



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2016년08월08일

(11) 등록번호 10-1646608

(24) 등록일자 2016년08월02일

(51) 국제특허분류(Int. Cl.)

G06T 1/00 (2006.01) G06K 9/36 (2006.01)

H04N 5/217 (2016.01)

(21) 출원번호 10-2013-7003318

(22) 출원일자(국제) 2011년07월07일

심사청구일자 2015년04월27일

(85) 번역문제출일자 2013년02월07일

(65) 공개번호 10-2013-0098298

(43) 공개일자 2013년09월04일

(86) 국제출원번호 PCT/IB2011/001942

(87) 국제공개번호 WO 2012/004672

국제공개일자 2012년01월12일

(30) 우선권주장

12/959,281 2010년12월02일 미국(US)

61/362,247 2010년07월07일 미국(US)

(56) 선행기술조사문헌

이상훈 외 3명, 한국정보기술학회논문지 제7호
제1권 (2009.02.28)*

JP2002503855 A*

*는 심사관에 의하여 인용된 문헌

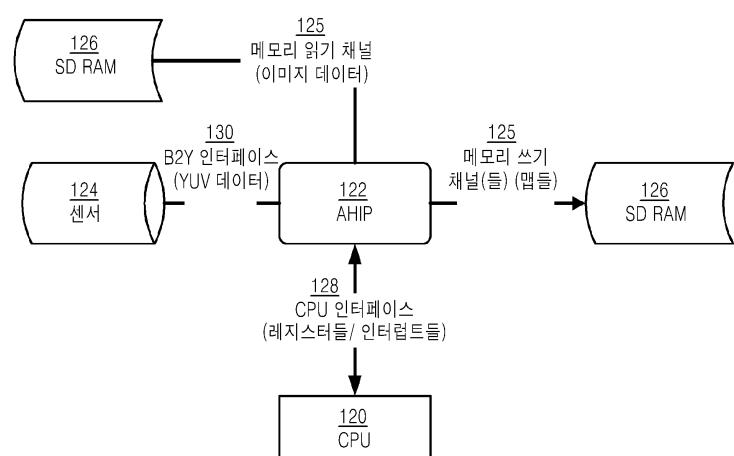
전체 청구항 수 : 총 20 항

심사관 : 김용권

(54) 발명의 명칭 실시간 비디오 프레임 선-프로세싱 하드웨어

(57) 요 약

동적으로 재설정 가능한 이종의 시스톨릭 (heterogeneous systolic) 어레이로서, 제1 이미지 프레임을 프로세싱하고, 그리고 상기 이미지 프레임으로부터 이미지 프로세싱 프리메이티브 (primitive)들을 생성하며, 그리고 상기 프리메이티브들 및 상기 대응 이미지 프레임을 메모리 저장부에 저장하도록 구성된다. 이미지 프레임의 특성이 결정된다. 그 특성을 기반으로 하여, 상기 어레이에는 다음의 이미지 프레임을 프로세싱하도록 재설정 가능하다.

대 표 도 - 도6

명세서

청구범위

청구항 1

하나 이상의 하드웨어 프로세서들, 메모리, 및 하나 이상의 모듈들을 포함하는 장치로서:

동적으로 재설정 가능한 이종의 시스톨릭 (heterogeneous systolic) 어레이에서, 제1 이미지 프레임을 프로세싱하여 적어도 하나의 이미지 맵을 포함하는 복수의 이미지 프로세싱 프리미티브 (primitive)들을 상기 제1 이미지 프레임으로부터 생성하며;

상기 복수의 이미지 프로세싱 프리미티브들 및 상기 제1 이미지 프레임을 메모리 저장부에 저장하며;

상기 복수의 이미지 프로세싱 프리미티브들을 적어도 부분적으로 기반으로 하여, 상기 제1 이미지 프레임의 하나 이상의 특성들을 결정하며;

상기 제1 이미지 프레임에 대해서 결정된 상기 하나 이상의 특성들을 기반으로 하여, 상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 재설정하며; 그리고

상기 재설정된 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 이용하여, 상기 하나 이상의 특성들을 기반으로 하여 제2 이미지 프레임을 프로세싱하도록 구성된, 장치.

청구항 2

제1항에 있어서,

상기 복수의 이미지 프로세싱 프리미티브들 중의 이미지 프로세싱 프리미티브가, 상기 제1 이미지 프레임의 서브 구역을 식별하고 그 서브 구역으로부터 상기 이미지 프로세싱 프리미티브를 생성함에 의해서 생성되는, 장치.

청구항 3

제1항에 있어서,

상기 제1 이미지 프레임 및 제2 이미지 프레임은 비디오 프레임들인, 장치.

청구항 4

제1항에 있어서,

상기 복수의 이미지 프로세싱 프리미티브들은 프로세싱된 이미지 맵들 또는 다른 프로세싱된 하드웨어 모듈 데이터, 또는 그 두 가지 모두를 포함하는, 장치.

청구항 5

제1항에 있어서,

상기 장치는 포맷된 이미지 데이터를 메모리로 제공하도록 구성된, 장치.

청구항 6

제4항에 있어서,

상기 프로세싱된 이미지 맵들은 문턱값 맵들 또는 인테그랄 (integral) 이미지 맵들, 또는 두 가지 모두를 포함하는, 장치.

청구항 7

제1항에 있어서,

상기 복수의 이미지 프로세싱 프리미티브들은 구역적인 프리미티브들 또는 프레임 데이터 프리미티브들, 또는

두 가지 모두를 포함하는, 장치.

청구항 8

제1항에 있어서,

상기 장치는 포맷된 이미지 데이터를 메모리로부터 수신하도록 더 구성된, 장치.

청구항 9

제1항에 있어서, 상기 장치는,

프로세싱된 이미지 데이터를 수신하도록 구성되며 그리고 압축된 이미지 또는 비디오 데이터, 또는 두 가지 모두를 메모리에 제공하거나 메모리로부터 수신하도록 구성된 트랜스코드 모듈을 더 포함하는, 장치.

청구항 10

제9항에 있어서,

상기 트랜스코드 모듈은 이미지 데이터 또는 비디오 데이터, 또는 두 가지 모두의 선택된 서브셋트들을 디스플레이로 출력하도록 더 구성된, 장치.

청구항 11

제1항에 있어서,

상기 장치는 단일의 클록 사이클 내에 이미지 데이터를 프로세싱하도록 구성된, 장치.

청구항 12

제1항에 있어서,

상기 장치는 하드웨어 모듈 또는 IPP (image processing pipeline) 모듈, 또는 두 가지 모두를 프로그램하도록 더 구성된, 장치.

청구항 13

이미지 획득 및 프로세싱 기기로서,

프로세서;

디지털 이미지 프레임들을 획득하기 위한 렌즈 및 이미지 센서; 및

동적으로 재설정 가능한 이종의 시스톨릭 어레이를 포함하며,

상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이는:

상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이에서, 제1 이미지 프레임을 프로세싱하여 복수의 이미지 프로세싱 프리미티브들을 상기 제1 이미지 프레임으로부터 생성하도록 구성되며; 상기 복수의 이미지 프로세싱 프리미티브들 중의 이미지 프로세싱 프리미티브가, 상기 제1 이미지 프레임의 서브 구역을 식별하고 그 서브 구역으로부터 상기 이미지 프로세싱 프리미티브를 생성함에 의해서 생성되며; 상기 복수의 이미지 프로세싱 프리미티브들은 적어도 하나의 이미지 맵을 포함하며;

상기 복수의 이미지 프로세싱 프리미티브들 및 상기 제1 이미지 프레임을 메모리 저장부에 저장하도록 구성되며;

상기 복수의 이미지 프로세싱 프리미티브들을 적어도 부분적으로 기반으로 하여, 상기 제1 이미지 프레임의 하나 이상의 특성들을 결정하도록 구성되며;

상기 제1 이미지 프레임에 대해서 결정된 상기 하나 이상의 특성들을 기반으로 하여, 상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 재설정하도록 구성되며; 그리고

상기 재설정된 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 이용하여, 상기 하나 이상의 특성들을 기반으로 하여 제2 이미지 프레임을 프로세싱하도록 구성된, 이미지 획득 및 프로세싱 기기.

청구항 14

제13항에 있어서,

상기 복수의 이미지 프로세싱 프리미티브들 중의 이미지 프로세싱 프리미티브는, 상기 제1 이미지 프레임의 서브 구역을 식별하고 그 서브 구역으로부터 상기 이미지 프로세싱 프리미티브를 생성함에 의해서 생성되는, 이미지 획득 및 프로세싱 기기.

청구항 15

제14항에 있어서,

상기 제1 이미지 프레임 및 제2 이미지 프레임은 비디오 프레임들인, 이미지 획득 및 프로세싱 기기.

청구항 16

동적으로 재설정 가능한 이종의 시스톨릭 (heterogeneous systolic) 어레이에서, 제1 이미지 프레임을 프로세싱 하여 적어도 하나의 이미지 맵을 포함하는 복수의 이미지 프로세싱 프리미티브 (primitive)들을 상기 제1 이미지 프레임으로부터 생성하고;

상기 복수의 이미지 프로세싱 프리미티브들 및 상기 제1 이미지 프레임을 메모리 저장부에 저장하며;

상기 복수의 이미지 프로세싱 프리미티브들을 적어도 부분적으로 기반으로 하여, 상기 제1 이미지 프레임의 하나 이상의 특성들을 결정하며;

상기 제1 이미지 프레임에 대해서 결정된 상기 하나 이상의 특성들을 기반으로 하여, 상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 재설정하며; 그리고

상기 재설정된 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 이용하여, 상기 하나 이상의 특성들을 기반으로 하여 제2 이미지 프레임을 프로세싱하는 것을 포함하는 방법으로,

상기 방법은 하나 이상의 컴퓨팅 기기를 이용하여 수행되는, 방법.

청구항 17

제16항에 있어서,

상기 복수의 이미지 프로세싱 프리미티브들 중의 이미지 프로세싱 프리미티브가, 상기 제1 이미지 프레임의 서브 구역을 식별하고 그 서브 구역으로부터 상기 이미지 프로세싱 프리미티브를 생성함에 의해서 생성되는, 방법.

청구항 18

제16항에 있어서,

상기 제1 이미지 프레임 및 제2 이미지 프레임은 비디오 프레임들인, 방법.

청구항 19

제16항에 있어서,

하드웨어 기반의 이미지 프로세싱 모듈을 프로그래밍하는 것을 더 포함하는, 방법.

청구항 20

하나 이상의 명령어들을 저장하는 비-일시적 컴퓨터-판독가능 저장 매체로서,

상기 하나 이상의 명령어들은 하나 이상의 프로세서들에 의해서 실행될 때에, 상기 하나 이상의 프로세서들로 하여금 제16항 내지 제19항 중 어느 한 항에 따른 방법을 수행하도록 하는, 비-일시적 컴퓨터-판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 우선권

[0002] 본원은 2010.12.02.에 출원된 미국 특허 출원 번호 12/959,281의 출원 그리고 2010.07.07.에 출원된 미국 임시 특허 출원 번호 61/362,247에 대한 우선권을 주장한다.

배경 기술

[0003] 애프터 이미지들은 디지털 이미지 시스템 내 이미지 센서들에 의해 획득되며, 그 이미지들은 디스플레이 또는 기기 상에서 저장되기 이전에 프로세싱되는 것이 보통이다. 전형적인 이미지 프로세싱 체인 또는 이미지 프로세싱 파이프라인 (image processing pipeline), 또는 IPP는 도 1에 도시된다. 도 1에 도시된 예시의 IPP는 노출 및 화이트 밸런스 모듈 (2), 디모자이크 블록 (4), 컬러 교정 블록 (6), 감마 교정 블록 (8), 컬러 변환 블록 (10) 그리고 다운샘플링 모듈 (12)을 포함한다.

[0004] 실시간 비디오 이미징 시스템들 구현할 것을 필요로 할 때에, 그런 IPP에 대한 커다란 강제가 종종 존재하며, 이는 이미지 데이터는 상기 IPP의 각 단계에서 메모리로부터 읽혀지며 그리고 몇몇의 오퍼레이션들 이후에 반대로 써지는 것이 보통이기 때문이다. HD 비디오에 대해, 메모리 대역폭은 커다란 도전들을 겪는다. 그래서, 비디오 획득 기기들에서의 하드웨어 실시예들에서 IPP의 엘리먼트들을 직접적으로 구현하는 것이 필요하다. 이것은 상기 IPP의 엘리먼트들이 프로세싱의 각 단계 이후에 메모리에 이미지 데이터를 기록하고 그리고 각 이어지는 IPP 오퍼레이션에 대한 데이터를 거꾸로 읽어내는 경쟁을 피하게 하는 이점을 가질 것이다. 그러나, 상기 IPP의 각 단계에 적용되는 방법이 덜 적응적일 수 있다는 것을 의미하며, 이는 단일의 이미지 프레임으로부터의 데이터를 입력하기 이전에 전체 IPP 체인이 설정될 것이기 때문이다.

[0005] 현대의 디지털 스틸 카메라 (digital still cameras (DSC))들은 도 1에서 몇몇의 예시 블록들로 도시된 것과 같은 기본적인 IPP에 의해서 제공될 수 있는 것보다 더욱 복잡한 이미지 및 장면 (scene) 분석을 실행한다. 특히, 이미지 획득 기기들은 이미지 장면 내에서 얼굴 구역들을 탐지하고 추적할 수 있으며 (미국 특허들 7,620,218, 7,460,695, 7,403,643, 7,466,866 및 7,315,631, 그리고 미국 특허 공개 2009/0263022, 2010/0026833, 2008/0013798, 2009/0080713, 2009/0196466 및 2009/0303342 그리고 미국 출원 번호 12/374,040 및 12/572,930 참조, 이 특허들, 공개 출원 및 특허 출원들은 본원과 동일한 양수인에게 양수되었으며 그리고 본원에 참조로서 편입된다) 그리고 이 기기들은 그런 구역들 내 오점들과 결함들을 분석하고 탐지하고 그리고 그런 흠집들을 즉시로 교정할 수 있다 (미국 특허 7,565,030 그리고 미국 특허 번호 2009/0179998 출원들 참조, 이것들은 모두 동일한 양수인에게 양수되었으며 그리고 본원에 참조로서 편입된다). 먼저 오점들 또는 "픽시들 (pixels)"과 같은 전체적인 결함들은 탐지되고 그리고 교정될 수 있다 (예를 들면, 미국 특허 출원 일련번호 12/710,271 및 12/558,227의 출원들 그리고 미국 특허들 7,206,461, 7,702,236, 7,295,233 및 7,551,800 참조, 이것들 모두는 동일한 양수인에게 양수되었으며, 본원에 참조로서 편입된다). 안면 향상 (Facial enhancement)이 적용될 수 있다. 이미지 블러 그리고 병진 및 회전하는 이미지 모션이 판별되고 보상될 수 있다 (예를 들면, 미국 특허 7,660,478 그리고 미국 특허 공개 2009/0303343, 2007/0296833, 2008/0309769, 2008/0231713 및 2007/0269108 그리고 WO/2008/131438의 출원들 참조, 이것들 모두는 본원에 참조로서 편입된다). 얼굴 구역들은 식별될 수 있고 그리고 알려진 사람들과 연관될 수 있다 (예를 들면, 미국 특허 7,567,068, 7,515,740 및 7,715,597 그리고 US2010/0066822, US2008/0219517 및 US2009/0238419 그리고 미국 일련번호 12/437,464 출원들 참조, 이것들 모두는 본원에 참조로서 편입된다). 이런 기술들 그리고 다른 기술들 (예를 들면, 미국 특허들 6,407,777, 7,587,085, 7,599,577, 7,469,071, 7,336,821, 7,606,417 및 2009/0273685, 2007/0201725, 2008/0292193, 2008/0175481, 2008/0309770, 2009/0167893, 2009/0080796, 2009/0189998, 2009/0189997, 2009/0185753, 2009/0244296, 2009/0190803, 2009/0179999 그리고 미국 출원 일련번호 12/636,647 참조, 이것들은 동일한 양수인에게 양수되었으며 그리고 본원에 참조로서 편입된다) 모두는 이미지 장면 분석에 의존한다. 전형적으로, 이는 메모리 저장부로부터 이미지 데이터의 블록들을 읽는 단계 그리고 이 데이터에 대한 이어지는 다양한 프로세싱 단계들을 포함한다. 각 장면 분석 알고리즘을 용이하게 하기 위해서 중간의 데이터 구조들이 상기 이미지 저장부 내에 임시로 저장될 수 있을 것이다. 몇몇의 경우들에서, 이런 데이터들은 단일의 알고리즘에 특유한 것이며, 반면에 다른 경우들에서, 데이터 구조들은 여러 상이한 장면 분석 알고리즘들에 걸쳐서 지속될 수 있을 것이다. 이런 경우들에서, 다양한 이미지 프로세싱 오퍼레이션들을 수행하기 위해서 이미지 데이터는 이미지 저장 메모리와 CPU 사이에서 이동된다. 다중의 알고리즘들이 적용되는 경우에, 이미지 데이터는 각

이미지 상의 상이한 이미지 및 장면 프로세싱 오퍼레이션들을 수행하기 위해서 여러 차례 읽혀지는 것이 보통이다.

[0006] 상기 기술들의 대부분에 대해서, 분석은 대부분의 디지털 카메라들에 의해서 캡쳐되고 그리고 그 카메라 디스플레이 상에 실시간 디스플레이를 제공하기 위해서 사용된 상대적으로 낮은 해상도의 스트림인 미리 보기 이미지 스트림을 포함할 수 있을 것이다. 그래서, 메인 이미지 장면을 알맞게 분석하기 위해서, 이용 가능한 실질적으로 동일한 장면의 적어도 두 개의 이미지들을 구비하는 것이 유용하다. 하나 또는 그 이상의 미리 보기 이미지들이 또한 저장되는 경우에, 이것들은 메인으로 획득된 (풀 해상도) 이미지와 결합하여 여러 기회들에서 또한 읽혀지는 것이 보통이다. 추가로, 다양한 장면 분석 알고리즘들을 용이하게 하기 위해서 프로세싱은 미리 보기 이미지들의 업샘플링된 사본들 그리고 메인으로 획득된 이미지들의 다운샘플링된 사본들을 일시적으로 저장하는 것을 포함할 수 있을 것이다.

[0007] 디지털 카메라 내에서, 이미지들은 보통은 개별적으로 획득되는 것이 보통이며 그리고 보통은 1초 또는 그 이상의 순서인 실질적인 실시간 간격은 장면 분석을 위한 이미지 획득들 및 개별 이미지들에 대한 포스트 프로세싱 사이에 이용 가능하다. 예를 들면, 프로페셔널 DSC의 버스트 모드에서 다중의 이미지들이 시간적으로 아주 근접하게 획득되는 경우에조차, 제한된 메모리로 인해서 한정된 수의 이미지들이 획득된다. 더욱이, 이 이미지들은 버스트 획득 동안에는 프로세싱될 수 없지만, 그러나 더욱 복잡한 장면-기반의 프로세싱이 실행될 수 있기 이전에 그것이 완료될 때까지 종종 대기한다.

[0008] 현재의 비디오 설비 내에서, 데이터는 30 fps 또는 그 이상의 프레임 레이트들로 종종 프로세싱되며, 그리고 메모리 강체들로 인해서, 상기 데이터는 디지털 방식으로 압축되며 그리고 거의 즉각적으로 장기 (long-term) 메모리 저장부에 쓰여진다. 또한, 낮은-해상도 미리 보기 스트림은 DSC의 경우에서처럼 일반적으로 이용 가능한 것은 아니다. 결국, 풀-HD 비디오 스트림을 처리하기 위한 요구 사항들은 그런 설비 내에서는 메모리 대역폭이 도전 대상이라는 것을 암시한다.

[0009] HD 비디오 획득 기기를 위해서 DSC 내에서 현재 이용 가능한 것과 같은 현대의 장면 분석 기술들의 이점들을 얻기 위해서, 우리는 여러 주요한 도전들을 식별할 수 있다. 먼저, 비디오 프레임 획득들 사이에 이용 가능한 시간 내에 풀 HD에 대한 복잡한 장면 분석을 저장하고 수행하는 것이 어렵다. 이는 단순하게 CPU 과워의 문제가 아니라, 아마도 데이터 대역폭이라는 더욱 중요한 문제이다. 풀 HD 이미지들의 크기는 그런 이미지들을 IPP를 통해서 비디오 압축 유닛의 장기간 저장부로 단순하게 이동시키는 것이 큰 도전이라는 것을 의미한다. 상기 IPP에 부가된 하드웨어를 통해서 몇몇의 제한 장면 분석이 가능할 수 있을 것이지만, 이는 상기 비디오 스트림의 실-시간 획득을 시작하기 이전에 고정된 많은 세팅들 및 설정들을 포함할 것이며, 그래서 그 장면 분석들은 진행되는 장면 분석에 동적으로 적응 가능하지 않을 수 있을 것이며 그리고 응답하지 않을 수 있을 것이다.

[0010] 두 번째로, 상기 IPP에 아주 큰 공유 메모리 버퍼들을 도입하지 않고는 장면 분석 알고리즘들 사이에 이미지 프로세싱 데이터 프리메이티브들 (primitives)을 공유하기 위한 어떤 기회도 존재하지 않는다. 이는 단일의 IC 내에 도 2에 도시된 혼란 기술을 무모하게 그리고 효율적으로 흉내내는 하드웨어 설계 요구 사항들로 이끌 수 있을 것이다. 도 2는 IPP 및 소프트웨어로 된 다른 높은 레벨의 기능들을 구현하기 위한 통상적인 하드웨어를 도시한다. 메모리 (14)는 이미지와 데이터 캐시 (16) 그리고 장기 (long term) 데이터 저장부 (18)를 포함하는 것으로 보여진다. 상기 캐시 (16)는 로 (raw) 데이터 (20), RGB 포맷된 데이터 (22) 및 RGB 프로세싱된 데이터 (24)를 저장할 수 있으며, 반면에 상기 장기 데이터 저장부 (18)는 MPEG 이미지들 (26) 및/또는 JPEG 이미지들 (28)을 유지할 수 있을 것이다. 센서 (32)는 로 데이터를 상기 메모리 (14)로 그리고 상기 IPP (34)로 전달한다. 또한 상기 IPP (34)는 상기 메모리 (14)로부터 데이터를 수신한다. 상기 IPP (34)는 상기 메모리 (14, 16)로 RGB 데이터 (22)를 제공한다. 프로세싱된 RGB 데이터 (24)를 상기 메모리 (14)에 제공하고 그리고 RGB 데이터 (22)를 트랜스코드 모듈 (38)로 제공하는 CPU (36)에 의해서 RGB 데이터 (22, 24)는 또한 인출된다. 상기 트랜스코드 모듈 (38)은 상기 메모리 (14, 18)로 데이터를 제공하고 그리고 그 메모리로부터 데이터를 인출한다. 상기 트랜스코드 모듈은, 예를 들면, LCD/TFT 디스플레이 (40) 상에 보여질 데이터를 또한 제공한다.

[0011] 다양한 실제적인 이유들로 인해서, 이는 최적화된 이미지 프로세싱 메커니즘을 제공하지 않는다. 대안으로는 각 장면 분석 알고리즘을 위해 개별적인 하드웨어 구현을 구비하는 것이지만, 그러나 이는 각 알고리즘이 완전한 장면 분석을 수행하기 위해서 풀 이미지 프레임 버퍼를 사용해야 할 것이기 때문에 매우 큰 하드웨어 크기로 또한 이끌 것이다.

[0012] 이런 넓은 영역들 각각 내에 많은 추가적인 엔지니어링 난점들이 존재하지만, 넓은 범위의 도전을 식별하는 것이 가능하며, 이 경우에 현재의 장면 분석 기수들 및 그 결과의 이미지 향상의 이점들은 현재의 기술들을 이용

하는 실시간 비디오에 느끼기 쉽게 적용되지 않는다. 그러므로 실시예들의 유리한 세트가 아래에서 제공된다.

선행기술문헌

특허문현

[0013]

- (특허문현 0001) 미국 특허 7,620,218
- (특허문현 0002) 미국 특허 7,460,695
- (특허문현 0003) 미국 특허 7,403,643
- (특허문현 0004) 미국 특허 7,466,866
- (특허문현 0005) 미국 특허 7,315,631
- (특허문현 0006) 미국 특허 공개 2009/0263022
- (특허문현 0007) 미국 특허 공개 2010/0026833
- (특허문현 0008) 미국 특허 공개 2008/0013798
- (특허문현 0009) 미국 특허 공개 2009/0080713
- (특허문현 0010) 미국 특허 공개 2009/0196466
- (특허문현 0011) 미국 특허 공개 2009/0303342
- (특허문현 0012) 미국 출원 번호 12/374,040
- (특허문현 0013) 미국 특허 출원 12/572,930
- (특허문현 0014) 미국 특허 7,565,030
- (특허문현 0015) 미국 특허 공개 2009/0179998
- (특허문현 0016) 미국 특허 출원 12/710,271
- (특허문현 0017) 미국 특허 출원 12/558,227
- (특허문현 0018) 미국 특허 7,206,461
- (특허문현 0019) 미국 특허 7,702,236
- (특허문현 0020) 미국 특허 7,295,233
- (특허문현 0021) 미국 특허 7,551,800
- (특허문현 0022) 미국 특허 7,660,478
- (특허문현 0023) 미국 특허 공개 2009/0303343
- (특허문현 0024) 미국 특허 공개 2007/0296833
- (특허문현 0025) 미국 특허 공개 2008/0309769
- (특허문현 0026) 미국 특허 공개 2008/0231713
- (특허문현 0027) 미국 특허 공개 2007/0269108
- (특허문현 0028) WO/2008/131438
- (특허문현 0029) 미국 특허 7,567,068
- (특허문현 0030) 미국 특허 7,515,740
- (특허문현 0031) 미국 특허 7,715,597
- (특허문현 0032) 미국 특허 공개 US2010/0066822

- (특허문현 0033) 미국 특허 공개 US2008/0219517
(특허문현 0034) 미국 특허 공개 US2009/0238419
(특허문현 0035) 미국 특허 출원 12/437,464
(특허문현 0036) 미국 특허 6,407,777
(특허문현 0037) 미국 특허 7,587,085
(특허문현 0038) 미국 특허 7,599,577
(특허문현 0039) 미국 특허 7,469,071
(특허문현 0040) 미국 특허 7,336,821
(특허문현 0041) 미국 특허 7,606,417
(특허문현 0042) 미국 특허 공개 2009/0273685
(특허문현 0043) 미국 특허 공개 2007/0201725
(특허문현 0044) 미국 특허 공개 2008/0292193
(특허문현 0045) 미국 특허 공개 2008/0175481
(특허문현 0046) 미국 특허 공개 2008/0309770
(특허문현 0047) 미국 특허 공개 2009/0167893
(특허문현 0048) 미국 특허 공개 2009/0080796
(특허문현 0049) 미국 특허 공개 2009/0189998
(특허문현 0050) 미국 특허 공개 2009/0189997
(특허문현 0051) 미국 특허 공개 2009/0185753
(특허문현 0052) 미국 특허 공개 2009/0244296
(특허문현 0053) 미국 특허 공개 2009/0190803
(특허문현 0054) 미국 특허 공개 2009/0179999
(특허문현 0055) 미국 출원 12/636,647
(특허문현 0056) 미국 특허 6,873,743
(특허문현 0057) 미국 특허 7,315,631
(특허문현 0058) 미국 특허 7,466,866
(특허문현 0059) 미국 특허 공개 US2009/0303342
(특허문현 0060) 미국 특허 공개 US2009/0263022
(특허문현 0061) 미국 특허 7,460,695
(특허문현 0062) 미국 특허 7,403,643
(특허문현 0063) 미국 특허 7,315,631
(특허문현 0064) 미국 특허 공개 US2009/00123063
(특허문현 0065) 미국 특허 공개 US2009/0190803
(특허문현 0066) 미국 특허 공개 US2009/0189998
(특허문현 0067) 미국 특허 공개 US2009/0179999
(특허문현 0068) 미국 특허 출원 12/572,930

- (특허문헌 0069) 미국 특허 출원 12/824,214
- (특허문헌 0070) 미국 특허 출원 12/815,396
- (특허문헌 0071) 미국 특허 출원 12/790,594
- (특허문헌 0072) 미국 특허 출원 12/825,280
- (특허문헌 0073) 미국 특허 7,315,631
- (특허문헌 0074) 미국 특허 공개 US 2002/0102024
- (특허문헌 0075) 미국 특허 공개 2010/0053368

발명의 내용

해결하려는 과제

[0014] 본 발명은 상기에서의 문제점들을 해결할 수 있는 실시간 비디오 프레임 선-프로세싱을 제공한다.

과제의 해결 수단

[0015] 폴 이미지 프레임을 획득하는 것에 연관된 것보다 두드러지게 더 작은 지역을 가지면서 굑센 단위로 시리얼로 입력되는 이미지 프레임으로부터 실시간으로 유도된 이미지 프로세싱 프리메이티브들 (primitives)의 범위를 생성하는 하드웨어 서브-시스템을 포함하는 실시예들이 아래에서 설명된다. 이 프리메이티브들은 매우 일찍 또는 이미지 프레임이 획득된 거의 직후에 조차 이용 가능하며 그리고 다음 이미지 프레임이 획득되고 있을 때에 이 이미지 프레임을 추가로 프로세싱하기 위해서 사용될 수 있다.

[0016] 추가로, 이전 이미지 프레임을 시퀀스로 프로세싱하는 것으로부터 결정된 데이터는 이전 이미지 프레임으로부터 유도된 이미지 프로세싱 프리메이티브들과 결합하기 위해서 이용 가능하게 만들어질 수 있다. (비록 그런 것이 특정 실시예들에서는 옵션으로 활용될 수 있을 것이지만) 이는 이미지들의 낮은 해상도의 미리 보기 스트림을 개별적으로 캡처하지 않고서도 상세한 프레임 단위의 장면 프로세싱을 가능하게 한다.

[0017] 실시예들은 클럭 사이클 입력 당 하나의 굑센을 사용하여 동작하고 그리고/또는 현재 이미지/비디오 프레임에 관해서 유용한 지식을 제공하는 이미지 프로세싱 프리메이티브들의 여러 상이한 유형들을 생성하는 것으로 또한 설명된다. 각 프리메이티브들은 하나 또는 그 이상의 굑센 프로세싱 블록들 또는 모듈들을 포함하는 프로세싱 체인에 대해서 생성된다. 이것들은 동적으로 스위치될 수 있을 복수의 내부 데이터 버스들에 대해서 서로 링크된다. 특정의 덜 복잡한 실시예들에서, 비록 상기 프라이머리 입력 모듈들이 상기 이미지 센서/IPP로부터의 공통 입력 데이터 경로를 공유할 수 있을 것이지만, 모듈들은 데이터 경로들과 직접적으로 링크될 수 있을 것이다.

[0018] 다중 모듈들은 동일한 입력 데이터를 공유할 수 있을 것이다. 또한, 개별 프로세싱 블록들의 출력은 논리적으로 결합될 수 있을 것이다. 다중의 프로세싱 체인들로부터의 개별 출력들은 외부 메모리 (SDRAM)로 출력되기 이전에 단일의 데이터 워드로 결합되는 것이 보통이며, 이는 메모리 및 외부 메모리 버스들의 최적 이용을 용이하게 한다. 프로세싱 체인들 사이의 프로세싱 시간에 있어서의 차이들 때문에, 상기 출력 데이터의 올바른 정렬을 보장하기 위해서 동기화 모듈은 논리적 회로와 통합된다.

[0019] 상기 생성된 이미지 프리메이티브들은, 더욱 적용 가능한 많은 이미지 프로세싱 기술들 중에서, 적목 검출, 얼굴 검출 및 식별, 얼굴 미화, 프레임-대-프레임 이미지 등록 (registration), 그리고 파노라마 이미지들을 생성하기 위한 다중-프레임 이미지 결합을 포함하는 이미지 프로세싱 오퍼레이션들의 범위의 성능을 가속하기 위해서 유리하게 사용될 수 있다. 또한, 이 프리메이티브들의 가용성은 장면 분석 및 프로세싱 알고리즘들의 범위의 구현을 크게 단순화시킨다. 이는 특히 CPU 또는 GPU 상에서 연속된 프로세싱을 위해서 메모리 저장부로부터 풀 이미지 프레임을 읽고 그리고 쓰는 경향을 유리하게 줄일 수 있다. 대부분의 경우들에, 특정 알고리즘을 이용하여 이미지를 분석하고 그리고/또는 향상시키기 위해서, 관련된 이미지 프리메이티브(들) 그리고 메인 이미지는 일단 읽혀지기만 한다. 다중의 알고리즘들을 단일의 이미지 읽기 상에서 실행시키기 위해서, 상기 다중의 알고리즘들로부터의 프리메이티브들을 상기 메인으로 획득된 이미지의 단일 읽기와 함께 로드하는 것이 또한 가능하다. 이는 비디오 스트림을 프로세싱하기 위해서 활용되는 메모리 대역폭을 크게 줄인다. 별도의 읽기/쓰기 버스들이 이용 가능한 경우에, 하나의 이미지 프레임을 메인 CPU/GPU 상에서 프로세싱하면서, 두 번째 이미지 프

레이임들이 획득되고 있으며 그리고 IPP 및 AHIP 모듈들에 의해서 선-프로세싱 (pre-processing)되는 것이 또한 가능하다.

[0020] 또한, 이 시스템 설정은 다음의 이미지 프레임의 선-프로세싱을 적응시키기 위해, 상기 CPU/GPU에 의해 프로세싱되고 있는 이미지 프레임의 분석으로부터 유도된 데이터가 상기 IPP 또는 상기 AHIP 모듈로 피드백되는 것을 가능하게 한다. 상기 IPP에 의해서 적용된 글로벌 이미지 프로세싱 그리고 상기 AHIP에 의해서 적용된 장면-특유 이미지 프로세싱의 두 가지 모두의 상세한 적용은 비디오 획득 설비의 더욱 빠르고 더욱 반응에 민감한 성능을 가능하게 한다. 이는 조명 상태들이, 예를 들면, 얼굴 구역들의 분석 및 연관된 피부의 컬러 맵들을 기반으로 하여 변하고 있는 경우인 상황들에서 비디오 획득에 대한 더욱 빠른 적용을, 차례로, 허용한다. 특정 실시예들에 따라서 그런 기술들은 비디오 획득에 유리하게 적용 가능하다.

[0021] 이런 면에서, 특정 실시예들에서 프레임 카운터 및 연관된 로직이 또한 활용될 수 있을 것이다. 각 프레임 프로세싱 사이클의 끝 부분에서, 내부적인 픽셀 프로세싱 체인들을 재설정하는 것이 가능하다. 이는 새로운 LTU들을 로딩하고, 개별 픽셀 프로세싱 블록들의 프로세싱 파라미터들을 변경하고 또는 몇몇의 경우들에서 프로세싱 체인 내 블록들의 논리적인 결합의 순서를 재설정하는 것을 포함할 수 있을 것이다. 특정 실시예들에서, 모듈들은 선택되거나 또는 우회되는 것 중의 어느 하나이다. 더욱 복잡한 실시예들에서, 데이터 프로세싱 모듈들은 하나 또는 그 이상의 내부 데이터-버스들 상에서 I/O 포트를 공유한다. 특정 실시예들에서, 모듈로의/모듈로부터의 인접 동시 읽기/쓰기 오퍼레이션들을 가능하게 하기 위해서 이중-버퍼 I/O가 채택될 수 있을 것이다.

[0022] 특정 실시예들에 따라, 동적으로 재설정 가능한 이종의 시스톨릭 (heterogeneous systolic) 어레이는 제1 이미지 프레임을 프로세싱하고, 그리고 그 이미지 프레임으로부터 이미지 프로세싱 프리메이티브들을 생성하고, 그리고 그 프리메이티브들 및 상기 대응 이미지 프레임을 메모리 저장부에 저장하도록 설정된다. 상기 이미지 프레임의 결정된 특성을 기반으로 하여, 상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이는 다음의 이미지 프레임을 프로세싱하도록 재설정 가능하다.

[0023] 상기 이미지 프레임의 상기 특성을 결정하는 것은 상기 시스톨릭 어레이의 외부에서 수행될 수 있을 것이다. 상기 적어도 하나의 특성을 결정하는 것은 상기 이미지 프레임의 적어도 일부를 그리고 상기 이미지 프로세싱 프리메이티브들 중 적어도 하나를 기반으로 할 수 있을 것이다.

[0024] 특정 실시예들에 따른 이미지 획득 및 프로세싱 기기는 프로세서, 렌즈 및 디지털 이미지 프레임들을 획득하기 위한 이미지 센서, 그리고 여기에서의 실시예들 중 어느 하나에 따라서 설명된 것과 같은 설정된 상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이들 중 어느 하나를 포함한다.

[0025] 특정 실시예들에 따른 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 이용하는 이미지 프로세싱 방법은 제1 이미지 프레임을 획득하고 프로세싱하는 단계를 포함한다. 이미지 프로세싱 프리메이티브들은 상기 이미지 프레임으로부터 생성된다. 상기 프리메이티브들 및 상기 대응 이미지 프레임은 메모리 저장부에 저장된다. 상기 이미지 프레임의 적어도 하나의 특성이 결정되며, 그리고 그 적어도 하나의 특성을 기반으로 하여, 상기 방법은 다음의 이미지 프레임을 프로세싱하기 위해서 상기 어레이를 재설정하는 단계를 포함한다.

[0026] 상기 적어도 하나의 특성을 결정하는 것은 상기 시스톨릭 어레이의 외부에서 수행될 수 있을 것이다. 상기 적어도 하나의 특성을 결정하는 것은 상기 이미지 프레임의 적어도 일부 및 상기 이미지 프로세싱 프리메이티브들 중 적어도 하나에서의 프로세싱을 기반으로 할 수 있을 것이다.

[0027] 추가의 디지털 이미지 획득 및 프로세싱 기기가 특정 실시예들에 따라서 제공된다. 상기 기기는 렌즈 및 디지털 이미지들을 획득하기 위한 이미지 센서를 포함한다. 상기 기기의 이미지 프로세싱 파이프라인 (IPP) 모듈은 이미지 프레임의 로 (raw) 이미지 데이터를 수신하고 그리고 대응하는 포맷된 이미지 데이터를 메모리 저장부 및 이미지 프로세싱을 수행하도록 구성된 하드웨어 모듈로 분배하도록 구성된다. 상기 하드웨어 모듈은 상기 IPP 모듈로부터 상기 이미지 프레임의 상기 포맷된 이미지 데이터를 수신하고 그리고 장면 프로세싱 프리메이티브들을 상기 메모리로 제공한다. 프로세싱 유닛은 상기 메모리 저장부로부터 수신한 그리고 상기 어레이가 제공한 상기 장면 프로세싱 프리메이티브들을 기반으로 하여 다음의 이미지 프레임을 프로세싱하도록 구성된다.

[0028] 상기 장면 프로세싱 프리메이티브들은 프로세싱된 이미지 맵들 또는 다른 프로세싱된 하드웨어 모듈 데이터, 또는 두 가지 모두를 포함할 수 있을 것이다. 상기 하드웨어 모듈은 포맷된 이미지 데이터를 상기 메모리로 제공하도록 또한 구성될 수 있을 것이다. 상기 프로세싱된 이미지 맵들은 문턱값 지도들 또는 인테그랄 (integral) 이미지 맵들 또는 두 가지 모두를 포함할 수 있을 것이다. 상기 장면 프로세싱 프리메이티브들은 구역적인 프리메이티브들이나 또는 프레임 데이터 프리메이티브들 또는 둘 모두를 포함할 수 있을 것이다. 상기 프로세싱 유

낮은 상기 메모리로부터 포맷된 이미지 데이터를 또한 수신할 수 있을 것이다.

- [0029] 상기 기기는 프로세싱 유닛으로부터 프로세싱된 이미지 데이터를 수신하고 그리고 상기 메모리로부터 압축된 이미지 데이터 또는 비디오 데이터, 또는 두 가지 모두를 제공하거나 또는 수신하도록 구성된 트랜스코드 모듈을 포함할 수 있을 것이다. 상기 트랜스코드 모듈은 이미지 데이터 또는 비디오 데이터, 또는 두 가지 모두의 선택된 서브셋트들을 디스플레이로 출력하도록 구성될 수 있을 것이다.
- [0030] 상기 하드웨어 모듈은 단일의 클록 사이클 내에 이미지 데이터를 프로세싱하도록 구성될 수 있을 것이다. 상기 프로세싱 유닛은 상기 하드웨어 모듈 또는 상기 IPP 모듈, 또는 둘 모두를 프로그램하도록 더 구성될 수 있을 것이다.
- [0031] 디지털 이미지들을 획득하고 그리고 프로세싱하는 다른 방법이 특정 실시예들에 따라서 제공되며, 이는 디지털 이미지들을 획득하고, 그리고 이미지 프레임의 로 (raw) 이미지 데이터를 수신하고 그리고 대응하는 포맷된 이미지 데이터를 메모리 및 하드웨어-기반의 이미지 프로세싱 모듈로 분배하는 단계를 포함한다. 상기 방법은 상기 이미지 프레임의 포맷된 이미지 데이터를 상기 하드웨어-기반의 이미지 프로세싱 모듈에서 수신하고 그리고 장면 프로세싱 프리메이티브들을 메모리로 제공하며, 그리고 상기 하드웨어-기반의 이미지 프로세싱 모듈에 의해서 제공된 상기 장면 프로세싱 프리메이티브들을 상기 메모리에 저장하는 단계를 포함한다. 다음의 이미지 프레임은 상기 메모리로부터 수신한 그리고 상기 하드웨어-기반의 이미지 프로세싱 모듈에 의해서 제공된 상기 장면 프리메이티브들을 기반으로 프로세싱된다.
- [0032] 상기 장면 프로세싱 프리메이티브들은 프로세싱된 이미지 맵들 또는 다른 프로세싱된 하드웨어 모듈 데이터, 또는 두 가지 모두를 포함할 수 있을 것이다. 포맷된 이미지 데이터는 상기 하드웨어-기반의 이미지 프로세싱 모듈로부터 상기 메모리로 제공될 수 있을 것이다. 문턱값 지도들 또는 인테그랄 이미지 맵들 또는 두 가지 모두를 포함하는 이미지 맵들이 프로세싱될 수 있을 것이다. 상기 장면 프로세싱 프리메이티브들은 구역적인 프리메이티브들 (regional primitives)이나 또는 프레임 데이터 프리메이티브들 또는 둘 모두를 포함할 수 있을 것이다. 포맷된 이미지 데이터는 상기 메모리로부터 수신될 수 있을 것이다. 프로세싱된 이미지 데이터는 트랜스코드 모듈로 전송될 수 있을 것이며 그리고 압축된 이미지 또는 비디오 데이터 또는 그것들의 결합이 상기 트랜스코드 모듈로 전달되거나 또는 상기 트랜스코드 모듈로부터 전달될 수 있을 것이다. 상기 이미지 데이터 또는 비디오 데이터, 또는 둘 모두의 선택된 서브셋트들을 디스플레이로 출력될 수 있을 것이다. 이미지 데이터는 상기 하드웨어-기반의 이미지 데이터 프로세싱 모듈을 이용하여 단일의 클럭 사이클 내에서 프로세싱될 수 있을 것이다. 상기 하드웨어-기반의 이미지 프로세싱 모듈은 프로그램 가능하고 그리고/또는 재설정 가능할 수 있을 것이다.
- [0033] 하드웨어-기반의 이미지 프로세싱 모듈이 이미지 획득 및 프로세싱 기기 내에서 동작하도록 제공되며, 상기 기기는 이미지 데이터를 획득하고 그리고 대응하는 포맷된 이미지 데이터를 상기 하드웨어-기반의 이미지 프로세싱 모듈로 분배하며, 상기 하드웨어-기반의 이미지 프로세싱 모듈은, 메모리로부터 수신된 그리고 상기 하드웨어-기반의 이미지 프로세싱 모듈에 의해서 제공된 장면 프로세싱 프리메이티브들을 기반으로 하여 다음의 이미지 프레임들을 프로세싱하기 위해 메모리에 저장될 포맷된 이미지 데이터를 기반으로 하여 장면 프로세싱 프리메이티브들을 생성하고 상기 메모리로 제공하도록 구성된다.
- [0034] 상기 장면 프로세싱 프리메이티브들은 프로세싱된 이미지 맵들 또는 다른 프로세싱된 하드웨어 모듈 데이터, 또는 두 가지 모두를 포함할 수 있을 것이다. 상기 모듈은 포맷된 이미지 데이터를 상기 메모리로 제공하고 그리고/또는 문턱값 지도들 또는 인테그랄 이미지 맵들 또는 두 가지 모두를 포함하는 이미지 맵들을 프로세싱하도록 구성될 수 있을 것이다. 상기 장면 프로세싱 프리메이티브들은 구역적인 프리메이티브들 (regional primitives)이나 또는 프레임 데이터 프리메이티브들 또는 둘 모두를 포함할 수 있을 것이다. 상기 모듈은 포맷된 이미지 데이터를 상기 메모리로부터 수신하고 그리고 프로세싱하며 그리고/또는 프로세싱된 이미지 데이터를 트랜스코드 모듈로 전송하고 그리고 압축된 이미지 또는 비디오 데이터를 상기 트랜스코드 모듈로 또는 상기 트랜스코드 모듈로부터 또는 결합하여 전달할 수 있을 것이다. 상기 모듈은 이미지 데이터 또는 비디오 데이터, 또는 두 가지 모두의 선택된 서브셋트들을 디스플레이로 출력하고 그리고/또는 상기 하드웨어-기반의 이미지 프로세싱 모듈을 이용하여 단일의 클록 사이클 내에서 이미지 데이터를 프로세싱하도록 또한 구성될 수 있을 것이다. 상기 모듈은 프로그램 가능하고 그리고/또는 재설정 가능할 수 있을 것이다.
- [0035] 동적으로 재설정 가능한 이종의 시스톨릭 어레이에는 특정 실시예들에 따라 제1 이미지 프레임을 프로세싱하고; 상기 이미지 프레임으로부터 복수의 이미지 프로세싱 프리메이티브 (primitive)들을 생성하며; 상기 이미지 프레임의 적어도 하나의 서브구역을 식별하고 그리고 상기 서브구역으로부터 적어도 하나의 이미지 프리메이티브

를 생성하고; 그리고 상기 프리메이티브들을 이미리 프레임 및 서브구역 위치와 함께 메모리 저장부에 저장하도록 구성된다. 상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이는 상기 이미지 프레임 또는 서브구역 또는 둘 모두의 하나 또는 그 이상의 결정된 특성들을 기반으로 하여 다음의 이미지 프레임을 프로세싱하도록 재설정 가능하다.

[0036] 상기 적어도 하나의 특성을 결정하는 것은 상기 시스톨릭 어레이의 외부에서 수행될 수 있을 것이다. 상기 적어도 하나의 특성을 결정하는 것은 상기 이미지 프레임의 적어도 일부에서의 그리고 상기 이미지 프로세싱 프리메이티브들 중 적어도 하나에서의 프로세싱을 기반으로 할 수 있을 것이다.

[0037] 특정 실시예들에 따른 이미지 획득 및 프로세싱 기기는, 프로세서; 디지털 이미지 프레임들을 획득하기 위한 렌즈 및 이미지 센서; 그리고 여기에서 설명된 실시예들 중 어느 하나에 따라서 구성된 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 포함한다.

[0038] 동적으로 재설정 가능한 이종의 시스톨릭 어레이를 이용한 이미지 프로세싱 방법이 제공되며, 이 방법은, 제1 이미지 프레임을 획득하고 프로세싱하는 단계; 상기 이미지 프레임으로부터 복수의 이미지 프로세싱 프리메이티브들을 생성하는 단계; 상기 이미지 프레임의 적어도 하나의 서브구역을 식별하고 그리고 상기 서브구역으로부터 적어도 하나의 이미지 프리메이티브를 생성하는 단계; 상기 프리메이티브들을 이미리 프레임 및 서브구역 위치와 함께 메모리 저장부에 저장하고, 상기 이미지 프레임 또는 서브구역, 또는 둘 모두의 적어도 하나의 특성을 결정하는 단계; 및 상기 적어도 하나의 특성을 기반으로 하여, 상기 어레이를 재설정하여 다음의 이미지 프레임을 프로세싱하도록 하는 단계를 포함한다.

[0039] 상기 적어도 하나의 특성을 결정하는 것은 상기 동적으로 재설정 가능한 이종의 시스톨릭 어레이의 외부에서 수행될 수 있을 것이다. 상기 적어도 하나의 특성을 결정하는 것은 상기 이미지 프레임의 적어도 일부 및 상기 이미지 프로세싱 프리메이티브들 중 적어도 하나에서의 프로세싱을 기반으로 할 수 있을 것이다.

발명의 효과

[0040] 본 발명의 효과는 본 명세서의 해당되는 부분들에 개별적으로 명시되어 있다.

도면의 간단한 설명

[0041] 도 1은 통상적인 이미지 프로세싱 파이프라인 (image processing pipeline (IPP))을 도시한다.

도 2는 IPP를 구현하고 그리고 다른 하이 레벨 기능들을 소프트웨어로 구현하기 위한 통상적인 하드웨어를 도시한다.

도 3은 이종의 시스톨릭 어레이 구조를 도시한다.

도 4는 특정 실시예들에 따른, 패스-스루 특성을 포함하는, 이미지 프로세싱을 위한 진보된 하드웨어 (advanced hardware for image processing), 또는 AHIP를 갖춘 IPP 하드웨어를 도시한다.

도 5는 본 발명의 특정 실시예들에 따른, 다양한 이미지 프로세싱 프리메이티브들을 위한 프로세싱 체인들에 배치된 몇몇의 일반적인 프로세싱 모듈을 포함하는 AHIP 모듈을 도시한다.

도 6은 본 발명의 특정 실시예들에 따른, 메인 CPU, 하드웨어 모듈, 이미지 센서 및 SDRAM 읽기/쓰기 채널들 사이의 상호-관계를 도시한다.

도 7은 본 발명의 특정 실시예들에 따른, 이미지를 포함하는 메모리 저장부와 AHIP 모듈을 갖춘 데이터 캐시 사이의 상호 관계를 도시한다.

도 8은 본 발명의 특정 실시예들에 따른 컬러 문턱값 모듈을 도시한다.

도 9는 본 발명의 특정 실시예들에 따른, 원래의 이미지 및 16개 문턱값들 (4 비트)을 가진 스킨-컬러 맵을 도시한다.

도 10은 컬러 공간 변환, 컬러 문턱값, 프레임-대-프레임 레지스트레이션 (registration), 인테그럴 (integral) 이미지 및/또는 인터크렐 제곱 이미지 프리메이티브들 그리고/또는 이미지 프레임 히스토그램 데이터를 출력하는 특정 실시예들에 따른 AHIP 모듈을 도시한다.

도 11은 본 발명의 특정 실시예들에 따른, AHIP 모듈 그리고 다른 이미지 프로세싱 컴포넌트들 사이의 관계들을

도시한다.

발명을 실시하기 위한 구체적인 내용

[0042] 시스톨릭 어레이들 (SYSTOLIC ARRAYS)

[0043] 시스톨릭 어레이 전형, 즉, 데이터 카운터들에 의한 데이터-스트림-구동 (data-stream-driven)은 폰 노이만 전형, 즉, 프로그램 카운터에 의한 명령-스트림 -구동 (instruction-stream-driven)의 대응부 (counterpart)이다. 시스톨릭 어레이는 다중의 데이터 스트림들을 보통 송신하고 그리고 수신하기 때문에, 그리고 이 데이터 스트림들을 생성하기 위해서 다중의 데이터 카운터들이 사용되기 때문에, 그것은 데이터 병행을 지원한다. 그 이름은 심장에 의한 규칙적인 혈액 펌프를 이용한 유사로부터 비롯된 것이다.

[0044] 시스톨릭 어레이는 셀들로 불리는 데이터 프로세싱 유닛들의 행렬-유사한 행들로 구성된다. 데이터 프로세싱 유닛들, 또는 PDU들은 프로그램 카운터를 제외하면 중앙 프로세싱 유닛 (CPU)들과 유사하며, 이는 오퍼레이션이, 즉, 데이터 객체의 도착에 의해서 전송-유발되기 (transport-triggered) 때문이다. 각 셀은 프로세싱 직후에 자신의 이웃들과 정보를 공유한다. 상기 시스톨릭 어레이는 종종 직사각형이며 또는 그렇지 않다면 열들 및/또는 행들로 배치된 자신의 셀들을 구비하며, 이 경우 데이터는 이웃 DPU들 사이에서 상기 어레이를 지나서, 때로는 상이한 방향들에서 흐르는 상이한 데이터로, 흐른다. 도 3은 균일한 시스톨릭 어레이 구조의 그런 예를 도시한다. 상기 어레이의 포트들에 진입하고 떠나는 데이터 스트림들은 자동-시퀀스 메모리 유닛들 (auto-sequencing memory units), 또는 ASM들에 의해서 생성된다. 각 ASM은 데이터 카운터를 포함한다. 임베디드 시스템들에서, 또한 데이터 스트림은 외부 소스로부터의 입력 그리고/또는 외부 소스로의 출력일 수 있을 것이다.

[0045] 시스톨릭 어레이들은 메시 (mesh)-유사한 위상 (topology)에서 가장 가까운 작은 개수의 이웃 DPU들에 연결된다. DPU들은 자신들 사이에서 흐르는 데이터 상에서 일련의 시퀀스들을 수행한다. 전통적인 시스톨릭 어레이 합성 방법들이 대수학적인 알고리즘들에 의해서 실행되기 때문에, 선형 파이프들만을 구비한 균일한 어레이들만이 획득될 수 있으며, 그래서 상기 구조는 모든 DPU들에서 동일하다. 결과는 규칙적인 데이터 의존성을 가진 애플리케이션들만이 클래식한 시스톨릭 어레이들 상에서 일반적으로 구현된다는 것이다.

[0046] SIMD (single instruction/multiple data) 머신들과 비슷하게, 클록을 공급받는 시스톨릭 어레이는 각 프로세서가 대안의 계산/통신 위상들을 떠맡는 "락-스텝 (lock-step)"으로 계산한다.

[0047] 그러나, DPE들 사이에서 비동기식 핸드세이킹을 하는 시스톨릭 어레이들은 종종 웨이브프론트 어레이들로 불린다. 한가지 잘-알려진 시스톨릭 어레이는 Carnegie Mellon 대학의 iWarp 프로세서로, 이는 인텔에 의해서 제조되었다. iWarp 시스템은 양 방향으로 진행하는 데이터 버스들에 의해서 연결된 선형 어레이 프로세서를 구비한다.

[0048] AHIP (Advanced Hardware for Image Processing)

[0049] 도 4는 AHIP (advanced hardware for image processing)를 구비한 IPP 하드웨어를 포함하는 일 실시예를 개략적으로 도시한다. 도 4에서 도시된 AHIP는 패스-스루 (pass-through) 속성을 가진다. 도 4는 이미지 및 데이터 캐시 (46) 그리고 장기 (long-term) 데이터 저장부 (48)를 포함하는 메모리 저장부 (44)를 보여준다. 상기 캐시 (46)는 로 (raw) 데이터 (50), RGB 데이터 (52) 및 프로세싱된 RGB 데이터 (54)를 포함하며, 그리고 상기 장기 데이터 저장부는 MPEG 이미지들 (56) 그리고/또는 JPEG 이미지들 (58)을 포함할 수 있을 것이다. 도 4에서 도시된 실시예는, 예를 들면, 구역적인 프리메이티브들 (62), 프레임 데이터 프리메이티브들 (64), 문턱값 맵들 (66) 그리고/또는 인터그럴 이미지 맵들 (68)을 포함하는 장면 프로세싱 프리메이티브들 (670)을 또한 유리하게 도 포함한다.

[0050] 도 4는 로 데이터를 상기 메모리 (44) 및 IPP (74)로 전달하는 센서 (72)를 보여준다. 상기 IPP (74)는 상기 메모리 (44)로부터 로 데이터를 또한 수신한다. 상기 IPP(74)는 RGB 데이터 (52)를 상기 메모리 (44, 46)로 제공한다. RGB 데이터는 상기 IPP (74)에 의해서 유리한 AHIP 모듈 (75)로 제공된다. 상기 AHIP 모듈 (75)은 프로세싱된 이미지 맵들, AHIP 모듈 데이터 그리고 RGB 데이터를 상기 메모리 (44, 46)로 제공한다. 상기 메모리 (44, 46)는 RGB 데이터, 이미지 맵들 그리고 AHIP 모듈 데이터를 상기 CPU/GPU (76)로 제공한다. 상기 CPU (76)는

프로세싱된 RGB 데이터 (54)를 상기 메모리 (44)로 그리고 트랜스코드 모듈 (78)로 제공한다. 상기 CPU (76)는 도 4에서 개략적으로 도시된 것처럼, 상기 IPP 모듈 (74) 및/또는 상기 AHIP 모듈 (75)을 또한 프로그램할 수 있다. 상기 트랜스코드 모듈 (78)은 상기 메모리 (44, 48)로 데이터를 제공하고 그리고 그 메모리로부터 데이터를 인출한다. 상기 트랜스코드 모듈 (78)은, 예를 들면, LCD/TFT 디스플레이 (80) 상에서 보여주기 위해서 데이터를 또한 제공한다.

[0051] 유리하게도, 특정 실시예들에서, 클럭 사이클 당 하나의 표준 이미지 픽셀이 취해지며 그리고 이 픽셀은 다양한 방식들 중의 하나 또는 그 이상의 방식으로 프로세싱된다. 출력의 여러 상이한 유형들은 각 개별 픽셀의 프로세싱으로부터 별도로 생성될 수 있을 것이다. 각 출력 유형의 하나 이상의 예는 하드웨어 엘리먼트들의 복제에 의해서 제공될 수 있다 이 하드웨어 서브-시스템이 매 클럭 사이클 상에서 하나의 픽셀을 프로세싱할 수 있기 때문에, 그것은 상기 센서로부터 이미지 픽셀들을 전송하는 것을 자연시키지 않으며 그래서 그것은 상기 IPP의 어떤 단계에라도 삽입될 수 있다.

[0052] 여러 일반적인 유형의 이미지 프로세싱 프리메이티브들은 상기 AHIP 모듈에 의해서 식별될 수 있고 생성된다. 다음의 설명을 명확하게 하기 위해서, 이미지 데이터는 "픽셀들" (픽쳐 엘리먼트들)로서 언급될 수 있을 것이며 그리고 출력 프리메이티브들 내의 데이터 값들은 "맵-픽셀들 (map-pixels)"로서 언급될 수 있을 것이다. 전형적으로 맵-픽셀은 한 픽셀 (24 또는 32 비트)보다 아주 더 작을 것이다. 예로서, 스킨 맵들을 위해서 사용된 맵-픽셀의 한가지 모습은 스킨 픽셀인 원래의 픽셀의 4개 확률들에 대응한 2-비트만을 구비한다. 다른 맵-픽셀은 미리 정의된 컬러 공간에서 그것이 특별한 컬러에 얼마나 유사한가를 기술하는 16개 문턱값 (threshold)들에 대응하는 4 비트를 구비한다. 이 16개 레벨의 유사성에 대응하는 컬러-공간 문턱값들은 맵-픽셀들을 포함하는 최종 출력 데이터 프리메이티브 맵들과 함께 LUT 내에 저장된다.

[0053] 첫 번째의 그런 프리메이티브는 직접적인 픽셀 대 맵-픽셀 매핑을 포함한다. 특정 실시예들에서, 이것은 컬러 또는 휘도 문턱값을 포함할 수 있을 것이며, 이는 특별한 픽셀이 컬러 공간에서 미리 정해진 값에 얼마나 가깝게 있는가를 결정한다. 특정 실시예들에서, 이 데이터는 15개 문턱값들의 범위로서 캡쳐될 수 있을 것이며 그리고 4-비트 이미지 맵으로 써질 수 있을 것이다. 이 문턱값들은 상기 CPU/GPU에 의해서 이미지 프레임으로부터 이미지 프레임으로 조절될 수 있다.

[0054] 예시의 실시예에서, 각 문턱값의 데이터 값들은 이미지 픽셀들이 스킨 컬러에 얼마나 가까운가를 측정하기 위해서 세팅된다. 그런 이미지 맵은 얼굴 구역의 상이한 스킨 영역들을 구별하기 위해서 유리하게도 사용될 수 있으며 그리고 얼굴 추적 및 얼굴 미화 (facial beautification)와 같은 애플리케이션들을 위해서 유용할 수 있다.

[0055] 이미지 프리메이티브의 이 모습은 클럭 사이클들의 면에서 작은 고정된 지연을 일으킬 뿐이다. 출력 이미지 맵은 이미지 프레임들의 마지막 픽셀이 상기 AHIP로 입력된 이후에 수십 개의 클럭 사이클들 내에 이용 가능한 것이 보통이다.

[0056] 픽셀 대 맵-픽셀 프로세싱에 관한 한가지 변형은 다중의 픽셀들이 프로세싱될 때에, 단일의 출력 픽셀을 생성하는 것이다. 이는 로 (RAW) 입력 이미지의 서브샘플링에 대응한다. 몇몇의 실시예들에서, 픽셀들의 블록은 대응하는 출력 프리메이티브 데이터를 생성하기 위해서 프로세싱되기 이전에 하드웨어 라인 버퍼들에 임시로 저장된다. 대안의 실시예들에서, 픽셀들은 개별적으로 계속해서 프로세싱되지만, 각 개별 픽셀을 프로세싱하는 것으로부터 출력들은 단일의 맵-픽셀을 생성하기 위해서 몇몇의 미리 정해진 방식으로 결합된다.

[0057] 프리메이티브의 두 번째 모습은 커널 유도된 프리메이티브이다. 그런 프리메이티브들에 대한 맵-픽셀들은 그 이미지의 현재의 이미지 픽셀 그리고 적어도 하나의 이전 픽셀에 대한 지식으로부터 유도된다. 많은 기술들이 NxN 커널들을 규정하며, 이는 현재 픽셀에 대응하는 출력 값이 그 이미지 내 수평 및 수직 방향 둘 모두에서 N 개의 인접한 픽셀들로부터 결정된다는 것을 의미한다. 특정 실시예들에서 픽셀 데이터가 상기 AHIP 모듈에만 이용 가능하기 때문에, 그런 실시예들에서 이미지 센서로부터 이미지 데이터의 IPP 전체 행들 (Full rows)을 통해서 행 단위로 (그리고/또는 열 단위로) 직접적으로 클록이 공급될 것이며, 커널 유도된 프리메이티브들을 지원하기 위해서 이 실시예들에서는 이미지 데이터의 IPP 전체 행들은 보통은 버퍼링될 것이다.

[0058] 특정 실시예들에서, 이미지 데이터의 7개 행들이 그 전체적으로 저장되고 그리고 8번째 이미지 행은 나가떨어진다. 이는 상기 모듈이 8x8 까지의 커널로부터 유도된 이미지 프로세싱 프리메이티브들을 생성하는 것을 가능하게 한다. 이 실시예에서, 현재 이미지 프레임에 대해 폴 커널 프리메이티브 이미지 맵이 이용 가능하기 이전에 8 곱하기 상기 이미지의 행 크기 (1080p에 대해 8 x 1920)의 차수의 지연이 존재한다. 그럼에도 불구하고 이것은 폴 이미지 프레임 (1000개 픽셀 행들)을 획득하기 위해서 취해진 전체 시간의 1%보다 여전히 더 작으며, 이

미지 프레임 프리메이티브 데이터는 최종 프레임 획득이 완료된 이후에 아주 짧게 이용 가능하다.

[0059] 커널 유도된 이미지 프로세싱 프리메이티브의 한 가지 특별한 예는 적목 세그먼테이션 (red-eye segmentation)이다. 예를 들면, 본원에 참조로 편입되는 미국 특허 6,873,743에서, 이미지 상에 2x2 세그먼테이션을 수행하기 위한 기술이 설명된다. 이는 LAB 컬러 공간상에서 동작한다. 커널 유도된 프리메이티브의 다른 예는 이미지의 휘도 성분상에서 수행되는 인테그랄 (integral) 이미지의 계산이다. 간략하게 설명될 것처럼, 특정 실시예에서 상기 AHIP는, 입력 픽셀 데이터를 여러 공통적으로 사용된 컬러 공간들 사이에서 온-더-플라이 (on-the-fly) 변환을 가능하게 하는 컬러-공간 변환 모듈을 통합한다. 그래서 개별적인 RGB 픽셀 데이터는 상기 AHIP 내에서 무시할 수 있는 지연을 가지고 YCC 또는 LAB 컬러 공간으로 변환될 수 있다.

[0060] 프리메이티브의 세 번째 모습은 프레임 유도된 프리메이티브들을 포함한다. 이것들은 데이터 프리메이티브들의 예들이며, 이 경우에 단일의 픽셀 또는 픽셀들의 블록은 대응 단일 맵-픽셀 출력을 생성하지 않는다. 이미지 프로세싱 프리메이티브의 이런 모습의 일 예는, 여러 빈들 (bins)로 미리 설정된 히스토그램 모듈이다. 입력 픽셀 데이터는 문턱값들의 세트에 대해서 분석되며 그리고 그것의 값을 기반으로 하여 히스토그램 빈으로 분류된다. 이미지 프레임의 끝 부분에서, 각 히스토그램 빈은 자신의 상단 및 하단 문턱값 한계들을 만족시키는 픽셀들 개수의 카운트를 포함한다.

[0061] 프리메이티브 유형 하나를 위해서 주어진 예와 결합될 때에, 16개 스킨-컬러 히스토그램 빈들의 세트 내 특별한 이미지 프레임 내에 얼마나 많은 픽셀들이 있는지를 측정하는 것이 가능하다.

[0062] 이는, 예를 들면, 추적된 얼굴 구역 내에서 너무 많은, 또는 너무 작은 스킨 픽셀들이 탐지된다면, 스킨 컬러 문턱값들이 다음 이미지 프레임에 대해서 조절될 것을 필요로 한다는 것을 암시할 수 있을 것이다. 상기 AHIP 내 하드웨어 구조는, 이미지 프레임을 프로세싱하기 이전에 프로세싱 블록들이 동적으로 재설정되는 것을 가능하게 하도록 설계된다. 본 발명의 특정 실시예들에 따라 추가적인 병렬 프로세싱 블록들이 그런 하드웨어 구조에 추가될 수 있다.

[0063] 다른 실시예들에서, 색상 상관도표 (color correlogram) 또는 히스토그램-유사한 데이터의 다른 모습들이 상기 AHIP에 의해서 결정될 수 있을 것이다. 그런 프리메이티브들은 커널 데이터를 생성하기 위해서 사용된 행 베틀들의 동일한 세트로부터 유리하게도 결정될 수 있을 것이며, 그 차이는 히스토그램 또는 상관도표 데이터는, 입력을 커널 유도된 프리메이티브들에 의해서 제공된 출력 픽셀들로 1-대-1 매핑하는 것이 아니라, 다중의 픽셀들로부터 결정된 프레임 유도된 프리메이티브들을 제공한다는 것이다.

[0064] 프레임 유도된 프리메이티브의 다른 모습은 픽셀 행 및 픽셀 열 값들의 더하기를 수행하는 것이다. 이는 현재 이미지 프레임의 하나 또는 그 이상의 이전 이미지 프레임들과의 상관을 가능하게 한다. 그런 프리메이티브들은 AHIP의 다른 모습을 도입하며, 이 경우 현재의 프레임으로부터 결정된 프리메이티브들의 하나 또는 그 이상은 하나 또는 그 이상의 연속 이미지 프레임들에 대해서 기억될 수 있을 것이다. 그런 프리메이티브는 전체 이미지 프레임보다 훨씬 더 작을 수 있을 것이며 또는 실시간 프로세싱의 유리한 점들은 완전하게 파악되지 않을 것이다. 특정 실시예들에서, 그런 프리메이티브들에 대한 전형적인 크기 한계는 그것들이 프로세싱된 이미지의 최대 행 크기보다 더 크지 않도록 한다.

[0065] 특정 실시예들에서, 그런 데이터는 외부 메모리 저장부로 써지는 것이 아니라 상기 AHIP 내에서 유지될 수 있을 것이다. 그래서 이미지 프레임 유도된 데이터 및/또는 픽셀 유도된 데이터는 이미지 프레임 프리메이티브들의 더 빠른 프로세싱을 용이하게 하기 위해서 상기 AHIP 내에서 축적될 수 있을 것이다.

[0066] 프리메이티브의 네 번째 모습은 상기 메인 이미지 프레임의 특유의 공간적인 구역으로부터 유도된다. 이 프리메이티브들은 속성상 더욱 복잡할 수 있을 것이며 그리고 구역의 더욱 복잡한 하드웨어 프로세싱을 몇몇의 기준 프리메이티브들 그리고 상기 CPU/GPU로부터 유도된 외부 데이터와 결합할 수 있을 것이며 그리고 하나 또는 그 이상의 앞선 이미지 프레임들과 관련시킨다.

[0067] 디지털 이미지들 내에서 얼굴 구역들의 위치를 예측하는 (미국 특허 7,315,631 및 그것의 자손, 그리고 7,466,866 참조, 이것들은 상기에서 참조로 편입되었다) 것을 포함하여 프로세싱하는데 있어서 하드웨어 베틀들이 사용될 수 있을 것이다. 특정 실시예들에서, 현재의 이미지 프레임 내에서 얼굴이 탐지될 것으로 기대되는 곳에 관하여 하나 또는 그 이상의 이전 프레임들에서 지식이 수집된다. 이런 접근 방식은 현재 프레임에서 얼굴 탐지를 수행하는 것보다 더욱 빠르다는 유리함을 가지며, 그리고 상기 모든 정보는 현재 이미지 프레임 이전에라도 다양한 목적들을 위해서 사용될 수 있을 것이다.

[0068] 특히, 이미지의 첫 번째 프로세싱 동안에 그런 구역의 고도로 정밀한 위치를 결정하는 것은 일반적으로 매우 어

려우며, 이는 그것이 상기 CPU/GPU 상에서 소프트웨어로 수행될 추가적인 이미지 프로세싱에 의존하기 때문이다. 결과적으로, 이미지 프레임에 대한 첫 번째 하드웨어 프로세싱 동안에 상기 AHIP에 의해서 공간적인 구역이 어디인가만을 대략적으로 결정하는 것이 일반적이다. 그러나, 이 대략적인 위치들은 유리하게 마킹될 수 있으며 그리고 상기 메인 이미지보다는 훨씬 더 작은 것이 보통이다. 일 실시예에서, 그렇게 예측된 여러 구역들은 다음의 프레임 사이클 상에서 추가로 프로세싱하기 위해서 상기 AHIP의 버퍼들 내에 저장될 수 있을 것이다. 대안의 실시예에서, 이것들은 메인 이미지와 함께 메모리에 쓰여지지만, 그런 구역들을 특별하게 프로세싱하도록 설정된 두 번째 AHIP 모듈을 통해서 다시 로딩된다. 이 두 번째 실시예에서, 메모리 서브시스템이 듀얼-포트라는 사실에서 유리함이 있다. 그래서, 다음의 이미지 프레임이 주된 AHIP에 의해서 프로세싱되고 그리고 메모리로 써지고 있을 때에, 이전의 이미지 프레임으로부터의 하나 또는 그 이상의 예측된 구역들이 더욱 특화된 프로세싱을 위해서 상기 두 번째 AHIP 모듈로 거꾸로 읽혀질 수 있을 것이다. 이 실시예에서, 다음의 이미지 프레임이 일반적으로 프로세싱되고 있을 때에만 특별한 이미지 구역들이 보통은 프로세싱될 것이다. 그럼에도 불구하고, 단일의 프레임 지연은 쉽게 보상될 수 있으며 그리고 비디오 시퀀스의 실시간 프로세싱에 근접하게 달성하는 목표를 손상시키지 않는다.

[0069] 하나의 매우 공통적인 공간적 구역은 예측된 얼굴 구역이다. 이는, 그 내부에 얼굴 구역이 위치할 것이라는 확률이 아주 높은 현재 이미지 프레임의 구역이다. 그런 구역들은 얼굴 추적 알고리즘들에서 빈번하게 사용된다 (상기에서 참조로서 편입된 미국 특허 7,315,631 및 그것의 자손 특허를 다시 참조). 그런 구역들의 한 가지 공통적인 이용은 적목 탐지와 같은 이미지 프로세싱 알고리즘의 적용을 얼굴이 있을 높은 확률이 존재하는 곳인 이미지 구역으로 한정하는 것이다.

[0070] 도 5는 다양한 이미지 프로세싱 프리메이티브들을 위한 프로세싱 체인들로 배치된 여러 가지의 일반적인 프로세싱 모듈들을 구비한 AHIP 모듈을 개략적으로 도시한다. 이미지 센서 (82), SDRAM (84), AHIP 모듈 (85) 그 자체, 그리고 CPU (86)가 도 5에서 보인다. 상기 AHIP 모듈은 상기 CPU (86)와 통신하는 AHIP 설정 관리자 (90)를 포함한다. 상기 AHIP 모듈 (85)은 룩-업 테이블 (look-up table (LUT)) 모듈 (92), 데이터 설정 모듈 (94), 로직 모듈 (96), 그리고 동기 모듈 (98)을 포함한다. 도 4에서 이전에 도시된 것처럼, 특정 RGB 데이터 (102)는 상기 메모리 (84)로 즉시 저장된다. 그러나, 특정한 다른 RGB 데이터 (103)는 상기 AHIP 모듈 (85)에 의해서 하나 또는 그 이상의 픽셀 프로세싱 모듈들 (106), 하나 또는 그 이상의 프레임 프로세싱 모듈들 (107), 하나 또는 그 이상의 구역 프로세싱 모듈들 (108) 및 하나 또는 그 이상의 커널 프로세싱 모듈들 (110)에서 프로세싱된다. 특정 RGB 데이터 (103)는 프레임 프로세싱 모듈에서 프로세싱될 수 있을 것이며, 그리고 그 프레임 데이터 (112)는 메모리 (84)에 저장된다. 특정 RGB 데이터 (103)는 하나 또는 그 이상의 픽셀 프로세싱 모듈들 (106)에서 프로세싱될 수 있을 것이며, 그리고 픽셀 데이터 (114)는 상기 메모리 (84)에 저장되거나 아니면 그 데이터는 커널 프로세싱 모듈 (110)에서 그리고/또는 프레임 프로세싱 모듈 (107)에서 그리고/또는 구역 프로세싱 모듈 (108)에서 더 프로세싱된다. (N-1로 표시된) 이전 프레임과 같은 인접한 프레임의 RGB 데이터 (116)는 구역 프로세싱 모듈 (108)에서 참조번호 103의 RGB 데이터와 같이 프로세싱될 수 있을 것이다. 상기 구역 프로세싱 모듈 (108)에서 프로세싱된 데이터는 그러면 구역 데이터 (118)로서 메모리 (84)에 저장될 수 있을 것이다.

[0071] 예를 들면, 심지어 얼굴 구역의 정밀한 위치에 대한 완전한 확인을 만드는 것이 가능하기 이전에, 때로는 이미지의 일부에 알고리즘을 적용하는 것이 바람직할 수 있을 것이다. 이 예측된 얼굴 구역들은 이전의 이미지 프레임들로부터 결정될 수 있으며 그리고 얼굴 및 카메라의 이전 여러 개의 프레임들에 걸친 이동의 이력이라는 유리함을 취할 수 있다. 이런 면에서, 프레임-투-프레임 dX 및 dY 변위들이 상기 AHIP 모듈에 의해서 결정될 수 있을 것이며 그리고 어떤 프레임의 마지막 픽셀이 프로세싱된 이후의 짧은 지연 이내에 이용가능할 수 있을 것이다. 유사하게, 얼굴 구역의 위치 및 크기는 얼굴 추적기 알고리즘의 하나 또는 그 이상의 프레임들의 마지막 개수에 대해서 정밀하게 알려질 수 있을 것이며 그리고 이 데이터들 둘 모두는 새로운 이미지 프레임 프로세싱이 AHIP에 의해서 시작된 이후에 아주 빨리 이용 가능할 수 있다. 이 데이터들은 현재의 이미지 프레임에 대해서 예측된 얼굴 구역들을 정확하고 동적으로 추정하는 것을 유리하게도 가능하게 한다 (미국 공개 특허 US2009/0303342 및 US2009/0263022, 그리고 US 특허들 7,460,695, 7,403,643 및 7,315,631, 그리고 US2009/00123063, US2009/0190803, US2009/0189998, US20090179999 및 미국 출원 일련번호 12/572,930, 12/824,214 및 12/815,396 참조, 이것들은 동일한 양수인에게 양수된 것이며 그리고 본원에 참조로서 편입된다).

[0072] 얼굴 구역들 또는 눈 구역들 그리고/또는 입 구역들 그리고/또는 반 얼굴 구역들과 같은 얼굴 특정 구역들 (미국 출원 12/790,594 및 12/825,280 참조, 이것들은 동일한 양수인에게 양수된 것이며 그리고 본원에 참조로서

편입된다)은 처음에는 로컬 메모리 버퍼에 저장될 수 있다. 보통, 메모리는 하드웨어 코어로는 비싸기 때문에, 제한된 개수의 "얼굴 버퍼들"이 존재할 수 있을 것이며, 그리고 그것들은 크기를 최소화하기 위해서 최적화될 수 있을 것이다. 실제로 몇몇의 실시예들에서는 그런 버퍼들이 상기 AHIP의 외부에 존재할 수 있을 것이며 그리고 이 구역들을 상기 외부의 메인 메모리 저장부에 쓰기 위한 대역폭이 결부될 수 있을 것이다. 다른 실시예들에서, 메인 이미지 내에 그것들의 위치들은 기록될 수 있을 것이며, 그래서 그것들이 상기 메인 이미지 프레임 내에서 상기 메인 메모리 저장부로부터 나중에 액세스될 수 있도록 할 수 있을 것이다. 일 실시예에서, 그런 구역들은 상기 AHIP 모듈 내에 내부적으로 저장될 것이다. 그러나, AHIP 버퍼의 크기가 풀 HD 비디오에 대해서 너무 큼 수도 있고 아닐 수도 있을 것이기 때문에, 대안의 실시예에서, 이 얼굴 구역들은 메인 메모리 내 메인 이미지와 함께 저장되며, 그리고 다음의 이미지 프레임이 프로세싱되고 그리고 메인 메모리에 써질 때에는 제2 AHIP 모듈로 다시 로딩된다.

[0073] 이 실시예들에서, 많은 얼굴 탐지 및 인식 기술들이 고정된 크기의 얼굴 이미지에 적용될 수 있기 때문에, 이 메모리 버퍼들은 그런 고정된 크기의 얼굴 구역을 몇몇의 추가적인 용량에 수용하기에 충분하게 를 수 있을 것이다. 그 추가의 용량은 여러 팩터들을 보상하기 위해서 사용될 수 있을 것이다: (i) 이미지 구역이 예측되고 그리고 얼굴의 고도로 정밀한 위치는 상기 이미지의 초기 AHIP 프로세싱의 시점에서는 알려지지 않는다; (ii) 상기 얼굴 구역은 바르게 똑바로 서지 않을 수 있을 것이며 그래서 적절하게 바로 선 얼굴 구역을 획득하기 위해서 평면에서 회전될 필요가 있을 수 있을 것이다; 이 정보는 얼굴 구역들의 정밀한 위치가 결정되고 그리고 이 얼굴 구역 내 눈-구역들의 위치가 결정된 이후의 더 나중에 이용 가능하게 될 수 있을 것이다; 그런 시점에, 평면 내 교정 각도 및 리-사이징 정보는 상기 AHIP로 제공될 수 있지만, 상기 얼굴 구역이 대각선으로 위치한 얼굴 구역들을 수용하기 위한 추가의 메모리 공간을 포함하는 평면 내 각도에 위치할 그런 시각까지는 제공되지 않는다; (iii) 상기 얼굴 구역은 프레임마다 글로벌 크기에 있어서 변할 수 있을 것이다; 과거의 이미지 프레임들의 이력으로부터 몇몇의 경향들이 알려질 수 있을 것이지만, 이런 경향들로부터의 변경이 존재할 수 있을 것이라는 것 역시 가능하다. 그래서 메인 이미지 프레임 프로세싱이 완료된 이후에 상기 구역적 데이터가 CPU/GPU에 의해서 프로세싱될 때까지 얼굴의 정밀한 크기는 알려질 수 있을 것이며 또는 알려지지 않을 수 있을 것이다. 다음로 (RAW) 이미지 프레임이 프로세싱되고 쓰여지는 동안에 얼굴 구역들이 로딩되고 그리고 프로세싱되는 대안의 실시예에서, 상기 메인 이미지 및 그것의 연관된 프리메이티브들에 대한 몇몇의 추가적인 프로세싱이 상기 CPU/GPU로부터 이용 가능하다. 그런 경우들에, 얼굴 구역의 위치의 더욱 정밀한 추정이 이용 가능할 수 있을 것이다.

[0074] 다른 예시의 실시예에서, 최대 크기 32x32의 최종 고정된-크기의 얼굴 구역이 결부된다고 가정한다. 메모리 버퍼는 1.99까지의 잠재적인 단편적인 리사이징 그리고 45도까지의 회전을 수용하기 위해서 64x64 얼굴 구역에 대해서 사용될 수 있을 것이다. 이는 정밀하게 위치한 얼굴 구역에 대한 버퍼 크기를 정의하지만, 현재 프레임이 먼저 AHIP에 의해서 프로세싱될 때에 우리는 얼굴이 존재할 것 같은 곳인 그 이미지 프레임의 예측된 구역을 알뿐이다. 따라서, 더 큰 구역을 저장하는 것이 유용할 수 있을 것이다. 예를 들면, 96x96 또는 128x128 픽셀들이 사용되어 예상되는 얼굴 구역의 치수들보다 50% 또는 100% 더 큰 예측된 얼굴 구역 버퍼들을 제공할 수 있을 것이다.

[0075] 예측된 얼굴 구역들의 위치 및 크기에 관한 데이터가 이전 이미지 프레임의 소프트웨어 프로세싱으로부터 상기 AHIP로 지나갈 때에, 이 예측된 얼굴 구역들은 96x96 또는 128x128의 개별 AHIP 버퍼들로 정수 다운사이즈된다 (integer downsized). 얼굴 구역들을 프로세싱하기 위해서 보통은 휘도 값들을 유지하는 것만이 필요하며 그래서 이 버퍼들은 단일-값이며 그리고 풀 컬러 데이터는 보통은 유지되지 않는다. 그러나 몇몇의 실시예들에서 스키-맵의 모습이 생성되며 그리고 그 얼굴 구역의 휘도 이미지와 함께 저장될 수 있을 것이다. 전형적으로 얼굴 추적 목적들을 위한 스키-맵은 스키-픽셀의 3개 또는 4개 상태들만을 나타내는 단지 2비트일 수 있을 것이다.

[0076] 그런 프리메이티브의 예는 얼굴 분류기들을 추적된 얼굴 구역에 적용하는 것이다. 보통 얼굴 구역은 정확하게 수평이 아닐 것이며 그래서 예측된 얼굴 구역으로부터의 데이터는 상기 얼굴 분류기들을 적용하기 이전에 먼저 수평 방위로 회전될 수 있을 것이다. 그런 조작들은 이미지 데이터를 위한 더 큰 버퍼들을 필요로 하며 그래서 구역적인 버퍼들의 개수 및 각각의 크기는 다소 제한된다. 이 예에서 상기 프리메이티브들 출력은, (i) 이 구역은 확인된 얼굴을 여전히 포함한다; 그리고/또는 (ii) 특정 벡터 세트는, 매치할 것 같다는 것을 표시하기 위해서, 알려진 얼굴들의 세트에 대해서 매치될 수 있다는 것을 확인하기 위해서 사용될 수 있는 특정 벡터 (feature vector) 세트에 대응하는 데이터 값들의 세트이다:

[0077] 도 6은 메인 CPU, AHIP 하드웨어 모듈, 이미지 센서 및 SDRAM 메모리 (126)의 읽기/쓰기 채널들 (125) 사이의 상호 관계들을 도시한다. AHIP/CPU 인터페이스 (128)는 특정 레지스터들 및/또는 인터럽트들을 제공한다. YUV

데이터를 위한 B2Y 인터페이스 (130)는 도 6에서 상기 센서 (124)와 상기 AHIP 블록 (122) 사이에서 보인다.

[0078] 이미지 프레임 프로세싱의 몇 가지 원칙 유형들은 상기 AHIP 내에서 구현된다. 이것들 중 특정의 원칙 유형들이 이제 아래에서 설명된다.

[0079] 프레임-투-프레임 AHIP 프로세싱 (FRAME-TO-FRAME AHIP PROCESSING)

[0080] 도 7은 현재의 프레임 N과 결합된 프레임 N-1로부터의 데이터를 사용하는 것을 도시한다. 메모리 저장부 (이미지 데이터 및 캐시) (140)는 도 7의 예시에서 두 개의 프레임들로 세그먼트된다. 상기 도면들 중의 몇몇은 도 4를 참조하여 설명되었으며 여기에서는 반복해서 설명하지 않는다. 예로서 얼굴 추적에 관련하여, 예측된 구역들 (142)은 프레임 N-1 및 프레임 N 둘 모두에 대해서 메모리 (140)에 저장된다. 상기 메모리 (140)의 프레임 N-1 부분 내에서, 개선된 구역들 및 구역적 프리미티브 데이터 (143)가 저장된다. 이것들은 프레임 N에 대해서 RGB 이미지 프레임 데이터 및 프레임 프리메이티브들 (145)을 생성하기 위해서 사용된다. 프레임 N-1 구역적 데이터 (144)는 프레임 N RGB 데이터 (149)와 함께 상기 AHIP 블록 (148)으로 입력된다. 상기 AHIP 블록은 블록 150에서 프레임 N-1 구역 데이터의 일정 부분을 서브샘플링하고 그리고/또는 컬러 공간 리-맵 (re-map)하며, 그리고 상기 프레임 N-1 구역 데이터 (144) 전부를 픽셀 프로세싱 블록 (154)을 통해서 이동시켜서 프레임 N 부분에서 메모리 (140)로 출력되도록 한다. 상기 프레임 N RGB 데이터 (149)는 서브-샘플링되고 (156) 그리고/또는 컬러-공간 리-맵되며 (158) 또는 그 이후에 블록 160에서 픽셀 프로세싱 될 수 있을 것이다. 상기 픽셀 프로세싱 (160)은 누적 픽셀 프로세싱, 직접 픽셀 프로세싱, 커널 픽셀 프로세싱, 구역 픽셀 프로세싱 및/또는 하이브리드 픽셀 프로세싱을 포함할 수 있을 것이다. 그러면 상기 데이터는 상기 메모리 (140)의 프레임 N 부분에서 저장된다.

[0081] 특정 실시예들에 따른 상기 AHIP 모듈의 다양한 서브-엘리먼트들의 일정 부분에 대한 설명들이 이제 간략하게 기술된다. 상기 AHIP 모듈은 이 예의 서브-엘리먼트들 중의 어느 것이나 모두를 포함할 수 있고 또는 전혀 포함하지 않을 수 있을 것이다.

[0082] 컬러 맵 유닛 (THE COLOR MAP UNIT)

[0083] 컬러 맵 유닛 모듈은 유사한 컬러들을 구비한 픽셀들의 맵을 생성한다. 레퍼런스 포인트 R은 입력 컬러 공간 (예를 들면, RGB)에서 미리-프로그램될 수 있을 것이다. 그러면, 각 입력 픽셀 P에 대해서, 컬러맵 모듈은 P와 R 사이의 유클리드 거리 d를 계산할 수 있을 것이며, 그리고 그것을 15개의 프로그램 가능한 문턱값들 (T1 내지 T15)과 비교할 수 있을 것이다. 이것들은 캐스케이드 식으로 추정될 수 있을 것이며, 그래서 그것들은 증가하는 순서 ($T_1 < T_2 \dots < T_{15}$ - 매우 엄밀한 것으로부터 매우 임의적으로)로 유리하게 배치될 수 있을 것이다. $d < T_n$ 이면 그러면 상기 모듈은 $16-n$ 의 값을 출력한다. 어떤 문턱값도 매치되지 않으면, 그러면 0이 송신된다. 도 8은 특정 실시예들에 따른 컬러 문턱값 모듈을 도시한다.

[0084] 컬러맵의 출력은 보통은 4 비트로 표현될 수 있을 것이다. 몇몇의 실시예들에서, 예를 들면, 8개 (3 비트), 또는 4개 (2 비트) 또는 2개 (1 비트)만을 이용하여 문턱값들의 더 작은 세트가 구현될 수 있을 것이다. 상기 컬러맵 모듈은 풀 해상도 이미지 입력 상에서 동작할 수 있을 것이다. 출력 맵은 입력과 동일한 해상도를 가질 수 있을 것이며 그리고 각 픽셀은 매칭 출력을 생성하여 그것이 3D 컬러 공간에서 미리-프로그램된 레퍼런스 픽셀에 얼마나 가까운가를 표시할 수 있을 것이다.

[0085] 대안의 실시예들에서, 추가적인 병렬의 컬러 모듈들은 상기 입력 채널들 중 하나를 0 또는 중간-포인트 값으로 세팅함으로써 어떤 픽셀이 3개 컬러 채널들 중 2개에 얼마나 가깝게 매치되었는가를 결정하는 것을 가능하게 할 수 있을 것이다. 다른 실시예들은 비교를 위해서 3개 채널들 중 2개만이 사용되는 경우에 2D 컬러 모듈을 구현한다.

[0086] 컬러맵 모듈의 기능이 유클리언 거리에 의존하기 때문에, 입력 컬러 공간은 이 거리가 의미있는 곳이 되어야만 한다 (근접한 것은 시각적인 유사성과 동등하다). 보통은, 상기 모듈은 RGB 입력과 함께 사용되어야만 하지만, 그것으로 한정되는 것은 아니다.

[0087] 도 9에서, 이미지에 적용되고 있는 컬러맵 모듈의 예가 제공된다. 이 예에서 레퍼런스 포인트는 RGB 좌표에서 주어진 스킨-유사한 컬러이다. 도 9의 왼쪽에 입력 이미지가 보이며 그리고 오른쪽에 컬러맵이 존재한다 (흰색은 최대 값을 나타내며, 그 범위는 0 내지 15이다). 도 9의 이 예에서, 16개의 문턱값들 (4 비트)이 사용되었다.

[0088]

[0089] 컬러 공간 변환 유닛 (COLOR SPACE CONVERSION (CSC) UNIT)

[0090] CSC는 다음과 같이 정의된 프로그램 가능한 매트릭스 곱셈기를 구현하기 위해서 사용될 수 있을 것이다.

[0091] A CSC may be used to implements a programmable matrix multiplier defined as follows:

$$Ax + B = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \times \begin{bmatrix} Y \\ U \\ V \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} a_{11}Y + a_{12}U + a_{13}V + b_1 \\ a_{21}Y + a_{22}U + a_{23}V + b_2 \\ a_{31}Y + a_{32}U + a_{33}V + b_3 \\ a_{41}Y + a_{42}U + a_{43}V + b_4 \end{bmatrix}$$

[0092]

[0093] $x = [Y, U, V]$ = 입력 픽셀, A 및 B = 프로그램 가능한 계수들을 가진 매트릭스들.

[0094]

이 구조는 YUV-2-RGB 또는 RGB-2-YUV와 같은 변환들을 수행할 수 있다. 그것은 YUV와 커스텀 컬러 공간들 사이에서의 변환들을 또한 수행할 수 있다. 이 변환들은 특별한 특성들을 가진 픽셀들을 식별하기 위해서 다른 모듈들과 결합하여 사용될 수 있을 것이다.

[0095]

(값들을 0 내지 255의 정수 범위로 제한하기 위해) 채도 기능은 상기 CSC의 출력 단에서 구현될 수 있을 것이다.

[0096]

도 10은 이미지 센서 (172)로부터 RGB 데이터를 수신하고 그리고 픽셀 데이터 그리고/또는 프레임 데이터를 포함하는 데이터를 메모리 (174)로 제공하는 AHIP 모듈 (170)을 도시한다. 도 10의 예시의 AHIP 모듈 (170)은 픽셀 프로세싱 모듈들 (176)을 포함한다. 상기 모듈들 (176) 중의 둘은 문턱값 모듈들 (178)로 피드 (feed)하며 그리고 상기 모듈들 (176) 중의 둘은 카운팅 모듈들, 하나의 히스토그램 (180) 및 다른 레지스트레이션 (182)으로 피드한다. 로직 모듈 (184)은 상기 문턱값 모듈들 (178)로부터 입력을 수신하고 그리고 메모리 (174)로 픽셀 데이터를 출력한다. 빙 카운터 (186)는 카운팅 (히스토그램) 모듈 (180)로부터 입력을 수신하고 그리고 메모리 (174)로 프레임 데이터를 출력한다. dX, dY 오프셋 블록 (188)은 카운팅 (레지스트레이션) 모듈 (182)로부터 입력을 수신하고 그리고 상기 메모리 (174)로 프레임 데이터를 출력한다. 제1 RGB-2-YUV 블록 (190)은 누산기 (인테그랄 이미지) 모듈 (192)로 출력하고, 이는 메모리 (174)로 픽셀 데이터를 출력한다. 제2 RGB-2-YUV 블록 (190)은 제곱 모듈 (194)로 그리고 누산기 (인테그랄 제곱 이미지) 모듈 (196)로 출력하며, 이는 차례로 메모리 (174)로 픽셀 데이터를 출력한다.

[0097]

도 11은 CPU (202)와 통신하는 AHIP 블록 (200)을 도시하며, 이는 상기 CPU (202)로 인터럽트들 (204)을 제공하고 그리고 상기 CPU (202)로부터 설정 명령들 (206)을 수신하는 것을 포함한다. 상기 AHIP 블록 (200)은 B2Y 블록 (210)을 경유하여 센서/B2Y (208)로부터 데이터를 순하고 그리고 메모리 (212)로 출력한다.

[0098]

문턱값 (THRESHOLDING (THR))

[0099]

이 모듈은 다른 모듈들로부터의 출력들에 문턱값을 정하고 그리고 그것들을 이진의 값들 (1비트)로 변환하기 위한 4개의 8x1 LTU들을 포함한다. 다중의 문턱값 유닛들이 상기 AHIP 내에서 통합될 수 있다.

[0100] 로직 함수 (LOGIC FUNCTION (LF))

[0101] 이 모듈은 6x6 LTU를 포함하며, 이는 상기 THR 또는 컬러맵 모듈과 같은 다른 AHIP 모듈의 결과들을 더 결합하기 위해서 사용될 수 있다. 그것은 6개의 상이한 로직 함수들을 상기 THR 및 스킨 모듈들로부터의 입력들과 함께 구현할 수 있다. 다중의 로직 유닛들은 상기 AHIP 내에서 통합될 수 있다.

[0102] 히스토그램들 (HISTOGRAMS (HIST))

[0103] 컴퓨터 히스토그램들은 원래의 입력 데이터 (YUV)에 그리고/또는 상기 CSC 모듈의 출력에 적용될 수 있을 것이다. 이 모듈들은 각 픽셀로부터 카운트 값을 축적하며 그리고 그것들의 출력은 상기 이미지 프레임의 각 픽셀이 상기 AHIP를 통해서 블록을 공급받은 이후에 이용 가능하다. 히스토그램 빈들의 개수는 보통은 16개이며 그리고 각 빈의 문턱값은 프로그램 가능하다.

[0104] 인테그랄 이미지 누산기 (INTEGRAL IMAGES ACCUMULATORS) (II) - AKA

[0105] 영역 계산 모듈들 (AREA COMPUTATION MODULES)

[0106] 이 모듈들은 인테그랄 이미지를 계산하기 위한 블록들을 포함한다 (직사각형 구역들 내에서 빠른 영역 계산들을 허용한다; 예를 들면 상기에서 참조로 편입된 미국 특허 7,315,631 그리고 본원에 참조로서 편입된 미국 특허 출원 US 2002/0102024 참조). 그들은 일반적으로 실시간 오브젝트 탐지 알고리즘들에서 채택된다. 세 개의 표준 블록들 중의 어떤 것도 이용 가능할 수 있을 것이다:

[0107] II (원래의 Y 채널에 대해서 합해진 영역, 영역 계산을 위해서 사용됨)

[0108] II2 (Y 제곱을 통한 합, 변이 계산을 위해서 사용됨)

[0109] 스킨 (skin) II (스킨 맵 상의 인테그랄 이미지, 영역 내 스킨 밀도를 준다).

[0110] 이 모듈들은 각 입력 픽셀 상에서 자신들의 값을 축적하고 그리고 현재의 축적된 값을 출력하여 상기 인테그랄 이미지 또는 인테그랄 변이 또는 스킨 인테그랄 이미지 맵들의 대응 픽셀 값을 제공하도록 한다 (미국 특허 출원 2010/0053368 참조, 이는 참조로서 편입된다).

[0111] 복소수 다운샘플러 (COMPLEX DOWNSAMPLER (DS))

[0112] 이 다운샘플러는 다른 모듈들을 위해서 상기 이미지를 크기 조절한다. 상기 모듈은 낮은 게이트-카운트를 구비하면서도 합리적인 레벨의 품질을 제공하려고 설계되었다. 그것은 상기 메인 이미지의 가변 다운사이징을 달성하기 위해서 프로그램될 수 있다.

[0113] 고정된 다운샘플러들 (FIXED DOWNSAMPLERS (XDS))

[0114] 더 단순한 다운샘플러 구현은 가장 가까운-이웃 보간법 (nearest-neighbor interpolation)을 이용하여 또한 이용 가능하다. 이것은 고정된 해상도로 맵들을 동시에 계산하는 것을 허용한다. 보통 x2, x4 및 x8 다운샘플러들이 이용 가능할 것이다.

[0115] 워크플로우 (WORKFLOW)

[0116] AHIP는 디-베이어 (De-Bayer) 모듈 (센서 인터페이스) 이후에 상기 이미지 파이프라인의 끝 부분에서 통합되도록 설계될 수 있을 것이다. YUV가 AHIP의 고유한 컬러 공간인 것으로 간주되지만, 몇몇의 내부 모듈들은 다른 컬러 공간들도 처리할 수 있다. 상기 센서 인터페이스에 추가로, AHIP는 메모리로부터 데이터를 읽기 위한 플레이백 인터페이스를 제공한다. 미리 보기 모드에서, AHIP는 센서 클록 레이트 (픽셀 당 1 클럭)에서 데이터를 처리하도록 구성될 수 있을 것이다. AHIP에 의해 프로세싱된 데이터는 타겟 시스템 (SDRAM)의 메인 메모리에 써질 수 있을 것이다. 하나 또는 그 이상의 상기 모듈들은 레지스터들로 값들을 출력할 수 있을 것이다. AHIP는 다른 모듈들로 스트리밍 출력 인터페이스를 제공하도록 구성될 수 있을 것이며 또는 구성되지 않을 수 있을 것이다.

[0117] AHIP를 갖춘 전형적인 워크플로우의에는 다음과 같다:

- 센서로부터 또는 메모리로부터의 라이브 데이터는 AHIP로 송신된다 (한번에 한 픽셀).

- 프레임의 주기 동안에, AHIP는 자신의 계산들을 수행한다.

- 하나 또는 그 이상의 모듈들은 내부 통계치들을 업데이트한다.

- 하나 또는 그 이상의 다른 모듈들은 시스템 메모리에 데이터 (맵들)를 쓴다. 이는 입력에 관련하여 최소의 지연으로 행해지는 것이 보통일 것이다 (즉, 픽셀들이 들어오면 데이터는 나간다).

- 프레임의 끝 부분 이후에, AHIP는 CPU로의 인터럽트를 일으킨다.

- 보통의 상태들에서, 상기 인터럽트는 상기 CPU에게 프레시 데이터가 이용 가능하다는 것을 통지한다.

- 오류 상태들을 시그널링하는 인터럽트들 또한 존재할 수 있을 것이다.

- CPU 인터럽트 핸들러가 호출된다.

- 그것은 상기 인터럽트의 원인을 찾기 위해서 AHIP 레지스터들을 읽는다.

- 그것은 그러면 옵션으로 AHIP를 재설정하고 (예를 들면, 그렇게 하는 것이 유리할 것이라는 결정을 기반으로 한다), 그리고 상기 인터럽트 수신 확인을 한다.

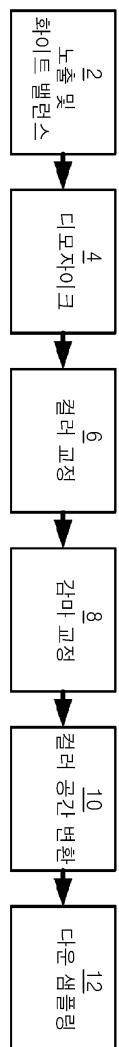
- 그것은 하드웨어로부터의 데이터가 이용 가능하다는 것을 (그리고 알고리즘들이 그것을 사용하기 시작했다는 것을) 다른 CPU 스레스들에게 또한 시그널링한다.

[0129] 본 발명의 예시의 도면들 및 특정 실시예들이 설명되고 도시되었지만, 본 발명의 범위는 설명된 그런 특정 실시예들로 한정되는 것이 아니라는 것이 이해되어야만 한다. 그래서, 상기 실시예들은 제한적인 것이 아니라 예시적인 것으로서 여겨질 것이며, 그리고 본 발명의 범위를 벗어나지 않으면서도 본 발명이 속한 기술 분야에서의 통상의 지식을 가진 자들에 의해 그 실시예들에 수정들을 할 수 있다는 것이 이해되어야만 한다.

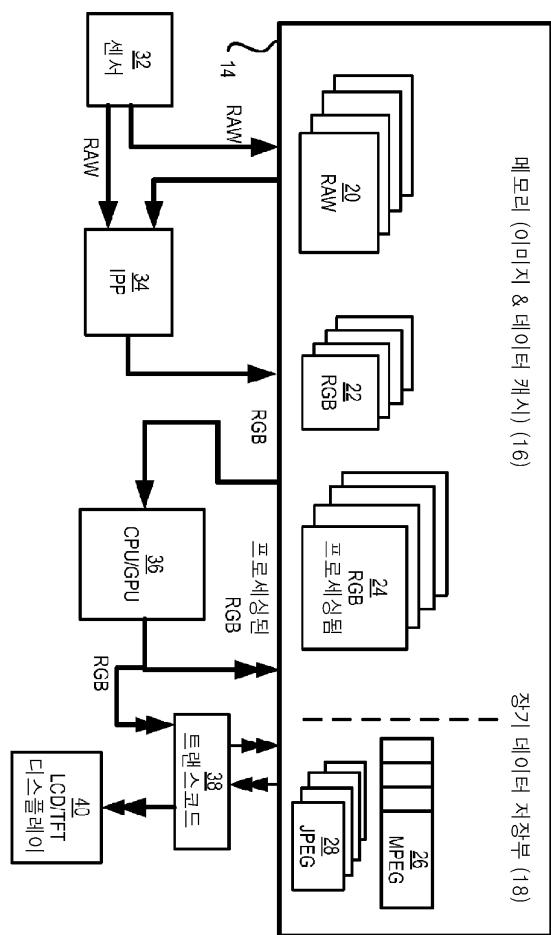
[0130] 추가로, 여기에서의 바람직한 실시예들에 따라 수행될 수 있을 그리고 상기에서 설명되었을 방법들에서, 상기 오퍼레이션들은 선택된 인쇄상 시퀀스들로 설명되었다. 그러나, 그 시퀀스들은 작성 상의 편의를 위해서 선택되고 그리고 그렇게 순서가 정해진 것이며 그리고, 특별한 순서가 명백하게 제시된 경우 또는 본 발명이 속한 기술 분야에서의 통상의 지식을 가진 자들이 특별한 순서가 필요한 것으로 간주할 수 있는 경우를 제외하면, 그 오퍼레이션들을 수행하기 위한 어떤 특별한 순서를 의미하려고 의도된 것은 아니다.

도면

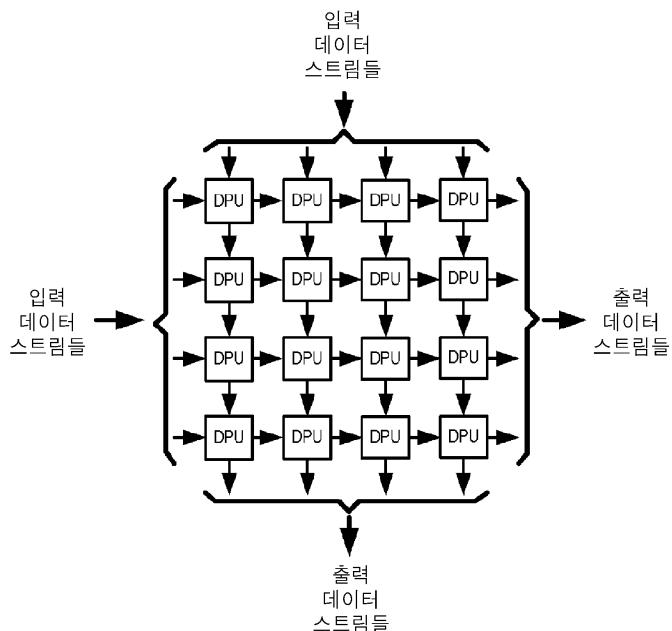
도면1



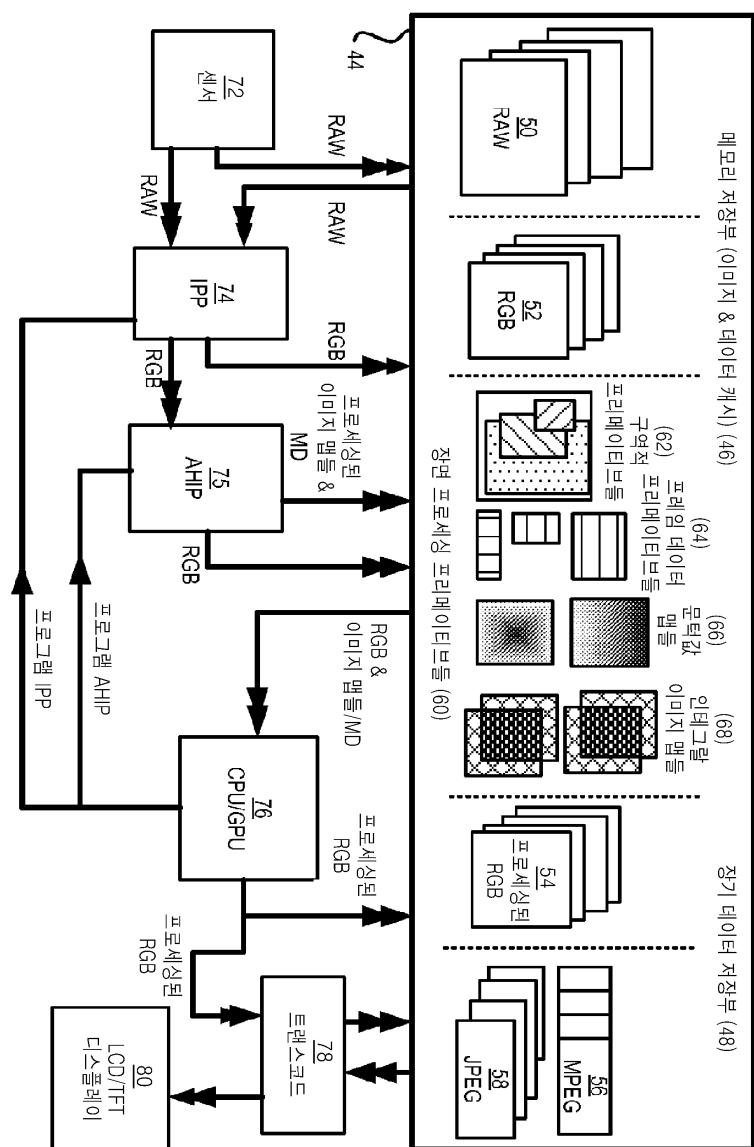
도면2



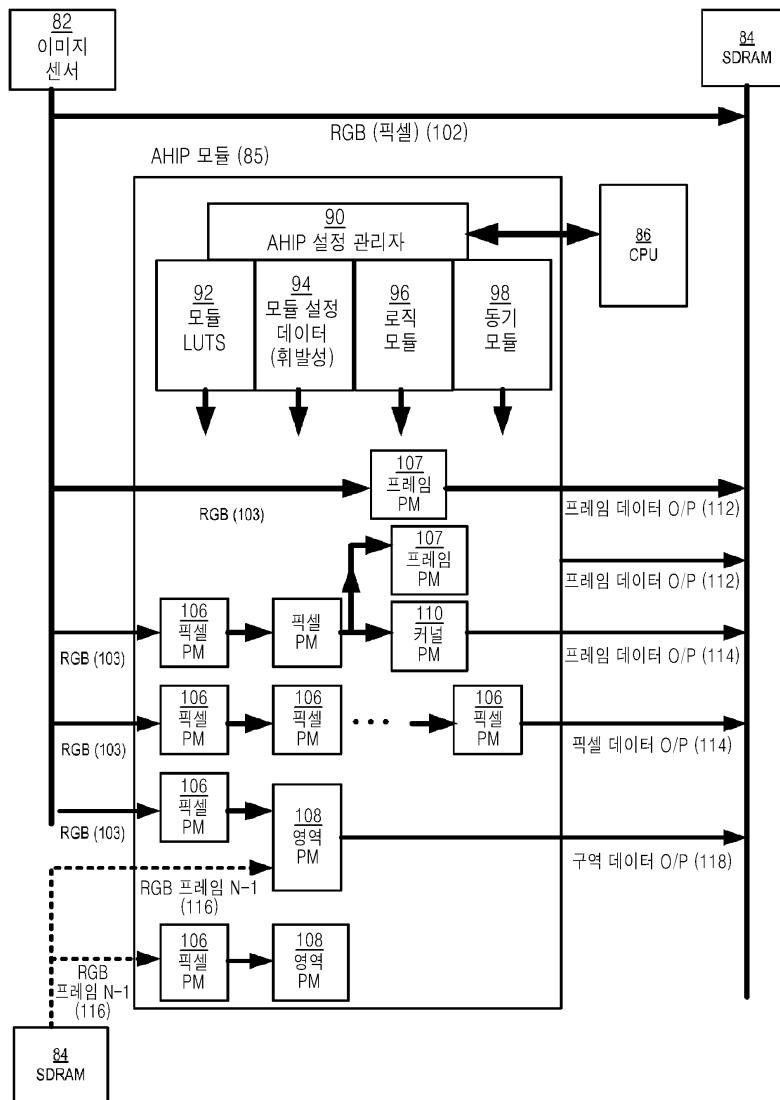
도면3



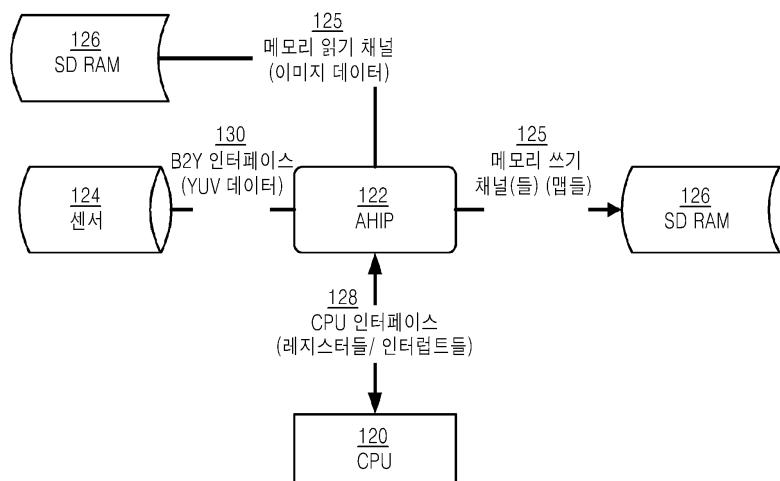
도면4



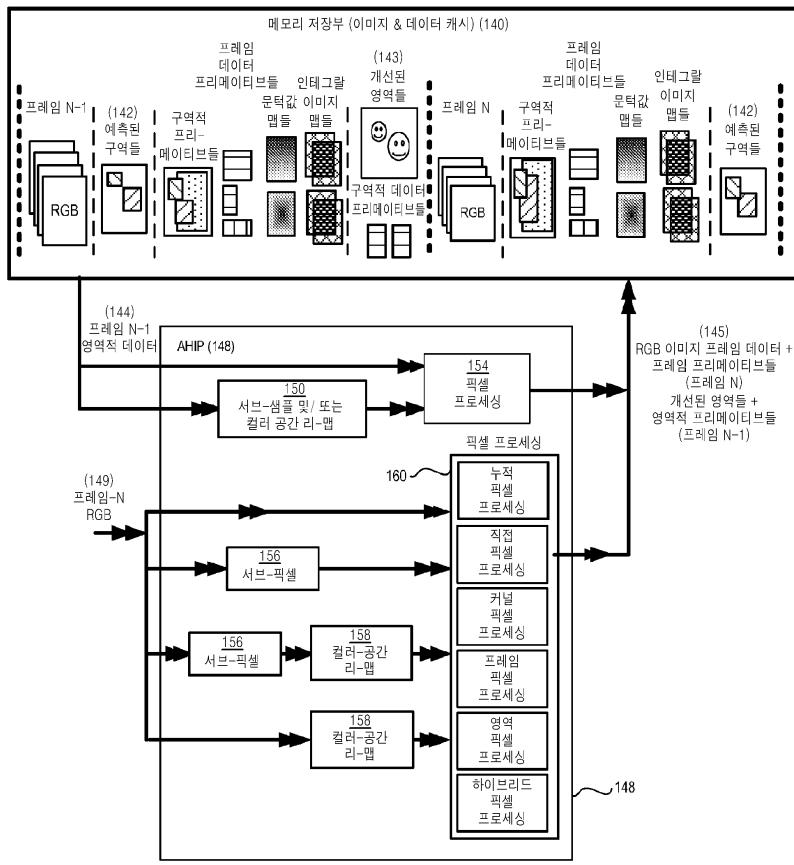
도면5



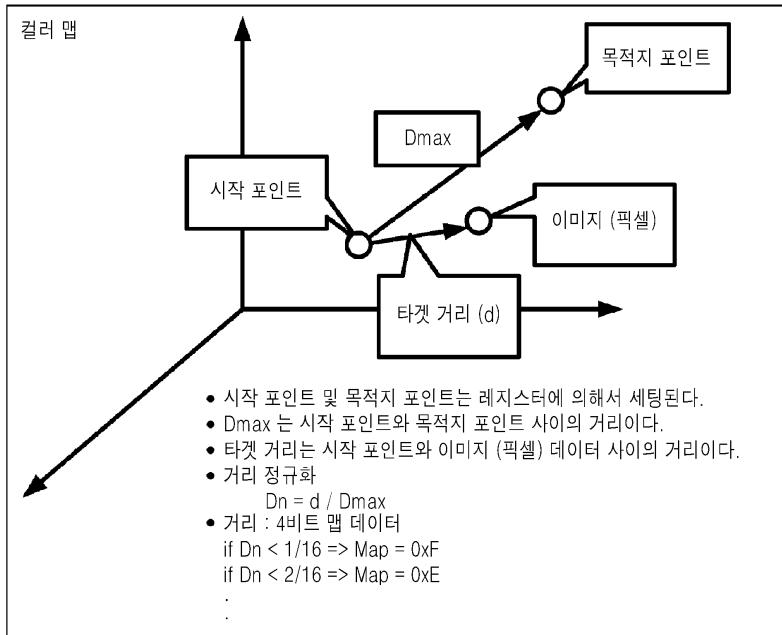
도면6

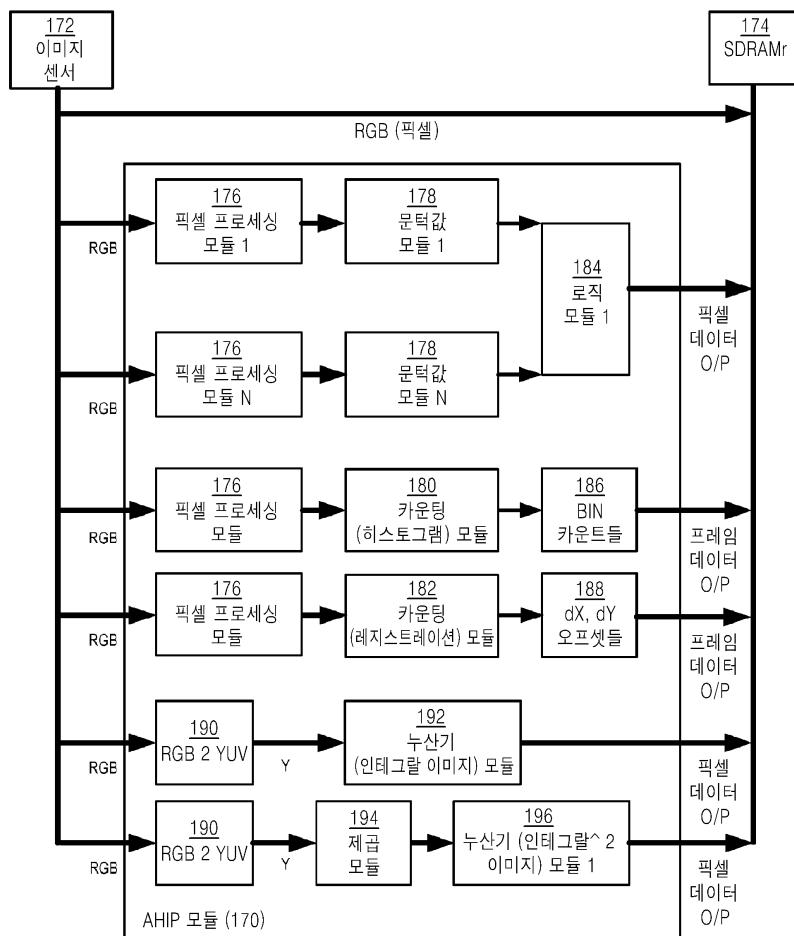


도면7



도면8



도면9**도면10****도면11**