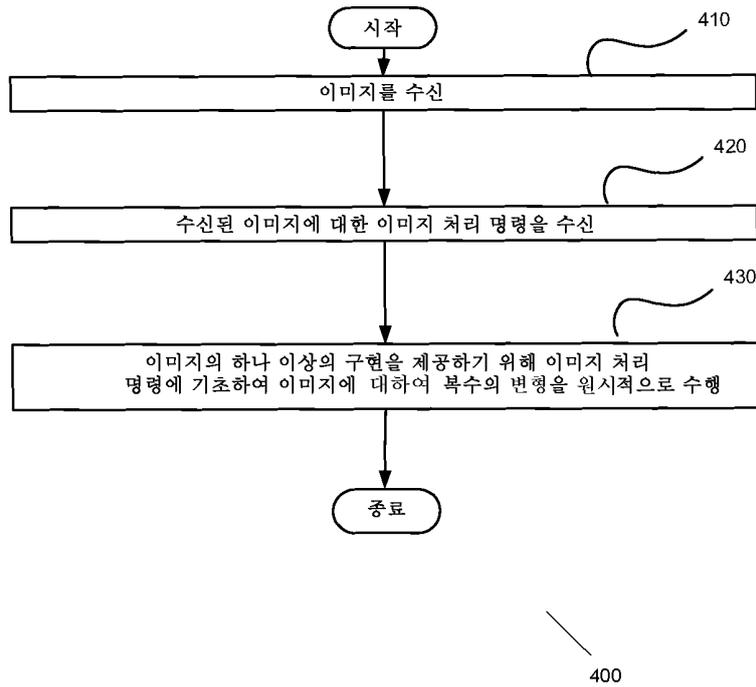
도면3A



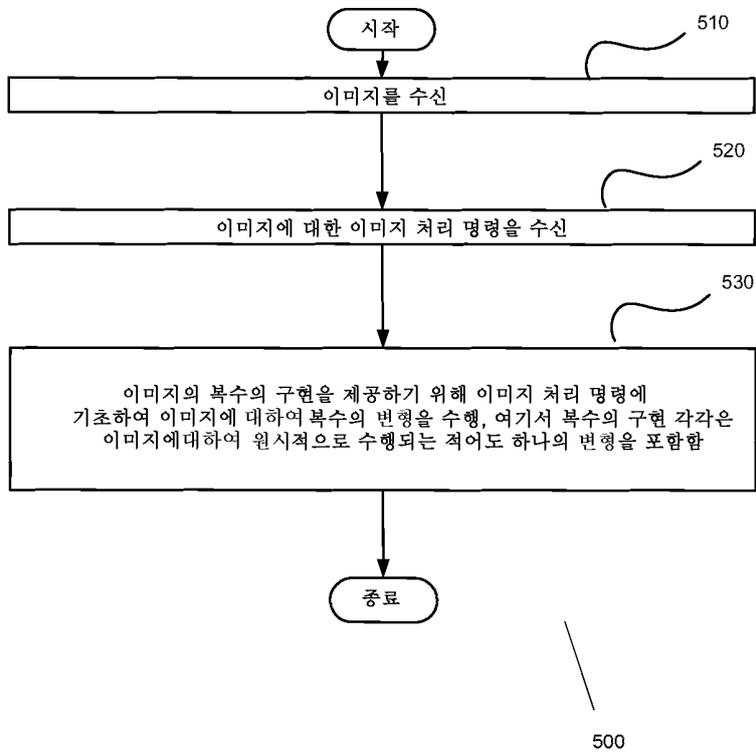
도면3B



도면4



도면5



도면6

