



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년05월31일
(11) 등록번호 10-1626236
(24) 등록일자 2016년05월25일

(51) 국제특허분류(Int. Cl.)
G06T 15/40 (2011.01) G06T 1/00 (2006.01)
(21) 출원번호 10-2014-7027414
(22) 출원일자(국제) 2013년02월08일
심사청구일자 2015년09월08일
(85) 번역문제출일자 2014년09월29일
(65) 공개번호 10-2014-0139526
(43) 공개일자 2014년12월05일
(86) 국제출원번호 PCT/US2013/025375
(87) 국제공개번호 WO 2013/130241
국제공개일자 2013년09월06일
(30) 우선권주장
13/409,993 2012년03월01일 미국(US)
(56) 선행기술조사문헌
US20090102857 A1
US20040012609 A1
US20070040851 A1
JP2009181582 A

(73) 특허권자
켈컴 인코퍼레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하
우스 드라이브 5775
(72) 발명자
그루버 앤드류
미국 92121-1714 캘리포니아주 샌디에고 모어하
우스 드라이브 5775
(74) 대리인
특허법인코리아나

전체 청구항 수 : 총 44 항

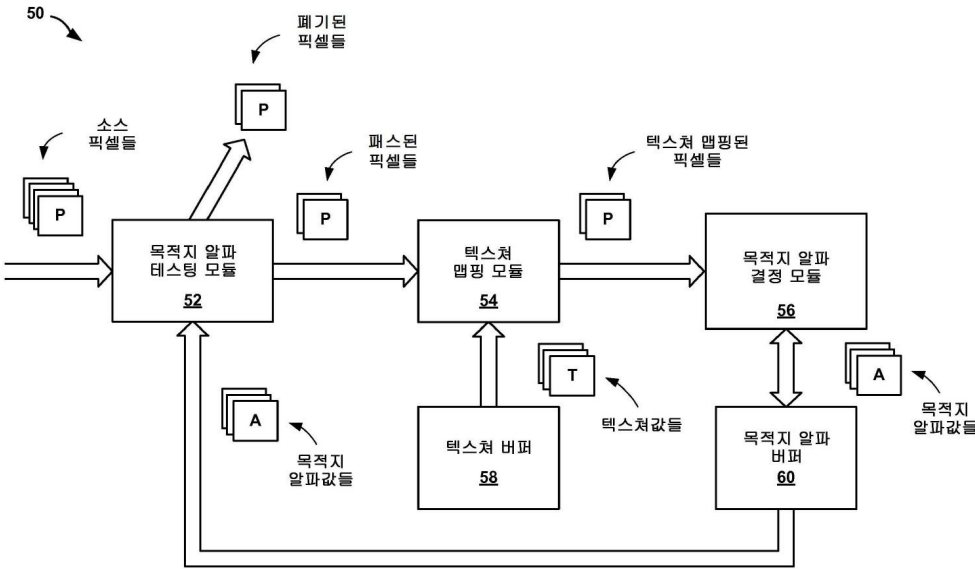
심사관 : 조우연

(54) 발명의 명칭 목적지 알파값들에 기초한 그래픽스 프로세싱 시스템에서 메모리 액세스 대역폭을 감소시키기 위한 기법들

(57) 요약

본 개시물은 목적지 알파값들에 기초하여 그래픽스 프로세싱 시스템에서 메모리 액세스 대역폭을 감소시키기 위한 기법들을 기재한다. 기법들은 빈 버퍼로부터 목적지 알파값을 추출하는 것을 포함할 수도 있으며, 목적지 알파값은 제 1 프리미티브와 연관된 제 1 픽셀을 프로세싱하는 것에 응답하여 생성된다. 기법들은 또한, 목적지 알파값에 기초하여 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하는 것을 포함할 수도 있다. 일부 예들에서, 액션은 제 2 픽셀이 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에 픽셀 프로세싱 파이프라인으로부터 제 2 픽셀을 폐기하는 것을 포함할 수도 있다. 제 2 픽셀은 제 1 프리미티브와 상이한 제 2 프리미티브와 연관될 수도 있다.

대표도



명세서

청구범위

청구항 1

그래픽 프로세싱을 위한 방법으로서,

그래픽스 프로세서의 그래픽스 파이프라인에 포함된 래스터화 블록에 의해, 제 1 프리미티브에 대응하는 하나 이상의 소스 픽셀들을 생성하도록 상기 제 1 프리미티브를 래스터화하는 단계로서, 상기 제 1 프리미티브에 대응하는 상기 하나 이상의 소스 픽셀들은 제 1 소스 픽셀을 포함하는, 상기 제 1 프리미티브를 래스터화하는 단계;

상기 래스터화 블록에 의해, 상기 제 1 프리미티브와 상이한 제 2 프리미티브에 대응하는 하나 이상의 소스 픽셀들을 생성하도록 상기 제 2 프리미티브를 래스터화하는 단계로서, 상기 제 2 프리미티브에 대응하는 상기 하나 이상의 소스 픽셀들은 제 2 소스 픽셀을 포함하는, 상기 제 2 프리미티브를 래스터화하는 단계;

상기 그래픽스 프로세서의 상기 그래픽스 파이프라인에 포함된 픽셀 프로세싱 파이프라인에 의해, 빈 버퍼(bin buffer)로부터 목적지 알파값을 추출하는 단계로서, 상기 목적지 알파값은 상기 래스터화 블록에 의해 생성되고 상기 제 1 프리미티브와 연관된 상기 제 1 소스 픽셀을 프로세싱하는 것에 응답하여 생성되는, 상기 목적지 알파값을 추출하는 단계;

상기 픽셀 프로세싱 파이프라인에 의해 그리고 상기 목적지 알파값에 기초하여, 상기 래스터화 블록에 의해 생성된 상기 제 2 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하는 단계로서, 상기 제 2 소스 픽셀은 상기 제 2 프리미티브와 연관되고, 상기 그래픽스 프로세서 및 상기 빈 버퍼는 제 1 마이크로칩 상에 상주하고, 상기 텍스처 버퍼는 상기 제 1 마이크로칩과 상이한 제 2 마이크로칩 상에 상주하는, 상기 액션을 수행할지 여부를 결정하는 단계; 및

조건이 만족되는지를 결정하기 위해 상기 목적지 알파값을 평가하여, 상기 조건이 만족되는 경우, 상기 픽셀 프로세싱 파이프라인에 의해, 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 상기 액션을 수행하는 단계를 포함하는, 그래픽 프로세싱을 위한 방법.

청구항 2

제 1 항에 있어서,

상기 액션을 수행할지 여부를 결정하는 단계는, 상기 제 2 소스 픽셀이 상기 목적지 알파값에 기초하여 상기 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에, 상기 픽셀 프로세싱 파이프라인으로부터 상기 제 2 소스 픽셀을 폐기할지 여부를 결정하는 단계를 포함하고,

상기 액션을 수행하는 단계는, 상기 목적지 알파값에 기초하여 상기 제 2 소스 픽셀을 폐기하도록 결정하는 것에 응답하여 상기 제 2 소스 픽셀이 상기 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에, 상기 픽셀 프로세싱 파이프라인으로부터 상기 제 2 소스 픽셀을 폐기하는 단계를 포함하는, 방법.

청구항 3

제 2 항에 있어서,

상기 목적지 알파값에 기초하여 상기 제 2 소스 픽셀을 폐기하지 않도록 결정하는 것에 응답하여 텍스처 맵핑 스테이지를 포함하는 하나 이상의 프로세싱 스테이지들의 세트로 상기 제 2 소스 픽셀을 패스하는 단계를 더 포함하는, 방법.

청구항 4

제 1 항에 있어서,

상기 액션은 제 1 액션이고,

상기 방법은,

상기 제 1 액션을 수행하지 않도록 결정하는 것에 응답하여 상기 제 2 소스 픽셀에 대한 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되게 하는 제 2 액션을 수행하는 단계를 더 포함하는, 방법.

청구항 5

제 1 항에 있어서,

상기 액션을 수행할지 여부를 결정하는 단계는,

상기 목적지 알파값을 임계값과 비교하는 단계; 및

상기 목적지 알파값을 상기 임계값과 비교하는 단계에 의해 산출된 결과에 기초하여 상기 액션을 수행할지 여부를 결정하는 단계를 포함하는, 방법.

청구항 6

제 5 항에 있어서,

상기 목적지 알파값을 상기 임계값과 비교하는 단계는, 상기 목적지 알파값이 상기 임계값 이하인지 여부를 결정하는 단계를 포함하는, 방법.

청구항 7

제 6 항에 있어서,

상기 액션을 수행하는 단계는, 상기 목적지 알파값이 상기 임계값 이하라고 결정하는 것에 응답하여 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 상기 액션을 수행하는 단계를 포함하고,

상기 방법은, 상기 목적지 알파값이 상기 임계값 이하가 아니라고 결정하는 것에 응답하여 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되게 하는 액션을 수행하는 단계를 더 포함하는, 방법.

청구항 8

제 5 항에 있어서,

상기 임계값은 구성가능한 임계값인, 방법.

청구항 9

제 1 항에 있어서,

상기 빈 버퍼로부터 상기 목적지 알파값을 추출하는 단계는, 제 1 통신 인터페이스를 통해 상기 빈 버퍼로부터 상기 목적지 알파값을 추출하는 단계를 포함하고,

상기 그래픽스 프로세서는 상기 제 1 통신 인터페이스와 상이한 제 2 통신 인터페이스를 통해 상기 텍스처 버퍼로부터 상기 하나 이상의 텍스처값들을 추출하도록 구성된, 방법.

청구항 10

제 9 항에 있어서,

상기 제 1 통신 인터페이스는 상기 제 2 통신 인터페이스의 대역폭보다 큰 대역폭을 갖는, 방법.

청구항 11

제 1 항에 있어서,

상기 빈 버퍼는 프레임 버퍼에 포함된 데이터의 서브세트를 저장하는, 방법.

청구항 12

제 1 항에 있어서,

그래픽스 이미지는 복수의 목적지 픽셀들을 포함하고, 상기 그래픽스 프로세서는 복수의 렌더링 패스들에서 상기 그래픽스 이미지들을 렌더링하도록 구성되고, 상기 렌더링 패스들의 각각이 상기 목적지 픽셀들의 각 서브세트와 연관되고, 상기 빈 버퍼의 용량이 상기 그래픽스 이미지와 연관된 상기 복수의 목적지 픽셀들 모두에 대한 픽셀 데이터를 저장하기 위해 필요한 최소 용량보다 적으며, 그리고 각각의 렌더링 패스에 대하여, 상기 빈 버퍼가 각각의 상기 렌더링 패스와 연관된 목적지 픽셀들의 서브세트에 대한 픽셀 데이터를 저장하는, 방법.

청구항 13

제 1 항에 있어서,

상기 액션을 수행할지 여부를 결정하는 단계는,

상기 목적지 알파값에 기초하여 그리고 스텐실 테스트 블록 또는 z/스텐실 테스트 블록 중 하나에 의해, 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하는 단계를 포함하는, 방법.

청구항 14

제 1 항에 있어서,

상기 액션을 수행할지 여부를 결정하는 단계는,

상기 목적지 알파값에 기초하여 그리고 셰이더 (shader) 유닛 상에서 실행하는 셰이더 프로그램에 의해, 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하는 단계를 포함하는, 방법.

청구항 15

제 1 항에 있어서,

상기 액션을 수행할지 여부를 결정하는 단계는,

상기 목적지 알파값에 기초하여, 상기 제 2 소스 픽셀에 대응하는 목적지 픽셀이 상기 목적지 픽셀에 대응하는 하나 이상의 미리 프로세싱된 소스 픽셀들로 인해 불투명한지 여부를 결정하는 단계를 포함하는, 방법.

청구항 16

제 1 항에 있어서,

상기 제 2 소스 픽셀은 픽셀 로케이션과 연관되고, 상기 제 2 소스 픽셀에 대한 목적지 알파값은 상기 픽셀 로케이션과 연관된 목적지 픽셀의 불투명도를 표시하는 값이고, 상기 목적지 픽셀의 불투명도는 상기 픽셀 로케이션과 연관된 하나 이상의 미리 프로세싱된 소스 픽셀들에 의해 적어도 부분적으로 결정되는, 방법.

청구항 17

그래픽 프로세싱을 위한 디바이스로서,

제 1 프리미티브에 대응하는 하나 이상의 소스 픽셀들을 생성하기 위해 상기 제 1 프리미티브를 래스터화하고, 상기 제 1 프리미티브와 상이한 제 2 프리미티브에 대응하는 하나 이상의 소스 픽셀들을 생성하기 위해 상기 제 2 프리미티브를 래스터화하도록 구성된 래스터화 블록으로서, 상기 제 1 프리미티브에 대응하는 상기 하나 이상의 소스 픽셀들은 제 1 소스 픽셀을 포함하고, 상기 제 2 프리미티브에 대응하는 상기 하나 이상의 소스 픽셀들은 제 2 소스 픽셀을 포함하는, 상기 래스터화 블록; 및

빈 버퍼 (bin buffer)로부터 목적지 알파값을 추출하도록 구성된 목적지 알파 테스트 모듈로서, 상기 목적지 알파값은 상기 래스터화 블록에 의해 생성되고 상기 제 1 프리미티브와 연관된 상기 제 1 소스 픽셀을 프로세싱하는 것에 응답하여 생성되고, 상기 목적지 알파 테스트 모듈은 또한, 상기 목적지 알파값에 기초하여, 상기 래스터화 블록에 의해 생성된 상기 제 2 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하고, 조건이 만족되는지를 결정하기 위해 상기 목적지 알파값을 평가하여, 상기 조건이 만족되는 경우, 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼

로부터 추출되지 않게 하는 상기 액션을 수행하도록 구성되며, 상기 제 2 소스 픽셀은 상기 제 2 프리미티브와 연관되는, 상기 목적지 알파 테스트 모듈을 포함하고,

상기 목적지 알파 테스트 모듈은 픽셀 프로세싱 파이프라인에 포함되며, 상기 래스터화 블록 및 상기 픽셀 프로세싱 파이프라인은 상기 그래픽스 프로세싱 디바이스의 그래픽스 파이프라인에 포함되고, 상기 그래픽스 프로세싱 디바이스 및 상기 빈 버퍼는 제 1 마이크로칩 상에 상주하고, 상기 텍스처 버퍼는 상기 제 1 마이크로칩과 상이한 제 2 마이크로칩 상에 상주하는, 그래픽 프로세싱을 위한 디바이스.

청구항 18

제 17 항에 있어서,

상기 목적지 알파 테스트 모듈은 또한, 상기 제 2 소스 픽셀이 상기 목적지 알파값에 기초하여 상기 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에 상기 픽셀 프로세싱 파이프라인으로부터 상기 제 2 소스 픽셀을 폐기할지 여부를 결정하고, 그리고 상기 목적지 알파값에 기초하여 상기 제 2 소스 픽셀을 폐기하도록 결정하는 것에 응답하여 상기 제 2 소스 픽셀이 상기 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에 상기 픽셀 프로세싱 파이프라인으로부터 상기 제 2 소스 픽셀을 폐기하도록 구성되는, 그래픽스 프로세싱 디바이스.

청구항 19

제 18 항에 있어서,

상기 목적지 알파 테스트 모듈은 또한, 상기 목적지 알파값에 기초하여 상기 제 2 소스 픽셀을 폐기하지 않도록 결정하는 것에 응답하여 텍스처 맵핑 스테이지를 포함하는 하나 이상의 프로세싱 스테이지들의 세트에 상기 제 2 소스 픽셀을 패스하도록 구성되는, 그래픽스 프로세싱 디바이스.

청구항 20

제 17 항에 있어서,

상기 액션은 제 1 액션이고,

상기 목적지 알파 테스트 모듈은 또한, 상기 제 1 액션을 수행하지 않도록 결정하는 것에 응답하여 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되게 하는 제 2 액션을 수행하도록 구성되는, 그래픽스 프로세싱 디바이스.

청구항 21

제 17 항에 있어서,

상기 목적지 알파 테스트 모듈은 또한, 상기 목적지 알파값을 임계값과 비교하고, 그리고 상기 목적지 알파값을 상기 임계값과 비교하는 것에 의해 산출된 결과에 기초하여 상기 액션을 수행할지 여부를 결정하도록 구성되는, 그래픽스 프로세싱 디바이스.

청구항 22

제 21 항에 있어서,

상기 목적지 알파 테스트 모듈은 또한, 상기 목적지 알파값이 상기 임계값 이하인지 여부를 결정하도록 구성되는, 그래픽스 프로세싱 디바이스.

청구항 23

제 22 항에 있어서,

상기 목적지 알파 테스트 모듈은 또한, 상기 목적지 알파값이 상기 임계값 이하라고 결정하는 것에 응답하여 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 상기 액션을 수행하고, 그리고 상기 목적지 알파값이 상기 임계값 이하가 아니라고 결정하는 것에 응답하여 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되게 하는 액션을 수행하도록 구

성되는, 그래픽스 프로세싱 디바이스.

청구항 24

제 21 항에 있어서,

상기 임계값은 구성가능한 임계값인, 그래픽스 프로세싱 디바이스.

청구항 25

제 17 항에 있어서,

상기 목적지 알파 테스트 모듈은 또한, 제 1 통신 인터페이스를 통해 상기 빈 버퍼로부터 상기 목적지 알파값을 추출하도록 구성되고,

상기 그래픽스 프로세싱 디바이스는 상기 제 1 통신 인터페이스와 상이한 제 2 통신 인터페이스를 통해 상기 텍스처 버퍼로부터 상기 하나 이상의 텍스처값들을 추출하도록 구성되는, 그래픽스 프로세싱 디바이스.

청구항 26

제 25 항에 있어서,

상기 제 1 통신 인터페이스는 상기 제 2 통신 인터페이스의 대역폭보다 큰 대역폭을 갖는, 그래픽스 프로세싱 디바이스.

청구항 27

제 17 항에 있어서,

상기 빈 버퍼는 프레임 버퍼에 포함된 데이터의 서브세트를 저장하는, 그래픽스 프로세싱 디바이스.

청구항 28

제 17 항에 있어서,

그래픽스 이미지는 복수의 목적지 픽셀들을 포함하고, 상기 그래픽스 프로세싱 디바이스는 복수의 렌더링 패스들에서 상기 그래픽스 이미지를 렌더링하도록 구성되고, 상기 렌더링 패스들의 각각이 상기 목적지 픽셀들의 각 서브세트와 연관되고, 상기 빈 버퍼의 용량이 상기 그래픽스 이미지와 연관된 상기 복수의 목적지 픽셀들 모두에 대한 픽셀 데이터를 저장하기 위해 필요한 최소 용량보다 적으며, 그리고 렌더링 패스 각각에 대하여, 상기 빈 버퍼가 각각의 상기 렌더링 패스와 연관된 목적지 픽셀들의 서브세트에 대한 픽셀 데이터를 저장하는, 그래픽스 프로세싱 디바이스.

청구항 29

제 17 항에 있어서,

상기 목적지 알파 테스트 모듈은,

상기 목적지 알파값에 기초하여, 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하도록 구성된, 스텐실 테스트 블록 또는 z/스텐실 테스트 블록 중 하나를 포함하는, 그래픽스 프로세싱 디바이스.

청구항 30

제 17 항에 있어서,

상기 목적지 알파 테스트 모듈은,

상기 목적지 알파값에 기초하여, 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하는 셰이더 (shader) 프로그램을 실행하도록 구성된 셰이더 유닛을 포함하는, 그래픽스 프로세싱 디바이스.

청구항 31

제 17 항에 있어서,

상기 목적지 알파 테스트 모듈은 또한, 상기 목적지 알파값에 기초하여, 상기 제 2 소스 픽셀에 대응하는 목적지 픽셀이 상기 목적지 픽셀에 대응하는 하나 이상의 미리 프로세싱된 소스 픽셀들로 인해 불투명한지 여부를 결정하도록 구성되는, 그래픽 프로세싱 디바이스.

청구항 32

제 17 항에 있어서,

상기 제 2 소스 픽셀은 픽셀 로케이션과 연관되고, 상기 제 2 소스 픽셀에 대한 목적지 알파값은 상기 픽셀 로케이션과 연관된 목적지 픽셀의 불투명도를 표시하는 값이고, 상기 목적지 픽셀의 불투명도는 상기 픽셀 로케이션과 연관된 하나 이상의 미리 프로세싱된 소스 픽셀들에 의해 적어도 부분적으로 결정되는, 그래픽 프로세싱 디바이스.

청구항 33

그래픽 프로세싱을 위한 장치로서,

제 1 프리미티브에 대응하는 하나 이상의 소스 픽셀들을 생성하도록 상기 제 1 프리미티브를 래스터화하는 수단으로서, 상기 제 1 프리미티브에 대응하는 상기 하나 이상의 소스 픽셀들은 제 1 소스 픽셀을 포함하는, 상기 제 1 프리미티브를 래스터화하는 수단;

상기 제 1 프리미티브와 상이한 제 2 프리미티브에 대응하는 하나 이상의 소스 픽셀들을 생성하도록 상기 제 2 프리미티브를 래스터화하는 수단으로서, 상기 제 2 프리미티브에 대응하는 상기 하나 이상의 소스 픽셀들은 제 2 소스 픽셀을 포함하는, 상기 제 2 프리미티브를 래스터화하는 수단;

빈 버퍼 (bin buffer)로부터 목적지 알파값을 추출하는 수단으로서, 상기 목적지 알파값은 상기 제 1 프리미티브를 래스터화하는 수단에 의해 생성되고 상기 제 1 프리미티브와 연관된 상기 제 1 소스 픽셀을 프로세싱하는 것에 응답하여 생성되는, 상기 목적지 알파값을 추출하는 수단;

상기 목적지 알파값에 기초하여, 상기 제 2 프리미티브를 래스터화하는 수단에 의해 생성된 상기 제 2 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하는 수단으로서, 상기 제 2 소스 픽셀은 상기 제 2 프리미티브와 연관되고, 상기 제 1 프리미티브를 래스터화하는 수단 및 상기 빈 버퍼는 제 1 마이크로칩 상에 상주하고, 상기 텍스처 버퍼는 상기 제 1 마이크로칩과 상이한 제 2 마이크로 칩 상에 상주하는, 상기 액션을 수행할지 여부를 결정하는 수단; 및

조건이 만족되는지를 결정하기 위해 상기 목적지 알파값을 평가하여, 상기 조건이 만족되는 경우, 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 상기 액션을 수행하는 수단을 포함하는, 그래픽 프로세싱을 위한 장치.

청구항 34

제 33 항에 있어서,

상기 액션을 수행할지 여부를 결정하는 수단은, 상기 제 2 소스 픽셀이 상기 목적지 알파값에 기초하여 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에, 상기 픽셀 프로세싱 파이프라인으로부터 상기 제 2 소스 픽셀을 폐기할지 여부를 결정하는 수단을 포함하고,

상기 액션을 수행하는 수단은, 상기 목적지 알파값에 기초하여 상기 제 2 소스 픽셀을 폐기하도록 결정하는 것에 응답하여 상기 제 2 소스 픽셀이 상기 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에, 상기 픽셀 프로세싱 파이프라인으로부터 상기 제 2 소스 픽셀을 폐기하는 수단을 포함하는, 장치.

청구항 35

제 34 항에 있어서,

상기 목적지 알파값에 기초하여 상기 제 2 소스 픽셀을 폐기하지 않도록 결정하는 것에 응답하여 텍스처 맵핑 스테이지를 포함하는 하나 이상의 프로세싱 스테이지들의 세트로부터 상기 제 2 소스 픽셀을 패스하는 수단을 더 포함하는, 장치.

청구항 36

제 33 항에 있어서,

상기 제 2 소스 픽셀은 픽셀 로케이션과 연관되고, 상기 제 2 소스 픽셀에 대한 목적지 알파값은 상기 픽셀 로케이션과 연관된 목적지 픽셀의 불투명도를 표시하는 값이고, 상기 목적지 픽셀의 불투명도는 상기 픽셀 로케이션과 연관된 하나 이상의 미리 프로세싱된 소스 픽셀들에 의해 적어도 부분적으로 결정되는, 장치.

청구항 37

제 33 항에 있어서,

상기 빈 버퍼로부터 상기 목적지 알파값을 추출하는 수단은, 제 1 통신 인터페이스를 통해 상기 빈 버퍼로부터 상기 목적지 알파값을 추출하는 수단을 포함하고,

상기 장치는, 상기 제 1 통신 인터페이스와 상이한 제 2 통신 인터페이스를 통해 상기 텍스처 버퍼로부터 상기 하나 이상의 텍스처값들을 추출하는 수단을 더 포함하는, 장치.

청구항 38

제 37 항에 있어서,

상기 제 1 통신 인터페이스는 상기 제 2 통신 인터페이스의 대역폭보다 큰 대역폭을 갖는, 장치.

청구항 39

그래픽 프로세싱을 위한 비일시적 컴퓨터 판독가능 매체로서,

하나 이상의 프로세서들로 하여금,

그래픽스 파이프라인에 포함된 래스터화 블록에 의해, 제 1 프리미티브에 대응하는 하나 이상의 소스 픽셀들을 생성하도록 상기 제 1 프리미티브를 래스터화하게 하는 것으로서, 상기 제 1 프리미티브에 대응하는 상기 하나 이상의 소스 픽셀들은 제 1 소스 픽셀을 포함하는, 상기 제 1 프리미티브를 래스터화하게 하고;

상기 래스터화 블록에 의해, 상기 제 1 프리미티브와 상이한 제 2 프리미티브에 대응하는 하나 이상의 소스 픽셀들을 생성하도록 상기 제 2 프리미티브를 래스터화하게 하는 것으로서, 상기 제 2 프리미티브에 대응하는 상기 하나 이상의 소스 픽셀들은 제 2 소스 픽셀을 포함하는, 상기 제 2 프리미티브를 래스터화하게 하고;

빈 버퍼 (bin buffer) 로부터 목적지 알파값을 추출하게 하는 것으로서, 상기 목적지 알파값은 상기 래스터화 블록에 의해 생성되고 상기 제 1 프리미티브와 연관된 상기 제 1 소스 픽셀을 프로세싱하는 것에 응답하여 생성되는, 상기 목적지 알파값을 추출하게 하고;

상기 목적지 알파값에 기초하여, 상기 래스터화 블록에 의해 생성된 상기 제 2 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하게 하는 것으로서, 상기 제 2 소스 픽셀은 상기 제 2 프리미티브와 연관되고, 상기 그래픽스 파이프라인 및 상기 빈 버퍼는 제 1 마이크로칩 상에 상주하고, 상기 텍스처 버퍼는 상기 제 1 마이크로칩과 상이한 제 2 마이크로칩 상에 상주하는, 상기 액션을 수행할지 여부를 결정하게 하고; 및

조건이 만족되는지를 결정하기 위해 상기 목적지 알파값을 평가하여, 상기 조건이 만족되는 경우, 상기 제 2 소스 픽셀에 대한 상기 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 상기 액션을 수행하게 하는,

명령들을 포함하는, 그래픽 프로세싱을 위한 비일시적 컴퓨터 판독가능 매체.

청구항 40

제 39 항에 있어서,

상기 하나 이상의 프로세서들로 하여금 상기 액션을 수행할지 여부를 결정하게 하는 명령들은, 상기 하나 이상의 프로세서들로 하여금 상기 제 2 소스 픽셀이 상기 목적지 알파값에 기초하여 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에, 상기 픽셀 프로세싱 파이프라인으로부터 상기 제 2 소스 픽셀을 폐기할지

여부를 결정하게 하는 명령들을 포함하고,

상기 하나 이상의 프로세서들로 하여금 상기 액션을 수행하게 하는 명령들은, 상기 하나 이상의 프로세서들로 하여금 상기 목적지 알파값에 기초하여 상기 제 2 소스 픽셀을 폐기하도록 결정하는 것에 응답하여 상기 제 2 소스 픽셀이 상기 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에, 상기 픽셀 프로세싱 파이프라인으로부터 상기 제 2 소스 픽셀을 폐기하게 하는 명령들을 포함하는, 비일시적 컴퓨터 판독가능 매체.

청구항 41

제 40 항에 있어서,

상기 하나 이상의 프로세서들로 하여금,

상기 목적지 알파값에 기초하여 상기 제 2 소스 픽셀을 폐기하지 않도록 결정하는 것에 응답하여 텍스처 맵핑 스테이지를 포함하는 하나 이상의 프로세싱 스테이지들의 세트로 상기 제 2 소스 픽셀을 패스하게 하는 명령들을 더 포함하는, 비일시적 컴퓨터 판독가능 매체.

청구항 42

제 39 항에 있어서,

상기 제 2 소스 픽셀은 픽셀 로케이션과 연관되고, 상기 제 2 소스 픽셀에 대한 목적지 알파값은 상기 픽셀 로케이션과 연관된 목적지 픽셀의 불투명도를 표시하는 값이고, 상기 목적지 픽셀의 불투명도는 상기 픽셀 로케이션과 연관된 하나 이상의 미리 프로세싱된 소스 픽셀들에 의해 적어도 부분적으로 결정되는, 비일시적 컴퓨터 판독가능 매체.

청구항 43

제 39 항에 있어서,

상기 하나 이상의 프로세서들로 하여금 상기 빈 버퍼로부터 상기 목적지 알파값을 추출하게 하는 명령들은, 상기 하나 이상의 프로세서들로 하여금 제 1 통신 인터페이스를 통해 상기 빈 버퍼로부터 상기 목적지 알파값을 추출하게 하는 명령들을 포함하고,

상기 하나 이상의 프로세서들은 상기 제 1 통신 인터페이스와 상이한 제 2 통신 인터페이스를 통해 상기 텍스처 버퍼로부터 상기 하나 이상의 텍스처값들을 추출하도록 구성되는, 비일시적 컴퓨터 판독가능 매체.

청구항 44

제 43 항에 있어서,

상기 제 1 통신 인터페이스는 상기 제 2 통신 인터페이스의 대역폭보다 큰 대역폭을 갖는, 비일시적 컴퓨터 판독가능 매체.

청구항 45

삭제

청구항 46

삭제

발명의 설명

기술 분야

[0001] 본 개시물은 그래픽스 프로세싱 시스템들에 관한 것이고, 보다 상세하게는 그래픽스 프로세싱 시스템에서 픽셀들을 프로세싱하는 것에 관한 것이다.

배경 기술

[0002] 컴퓨팅 디바이스들을 디스플레이 상으로 그래픽스 데이터의 렌더링을 가속화하기 위해 그래픽스 프로세싱 유닛

(GPU) 을 종종 사용한다. 그러한 컴퓨팅 디바이스들은, 예를 들어 컴퓨터 워크스테이션들, 이동 전화기들, 임베딩된 시스템들, 개인용 컴퓨터들 및 비디오 게임 콘솔들을 포함할 수도 있다. GPU들은 또한 윈도우 컴포지션 (window composition) 동작들을 수행하기 위해 윈도우들 기반 오퍼레이팅 시스템들에 의해 사용된다.

윈도우들 기반 오퍼레이팅 시스템은, 각각이 그 자신의 어플리케이션 윈도우를 운영할 수도 있는, 몇몇 상이한 오픈 어플리케이션 프로그램들에 의해 사용자가 작업하게 하고, 그 프로그램들 사이에서 스위칭하게 하는, 윈도우 시스템을 포함하는 그래픽 사용자 인터페이스 (GUI) 를 사용한다. 그러한 시스템에서, 개별 윈도우들은 다양한 중첩 및/또는 비중첩 구성들로 사용자에게 의해 포지셔닝되고 재사이징될 수도 있다.

[0003]

윈도우 관리자는 윈도우들 기반 오퍼레이팅 시스템에서 윈도우들의 배치 및 외관의 양자를 제어하기 위해 사용되는 소프트웨어 프로그램의 유형이다. 윈도우 관리자는 또한 GPU 와 상호작용하여 윈도우들로 하여금 스크린으로 드로잉되게 하고 윈도우들 중 어느 하나의 포지션, 사이즈 또는 외관의 변화에 응답하거나 또는 주기적으로 윈도우들을 리프레시하게 한다. 각각의 어플리케이션은 통상 시스템 메모리에서의 표면과 같은 어플리케이션의 윈도우의 현재의 외관의 표현을 저장한다. 표면은 픽셀들의 어레이에 대한 컬러 데이터의 수집을 지칭할 수도 있다. 디스플레이를 리프레시하기 위해서, 윈도우 관리자는 통상 스크린 상으로 모든 오픈 어플리케이션 윈도우들과 연관된 모든 표면들을 재드로잉한다. 주어진 시점에서 디스플레이 상에 디스플레이되는 전체 표면들의 세트가 서로 중첩하는 표면들을 포함할 수도 있기 때문에, 윈도우들 관리자는 통상 표면 컴포지션 동작을 수행하며, 이는 다중의 독립 표면들을 디스플레이 상의 표현을 위한 단일 스크린 표면으로 컨버팅한다. 표면 컴포지션을 수행하기 위한 하나의 기법은, 예를 들어 페인터의 알고리즘에 따라, 후방-전방 순서로 스크린 상으로 표면들을 드로잉하는 것을 수반한다. 이러한 방식으로, 2 개의 표면들이 중첩 부분들을 포함한다면, 최전방 표면의 중첩 부분들은 최후방 표면의 중첩 부분들의 상부 상에 드로잉되게 되며, 이로써 뷰어가 보고자 할 때, 제 1 표면의 중첩 부분들은 가시적이도록 하고, 제 2 표면의 중첩 부분들은 가시적이지 않도록 한다. 따라서, 윈도우 관리자는 가시적이든 가시적이지 않든, 특정 순서로, 각각의 어플리케이션 표면을 스크린으로 드로잉함으로써 표면 컴포지션을 수행할 수도 있다.

[0004]

각각의 어플리케이션 표면을 드로잉하기 위하여, 윈도우 관리자는 GPU 로 하여금 표면과 연관된 하나 이상의 프리미티브들을 렌더링하게 하고, 그 프리미티브들에 텍스처들의 다양한 조합들을 적용하게 하여 표면들에 원하는 외관을 부여할 수도 있다. 프리미티브들의 각각에 적용된 텍스처들은 프리미티브와 연관된 어플리케이션 표면에 함유된 컬러값들에 대응하거나 및/또는 그 컬러값들로부터 도출된 텍스처값들에 의해 정의될 수도 있다. 게다가, 윈도우 관리자는 특정 시각 효과들, 예를 들어, 음영들, 보더들, 투명도, 조광 등을 달성하기 위해 텍스처값들을 변경하고 및/또는 부가적인 텍스처들을 적용할 수도 있다. 프리미티브에 적용되는 각각의 텍스처에 대해, GPU 는 통상 오프 칩 메모리로부터 프리미티브와 연관된 픽셀 마다 텍스처값들에 액세스하며, 이는 많은 양의 메모리 대역폭 사용을 초래할 수도 있다.

발명의 내용

해결하려는 과제

[0005]

본 개시물은 목적지 알파값들에 기초한 그래픽스 프로세싱 시스템에서 메모리 액세스 대역폭을 감소시키기 위한 기법들을 기재한다. 본 개시물의 기법들에 따라 개별 소스 픽셀을 프로세싱할 때, 목적지 알파 테스트 모듈은 빈 버퍼 (bin buffer) 로부터 소스 픽셀과 연관된 목적지 알파값을 추출하고, 목적지 알파값에 기초하여 소스 픽셀에 대한 텍스처값들이 텍스처 버퍼로부터 추출되게 할지 여부를 결정할 수도 있다. 일부 예들에서, 목적지 알파값은 소스 픽셀과 동일한 스크린 로케이션을 갖는 목적지 픽셀의 용량 레벨을 표시할 수도 있다. 대응하는 목적지 알파값들에 기초하여 소스 픽셀들에 대한 텍스처값들을 선택적으로 추출함으로써, 본 개시물의 기법들은 종래의 그래픽스 시스템에서 텍스처 버퍼에 관하여 통상 발생하는 소스 판독들의 수와 비교할 때 텍스처 버퍼에 관하여 발생하여야 하는 소스 판독들의 수를 효과적으로 감소시킬 수도 있다. 이러한 방식으로, 그래픽스 프로세싱 시스템에 대한 메모리 액세스 대역폭이 감소될 수도 있다.

과제의 해결 수단

[0006]

일 예에서, 본 개시물은, 그래픽스 프로세서에 의해 빈 버퍼 (bin buffer) 로부터 목적지 알파값을 추출하는 단계를 포함하는 방법을 기재한다. 목적지 알파값은 제 1 프리미티브와 연관된 제 1 픽셀을 프로세싱하는 것에 응답하여 생성된다. 방법은, 그래픽스 프로세서에 의해 그리고 목적지 알파값에 기초하여, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하는 단계를 더 포함한다. 제 2 픽셀은 제 1 프리미티브와 상이한 제 2 프리미티브와 연관된다. 방법은, 그래픽

스 프로세서에 의해, 액션을 수행하도록 결정하는 것에 응답하여 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행하는 단계를 더 포함한다.

[0007] 또 다른 예에서, 본 개시물은, 빈 버퍼로부터 목적지 알파값을 추출하도록 구성된 목적지 알파 테스트 모듈을 포함하는 그래픽스 프로세싱 디바이스를 기재한다. 목적지 알파값은 제 1 프리미티브와 연관된 제 1 픽셀을 프로세싱하는 것에 응답하여 생성된다. 목적지 알파 테스트 모듈은 또한, 목적지 알파값에 기초하여, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하고, 그리고 액션을 수행하도록 결정하는 것에 응답하여 제 2 픽셀에 대한 하나 이상의 텍스처값들이 상기 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행하도록 구성된다. 제 2 픽셀은 제 1 프리미티브와 상이한 제 2 프리미티브와 연관된다.

[0008] 또 다른 예에서, 본 개시물은, 빈 버퍼로부터 목적지 알파값을 추출하는 수단을 포함하는 장치를 기재한다. 목적지 알파값은 제 1 프리미티브와 연관된 제 1 픽셀을 프로세싱하는 것에 응답하여 생성된다. 장치는, 목적지 알파값에 기초하여, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하는 수단을 더 포함한다. 제 2 픽셀은 제 1 프리미티브와 상이한 제 2 프리미티브와 연관된다. 장치는, 액션을 수행하도록 결정하는 것에 응답하여 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행하는 수단을 더 포함한다.

[0009] 또 다른 예에서, 본 개시물은, 하나 이상의 프로세서들로 하여금, 빈 버퍼로부터 목적지 알파값을 추출하게 하는 명령들을 포함하는 컴퓨터 판독가능 매체를 기재한다. 목적지 알파값은 제 1 프리미티브와 연관된 제 1 픽셀을 프로세싱하는 것에 응답하여 생성된다. 컴퓨터 판독가능 매체는, 하나 이상의 프로세서들로 하여금, 목적지 알파값에 기초하여, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하게 하게 하는 명령들을 더 포함한다. 제 2 픽셀은 제 1 프리미티브와 상이한 제 2 프리미티브와 연관된다. 컴퓨터 판독가능 매체는, 액션을 수행하도록 결정하는 것에 응답하여 하나 이상의 프로세서들로 하여금 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행하게 하는 명령들을 더 포함한다.

[0010] 하나 이상의 예들의 상세들은 첨부 도면들 및 하기의 상세한 설명에서 기술된다. 다른 특징들, 목적들, 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명백해질 것이다.

도면의 간단한 설명

[0011] 도 1 은 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들을 구현하는데 사용될 수도 있는 일 예의 컴퓨팅 디바이스를 도시하는 블록 다이어그램이다.

도 2 는 도 1 의 컴퓨팅 디바이스의 CPU, GPU, 시스템 메모리를 더욱 상세하게 도시하는 블록 다이어그램이다.

도 3 은 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들을 구현하는데 사용될 수도 있는 일 예의 픽셀 프로세싱 파이프라인을 도시하는 블록 다이어그램이다.

도 4 는 도 3 의 픽셀 프로세싱 파이프라인을 구현하는데 사용될 수도 있는 일 예의 픽셀 프로세싱 파이프라인을 도시하는 블록 다이어그램이다.

도 5 는 본 개시물의 기법들을 구현하는데 사용될 수도 있는 일 예의 목적지 알파 테스트 모듈을 도시하는 블록 다이어그램이다.

도 6 은 본 개시물의 기법들을 구현하는데 사용될 수도 있는 또 다른 예의 목적지 알파 테스트 모듈을 도시하는 블록 다이어그램이다.

도 7 은 목적지 알파 테스트를 수행하고 본 개시물의 선택적 텍스처값 추출 기법들을 구현하기 위해 구성가능한 스텐실 테스트 블록을 사용하는 일 예의 픽셀 프로세싱 파이프라인을 도시하는 블록 다이어그램이다.

도 8 은 도 7 의 픽셀 프로세싱 파이프라인을 구현하는데 사용될 수도 있는 일 예의 구성가능한 스텐실 테스트 블록을 도시하는 블록 다이어그램이다.

도 9 는 본 개시물의 선택적 텍스처값 추출 기법들에 따른 목적지 알파 테스트를 수행하기 위해 구성가능한 z/스텐실 테스트 블록을 사용하는 일 예의 픽셀 프로세싱 파이프라인을 도시하는 블록 다이어그램이다.

도 10 은 도 9 의 픽셀 프로세싱 파이프라인을 구현하는데 사용될 수도 있는 일 예의 구성가능한 z/스텐실 테스트

팅 블록을 도시하는 블록 다이어그램이다.

도 11 은 본 개시물의 선택적 텍스처값 추출 기법들에 따른 목적지 알파 테스트를 수행하기 위해 프로그램가능한 셰이더 (shader) 유닛을 사용하는 일 예의 픽셀 프로세싱 파이프라인을 도시하는 블록 다이어그램이다.

도 12 는 본 개시물의 선택적 텍스처값 추출 기법들에 따른 텍스처 맵핑 및 목적지 알파 테스트를 수행하기 위해 프로그램가능한 셰이더 유닛을 사용하는 다른 예의 픽셀 프로세싱 파이프라인을 도시하는 블록 다이어그램이다.

도 13 은 본 개시물에 따른 목적지 알파값에 기초하여 소스 픽셀에 대한 텍스처값들을 선택적으로 추출하기 위한 일 예의 기법을 도시하는 플로우 다이어그램이다.

도 14 는 도 13 에 나타난 기법을 구현하는데 사용될 수도 있는 목적지 알파 테스트를 수행하기 위한 일 예의 기법을 도시하는 플로우 다이어그램이다.

도 15 는 도 14 에 나타난 기법을 구현하는데 사용될 수도 있는 목적지 알파값을 임계와 비교하기 위한 일 예의 기법을 도시하는 플로우 다이어그램이다.

도 16 은 본 개시물에 따른 목적지 알파값 기반 선택적 텍스처값 추출 렌더링 모드를 자동으로 인에이블시키기 위한 일 예의 기법을 도시하는 플로우 다이어그램이다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 개시물은 목적지 알파값에 기초한 그래픽스 프로세싱 시스템에서 메모리 액세스 대역폭을 감소시키기 위한 기법들을 개시한다. 본 개시물의 기법들에 따른 개별 소스 픽셀을 프로세싱할 때, 목적지 알파 테스트 모듈은 소스 픽셀과 연관된 목적지 알파값에 기초하여 소스 픽셀에 대한 텍스처값들이 버퍼로부터 추출되게 하는지 여부를 결정할 수도 있다. 일부 예들에서, 목적지 알파값은 소스 픽셀과 동일한 스크린 로케이션을 갖는 목적지 픽셀의 불투명도의 레벨을 표시할 수도 있다. 그 대응 목적지 알파값에 기초하여 소스 픽셀들에 대한 텍스처값들을 선택적으로 추출함으로써, 본 개시물의 기법들은 종래 그래픽스 시스템에서 텍스처 버퍼에 관하여 통상 발생하는 소스 판독들의 수와 비교할 때, 텍스처 버퍼에 관하여 발생하여야 하는 소스 판독들의 수를 효과적으로 감소시킬 수도 있다. 이러한 방식으로, 그래픽스 프로세싱 시스템에 대한 메모리 액세스 대역폭이 감소될 수도 있다.

[0013] 본 개시물의 기법들은 윈도우들 컴포지션이, 예를 들어 역 페인터의 알고리즘에 따라, 전방-후방 방식으로 수행되는 그래픽스 프로세싱 환경에 특히 유용할 수도 있다. 그러한 환경에서, 목적지 픽셀이 하나 이상의 이전에 프로세싱된 표면들로 인하여 이미 불투명하면, 그 동일한 픽셀 로케이션에 대하여 프로세싱되는 임의의 후속 소스 픽셀들이 목적지 픽셀의 전체 외관에 기여하지 않게 되는데, 이는 전방-후방 렌더링으로 인하여, 그러한 소스 픽셀들이 그 로케이션에 대해 이전에 프로세싱된 픽셀들 뒤에 위치되고, 이에 따라 가시적이지 않기 때문이다. 이로써, 그 픽셀 로케이션에 대한 임의의 후속 소스 픽셀들은 렌더링된 이미지의 외관에 영향을 미치지 않으면서 안전하게 폐기될 수도 있다. 이에 따라, 전방-후방 컴포지팅 환경에서 본 개시물의 기법들을 이용함으로써, 본 개시물의 기법들이 사용되어, 일부 예들에서, 결과의 렌더링된 이미지의 품질을 훼손하지 않으면서 그래픽스 프로세싱 시스템에 대한 메모리 액세스 대역폭을 감소시킬 수도 있다.

[0014] 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들은 빈 렌더링의 컨텍스트, 예를 들어 그래픽 이미지가 복수의 작은 부분들 (예를 들어, 목적지 픽셀들의 서브세트, 픽셀들의 타이들, 또는 빈들) 로 세분화되고, 이미지의 각 부분이 별도의 렌더링 패스로서 렌더링되는 렌더링 컨텍스트에 특히 유용할 수도 있다. 그러한 렌더링 컨텍스트에서, 특정 빈에 대한 목적지 알파값은, 그러한 값들이 목적지 알파 테스트 모듈에 의해 필요할 때, 빈 버퍼를 통해 액세스가능할 수도 있다. 이것은 빈 버퍼보다 낮은 유효 대역폭을 갖는 메모리에 로케이팅될 수도 있는, 프레임 버퍼로부터 목적지 알파값들에 액세스할 필요성을 완화시킨다. 이에 따라, 목적지 알파값들이 빈 버퍼로부터 추출되는 비닝 (binning) 컨텍스트에서 본 개시물의 기법들을 구현함으로써, 목적지 알파값들을 추출해야 할 수도 있는 오버헤드 대역폭이 텍스처 버퍼에 대한 소스 판독들의 수를 감소시킴으로써 달성되는 대역폭 절약들에 비해 상대적으로 최소가 될 수도 있다.

[0015] z 버퍼링 기법들은 동일한 픽셀 로케이션에서 다른 소스 픽셀들 뒤에 있는 소스 픽셀들을 폐기함으로써 컴포지팅 대역폭을 감소시키기 위해 사용될 수도 있지만, 그러한 기법들은 픽셀들이 픽셀당 소스 알파값들을 가질 때, 표면들을 적절하게 렌더링하지 않으며, 이는 본질적으로 중첩 픽셀들이 반투명이 되게 한다. 하지만, 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들은 모든 표면들이 픽셀당 소스 알파값들을 가질 때에도

컴포지팅 대역폭을 감소시킴으로써, 보다 강건한 대역폭 감소 메커니즘을 제공할 수도 있다.

- [0016] 본 개시물의 선택적 텍스처값 추출 기법들을 구현하는데 사용되는 목적지 알파 테스트 모듈은 다양한 방식으로 구현될 수도 있다. 예를 들어, 목적지 알파 테스트 모듈은 전용된 고정 함수 하드웨어 블록으로서, 셰이더 유닛 상에서 실행하는 셰이더 프로그램으로서, 목적지 알파 테스트를 수행하도록 구성된 구성가능한 스텐실 테스트 블록으로서, 및/또는 목적지 알파 테스트를 수행하도록 구성된 구성가능한 z/스텐실 테스트 블록으로서 구현될 수도 있다. 이러한 다양한 구성들은 본 개시물에서 이후 더 상세하게 기재될 것이다.
- [0017] 도 1 은 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들을 구현하는데 사용될 수도 있는 일 예의 컴퓨팅 디바이스 (2) 를 도시하는 블록 다이어그램이다. 컴퓨팅 디바이스 (2) 는 예를 들어, 개인용 컴퓨터, 데스크탑 컴퓨터, 랩탑 컴퓨터, 컴퓨터 워크스테이션, 비디오 게임 플랫폼 또는 콘솔, 예를 들어 셀룰러 또는 위성 전화기와 같은 이동 전화기, 랜드라인 전화기, 인터넷 전화기, 핸드헬드 디바이스, 예컨대 포터블 비디오 게임 디바이스 또는 개인 휴대정보 단말기 (PDA), 개인 뮤직 플레이어, 비디오 플레이어, 디스플레이 디바이스, 텔레비전, 텔레비전 셋톱 박스, 서버, 중계 네트워크 디바이스, 메인프레임 컴퓨터 또는 그래픽 데이터 를 프로세싱 및/또는 디스플레이하는 임의의 다른 유형의 디바이스를 포함할 수도 있다.
- [0018] 도 1 의 예에 도시된 바와 같이, 컴퓨팅 디바이스 (2) 는 사용자 입력 인터페이스 (4), CPU (6), 메모리 제어기 (8), 시스템 메모리 (10), 그래픽스 프로세싱 유닛 (GPU)(12), GPU 캐시 (14), 디스플레이 인터페이스 (16), 디스플레이 (18) 및 버스들 (20, 22) 을 포함한다. 사용자 입력 인터페이스 (4), CPU (6), 메모리 제어기 (8), GPU (12) 및 디스플레이 인터페이스 (16) 는 버스 (20) 를 사용하여 서로 통신할 수도 있다. 메모리 제어기 (8) 및 시스템 메모리 (10) 는 또한 버스 (22) 를 사용하여 서로 통신할 수도 있다. 버스들 (20, 22) 은 제 3 세대 버스 (예를 들어, HyperTransport 버스 또는 InfiniBand 버스), 제 2 세대 버스 (예를 들어, 어드밴스드 그래픽스 포트 버스, PCI (Peripheral Component Interconnect) 익스프레스 버스, 또는 AXI (Advanced eXtensible Interface) 버스) 또는 버스 또는 디바이스 상호접속의 다른 유형과 같은, 다양한 버스 구조들 중 어느 하나일 수도 있다. 도 1 에 도시된 상이한 컴포넌트들 사이의 통신 인터페이스들 및 버스들의 특정 구성은 단지 예시일 뿐이고, 동일하거나 상이한 컴포넌트들을 갖는 컴퓨팅 디바이스들의 다른 구성들 및/또는 다른 그래픽스 프로세싱 시스템들이 본 개시물의 기법들을 구현하는데 사용될 수도 있음을 유념해야 한다.
- [0019] CPU (6) 는 컴퓨팅 디바이스 (2) 의 동작을 제어하는 범용 또는 특수 목적 프로세서일 수도 있다. 사용자는 CPU (6) 로 하여금 하나 이상의 소프트웨어 어플리케이션들을 실행하게 하기 위한 입력을 컴퓨팅 디바이스 (2) 에 제공한다. CPU (6) 상에서 실행하는 소프트웨어 어플리케이션들은, 예를 들어 오퍼레이팅 시스템, 워드 프로세서 어플리케이션, 이메일 어플리케이션, 스프레드 시트 어플리케이션, 미디어 플레이어 어플리케이션, 비디오 게임 어플리케이션, 그래픽 사용자 인터페이스 어플리케이션 또는 다른 프로그램을 포함할 수도 있다. 사용자는 키보드, 마우스, 마이크론, 터치 패드 또는 사용자 입력 인터페이스 (4) 를 통해 컴퓨팅 디바이스 (2) 에 커플링되는 다른 입력 디바이스와 같은 하나 이상의 입력 디바이스들 (도시되지 않음) 을 통해 컴퓨팅 디바이스 (2) 에 입력을 제공할 수도 있다.
- [0020] CPU (6) 상에서 실행하는 소프트웨어 어플리케이션들은 디스플레이 (18) 에 그래픽스 데이터의 렌더링을 야기시키도록 CPU (6) 에 명령하는 하나 이상의 그래픽스 렌더링 명령들을 포함할 수도 있다. 일부 예들에서, 소프트웨어 명령들은 그래픽스 어플리케이션 프로그래밍 인터페이스 (API), 예컨대 오픈 그래픽스 라이브러리 (OpenGL[®]) API, 오픈 그래픽스 라이브러리 임베딩된 시스템들 (OpenGL ES) API, Direct3D API, X3D API, a RenderMan API, WebGL API, 또는 임의의 다른 공중 또는 사설 표준 그래픽스 API에 따를 수도 있다. 그래픽스 렌더링 명령들을 프로세싱하기 위해서, CPU (6) 는 GPU (12) 로 하여금 그래픽스 데이터의 렌더링의 일부 또는 전부를 수행하게 하는 하나 이상의 그래픽스 렌더링 커맨드들을 GPU (12) 에 발행할 수도 있다. 일부 예들에서, 렌더링되는 그래픽스 데이터는 그래픽스 프리미티브들의 리스트, 예를 들어 포인트들, 라인들, 삼각형들, 쿼드라테럴(quadralateral)들, 삼각형 스트림들 등을 포함할 수도 있다.
- [0021] 메모리 제어기 (8) 는 시스템 메모리 (10) 로 들어가고 이 메모리로부터 나오는 데이터의 전달을 용이하게 한다. 예를 들어, 메모리 제어기 (8) 는 메모리 판독 및 기록 커맨드들을 수신하고, 컴퓨팅 디바이스 (2) 에서 컴포넌트들에 메모리 서비스들을 제공하기 위해서 메모리 시스템 (10) 에 관한 그러한 커맨드들을 서비스할 수도 있다. 메모리 제어기 (8) 는 메모리 버스 (22) 를 통해 시스템 메모리 (10) 에 통신으로 커플링된다. 도 1 에서는 메모리 제어기 (8) 가 CPU (6) 및 시스템 메모리 (10) 의 양자와 이격되는 프로세싱 모듈인 것으로서 도시되지만, 다른 예들에서, 메모리 제어기 (8) 의 기능성의 일부 또는 전부가 CPU (6) 및 시스템

메모리 (10) 의 하나 또는 양자 상에서 구현될 수도 있다.

[0022] 시스템 메모리 (10) 는 CPU (6) 에 의한 실행을 위해 액세스 가능한 프로그램 모듈들 및/또는 명령들 및/또는 CPU (6) 상에서 실행하는 프로그램들에 의한 사용을 위한 데이터를 저장할 수도 있다. 예를 들어, 시스템 메모리 (10) 는 디스플레이 (18) 상에 그래픽 사용자 인터페이스 (GUI) 를 제시하기 위해 CPU (6) 에 의해 사용되는 윈도우 관리자 어플리케이션을 저장할 수도 있다. 게다가, 시스템 메모리 (10) 는 사용자 어플리케이션들 및 이 어플리케이션들과 연관된 어플리케이션 표면 데이터를 저장할 수도 있다. 시스템 메모리 (10) 는 컴퓨팅 디바이스 (2) 의 다른 컴포넌트들에 의한 사용을 위한 및/또는 이 컴포넌트들에 의해 생성된 정보를 부가적으로 저장할 수도 있다. 예를 들어, 시스템 메모리 (10) 는 GPU (12) 에 대한 디바이스 메모리로서 작용할 수도 있고, GPU (12) 에 의해 수행된 동작들로부터 유발되는 데이터 뿐만 아니라 GPU (12) 에 의해 동작되는 데이터를 저장할 수도 있다. 예를 들어, 시스템 메모리 (10) 는 텍스처 버퍼들, 깊이 버퍼들, 스텐실 버퍼들, 버텍스 버퍼들, 프레임 버퍼들 등의 임의의 조합을 저장할 수도 있다. 시스템 메모리 (10) 는, 예를 들어 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 리드 온니 메모리 (ROM), 소거가능 프로그램가능 ROM (EPROM), 전기적 소거가능 프로그램가능 ROM (EEPROM), 플래시 메모리, 자기 데이터 매체들 또는 광학 저장 매체들과 같은, 하나 이상의 휘발성 또는 비휘발성 메모리들 또는 저장 디바이스들을 포함할 수도 있다.

[0023] GPU (12) 는 하나 이상의 그래픽스 프리미티브들을 디스플레이 (18) 에 렌더링하기 위해 그래픽스 동작들을 수행하도록 구성될 수도 있다. 이에 따라, CPU (6) 상에서 실행하는 소프트웨어 어플리케이션들 중 하나가 그래픽스 프로세싱을 필요로 할 때, CPU (6) 는 디스플레이 (18) 로의 렌더링을 위해 그래픽스 커맨드들 및 그래픽스 데이터를 GPU (12) 에 제공할 수도 있다. 그래픽스 데이터는, 예를 들어, 드로잉 커맨드들, 상태 정보, 프리미티브 정보, 텍스처 정보 등을 포함할 수도 있다. GPU (12) 는, 일부 경우들에서, CPU (6) 보다 복잡한 그래픽 관련 동작들의 보다 효율적인 프로세싱을 제공하는 고도 병렬 구조로 내장될 수도 있다. 예를 들어, GPU (12) 는 병렬 방식으로 다중 버텍스들 또는 픽셀들 상에서 동작하도록 구성되는 복수의 프로세싱 엘리먼트들을 포함할 수도 있다. GPU (12) 의 고도 병렬 성질은, 일부 경우들에서, GPU (12) 가 CPU (6) 를 사용하여 디스플레이 (18) 에 장면들을 직접 드로잉하는 것보다 더 빨리 디스플레이 (18) 상으로 그래픽스 이미지들 (예를 들어, GUI들 및 2 차원 (2D) 및/또는 3 차원 (3D) 그래픽스 장면들) 을 드로잉하게 할 수도 있다.

[0024] 본 개시물에 따라, GPU (12) 는 소스 픽셀과 연관된 목적지 알파값에 기초하여 소스 픽셀에 대한 텍스처값들을 선택적으로 추출하도록 구성될 수도 있다. 예를 들어, GPU (12) 는 제 1 래스터화된 프리미티브와 연관된 제 1 픽셀을 프로세싱하는 것에 응답하여 목적지 알파값을 생성하고, 빈 버퍼에 목적지 알파값을 저장하고, 제 1 픽셀과 상이한 제 2 픽셀을 프로세싱하는 것에 응답하여 빈 버퍼로부터 목적지 알파값을 추출하고, 목적지 알파값에 기초하여, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하고, 그리고 액션을 수행하도록 결정하는 것에 응답하여 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션을 수행할 수도 있다. 제 2 픽셀은 제 1 래스터화된 (rasterized) 프리미티브와 상이한 제 2 래스터화된 프리미티브와 연관될 수도 있다. 게다가, 제 2 픽셀은 제 1 픽셀과 동일한 픽셀 로케이션을 가질 수도 있다 (예를 들어, 제 1 및 제 2 픽셀들이 스크린 스페이스에서 동일한 포지션 좌표들을 가질 수도 있다). 그 대응하는 목적지 알파값들에 기초하여 소스 픽셀들에 대한 텍스처값들을 선택적으로 추출함으로써, GPU (12) 는 텍스처값들에 관하여 발생하여야 하는 소스 판독들의 수를 효과적으로 감소시킴으로써, 그래픽스 이미지를 렌더링하기 위해 GPU (12) 에 의해 필요한 메모리 액세스 대역폭을 감소시킬 수도 있다.

[0025] 일부 예들에서, GPU (12) 는 목적지 알파값을 임계값과 비교하고, 목적지 알파값을 임계값과 비교함으로써 산출된 결과에 기초하여 액션을 수행할지 여부를 결정함으로써, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 적어도 부분적으로 텍스처 버퍼로부터 추출되어야 하는지 여부를 결정할 수도 있다. 일부 구현들에서, GPU (12) 는 목적지 알파값이 임계값 이하인지 여부를 결정함으로써 적어도 부분적으로 목적지 알파값을 임계값과 비교할 수도 있다. 그러한 예들에서, GPU (12) 는 목적지 알파값이 임계값 이하라고 결정하는 것에 응답하여 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행하고, 목적지 알파값이 임계값 이하가 아니라고 결정하는 것에 응답하여 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되게 하는 액션을 수행할 수도 있다. 임계값은, 예를 들어, 임계 레벨에서 또는 그 아래에서 목적지 픽셀 뒤에 위치되는 픽셀들이 가시적이지 않게 되고 및/또는 디스플레이되지 않아야 하는 그 목적지 픽셀에 대한 불투명도의 임계 레벨을 표시하는 값일 수도 있다. 일부 예들에서, 임계값은 목적지 픽셀이 완전히 불투명한, 예를 들어 0 의 값인, 불투명도의 레벨을 표시할 수도 있다. 임계값은 CPU (6) 상에서 실행하는 소프

트웨어에 의해 및/또는 GPU (12) 상의 하드웨어 또는 셰이더 컴포넌트들에 의해 구성가능할 수도 있다.

[0026] 일부 구현들에서, 목적지 알파값, α_D 는 0 에서 1 을 포함하는 범위 (즉, $0 \leq \alpha_D \leq 1$) 내에 있을 수도 있으며, 여기서 0 의 값은 완전히 불투명한 픽셀 (즉, 실질적으로 투명성을 갖지 않는 - 100% 불투명하고, 0% 투명한, 픽셀) 을 표시하고, 1 의 값은 완전히 투명한 목적지 픽셀 (즉, 실질적으로 불투명성을 갖지 않는 100% 투명한, 0% 불투명한, 픽셀) 을 표시한다. 목적지 알파값이 0 에서 1 까지 증가하기 때문에, 그러한 예들에서, 불투명도의 레벨은 0 의 목적지 알파값에서 실질적으로 완전한 불투명도에서 1 의 목적지 알파값에서 실질적으로 불투명도가 없는 것으로 감소한다. 부가적인 구현들에서, 0 의 목적지 알파값은 완전히 투명한 픽셀을 표시할 수도 있고, 1 의 목적지 알파값은 완전히 불투명한 픽셀을 표시할 수도 있다. 목적지 알파값들의 다른 예들 및 범위들이 가능하며 본 개시물의 범위 내에 있다.

[0027] 목적지 알파값은, 일부 예들에서, 그래픽스 파이프라인에서 종래 알파 블렌딩 유닛에 의해 생성된 목적지 알파값에 대응할 수도 있다. 추가 예들에서, 목적지 알파값은 전방-후방 알파 블렌딩 함수에 따라 블렌딩하는 알파 블렌딩 유닛에 의해 생성된 목적지 알파값에 대응할 수도 있다. 추가 예들에서, 목적지 알파값은 본 개시물에서 더 상세하게 설명되는 바와 같이, 프레임 버퍼 및/또는 빈 버퍼에 저장된 목적지 알파값에 대응할 수도 있다. 목적지 알파값은, 추가 예들에서, 상기 예들 중 어느 하나의 조합일 수도 있다.

[0028] 추가 예들에서, GPU 의 픽셀 프로세싱 파이프라인은 픽셀 샘플 레벨에서 프로세싱을 수행할 수도 있다. 그러한 예들에서, 각 픽셀은 복수의 픽셀 샘플들과 연관될 수도 있고, 각 픽셀 샘플은 단일 픽셀과 연관될 수도 있다. 게다가, 각 픽셀 샘플은 각각의 목적지 알파값을 가질 수도 있다. 제 2 픽셀에 대한 하나 이상의 텍스처값들이 그러한 예들에서 버퍼로부터 추출되어야 할지 여부를 결정하기 위해서, GPU (12) 는 픽셀과 연관된 샘플들 모두에 대한 목적지 알파값들이 임계값 이하인지 여부를 결정할 수도 있다. 픽셀과 연관된 샘플들 모두에 대한 목적지 알파값들이 임계값 이하이면, GPU (12)는 텍스처값들이 픽셀에 대해 추출되지 않게 할 수도 있다. 이와 달리, 픽셀과 연관된 샘플들 중 적어도 하나에 대한 목적지 알파값이 임계값 이하가 아니면, GPU (12) 는 텍스처값들이 픽셀에 대해 추출되게 할 수도 있다.

[0029] 일부 예들에서, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션은, 제 2 픽셀이 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에, 픽셀 프로세싱 파이프라인으로부터 제 2 픽셀을 폐기하는 액션일 수도 있다. 그러한 예들에서, GPU (12) 는, 목적지 알파값에 기초하여 픽셀 프로세싱 파이프라인으로부터 제 2 픽셀을 폐기할지 여부를 결정하고, 목적지 알파값에 기초하여 제 2 픽셀을 폐기하도록 결정하는 것에 응답하여, 제 2 픽셀이 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지에 도달하기 전에 픽셀 프로세싱 파이프라인으로부터 제 2 픽셀을 폐기하도록 구성될 수도 있다. GPU (12) 는 또한, 그러한 예들에서, 목적지 알파값에 기초하여 제 2 픽셀을 폐기하지 않도록 결정하는 것에 응답하여 텍스처 맵핑 스테이지를 포함하는 하나 이상의 프로세싱 스테이지들의 세트에 제 2 픽셀을 패스하도록 구성될 수도 있다.

[0030] 추가 예들에서, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션은 텍스처 맵핑이 제 2 픽셀에 대해 수행되지 않게 하는 액션일 수도 있다. 예를 들어, 액션은 텍스처값들이 제 2 픽셀에 대해 추출되지 않게 하는 제어 신호를 어서트 (assert) 하는 액션일 수도 있다. 제어 신호는, 예를 들어, 픽셀 프로세싱 파이프라인 내의 또 다른 프로세싱 스테이지가 픽셀을 폐기하게 하는 제어 신호일 수도 있다. 또 다른 예로서, 제어 신호는 픽셀 프로세싱 파이프라인 내에서 특정 픽셀에 대한 텍스처 맵핑 스테이지를 인에이블 또는 디스에이블시킬 수도 있다. 더 많은 예들에서, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션은 텍스처 맵핑이 픽셀에 대해 디스에이블되게 하는 값으로 제 2 픽셀에 대한 텍스처 맵핑 디스에이블 속성을 설정하는 액션일 수도 있다.

[0031] 본 개시물의 선택적 텍스처값 추출 기법들을 구현하는데 사용되는 목적지 알파 테스트 모듈은 다양한 방식으로 구현될 수도 있다. 예를 들어, 목적지 알파 테스트 모듈은, 전용된 고정 함수 하드웨어 블록으로서, 셰이더 유닛 상에서 실행하는 셰이더 프로그램으로서, 목적지 알파 테스트를 수행하도록 구성된 구성가능한 스텐실 테스트 블록으로서, 및/또는 목적지 알파 테스트를 수행하도록 구성된 구성가능한 z/스텐실 테스트 블록으로서 구현될 수도 있다. 이러한 다양한 구성들은 본 개시물에서 이후 더 상세하게 설명될 것이다.

[0032] GPU (12) 는, 일부 경우들에서, 컴퓨팅 디바이스 (2) 의 마더보드에 집적될 수도 있다. 다른 경우들에서, GPU (12) 는 컴퓨팅 디바이스 (2) 의 마더보드에서 포트에 설치되는 그래픽스 카드 상에 존재할 수도 있고, 그렇지 않으면 컴퓨팅 디바이스 (2) 와 상호동작하도록 구성된 주변 디바이스 내에 통합될 수도 있다. GPU (12) 는 하나 이상의 프로세서들, 예컨대 하나 이상의 마이크로프로세서들, 주문형 집적회로(ASIC)들, 필드 프로그래밍 가능 게이트 어레이(FPGA)들, 디지털 신호 프로세서(DSP)들, 또는 다른 등가 집적 또는 이산 로직 회로들

포함할 수도 있다.

[0033] GPU (12) 는 GPU 캐시 (14) 에 직접 커플링될 수도 있다. 이로써, GPU (12) 는 버스 (20) 를 사용하지 않고 GPU 캐시 (14) 로부터 데이터를 관독하고 GPU 캐시 (14) 에 데이터를 기록할 수도 있다. 즉, GPU (12) 는 칩 오프 메모리 대신 로컬 스토리지를 사용하여 데이터를 국부적으로 프로세싱할 수도 있다. 이는 과중한 버스 트래픽을 경험할 수도 있는, 버스 (20) 를 통한 데이터의 관독 및 기록에 대한 GPU (12) 의 필요성을 제거함으로써, GPU (12) 가 보다 효율적으로 동작하게 한다. 하지만, 일부 경우들에서, GPU (12) 는 별도의 메모리를 포함하지 않을 수도 있지만, 대신 버스 (20) 를 통해 시스템 메모리 (10) 를 사용할 수도 있다. GPU 캐시 (14) 는, 예를 들어 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 소거가능 프로그램가능 ROM (EPROM), 전기적 소거가능 프로그램가능 ROM (EEPROM), 플래시 메모리, 자기 데이터 매체들 또는 광학 저장 매체들과 같은, 하나 이상의 휘발성 또는 비휘발성 메모리들 또는 저장 디바이스들을 포함할 수도 있다.

[0034] CPU (6) 및/또는 GPU (12) 는 시스템 메모리 (10) 내에 할당되는 프레임 버퍼에 렌더링된 이미지 데이터를 저장할 수도 있다. 디스플레이 인터페이스 (16) 는 프레임 버퍼로부터 데이터를 추출하고 렌더링된 이미지 데이터에 의해 나타난 이미지를 디스플레이하도록 디스플레이 (18) 를 구성할 수도 있다. 일부 예들에서, 디스플레이 인터페이스 (16) 는 프레임 버퍼로부터 추출된 디지털값들을 디스플레이 (18) 에 의해 소모가능한 아날로그 신호로 컨버팅하도록 구성된 디지털-아날로그 컨버터 (DAC) 를 포함할 수도 있다. 다른 예들에서, 디스플레이 인터페이스 (16) 는 프로세싱을 위해 디지털값들을 디스플레이 (18) 에 직접 패스할 수도 있다. 디스플레이 (18) 는 모니터, 텔레비전, 프로젝션 디바이스, 액정 디스플레이 (LCD), 플라즈마 디스플레이 패널, 발광 다이오드 (LED) 어레이, 캐소드 레이 튜브 (CRT) 디스플레이, 전자 페이퍼, 표면 전도 전자 방출 디스플레이 (SED), 레이저 텔레비전 디스플레이, 나노결정 디스플레이 또는 디스플레이 유닛의 다른 유형을 포함할 수도 있다. 디스플레이 (18) 는 컴퓨팅 디바이스 (2) 내에 통합될 수도 있다. 이를 테면, 디스플레이 (18) 는 이동 전화기의 스크린일 수도 있다. 대안으로, 디스플레이 (18) 는 유선 또는 무선 통신들 링크를 통해 컴퓨터 디바이스 (2) 에 커플링된 독립형 디바이스일 수도 있다. 이를 테면, 디스플레이 (18) 는 케이블 또는 무선 링크를 통해 개인용 컴퓨터에 접속된 평판 디스플레이 또는 컴퓨터 모니터일 수도 있다.

[0035] 도 2 는 도 1 의 CPU (6), GPU (12) 및 시스템 메모리 (10) 를 더 상세하게 도시하는 블록 다이어그램이다. CPU (6) 는 적어도 하나의 소프트웨어 어플리케이션 (24), 그래픽스 API (26), 및 GPU 드라이버 (28) 를 포함하고, 그 각각은 CPU (6) 상에서 실행하는 하나 이상의 소프트웨어 어플리케이션들 또는 서비스들일 수도 있다. GPU (12) 는 그래픽스 프로세싱 커맨드들을 실행하기 위해 함께 동작하는 복수의 그래픽스 프로세싱 스테이지들을 포함하는 그래픽스 프로세싱 파이프라인 (30) 을 포함한다. 도 2 에 나타난 바와 같이, 그래픽스 프로세싱 파이프라인 (30) 은 커맨드 엔진 (32), 지오메트리 프로세싱 스테이지 (34), 래스터화 스테이지 (36), 및 픽셀 프로세싱 파이프라인 (38) 을 포함한다. 그래픽스 프로세싱 파이프라인 (30) 에서 컴포넌트들의 각각은 고정 함수 컴포넌트들, 프로그램가능 컴포넌트들로서 (즉, 프로그램가능 셰이더 유닛 상에서 실행하는 셰이더 프로그램의 부분으로서), 또는 고정 함수 및 프로그램가능 컴포넌트들의 조합으로서 구현될 수도 있다. 시스템 메모리 (10) 는 렌더링된 이미지 데이터를 저장할 수도 있는 프레임 버퍼 (40) 및 복수의 텍스처값들을 저장할 수도 있는 텍스처 버퍼 (42) 를 포함한다.

[0036] 본 개시물의 일 예의 실시형태에 의하면, 소프트웨어 어플리케이션 (24) 은 GUI 에서 하나 이상의 어플리케이션 윈도우들의 외관 및 포지션을 제어하는 윈도우 관리자 어플리케이션일 수도 있다. 하지만, 다른 예들에서, 소프트웨어 어플리케이션 (24) 은 GPU (12) 의 기능성을 이용하는 임의의 어플리케이션일 수도 있다. 예를 들어, 소프트웨어 어플리케이션 (24) 은 GUI 어플리케이션, 오퍼레이팅 시스템, 포터블 맵핑 어플리케이션, 엔지니어링 또는 아트 어플리케이션들을 위한 컴퓨터 보조 디자인 프로그램, 비디오 게임 어플리케이션, 또는 2D 또는 3D 그래픽스를 사용하는 소프트웨어 어플리케이션의 또 다른 유형일 수도 있다.

[0037] 소프트웨어 어플리케이션 (24) 은 그래픽 사용자 인터페이스 (GUI) 및/또는 그래픽스 장면을 렌더링하도록 GPU (12) 에 명령하는 하나 이상의 드로잉 명령들을 포함할 수도 있다. 예를 들어, 드로잉 명령들은 GPU (12) 에 의해 렌더링되는 하나 이상의 그래픽스 프리미티브들의 세트를 정의하는 명령들을 포함할 수도 있다. 일부 예들에서, 드로잉 명령들은, GUI 에서 사용된 복수의 윈도우 표면들의 전부 또는 부분을 집합적으로 정의할 수도 있다. 부가 예들에서, 드로잉 명령들은, 어플리케이션에 의해 정의된 모델 스페이스 또는 월드 스페이스 내에 하나 이상의 그래픽스 오브젝트들을 포함하는 그래픽스 장면의 전부 또는 부분을 집합적으로 정의할 수도 있다.

[0038] 소프트웨어 어플리케이션 (24) 은 그래픽스 API (26) 를 통해 GPU 드라이버 (28) 를 발동시켜, 하나 이상의 그

래픽스 프리미티브들을 디스플레이 가능한 그래픽스 이미지들로 렌더링하기 위한 하나 이상의 커맨드들을 GPU (12)에 발행할 수도 있다. 예를 들어, 소프트웨어 어플리케이션 (24)은 그래픽스 API (26)를 통해 GPU 드라이버 (28)를 발송시켜, GPU (12)에 프리미티브 정의들을 제공할 수도 있다. 일부 경우, 프리미티브 정의들은 드로잉 프리미티브들의 리스트, 예를 들어, 삼각형들, 사각형들, 삼각형 팬들, 삼각형 스트립들 등의 형태로 GPU (12)에 제공될 수도 있다. 프리미티브 정의들은 렌더링되는 프리미티브들과 연관된 하나 이상의 버텍스들을 특징하는 버텍스 사양들을 포함할 수도 있다. 버텍스 사양들은, 각 버텍스에 대한 포지션 좌표, 그리고 일부 경우, 예를 들어 컬러 좌표들, 정규 벡터들, 및 텍스처 좌표들과 같은, 버텍스와 연관된 다른 속성들을 포함할 수도 있다. 프리미티브 정의들은 또한, 프리미티브 유형 정보 (예를 들어, 삼각형, 사각형, 삼각형 팬, 삼각형 스트립 등), 스케일링 정보, 회전 정보 등을 포함할 수도 있다. 소프트웨어 어플리케이션 (24)에 의해 GPU 드라이버 (28)에 발행된 명령들에 기초하여, GPU 드라이버 (28)는 프리미티브를 렌더링하기 위해 수행하도록 GPU (12)에 대한 하나 이상의 동작들을 특징하는 하나 이상의 커맨드들을 공식화할 수도 있다. GPU (12)가 CPU (6)로부터 커맨드를 수신할 때, 그래픽스 프로세싱 파이프라인 (30)은 커맨드를 디코딩하고 커맨드에서 특정된 동작을 수행하도록 그래픽스 프로세싱 파이프라인 (30)내에 하나 이상의 프로세싱 엘리먼트들을 구성한다. 특정된 동작들을 수행한 후에, 그래픽스 프로세싱 파이프라인 (30)은 렌더링된 데이터를 디스플레이 디바이스와 연관된 프레임 버퍼 (40)로 출력한다.

[0039] 소프트웨어 어플리케이션 (24)이 윈도우 관리자를 포함하는 예들에서, 소프트웨어 어플리케이션 (24)은 GPU (12)가 개별 사용자 어플리케이션들에 대한 오픈 GUI 윈도우들과 연관된 하나 이상의 표면들을 렌더링하게 하도록 구성될 수도 있다. 어플리케이션 표면들의 각각은, 사용자 어플리케이션을 위해 실행하는 오픈 윈도우에 대응할 수도 있고, 본 명세서에서 표면 컬러 데이터로서 지칭될 수도 있는, 윈도우에 대한 컬러 데이터의 현재 상태를 포함할 수도 있다. 윈도우 관리자가 어플리케이션 표면을 렌더링하기 위해서, 소프트웨어 어플리케이션 (24)은 표면 컬러 데이터의 전부 또는 부분이 GPU (12)에 대한 하나 이상의 텍스처 버퍼들, 예를 들어 텍스처 버퍼 (42)에 배치되게 하는 하나 이상의 명령들을 GPU 드라이버 (28) 및/또는 메모리 제어기 (8)에 발행할 수도 있다. 게다가, 소프트웨어 어플리케이션 (24)은 특정 윈도우들에 대한 특정 시각적 효과들, 예를 들어, 음영들, 보더들, 투명도, 조광 등을 달성하기 위해 컬러 데이터를 변경하고 및/또는 부가 텍스처들을 적용할 수도 있다. 텍스처 버퍼들을 셋업한 후에, 소프트웨어 어플리케이션 (24)은 GPU (12)로 하여금 표면에 대응하는 하나 이상의 프리미티브들을 렌더링하게 하고, 프리미티브들의 일부 또는 전부에 하나 이상의 텍스처들을 적용하게 하는 하나 이상의 명령들을 발행할 수도 있다. 표면 컬러 데이터는 시스템 메모리 (10)에 또는 CPU (6)에 대한 캐시에 저장될 수도 있다. 텍스처 버퍼들은, 일부 예들에서, 시스템 메모리 (10)에 로케이팅될 수도 있다. 이로써, 일부 예들에서, 표면 컬러 데이터 및 하나 이상의 텍스처 버퍼들 양자가 동일한 메모리 서브시스템의 부분일 수도 있지만, 다른 예들에서, 표면 컬러 데이터 및 하나 이상의 텍스처 버퍼가 상이한 메모리 서브시스템들 상에 위치될 수도 있다.

[0040] 본 개시물의 예시의 실시형태들에 의하면, 소프트웨어 어플리케이션 (24)이 GPU (12)내의 알파 블렌딩 블록으로 하여금 하나 이상의 전방-후방 알파 블렌딩 함수들의 세트를 이용하여 알파 블렌딩을 수행하게 하는 하나 이상의 커맨드들을 발생할 수도 있다. 전방-후방 알파 블렌딩 함수는 전방-후방 방식으로 프로세싱되는 픽셀들을 블렌딩 (즉, 뷰어에 근접한 픽셀들이 뷰어로부터 멀리 있는 픽셀들 보다 먼저 프로세싱됨) 하도록 구성되고, GPU 들에 보통 사용되는 후방-전방 알파 블렌딩 함수들과 구별되는 알파 블렌딩 함수들로 지칭할 수도 있다. 일 예의 전방-후방 알파 블렌딩 함수들의 세트는 다음의 함수들을 포함할 수도 있다:

$$Cdst[n] = (Adst[n-1]) \times (Asrc[n] \times Csrc[n]) - Cdst[n-1] \quad (1)$$

$$Adst[n] = (1 - Asrc[n]) \times (Adst[n-1]) \quad (2)$$

[0041]

[0042] 식 중, $p[0], p[1] \dots p[n-1]$ 은 n 개의 이전에 프로세싱된 소스 픽셀들이고, $p[n]$ 은 현재 프로세싱된 소스 픽셀 (즉, 현재 소스 픽셀)이고, $Cdst[n]$ 은 현재의 소스 픽셀을 프로세싱하는 것에 응답하여 생성된 목적지 컬러값이고, $Adst[n]$ 은 현재의 소스 픽셀을 프로세싱하는 것에 응답하여 생성된 목적지 알파값이고, $Cdst[n-1]$ 은 이전에 프로세싱된 소스 픽셀 (즉, $p[n-1]$)을 프로세싱하는 것에 응답하여 생성된 목적지 컬러값이고, $Adst[n-1]$ 은 이전에 프로세싱된 소스 픽셀 (즉, $p[n-1]$)을 프로세싱하는 것에 응답하여 생성된 목적지 알파값이고, $Csrc[n]$ 은 현재의 소스 픽셀에 대응하는 소스 컬러값이고, $Asrc[n]$ 은 현재의 소스 픽셀에 대응하는 소스 알파값이다. 식들 (1) 및 (2)에서, 임의의 값 i 에 대하여, $Adst[i] = 0$ 은 완전히 불투명한 목적지 픽

셀을 표시할 수도 있고, $Adst[i] = 1$ 은 완전히 투명한 목적지 픽셀을 표시할 수도 있다. $Adst[-1]$ 및 $Cdst[-1]$ 은 적절한 디폴트값들 (예를 들어, 각각 1 및 0 의 값들) 로 초기화될 수도 있다. 일부 예들에서, $Asrc[n] \times Csrc[n]$ 곱이 셰이더 (예를 들어, 픽셀 셰이더) 에서 수행될 수도 있지만, 다른 수학 연산들이 알파 블렌딩 모듈의 부분으로서 수행된다.

[0043] GPU 드라이버 (28) 는 또한 하나 이상의 셰이더 프로그램들을 컴파일하고, GPU (12) 내에 함유된 하나 이상의 프로그램가능한 셰이더 유닛들 상으로 컴파일된 셰이더 프로그램들을 다운로드하도록 구성될 수도 있다. 셰이더 프로그램들은, 예를 들어 OpenGL 셰이딩 언어 (GLSL), 하이 레벨 셰이딩 언어 (HLSL), 그래픽스에 대한 C (Cg) 셰이딩 언어 등과 같은, 하이 레벨 셰이딩 언어로 기록될 수도 있다. 컴파일된 셰이더 프로그램들은 GPU (12) 내의 프로그램 가능한 셰이더 유닛의 동작을 제어하는 하나 이상의 명령들을 포함할 수도 있다. 예를 들어, 셰이더 프로그램들은 버텍스 셰이더 프로그램들 및/또는 픽셀 셰이더 프로그램들을 포함할 수도 있다. 버텍스 셰이더 프로그램은 프로그램 가능한 버텍스 셰이더 유닛 또는 통합된 셰이더 유닛의 실행을 제어할 수도 있고, 하나 이상의 버텍스 마다의 동작들을 특정하는 명령들을 포함할 수도 있다. 픽셀 셰이더 프로그램은 프로그램 가능한 픽셀 셰이더 유닛 또는 통합된 셰이더 유닛의 실행을 제어하는 픽셀 셰이더 프로그램들을 포함할 수도 있고, 하나 이상의 픽셀 마다의 동작들을 특정하는 명령들을 포함할 수도 있다. 본 개시물의 일부 실시형태들에 의하면, 픽셀 셰이더 프로그램은 또한 선택적으로 텍스처값들이 소스 픽셀들에 대한 대응 목적지 알파값들에 기초하여 소스 픽셀들에 대해 취출되게 하는 명령들을 포함할 수도 있다.

[0044] 그래픽스 프로세싱 파이프라인 (30) 은 그래픽스 드라이버 (28) 를 통해 CPU (6) 로부터 하나 이상의 그래픽스 프로세싱 커맨드들을 수신하고, 디스플레이 가능한 그래픽스 이미지들을 생성하기 위해 그래픽스 프로세싱 커맨드들을 실행하도록 구성될 수도 있다. 위에서 논의된 바와 같이, 그래픽스 프로세싱 파이프라인 (30) 은 그래픽스 프로세싱 커맨드들을 실행하기 위해 함께 동작하는 복수의 스테이지들을 포함한다. 하지만, 그러한 스테이지들이 별도의 하드웨어 블록들에서 반드시 구현되어야 하는 것은 아님을 유념해야 한다. 예를 들어, 픽셀 프로세싱 파이프라인 (38) 및 지오메트리 프로세싱 스테이지 (34) 의 부분들은 통합된 셰이더 유닛의 부분으로서 구현될 수도 있다.

[0045] 커맨드 엔진 (32) 은 그래픽스 프로세싱 커맨드들을 수신하고 그래픽스 프로세싱 커맨드들을 수행하기 위한 다양한 동작들을 수행하기 위해 그래픽스 프로세싱 파이프라인 (30) 내의 잔여 프로세싱 스테이지들을 구성할 수도 있다. 그래픽스 프로세싱 커맨드들은, 예를 들어 드로잉 커맨드들 및 그래픽스 상태 커맨드들을 포함할 수도 있다. 드로잉 커맨드들은 하나 이상의 버텍스들에 대한 포지션 좌표들을 특정하는 버텍스 사양 커맨드들, 그리고 일부 경우, 예를 들어, 컬러 좌표들, 정규 벡터들, 텍스처 좌표들 및 포그 좌표들과 같은 버텍스들 각각과 연관된 다른 속성값들을 포함할 수도 있다. 그래픽스 상태 커맨드들은 프리미티브 유형 커맨드들, 변환 커맨드들, 조명 커맨드들 등을 포함할 수도 있다. 프리미티브 유형 커맨드들은 렌더링되는 프리미티브의 유형 및/또는 버텍스들이 프리미티브를 형성하기 위해 어떻게 결합되는지를 특정할 수도 있다. 변환 커맨드들은 버텍스들 상에서 수행하기 위해 변환들의 유형들을 특정할 수도 있다. 조명 커맨드들은 그래픽스 장면 내의 상이한 광들의 유형, 방향 및/또는 배치를 특정할 수도 있다. 커맨드 엔진 (32) 은 지오메트리 프로세싱 스테이지 (34) 가 하나 이상의 수신된 커맨드들과 연관된 버텍스들 및/또는 프리미티브들에 관하여 지오메트리 프로세싱을 수행하게 할 수도 있다.

[0046] 지오메트리 프로세싱 스테이지 (34) 는 래스터화 스테이지 (36) 에 대한 프리미티브 데이터를 생성하기 위해서 하나 이상의 버텍스들 상에서 버텍스 마다의 동작들 및/또는 프리미티브 셋업 동작들을 수행할 수도 있다. 각 버텍스는, 예를 들어 포지션 좌표들, 컬러값들, 정규 벡터 및 텍스처 좌표들과 같은 속성들의 세트와 연관될 수도 있다. 지오메트리 프로세싱 스테이지 (34) 는 다양한 버텍스 마다의 동작들에 따라 이러한 속성들의 하나 이상을 변경한다. 예를 들어, 지오메트리 프로세싱 스테이지 (34) 는 변경된 버텍스 포지션 좌표들을 산출하기 위해 버텍스 포지션 좌표들 상에서 하나 이상의 변환들을 수행할 수도 있다. 지오메트리 프로세싱 스테이지 (34) 는, 예를 들어 모델링 변환, 뷰잉 변환, 프로젝션 변환, 모델뷰 변환, 모델뷰프로젝션 변환, 뷰포트 변환 및 깊이 범위 스케일링 변환 중 하나 이상을 버텍스 포지션 좌표들에 적용하여 변경된 버텍스 포지션 좌표들을 생성할 수도 있다. 일부 경우, 버텍스 포지션 좌표들은 모델 스페이스 좌표들일 수도 있고, 변경된 버텍스 포지션 좌표들은 스크린 스페이스 좌표들일 수도 있다. 스크린 스페이스 좌표들은 모델링, 뷰잉, 프로젝션 및 뷰포트 변환들의 적용 후에 획득될 수도 있다. 일부 경우, 지오메트리 프로세싱 스테이지 (34) 는 또한 버텍스들 상에서 버텍스 마다의 조명 동작들을 수행하여 버텍스들에 대한 변경된 컬러 좌표들을 생성할 수도 있다. 지오메트리 프로세싱 스테이지 (34) 는 또한, 예를 들어 정규 변환들, 정규 정규화 동작들, 뷰볼륨 클리핑, 균질한 분할 및/또는 백페이스 클리핑 동작들을 포함하는 다른 동작들을 수행할 수도 있다.

- [0047] 지오메트리 프로세싱 스테이지 (34) 는 프리미티브를 형성하기 위해 버텍스들이 어떻게 결합되는지를 특징하는 데이터 뿐만 아니라 래스터화되는 프리미티브를 정의하는 하나 이상의 변경된 버텍스들의 세트를 포함하는 프리미티브 데이터를 산출할 수도 있다. 변경된 버텍스들의 각각은, 예를 들어 버텍스와 연관된 프로세싱된 버텍스 속성값들 및 변경된 버텍스 포지션 좌표들을 포함할 수도 있다. 프리미티브 데이터는 그래픽스 프로세싱 파이프라인 (30) 의 추가 스테이지들에 의해 래스터화되는 프리미티브에 집합적으로 대응할 수도 있다. 개념적으로, 각 버텍스는 프리미티브의 2 개의 에지들이 만나는 프리미티브의 코너에 대응할 수도 있다. 지오메트리 프로세싱 스테이지 (34) 는 추가 프로세싱을 위해 래스터화 스테이지 (36) 에 프리미티브 데이터를 제공할 수도 있다.
- [0048] 일부 예들에서, 지오메트리 프로세싱 스테이지 (34) 의 전부 또는 부분이 하나 이상의 셰이더 유닛들 상에서 실행하는 하나 이상의 셰이더 프로그램들에 의해 구현될 수도 있다. 예를 들어, 지오메트리 프로세싱 스테이지 (34) 는, 그러한 예들에서, 버텍스 셰이더, 지오메트리 셰이더 또는 그 임의의 조합에 의해 구현될 수도 있다. 다른 예들에서, 지오메트리 프로세싱 스테이지 (34) 는 고정 함수 하드웨어 프로세싱 파이프라인으로서 또는 하나 이상의 셰이더 유닛들 상에서 실행하는 하나 이상의 셰이더 프로그램들 및 고정 함수 하드웨어의 조합으로서 구현될 수도 있다.
- [0049] 래스터화 스테이지 (36) 는 지오메트리 프로세싱 스테이지 (34) 로부터, 래스터화되는 프리미티브를 나타내는 프리미티브 데이터를 수신하고, 프리미티브를 래스터화하여 래스터화된 프리미티브에 대응하는 복수의 소스 픽셀들을 생성하도록 구성된다. 일부 예들에서, 래스터화 스테이지 (36) 는 스크린 픽셀 로케이션들이 래스터화되는 프리미티브에 의해 커버되는 것을 결정하고, 프리미티브에 의해 커버되도록 결정된 각 스크린 픽셀 로케이션에 대한 소스 픽셀을 생성할 수도 있다. 래스터화 스테이지 (36) 는, 스크린 픽셀 로케이션들이, 예를 들어 에지 워킹 기법, 에지 방정식들의 평가 등과 같은 당업자에게 공지된 기법들을 이용함으로써, 프리미티브에 의해 커버되는 것을 결정할 수도 있다. 래스터화 스테이지 (36) 는 추가 프로세싱을 위해 픽셀 프로세싱 파이프라인 (38) 에 결과의 소스 픽셀들을 제공할 수도 있다.
- [0050] 래스터화 스테이지 (36) 에 의해 생성된 각각의 소스 픽셀은 스크린 픽셀 로케이션, 즉 목적지 픽셀에 대응할 수도 있고, 하나 이상의 컬러 속성들과 연관될 수도 있다. 특정 래스터화된 프리미티브에 대해 생성된 소스 픽셀들의 모두가 래스터화된 프리미티브와 연관되도록 언급될 수도 있다. 프리미티브에 의해 커버되는 래스터화 스테이지 (36) 에 의해 결정되는 픽셀들은 개념적으로 프리미티브의 버텍스들을 나타내는 픽셀들, 프리미티브의 에지들을 나타내는 픽셀들 및 프리미티브의 내부를 나타내는 픽셀들을 포함할 수도 있다.
- [0051] 픽셀 프로세싱 파이프라인 (38) 은 래스터화된 프리미티브와 연관된 소스 픽셀을 수신하고, 소스 픽셀 상에서 하나 이상의 픽셀 마다의 동작들을 수행하도록 구성될 수도 있다. 본 개시물에 의하면, 픽셀 프로세싱 파이프라인 (38) 에 의해 수행된 픽셀 마다의 동작들은 목적지 알파 테스트, 텍스처 맵핑 동작, 및 목적지 알파값 생성 동작을 포함할 수도 있다. 목적지 알파값 생성 동작은, 일부 예들에서, 알파 블렌딩 동작에 대응할 수도 있다. 일부 예들에서, 개별 소스 픽셀을 프로세싱할 때, 픽셀 프로세싱 파이프라인 (38) 은 텍스처 맵핑 동작 전에 목적지 알파 테스트를 수행할 수도 있다. 픽셀 프로세싱 파이프라인 (38) 에 의해 수행될 수도 있는 다른 픽셀 마다의 동작들은, 예를 들어 컬러 계산, 픽셀 새이딩, 픽셀 마다의 조명, 포그 프로세싱, 블렌딩, 픽셀 오너십 테스트, 소스 알파 테스트, 스텐실 테스트, 깊이 테스트, 가위 테스트 및/또는 스트리플링 (stripping) 동작들을 포함한다. 게다가, 픽셀 프로세싱 파이프라인 (38) 은 하나 이상의 픽셀 마다의 동작들을 수행하기 위해 하나 이상의 픽셀 셰이더 프로그램을 실행할 수도 있다. 픽셀 프로세싱 파이프라인 (38) 에 의해 산출된 결과의 데이터는 본 명세서에서 목적지 픽셀로서 지칭될 수도 있고 프레임 버퍼 (40) 에 저장될 수도 있다. 목적지 픽셀 데이터는 프로세싱되었던 소스 픽셀과 동일한 디스플레이 로케이션을 갖는 프레임 버퍼 (40) 의 목적지 픽셀과 연관될 수도 있다. 목적지 픽셀 데이터는, 예를 들어 컬러값들, 목적지 알파값들, 깊이값들 등과 같은 데이터를 포함할 수도 있다.
- [0052] 본 개시물에 의하면, 픽셀 프로세싱 파이프라인 (38) 은 제 1 래스터화된 프리미티브와 연관된 제 1 픽셀을 프로세싱하는 것에 응답하여 목적지 알파값을 생성하고, 목적지 알파값을 빈 버퍼에 저장하고, 제 1 픽셀과 상이한 제 2 픽셀을 프로세싱하는 것에 응답하여 빈 버퍼로부터 목적지 알파값을 추출하고, 목적지 알파값에 기초하여, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게하는 액션을 수행할지 여부를 결정하고, 그리고 액션을 수행하도록 결정하는 것에 응답하여 제 2 픽셀에 대한 하나 이상이 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션을 수행할 수도 있다. 제 2 픽셀은 제 1 래스터화된 프리미티브와 상이한 제 2 래스터화된 프리미티브와 연관될 수도 있다. 게다가, 제 2 픽셀은 제 1 픽셀과 동일한 스크린 픽셀 로케이션들 가질 수도 있다 (예를 들어, 제 1 및 제 2 픽셀들이 스크린 스페이스에서 동일한 포지션 좌표들을 가질 수도

있다). 목적지 알파값은 제 1 및 제 2 소스 픽셀들과 동일한 스크린 픽셀 로케이션을 갖는 목적지 픽셀과 연관될 수도 있다. 예를 들어, 소스 픽셀에 대한 목적지 알파값은 소스 픽셀과 동일한 스크린 로케이션을 갖는 목적지 픽셀의 불투명도를 표시하는 값일 수도 있다. 목적지 픽셀의 불투명도는 픽셀 로케이션과 연관된 하나 이상의 이전에 프로세싱된 소스 픽셀들에 의해 적어도 부분적으로 결정될 수도 있다. 그 대응하는 목적지 알파값들에 기초하여 소스 픽셀들에 대한 텍스처값들을 선택적으로 추출함으로써, 픽셀 프로세싱 파이프라인 (38)은 텍스처값들에 관하여 발생하여야 하는 소스 판독들의 수를 효과적으로 감소시킴으로써, 그래픽스 프로세싱 시스템을 위해 필요한 메모리 액세스 대역폭을 감소시킬 수도 있다.

[0053] 일부 예들에서, 각 소스 픽셀에 대하여, 픽셀 프로세싱 파이프라인 (38)은 목적지 알파 테스트를 수행할 수도 있고, 그 테스트의 결과에 기초하여 소스 픽셀에 대한 텍스처값들을 추출하거나 추출하지 않을 수도 있다. 목적지 알파 테스트는, 예를 들어 소스 픽셀에 대응하는 목적지 알파값을 임계와 비교하는 것을 수반할 수도 있다. 그러한 예들에서, 임계는 소프트웨어 어플리케이션 (24), GPU 드라이버 (28) 및 GPU (12) 내의 하나 이상의 컴포넌트들 중 하나 이상의 컴포넌트들에 의해 구성가능할 수도 있다. 임계값은, 예를 들어 임계 레벨에서 또는 그 아래에서 목적지 픽셀 뒤에 위치되는 픽셀들이 가시적이지 않게 되고 및/또는 디스플레이될 필요가 없는 그 목적지 픽셀에 대한 불투명도의 임계 레벨을 표시하는 값일 수도 있다.

[0054] 일부 예들에서, 픽셀 프로세싱 파이프라인 (38)은 목적지 알파 테스트의 결과들에 기초한 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지 전에, 픽셀 프로세싱 파이프라인 (38)에서 소스 픽셀들을 선택적으로 폐기함으로써 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들을 구현할 수도 있다. 추가 예들에서, 픽셀 프로세싱 파이프라인 (38)은 목적지 알파 테스트의 결과들에 기초하여 소스 픽셀에 대한 텍스처 맵핑을 선택적으로 인에이블 및 디스에이블함으로써 본 개시물의 기법들을 구현할 수도 있다. 예를 들어, 픽셀 프로세싱 파이프라인 (38)은 목적지 알파 테스트의 결과들에 기초하여 텍스처 맵핑을 디스에이블하는 제어 신호를 선택적으로 어서트할 수도 있고, 또는 목적지 알파 테스트의 결과들에 기초하여 텍스처 맵핑을 디스에이블하는 픽셀 속성을 선택적으로 설정할 수도 있다.

[0055] 프레임 버퍼 (40)는 GPU (12)에 대한 목적지 픽셀들을 저장한다. 각각의 목적지 픽셀은 고유의 스크린 픽셀 로케이션과 연관될 수도 있다. 일부 예들에서, 프레임 버퍼 (40)는 각각의 목적지 픽셀에 대한 목적지 알파값 및 컬러 성분들을 저장할 수도 있다. 예를 들어, 프레임 버퍼 (40)는, 각 픽셀에 대한 적, 녹, 청, 알파 (RGBA) 성분들을 저장할 수도 있으며, 여기서 "RGB" 성분들은 컬러값들에 대응하고 "A" 성분은 목적지 알파값에 대응한다. 일부 예들에서, 프레임 버퍼 (40)에 저장된 목적지 픽셀들의 서브세트 (예를 들어 전부 미만)가 GPU (12), 예를 들어 온칩 빈 버퍼에 관하여 시스템 메모리 (10)보다 높은 유효 대역폭을 갖는 메모리에 저장될 수도 있다.

[0056] 텍스처 버퍼 (42)는 GPU (12)에 의한 추출을 위한 하나 이상의 텍스처값들을 저장한다. 텍스처값들은, 일부 예들에서, 어플리케이션 표면에 정의된 컬러값들에 대응할 수도 있다. GPU (12)는 텍스처 버퍼 (42)로부터 텍스처값들을 추출하도록 시스템 메모리 (10)에 액세스할 수도 있다. 일부 예들에서, 텍스처 버퍼 (42)에서 콘텐츠의 서브세트는 GPU (12)에 의해 액세스가능한 텍스처 캐시, 예를 들어 GPU 캐시 (14)에 캐시될 수도 있다.

[0057] 시스템 메모리 (10)가 도 2의 예시의 그래픽스 프로세싱 시스템에서 프레임 버퍼 (40) 및 텍스처 버퍼 (42) 양자를 포함하는 것으로서 도시되지만, 다른 예들에서 프레임 버퍼 (40) 및 텍스처 버퍼 (42)의 하나 또는 양자가 CPU (6) 및 GPU (12)의 하나 또는 양자와 동일하거나 상이한 칩들 상에 있는 다른 메모리 서브시스템들에서 구현될 수도 있다. 게다가, 프레임 버퍼 (40) 및 텍스처 버퍼 (42)가 동일한 메모리 서브시스템의 부분으로서 구현될 필요가 없다.

[0058] 일부 예들에서, 그래픽스 프로세싱 파이프라인 (30)은 픽셀 입상 (granularity) 대신 픽셀 샘플 입상에서 데이터를 프로세싱할 수도 있다. 그러한 예들에서, 래스터화 블록 (36)은 각각의 소스 픽셀에 대한 복수의 소스 픽셀 샘플들을 생성할 수도 있고, 픽셀 프로세싱 파이프라인 (38)은 프레임 버퍼 (40)에 저장되는, 복수의 목적지 픽셀 샘플들을 산출하기 위해 소스 픽셀 샘플들 상에서 샘플 마다의 동작들을 수행할 수도 있다. 소스 픽셀 샘플들 및 목적지 픽셀 샘플들은, 데이터를 픽셀 전체에 적용하는 대신 픽셀 샘플에 적용하는 것을 제외하고, 소스 및 목적지 픽셀들에 관하여 각각 위에서 논의되었던 것과 실질적으로 유사한 그러한 데이터를 포함할 수도 있다. 디스플레이 인터페이스 (16)는 목적지 픽셀 샘플 데이터를 디스플레이 상에 디스플레이될 수 있는 픽셀 데이터로 컨버팅할 수도 있다. 일부 예들에서, 4개의 샘플들이 각각의 픽셀에 대해 생성될 수도 있고, 각각의 샘플은 단일 픽셀과 연관될 수도 있다. 하지만, 다른 예들에서, 임의의 수의 샘플들

이 각각의 픽셀에 대해 생성될 수도 있다.

[0059] 부가 예들에서, 그래픽스 프로세싱 파이프라인 (30) 은, 본 명세서에서 대안으로 타일 렌더링 기법으로서 지칭될 수도 있는, 빈 렌더링 기법에 따라 그래픽스 이미지를 렌더링할 수도 있다. 빈 렌더링 기법에 따라 렌더링할 때, 그래픽스 프로세싱 파이프라인 (30) 은 프리미티브들의 배치 (batch)(즉, 복수의 프리미티브들)을 수신하여 결과의 그래픽스 이미지들로 렌더링할 수도 있다. 프리미티브들의 배치를 렌더링하기 위해, 결과의 그래픽스 이미지가 복수의 소형 부분들 (예를 들어, 픽셀들의 타일들 또는 빈들)로 세분화될 수도 있고, 그래픽스 프로세싱 파이프라인 (30) 이 별도의 렌더링 패스로서 그래픽스 이미지의 각 부분을 렌더링할 수도 있다. 그래픽스 이미지의 특정 부분에 대한 단일 렌더링 패스의 부분으로서, 그래픽스 프로세싱 파이프라인 (30) 은 그래픽스 이미지의 목적지 픽셀들의 특정 서브세트 (예를 들어, 목적지 픽셀들의 특정 타일)에 관하여 프리미티브들의 배치의 전부 또는 서브세트를 렌더링할 수도 있다. 특정 렌더링 패스에 대한 목적지 픽셀들의 특정 서브세트는 본 명세서에서 렌더링 패스에 대한 목적지 픽셀 범위로서 지칭될 수도 있다. 제 1 목적지 픽셀 범위에 관하여 제 1 렌더링 패스를 수행한 후에, 그래픽스 프로세싱 파이프라인 (30) 은 제 1 목적지 픽셀 범위와 상이한 제 2 목적지 픽셀 범위에 관하여 제 2 렌더링 패스를 수행할 수도 있다. 타일 또는 빈마다 연관된 프리미티브들이 렌더링될 때까지, 그래픽스 프로세싱 파이프라인 (40) 이 타일들 또는 빈들을 관통하여 점증적으로 횡단할 수도 있다.

[0060] 특정 렌더링 패스를 수행하는 동안, 그 특정 렌더링 패스와 연관된 목적지 픽셀들의 서브세트에 대한 픽셀 데이터가 빈 버퍼에 저장될 수도 있다. 렌더링 패스를 수행한 후에, 그래픽스 프로세싱 파이프라인 (30) 이 빈 버퍼의 콘텐츠를 프레임 버퍼 (40) 로 전달할 수도 있다. 일부 경우, 그래픽스 프로세싱 파이프라인 (30) 은 빈 버퍼에 저장된 데이터로 프레임 버퍼 (40) 의 데이터의 부분을 덮어쓸 수도 있다. 다른 경우, 그래픽스 프로세싱 파이프라인 (30) 은 빈 버퍼에 저장된 데이터와 프레임 버퍼 (40) 의 데이터를 컴포지트 또는 결합할 수도 있다. 빈 버퍼의 콘텐츠를 프레임 버퍼 (40) 로 전달한 후에, 그래픽스 프로세싱 파이프라인 (30) 은 빈 버퍼를 디폴트값들로 초기화하고 상이한 목적지 픽셀 범위에 관하여 후속 렌더링 패스를 시작할 수도 있다.

[0061] 일부 구현들에서, 프리미티브들의 배치에서 수신된 상이한 프리미티브들을 적절한 빈들로 분류하기 위해서, 별도의 렌더링 패스들 전에 비닝 패스가 수행될 수도 있다. 예를 들어, 그래픽스 프로세싱 파이프라인 (30) 은, 각각의 프리미티브가 어느 타일들 (즉, 빈들)에 기여하는지 또는 중첩하는지를 결정하고, 결정에 기초하여 프리미티브를 하나 이상의 타일 특정 빈들에 할당할 수도 있다. 그 후, 특정 빈 동안 렌더링 패스가 수행될 때, 그래픽스 프로세싱 파이프라인 (30) 은 빈과 연관된 특정 목적지 픽셀 범위에 관하여 빈에 할당되는 프리미티브들의 각각을 렌더링할 수도 있다. 본 명세서에서 그래픽스 프로세싱 파이프라인 (30) 이 비닝 패스를 수행하는 것으로서 기재되지만, 다른 예들에서, 비닝 패스는 GPU (12) 에서의 다른 컴포넌트에 의해, CPU (6) 에서의 소프트웨어 모듈 (예를 들어, GPU 드라이버 (28))에 의해, 또는 이들의 임의의 조합에 의해 수행될 수도 있다.

[0062] 부가 구현들에서, 비닝 패스가 반드시 수행되지 않을 수도 있다. 대신, GPU 드라이버 (28) 및/또는 그래픽스 프로세싱 파이프라인 (30) 은 수신된 프리미티브들의 배치에서 모든 프리미티브들에 관하여 렌더링 패스들의 각각을 수행할 수도 있다. 각각의 렌더링 패스에 대하여, GPU 드라이버 (28) 및/또는 그래픽스 프로세싱 파이프라인 (30) 은 그 렌더링 패스 동안 주어진 렌더링 패스와 연관된 목적지 픽셀들이 업데이트되도록 상이한 가위 설정들 및/또는 상이한 메모리 어드레스 오프셋들을 사용할 수도 있다.

[0063] 일부 예들에서, 그래픽스 프로세싱 파이프라인 (30) 은 제 1 통신 인터페이스를 통해 빈 버퍼에 액세스할 수도 있고, 제 2 통신 인터페이스를 통해 텍스처 버퍼 (42) 에 액세스할 수도 있으며, 제 3 통신 인터페이스를 통해 프레임 버퍼 (40) 에 액세스할 수도 있다. 그러한 예들에서, 제 1 통신 인터페이스는, 일부 예들에서, 제 2 통신 인터페이스 및/또는 제 3 통신 인터페이스보다 높은 대역폭을 가질 수도 있다. 일부 경우, 제 2 통신 인터페이스는 제 3 통신 인터페이스와 동일한 통신 인터페이스일 수도 있고, 예를 들어, 프레임 버퍼 (40) 및 텍스처 버퍼 (42) 가 시스템 메모리 (10) 에서 양자 구현된다. 제 2 통신 인터페이스 및 제 3 통신 인터페이스는, 일부 예들에서, 도 1의 버스들 (20 및/또는 22)에 대응할 수도 있다. 빈 버퍼가 온칩 빈 버퍼일 때, 제 1 통신 인터페이스는 GPU (12) 내부에 있는 통신 인터페이스일 수도 있다.

[0064] 본 명세서에서 사용된 바와 같이, 대역폭은 통신 인터페이스가 2 개의 컴포넌트들, 예를 들어 메모리 컴포넌트 및 GPU (12) 사이에서 데이터를 전달할 수 있는 레이트를 지칭할 수도 있다. 대역폭에 대한 유닛들은, 일부 예들에서, 시간의 단위 당 비트들의 개수, 예를 들어 초당 기가비트들 (Gb/s)로서 주어질 수도 있다. 다중

비트들의 버스 폭을 갖는 버스가 통신 인터페이스의 부분으로서 사용될 때, 일부 예들에서, 대역폭은 단일 비트 라인을 따라 데이터가 전달되는 레이트로 곱해진 버스의 폭의 적(product)과 같다. 예를 들어, 버스가 16 비트들 와이드이고, 버스의 각 비트 라인이 2 Gb/s 의 레이트에서 데이터를 전달할 수 있는 경우, 버스의 대역폭은 32 Gb/s 와 동등할 수도 있다. 다중 버스들이 2 개의 컴포넌트들 사이에서 통신 인터페이스를 형성하면, 통신 인터페이스의 대역폭은 다중 버스들의 각각의 대역폭, 예를 들어 개별 버스들의 각각의 최소 대역폭의 함수일 수도 있다.

[0065] 빈 버퍼에 액세스하는데 사용된 통신 인터페이스에 대한 높은 대역폭을 달성하기 위해, 빈 버퍼는, 일부 예들에서, GPU (12) 와 동일한 마이크로칩 상에 구현될 수도 있다. 그러한 빈 버퍼는 온 칩 빈 버퍼로서 지칭될 수도 있다. 빈 버퍼가 GPU (12) 와 동일한 칩 상에서 구현될 때, GPU (12) 가 반드시 시스템 및 메모리 버스들 (예를 들어, 도 1 에서 버스들 (20, 22)) 를 통해 빈 버퍼에 액세스할 필요는 없지만, 대신 GPU (12) 와 동일한 칩 상에서 구현되는 내부 통신 인터페이스 (예를 들어, 버스) 를 통해 빈 버퍼에 액세스할 수도 있다.

그러한 인터페이스는 온 칩이기 때문에, 시스템 및 메모리 버스들 보다 높은 대역폭에서 동작할 수도 있다.

상술한 기법은 프레임 버퍼 (40) 및/또는 텍스처 버퍼 (42) 에 액세스하는데 사용된 통신 인터페이스의 대역폭을 초과하는 빈 버퍼에 대한 통신 인터페이스를 달성하는 하나의 방식이지만, 다른 기법들이 가능하며 본 개시물의 범위 내에 있다.

[0066] 빈 버퍼의 용량은, 일부 예들에서, 컴퓨팅 디바이스들의 소정의 유형들, 예를 들어 모바일 디바이스들 상에서 유효한 면적에 의해 제한될 수도 있다. 게다가, 빈 버퍼가 GPU (12) 와 동일한 칩 상에서 구현될 때, 동일한 칩 상에서 빈 버퍼를 구현하는데 유효한 면적의 양은 칩 상에서 구현되는 다른 기능성으로 인하여 제한될 수도 있다. 일부 예들에서, 빈 버퍼는 빈 버퍼의 용량을 더욱 제한하는 텍스처 버퍼 (42) 및/또는 프레임 버퍼 (40) 중 하나 이상의 비트 밀도보다 낮은 비트 밀도를 가질 수도 있다. 이들 및/또는 다른 인자들 때문에, 빈 버퍼의 용량이, 일부 경우, 프레임 버퍼 (40) 및 텍스처 버퍼 (42) 중 하나 또는 양자의 사이즈보다 적을 수도 있다. 따라서, 빈 버퍼의 용량은, 그러한 예들에서, 그래픽스 이미지와 연관된 복수의 목적지 픽셀들 모두에 대한 픽셀 데이터를 저장하여야 하는 최소 용량보다 적을 수도 있다. 메모리 컴포넌트의 용량은 메모리 컴포넌트에 저장될 수 있는 데이터의 최대량 (예를 들어 비트들의 최대 수) 를 지칭할 수도 있다. 프레임 버퍼 (40) 및 텍스처 버퍼 (42) 의 사이즈는 프레임 버퍼 (40) 및 텍스처 버퍼 (42) 에 각각 할당된 메모리 범위 내에 저장된 데이터의 양 (예를 들어, 비트들의 수) 을 지칭할 수도 있다. 비트 밀도는 특정 면적양에 저장될 수 있는 비트들의 수를 지칭할 수도 있다.

[0067] 위에서 논의된 바와 같이, 빈 렌더링 기법에 따라 렌더링할 때, 그래픽스 프로세싱 파이프라인 (30) 은 별도의 렌더링 패스로서 그래픽스 이미지의 각 부분을 렌더링할 수도 있다. 예를 들어, 그래픽스 이미지의 특정 부분에 대한 단일 렌더링 패스의 부분으로서, 그래픽스 프로세싱 파이프라인 (30) 은 특정 목적지 픽셀 범위 (예를 들어, 그래픽스 이미지의 목적지 픽셀들의 특정 서브세트) 에 관하여 프리미티브들의 배치의 전부 또는 서브세트를 렌더링할 수도 있다. 빈 버퍼의 용량은 목적지 픽셀의 사이즈 이상이 되도록 구성될 수도 있다.

이에 따라, 단일 렌더링 패스 동안, 렌더링 패스에 대한 목적지 픽셀 범위와 연관된 모든 목적지 픽셀 데이터 (예를 들어, 목적지 알파값들 및 목적지 컬러값들) 가 프레임 버퍼 (40) 에 반드시 액세스할 필요 없이도 빈 버퍼에서 이용가능할 수도 있다. 따라서, 단일 렌더링 패스 동안, 그래픽스 프로세싱 파이프라인 (30) 은 상대적으로 낮은 대역폭 통신 인터페이스를 통해 프레임 버퍼 (40) 로부터 그러한 데이터를 판독하여야 하는 대신 상대적으로 높은 대역폭 통신 인터페이스를 통해 빈 버퍼로부터 목적지 알파값들을 판독할 수도 있다.

[0068] 빈 렌더링을 수행하지 않는 일부 그래픽스 시스템들이 하드웨어 기반 온 칩 캐시를 사용하여 프레임 버퍼의 부분을 캐시하는 것이 가능할 수도 있지만, 그러한 캐시들은 필요할 때 주어진 픽셀에 대한 목적지 알파값이 이용가능하게 되는 것을 보장하지 않는다. 이는 다중 목적지 픽셀들이 하드웨어 기반 캐시에서 동일한 어드레스에 맵핑할 수도 있기 때문이다. 이러한 경우에 빈 렌더링이 사용되지 않으면, 하드웨어 기반 캐시의 현재 상태는 현재 프로세싱된 프리미티브와 연관된 목적지 픽셀값들을 반드시 포함하지 않을 수도 있지만, 대신 그래픽스 이미지의 다른 면적들에서 이전에 프로세싱된 프리미티브들과 연관된 목적지 픽셀값들을 포함할 수도 있다.

[0069] 다중 목적지 픽셀이 동일한 캐시 로케이션에 맵핑하는 하드웨어 기반 캐시와 대조적으로, 주어진 렌더링 패스에 대하여 빈 버퍼에 저장된 목적지 픽셀들은, 일부 예들에서, 고유하게 어드레스가능할 수도 있다. 즉, 주어진 렌더링 패스에 대하여, 일 대 일 맵핑은 그 렌더링 패스에 대해 사용된 목적지 픽셀들 및 빈 버퍼에서의 어드레스가능한 스토리지 슬롯들 사이에 정의될 수도 있다. 따라서, 빈 렌더링 기법에 따라 렌더링할 때, 주어진 비닝 패스에 대한 모든 목적지 알파값들은, 일부 예들에서, 상대적으로 낮은 대역폭 통신 인터페이스를 통

해 빈 버퍼로부터 이용가능할 수도 있다. 게다가, 하드웨어 기반 캐시 시스템들과 달리, 빈버퍼에서 고유하게 어드레스가능한 데이터 때문에, 캐시 누락들이 발생하지 않고, 캐시 누락의 이벤트에서 비용이 높은 대역폭 프레임 버퍼 액세스들에 의지하는 필요성을 완화시킨다.

[0070] 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들은, 특정 빈에 대한 목적지 알파값들이, 필요할 때 목적지 알파 테스트 모듈에 의해 빈 버퍼로 이미 로딩될 수도 있기 때문에, 빈 렌더링의 컨텍스트에서 특히 유용할 수도 있다. 이는 낮은 유효 대역폭을 갖는 메모리, 예를 들어 시스템 메모리 (10) 에 로케이팅될 수도 있는, 프레임 버퍼 (40) 로부터 그러한 값들에 액세스할 필요성을 완화시킬 수도 있다. 따라서, 비닝 컨텍스트에서, 목적지 알파값들을 추출할 필요가 있을 수도 있는 오버헤드 대역폭은, 본 개시물의 기법들에 따라 텍스처 버퍼에 대한 소스 판독들의 수를 감소시킴으로써 달성되는 대역폭 절약들에 비해 상대적으로 최소일 수도 있다.

[0071] 추가 예들에서, GPU 드라이버 (28) 는 픽셀 프로세싱 파이프라인 (38) 에 의해 구현되는, 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들을 선택적으로 인에이블 및 디스에이블 시킬 수도 있다. 일부 예들에서, 소프트웨어 어플리케이션 (24) 은 본 개시물의 기법들에 따라 목적지 알파값 기반 선택적 텍스처값 추출 모드를 인에이블시키도록 GPU 드라이버 (28) 에 명령하는 커맨드를 GPU 드라이버 (28) 에 발행할 수도 있다.

부가 예들에서, GPU 드라이버 (28) 는, 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들을 사용하면 결과의 이미지의 품질과 간섭하지 않게 되는 상황들을 검출할 수도 있으며, 예를 들어 GPU 드라이버 (28) 가 전방-후방 렌더링 스킴이 소프트웨어 어플리케이션 (24) 에 의해 구현될 때를 검출할 수도 있다. 그러한 검출에 응답하여, GPU 드라이버 (28) 는 본 개시물의 기법들에 따라 GPU (12) 에 대한 목적지 알파값 기반 선택적 텍스처값 추출 모드를 인에이블시킬 수도 있다.

[0072] 본 명세서에서 사용된 바와 같이, 소스 픽셀은 래스터화 블록 (36) 에 의해 생성되고 프레임 버퍼 (40) 또는 대응 빈 버퍼 중 어느 하나에 아직 저장되지 않은 픽셀 데이터를 지칭할 수도 있다. 프레임 버퍼 (40) 또는 대응 빈 버퍼에 기록된 픽셀 데이터는 목적지 픽셀로서 지칭될 수도 있다. 목적지 픽셀은 상이한 프리미티브들과 연관된 다중 소스 픽셀들로부터 컴포지션된 픽셀 데이터를 포함할 수도 있다. 따라서, 픽셀 프로세싱 파이프라인 (38) 은 소스 픽셀들 상에서 동작하고, 소스 픽셀들을 프로세싱하는 것에 응답하여 프레임 버퍼 (40) 또는 대응 빈 버퍼에서 목적지 픽셀 데이터를 업데이트한다. 모든 픽셀 데이터들이 목적지 픽셀들이 되는 것이 반드시 보장되지는 않음을 유념해야 한다. 예를 들어, 이전에 프로세싱된 픽셀들에 의해 폐쇄되는 임의의 후속으로 프로세싱된 소스 픽셀들이 반드시 목적지 픽셀이 되지 않을 수도 있다. 대신, z 테스트 및/또는 목적지 알파 테스트가 그러한 폐쇄된 픽셀들을 폐기할 수도 있다. 일부 경우, 용어 "프래그먼트" 는 "소스 픽셀" 로 지칭하도록 당업자에 의해 사용될 수도 있고, 용어 "픽셀" 은 "목적지 픽셀" 을 지칭하도록 당업자에 의해 사용될 수도 있다. 본 개시물에 기재된 픽셀들 중 어느 것이 소스 픽셀인지 또는 목적지 픽셀인지는, 픽셀이 기재되는 맥락으로부터 명백히 식별될 것이고 또는 결정될 수 있어야 한다.

[0073] 특정 픽셀 로케이션에 대한 목적지 알파값은 특정 스크린 로케이션에 대응하는 소스 픽셀이 프로세싱을 완료한 시간 마다 증가하여 업데이트될 수도 있다. 업데이트된 목적지 알파값은 동일한 픽셀 로케이션에 대한 이전의 목적지 알파값의 함수일 수도 있다. 따라서, 목적지 알파값은 주어진 시점까지 프로세싱을 완료한 목적지 픽셀에 대응하는 모든 소스 픽셀들에 기초하여 주어진 시점에서 목적지 픽셀의 불투명도를 표시하는 "컴포지션" 알파값을 나타낼 수도 있다.

[0074] 도 3은 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들을 구현하는데 사용될 수도 있는 일 예의 픽셀 프로세싱 파이프라인 (50) 을 도시하는 블록 다이어그램이다. 픽셀 프로세싱 파이프라인 (50) 은 소스 픽셀들에 대응하는 목적지 알파값들에 기초한 텍스처 맵핑 스테이지 전에, 소스 픽셀들을 선택적으로 폐기하도록 구성될 수도 있다. 픽셀 프로세싱 파이프라인 (50) 은 목적지 알파 테스트 모듈 (52), 텍스처 맵핑 모듈 (54), 목적지 알파 결정 모듈 (56), 텍스처 버퍼 (58) 및 목적지 알파 버퍼 (60) 를 포함한다. 목적지 알파 테스트 모듈 (52), 텍스처 맵핑 모듈 (54), 및 목적지 알파 결정 모듈 (56) 은 하나 이상의 고정 함수 프로세싱 스테이지들, 하나 이상의 프로그램가능 스테이지들 또는 그 임의의 조합으로서 하나 이상의 프로세서들 상에서 구현될 수도 있다. 텍스처 버퍼 (58) 및 목적지 알파 버퍼 (60) 는 하나 이상의 메모리 저장 유닛들로서 각각 구현될 수도 있다.

[0075] 목적지 알파 테스트 모듈 (52) 이전의 프로세싱 스테이지로부터 소스 픽셀을 수신하고, 수신된 픽셀 상에서 목적지 알파 테스트를 수행하고, 목적지 알파 테스트의 결과들에 기초하여 소스 픽셀을 선택적으로 폐기하도록 구성된다. 예를 들어, 목적지 알파 테스트 모듈 (52) 은, 소스 픽셀을 폐기할지 또는 소스 픽셀에 대응하는

목적지 알파값에 기초한 추가 프로세싱을 위해 하나 이상의 후속 프로세싱 스테이지들의 세트로 소스 픽셀을 패스할지를 결정할 수도 있다. 이러한 방식으로, 목적지 알파 테스트 모듈 (52) 은 수신된 소스 픽셀에 대응하는 목적지 픽셀이 목적지 픽셀에 대응하는 하나 이상의 이전에 프로세싱된 소스 픽셀들로 인해 이미 불투명한지 여부를 효과적으로 결정할 수도 있다.

[0076] 일부 예들에서, 이전의 프로세싱 스테이지는 래스터화 블록 (36) 일 수도 있다. 추가 예들에서, 이전의 프로세싱 스테이지는 이전의 픽셀 프로세싱 파이프라인 스테이지, 예를 들어 픽셀 소유 테스트 블록, 가위 테스트 블록, 소스 알파 테스트 블록, 픽셀 셰이더 스테이지 또는 그래픽스 프로세서에서 발견되는 임의의 다른 픽셀 프로세싱 스테이지일 수도 있다. 하나 이상의 후속 프로세싱 스테이지들의 세트는 텍스처 맵핑 모듈 (54) 을 포함할 수도 있다. 일부 예들에서, 목적지 알파 테스트 모듈 (52) 은 목적지 알파 테스트 모듈 (52) 과 텍스처 맵핑 모듈 (54) 사이의 하나 이상의 중계 프로세싱 스테이지들로 소스 픽셀을 패스함으로써 텍스처 맵핑 모듈 (54) 로 소스 픽셀을 패스할 수도 있다. 부가 예들에서, 목적지 알파 테스트 모듈 (52) 은 텍스처 맵핑 모듈 (54) 로 소스 픽셀을 직접 패스할 수도 있다. 하나 이상의 후속 프로세싱 스테이지들의 세트로 패스된 픽셀들은 본 명세서에서 패스된 픽셀들로서 지칭될 수도 있다.

[0077] 본 개시물에 의하면, 목적지 알파 테스트 모듈 (52) 은, 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하고, 액션을 수행하도록 결정하는 것에 응답하여 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하는 액션을 수행하도록 구성될 수도 있다. 예를 들어, 소스 픽셀을 수신할 때, 목적지 알파 테스트 모듈 (52) 은 소스 픽셀에 대응하는 목적지 알파 버퍼 (60) 로부터 목적지 알파값을 추출할 수도 있으며, 예를 들어 목적지 알파 테스트 모듈 (52) 은 소스 픽셀로 포함된 포지션 좌표들을 사용하여 소스 픽셀에 대한 스크린 픽셀 로케이션을 결정하고, 스크린 픽셀 로케이션과 연관된 목적지 알파값을 추출할 수도 있다. 추출된 목적지 알파값에 기초하여, 목적지 알파 테스트 모듈 (52) 은 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하고, 그렇게 결정된다면 그 액션을 수행할 수도 있다.

[0078] 도 3에 나타난 예시의 픽셀 프로세싱 파이프라인에서, 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하는 목적지 알파 테스트 모듈 (52) 에 의해 수행된 액션이 텍스처 맵핑 모듈 (54) 에 의해 나타나는 텍스처 맵핑 스테이지 전에, 픽셀 프로세싱 파이프라인 (50) 으로부터 소스 픽셀을 폐기하는 것을 포함한다. 하지만, 다른 예시의 실시형태들에서, 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하도록 목적지 알파 테스트 모듈 (52) 에 의해 다른 액션들이 수행될 수도 있다. 예를 들어, 목적지 알파 테스트 모듈 (52) 은, 일부 예들에서, 픽셀 프로세싱 파이프라인 (50) 내의 또 다른 프로세싱 스테이지가 픽셀을 폐기하게 하는 제어 신호를 어서트할 수도 있다. 부가 예들에서, 목적지 알파 테스트 모듈 (52) 은 픽셀 프로세싱 파이프라인 (50) 내의 특정 픽셀에 대한 텍스처 맵핑 모듈 (54) 을 인에이블 또는 디스에이블시키는 제어 신호를 어서트할 수도 있다. 추가 예들에서, 목적지 알파 테스트 모듈 (52) 은, 텍스처 맵핑이 소스 픽셀에 대해 디스에이블되게 하는 값으로 소스 픽셀에 대한 텍스처 맵핑 디스에이블 속성을 설정하고, 픽셀 프로세싱 파이프라인 (50) 에서 하나 이상의 후속 프로세싱 스테이지들로 픽셀을 패스할 수도 있다. 그러한 예들에서, 텍스처 맵핑 모듈 (54) 은 일 세트의 텍스처 맵핑 디스에이블 속성을 갖는 픽셀들에 대한 텍스처 맵핑을 디스에이블시키도록 구성될 수도 있다.

[0079] 일부 예들에서, 목적지 알파 테스트 모듈 (52) 은, 추출된 목적지 알파값을 임계값과 비교함으로써 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정할 수도 있다. 비교의 결과들에 기초하여, 목적지 알파 테스트 모듈 (52) 은 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정할 수도 있다. 예를 들어, 목적지 알파 테스트 모듈 (52) 은 목적지 알파값이 임계 이하인지 여부를 결정할 수도 있다. 목적지 알파값이 임계 이하라고 결정하는 것에 응답하여, 목적지 알파 테스트 모듈 (52) 은 소스 픽셀에 대한 하나 이상의 픽셀값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하는, 예를 들어 텍스처 맵핑 스테이지 (54) 전에 소스 픽셀을 폐기하는 액션을 수행할 수도 있다. 한편, 목적지 알파값이 임계 이하가 아니라고 결정하는 것에 응답하여, 목적지 알파 테스트 모듈 (52) 은 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되게 하는 액션을 수행할 수도 있다. 임계값은, 예를 들어 임계 레벨에서 또는 그 아래에서 목적지 픽셀 뒤에 위치되는 픽셀들이 가시적이지 않게 되고 및/또는 디스플레이될 필요가 없는 그 목적지 픽셀에 대한 불투명도의 임계 레벨을 표시하는 값일 수도 있다.

[0080] 임계값은, 일부 예들에서, 목적지 알파 모듈 (52) 에 하드웨어 내장된 고정 값일 수도 있다. 추가 예들에서, 임계값은 그래픽스 프로세서 내의 다른 컴포넌트들에 의해 구성될 수도 있고 및/또는 그래픽스 드라

이버 또는 사용자 어플리케이션에 의해 프로그램가능할 수도 있다.

- [0081] 일부 구현들에서, 목적지 알파값, α_D 는 0 에서 1 을 포함하는 범위 (즉, $0 \leq \alpha_D \leq 1$) 내에 있을 수도 있으며, 여기서 0 의 값은 완전히 불투명한 픽셀을 표시하고, 1 의 값은 완전히 투명한 목적지 픽셀을 표시한다. 목적지 알파값이 0 에서 1 까지 증가하기 때문에, 그러한 예들에서, 불투명도의 레벨은 0 의 목적지 알파값의 실질적으로 완전한 불투명도에서 1 의 목적지 알파값의 실질적으로 불투명성이 없는 것으로 감소한다.
- [0082] 일부 예들에서, 임계값은 완전히 불투명한 목적지 픽셀, 예를 들어 0 의 값을 표시할 수도 있다. 임계값을 0 으로 설정함으로써, 목적지 알파 테스트 블록은 이미 완전히 불투명한 목적지 픽셀들에 대응하는 소스 픽셀들을 폐기할 수도 있다. 전방-후방 방식으로 표면들이 드로잉될 때, 목적지 픽셀에 대해 후속으로 프로세싱되는 임의의 소스 픽셀들은 목적지 픽셀의 전체 외관에 기여하지 않게 되며, 이는 임의의 그러한 소스 픽셀들이 목적지 픽셀에 대해 이미 프로세싱된 소스 픽셀들 뒤에 로케이팅되기 때문이다. 이러한 소스 픽셀들을 폐기하면 전체 이미지에 기여하지 않는 텍스처값들을 추출할 필요가 없는 것으로 인해 그래픽스 프로세싱 시스템에서 메모리 액세스 대역폭을 자유화할 수도 있다.
- [0083] 목적지 알파 테스트 모듈 (52) 은 다양한 방식으로 구현될 수도 있다. 예를 들어, 목적지 알파 테스트 모듈 (52) 은 전용된 고정 함수 하드웨어 블록으로서, 셰이더 유닛 상에서 실행하는 셰이더 프로그램으로서, 목적지 알파 테스트를 수행하도록 구성된 구성가능한 스텐실 테스트 블록으로서, 및/또는 목적지 알파 테스트를 수행하도록 구성된 구성가능한 z/스텐실 테스트 블록으로서 구현될 수도 있다. 이러한 다양한 구성들은 본 개시물에서 이후 더 상세하게 기재될 것이다.
- [0084] 텍스처 맵핑 모듈 (54) 은 소스 픽셀들, 예를 들어 패스된 소스 픽셀들을 수신하고, 소스 픽셀들 상에서 텍스처 맵핑 동작들을 수행하여 텍스처 맵핑된 픽셀들을 산출하도록 구성된다. 각 소스 픽셀에 대하여, 텍스처 맵핑 모듈 (54) 은 수신된 소스 픽셀에 포함된 하나 이상의 텍스처 좌표들에 기초하여 텍스처 버퍼 (58) 로부터 하나 이상의 텍스처값들을 추출할 수도 있다. 텍스처 좌표들은 어느 텍스처값들을 텍스처 버퍼 (58) 로부터 추출할지를 특정하고 및/또는 결정할 수도 있다. 텍스처 좌표들을 추출한 후, 텍스처 맵핑 모듈 (54) 은 변경된 컬러 성분들을 갖는 결과의 텍스처 맵핑된 픽셀을 생성하기 위해 텍스처 맵핑 함수에 따라 하나 이상의 추출된 텍스처값들에 기초하여 소스 픽셀의 하나 이상의 컬러 좌표들 (예를 들어, RGB 성분들) 을 변경할 수도 있다. 텍스처 맵핑 함수는 텍스처 맵핑된 컬러값들을 소스 픽셀의 포지션 좌표들 및 텍스처 버퍼 (58) 로부터 추출된 텍스처값들의 함수로서 특정할 수도 있다. 텍스처 맵핑 함수는 사용자 특정 함수일 수도 있다. 다양한 텍스처 맵핑 함수들은 당업계에 알려져 있으며 본 개시물에서 더 상세하게 설명되지 않는다. 일부 예들에서, 텍스처 맵핑 모듈 (54) 은 셰이더 유닛 상에서 실행하는 셰이더 프로그램, 고정 함수 하드웨어 프로세싱 스테이지 또는 그 임의의 조합으로서 구현될 수도 있다.
- [0085] 텍스처 맵핑 모듈 (54) 은 텍스처 맵핑된 픽셀들을 추가 프로세싱을 위한 하나 이상의 후속 프로세싱 스테이지들로 패스할 수도 있다. 목적지 알파 결정 모듈 (56) 이 도 3에서 텍스처 맵핑 모듈 (54) 에 대한 후속 프로세싱 스테이지로서 도시되지만, 임의의 수의 중간 프로세싱 스테이지들이 텍스처 맵핑 모듈 (54) 과 목적지 알파 결정 모듈 (56) 사이에 포함될 수도 있다. 유사하게, 목적지 알파 테스트 모듈 (52) 이 도 3에서 텍스처 맵핑 모듈 (54) 에 대한 이전의 프로세싱 스테이지로서 도시되지만, 임의의 수의 중간 프로세싱 스테이지들이 목적지 알파 테스트 모듈 (52) 과 텍스처 맵핑 모듈 (54) 사이에 포함될 수도 있다.
- [0086] 목적지 알파 결정 모듈 (56) 은 소스 픽셀을 수신하고, 소스 픽셀과 동일한 스크린 픽셀 로케이션 (즉, 스크린 스페이스에서의 포지션 좌표들) 을 갖는 목적지 픽셀에 대한 목적지 알파값을 생성하도록 구성된다. 일부 예들에서, 목적지 알파 결정 모듈 (56) 은 목적지 알파 버퍼 (60) 로부터 스크린 픽셀 로케이션에 대한 이전의 목적지 알파값을 추출하고, 소스 픽셀의 알파 성분들 및/또는 하나 이상의 컬러 성분들 및 이전의 목적지 알파값의 함수로서 스크린 픽셀 로케이션에 대한 후속 목적지 알파값을 생성할 수도 있다. 예를 들어, 목적지 알파 결정 모듈 (56) 은 이전의 목적지 알파값 (즉, 목적지 알파 버퍼 (60) 로부터 추출된 "A" 성분) 및 소스 픽셀의 소스 알파값 (즉, "A" 성분) 의 함수로서 스크린 픽셀 로케이션에 대한 후속 목적지 알파값을 생성할 수도 있다.
- [0087] 일부 예들에서, 목적지 알파 결정 모듈 (56) 은 알파 블렌딩 모듈일 수도 있고, 목적지 알파 결정 모듈 (56) 은 수신된 소스 픽셀을 알파 블렌딩 함수에 기초하여 프로세싱하는 것에 응답하여 목적지 알파값을 생성할 수도 있다. 목적지 알파값을 업데이트하는데 사용된 특정 함수는 그래픽 프로세서에 의해 통상 구성가능하다. 알파 블렌딩 함수는, 일부 경우, 전방-후방 알파 블렌딩 함수일 수도 있다. 일 예의 전방-후방 알파 블렌딩

함수들의 세트는 다음의 함수들을 포함할 수도 있다:

$$Cdst[n] = (Adst[n-1]) \times (Asrc[n] \times Csrc[n]) - Cdst[n-1] \quad (3)$$

$$Adst[n] = (1 - Asrc[n]) \times (Adst[n-1]) \quad (4)$$

[0088]

[0089]

식 중, $p[0], p[1] \dots p[n-1]$ 은 n 개의 이전에 프로세싱된 소스 픽셀들이고, $p[n]$ 은 현재 프로세싱된 소스 픽셀 (즉, 현재 소스 픽셀) 이고, $Cdst[n]$ 는 현재의 소스 픽셀을 프로세싱하는 것에 응답하여 생성된 목적지 컬러값이고, $Adst[n]$ 은 현재의 소스 픽셀을 프로세싱하는 것에 응답하여 생성된 목적지 알파값이고, $Cdst[n-1]$ 은 이전에 프로세싱된 소스 픽셀 (즉, $p[n-1]$) 을 프로세싱하는 것에 응답하여 생성된 목적지 컬러값이고, $Adst[n-1]$ 은 이전에 프로세싱된 소스 픽셀 (즉, $p[n-1]$) 을 프로세싱하는 것에 응답하여 생성된 목적지 알파값이고, $Csrc[n]$ 은 현재의 소스 픽셀에 대응하는 소스 컬러값이고, $Asrc[n]$ 은 현재의 소스 픽셀에 대응하는 소스 알파값이다. 식들 (3) 및 (4) 에서, 임의의 값 i 에 대하여, $Adst[i] = 0$ 은 완전히 불투명한 목적지 픽셀을 표시할 수도 있고, $Adst[i] = 1$ 은 완전히 투명한 목적지 픽셀을 표시할 수도 있다. $Adst[-1]$ 및 $Cdst[-1]$ 은 적절한 디폴트값들 (예를 들어, 각각 1 및 0 의 값들) 로 초기화될 수도 있다. 일부 예들에서, $Asrc[n] \times Csrc[n]$ 곱이 셰이더 (예를 들어, 픽셀 셰이더) 에서 수행될 수도 있지만, 다른 수학 연산들이 목적지 알파 결정 모듈 (56) 의 부분으로서 수행된다.

[0090]

텍스처 버퍼 (58) 는 텍스처 맵핑 모듈 (54) 에 의한 취출을 위한 텍스처값들을 저장하도록 구성된다. 텍스처값들은, 일부 예들에서, 어플리케이션 표면에서 정의된 컬러값들에 대응할 수도 있다. 텍스처 버퍼 (58) 는 일부 예들에서 시스템 메모리 (10) 에서 구현될 수도 있고, 또는 다른 예들에서는 또 다른 메모리 (미도시) 에서 구현될 수도 있다. 일부 구현들에서, 텍스처 버퍼 (58) 의 전부 또는 부분은 캐시에 저장될 수도 있다. 텍스처 버퍼 (58) 는 텍스처 맵핑 모듈 (54) 에 통신으로 커플링된다.

[0091]

목적지 알파 버퍼 (60) 는 목적지 알파 결정 모듈 (56) 에 의해 생성된 목적지 알파값들을 저장하고, 추가 프로세싱을 위해 저장된 목적지 알파값들을 목적지 알파 테스트 모듈 (52) 및 목적지 알파 결정 모듈 (56) 에 제공하도록 구성된다. 목적지 알파 버퍼 (60) 는, 목적지 알파값들의 각각이 스크린 스페이스에서 특정 픽셀 로케이션에 대응하는 복수의 목적지 알파값들을 포함할 수도 있다.

[0092]

일부 예들에서, 목적지 알파 버퍼 (60) 는 그래픽스 이미지를 형성하는 목적지 픽셀들에 대한 목적지 알파값들 및/또는 컬러값들을 포함하는 프레임 버퍼 (예를 들어, 도 2의 프레임 버퍼 (40)) 일 수도 있다. 예를 들어, 프레임 버퍼는 복수의 목적지 픽셀들에 대한 적, 녹, 청 및 알파 (RGBA) 성분들 저장할 수도 있다. 특정 목적지 픽셀에 대한 RGBA 성분들 중 "A" 성분은 그 픽셀에 대한 목적지 알파값에 대응할 수도 있다. 일부 경우, 프레임 버퍼는 컬러 서브 버퍼 및 목적지 알파 서브 버퍼로 세분화될 수도 있고, 컬러 데이터 (예를 들어, RGB) 는 컬러 서브 버퍼에 저장될 수도 있고 목적지 알파값들 (예를 들어, "A" 성분) 은 목적지 알파 서브 버퍼에 저장될 수도 있다.

[0093]

추가 예들에서, 목적지 알파 버퍼 (60) 는 그래픽스 이미지들의 목적지 픽셀들의 서브 세트에 대한 목적지 알파값들 및/또는 컬러값들을 포함하는 빈 버퍼일 수도 있다. 빈 버퍼는 프레임 버퍼에 관하여 상술한 바와 같이 목적지 픽셀들의 서브 세트에 대한 유사한 데이터를 저장할 수도 있다. 빈 버퍼는, 일부 예들에서, 프레임 버퍼에 액세스하는데 사용된 통신 인터페이스보다 높은 대역폭을 갖는 통신 인터페이스를 통해 목적지 알파 테스트 모듈 (52) 에 액세스가능할 수도 있다.

[0094]

목적지 알파 테스트 모듈 (52), 텍스처 맵핑 모듈 (54) 및 목적지 알파 결정 모듈 (56) 에 의해 수신된 각각의 소스 픽셀은 스크린 스페이스에서 특정 픽셀 로케이션에 대응한다. 예를 들어, 각각의 소스 픽셀은 픽셀에 대응하는 픽셀 로케이션을 표시하는 포지션 좌표들을 포함할 수도 있다. 따라서, 각각의 소스 픽셀은 목적지 알파 버퍼 (60) 에서 대응 목적지 픽셀 및 목적지 알파값에 맵핑한다.

[0095]

일부 예들에서, 목적지 알파 버퍼 (60) 는 텍스처 버퍼 (58) 의 유효 판독 대역폭보다 높은 유효 판독 대역폭을 가질 수도 있다. 예를 들어, 텍스처 버퍼 (58) 는 하나 이상의 버스들 (예를 들어, 도 1에서 버스들 (20 및/또는 22)) 에 의해 GPU (12) 에 커플링된 오프 칩 메모리일 수도 있고, 목적지 알파 버퍼 (60) 는 하나 이상의 버스들을 통해 GPU (12) 에 반드시 커플링되지 않는 온 칩 캐시일 수도 있다. 텍스처 맵핑 모듈 (54) 이전에 소스 픽셀들을 폐기함으로써, 텍스처 버퍼 (58) 로부터 수행되어야 하는 소스 판독들의 수가 감소됨으로써, 그래픽스 프로세싱 파이프라인 (50) 에 대해 필요한 전체 메모리 액세스 대역폭을 감소시킬 수도 있다.

- [0096] 픽셀 프로세싱 파이프라인 (50) 은, 일부 예들에서, 픽셀 레벨과 반대인 샘플 레벨에서 프로세싱을 수행할 수도 있다. 그러한 예들에서, 각 픽셀은 복수의 샘플들과 연관될 수도 있고, 각 샘플은 단일 픽셀과 연관될 수도 있다. 게다가, 각각의 샘플은 목적지 알파 버퍼 (60) 에서 그 자신의 목적지 알파값과 연관될 수도 있다. 소스 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼 (58) 로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하기 위해서, 목적지 알파 테스트 모듈 (52) 은 소스 픽셀의 각 샘플과 연관된 목적지 알파값을 임계값과 비교하고, 소스 픽셀과 연관된 샘플들 모두에 대한 목적지 알파값들이 임계값 이하인지 여부를 결정할 수도 있다. 소스 픽셀과 연관된 샘플들 모두에 대한 목적지 알파값들이 임계값 이하이면, 목적지 알파 테스트 모듈 (52) 은 소스 픽셀과 연관된 모든 샘플들을 폐기할 수도 있다. 그렇지 않고, 소스 픽셀들과 연관된 샘플들 중 적어도 하나에 대한 목적지 알파값들이 임계값 이하가 아니면, 목적지 알파 테스트 모듈 (52) 은 추가 프로세싱을 위해 후속 프로세싱 스테이지들의 세트에 소스 픽셀과 연관된 샘플들 모두를 패스할 수도 있다.
- [0097] 다중 소스 픽셀들은 동일한 스크린 픽셀 로케이션과 연관될 수도 있는데, 이는 다중 중첩 표면들 및/또는 프리미티브들이 단일 그래픽스 이미지들에 대해 렌더링될 수도 있기 때문이다. 본 명세서에서 사용된 바와 같이, 양 표면들이 스크린 스페이스에서 동일한 픽셀 로케이션에 맵핑하는 적어도 하나의 픽셀을 갖는 경우, 일 표면이 다른 표면 내에서 중첩할 수도 있다. 특정 스크린 픽셀 로케이션에 대한 목적지 알파값은 특정 스크린 로케이션에 대응하는 소스 픽셀이 목적지 알파 결정 모듈 (56) 에 의해 프로세싱될 때마다 증가하여 업데이트될 수도 있다. 업데이트된 목적지 알파값은 동일한 픽셀 로케이션에 대한 이전의 목적지 알파값에 의존할 수도 있다. 따라서, 목적지 알파 버퍼 (60) 에 저장된 목적지 알파값은 주어진 시간의 지점까지 그 목적지 픽셀에 대한 목적지 알파 결정 모듈 (56) 에 의해 프로세싱된 모든 픽셀들에 기초하여 주어진 시간의 지점에서 목적지 알파의 불투명도를 표시하는 "컴포지트" 알파값을 나타낸다.
- [0098] 일부 예들에서, 목적지 알파값이 수신된 소스 픽셀에 대한 목적지 알파 테스트 모듈 (52) 에 의해 추출될 때, 픽셀 프로세싱 파이프라인 (50) 에 의해 여전히 프로세싱되고 있고 목적지 알파 결정 모듈 (56) 에는 아직 도달하지 않은 다른 소스 픽셀들이 있을 수도 있다. 이로써, 특정 소스 픽셀에 대한 목적지 알파 테스트 모듈 (52) 에 의해 추출된 목적지 알파값은 현재의 소스 픽셀 이전에 픽셀 프로세싱 파이프라인 (50) 에서 프로세싱을 시작한 소스 픽셀마다 기여를 반영하지 않을 수도 있다. 즉, 목적지 알파 테스트 모듈 (52) 에 의해 추출된 목적지 알파값은 "스테일" 또는 "올드" 목적지값일 수도 있다. 그럼에도 불구하고, 이른바 "스테일" 목적지 알파값이 추출된다고 하더라도, 그러한 목적지 알파값이 목적지 픽셀이 불투명도의 임계 레벨에 도달한 것을 표시하는 경우, 목적지 알파 테스트 모듈 (52) 은 여전히 안전하게 소스 픽셀을 폐기할 수도 있는데, 이는 목적지 알파값이 불투명도의 임계 레벨을 달성하면, 목적지 알파값이 동일한 프레임을 렌더링 하면서, 후속하여 덜 불투명해지기 때문이다. 따라서, 본 개시물의 기법은 가장 최근의 목적지 알파값이 추출되지 않게 되더라도, 시스템 대역폭을 감소시키는데 있어서 여전히 효과적인 수도 있다.
- [0099] 본 개시물에서 목적지 알파 테스트 기법들이 그래픽스 프로세서들에서 보통 발견되는 소스 알파 테스트들과 상이하고, 그래픽스 프로세서들에서 보통 발견되는 알파 블렌딩 함수와 상이함을 또한 유념해야 한다. 소스 알파 테스트는 프로세싱되는 각 소스 픽셀에 포함된 소스 알파값에 기초하여 픽셀들을 폐기할 수도 있다. 주어진 시점까지 주어진 픽셀 로케이션에 대해 프로세싱된 모든 픽셀들에 대한 컴포지션 알파값인 목적지 알파값과 달리, 소스 알파값은 각 소스 픽셀에 대해 특정되고 컴포지션 알파값이 아닌 알파값이다. 게다가, 목적지 알파 테스트 모듈 (52) 에 의해 사용된 목적지 알파값이 목적지 알파 버퍼 (60), 예를 들어 프레임 버퍼 또는 빈 버퍼로부터 추출되고, 소스 알파 테스트들에 의해 사용된 소스 알파값들은 통상 수신된 소스 픽셀 그 자체 내에 포함된다. 이에 따라, 종래 소스 알파 테스트에 사용된 소스 알파값들은 통상 이전 프로세싱 스테이지로부터 직접 수신되고 프레임 버퍼 또는 빈 버퍼로부터 추출되지 않는다.
- [0100] 종래 기술의 소스 알파 테스트 블록 및 종래 기술의 알파 블렌딩 블록이 본 개시물에 기재된 목적지 알파 테스트 모듈과 상이하지만, 소스 알파 테스트 블록 및 알파 블렌딩 블록 중 하나 또는 양자가 본 개시물의 목적지 알파 테스트 모듈과 함께 사용될 수도 있다. 전방-후방 렌더링 순서가 사용되는 경우, 알파 블렌딩 블록은 페인터의 알고리즘에 따라 렌더링될 때 통상 사용되는, 후방-전방 알파 블렌딩 방정식 대신 전방-후방 알파 블렌딩 방정식을 사용하도록 구성될 수도 있다.
- [0101] 도 4는 도 3의 픽셀 프로세싱 파이프라인 (50) 을 구현하는데 사용될 수도 있는 일 예의 픽셀 프로세싱 파이프라인 (62) 을 도시하는 블록 다이어그램이다. 픽셀 프로세싱 파이프라인 (62) 은 목적지 알파 테스트 모듈 (52), 텍스처 맵핑 모듈 (54), 목적지 알파 결정 모듈 (56), 텍스처 버퍼 (58), 빈 버퍼 (64) 및 프레임 버퍼 (66) 를 포함한다. 도 4에 나타난 바와 같이, 픽셀 프로세싱 파이프라인 (62) 의 아키텍처는, 목적지 알파 버퍼 (60) 가 빈 버퍼 (64) 및 프레임 버퍼 (66) 로 대체된 것을 제외하고, 도 3에 나타난 픽셀 프로세싱 파이프라인 (50) 과 유사하다.

프라인 (50) 과 유사하다. 도 3 및 도 4 에서 유사하게 넘버링된 컴포넌트들은 동일하거나 유사한 기능 및 구성을 갖는다. 따라서, 간결성을 위해 그리고 중복을 피하기 위해, 이러한 공유된 컴퓨넌트들의 동작 및 구성이 더 상세하게 설명되지 않을 것이다.

- [0102] 빈 버퍼 (64) 는 프레임 버퍼 (66) 에서 목적지 픽셀들의 서브세트에 관하여 도 3에서 목적지 알파 버퍼 (60) 의 기능성을 구현할 수도 있다. 프레임 버퍼 (66) 는 도 2에 관하여 도시되고 기재된 프레임 버퍼 (40) 에 대응할 수도 있다. 프레임 버퍼 (66) 는 스크린 스페이스에 목적지 픽셀들 각각에 대한 컬러 성분들 및 목적지 알파값을 저장할 수도 있다. 예를 들어, 프레임 버퍼 (66) 는 목적지 픽셀들 각각에 대한 RGBA 성분들을 저장할 수도 있고, RGBA 의 알파 ("A") 성분은 목적지 알파 테스트 모듈 (52) 에 의해 사용된 목적지 알파값들에 대응할 수도 있다. 프레임 버퍼 (66) 가 시스템 메모리 (10) 에 상주할 수도 있다. 일부 경우, 프레임 버퍼 (66) 는 컬러 서브 버퍼 및 목적지 알파 서브 버퍼로 세분화될 수도 있고, 컬러 데이터 (예를 들어, RGB) 는 컬러 서브 버퍼에 저장될 수도 있으며, 목적지 알파값들 (예를 들어, "A" 성분) 은 목적지 알파 서브 버퍼에 저장될 수도 있다.
- [0103] 빈 버퍼 (64) 는 픽셀들의 서브 세트에 대한 목적지 알파값 및 컬러 성분들을 프레임 버퍼 (66) 에 저장할 수도 있다. 예를 들어, 빈 버퍼 (64) 는 목적지 픽셀들의 특정화된 사이즈, 예를 들어 16 x 16 타일의 픽셀들의 타일 내에 목적지 픽셀들에 대한 RGBA 성분들을 저장할 수도 있다. 일부 예들에서, 빈 버퍼 (64) 는 온 칩 메모리일 수도 있다. 즉, 빈 버퍼 (64) 는 GPU (12) 의 픽셀 프로세싱 파이프라인 (62) 과 동일한 마이크로칩 상에 상주할 수도 있다. 그러한 예들에서, 프레임 버퍼 (40) 는 오프 칩 메모리일 수도 있으며, 예를 들어 프레임 버퍼 (40) 가 GPU (12) 및 빈 버퍼 (64) 가 상주하는 마이크로칩과 상이한 제 2 마이크로칩 상에 상주할 수도 있다.
- [0104] 빈 버퍼 (64) 의 초기화 및 빈 버퍼로부터의 데이터를 프레임 버퍼 (66) 에 기록하는 것이, 일부 예들에서, 프리미티브들의 타일들 또는 빈들의 프로세싱과 동기화될 수도 있다. 예를 들어, 프리미티브들의 새로운 빈이 대략 프로세싱될 때, 픽셀 프로세싱 파이프라인 (62) 은 빈 버퍼로 하여금 이전에 프로세싱된 프리미티브들에 대해 빈 버퍼 (64) 에 저장된 목적지 픽셀 데이터를 프레임 버퍼 (66) 에 기록하게 하고, 빈 버퍼의 데이터값들을 디폴트값들로 초기화할 수도 있다.
- [0105] 일부 예들에서, 목적지 알파 테스트 모듈 (52) 은 제 1 통신 인터페이스를 통해 빈 캐시 (64) 로부터 목적지 알파값들을 추출할 수도 있고, 텍스처 맵핑 모듈 (54) 은 제 2 통신 인터페이스를 통해 텍스처 버퍼 (58) 로부터 텍스처값들을 추출할 수도 있으며, 프레임 버퍼 (66) 는 제 3 통신 인터페이스를 통해 픽셀 프로세싱 파이프라인 (64) 에 액세스가능할 수도 있다. 그러한 예들에서, 제 1 통신 인터페이스는, 일부 예들에서, 제 2 통신 인터페이스 및/또는 제 3 통신 인터페이스보다 높은 대역폭을 가질 수도 있다. 일부 경우, 제 2 통신 인터페이스는, 예를 들어 프레임 버퍼 (40) 및 텍스처 버퍼 (42) 의 양자가 시스템 메모리 (10) 에서 구현될 때, 제 3 통신 인터페이스와 동일한 통신 인터페이스일 수도 있다. 제 2 통신 인터페이스 및 제 3 통신 인터페이스는, 일부 예들에서, 도 1 의 버스들 (20 및/또는 22) 에 대응할 수도 있다. 빈 버퍼 (64) 가 온 칩 빈 버퍼일 때, 제 2 통신 인터페이스는 GPU (12) 내부에 있는 통신 인터페이스일 수도 있다.
- [0106] 부가 예들에서, 빈 버퍼 (64) 의 용량은, 그래픽스 이미지와 연관된 복수의 목적지 픽셀들 모두를 저장하도록 구성될 수도 있는, 프레임 버퍼 (66) 의 용량 미만일 수도 있다. 따라서, 그러한 예들에서, 빈 버퍼 (64) 의 용량은 그래픽스 이미지와 연관된 복수의 목적지 픽셀들 모두에 대한 픽셀 데이터를 저장하여야 하는 최소 용량 미만일 수도 있다.
- [0107] 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들은 특히 도 4에 나타난 비닝 아키텍처의 컨텍스트에서 유용할 수도 있는데, 이는 특정 빈에 대한 목적지 알파값들이 목적지 알파 테스트 모듈 (52) 에 의해 필요로 할 때 이미 빈 버퍼 (64) 에 로딩될 수도 있다. 이것은 낮은 유효 대역폭을 갖는 통신 인터페이스를 통해 액세스 가능한 메모리, 예를 들어 시스템 메모리 (10) 에 로케이팅될 수도 있는, 프레임 버퍼 (66) 로부터 그러한 값들에 액세스할 필요성을 완화시킬 수도 있다. 따라서, 예를 들어 도 4 에 나타난 바와 같이, 비닝 컨텍스트에서 본 개시물의 기법들을 구현함으로써, 목적지 알파값들을 추출할 필요가 있을 수도 있는 오버헤드 대역폭이 텍스처 버퍼 (58) 에 대한 소스 판독들의 수를 감소시킴으로써 달성된 대역폭 절약들에 비해 상대적으로 최소일 수도 있다.
- [0108] 도 5 는 본 개시물의 기법들을 구현하는데 사용될 수 있는 일 예의 목적지 알파 테스트 모듈 (68) 을 도시하는 블록 다이어그램이다. 일부 예들에서, 목적지 알파 테스트 모듈 (68) 은 도 3 및 도 4 에 도시된 목적지 알파 테스트 모듈 (52) 에 대응할 수도 있다. 목적지 알파 테스트 모듈 (68) 은 하나 이상의 소스 픽셀들을

수신하고, 수신된 픽셀들 상에서 목적지 알파 테스트를 수행하며, 목적지 알파 테스트의 결과들에 기초하여 개별 픽셀들을 선택적으로 폐기하도록 구성된다. 목적지 알파 테스트 모듈 (68) 은 목적지 알파 추출 모듈 (70), 비교 모듈 (72), 및 임계 레지스터 (74) 를 포함한다.

[0109] 목적지 알파 추출 모듈 (70) 은 포지션 좌표 데이터를 포함하는 소스 픽셀을 수신하고, 포지션 좌표 데이터에 기초하여 소스 픽셀의 포지션 좌표들을 결정하고, 소스 픽셀의 포지션 좌표들에 기초하여 목적지 알파 버퍼 (60) 로부터 목적지 알파값을 추출하고, 비교를 위해 비교 모듈 (72) 에 추출된 목적지 알파값을 제공하도록 구성될 수도 있다. 추출된 목적지 알파값은 소스 픽셀과 동일한 포지션 좌표들을 가질 수도 있다.

[0110] 비교 모듈 (72) 은 목적지 알파 추출 모듈 (70) 로부터의 목적지 알파값 및 임계 레지스터 (74) 로부터의 임계값의 수신하고, 목적지 알파값을 임계값과 비교하도록 구성될 수도 있다. 비교에 기초하여, 비교 모듈 (72) 은 하나 이상의 프로세싱 스테이지들로 소스 픽셀을 패스할 수도 있고 또는 그 픽셀을 폐기, 즉 어떠한 후속 프로세싱 스테이지들로도 소스 픽셀을 패스하지 않을 수도 있다. 일부 예들에서, 비교 모듈 (72) 은 목적지 알파값이 임계 이하인지 여부를 결정함으로써 목적지 알파값을 임계와 비교할 수도 있다. 목적지 알파값이 임계 이하이면, 비교 모듈 (72) 은 일부 예들에서 소스 픽셀을 폐기할 수도 있다. 반대로, 목적지 알파값이 임계 이하가 아니면, 비교 모듈 (72) 은 일부 예들에서 하나 이상의 후속 프로세싱 스테이지들로 픽셀을 패스, 즉 소스 픽셀을 폐기하지 않을 수도 있다. 다른 임계 비교 함수들이 다른 예들에서 사용될 수도 있다. 이러한 방식으로 목적지 알파 테스트 모듈 (68) 은 소스 픽셀들에 대응하는 목적지 알파값들에 기초하여 소스 픽셀들을 선택적으로 폐기할 수도 있다.

[0111] 임계 레지스터 (74) 는 불투명도의 임계 레벨을 표시하는 임계값을 저장할 수도 있으며, 임계 레벨에서 또는 그 아래에서 소스 픽셀이 폐기될 수도 있다. 예를 들어, 목적지 알파 테스트 모듈 (68) 이 특정 스크린 픽셀 로케이션을 갖는 소스 픽셀을 프로세싱할 때, 불투명도의 임계 레벨은 대응하는 목적지 픽셀에 대한 불투명도의 레벨을 표시할 수도 있으며, 레벨에서 또는 레벨 아래에서 소스 픽셀은 소스 픽셀이 디스플레이될 필요가 없도록 가시적이지 않게 된다. 도 5에 나타난 바와 같이, 임계 레지스터 (74) 는 특정 임계값을 표시하는 정보를 포함할 수도 있는 구성 정보를 수신하여 임계 레지스터 (74) 에 저장하도록 구성된다. 임계 레지스터 (74) 는 구성 정보를, 예를 들어 호스트 CPU 상에서 실행하는 사용자 소프트웨어 어플리케이션, 호스트 CPU 상에서 실행하는 GPU 드라이버, GPU 상에 상주하는 하드웨어 컴포넌트 및/또는 GPU 상에서 실행하는 셰이더로부터 수신할 수도 있다.

[0112] 도 6 은 본 개시물의 기법들을 수행하는데 사용될 수도 있는 또 다른 예의 목적지 알파 테스트 모듈 (76) 을 도시하는 블록 다이어그램이다. 일부 예들에서, 목적지 알파 테스트 모듈 (76) 은 도 3 및 도 4 에 도시된 목적지 알파 테스트 모듈 (52) 을 대체할 수도 있다. 목적지 알파 테스트 모듈 (76) 은 하나 이상의 소스 픽셀들을 수신하고, 소스 픽셀들 상에서 목적지 알파 테스트를 수행하고, 비교에 응답하여 제어 신호를 선택적으로 활성화시키도록 구성된다. 목적지 알파 테스트 모듈 (68) 은 목적지 알파 추출 모듈 (70), 비교 모듈 (78) 및 임계 레지스터 (74) 를 포함한다. 도 6 에 나타난 바와 같이, 목적지 알파 테스트 모듈 (76) 의 아키텍처는 비교 모듈 (72) 이 비교 모듈 (78) 로 대체되고, 모든 소스 픽셀들이 후속 프로세싱 스테이지로 패스되는 것을 제외하고, 도 5 에 나타난 목적지 알파 테스트 모듈 (68) 과 유사하다. 도 5 및 도 6 에서 유사하게 넘버링된 컴포넌트들은 동일하거나 유사한 기능성 및 구성을 갖는다. 따라서, 간결성을 위해 그리고 중복을 회피하기 위해, 이러한 공유된 컴포넌트들의 동작 및 구성은 더 상세하게 설명되지 않을 것이다.

[0113] 비교 모듈 (78) 은 도 5 에서 비교 모듈 (72) 에 관하여 상술한 바와 유사한 비교 동작들을 수행할 수도 있다. 하지만, 비교 동작들에 응답하여 소스 픽셀들을 선택적으로 폐기하는 대신, 비교 모듈 (78) 은 제어 신호를 선택적으로 활성화 및 비활성화시킬 수도 있다. 제어 신호는 대응 소스 픽셀에 대한 텍스처값들의 추출을 선택적으로 인에이블 및 디스에이블시키기 위해 픽셀 프로세싱 파이프라인에서 하나 이상의 후속 프로세싱 스테이지들에 공급될 수도 있다. 예를 들어, 목적지 알파 테스트 모듈 (76) 에서 소스 픽셀을 폐기하는 대신, 또 다른 프로세싱 스테이지가 제어 신호에 기초하여 소스 픽셀을 폐기할 수도 있다. 또 다른 예에서, 텍스처 맵핑 모듈은 제어 신호를 수신하고, 제어 신호에 기초하여 소스 픽셀에 대한 텍스처값 추출을 선택적으로 인에이블 또는 디스에이블할 수도 있다.

[0114] 도 6 에서 예시의 목적지 알파 테스트 모듈 (76) 은, 도 5 에 도시된 구성 대신에 또는 그 구성에 추가하여 픽셀 프로세싱 파이프라인에서 텍스처값들의 추출을 선택적으로 제어하기 위해 사용될 수도 있는 또 다른 예의 목적지 알파 테스트 모듈 구성을 도시한다. 다른 목적지 알파 테스트 모듈 구성들이 고려되며 본 개시물의 범위 내에 있다. 예를 들어, 또 다른 예시의 목적지 알파 테스트 모듈 구성은 텍스처값들이 소스 픽셀에 대해

취출되어야 하는지를 표시하는 픽셀 속성을 변경할 수도 있고, 텍스처 맵핑 모듈은 픽셀 속성에 기초하여 소스 픽셀에 대한 텍스처값들을 선택적으로 취출할 수도 있다.

- [0115] 일부 예들에서, 본 개시물의 기법들에 따라 설계된 픽셀 프로세싱 파이프라인은 스텐실 테스트를 수행하는 프로세싱 유닛을 포함할 수도 있다. 프로세싱 유닛은, 예를 들어, 스텐실 테스트 블록 및/또는 결합된 z/스텐실 테스트 블록일 수도 있다. 그러한 예들에서, GPU 는 일부 예들에서 스텐실값 대신에 목적지 알파값을 수신하도록 스텐실 테스트를 수행하는 프로세싱 유닛을 구성함으로써 본 개시물에 기재된 목적지 알파 테스트 기능을 구현할 수도 있다. 그 후, 프로세싱 유닛은 목적지 알파 테스트를 수행하고, 선택적으로 텍스처값이 본 개시물의 기법들에 따라 목적지 알파값에 기초하여 소스 픽셀에 대해 취출되게 하기 위해 사용될 수도 있다. 이러한 기법들은 도 7 내지 도 10 에 관하여 더 상세하게 설명된다.

- [0116] 도 7 은 목적지 알파 테스트를 수행하고 본 개시물의 선택적 텍스처값 취출 기법들을 구현하기 위해 구성가능한 스텐실 세스팅 블록을 사용하는 일 예의 픽셀 프로세싱 파이프라인 (80) 을 도시하는 블록 다이어그램이다. 픽셀 프로세싱 파이프라인 (80) 은 목적지 알파 테스트 모듈 (82), 스텐실 버퍼 (84) 및 목적지 알파 버퍼 (60) 를 포함한다. 목적지 알파 테스트 모듈 (82) 은 구성가능한 스텐실 테스트 블록 (86) 을 포함한다. 스텐실 버퍼 (84) 는 픽셀 프로세싱 파이프라인 (80) 에 대한 스텐실값들을 저장하도록 구성될 수도 있다. 일부 예들에서, 스텐실 버퍼 (84) 는 오프 칩 시스템 메모리, 예를 들어 도 10 에 나타난 시스템 메모리 (10) 에 상주할 수도 있다. 추가 예들에서, 스텐실 버퍼 (84) 는 도 9 에 관하여 기재된 z/스텐실 버퍼와 유사한 z/스텐실 버퍼일 수도 있고, 프레임 버퍼에서 목적지 픽셀들과 연관된 스텐실값들 및 z값들을 저장할 수도 있다. 목적지 알파 버퍼 (60) 는, 예를 들어 프레임 버퍼 또는 빈 버퍼일 수도 있다. 목적지 알파 버퍼 (60) 가 빈 버퍼일 때, 목적지 알파 버퍼 (60) 는 일부 예들에서, 픽셀 프로세싱 파이프라인 (80) 을 포함하는 GPU 와 동일한 마이크로칩 상에 상주할 수도 있다.

- [0117] 목적지 알파 테스트 모듈 (82) 은 GPU 에서 이전의 프로세싱 스테이지로부터 소스 픽셀을 수신하고, 소스 픽셀에 대응하는 목적지 알파값을 취출하고, 목적지 알파값을 임계와 비교하고, 비교 결과에 기초하여 소스 픽셀을 선택적으로 폐기하도록 구성될 수도 있다. 도 7 에 나타난 바와 같이, 목적지 알파 테스트 모듈 (82) 은 구성가능한 스텐실 테스트 블록 (86) 으로서 구현된다. 구성가능한 스텐실 테스트 블록 (86) 은 스텐실 세트링 스테이지에서 그리고 목적지 알파 테스트 상태에서 동작하도록 구성가능할 수도 있다. 스텐실 테스트 상태에서 동작하도록 구성될 때, 스텐실 테스트 블록 (86) 은 이전의 프로세싱 스테이지로부터 소스 픽셀을 수신하고, 스텐실 버퍼 (84) 로부터 수신된 소스 픽셀에 대응하는 스텐실값을 취출하고, 취출된 스텐실값을 임계와 비교하고, 그리고 비교 결과에 기초하여 소스 픽셀을 선택적으로 폐기할 수도 있다. 목적지 알파 테스트 상태에서 동작하도록 구성될 때, 스텐실 테스트 블록 (86) 은 스텐실 버퍼 (84) 로부터 대응 스텐실값을 취출하는 대신, 스텐실 테스트 블록 (86) 이 목적지 알파 버퍼 (60) 로부터 대응 목적지 알파값을 취출할 수도 있는 것을 제외하고, 스텐실 테스트 상태에서 구성될 때와 본질적으로 동일한 동작들을 수행할 수도 있다. 따라서, 스텐실 테스트 블록 (86) 은 목적지 알파값에 기초하여, 소스 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 취출되지 않게 하는 액션을 수행할지 여부를 결정하고, 본 개시물의 기법들에 따라 액션을 수행하도록 결정하는 것에 응답하여 액션을 수행할 수도 있다.

- [0118] 스텐실 테스트 블록 (86) 은 스텐실 테스트 블록 (86) 에 의해 수신된 구성 정보에 기초하여 구성 상태들 사이에서 스위칭하도록 구성될 수도 있다. 구성 정보는 GPU 내의 또 다른 컴포넌트로부터, 호스트 CPU 상에서 실행하는 GPU 드라이버로부터, 및/또는 호스트 CPU 상에서 실행하는 사용자 소프트웨어 어플리케이션으로부터 수신될 수도 있다.

- [0119] 일부 예들에서, 스텐실 테스트 블록 (86) 에 의해 수신된 구성 정보는 스텐실 테스트 블록 (86) 이 입력값들 (예를 들어, 스텐실값들 및/또는 목적지 알파값들) 을 취출하여야 하는 메모리 어드레스를 특정할 수도 있다. 그러한 예들에서, 스텐실 테스트 블록 (86) 은 스텐실 버퍼 (84) 를 가리키는 스텐실 테스트 블록 (86) 에 메모리 어드레스를 제공함으로써 스텐실값들을 수신하도록 구성될 수도 있다. 게다가, 스텐실 테스트 블록 (86) 은 스텐실 버퍼 (84) 대신 목적지 알파 버퍼 (60) 를 가리키는 스텐실 테스트 블록 (86) 에 메모리 어드레스를 제공함으로써 스텐실값들 대신에 목적지 알파값들을 수신하도록 구성될 수도 있다.

- [0120] 본 개시물의 기법들에 따라 스텐실 알파값들 대신에 목적지 알파값들을 취출하도록 스텐실 테스트 블록을 재구성함으로써, 하드웨어 기반 목적지 알파 테스트 블록은, 그래픽스 파이프라인에 부가 하드웨어 블록들을 반드시 부가할 필요없이도 그래픽스 프로세싱 파이프라인에서 구현될 수도 있다. 또한, 그래픽스 프로세서는 2 개의 상이한 렌더링 모드들 - 하드웨어 기반 스텐실 테스트를 수행하는 제 1 모드 및 하드웨어 기반 목적지 알파

테스팅을 수행하는 제 2 모드 사이에서 선택적으로 스위칭할 수도 있다.

- [0121] 도 8 은 도 7 의 픽셀 프로세싱 파이프라인을 구현하는데 사용될 수도 있는 일 예의 스텐실 테스트 블록 (86) 을 도시하는 블록 다이어그램이다. 스텐실 테스트 블록 (86) 은 어드레스 레지스터 (88), 임계 레지스터 (90), 입력값 추출 모듈 (92) 및 비교 모듈 (94) 을 포함한다.
- [0122] 어드레스 레지스터 (88) 는 입력값 추출 모듈 (92) 이 입력값들을 추출해야 하는 메모리 어드레스를 저장하도록 구성될 수도 있다. 사용자 어플리케이션, GPU 드라이버 및/또는 GPU 컴포넌트는 스텐실 버퍼 (84) 에 대한 어드레스 공간을 가리키는 어드레스 레지스터 (88) 에 어드레스값을 로딩함으로써 스텐실값들을 추출하도록 스텐실 테스트 블록 (86) 을 구성할 수도 있다. 유사하게, 사용자 어플리케이션, GPU 드라이버 및/또는 GPU 컴포넌트는 목적지 알파 버퍼 (60) 에 대한 어드레스 공간을 가리키는 어드레스 레지스터 (88) 에 어드레스값을 로딩함으로써 목적지 알파값들을 추출하도록 스텐실 테스트 블록 (86) 을 구성할 수도 있다.
- [0123] 임계 레지스터 (90) 는 비교 모듈 (94) 에 의한 사용을 위해 임계값을 저장할 수도 있다. 스텐실 테스트 블록 (86) 이 스텐실 테스트를 수행하도록 구성될 때, 임계값은 스텐실 테스트를 위한 레퍼런스값을 표시할 수도 있다. 스텐실 테스트 블록 (86) 이 목적지 알파 테스트를 수행하도록 구성될 때, 임계값은 불투명도의 임계 레벨을 수도 있으며, 이 임계 레벨에서 또는 그 아래에서 소스 픽셀이 폐기될 수도 있다. 사용자 어플리케이션, GPU 드라이버 및/또는 GPU 컴포넌트는 임계 레지스터 (90) 에 임계값을 로딩함으로써 특정 임계 레벨을 저장하도록 임계 레지스터 (90) 를 구성할 수도 있다.
- [0124] 입력값 추출 모듈 (92) 은 포지션 좌표 데이터를 포함하는 소스 픽셀을 수신하고, 포지션 좌표 데이터에 기초하여 소스 픽셀의 포지션 좌표들을 결정하고, 포지션 좌표들에 기초하여 메모리 또는 캐시로부터 입력값을 추출하고, 그리고 비교를 위해 비교 모듈 (94) 에 입력값을 제공하도록 구성될 수도 있다. 수신된 입력값은 소스 픽셀과 동일한 포지션 좌표들과 연관될 수도 있다. 입력값 추출 모듈 (92) 은 어드레스 레지스터 (88) 에 저장된 메모리 어드레스에 의해 적어도 부분적으로 결정되는 메모리 로케이션 및/또는 캐시 로케이션으로부터 입력값을 추출할 수도 있다. 어드레스 레지스터 (88) 에 저장된 메모리 어드레스가 스텐실 버퍼 (84) 를 가리킬 때, 입력값 추출 모듈 (92) 은 소스 픽셀에 대응하는 스텐실값을 추출하고, 비교를 위해 비교 모듈 (94) 에 스텐실값을 제공할 수도 있다. 유사하게, 어드레스 레지스터 (88) 에 저장된 메모리 어드레스가 목적지 알파 버퍼 (60) 를 가리킬 때, 입력값 추출 모듈 (92) 은 소스 픽셀에 대응하는 목적지 알파값을 추출하고, 비교를 위해 비교 모듈 (94) 에 목적지 알파값을 제공할 수도 있다.
- [0125] 비교 모듈 (94) 은 입력 추출 모듈 (92) 로부터의 입력값 및 임계 레지스터 (90) 로부터의 임계값을 수신하고, 입력값을 임계값과 비교하도록 구성될 수도 있다. 비교에 기초하여, 비교 모듈 (94) 은 소스 픽셀을 하나 이상의 프로세싱 스테이지들로 패스하거나 픽셀을 폐기, 즉 소스 픽셀을 어떠한 후속 프로세싱 스테이지들로 패스하지 않을 수도 있다. 일부 예들에서, 비교 모듈 (94) 은 입력값이 임계 이하인지 여부를 결정함으로써 입력값을 임계치와 비교할 수도 있다. 입력값이 임계 이하라면, 비교 모듈 (94) 은 일부 예들에서 소스 픽셀을 폐기할 수도 있다. 반대로, 입력값이 임계 이하가 아니라면, 비교 모듈 (94) 은 일부 예들에서 픽셀을 하나 이상의 후속 프로세싱 스테이지들로 패스, 즉 소스 픽셀을 폐기하지 않을 수도 있다. 다른 임계 비교 함수들이 다른 예들에서 사용될 수도 있다. 또한, 입력값은 어느 메모리 스페이스 입력값 추출 모듈 (92) 이 데이터를 추출했는지에 의존하여 스텐실값 또는 목적지 알파값에 대응할 수도 있다. 이러한 방식으로, 스텐실 테스트 블록 (86) 은 스텐실 테스트 블록으로서 또는 목적지 알파 테스트 블록으로서 동작하도록 구성가능할 수도 있다.
- [0126] 도 9 는 본 개시물의 선택적 텍스처값 추출 기법들에 따라 목적지 알파 테스트를 수행하도록 구성가능한 z/스텐실 테스트 블록 (106) 을 사용하는 일 예의 픽셀 프로세싱 파이프라인 (100) 을 도시하는 블록 다이어그램이다. 픽셀 프로세싱 파이프라인 (100) 은 목적지 알파 테스트 모듈 (102), z/스텐실 버퍼 (104) 및 목적지 알파 버퍼 (60) 를 포함한다. 목적지 알파 테스트 모듈 (102) 은 결합된, 구성가능한 z/스텐실 테스트 블록 (106) 을 포함한다. Z/스텐실 버퍼 (104) 는 픽셀 프로세싱 파이프라인 (100) 에 대한 z 값들 및 스텐실값들을 저장하도록 구성될 수도 있다. 일부 예들에서, z/스텐실 버퍼 (104) 는 오프 칩 시스템 메모리, 예를 들어 도 1에 나타난 시스템 메모리에 상주할 수도 있다. 목적지 알파 버퍼 (60) 는 예를 들어 프레임 버퍼 또는 빈 버퍼일 수도 있다. 목적지 알파 버퍼 (60) 가 빈 버퍼일 때, 목적지 알파 버퍼 (60) 는 일부 예들에서 픽셀 프로세싱 파이프라인 (100) 을 포함하는 GPU 와 동일한 마이크로칩 상에 상주할 수도 있다.
- [0127] 목적지 알파 테스트 모듈 (102) 은 GPU 에서 이전의 프로세싱 스테이지로부터 소스 픽셀을 수신하고, 소스 픽셀에 대응하는 목적지 알파값을 추출하고, 목적지 알파값을 임계와 비교하고, 그리고 비교의 결과들에 기초하여

소스 픽셀을 선택적으로 폐기하도록 구성된다. 도 9에 나타난 바와 같이, 목적지 알파 테스트 모듈 (102)은 구성가능한 z/스텐실 테스트 블록 (106)으로서 구현된다. 구성가능한 z/스텐실 블록 (106)은 z 스텐실 테스트 상태에서 및 목적지 알파 테스트 상태에서 동작하도록 구성가능할 수도 있다. z 스텐실 테스트 상태에서 동작하도록 구성될 때, z/스텐실 테스트 블록 (106)은 수신된 소스 픽셀 상에서 스텐실 테스트 및 z 테스트 중 하나 또는 양자를 수행할 수도 있다. 스텐실 테스트는 스텐실값들이 스텐실 버퍼 (84) 대신 z/스텐실 버퍼 (104)로부터 추출되는 것을 제외하고, 도 7에서 스텐실 블록 (86)에 관하여 위에 상술하였던 것과 실질적으로 유사한 방식으로 수행된다.

[0128] z 테스트를 수행하기 위해, z/스텐실 테스트 블록 (106)은 이전의 프로세싱 스테이지로부터 소스 픽셀을 수신하고, z/스텐실 버퍼 (104)로부터 수신된 소스 픽셀에 대응하는 z 값을 추출하고, 추출된 z 값을 소스 픽셀에 포함된 소스 z 값과 비교하고, 그리고 비교 결과들에 기초하여 소스 픽셀을 선택적으로 폐기할 수도 있다. 목적지 알파 테스트 상태에서 동작하도록 구성될 때, z/스텐실 버퍼 (104)는 스텐실 테스트에 의해 수행된 것과 본질적으로 동일한 동작들을 수행할 수도 있지만, z/스텐실 버퍼 (104)로부터 대응 스텐실값을 추출하는 대신, z/스텐실 테스트 블록 (106)은 목적지 알파 버퍼 (60)로부터 대응 목적지 알파값을 추출할 수도 있다. 따라서, z/스텐실 테스트 블록 (106)은 목적지 알파값에 기초하여, 소스 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정하고, 본 개시물에 기재된 기법들에 따라 액션을 수행하도록 결정하는 것에 응답하여 액션을 수행할 수도 있다.

[0129] Z/스텐실 테스트 블록 (106)은 z/스텐실 테스트 블록 (106)에 의해 수신된 구성 정보에 기초하여 구성 상태들 사이에서 스위칭하도록 구성될 수도 있다. 구성 정보는 GPU 내의 또 다른 컴포넌트로부터, 호스트 CPU 상에서 실행하는 GPU 드라이버로부터 및/또는 호스트 CPU 상에서 실행하는 사용자 소프트웨어 어플리케이션들로부터 수신될 수도 있다.

[0130] 일부 예들에서, z/스텐실 테스트 블록 (106)에 의해 수신된 구성 정보는 z/스텐실 테스트 블록 (106)이 입력값들 (예를 들어, 스텐실값들 및/또는 목적지 알파값들)을 추출하여야 하는 메모리 어드레스를 특정할 수도 있다. 그러한 예들에서, z/스텐실 테스트 블록 (106)은 z/스텐실 버퍼 (104)를 가리키는 z/스텐실 테스트 블록 (106)에 메모리 어드레스를 제공함으로써 목적지 알파값들을 수신하도록 구성될 수도 있다. 게다가, z/스텐실 테스트 블록 (106)은 z/스텐실 버퍼 (104) 대신 목적지 알파 버퍼 (60)를 가리키는 z/스텐실 테스트 블록 (106)에 메모리 어드레스를 제공함으로써 스텐실값들 및/또는 z 값들 대신에 목적지 알파값들을 수신하도록 구성될 수도 있다.

[0131] 본 개시물의 기법들에 따라 z 스텐실값들 대신 목적지 알파값들을 추출하도록 z 스텐실 테스트 블록을 재구성함으로써, 하드웨어 기반 목적지 알파 테스트 블록은 그래픽스 파이프라인에 부가의 하드웨어 블록들을 반드시 부가할 필요 없이도, 그래픽스 프로세싱 파이프라인에서 구현될 수도 있다. 게다가, 그래픽스 프로세서는 적어도 2개의 상이한 렌더링 모드들: 하드웨어 기반 결합된 z 스텐실 테스트를 수행하는 제 1 모드 및 하드웨어 기반 목적지 알파 테스트를 수행하는 제 2 모드 사이에서 선택적으로 스위칭할 수도 있다.

[0132] 일부 예들에서, z/스텐실 테스트 블록 (106)이 목적지 알파 테스트 모드에서 동작하도록 구성될 때, z/스텐실 테스트 블록 (106)의 z 테스트 부분은, z 테스트에 기인하여 어떠한 픽셀들도 폐기되지 않도록 모든 픽셀들이 z 테스트를 패스하게 하는 "항시 패스" 모드에 배치될 수도 있다. 즉, z 테스트가 효과적으로 디스에이블될 수도 있다. Z/스텐실 테스트 블록 (106)이 이러한 예들에서 "항시 패스" 모드로 배치될 수 있는데, 이는 z/스텐실 테스트 블록 (106)이 목적지 알파 버퍼 (60)로부터 데이터를 수신하도록 재구성될 때, z/스텐실 테스트 블록 (106)이 반드시 z 값들을 수신하지 않을 수도 있다. 오히려, z/스텐실 테스트 블록 (106)은 목적지 알파 버퍼 (60)로부터 컬러값들, 예를 들어 RGB값들을 수신할 수도 있다. z/스텐실 테스트 블록 (106)을 "항시 패스" 모드로 배치하면, z 테스트가 컬러값들에 기초하여 의도치않게 픽셀들을 폐기하지 않은 것을 보장한다.

[0133] 도 10은 도 9의 픽셀 프로세싱 파이프라인을 구현하는데 사용될 수도 있는 일 예의 z/스텐실 테스트 블록 (106)을 도시하는 블록 다이어그램이다. Z/스텐실 테스트 블록 (106)은 어드레스 레지스터 (108), 임계 레지스터 (110), z 테스트 디스에이블 레지스터 (112), 입력값 추출 모듈 (114) 및 비교 모듈 (116)을 포함한다.

[0134] 어드레스 레지스터 (108)는 입력값 추출 모듈 (114)이 입력값들을 추출해야 하는 메모리 어드레스를 저장하도록 구성될 수도 있다. 사용자 어플리케이션, GPU 드라이버 및/또는 GPU 컴포넌트는, z/스텐실 버퍼 (104)에 대한 어드레스 스페이스를 가리키는 어드레스 레지스터 (108)로 어드레스값을 로딩함으로써 z 스텐실값들

(즉, z 값들 및 스텐실값들)을 추출하도록 입력값 추출 모듈 (114)을 구성할 수도 있다. 유사하게, 사용자 어플리케이션, GPU 드라이버 및/또는 GPU 컴포넌트는 목적지 알파 버퍼 (60)에 대한 어드레스 스페이스를 가리키는 어드레스 레지스터 (108)로 어드레스값을 로딩함으로써 목적지 알파값들을 추출하도록 입력값 추출 모듈 (114)을 구성할 수도 있다.

[0135] 임계 레지스터 (110)는 비교 모듈 (116)에 의한 사용을 위한 임계값을 저장할 수도 있다. z /스텐실 테스트 블록 (106)이 z 스텐실 테스트 (즉, z 테스트 및/또는 스텐실 테스트)를 수행하도록 구성될 때, 임계값은 스텐실 테스트를 위한 레퍼런스값을 표시할 수도 있다. z /스텐실 테스트 블록 (106)이 목적지 알파 테스트를 수행하도록 구성될 때, 임계값은 임계 레벨을 표시할 수도 있으며, 그 레벨에서 또는 그 레벨 아래에서 소스 픽셀이 폐기될 수도 있다. 사용자 어플리케이션, GPU 드라이버 및/또는 GPU 컴포넌트는 임계값을 임계 레지스터 (110)로 로딩함으로써 특정 임계 레벨을 저장하도록 임계 레지스터를 구성할 수도 있다.

[0136] Z 테스트 디스에이블 레지스터 (112)는 z 테스트가 비교 모듈 (116)에 대해 인에이블 또는 디스에이블되어야 하는지를 표시하는 값을 저장할 수도 있다. 사용자 어플리케이션, GPU 드라이버 및/또는 GPU 컴포넌트는 z 테스트 디스에이블 레지스터 (112)에 구성값을 로딩함으로써 특정 구성값을 저장하도록 z 테스트 디스에이블 레지스터 (112)를 구성할 수도 있다.

[0137] 입력값 추출 모듈 (114)은 포지션 좌표 데이터를 포함하는 소스 픽셀을 수신하고, 포지션 좌표 데이터에 기초하여 소스 픽셀의 포지션 좌표들을 결정하고, 포지션 좌표들에 기초하여 메모리 또는 캐시로부터 입력값을 추출하고, 그리고 비교를 위해 입력값을 비교 모듈 (116)에 제공하도록 구성될 수도 있다. 추출된 입력값은 소스 픽셀과 동일한 포지션 좌표들과 연관될 수도 있다. 입력값 추출 모듈 (114)은 어드레스 레지스터 (108)에 저장된 메모리 어드레스에 의해 적어도 부분적으로 결정되는 메모리 로케이션 및/또는 캐시 로케이션으로부터 입력값을 추출할 수도 있다. 메모리 레지스터 (108)에 저장된 메모리 어드레스가 z /스텐실 버퍼 (104)를 가리킬 때, 입력값 추출 모듈 (114)은 소스 픽셀에 대응하는 z 스텐실값들을 추출하고 비교를 위해 z 스텐실값들을 비교 모듈 (116)에 제공할 수도 있다. 유사하게, 어드레스 레지스터 (108)에 저장된 메모리 어드레스가 목적지 알파 버퍼 (60)를 가리킬 때, 입력값 추출 모듈 (114)은 소스 픽셀에 대응하는 목적지 알파값을 추출하고 비교를 위해 목적지 알파값을 비교 모듈 (116)에 제공할 수도 있다.

[0138] 비교 모듈 (116)은 2개의 상이한 입력값들을 포함하는 입력값 추출 모듈 (114)로부터 입력 데이터를 수신하고, 입력값들의 각각에 대해 상이한 비교 테스트들을 수행하도록 구성될 수도 있다. 제 1 비교 테스트의 부분으로서, 비교 모듈 (116)은 입력값 추출 모듈 (114)로부터 수신된 제 1 입력값을 임계 레지스터 (110)로부터 수신된 임계값과 비교할 수도 있다. 제 2 비교 테스트의 부분으로서, 비교 모듈 (116)은 입력값 추출 모듈 (114)로부터 수신된 제 2 입력값을 소스 픽셀 내에 포함된 픽셀 속성과 비교할 수도 있다. 이들 비교 테스트들의 양자의 결과들에 기초하여, 비교 모듈 (116)은 소스 픽셀을 하나 이상의 프로세싱 스테이지들로 패스할 수도 있고, 또는 픽셀을 폐기, 즉 소스 픽셀을 임의의 후속 프로세싱 스테이지들로 패스하지 않을 수도 있다. 예를 들어, 소스 픽셀이 비교 테스트들의 양자를 패스하면, 비교 모듈 (116)은 추가 프로세싱을 위한 후속 스테이지로 소스 픽셀을 패스할 수도 있다. 대조적으로, 소스 픽셀이 테스트들 중 적어도 하나를 패스하지 않는 경우, 비교 모듈 (116)은 소스 픽셀을 폐기할 수도 있다. 비교 모듈 (116)은 z 테스트 디스에이블 레지스터 (112)에 저장된 구성값에 기초하여 "항시 패스" 모드로 제 2 비교 테스트를 선택적으로 구성할 수도 있다. 제 2 비교 테스트가 "항시 패스" 모드에 있을 때, 그러면 제 2 비교 테스트는, 수행되었을 경우, 그러한 비교의 실제 결과에 관계없이 모든 픽셀에 대해 패스된다고 고려된다. 따라서, 제 1 비교 테스트는 소스 픽셀이 폐기되게 될지 여부를 결정하게 된다.

[0139] 비교 모듈 (116)이 z 스텐실 테스트 구성에서 동작하도록 구성될 때, 비교 모듈 (116)은 (예를 들어, 도 8에서 비교 모듈 (94)에 관하여 상술하였던 것과 유사한 방식으로 입력값 추출 모듈 (114)로부터 수신된 스텐실값을 임계치와 비교함으로써) 스텐실 테스트를 수행하기 위해 제 1 비교 테스트를 사용하고, (예를 들어, 입력값 추출 모듈 (114)로부터 수신된 z 값을 소스 픽셀 내에 포함된 소스 z 값 속성과 비교함으로써) z 테스트를 수행하기 위해 제 2 비교 테스트를 사용할 수도 있다. 비교 모듈 (116)이 z 스텐실 테스트 구성에서 동작하도록 구성될 때, 비교 모듈 (116)은 (도 8에서 비교 모듈 (94)에 관하여 상술하였던 것과 유사한 방식으로 입력값 추출 모듈 (114)로부터 수신된 목적지 알파값을 임계치와 비교함으로써) 목적지 알파 테스트를 수행하기 위해 제 1 비교 테스트를 사용할 수도 있으며, 비교 모듈 (116)은 z 테스트 디스에이블 레지스터 (112)에 저장된 디스에이블 구성값에 기초하여 제 2 비교를 디스에이블 할 수도 있다. 이러한 방식으로, z /스텐실 테스트 블록 (106)은 z 스텐실 테스트 블록으로서 또는 목적지 알파 테스트 블록으로서 동작하도록 구성가능할

수도 있다.

- [0140] 일부 그래픽스 프로세서들에서, 주어진 픽셀 로케이션에 대한 z 값들 및 스텐실값들은 z /스텐실 버퍼 (104) 에서 단일 데이터로서 저장될 수도 있다. 예를 들어, 데이터 워드는 워드의 24 비트들을 구성하는 z 값 및 잔여 8 비트들을 구성하는 스텐실값을 갖는 32 비트들일 수도 있다. 그러한 그래픽스 프로세서들에서, RGBA 컴포넌트들은 프레임 버퍼에, 그리고 빈 버퍼에서의 확장에 의해, "RGB" 컴포넌트들이 24 비트들을 구성하고 "A" 컴포넌트가 잔여 8 비트들을 구성하는 32 비트 워드들로 저장될 수도 있다. 그러한 예들에서, 32 비트 RGBA 데이터 내의 알파값들의 포지션은 스텐실값이 통상 32 비트 z 값/스텐실 데이터 워드에 저장되는 포지션으로 맵핑할 수도 있다. 즉, 목적지 알파값은 z /스텐실 테스트 블록 (106) 이 통상 스텐실값들을 수신할 것을 예상하는 데이터 워드 내에서 비트 포지션들과 정렬할 수도 있고, RGB 값들은 z /스텐실 테스트 블록 (106) 이 통상 z 값들을 수신할 것을 예상하는 데이터 워드에서 비트 포지션들과 정렬할 수도 있다. 이러한 방식으로, z /스텐실 테스트 블록 (106) 은 z /스텐실 버퍼 (104) 에서 하부에 놓인 데이터를 반드시 재구성할 필요없이 목적지 알파값들 사이에서 동작가능할 수도 있다. 게다가, 목적지 알파 테스트 모드에서 동작할 때 z 테스트 기능을 디스에이블시킴으로써, z 값들 대신 RGB 값들을 수신하는 것으로부터의 간섭이 방지될 수도 있다.
- [0141] 도 11은 본 개시물의 선택적 텍스처값 추출 기법들에 따른 목적지 알파 테스트를 수행하기 위해 프로그램가능한 셰이더 유닛을 사용하는 일 예의 픽셀 프로세싱 파이프라인 (120) 을 도시하는 블록 다이어그램이다. 픽셀 프로세싱 파이프라인 (120) 은 프로그램가능한 셰이더 유닛 (122), 텍스처 맵핑 모듈 (54), 텍스처 버퍼 (58) 및 목적지 알파 버퍼 (60) 를 포함한다. 도 11의 예의 픽셀 프로세싱 파이프라인 (120) 에서, 텍스처 맵핑 모듈 (54) 은 고정 함수 하드웨어로서 구현되고, 목적지 알파 테스트 모듈 (126) 은 프로그램가능한 셰이더 유닛에서 구현된다.
- [0142] 도 11 에 나타난 바와 같이, 픽셀 프로세싱 파이프라인 (120) 의 아키텍처는, 목적지 알파 테스트 모듈 (52) 이 목적지 알파 테스트를 구현하는, 셰이더 유닛 (122) 에 대체되는 것을 제외하고, 도 3 에 나타난 픽셀 프로세싱 파이프라인 (50) 과 유사하다. 도 3 및 도 11 에서 유사하게 넘버링된 컴포넌트들은 동일하거나 유사한 기능성 및 구성을 갖는다. 따라서, 간결성을 위해 그리고 중복을 회피하기 위해, 이러한 공유된 컴포넌트들의 동작 및 구성은 더 상세하게 설명되지 않을 것이다.
- [0143] 셰이더 유닛 (122) 은 셰이더 프로그램 (124) 을 실행하도록 구성되는 하나 이상의 프로세싱 엘리먼트들을 포함할 수도 있다. 셰이더 유닛 (122) 이 복수의 프로세싱 엘리먼트들을 포함하는 경우, 각 프로세싱 엘리먼트는 일부 예들에서, 다른 프로세싱 엘리먼트들과 병렬로 셰이더 프로그램 (124) 을 실행하도록 구성될 수도 있다. 일부 예들에서, 셰이더 유닛 (122) 에서의 프로세싱 엘리먼트들은 병렬의 단일 명령, 최신 GPU들에서 보통 발견되는 다중 데이터 (SIMD) 파이프라인을 형성할 수도 있다.
- [0144] 셰이더 프로그램 (124) 은 셰이더 유닛 (122) 과 연관된 명령 메모리 또는 캐시에 저장되는 프로그램 명령들을 포함할 수도 있다. 셰이더 프로그램 (124) 은 호스트 CPU, 예를 들어 도 1에서 CPU (6) 상에서, 사용자 소프트웨어 어플리케이션, GPU 드라이버, 또는 호스트 상에서 실행하는 또 다른 서비스에 의해 컴파일될 수도 있고, 셰이더 프로그램 (124) 의 컴파일된 버전은 실행을 위해 셰이더 유닛 (122) 상으로 다운로드될 수도 있다.
- [0145] 목적지 알파 테스트 모듈 (126) 은 본 개시물에 기술된 다양한 목적지 알파 테스트 모듈들에 기인하는 기능들 중 어느 하나를 수행할 수도 있다. 예를 들어, 목적지 알파 테스트 모듈 (126) 은 소스 픽셀들 상에서 목적지 알파 테스트를 수행할 수도 있고, 목적지 알파 테스트의 결과들에 기초하여 텍스처 맵핑 모듈 (54) 이전에 개별 소스 픽셀들을 선택적으로 폐기할 수도 있다.
- [0146] 도 12 는 본 개시물의 선택적 텍스처값 추출 기법들에 따른 목적지 알파 테스트 및 텍스처 맵핑을 수행하기 위해 프로그램가능한 셰이더 유닛을 사용하는 또 다른 예의 픽셀 프로세싱 파이프라인 (128) 을 도시하는 블록 다이어그램이다. 픽셀 프로세싱 파이프라인 (128) 은 셰이더 유닛 (122), 텍스처 버퍼 (58) 및 목적지 알파 버퍼 (60) 를 포함한다. 도 12 에 나타난 바와 같이, 픽셀 프로세싱 파이프라인 (128) 의 아키텍처는, 텍스처 맵핑 모듈 (54) 이 고정 함수 파이프라인 스테이지의 부분으로서 대신 프로그램가능한 셰이더 유닛 상에서 구현된 것을 제외하고 도 11에 나타난 픽셀 프로세싱 파이프라인 (120) 과 유사하다. 도 11 및 도 12 에서 유사하게 넘버링된 컴포넌트들은 동일하거나 유사한 기능성 및 구성을 갖는다. 따라서, 간결성을 위해 그리고 중복을 회피하기 위해, 이러한 공유된 컴포넌트들의 동작 및 구성은 더 상세하게 설명되지 않을 것이다.
- [0147] 셰이더 유닛 (122) 은 셰이더 프로그램 (130) 을 포함하고, 차례로 목적지 알파 테스트 모듈 (126) 및 텍스처

맵핑 모듈 (132) 을 포함한다. 목적지 알파 테스트 모듈 (126) 은 본 개시물의 다양한 목적지 알파 테스트 모듈들에 기인하는 기능들 중 어느 하나를 수행할 수도 있다. 유사하게, 텍스처 맵핑 모듈 (132) 은 본 개시물의 다양한 텍스처 맵핑 모듈들에 기인하는 기능들 중 어느 하나를 수행할 수도 있다. 셰이더 유닛 (122) 이 소스 픽셀을 수신할 때, 셰이더 유닛 (122) 은 먼저 소스 픽셀이 폐기되어야 하는지를 결정하도록 목적지 알파 테스트 모듈 (126) 을 실행할 수도 있다. 목적지 알파 테스트 모듈 (126) 이 소스 픽셀이 폐기되어야 하는 것을 결정하면, 셰이더 유닛 (122) 은 픽셀을 폐기하고 텍스처 맵핑 모듈 (132) 이 소스 픽셀에 관하여 실행되지 않는다. 반면, 목적지 알파 테스트 모듈 (126) 이 소스 픽셀이 폐기되지 않아야 한다고 결정하면, 텍스처 맵핑 모듈 (132) 은 소스 픽셀에 대한 텍스처 맵핑을 수행하도록 소스 픽셀에 관하여 실행된다. 결과의 텍스처 맵핑된 소스 픽셀은 하나 이상의 후속 프로세싱 스테이지들로 패스된다.

[0148] 도 13 은 본 개시물의 목적지 알파값에 기초하여 소스 픽셀에 대한 텍스처값들을 선택적으로 추출하기 위한 일 예의 기법을 도시하는 플로우 다이어그램이다. 일부 예들에서, 도 13 에 도시된 기법은 도 1 내지 도 12 에 나타난 시스템들 또는 컴포넌트들 중 어느 하나로 구현될 수도 있다. 목적지 알파 테스트 모듈 (52) 은 빈 버퍼 (64) 로부터 목적지 알파값을 추출한다 (140). 목적지 알파값은 제 1 프리미티브와 연관된 제 1 픽셀을 프로세싱하는 것에 응답하여 목적지 알파 결정 모듈 (56) 에 의해 생성될 수도 있다. 목적지 알파 테스트 모듈 (52) 은 제 2 픽셀에 대한 하나 이상의 텍스처값들이 텍스처 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정한다 (142). 제 2 픽셀은 제 1 프리미티브와 상이한 제 2 프리미티브와 연관될 수도 있다. 제 2 픽셀은 제 1 픽셀의 픽셀 로케이션과 동일한 픽셀 로케이션을 가질 수도 있다.

[0149] 액션을 수행하기로 결정하는 것에 응답하여 (144 - 예), 목적지 알파 테스트 모듈 (52) 은 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션을 수행한다 (146). 예를 들어, 목적지 알파 테스트 모듈 (52) 은 픽셀이 픽셀 프로세싱 파이프라인 (50) 의 텍스처 맵핑 스테이지 (54) 에 의해 프로세싱되기 전에 제 2 픽셀을 폐기한다. 대조적으로, 액션을 수행하지 않도록 결정하는 것에 응답하여 (144 - 아니오), 목적지 알파 테스트 모듈 (52) 은 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션을 수행하지 않는다 (148). 예를 들어, 목적지 알파 테스트 모듈 (52) 은 하나 이상의 프로세싱 스테이지들의 세트에 제 2 픽셀을 패스할 수도 있으며, 스테이지들 중 하나는 텍스처 맵핑 스테이지 (54) 를 포함할 수도 있다.

[0150] 추가 예들에서, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션은 텍스처 맵핑이 제 2 픽셀에 대해 수행되지 않게 하는 액션일 수도 있다. 부가 예들에서, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션은, 텍스처값들이 제 2 픽셀에 대해 추출되지 않게 하는 제어 신호를 어서트하는 액션일 수도 있다. 예를 들어, 제어 신호는 픽셀 프로세싱 파이프라인 내의 또 다른 프로세싱 스테이지가 픽셀을 폐기하게 할 수도 있다. 또 다른 예에서, 제어 신호는 픽셀 프로세싱 파이프라인 내의 특정 픽셀에 대한 텍스처 맵핑 스테이지를 인에이블 또는 디스에이블할 수도 있다. 더 많은 예들에서, 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션은, 텍스처 맵핑이 픽셀에 대해 디스에이블되게 하는 값으로 제 2 픽셀에 대한 텍스처 맵핑 디스에이블 속성을 설정하는 액션일 수도 있다.

[0151] 부가 예들에서, 목적지 알파 테스트 모듈 (52) 은 제 2 통신 인터페이스를 통해 빈 버퍼 (64) 로부터 목적지 알파값을 추출할 수도 있고, 하나 이상의 텍스처값들은 제 2 통신 인터페이스를 통해 텍스처 버퍼 (58) 로부터 추출될 수도 있다. 그러한 예들에서, 제 1 통신 인터페이스는, 일부 예들에서, 제 2 통신 인터페이스보다 큰 대역폭을 가질 수도 있다.

[0152] 빈 버퍼 (64) 는, 일부 예들에서, 프레임 버퍼 (64) 에 저장된 목적지 픽셀들의 서브세트에 대한 픽셀 데이터를 저장할 수도 있다. 예를 들어, 빈 버퍼 (64) 에 저장된 목적지 픽셀들의 서브세트는 빈 렌더링 기법의 부분으로서 특정 렌더링 패스와 연관된 목적지 픽셀 범위에 대응할 수도 있다. 빈 버퍼 (64) 는 빈 버퍼에서 특정 저장 슬롯을 특정하는 메모리 어드레스를 포함하는 판독 요청을 수신하고, 그 저장 슬롯에 저장된 데이터를 리턴하도록 구성될 수도 있다. 저장 슬롯은 현재의 렌더링 패스와 연관된 특정 목적지 픽셀에 대응할 수도 있다. 일부 예들에서, 특정 렌더링 패스 동안, 빈 버퍼 (64) 에 대한 각각의 메모리 어드레스는 특정 렌더링 패스에 대한 목적지 픽셀 범위 내의 고유 목적지 픽셀에 대응한다.

[0153] 일부 예들에서, 렌더링된 그래픽스 이미지는 복수의 목적지 픽셀들을 포함할 수도 있고, 그래픽스 프로세서는 복수의 렌더링 패스들에서 그래픽스 이미지를 렌더링하도록 구성될 수도 있으며, 렌더링 패스들의 각각은 목적지 픽셀들의 각 서브세트와 연관될 수도 있다. 그러한 예들에서, 각각의 렌더링 패스에 대해, 빈 버퍼는 각

각의 렌더링 패스와 연관된 목적지 픽셀들의 서브세트에 대한 픽셀 데이터를 저장할 수도 있다. 부가 예들에서, 빈 버퍼의 용량은 그래픽스 이미지와 연관된 복수의 목적지 픽셀들 모두에 대한 픽셀 데이터를 저장하는데 필요한 최소 용량 미만일 수도 있다. 추가 예들에서, 각각의 렌더링 패스와 연관된 픽셀들의 개별 서브세트들은 픽셀들의 미리 결정된 서브세트들일 수도 있다.

[0154] 도 14 는 도 13 에 나타낸 기법들 구현하는데 사용될 수도 있는 목적지 알파 테스트를 수행하기 위한 일 예의 기법을 도시하는 플로우 다이어그램이다. 일부 예들에서, 도 14 에 나타낸 기법은 도 13에서 프로세스 박스 (142) 를 구현하는데 사용될 수도 있다. 목적지 알파 테스트 모듈 (52) 은 수신된 소스 픽셀과 연관된 픽셀 로케이션들을 결정한다 (150). 목적지 알파 테스트 모듈 (52) 은 목적지 알파 버퍼 (60) 로부터 픽셀 로케이션과 연관된 목적지 알파값을 추출한다 (152). 목적지 알파 테스트 모듈 (52) 은 목적지 알파값을 임계값과 비교한다 (154). 목적지 알파 테스트 모듈 (52) 은 제 2 픽셀에 대한 하나 이상의 텍스처값들이 비교 결과에 기초하여 버퍼로부터 추출되지 않게 하는 액션을 수행할지 여부를 결정한다 (156).

[0155] 도 15 는 도 14 에 나타낸 기법을 구현하는데 사용될 수도 있는 목적지 알파값을 임계와 비교하기 위한 일 예의 기법을 도시하는 플로우 다이어그램이다. 일부 예들에서, 도 14 에 나타낸 기법은 도 14 에서 프로세스 박스 (156) 을 구현하는데 사용될 수도 있다. 목적지 알파 테스트 모듈 (52) 은 목적지 알파값이 임계값 이하인지 여부를 결정한다 (158). 목적지 알파값이 임계값 이하이면 (160 - 예), 목적지 알파 테스트 모듈 (52) 은 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션을 수행하도록 결정한다 (162). 예를 들어, 목적지 알파 테스트 모듈 (52) 은 픽셀 프로세싱 파이프라인의 텍스처 맵핑 스테이지 전에 소스 픽셀을 폐기할 수도 있다. 대조적으로, 목적지 알파값이 임계값 이하가 아니면 (160 - 아니오), 제 2 픽셀에 대한 하나 이상의 텍스처값들이 버퍼로부터 추출되지 않게 하는 액션을 수행하지 않도록 결정한다 (164). 예를 들어, 목적지 알파 테스트 모듈 (52) 은 하나 이상의 후속 프로세싱 스테이지들로 픽셀을 패스할 수도 있으며, 스테이지들 중 하나는 텍스처 맵핑 스테이지를 포함할 수도 있다.

[0156] 도 16 은 본 개시물에 따른 목적지 알파값 기반 선택적 텍스처값 추출 렌더링 기법을 자동으로 인에이블 시키기 위한 일 예의 기법을 도시하는 플로우 다이어그램이다. 도 16 의 기법은, 예를 들어 도 2 에 나타낸 GPU 드라이버 (28) 와 같은, GPU 드라이버에서 구현될 수도 있다. GPU 드라이버 (28) 는 전방-후방 렌더링 모드가 인에이블되는지를 검출할 수도 있다 (166). 예를 들어, GPU 드라이버 (28) 는, 소프트웨어 어플리케이션 (24) 이 전방-후방 알파 블렌딩 방식식들을 사용하여 알파 블렌딩을 수행하기 위해 GPU (12) 에서 알파 블렌딩 모듈을 구성하는지를 검출할 수도 있다. 검출에 응답하여, GPU 드라이버 (28) 는 GPU (12) 에 대한 목적지 알파 기반 선택적 텍스처값 추출 렌더링 모드를 자동으로 인에이블시킬 수도 있다 (168). 목적지 알파 기반 선택적 텍스처값 추출 렌더링 모드는, 예를 들어 도 13 에 나타낸 기법들에 따라 프리미티브들을 렌더링할 수도 있다.

[0157] 위에서 이미 논의된 바와 같이, 본 개시물의 목적지 알파 테스트 기법들은, 중첩할 수도 있는 다중 표면들이 단 일 표면으로 컴포지션되는 "비닝 (binning)" 환경에서 특히 유용할 수도 있다. 비닝 환경에서, 컴포지션에 대한 제한 인자들은 오프 칩 "시스템 메모리" 에 저장된 텍스처들의 판독 및 기록에 기인하는 경향이 있다. 한편, 빈 버퍼의 판독들 및 기록들은 본질적으로 대역폭에 관하여 "프리" 이다. 표면들의 세트를 컴포지트 하기 위해, 표면들은 "전방-후방" 순서로 프로세싱될 수도 있고, 목적지 알파값은 유지될 수도 있다. 픽셀 이 불투명하게 되면 (예를 들어, 투명도들의 합계가 모든 추가 광 송신을 블록킹하기 때문이거나 불투명한 픽셀 이 렌더링되기 때문임), 일부 예들에서, 목적지 알파값은 "0" 과 동등하게 될 것이며, 이는 스크린 로케이션에 대해 프로세싱되는 임의의 후속 소스 픽셀들을 폐기하기 위해 목적지 알파 테스트 모듈을 트리거할 수도 있다.

[0158] z 버퍼링 기법들이 동일한 픽셀 로케이션에서 다른 소스 픽셀들 뒤에 있는 소스 픽셀들을 폐기함으로써 컴포지션 대역폭을 감소시키기 위해 사용될 수도 있지만, 그러한 기법들은 픽셀들이 픽셀 마다의 소스 알파값들을 가질 때 표면들을 적절하게 렌더링하지 않으며, 이는 본질적으로 중첩 픽셀들이 불투명하게 한다. 하지만, 본 개시물의 목적지 알파값 기반 선택적 텍스처값 추출 기법들은 모든 표면들이 픽셀당 소스 알파값들을 가질 때에도 컴포지션 대역폭을 감소시킬 수도 있으며, 이로써 보다 강건한 대역폭 감소 메커니즘을 제공한다.

[0159] 일부 예들에서, 본 개시물의 목적지 알파 테스트 기법들은, 예를 들어, 윈도우 표면들을 컴포지팅할 때, 그러한 기법들이 특히 유용한 경우 선택적으로 인에이블될 수도 있다. 추가 예들에서, 목적지 알파 테스트 기법들은 그러한 스텝이 발생하는 것을 컴포지터가 반드시 인지하지 않으면서 그래픽스 드라이버에 의해 및/또는 그래픽스 프로세서에 의해 인에이블될 수도 있다. 예를 들어, 드라이버 및/또는 그래픽스 프로세서는 사용자 어플리케이션이 전방-후방 방식으로 표면들을 컴포지트하려고 시도하는 것을 검출할 수도 있고, 그러한 검출에 응

답하여 본 개시물의 목적지 알파 테스트 기법들을 자동으로 인에이블시킬 수도 있다. 예를 들어, 드라이버 및/또는 그래픽스 프로세서는, 알파 블렌딩 모듈이 "전방-후방" 블렌딩 함수로 설정되고, 수 개의 표면 드로잉 커맨드들이 소프트웨어에 의해 호출되는 것을 검출할 수도 있고, 본 개시물의 목적지 알파 테스트 기법들을 자동으로 인에이블시킬 수도 있다.

[0160] 본 개시물의 목적지 알파 테스트 기법들은 텍스처값 추출 대역폭을 감소시키기 위한 다른 비목적지 알파값 기반 기법들에 걸쳐 하나 이상의 이점들을 제공할 수도 있다. 예를 들어, 하나의 기법은 "위" 그리고 최상위 불투명 표면을 포함하는 표면만을 테셀레이트 (tessellate) 하고 렌더링하는 것이다. 하지만, 이러한 기법은 모든 표면들이 투명하거나 픽셀 마다 알파를 가질 때 실패한다. 대역폭을 감소시키기 위한 또 다른 기법은 'Z' 를 각 표면으로 할당하고 Z 값을 사용하여 불투명 표면을 아래의 표면들을 렌더링하는 것을 회피하는 것을 포함한다. 대역폭을 감소시키기 위한 또 다른 기법은 표면 마다, 픽셀 알파 테스트 마다 하나의 패스에서 모든 표면들의 셰이더 기반 렌더링을 포함한다. 그러나 이러한 작업들은 어플리케이션에서 현저한 변화를 요구한다. 또한, 그것은 표면들의 수가 커지게 되거나 표면들의 형상이 직사각형과 상이하게 되면, 더 많은 셰이더 마력을 취하고 활력을 소진한다.

[0161] 본 개시물에 기재된 기법들은 적어도 부분적으로 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 예를 들어, 개시된 기법들의 다양한 양태들은, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP)들, 주문형 집적회로(ASIC)들, 필드 프로그램가능 로직 어레이(FPGA)들, 또는 다른 등가의 집적 또는 이산 로직 회로를 포함하는 하나 이상의 프로세서들 뿐만 아니라 그러한 컴포넌트들의 임의의 조합들 내에서 구현될 수도 있다. 용어 "프로세서" 또는 "프로세싱 회로" 는 일반적으로 전술한 로직 회로 단독 또는 다른 로직 회로와의 조합 중 어느 하나를 지칭할 수도 있다.

[0162] 그러한 하드웨어, 소프트웨어 및 펌웨어는 동일한 디바이스 내에 또는 별도의 디바이스들 내에 구현되어 본 개시물에 개시된 다양한 동작들 및 기능들을 지원할 수도 있다. 또한, 개시된 유닛들, 모듈들 또는 컴포넌트들 중 어느 하나가 함께 또는 이산으로서 별도로 구현될 수도 있지만 로직 디바이스들을 상호동작 가능하게 할 수도 있다. 모듈들로서 또는 유닛들로서 상이한 피쳐들의 도시는 상이한 기능적 양태들을 강조하려는 것이며, 그러한 모듈들 또는 유닛들이 별도의 하드웨어 또는 소프트웨어 컴포넌트들에 의해 실현되어야 하는 것을 반드시 의미하는 것은 아니다. 대신, 하나 이상의 모듈들 또는 유닛들과 연관된 기능성은 별도의 하드웨어, 펌웨어 및/또는 소프트웨어 컴포넌트들에 의해 수행될 수도 있고 또는 공통 또는 별개의 하드웨어 또는 소프트웨어 컴포넌트들 내에 집적될 수도 있다.

[0163] 본 개시물에 기재된 기법들은 또한 명령들을 저장하는 컴퓨터 판독가능 저장 매체와 같은 컴퓨터 판독가능 매체에 저장되고, 수록되고 또는 인코딩될 수도 있다. 컴퓨터 판독가능 매체에 임베딩되거나 인코딩된 명령들은 하나 이상의 프로세서들로 하여금, 예를 들어 명령들이 하나 이상의 프로세서들에 의해 실행될 때, 본 명세서에 기재된 기법들을 수행하게 할 수도 있다. 컴퓨터 판독가능 저장 매체는 랜덤 액세스 메모리 (RAM), 리드 온니 메모리 (ROM), 프로그램가능 리드 온니 메모리 (PROM), 소거가능 프로그램가능 리드 온니 메모리 (EPROM), 전기적 소거가능 프로그램가능 리드 온니 메모리 (EEPROM), 플래시 메모리, 하드 디스크, CD-ROM, 플로피 디스크, 카세트, 자기 매체들, 광학 매체들 또는 유형인 다른 컴퓨터 판독가능 저장 매체들을 포함할 수도 있다.

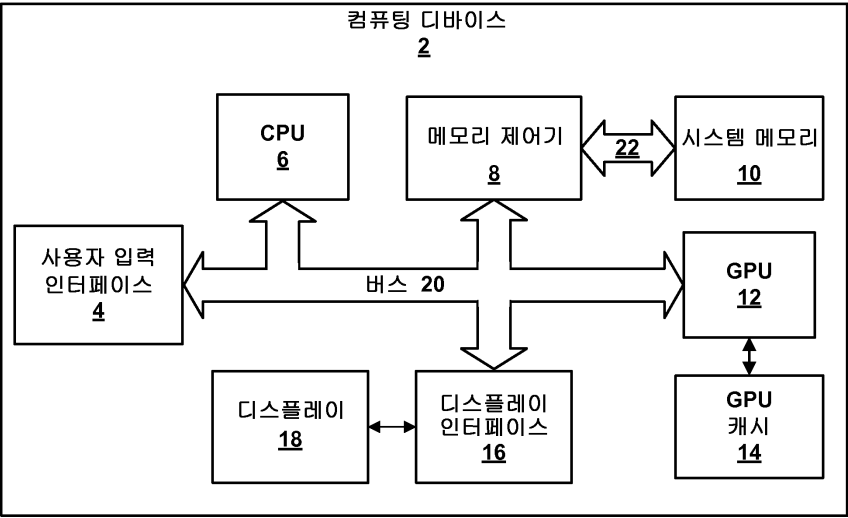
[0164] 컴퓨터 판독가능 매체는 위에 열거한 것들과 같은, 유형의 저장 매체에 대응하는 컴퓨터 판독가능 저장 매체들을 포함할 수도 있다. 컴퓨터 판독가능 매체는 또한, 예를 들어 통신 프로토콜에 따라 일 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 다른 매체를 포함하는 통신 매체들을 포함할 수도 있다.

이러한 방식으로, 구문 "컴퓨터 판독가능 매체" 는 일반적으로 (1) 비일시적인, 유형의 컴퓨터 판독가능 저장 매체, 또는 (2) 신호 또는 반송파와 같은, 통신 매체에 대응할 수도 있다.

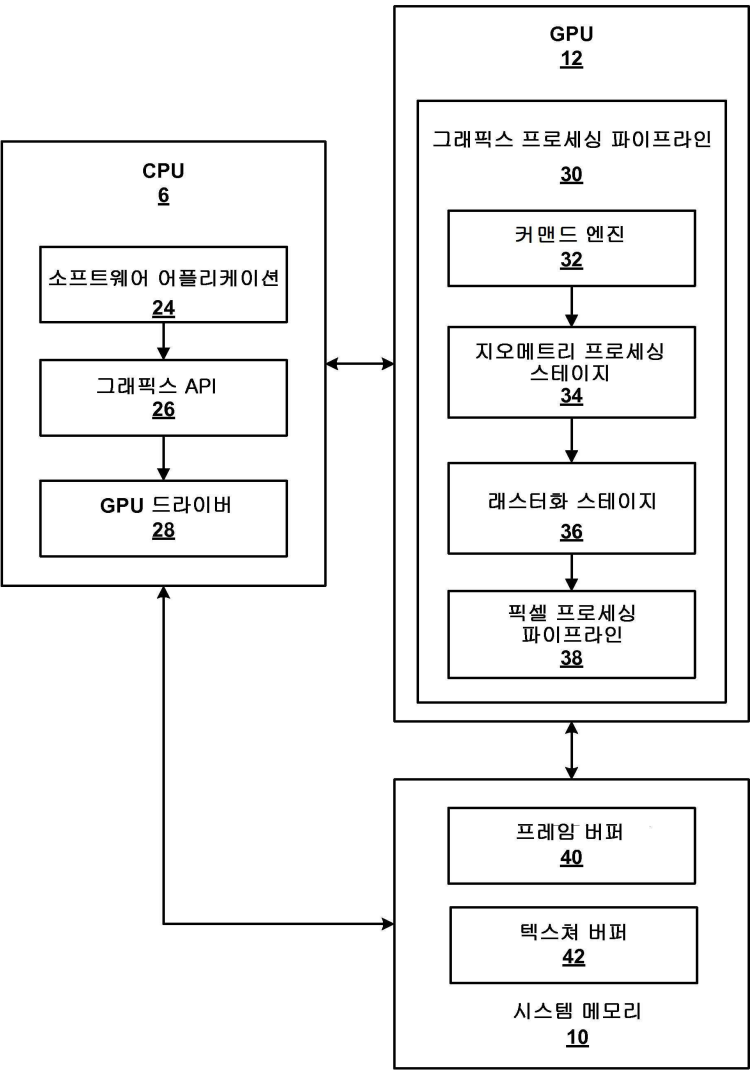
[0165] 다양한 양태들 및 예들이 기술되었다. 하지만, 다음의 청구항들의 범위를 벗어나지 않으면서 본 개시물의 구조 또는 기법들에 대해 변형들이 이루어질 수 있다.

도면

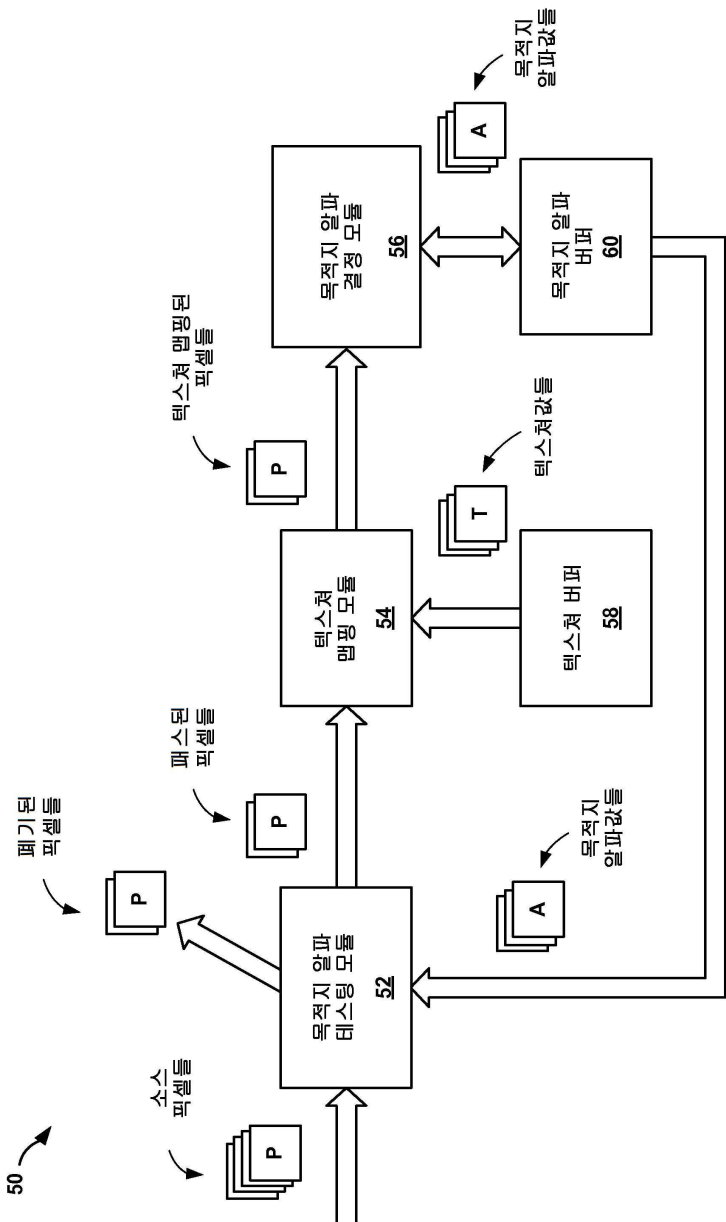
도면1



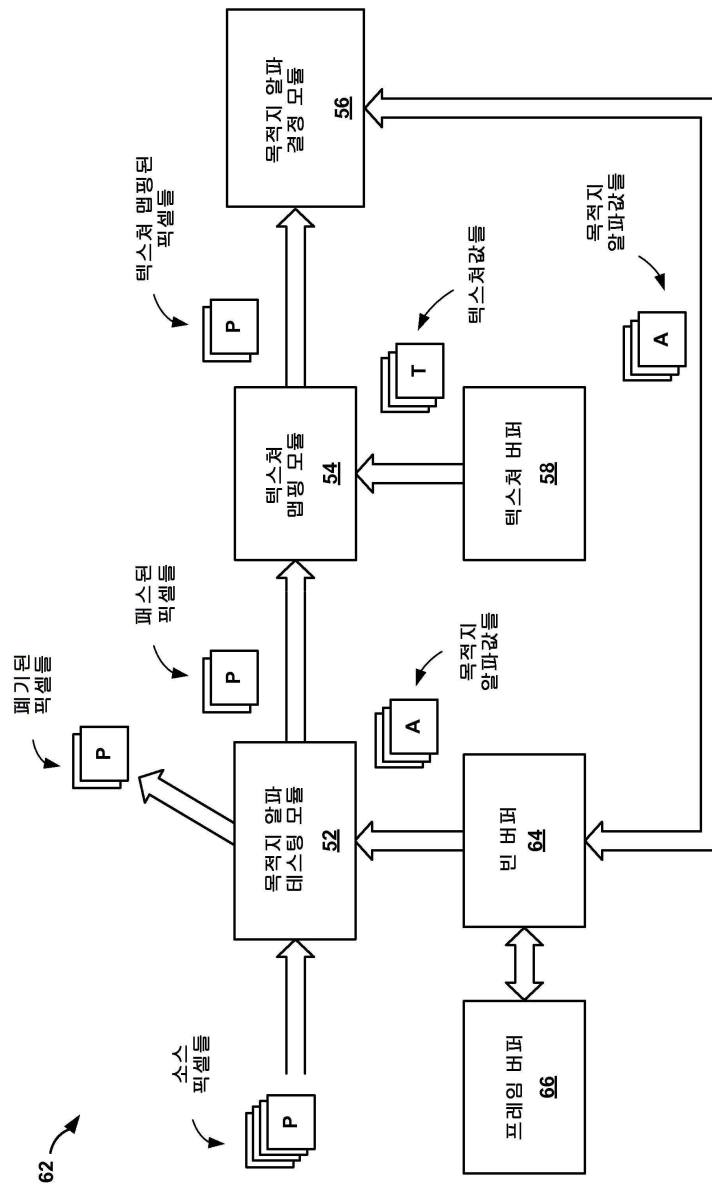
도면2



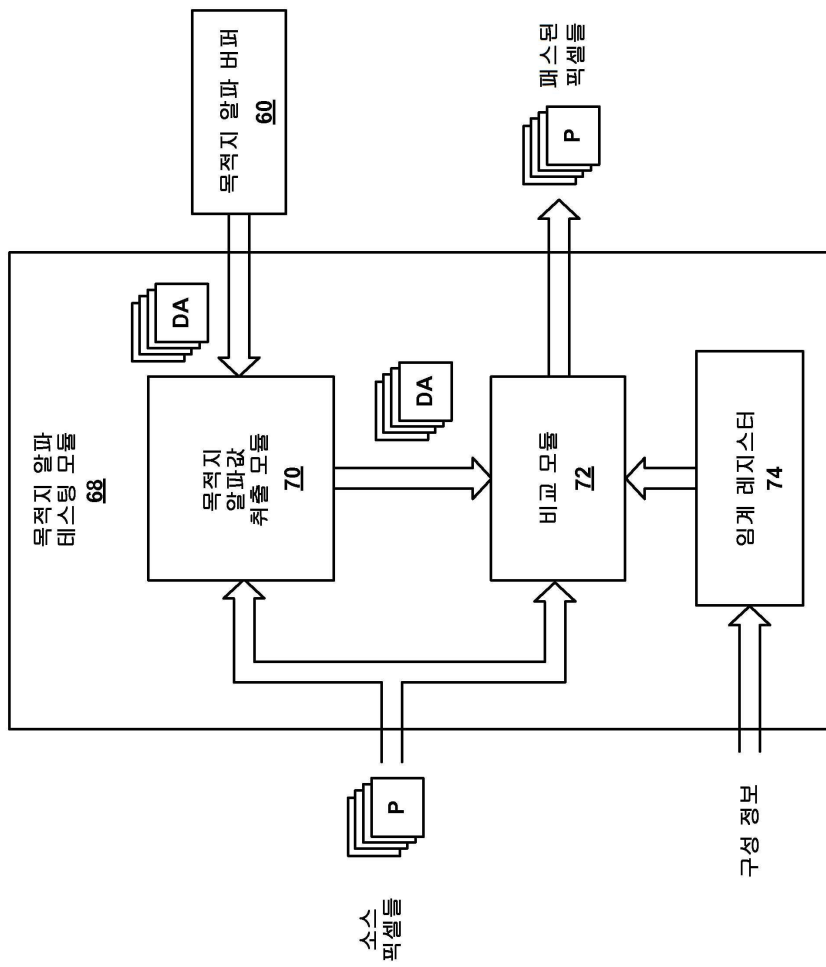
도면3



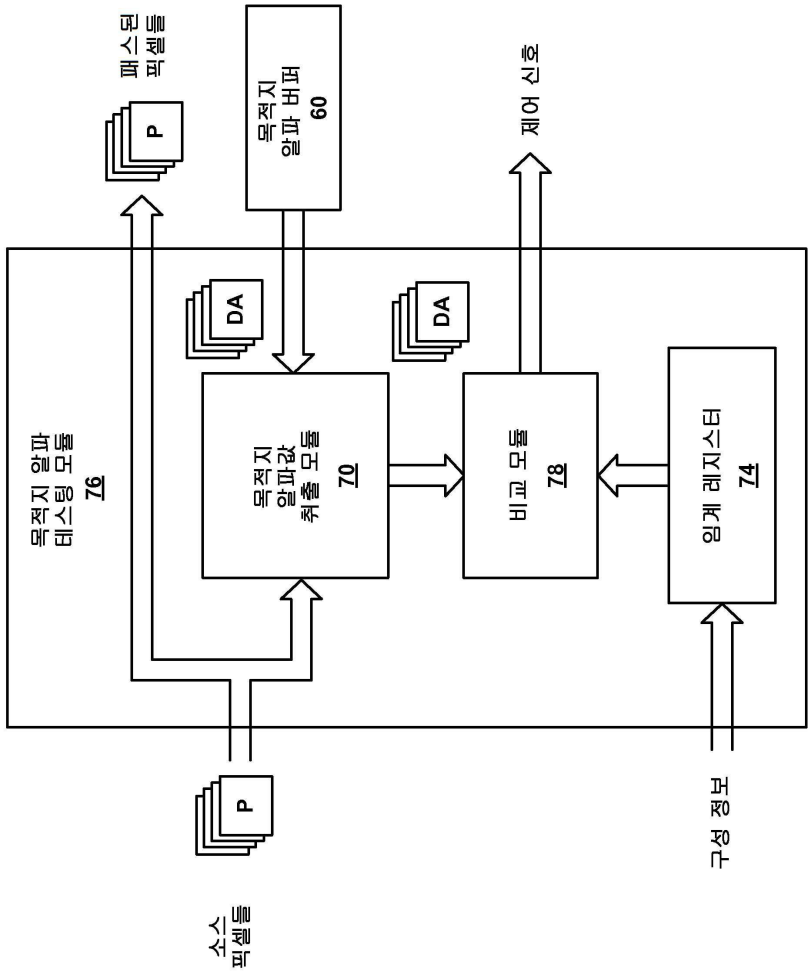
도면4



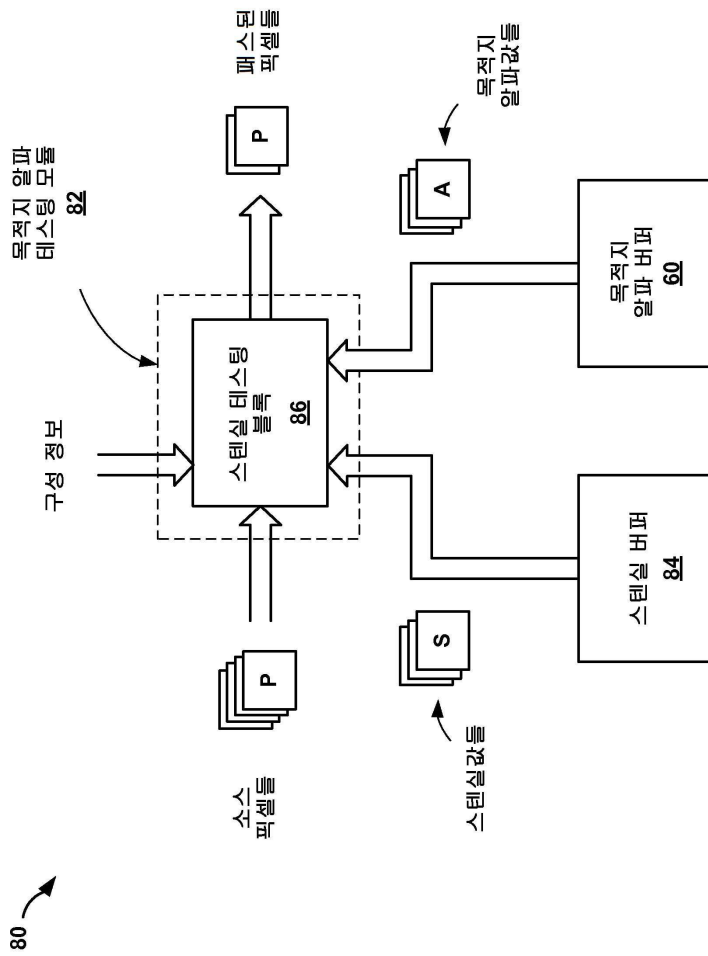
도면5



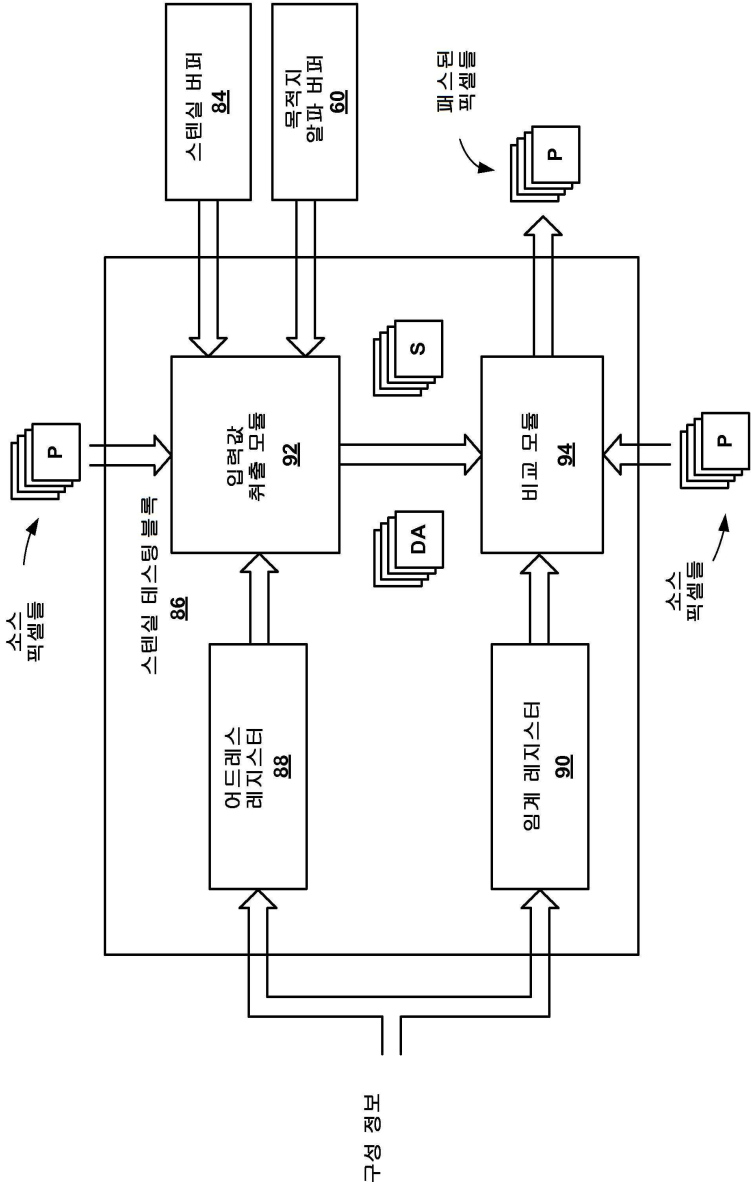
도면6



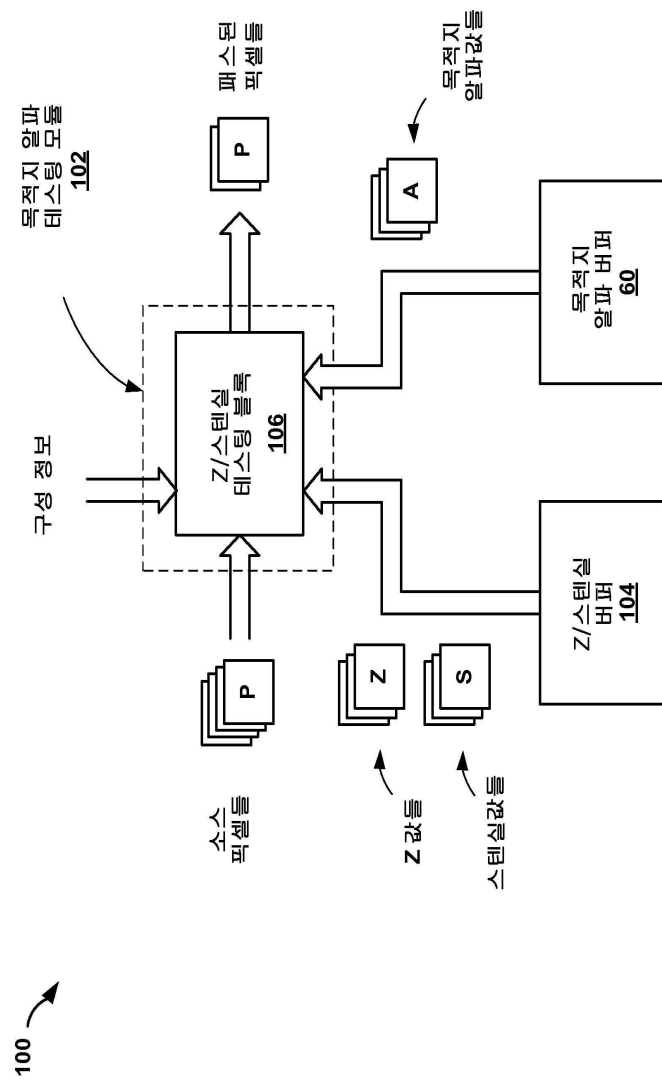
도면7



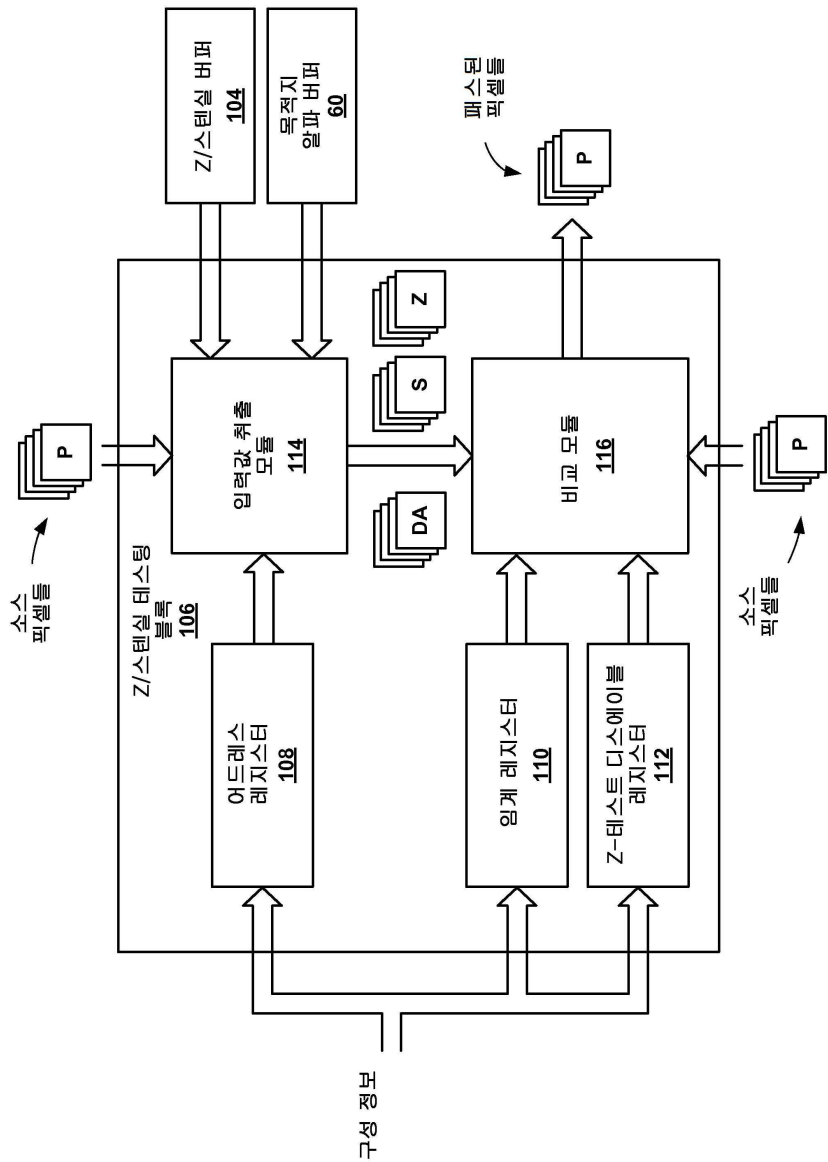
도면8



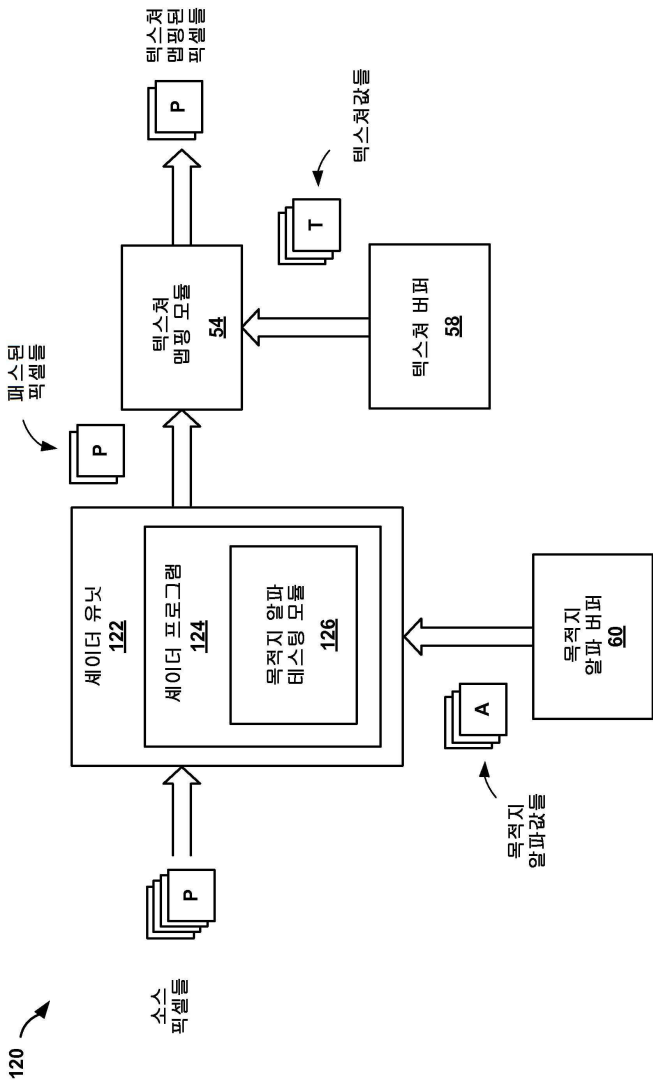
도면9



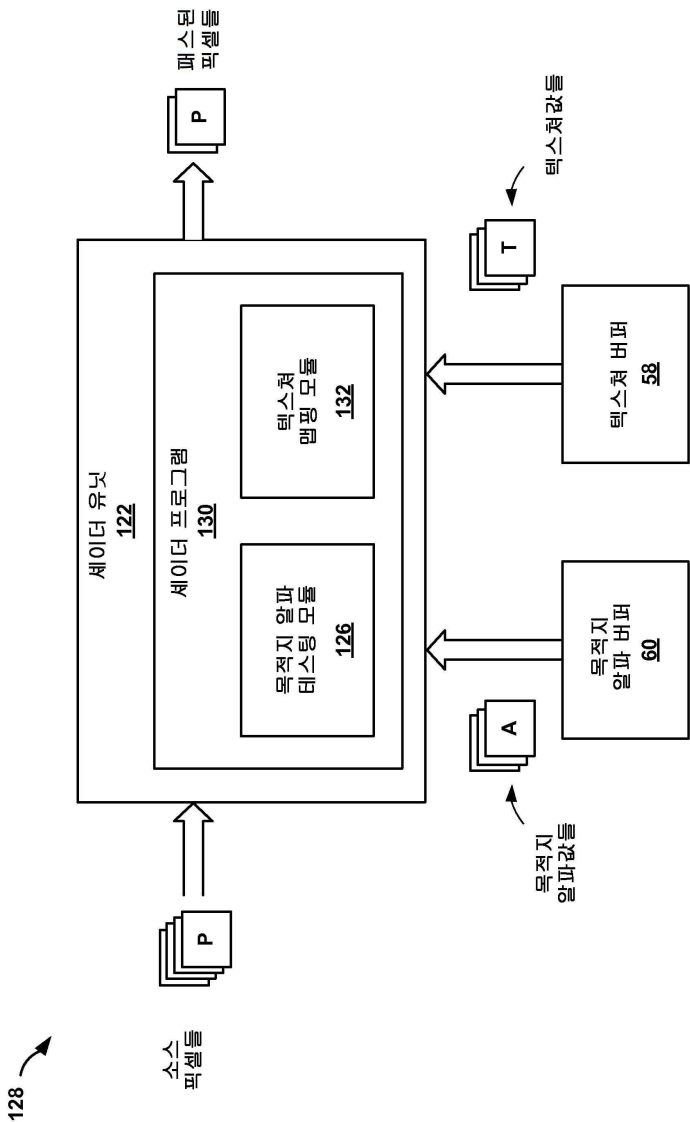
도면10



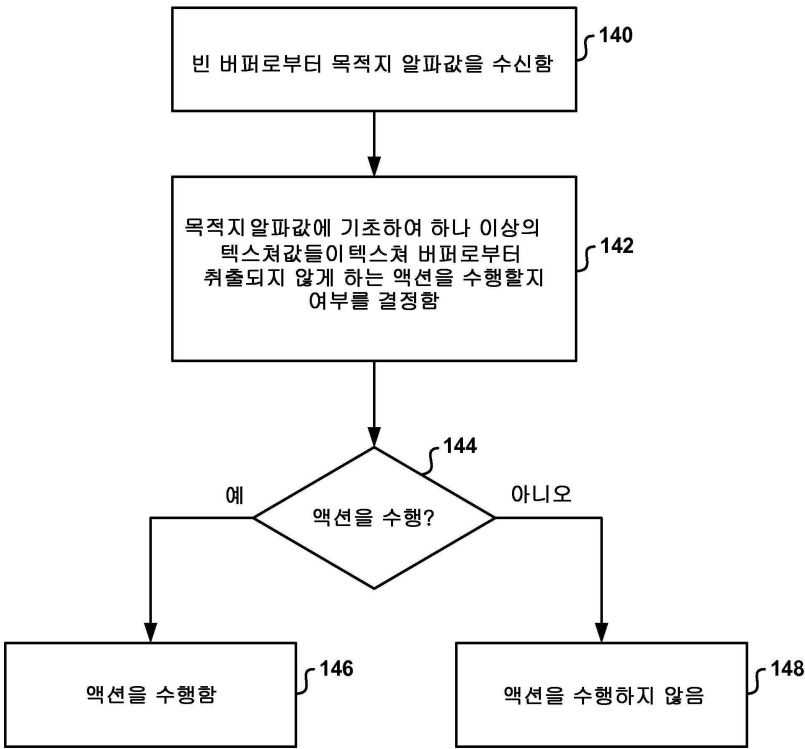
도면11



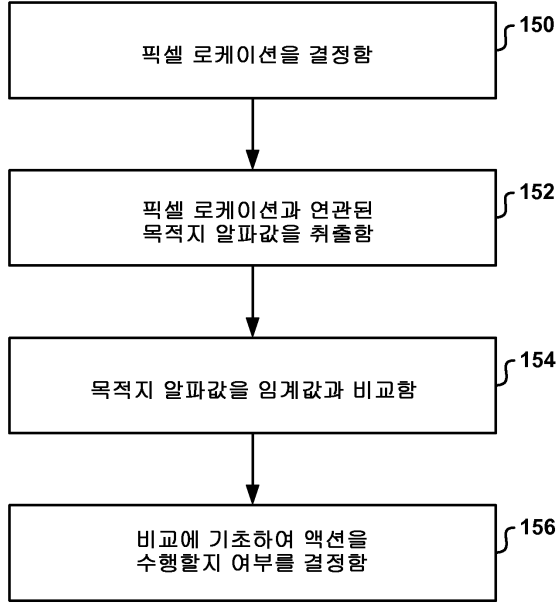
도면12



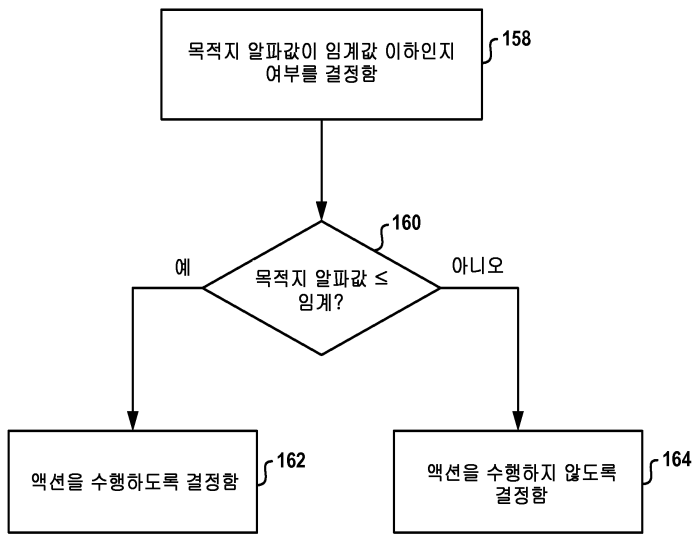
도면13



도면14



도면15



도면16

