



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0097515
(43) 공개일자 2012년09월04일

- | | |
|--|--|
| <p>(51) 국제특허분류(Int. Cl.)
G06T 9/00 (2006.01)</p> <p>(21) 출원번호 10-2012-7015254</p> <p>(22) 출원일자(국제) 2010년02월24일
심사청구일자 없음</p> <p>(85) 번역문제출일자 2012년06월13일</p> <p>(86) 국제출원번호 PCT/SE2010/050217</p> <p>(87) 국제공개번호 WO 2011/065886
국제공개일자 2011년06월03일</p> <p>(30) 우선권주장
61/263,922 2009년11월24일 미국(US)
61/266,316 2009년12월03일 미국(US)</p> | <p>(71) 출원인
텔레폰악티에블라겟엘엠에릭슨(펍)
스웨덴왕국 스톡홀름 에스-164 83</p> <p>(72) 발명자
스트롬 자콥
스웨덴 스톡홀름 에스-117 32 2 티알 헬레네보그
스가탄 6씨</p> <p>(74) 대리인
박병석, 서장찬, 최재철</p> |
|--|--|

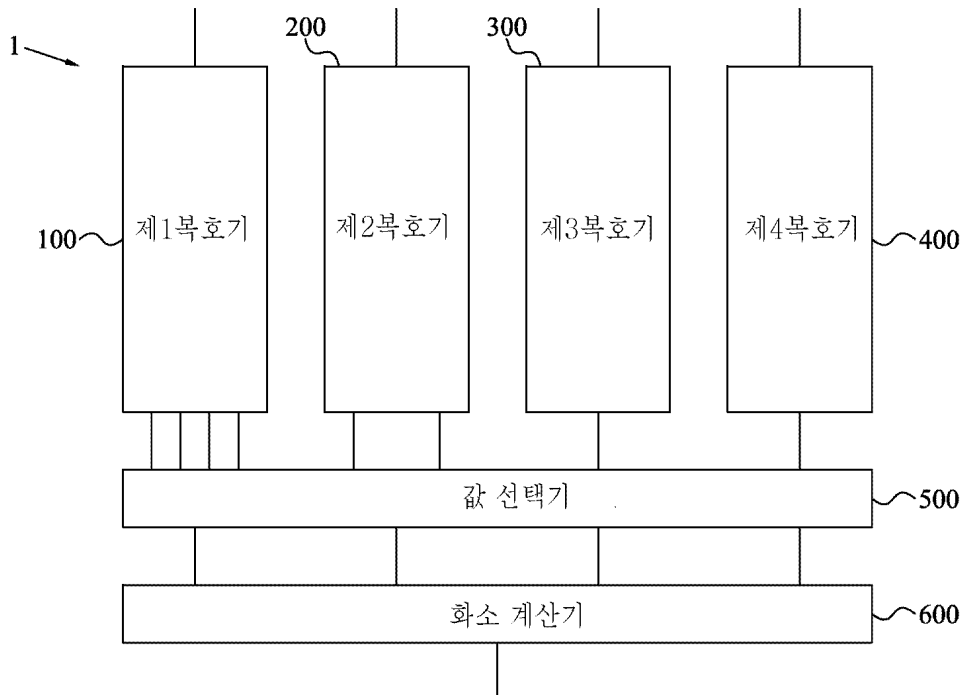
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 암호화된 텍셀 블록에서 동작할 수 있는 복호 시스템 및 방법

(57) 요약

복호 시스템(1)은, 각각이 다른 N-1개의 복호기들의 회로들과는 상이한 고유 회로를 가지는 N개의 상이한 복호기(100, 200, 300, 400)들을 포함한다. 복호기(100, 200, 300, 400) 각각은 입력 암호화된 텍셀블록(40)을 기반으로 적어도 하나의 텍셀값을 생성한다. 값 선택기(500)는 적어도 N개 텍셀(30 내지 36)들 중 적어도 하나를 포함하는 텍셀블록(10)의 경계에 관해 적어도 N개의 텍셀(30 내지 36)들의 위치를 기반으로 복호기(100, 200, 300, 400)들 중 적어도 하나로부터 적어도 N개의 텍셀값들을 선택적으로 출력하도록 구성된다. 화소 계산기(600)는 값 선택기(500)로부터 적어도 N개의 선택된 텍셀값들을 기반으로 복호된 화소의 화소값을 계산한다.

대표도



특허청구의 범위

청구항 1

적어도 하나의 암호화된 텍셀블록(40)을 기반으로 복호된 화소를 생성하기 위한 복호 시스템(1)에 있어서, 상기 복호 시스템(1)은:

각각이 다른 N-1개 복호기들의 회로들과는 상이한 회로를 가지고 또한 각각은 적어도 하나의 각 텍셀값을 생성하기 위해 암호화된 텍셀블록(40)을 복호하도록 구성되는, N개의 상이한 복호기(100, 200, 300, 400)들과;

상기 N개의 복호기(100, 200, 300, 400)들에 연결되고 또한 적어도 N개의 텍셀(30 내지 36)들 중 적어도 하나를 포함하는 텍셀블록(10)의 경계에 관해 상기 적어도 N개의 텍셀값들과 관련되는 상기 적어도 N개이 텍셀(30 내지 36)들의 위치들을 기반으로 상기 N개의 복호기(100, 200, 300, 400)들 중 적어도 하나로부터 적어도 N개의 텍셀값들을 선택적으로 출력하도록 구성되는 값 선택기(500)와; 그리고

상기 값 선택기(500)에 의해 선택적으로 출력되는 상기 적어도 N개의 텍셀값들을 기반으로 상기 복호된 화소의 복호된 화소값을 계산하도록 구성되는 화소 계산기(600)를 포함하는 것을 특징으로 하는 복호 시스템.

청구항 2

제1항에 있어서, 상기 값 선택기(500)는, 만일 상기 N개의 텍셀값들과 관련된 N개의 텍셀(30 내지 36)들이 텍셀블록(10)의 상기 경계 내측에 위치한다면 상기 N개의 복호기(100, 200, 300, 400)들 중 제1복호기(100)로부터 N개의 텍셀값들을 선택적으로 출력하도록 구성되는 것을 특징으로 하는 복호 시스템.

청구항 3

제1항 또는 제2항에 있어서, 상기 값 선택기(500)는, 만일 상기 N개의 텍셀값들과 관련된 N개의 텍셀(30 내지 36)들 중 반이 상기 텍셀블록(10)의 상기 경계 내측에 위치하고 또한 상기 N개의 텍셀(30 내지 36)들 중 반이 이웃하는 텍셀블록(12, 14)의 경계 내측에 위치한다면, 상기 N개의 복호기(100, 200, 300, 400)들 중 제1복호기(100)로부터 N개의 텍셀값들 중 반을 선택적으로 출력하고 또한 상기 N개의 텍셀값들 중 나머지 반은 상기 N개의 복호기(100, 200, 300, 400)들 중 제2복호기(200)로부터 선택적으로 출력하도록 구성되는 것을 특징으로 하는 복호 시스템.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 N은 4와 동일하고 또한 만일, 상기 네 개의 텍셀(30 내지 36)들 ND 하나가 상기 텍셀블록(10)의 상기 경계 내측에 위치하고 그리고 잔여 세 개의 텍셀들 각각이 이웃하는 텍셀블록(12, 14, 16)들의 경계 내측에 위치한다면, 상기 값 선택기(500)는 상기 네 개의 복호기(100, 200, 300, 400)들 중 제1복호기(100)로부터 상기 네 개의 텍셀값들 중 제1텍셀값을 선택적으로 출력하고, 상기 네 개의 텍셀값들 중 제2텍셀값을 상기 네 개의 복호기(100, 200, 300, 400)들 중 제2복호기(200)로부터 출력하고, 상기 네 개의 텍셀값들 중 제3텍셀값을 상기 네 개의 복호기(100, 200, 300, 400)들 중 제3복호기(300)로부터 출력하고 그리고 상기 네 개의 텍셀값들 중 제4텍셀값을 상기 네 개의 복호기(100, 200, 300, 400)들 중 제4복호기(400)로부터 출력하는 것을 특징으로 하는 복호 시스템.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서, 상기 N은 4와 동일하고 그리고

상기 네 개의 복호기(100, 200, 300, 400)들 중 제1복호기(100)는 제1회로를 가지고 또한 네 개의 텍셀값을 생성하기 위해 암호화된 텍셀블록(40)을 복호하도록 구성되고;

상기 네 개의 복호기(100, 200, 300, 400)들 중 제2복호기(200)는 제2회로를 가지고 또한 두 개의 텍셀값을 생성하기 위해 암호화된 텍셀블록(40)을 복호하도록 구성되고;

상기 네 개의 복호기(100, 200, 300, 400)들 주 제3복호기는 제3회로를 가지고 또한 하나의 텍셀값을 생성하기 위해 암호화된 텍셀블록(40)을 복호하도록 구성되고; 그리고

상기 네 개의 복호기(100, 200, 300, 400)들 중 제4복호기(400)는 제4회로를 가지고 또한 하나의 텍셀값을 생

성하기 위해 암호화된 텍셀블록(40)을 복호하도록 구성되는 것을 특징으로 하는 복호 시스템.

청구항 6

제1항 내지 제5항 중 어느 한 항에 있어서, 상기 암호화된 텍셀블록(40)은 제1색상 코드워드(41, 44, 47)와, 제2색상 코드워드(42, 45, 48)와 그리고 제3색상 코드워드(43, 46, 49)를 포함하는 것을 특징으로 하는 복호 시스템.

청구항 7

제5항 및 제6항에 있어서, 상기 제4복호기(400)는 상기 제1색상 코드워드(41, 44, 47)를 출력하기 위한 적어도 하나의 배선을 포함하는 것을 특징으로 하는 복호 시스템.

청구항 8

제6항 또는 제7항에 있어서, 상기 제3복호기(300)는:

상기 제2색상 코드워드(42, 45, 48)에 대응하는 비트열을 좌측으로 1비트 위치 이동시켜 좌측-이동된 비트열을 형성하도록 구성되는 좌측 이동 연산자(310)와;

1에 설정된 캐리 입력을 가지고 또한 상기 좌측 이동 연산자(310)에 연결되며 또한 상기 제2색상 코드워드(42, 45, 48)와 상기 좌측-이동된 비트열을 가산하도록 구성되는 제1가산기(320)와;

1에 설정된 캐리 입력을 가지고 또한 상기 제1가산기(320)에 연결되며 또한 상기 제1가산기(320)의 출력과 상기 제1색상 코드워드(41, 44, 47)를 가산하도록 구성되는 제2가산기(322)와; 그리고

상기 제2가산기(322)에 연결되고 또한 상기 제2가산기(322)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 우측-이동된 열을 형성하도록 구성되는 우측 이동 연산자(330)를 포함하는 것을 특징으로 하는 복호 시스템.

청구항 9

제6항 내지 제8항 중 어느 한 항에 있어서, 상기 제2복호기(200)는:

네 개의 텍셀(30 내지 36)들 중 절반이 상기 텍셀블록(10)의 상기 경계 내측에서 행 또는 열에 위치하는지를 기반으로 하여 상기 제2색상 코드워드(42, 45, 48)와 상기 제3색상 코드워드(43, 46, 49) 중 하나를 출력하도록 구성되는 제1멀티플렉서(250)와;

상기 제1색상 코드워드(41, 44, 47)를 부정하도록 구성되는 부정 연산기(240)와;

상기 제1멀티플렉서(250)와 상기 부정 연산자(240)에 연결되고 또한 상기 제1멀티플렉서(250)의 출력과 상기 부정된 제1색상 코드워드를 가산하도록 구성되는 제1가산기(220)와;

상기 제1가산기(220)에 연결되고 또한 상기 제1가산기(220)의 출력에 대응하는 비트열을 좌측으로 1비트 위치 이동시켜 좌측-이동된 비트열을 형성하도록 구성되는 좌측 이동 연산자(210)와;

상기 네 개의 텍셀(30 내지 36) 중 절반이 상기 텍셀블록(10)의 상기 경계 내측에서 행 또는 열로 위치하는지를 기반으로 텍셀(32, 24)의 행 좌표 또는 열 좌표 중 하나를 출력하도록 구성되는 제2멀티플렉서(252)와;

상기 좌측 이동 연산자(210)와 상기 제1가산기(220)에 연결되고 또한 상기 제2멀티플렉서(252)의 출력 중 최상위 비트를 기반으로 상기 제1가산기(220)의 상기 출력과 상기 좌측-이동된 비트열 중 하나를 출력하도록 구성되는 제3멀티플렉서(254)와;

상기 제2멀티플렉서(252)에 연결되고 또한 상기 제2멀티플렉서(252)의 상기 출력의 최상위 비트와 상기 제2멀티플렉서의 상기 출력의 최하위 비트를 수신하도록 구성되는 OR 연산자(260)와;

상기 제3멀티플렉서(254)와 상기 OR 연산자(260)에 연결되고 또한 상기 제3멀티플렉서(254)의 출력과 상기 OR 연산자(260)의 출력을 수신하도록 구성되는 AND 연산자(270)와;

상기 AND 연산자(270)에 연결되고 또한 상기 AND 연산자(270)의 출력과 그리고 상기 제1색상 코드워드(41, 44, 47)에 후행하는 두 개의 하위 비트들로서 10_{bin} 을 가지는 비트열을 가산하도록 구성되는 제2가산기(222)와;

상기 제1가산기(220)와 상기 제2가산기(222)에 연결되고 또한 상기 제1가산기(220)의 상기 출력과 상기 제2가산기(222)의 출력을 가산하도록 구성되는 제3가산기(224)와;

상기 제3가산기(224)에 연결되고 또한 상기 제3가산기(224)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제1 우측-이동된 열을 형성하도록 구성되는 제1우측 이동 연산자(230)와;

상기 제2가산기(222)에 연결되고 또한 상기 제2가산기(222)의 상기 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제2 우측-이동된 열을 형성하도록 구성되는 제2우측 이동 연산자(232)를 포함하는 것을 특징으로 하는 복호 시스템.

청구항 10

제6항 내지 제8항 중 어느 한 항에 있어서, 상기 제2복호기(200)는:

네 개의 텍셀(30 내지 36)들 중 절반이 상기 텍셀블록(10)의 상기 경계 내측에서 행 또는 열로 위치하는지를 기반으로 상기 제2색상 코드워드(42, 45, 48) 및 상기 제3색상 코드워드(43, 46, 49) 중 하나를 출력하도록 구성되는 제1멀티플렉서(250)와;

상기 제1색상 코드워드(41, 44, 47)를 부정하도록 구성되는 부정(not) 연산자와;

1에 설정된 캐리 입력을 가지고서 상기 제1멀티플렉서(250)와 상기 부정 연산자에 연결되고 또한 상기 제1멀티플렉서(250)의 출력과 상기 부정된 제1색상 코드워드를 가산하도록 구성되는 제1가산기(220)와;

상기 제1가산기(220)에 연결되고 또한 상기 제1가산기(220)의 출력에 대응하는 비트열을 좌측으로 1비트 위치 이동시켜 좌측-이동된 비트열을 형성하도록 구성되는 좌측 이동 연산자(210)와;

네 개의 텍셀(30 내지 36)들 중 상기 절반이 상기 텍셀블록(10)의 상기 경계 내측에서 행 또는 열로 위치하는지를 기반으로 텍셀(32, 24)의 행 좌표와 열 좌표 중 하나를 출력하도록 구성되는 제2멀티플렉서(252)와;

상기 좌측 이동 연산자(210)와 상기 제1가산기(220)에 연결되고 또한 상기 제2멀티플렉서(252)의 출력 중 최상위 비트를 기반으로 상기 제1가산기(220)의 상기 출력과 상기 좌측-이동된 비트열 중 하나를 출력하도록 구성되는 제3멀티플렉서(254)와;

상기 제2멀티플렉서(252)에 연결되고 또한 상기 제2멀티플렉서(252)의 상기 출력 중 최상위 비트와 상기 제2멀티플렉서의 상기 출력중 최하위 비트를 수신하도록 구성되는 OR 연산자(260)와;

상기 제3멀티플렉서(254)와 상기 OR 연산자(260)에 연결되고 또한 상기 제3멀티플렉서(254)의 출력과 상기 OR 연산자(260)의 출력을 수신하도록 구성되는 AND 연산자(270)와;

상기 AND 연산자(270)에 연결되고 또한 상기 AND 연산자(270)의 출력과 그리고 상기 제1색상 코드워드(41, 44, 47)에 후행하는 두 개의 하위 비트들을 가산하도록 구성되는 제2가산기(222)와;

상기 제1가산기(220)와 상기 제2가산기(222)에 연결되고 또한 상기 제1가산기(220)의 상기 출력과 상기 제2가산기(222)의 출력을 가산하도록 구성되는 제3가산기(224)와;

상기 제3가산기(224)에 연결되고 또한 상기 제3가산기(224)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제1 우측-이동된 열을 형성하도록 구성되는 제1우측 이동 연산자(230)와;

상기 제2가산기(222)에 연결되고 또한 상기 제2가산기(222)의 상기 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제2 우측-이동된 열을 형성하도록 구성되는 제2우측 이동 연산자(232)를 포함하는 것을 특징으로 하는 복호 시스템.

청구항 11

제6항 내지 제10항 중 어느 한 항에 있어서, 상기 제1복호기(100)는:

상기 제1색상 코드워드(41, 44, 47)를 부정하도록 구성되는 부정 연산자(140)와;

상기 부정 연산자(140)에 연결되고 또한 상기 제2색상 코드워드(42, 45, 48)와 상기 부정된 제1색상 코드워드를 가산하도록 구성되는 제1가산기(120)와;

상기 부정 연산자(140)에 연결되고 또한 상기 제3색상 코드워드(43, 46, 49)와 상기 부정된 제1색상 코드워드

를 가산하도록 구성되는 제2가산기(122)와;

상기 제1가산기(120)에 연결되고 또한 상기 제1가산기(120)의 출력에 대응하는 비트열을 좌측으로 1비트 위치 이동시켜 제1 좌측-이동된 열을 형성하도록 구성되는 제1좌측 이동 연산자(110)와;

상기 제2가산기(122)에 연결되고 또한 상기 제2가산기(122)의 출력에 대응하는 비트열을 좌측으로 1비트 위치 이동시켜 제2 좌측-이동 비트열을 형성하도록 구성되는 제2좌측 이동 연산자(112)와;

상기 제1가산기(120)와 상기 제1좌측 이동 연산자(110)에 연결되고 또한 텍셀(30)의 열 좌표의 최상위 비트를 기반으로 상기 제1가산기(120)의 상기 출력과 상기 제1 좌측-이동된 비트열 중 하나를 출력하도록 구성되는 제1멀티플렉서(150)와;

상기 제2가산기(122)와 상기 제2좌측 이동 연산자(112)에 연결되고 또한 상기 텍셀(30)의 행 좌표의 최우선 비트를 기반으로 상기 제2가산기(122)의 상기 출력과 상기 제1 좌측-이동된 비트열 중 하나를 출력하도록 구성되는 제2멀티플렉서(152)와;

상기 열 좌표의 상기 최상위 비트와 상기 열 좌표의 상기 최하위 비트를 수신하도록 구성되는 제1 OR 연산자(160)와;

상기 행 좌표의 상기 최상위 비트와 상기 행 좌표의 상기 최하위 비트를 수신하도록 구성되는 제2 OR 연산자(162)와;

상기 제1멀티플렉서(150)와 상기 제1 OR 연산자(160)에 연결되고 또한 상기 제1멀티플렉서(150)의 출력과 상기 제1 OR 연산자(160)의 출력을 수신하도록 구성되는 제1 AND 연산자(170)와;

상기 제2멀티플렉서(152)와 상기 제2 OR 연산자(162)에 연결되고 또한 상기 제2멀티플렉서(152)의 출력과 상기 제2 OR 연산자(162)의 출력을 수신하도록 구성되는 제2 AND 연산자(172)와;

상기 제1 AND 연산자(170)에 연결되고 또한 상기 제1 AND 연산자(170)의 출력과 그리고 상기 제1색상 코드워드(41, 44, 47)에 후행하는 두 개의 하위 비트들로서 10_{bin} 을 가지는 비트열을 가산하도록 구성되는 제3가산기(124)와;

상기 제1가산기(120)와 상기 제3가산기(124)에 연결되고 또한 상기 제1가산기(120)의 상기 출력과 상기 제3가산기(124)의 출력을 가산하도록 구성되는 제4가산기(126)와;

상기 제2가산기(122)와 상기 제2 AND 연산자(172)에 연결되고 또한 상기 제2가산기(122)의 상기 출력과 상기 제2 AND 연산자(172)의 출력을 가산하도록 구성되는 제5가산기(128)와;

상기 제3가산기(124)와 상기 제2 AND 연산자(172)에 연결되고 또한 상기 제3가산기(124)의 상기 출력과 상기 제2 AND 연산자(172)의 상기 출력을 가산하도록 구성되는 제6가산기(121)와;

상기 제4가산기(126)와 상기 제2 AND 연산자(172)에 연결되고 또한 상기 제4가산기(126)의 상기 출력과 상기 제2 AND 연산자(172)의 상기 출력을 가산하도록 구성되는 제7가산기(123)와;

상기 제3가산기(124)와 상기 제5가산기(128)에 연결되고 또한 상기 제3가산기(124)의 상기 출력과 상기 제5가산기(128)의 출력을 가산하도록 구성되는 제8가산기(125)와;

상기 제4가산기(126)와 상기 제5가산기(128)에 연결되고 또한 상기 제4가산기(126)의 상기 출력과 상기 제5가산기(128)의 상기 출력을 가산하도록 구성되는 제9가산기(127)와;

상기 제6가산기(121)에 연결되고 또한 상기 제6가산기(121)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제1 우측-이동된 열을 형성하도록 구성되는 제1우측 이동 연산자(130)와;

상기 제7가산기(123)에 연결되고 또한 상기 제7가산기(123)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제2 우측-이동된 열을 형성하도록 구성되는 제2우측 이동 연산자(132)와;

상기 제8가산기(125)에 연결되고 또한 상기 제8가산기(125)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제3 우측-이동된 열을 형성하도록 구성되는 제3우측 이동 연산자(134)와; 그리고

상기 제9가산기(127)에 연결되고 또한 상기 제9가산기(127)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제4 우측-이동된 열을 형성하도록 구성되는 제4우측 이동 연산자(136)를 포함하는 것을 특징으로 하는 복호 시스템.

청구항 12

제6항 내지 제10항 중 어느 한 항에 있어서, 상기 제1복호기(100)는:

상기 제1색상 코드워드(41, 44, 47)를 부정하도록 구성되는 부정 연산자와;

1에 설정된 캐리 입력을 가지고서 상기 부정 연산자에 연결되고 또한 상기 제2색상 코드워드(42, 45, 48)와 상기 부정된 제1색상 코드워드를 가산하도록 구성되는 제1가산기(120)와;

1에 설정된 캐리 입력을 가지고서 상기 부정 연산자에 연결되고 또한 상기 제3색상 코드워드(43, 46, 49)와 상기 부정된 제1색상 코드워드를 가산하도록 구성되는 제2가산기(122)와;

상기 제1가산기(120)에 연결되고 또한 상기 제1가산기(120)의 출력에 대응하는 비트열을 좌측으로 1비트 위치 이동시켜 제1 좌측-이동된 비트열을 형성하도록 구성되는 제1좌측 이동 연산자(110)와;

상기 제2가산기(122)에 연결되고 또한 상기 제2가산기(122)의 출력에 대응하는 비트열을 좌측으로 1비트 위치 이동시켜 제2 좌측-이동된 비트열을 형성하도록 구성되는 제2좌측 이동 연산자(112)와;

상기 제1가산기(120)와 상기 제1좌측 이동 연산자(110)에 연결되고 또한 텍셀(30)의 열 좌표의 최상위 비트를 기반으로 상기 제1가산기(120)의 상기 출력과 상기 제1 좌측-이동된 비트열 중 하나를 출력하도록 구성되는 제1멀티플렉서(150)와;

상기 제2가산기(122)와 상기 제2좌측 이동 연산자(112)에 연결되고 또한 상기 텍셀(30)의 행 좌표의 최상위 비트를 기반으로 상기 제2가산기(122)의 상기 출력과 상기 제2 좌측-이동된 비트열 중 하나를 출력하도록 구성되는 제2멀티플렉서(152)와;

상기 열 좌표의 상기 최상위 비트와 상기 열 좌표의 상기 최하위 비트를 수신하도록 구성되는 제1 OR 연산자(160)와;

상기 행 좌표의 상기 최상위 비트와 상기 행 좌표의 상기 최하위 비트를 수신하도록 구성되는 제2 OR 연산자(162)와;

상기 제1멀티플렉서(150)와 상기 제1 OR 연산자(160)에 연결되고 또한 상기 제1멀티플렉서(150)의 출력과 상기 제1 OR 연산자(160)의 출력을 수신하도록 구성되는 제1 AND 연산자(170)와;

상기 제2멀티플렉서(152)와 상기 제2 OR 연산자(162)에 연결되고 또한 상기 제2멀티플렉서(152)의 출력과 상기 제2 OR 연산자(162)의 출력을 수신하도록 구성되는 제2 AND 연산자(172)와;

상기 제1 AND 연산자(170)에 연결되고 또한 상기 제1 AND 연산자(170)의 출력과 그리고 상기 제1색상 코드워드(41, 44, 47)에 후행하는 두 개의 하위 비트들로서 10_{bin} 을 가지는 비트열을 가산하도록 구성되는 제3가산기(124)와;

상기 제1가산기(120)와 상기 제3가산기(124)에 연결되고 또한 상기 제1가산기(120)의 상기 출력과 상기 제3가산기(124)의 출력을 가산하도록 구성되는 제4가산기(126)와;

상기 제2가산기(122)와 상기 제2 AND 연산자(172)에 연결되고 또한 상기 제2가산기(122)의 상기 출력과 상기 제2 AND 연산자(172)의 출력을 가산하도록 구성되는 제5가산기(128)와;

상기 제3가산기(124)와 상기 제2 AND 연산자(172)에 연결되고 또한 상기 제3가산기(124)의 상기 출력과 상기 제2 AND 연산자(172)의 상기 출력을 가산하도록 구성되는 제6가산기(121)와;

상기 제4가산기(126)와 상기 제2 AND 연산자(172)에 연결되고 또한 상기 제4가산기(126)의 상기 출력과 상기 제2 AND 연산자(172)의 상기 출력을 가산하도록 구성되는 제7가산기(123)와;

상기 제3가산기(124)와 상기 제5가산기(128)에 연결되고 또한 상기 제3가산기(124)의 상기 출력과 상기 제5가산기(128)의 출력을 가산하도록 구성되는 제8가산기(125)와;

상기 제4가산기(126)와 상기 제5가산기(128)에 연결되고 또한 상기 제4가산기(126)의 상기 출력과 상기 제5가산기(128)의 상기 출력을 가산하도록 구성되는 제9가산기(127)와;

상기 제6가산기(121)에 연결되고 또한 상기 제6가산기(121)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제1 우측-이동된 열을 형성하도록 구성되는 제1우측 이동 연산자(130)와;

상기 제7가산기(123)에 연결되고 또한 상기 제7가산기(123)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제2 우측-이동된 열을 형성하도록 구성되는 제2우측 이동 연산자(132)와;

상기 제8가산기(125)에 연결되고 또한 상기 제8가산기(125)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제3 우측-이동된 열을 형성하도록 구성되는 제3우측 이동 연산자(134)와; 그리고

상기 제9가산기(127)에 연결되고 또한 상기 제9가산기(127)의 출력에 대응하는 비트열을 우측으로 2비트 위치 이동시켜 제4 우측-이동된 열을 형성하도록 구성되는 제4우측 이동 연산자(136)를 포함하는 것을 특징으로 하는 복호 시스템.

청구항 13

제6항 내지 제12항 중 어느 한 항에 있어서, 상기 값 선택기(500)는:

상기 제1복호기(100)에 연결되고 또한 제1텍셀값을 출력하도록 구성되는 제1멀티플렉서(510)와;

상기 제1복호기(100)와 상기 제2복호기(200)에 연결되고 또한 제2텍셀값을 출력하도록 구성되는 제2멀티플렉서(520)와;

상기 제1복호기(100)와, 상기 제2복호기(200)와 상기 제3복호기(300)에 연결되고 또한 제3텍셀값을 출력하도록 구성되는 제3멀티플렉서와; 그리고

상기 제1복호기(100)와, 상기 제2복호기(200)와 그리고 상기 제4복호기(400)에 연결되고 또한 제4텍셀값을 출력하도록 구성되는 제4멀티플렉서(540)를 포함하는 것을 특징으로 하는 복호 시스템.

청구항 14

적어도 하나의 암호화된 텍셀블록(40)을 복호함으로써 복호된 화소를 생성하는 방법에 있어서, 상기 방법은:

각각이 다른 N-1개 복호기들의 회로와는 상이한 회로를 가지고 또한, 적어도 N개 텍셀(30 내지 36)들 중 적어도 하나를 포함하는 텍셀블록(10)의 경계에 관해 적어도 N개의 텍셀값들과 관련된 상기 적어도 N개 텍셀(30 내지 36)들의 위치를 기반으로, 적어도 하나의 각 텍셀값을 생성하기 위해 암호화된 텍셀블록(40)을 복호하도록 각각 구성되는, N개의 상이한 복호기(100, 200, 300, 400)들 중 적어도 하나로부터 상기 적어도 N개의 텍셀값들을 선택적으로 출력하는 단계와; 그리고

상기 적어도 N개의 텍셀값들을 기반으로 상기 암호화된 화소의 복호된 화소값을 계산하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 15

제14항에 있어서, 상기 선택적으로 출력하는 단계는, 만일 상기 N개의 텍셀값들과 관련된 N개의 텍셀(30 내지 36)들이 텍셀블록(10)의 상기 경계 내측에 위치한다면 상기 N개의 복호기(100, 200, 300, 400)들 중 제1복호기(100)로부터 N개의 텍셀값들을 선택적으로 출력하는 것을 포함하는 것을 특징으로 하는 방법.

청구항 16

제14항 또는 제15항에 있어서, 상기 선택적으로 출력하는 단계는, 만일 상기 N개의 텍셀값들과 관련된 N개의 텍셀(30 내지 36)들 중 절반이 상기 텍셀블록(10)의 상기 경계 내측에 위치하고 또한 상기 N개의 텍셀(30 내지 36)들 중 절반이 이웃하는 텍셀블록(12, 14)의 경계 내측에 위치한다면, 상기 N개의 복호기(100, 200, 300, 400)들 중 제1복호기(100)로부터 N개의 텍셀값들 중 절반을 선택적으로 출력하고 또한 상기 N개의 복호기(100, 200, 300, 400)들 중 제2복호기(200)로부터 상기 N개의 텍셀값들 중 나머지 절반을 선택적으로 출력하는 것을 포함하는 것을 특징으로 하는 방법.

청구항 17

제14항 내지 제16항 중 어느 한 항에 있어서, 상기 N은 4와 동일하고, 그리고 상기 선택적으로 출력하는 단계는, 만일 상기 네 개의 텍셀(30 내지 36)들 중 하나가 상기 텍셀블록(10)의 상기 경계 내측에 위치하고 또한 나머지 세 개의 텍셀들이 이웃하는 각 텍셀블록(12, 14, 16)의 경계 내측에 위치한다면, 상기 네 개의 복호기(100, 200, 300, 400)들 중 제1복호기(100)로부터 상기 네 개의 텍셀값들 중 제1텍셀값을 선택적으로 출력하고, 상기 네 개의 복호기(100, 200, 300, 400)들 중 제2복호기(200)로부터 상기 네 개의 텍셀값들 중 제2텍셀

값을 선택적으로 출력하고, 상기 네 개의 복호기(100, 200, 300, 400)들 중 제3복호기(300)로부터 상기 네 개의 텍셀값들 중 제3텍셀값을 선택적으로 출력하고, 그리고 상기 네 개의 복호기(100, 200, 300, 400)들 중 제4복호기(400)로부터 상기 네 개의 텍셀값들 중 제4텍셀값을 선택적으로 출력하는 것을 포함하는 것을 특징으로 하는 방법.

명세서

기술분야

[0001] 본 발명은 복호화에 관한 것으로서, 특히 암호화된 텍셀 블록들에서 동작할 수 있는 복호 시스템 및 방법에 관한 것이다.

배경기술

[0002] 그래픽처리장치(graphics processing units:GPU)에 대한 성능을 향상시키고 할 때, 한 가지 해결책은, 메모리 대역폭 소비를 감소시키기 위한 다양한 기술들을 적용하는 것이다. 대역폭 감소는 또한, 처리전력에 대한 성능 성장률이 대역폭에 대한 성능 성장률과 무작위 액세스 메모리(RAM)에 대한 잠재정보보다 훨씬 크기 때문에 점점 더 중요해지고 있다.

[0003] 텍스처 이미지 압축(texture compression)은 대역폭 필요조건을 감소시키기 위한 한 가지 보편적인 방식이다. 메모리에 압축된 형식으로 텍스처를 저장하고 또한 버스(bus)를 통해 압축된 데이터의 블록들을 전송함으로써, 텍스처 대역폭은 실질적으로 감소된다.

[0004] 오늘날, 가장 많이 사용하는 텍스처 이미지 압축 시스템은 윈도우 기반 시스템들과 엑스박스(Xbox)를 위한 DXTC[1]와 이동 전화기용의 ETC[2]이다. 이들 두 시스템들은 텍스처로 표시되는 이미지를 4 x 4 텍셀의 텍셀(texel) 블록들로 분할, 그런 다음에 텍셀들의 적, 녹, 청(RGB) 데이터들은 (8+8+8)x16=384 비트에서 64비트로 압축된다. 그러므로, 각각의 텍셀 블록에는 동일 수의 비트들이 제공된다. 이는 중요한데, 압축해제에 사용되는 래스터라이저(rasterizer)가 텍스처의 소정 부분에 액세스할 필요성이 있을 수 있고 또한 원하는 텍셀 블록의 메모리 어드레스를 쉽게 계산할 수 있는 것이 필요하기 때문이다. 즉, 고정율 코덱(fixed rate codec), 즉 모든 텍셀 블록이 동일 양의 저장 공간을 차지하는 코덱이 매우 바람직하고 또한 오늘날 텍스처 압축 알고리즘 중에서 표준이다.

[0005] ETC 복호기의 개발은, 소위 T-모드, H-모드 및 PLANAR-모드를 처리하기 위한 여분의 기능을 가지는 ETC2 복호기[3]가 되었다. T- 및 H-모드들은 불균일한 색상의 분포를 가지는 텍셀 블록들과 두 그룹으로 배열되는 색상들을 각각 처리하도록 디자인된다. ETC 복호기의 하드웨어 회로는 추가적인 T- 및 H-모드들을 포함하도록 상당한 정도로 재사용할 수 있다. 비록 PLANAR-모드가 ETC 및 T- 및 H-모드들과 상당히 다르지만, 부드러운 색상 변위(color transition)을 텍셀 블록들을 처리하도록 설계된다 한다. PLANAR-모드를 위한 복합 하드웨어 회로는, ETC2 복호기의 복잡도가 ETC 복호기와 비교할 때 상당히 증가할 수 있다는 것을 의미한다.

[0006] 그러므로, PLANAR-모드의 효율적인 구현을 제공할 필요가 있고, 특히 비용 효율적인 회로를 사용하는 하드웨어 구현으로서 제공될 수 있는 구현들을 제공할 필요가 있다. 이러한 비용 효율적인 회로는, PLANAR-모드가 ETC-복호기와 T- 및 H-모드와 함께 사용되게 되어 ETC2 복호기를 형성한다면 특히 바람직하다.

[0007] 비슷한 문제가 기술분야에 존재하는데, 즉 구현의 복잡성을 감소시킬 목적으로 복호 시스템에 적용될 수 있는 기술에 대한 일반적인 필요성이 있다.

발명의 내용

해결하려는 과제

[0008] 본 발명의 목적은 암호화된 텍셀 블록들의 효율적인 복호를 제공하는 것이다.

[0009] 특히, 본 발명의 목적은 하드웨어에서 효율적으로 구현될 수 있는 복호구조를 제공하는 것이다.

과제의 해결 수단

[0010] 이들 및 다른 목적들은 여기에서 기술하는 실시예들에 의해 이루어진다.

- [0011] 간략하게, 한 실시예에는 암호화된 적어도 한 텍셀 블록을 기반으로 복호화된 화소(pixel)를 생성하기 위한 복호 시스템에 관련된다. 복호 시스템은, 각각이 하 이상의 텍셀 값들을 생성하기 위해 암호화된 텍셀 블록을 복호하도록 구성되는 N 개의 병렬 복호기들을 포함한다. 상기 실시예에 따라, N개의 복호기들의 회로들은 서로 간에 비교하였을 때 상이하여, N 개의 복호기들의 각 복호기는 다른 N-1 개의 복호기들의 회로와는 상이한 고유 회로를 가진다. 이는, 복호기 회로들이 상이한 수준의 복잡도를 가지도록 해주는데, 복호기들이 처리할 수 있는 텍셀 블록 내에서 텍셀 위치들에 대하여 복호기들이 상이한 용량을 가지기 때문이다.
- [0012] 복호 시스템은 또한 N개의 복호기들에 연결되고 또한 선택적으로 N개의 텍셀 값들을 출력하도록 구성되는 값 선택기(value selector)를 포함한다. 이들 N개의 텍셀 값들은 N개의 복호기들 중에서 하나 이상으로부터 더 출력된다. 어느 복호기(들)로부터 출력해야할 특정 텍셀 값들은, N개의 텍셀들 중 적어도 하나를 포함하는 텍셀 블록에 관해, N개의 텍셀 값들을 가지는 N개 텍셀들의 패치(patch)의 위치에 의존한다. 즉, 텍스처의 텍셀 블록 경계들에 대한 N개 텍셀들의 위치는, 값 선택기에 의해 선택되어, 화소 계산기로 전달되는 복호기 출력들에 영향을 준다. 화소 계산기는 값 선택기에 의해 선택적으로 출력되는 N개 화소 값들을 기반으로 복호된 화소 값을 계산하도록 구성된다.
- [0013] 값 선택기에 의한 조심스러운 선택은, N개 복호기들 중 적어도 다수가 다른 복호기보다 덜 복잡한 회로를 가질 수 있도록, 복호기들의 설계를 가능하게 한다. 이는, 복호기들 중 몇몇만이 텍셀 블록 내 소정의 텍셀 위치들에 대한 텍셀 값들을 복호하고 또한 출력하도록 제한함으로써 가능해진다. 따라서, N개 복호기들은 동일한 회로를 가질 필요가 없고, 반대로 회로에 대한 하드웨어 구조에 관해 상이한 수준의 복잡도를 가지는 상이한 회로를 가진다.
- [0014] 한 실시예는 적어도 하나의 암호화된 텍셀 블록을 복호함으로써 화소를 생성하는 방법을 규정한다. 방법은, 각각이, 다른 N-1 개 복호기들의 회로와 상이하고 또한 고유한 회로를 가지는 N개 상이한 복호기들 중 적어도 하나로부터 N개 텍셀 값들을 선택적으로 출력하는 것을 포함한다. 각각의 복호기는, 입력 암호화된 텍셀 블록을 기반으로 적어도 하나의 텍셀 값을 더 출력한다. 복호를 위해 화소의 화소값의 계산을 위해 N개 복호기들로부터 텍셀 값들 중 어느 것을 사용하느냐의 선택은, 텍셀 블록의 경계에 관해 N개 텍셀 값들과 관련된 N개 텍셀들의 위치를 기반으로 수행된다. 선택된 N개 텍셀 값들은 화소값을 계산하는데 사용된다.

발명의 효과

- [0015] 본 발명의 실시예들은, 병렬 복호기들 중 적어도 몇몇에 필요한 복잡도를 줄임으로써 화소 복호 구조의 효율적인 구현이 가능하게 한다. 이에 의해, 복호 구조의 전체 복잡도는 N개의 동일한 병렬 복호기를 가지는 선행 기술과 비교하면 줄어든다.

도면의 간단한 설명

- [0016] 도 1a 내지 1d는 이중선형(bilinear) 패치 관련 텍셀 경계들의 상이한 위치들을 설명하는 도면.
- 도 2는 한 실시예에 따른 복호 시스템을 포함하는 복호장치의 개략적인 개요도.
- 도 3은 한 실시예에 따른 복호 시스템의 설명도.
- 도 4는 한 실시예에 따른 암호화된 텍셀 블록을 설명하는 도면.
- 도 5는 한 텍셀 블록에서 텍셀 속성값들의 평면 근사화를 채용하는 개념을 설명하는 도면.
- 도 6은 도 3에서 제4복호기의 하드웨어 구현예를 보여주는 도면.
- 도 7은 도 3에서 제3복호기의 하드웨어 구현예를 보여주는 도면.
- 도 8은 도 3에서 제2복호기의 하드웨어 구현예를 보여주는 도면.
- 도 9는 도 3에서 제1복호기의 하드웨어 구현예를 보여주는 도면.
- 도 10은 7-8 비트 확장기의 하드웨어 구현예를 보여주는 도면.
- 도 11은 클램퍼의 하드웨어 구현을 보여주는 도면.
- 도 12는 클램퍼의 다른 하드웨어 구현을 보여주는 도면.
- 도 13은 도 3의 값 선택기의 하드웨어 구현예를 보여주는 도면.

도 14는 한 실시예에 따른 복호방법을 설명하는 흐름도.

발명을 실시하기 위한 구체적인 내용

- [0017] 첨부도면과 함께 다음의 상세한 설명을 참조함으로써, 본 발명의 추가 목적과 장점들과 함께 본 발명을 가장 잘 이해할 수 있을 것이다.
- [0018] 도면 전체를 통해, 비슷하거나 대응하는 요소들에는 동일 참조부호들을 사용한다.
- [0019] 실시예들은 복호된 화소값들을 생성할 목적으로 암호화된 또는 압축된 텍스처 요소(텍셀:texel) 블록들의 복호 또는 압축해제에 관련된다.
- [0020] 복호화는, 제한하고자 하는 것은 아니지만, 게임과 같은 3차원(3D) 그래픽과, 지도와 장면과, 예컨대, 만화 메시지(animated message)와 같은 3D 메시지, 화면보호기, 인간-기계인터페이스(MMI)에 사용하기에 매우 적합하다. 그러므로, 실시예들은 일차원(1D), 2차원(2D) 또는 3D 이미지와 같은 다른 유형의 이미지들 또는 그래픽들을 복호하는데 사용할 수 있다.
- [0021] 3D 그래픽 렌더링(rendering) 프로세서는 전형적으로 세 개의 부-단계(하위단계)를 포함한다. 간략하게 설명하면, 제1단계와, 응용단계와, 여러 삼각형들을 생성하는 단계를 포함한다. 이들 삼각형들의 모서리들은 제2 단계, 즉 기하단계(geometry stage)에서 변환되고(transformed), 투영되고(projected) 그리고 라이팅(lit)된다. 제3단계, 즉 래스터화 단계(rasterization stage)에서, 종종 텍스처(texture)라고 부르는 이미지들이 상기 삼각형들 상에 "부착(glued)"될 수 있어서, 렌더링된 영상의 사실성을 증가시킨다. 상기 제3단계는 또한 z-버퍼를 사용하여 소팅(sorting)을 수행한다.
- [0022] 그러나, 텍스처, 특히 3D 이미지와 그래픽들의 렌더링은, 그래픽 시스템들에 필요한 메모리 대역폭과 프로세싱 전력의 면에서 많은 자원이 필요한 임무이다. 예컨대, 텍스처들은 메모리의 면에서 많은 비용이 드는데, 텍스처들은 반드시 고속 온-칩 메모리(fast on-chip memory)에 위치하여야만 하기 때문이고, 또한 메모리 대역폭의 면에서도 많은 비용이 드는데, 텍스처들은 단일 화소를 작성(draw)하기 위해 반드시 여러 차례 액세스될 수 있기 때문이다. 또한 한 화소를 작성하기 위해 다중 텍스처들은 때때로 필요할 수 있다.
- [0023] 대역폭과 프로세싱 전력요구를 줄이기 위하여, 텍스처 암호화 방법 또는 시스템이 전형적으로 채용된다. 이러한 암호화 시스템은 렌더링 동안에 값비싼 온-칩 메모리의 효율적인 사용과 낮은 메모리 대역폭이 되게 하여, 저전력 소비 및/또는 고속 렌더링이 되게 한다. 이러한 대역폭과 프로세싱 전력요구의 감소는 특히, 작은 양의 메모리와, 아주 적은 대역폭과 제한된 전력, 즉 배터에 의해 공급되는 제한된 전력을 가지는, 이동유닛과 전화기와 같은 신 클라이언트(thin clients)에 대해 중요하다.
- [0024] 기술분야에서, 텍스처들은 소위 텍셀 블록들로 분할되는데, 블록들 각각은 다수의 텍셀들을 포함한다. 이러한 텍셀 블록은 전형적으로, $M_r \times M_c$ 의 크기와 같은 직사각형 또는 정사각형인데, 여기에서 M_r , M_c 들은, M_r 와 M_c 둘 다가 동시에 1이 아니라는 단서를 가지는 양의 정수들이다. 바람직하게 $M_r=2m_r$ 및 $M_c=2m_c$ 이고, 여기서 m_r 과 m_c 는 양의 정수이다. 전형적인 구현 $M_r=M_c$ 와 바람직한 이러한 블록 실시예들은 4 x 4 텍셀들일 수 있다.
- [0025] 기술분야에서, 텍스처의 텍스처 요소를 나타내기 위해 화소 또는 이미지요소들이 때때로 채용된다. 다음의 상세한 설명에서, 텍스처의 텍스처 요소를 나타내기 위해 텍셀이 사용된다고 하더라도, 래스터화와 관련해 이러한 텍셀들을 생성하는 영상 요소를 나타내기 위해 화소가 사용된다.
- [0026] 텍셀은 텍셀값과 관련되고 또한 화소는 화소값과 관련된다. 텍셀과 화소값들은 색상값들일 수 있다. 색상값(color value)은 적, 녹, 청(RGB) 색상 또는 휘도 및 색차와 같은 다양한 색상 포맷들로 표시될 수 있다. 게다가, 텍셀값은 정상(normal) 또는 Z값과 같은, 텍셀의 다른 특성들을 나타낼 수 있다. 다음의 상세한 설명에서, 텍셀값은 색상값으로 비제한적으로 예시된다.
- [0027] 전형적인 래스터라이저(rasterizer)는 한 번에 텍스처를 단일 텍셀로 액세스하지 않는다. 대신에, 예컨대, 2 x 2 텍셀과 같은, 텍셀들로 구성된 작은 집단이 채용된다. 예컨대, 이중선형 필터링 동안에, 이러한 네 개의 이웃하는 텍셀들이 판독되어 이중선형적으로 필터링된 값, 즉 화소값을 생산한다. 기술분야에서, 화소값을 생성하기 위해 함께 처리되는(co-processed) 이웃하는 텍셀들의 그룹은 전형적으로 패치(patch)로 표시된다. 그러므로, 이중선형 필터링에 채용되는 2 x 2 텍셀 그룹은 이중선형 패치로 표시된다. 이 개념은, 예컨대 삼중선형(trilinear) 필터링으로 더 확장할 수 있다. 삼중선형 필터링은 두 개의 상이한 mip맵(mipmap)들로부터 두 개의 이중선형 패치들을 사용한다.

- [0028] 텍셀들의 패치들을 처리하는데 있어서 한 가지 문제점은, 패치 내 텍셀들이 한 텍셀 블록보다 더 걸쳐 이어진다는(span) 것이다. 이를 도 1a 내지 도 1d에 도식적으로 설명하고 있다. 도 1a는 네 개의 텍셀(30 - 36)들로 된 이중선형 패치(20)가 단일 텍셀블록(10)에서 발견되는 경우를 설명한다. 도 1b와 1c는 이중선형 패치(20)가, 나란히(도 1b) 또는 서로 위에(도 1c)에 존재하는 두 개의 화소 블록들(10, 12, 14)에 걸쳐 이어지는 다른 상황을 나타내는 것이다. 도 1d에서는, 각각의 텍셀블록(10 - 16)들 내에 이중선형 패치(20)의 한 텍셀(30 - 36)들을 가지는 극한 경우를 도시하고 있다.
- [0029] 이러한 문제점을 해결하기 위한 전통적인 방식은, 네 개의 동일한 복호기들을 사용하는 것인데, 복호기 각각은 이중선형 패치 내 규정된 텍셀을 복호한다. 예컨대, 도 1a의 경우에 있어서, 텍셀블록(10)을 나타내는 암호화된 텍셀블록이 네 개의 복호기들에 입력된다. 제1복호기는 텍셀(1, 2)(30)를 출력하고, 텍셀(2, 2)는 제2복호기로부터 획득하고 그리고 제3 및 제4복호기들은 이중선형 패치의 텍셀(1, 3) 및 (2, 3)(36)을 생성한다. 텍셀블록(10)의 상측 좌측 텍셀에서 좌표시스템이 시작한다고 추정한다. 도 1d에서, 텍셀블록(10)의 암호화된 텍셀블록이 제1복호기에 입력되어 텍셀(3, 3)(10)을 생성하고, 텍셀블록(12)의 암호화된 텍셀블록이 제2복호기에 의해 처리되어 텍셀(0, 3)(32)을 얻는다. 텍셀(3, 0)(34)은 텍셀블록(14)에 대해 암호화된 텍셀블록을 기반으로 제3복호기로부터 얻고 그리고 제4복호기는 텍셀블록(16)의 암호화된 텍셀블록으로부터 텍셀(0, 0)(36)를 출력한다.
- [0030] 따라서, 이 방식에서 어떠한 유형의 구성(도 1a 내지 1d)이 나타난다 하더라도 매 클럭 사이클마다 2 x 2 이중선형 패치를 생성할 수 있다. 고성능 래스터화 하드웨어는 매 클럭 사이클마다 이러한 두 개의 이중선형 패치들을 생성하여 삼중선형 필터링이 가능하게 한다. 그러나, 이 경우에 여덟 개의 동일 복호기들이 선행기술에 따라 필요하다.
- [0031] 그러나, 병렬 복호기들에서 텍셀들을 합동으로 처리하는 개념은 복호 시스템의 복잡도를 극적으로 증가시키는데, 이는 신 클라이언트들에게는 문제점이다. 예컨대, ECT-복호 시스템에 앞서 언급한 PLANAR-모드를 부가하는 것은, 복호기 당 6개 가산기(adder) 내지 21개 가산기들[3]과 같이 가산기들의 숫자로 규정되는 복잡도를 증가시키게 된다. 그러므로, PLANAR 는 ETC보다 3.5 개 이상의 가산기들을 사용하고 또한 네 개의 ETC 복호기들과 네 개의 PLANAR 복호기들을 가지는 복호 시스템의 복잡도가 3.5 정도로 증가할 수 있다.
- [0032] 여기에서 기술한 실시예들은 동일한 병렬 복호기들을 사용하지 않고서 복호 시스템의 복잡도를 줄인다. 반대로, 만일 상이한 숫자의 텍셀들과 패치 내 상이한 텍셀 위치들을 처리하도록 복호기들이 디자인된다면 상이한 복호기 회로를 사용할 수 있다.
- [0033] 일반적으로 실시예에서, 복호 시스템은 N개의 상이한 복호기들을 포함한다. 이러한 복호 시스템(1)이 도 3에 도시되어 있고, N은 4로서 예시된다. 일반적인 실시예에서, N은 2도 동일하거나 또는 이보다 큰 정수이고, 바람직하게는 4이다. 각 복호기(100, 200, 300, 400)는 다른 N-1개 복호기들의 회로와는 상이한 회로를 가진다. 그러므로, 복호 시스템(1)은 선행기술에서와 같이 동일한 복호기들을 포함하지 않고, 반대로 상이한 복호기(100, 200, 300, 400)들은 상이한 고유 회로들을 가진다. N개 복호기(100, 200, 300, 400)들의 회로에서 차이점은, 복호기들 중 하나의회로 요소들의 조합이 다른 복호기들의 회로 요소들의 조합과 상이하다는 것을 의미한다. 따라서, 복호기(100, 200, 300, 400)들은 상이한 구현 비용과 복잡도를 가진다.
- [0034] 복호기(100, 200, 300, 400) 각각은 적어도 하나의 텍셀값을 생성하도록 암호화된 텍셀블록을 복호하도록 더 구성된다. 복호 시스템(1)은 또한 N개 복호기(100, 200, 300, 400)들에 연결되고 또한 적어도 N개 텍셀값들, 바람직하게는 N개 텍셀값을 선택적으로 출력하도록 구성되는 값 선택기(value selector)(500)를 포함한다. 이들 적어도 N개 텍셀값들은 복호기(100, 200, 300, 400)들 중 적어도 하나로부터 출력된다. 부수적으로, 값 선택기(50)는 적어도 N개 텍셀값들을 선택하고 또한, N개 텍셀들 중 적어도 하나를 포함하는 텍셀블록의 경계에 관련해 적어도 N개 텍셀값들과 관련된 적어도 N개 텍셀들의 위치들을 기반으로 텍셀값들이 출력되어야만 하는 복호기(들)를 선택한다. 그러므로, 텍셀블록에 관해, 그 텍셀값이 복호 시스템(1)에 의해 생성되어야만 하는 N개 텍셀들의 위치들은, N개 복호기(100, 200, 300, 400)들 중 어느 복호기로부터 또는 어느 복호기들로부터 텍셀값들이 패치(fetch)되어야 하는지를 결정한다. 복호 시스템(1)의 화소 계산기(600)는 값 선택기(500)에 연결되고 또한 값 선택기(500)에 의해 선택적으로 출력되는 적어도 N개 텍셀값들 기반으로 복호된 화소의 복호된 화소값을 계산하도록 구성된다.
- [0035] 상기 일반적인 실시예는 텍셀블록 내에서 소정의 텍셀 위치들에서 텍셀들을 처리할 수만 있는 복호기들 중의 제한된 몇몇에 의해 복호기(100, 200, 300, 400)의 복잡도를 줄일 수 있다. 따라서, 복호기(100, 200, 300, 400)의 회로는, 이러한 복호기들의 감소가 없는 선행기술의 경우와 비교하면 상당히 줄어들 수 있다.

- [0036] 바람직한 실시예에서, N개 복호기(100, 200, 300, 400)들 중 제1복호기(100)는 N개 텍셀값들을 생성하기 위해 입력 암호화된 텍셀블록을 복호하도록 구성된다. 그러면, 값 선택기(500)는, 제1복호기(100)로부터 이들 N개 텍셀값들을 선택적으로 출력하고 또한 만일 N개 텍셀들이 텍셀블록의 경계들 내측에 위치한다면 화소 계산기(500)로 선택적으로 출력하도록 구성된다. 도 1a는, N=4의 경우를 도시하고 있다. 그러므로, 만일 이중선형 패치(20)의 네 개의 텍셀(30 내지 36)들 모두가 단일 텍셀블록(10) 내측에 위치한다면, 제1복호기(100)에 의해 네 개의 텍셀값들이 모두 생성된다. 이는, 다른 복호기(200, 300, 400)들로부터의 소정의 출력은 값 선택기(500)에 의해 선택되지 않거나 또는 복호된 화소값을 계산하기 위해 화소 계산기(600)에 의해 사용되지 않는다는 것을 의미한다.
- [0037] 만일 N개 텍셀들 중 반(30, 34(30, 32))이 텍셀블록(10)의 경계들 내측에 위치하고 또한 나머지 반(32, 36(34, 36))들이 이웃하는 화소블록(12(14))의 경계들 내측에 위치한다면(도 1b(도 1c) 참조), 값 선택기(500)는 바람직하게, 제1복호기(100)로부터 N개 텍셀값들 중 반을 선택적으로 출력하고 또한 제2복호기(200)로부터 N개 텍셀값들 중 나머지 반을 선택적으로 출력하도록 구성된다.
- [0038] N=4인 바람직한 실시예에서, 제1복호기(100)는 네 개의 텍셀값들을 생성하기 위해 입력 암호화된 텍셀블록을 복호하도록 구성되는 제1회로를 가진다. 제2복호기(200)는 제2회로를 가지고 또한 입력 암호화된 텍셀블록을 기반으로 두 개의 텍셀값들을 출력한다. 제3(300) 및 제4복호기(400)기들은 제3 및 제4회로를 각각 포함한다. 제3(300) 및 제4복호기(400)들은 입력 암호화된 텍셀블록에 주어진 단일 텍셀값들 각각 출력한다.
- [0039] 도 1d에서 도시된 상황에서, 바람직하게 네 개의 모든 복호기(100, 200, 300, 400)들은 값 선택기(500)에 의해 선택되는 텍셀값들을 출력한다. 그러므로, 값 선택기(500)는 제1복호기(100)로부터는 제1텍셀값을, 제2복호기(200)로부터는 제2텍셀값을, 제3복호기(300)로부터는 제3텍셀값을 그리고 마지막으로 제4복호기(400)로부터는 제4텍셀값을 선택적으로 출력하도록 구성된다. 이 경우에, 네 개의 텍셀들(30 내지 36) 각각은 각 텍셀블록(10 내지 16) 내에 위치한다.
- [0040] 도 2는 한 실시예에 따른 복호 시스템(10)을 가지는 실현 구조를 개략적으로 도시하고 있다. 구조는 암호화된 텍셀블록(40)을 포함하는 메모리 위치들을 가지는 메모리(5), 바람직하게 RAM을 포함한다. 그래픽 처리장치(GPU)(4)가 메모리 버스를 통해 메모리(5)에 연결된다. GPU(4)는 텍셀블록을 식별하기 위한 텍셀블록 식별기(2)를 포함하고, 텍셀블록의 암호화된 블록(40)은 메모리(5)로부터 페치되어야 한다. 텍셀블록 식별기(2)는 기본적으로 메모리(5) 내 암호화된 텍셀블록(40)의 어드레스를 계산하고 그리고 버스를 통해 메모리(5)로 메모리 액세스요청을 전송한다. 텍셀블록에 할당된 또는 지정된 메모리위치 내에 존재하는 암호화된 텍셀블록(40)은 메모리(5)로부터 관독되고 또한 버스를 통해 복호 시스템(1) 또는 GPU(4)의 캐시(3)로 전송된다.
- [0041] 복호 시스템(1)은 암호화된 텍셀블록들 또는 복호된 텍셀값들을 저장하기 위한 캐시 위치들 또는 영역들을 가지는 적어도 하나의 캐시(3)를 포함하거나 또는 캐시에 연결된다. 다른 실시예에서, GPU(4)는 텍스처 캐시 및 색상 버퍼 캐시와 같은 다수의 캐시(3)들을 포함할 수 있다. GPU(4)는 전형적으로 디스플레이 스크린(6)에 연결되는데, 스크린 상에서 복호된 영상이 디스플레이된다.
- [0042] GPU(4)와 이의 내장 유닛(1, 2, 3)들은 예컨대 그래픽카드, 비디오카드 또는 마더보드의 하나 이상의 칩들에 제공될 수 있다. 도 2의 구현 구조는 예컨대, 컴퓨터, 게임콘솔, 이동전화와 같은 휴대용 장치 또는 미디어 처리장치를 포함하는 소정의 이미지 또는 텍스처 처리유닛 또는 장치에 구현될 수 있다.
- [0043] 다음에, 상이한 구현 예들과 관련해 복호 시스템의 실시예들을 상세히 기술하게 된다. 이들 예들은 [3, 4]DP 기술된 PLANAR에 따라 암호화된 텍셀블록들을 처리하기에 적합하도록 더 디자인된다. 간략하게, PLANAR는 색차(chrominance)들을 부드럽게 변경하면서 텍셀블록들을 처리하도록 디자인된다. PLANAR는 평면 근사(planar approximation)로 텍셀블록 내 색상 요소(적, 녹, 청)들을 근사한다. 평면을 명시하기 위해, 텍셀블록내 세 위치들에서 색상들을 명시하는 것이 충분하다. 도 5는, 텍셀블록(10)에 관해 소정의 위치들에서 세 개의 적색 요소(R_0 , R_H 및 R_V)들로 이 개념을 도시한다. 상기 세 개의 적색 요소들의 이 위치 지정으로, 다음의 식을 사용하여 텍셀블록(10) 내 어디에도 적색 요소를 계산할 수 있다:

[0044]
$$R(x, y) = \text{round} \left(R_0 + \frac{1}{4} (R_H - R_0)x + \frac{1}{4} (R_V - R_0)y \right)$$

[0045] 청색 및 녹색 요소들도 적색 요소들과 동일한 방식으로 계산한다. 따라서, 도 4에 도시된 것과 같은 암호화된 텍셀블록(40)은 색상 요소당 세 개의 색상 코드워드(41 내지 49), 즉 X_0 , X_H 및 X_V 를 포함하는데, X는 적색

요소에 대한 R, 녹색 요소에 대한 G 및 청색 요소에 대한 B 이다. 바람직한 구현에서, 적색 코드워드(41 내지 43)와 청색 코드워드(47 내지 49)들 각각에는 6비트들이 사용되는 것이 바람직하고 또한 녹색 코드워드(44 내지 46)들에 각각에는 7비트들이 사용되는 것이 바람직한데, 이는 적색과 청색 요소들과 비교하면 인간의 눈은 녹색 요소에 더 민감하기 때문이다. 암호화된 텍셀블록의 전체 크기는 57비트인데, 이는 ETC2의 57-비트 모드 [3]에 정확하게 맞는다. 만일 PLANAR가 단독으로 사용되거나 또는 ETC에 대해 특별한 모드로 사용되지 않는다면, 암호화된 텍셀블록(40)은, 8비트인 G_0 와 같은 색상 코드워드(41 내지 47)을 제외한 모두에 대해 7비트를 소비한다. 그러면, 전체 크기는 64비트가 된다.

[0046] PLANAR 구현에서, 도 3의 제1복호기(100)는 텍셀블록 내측에 네 개의 텍셀들 모두를 가지는 소정의 2 x 2 이중선형 패치(도 1a 참조)를 복호할 수 있다. 제1복호기(200)는 덜 진행되고 그리고 텍셀블록의 제1열(row) 또는 제1열(column) 내 텍셀들에 대한 텍셀값들을 생성한다(도 1b의 텍셀(32, 36) 및 도 1c의 텍셀(34, 236) 참조). 따라서, 제2복호기(200)는 한 번에 두 개의 텍셀값들을 출력하는데, 2 x 1패치 또는 1 x 2패치를 출력한다. 제3복호기(300)는 암호화된 텍셀블록을 복호하여 단일의 텍셀값, 바람직하게 좌표에서 상부 우측의 것(3, 0)을 생성한다(도 1d에서 텍셀(34) 참조). 제4복호기(400)는 또한 근원(origin) 텍셀(0, 0)의 단일 텍셀값을 출력한다(도 1d에서 텍셀(36) 참조).

[0047] 이는, 만일 이중선형 패치(20)의 모든 텍셀(30 내지 36)들이 도 1a에 도시한 바와 같이 텍셀블록(10)의 경계 내측에 있다면, 제1복호기(100)로부터의 출력은 값 선택기(500)에 의해 선택된다는 것을 의미한다, 만일 도 1b와 1c에 도시한 바와 같이, 두 개의 텍셀(30, 34(30, 32))들이 하나의 텍셀블록(10) 내에 있고 또한 두 개의 텍셀(32, 36(34, 36))들이 이웃하는 텍셀블록(12(14)) 내에 있다면, 좌측(30, 34) 또는 상측(30, 32) 두 텍셀들의 텍셀값들은 제1복호기(100)로부터 구하고 또한 두 개의 우측(32, 36) 또는 하측(34, 36) 텍셀들로부터 텍셀값들은 제2복호기(200)로부터 나온다. 마지막으로, 도 1d에 도시한 바와 같이, 만일 모든 네 개의 텍셀(30 내지 36)들이 상이한 텍셀블록(10 내지 16)들 내에 있다면, 값 선택기(500)는 제1복호기(100)로부터 상부 좌측 텍셀(30)에 대한 텍셀값을 선택하고, 제2복호기(200)로부터 상부 우측 텍셀(32)에 대한 텍셀값을 선택하고, 제3복호기(300)로부터 하부 우측 텍셀(34)에 대한 텍셀값을 선택하고 그리고 제4복호기(400)로부터 하부 좌측 텍셀(36)에 대한 텍셀값을 선택한다. 이러한 방식에서, 이중선형 패치(20)의 모든 네 개의 텍셀(30 내지 36)들을 단일 클록주기로 얻을 수 있는데, 제2 내지 제4복호기(200, 300, 400)들은 하드웨어를 거의 사용하지 않고 구현될 수 있다는 장점이 있다.

[0048] 다음의 식,
$$X(x,y) = \text{round}\left(X_0 + \frac{1}{4}(X_H - X_0)x + \frac{1}{4}(X_V - X_0)y\right)$$
 으로부터, 원점(0, 0) 내 텍셀은 특히 다음과 같이

매우 간단하게 계산할 수 있다,
$$X(0,0) = \text{round}\left(X_0 + \frac{1}{4}(X_H - X_0)0 + \frac{1}{4}(X_V - X_0)0\right) = X_0$$
 . 상기에서 논의 하였듯이,

제4복호기는 바람직하게 근원에서의 텍셀들만을 복호하도록 구성된다. 이는, 제4복호기의 회로는 극히 단순하게 만들 수 있다는 것을 의미한다. 도 6은 색상 요소들 중 하나에 대한 제4복호기(400)의 회로를 도시하고 있다. 도면으로부터 알 수 있듯이, 유일한 구조는 선택적이지만, 바람직하게는 입력 제1색상 코드워드(X_0)를 규정된 비트길이, 바람직하게 8비트로 확장시키도록 구성되는 확장기(extender)(48)이다. 도 10과 관련해 아래에서 더 기술하는 바와 같이, 이러한 확장기(48)는 소정의 하드웨어 구조들을 사용하는 일이 없이, 각 색상 요소들이 저장되는 곳에서부터 제4복호기(400)의 출력까지의 배선(wire)들만으로 구현될 수 있다.

[0049] 따라서, 제4복호기(400)는 어떠한 게이트(gate)들 없이 구현할 수 있다.

[0050] 상기의 논의에 따라, 제3복호기(300)는 위치(3, 0)에서의 텍셀들을 처리하여야 한다:

$$X(3,0) = \text{round}\left(X_0 + \frac{1}{4}(X_H - X_0) \times 3 + \frac{1}{4}(X_V - X_0) \times 0\right)$$
 . 마지막 항은 기여하지 않아 생략할 수 있다. 라운드 () 연산자를 제거하기 위하여, 결과는 (1/4)의 유닛들 내에서 계산되고, 2를 더하고 그리고 우측 두 항들을 이동시켜 $\gg 2$ 로 표시한다:
$$\gg 2: X(3,0) = ((X_0 \ll 2) + (X_H - X_0) \times 3 + 2) \gg 2$$
 .

[0051] 3으로 승산(multiplication)은 이동된 버전 자체에 수를 더함으로써 교환할 수 있다:

$$3t = 2t + t = (t \ll 1) + t \quad \text{with} \quad t = X_H - X_0$$
 . 따라서, 식은 다음과 같이 쓸 수 있다:
$$X(3,0) = ((X_0 \ll 2) + (t \ll 1) + t + 2) \gg 2$$
 .

[0052] 각 가산기(adder)에서, 0 대신에 캐리 입력(carry in)=1로 설정함으로써 1을 부가할 가능성이 있다.

그러므로, 2의 부가(addition)는 이 방식에서 두 개의 가산기들을 변경시킴으로써 대체될 수 있다. 아래 식에서, +!는 1로 설정하기 위해 캐리 입력을 가지는 가산기를 나타내는데 사용된다:

$X(3,0) = ((X_0 \ll 2) + !(t \ll 1) + !t) \gg 2$. 다르게 나타내면 다음과 같다:

[0053]
$$X(3,0) = \text{round}\left(X_0 + \frac{1}{4}(X_H - X_0) \times 3 + \frac{1}{4}(X_V - X_0) \times 0\right) = \text{round}\left(\frac{1}{4}X_0 + \frac{3}{4}X_H\right)$$
. 상기와 비슷하게, 식은 다음과 같이 적을 수 있다:

$X(3,0) = (X_0 + !((X_H \ll 1) + !X_H)) \gg 2$. 이 식은 상기에서 나타낸 것과 비교하면 하드웨어에서 보다 쉽게 구현된다.

[0054] 이동 연산자(shift operator)들은 배선들을 상이하게 연결함으로써 구현될 수 있다. $X(3,0)$ 을 계산하기 위해 필요한 유일한 잔여 연산은 3의 부가이다.

[0055] 도 7은 제3복호기(300)의 회로의 실시예를 도시하고 있다. 제3복호기(300)는 좌측-이동된 비트열(left-shifted bit sequence)를 얻기 위하여 제2색상 코드워드(X_H)를 좌측으로 1비트위치 이동시키기 위한 좌측 이동 연산자(310)를 포함한다. 좌측 이동 연산자(310)는 1에 설정된 캐리 입력을 가지는 제1가산기(320)에 연결된다. 이 제1가산기(320)는 좌측-이동된 비트열을 제2색상 코드워드(X_H)에 더한다. 최종 출력은 1에 설정된 캐리 입력을 가지고 또한 제1가산기(320)에 연결되는 제2가산기(322)에서 제1색상 코드워드(X_0)에 가산된다. 마지막으로, 제2가산기(322)는, 제2가산기(322)의 출력에 대응하는 비트열을 우측으로 2비트 이동시켜 제3복호기(300)의 최종 출력, 즉 텍셀값 $X(3,0)$ 을 형성하도록 구성되는 우측 이동 연산자(330)에 연결된다.

[0056] 선택적이지만 바람직한 실시예에서, 제1색상 코드워드(X_0)와 제2색상 코드워드(X_H)들은 바람직하게 6 또는 7비트에서 8비트로 먼저 확장되거나 또는 확장된다. 따라서, 제1확장기(380)는 바람직하게 제2가산기(322)에 연결되도록 배열되고 또한 제1색상 코드워드(X_0)를 선-규정된 비트크기, 바람직하게 8비트로 확장시키도록 구성된다. 제2확장기(322)는 바람직하게 좌측 이동 연산자(310)에 연결되도록 배열되고 또한 제2색상 코드워드(X_H)를 선-규정된 비트크기, 바람직하게 8비트로 신장 또는 확장하도록 구성된다.

[0057] 확장기는, a/-비트열($k < l$) 중 최상위 k비트들을 하위 k비트들로 복제하여 $1+k$ -비트열을 얻도록 단순히 배선함으로써 쉽게 구현할 수 있다. 도 10은 녹색 요소에 적합한 제1 및 제2확장기(380, 382)의 구현을 개략적으로 도시하고 있다. 도면에서 알 수 있듯이, 입력 7-비트열 중 최상위 비트(most significant bits)는 하위비트위치(least significant bits position)로 복제되어 8-비트열을 얻는다. 적색과 청색 요소들에 대해서, 최상위 두 비트들이 하위비트 위치들로 복제되어 6-비트열에서 8-비트열로 된다.

[0058] 암호화된 텍셀블록에 대해 앞서 언급한 비트크기로, 즉 적색 및 청색 요소들에 대해 색상 코드워드당 6비트와 청색 요소에 대한 색상 코드워드당 7비트와 그리고 두 개의 확장기의 사용으로, 제3복호기(300)의 하드웨어 회로는, 만일 각 가산기가 비트당 4.4 게이트들을 사용한다면 전체 250.8 게이트들로 구현될 수 있다.

[0059] R/G/B: 1 9-비트 가산기 → $9 \times 4.4 = 39.6$ 게이트

[0060] 1 10-비트 가산기 → $10 \times 4.4 = 44$ 게이트

[0061] 전체: $3 \times 83.6 = 250.8$ 게이트

[0062] 제1 및 제2복호기들의 바람직한 실시예에서와 같이 제3복호기(300)에는 클램퍼(clamper)들이 필요치 않고, 이는 여기에서 더 논의한다.

[0063] 도 8은 제2복호기(200)의 하드웨어 구현의 실시예를 도시하고 있다. 제2복호기(200)는 제2색상 코드워드(X_H)와 제3색상 코드워드(X_V) 중 하나를 출력하도록 구성되는 제1멀티플렉서(multiplexor)(250)를 포함한다. 색상 코드워드의 선택은, 네 개의 텍셀들 중 반이 텍셀블록의 경계 내측에서 제1행 또는 제1열에 위치하는지에 의존하고, 암호화된 텍셀블록은 제2복호기(200)에 입력된다. 만일 두 개의 텍셀들이 제1열에 위치한다면, 제2색상 코드워드(X_H)는 제1멀티플렉서(250)에서부터 출력되고 그렇지 않다면 제3색상 코드워드(X_V)가 출력된다. 부정 연산자(negation operator)(240)는 제1색상 코드워드(X_0)를 부정하도록 구성된다. 제1가산기(220)는 제1멀티플렉서(250)와 부정 연산자(240)에 연결되고 또한 이들 유닛(240, 250)들로부터의 출력을 가산하여 차이 $X_H - X_0$ 또는 $X_V - X_0$ 를 형성한다. 대안 구현에서, 부정 연산자는 NOT 연산자로 대체되고, 제1가산기(220)는 1에 설정된

캐리 입력을 가지는 가산기이다. NOT 연산자와 함께 1에 설정된 캐리 입력을 가지는 가산기의 사용은 일반적
으로, 하드웨어 복잡도의 면에서 정규 가산기를 가지는 부정 연산자보다 위에 있는 것이 바람직하다.

- [0064] 좌측 이동 연산자(210)는 제1가산기(220)에 연결되고 또한 가산기 출력에 대응하는 비트열을 좌측으로 1비트
위치 이동시키도록 구성된다.
- [0065] 제2멀티플렉서(252)는 텍셀의 행좌표(row coordinate)와 열좌표(column coordinate) 중 하나를 출력하도록 구
성된다. 이 텍셀은 텍셀블록 내에 존재하는 두 개의 텍셀들 중 첫 번째 텍셀이고, 이의 암호화된 텍셀블록은
제2복호기(200)에 입력된다. 만일 두 개의 텍셀들이 제1열에 위치한다면, 열좌표(S_x)가 제2멀티플렉서(252)로
부터 출력되고, 그렇지 않다면 열좌표(S_y)가 출력된다.
- [0066] 제3멀티플렉서(254)는 좌측 이동 연산자(210)와 제1가산기(220)에 연결된다. 제3멀티플렉서(254)는 $2x(X_{H/V}-$
 $X_0)=2d$, 즉 즉 좌측 이동 연산자로부터의 출력과 $(X_{H/V}-X_0)=d$, 즉 제1가산기(220)로부터의 출력 사이를 선택한
다. 이 선택은 제2멀티플렉서(252)로부터 출력된 열 또는 열좌표 중 최상위 비트의 값을 기반으로 한다.
- [0067] OR 연산자(360)는 제2멀티플렉서(352)에 연결되고 또한 제2멀티플렉서(352)로부터의 열 또는 행좌표 중에서
하위비트와 최상위 비트에 OR 연산을 수행하도록 구성된다. OR 연산자(360)와 제3멀티플렉서(254)는 AND 연
산자(270)에 더 연결된다. 이 AND 연산자(270)는 OR 연산자(260)로부터 단일 비트로 멀티플렉서 출력 비트에
관한 AND 연산을 수행한다. 여기서 결과는, 만일 OR 출력이 1_{bin} 이면 멀티플렉서 출력이 보존되고 또한 OR 출
력이 0_{bin} 이면 AND 연산자(270)로부터의 출력은 0이라는 것이다. 그러므로, AND 연산자(270)로부터 출력은
 $2d$, d 또는 0이다.
- [0068] 제2가산기(222)는 AND 연산자(270)에 연결되고 또한 AND 연산자(270)의 출력과 제1색상 코드워드(X_0)에 뒤이
은 하위 2비트로서 10_{bin} 을 가지는 비트열을 가산하도록 구성된다. 이는 $f=4xX_0+2$ 에 대응하고, 이는 제1색상
코드워드(X_0)의 비트열을 좌측으로 2비트 위치 이동시키고 또한 그렇지 않으면 이동 후 0이 되는, 마지막 2비
트로서 $2=10_{bin}$ 을 삽입함으로써 구현될 수 있다. 따라서, 제2가산기(222)로부터의 출력은 $2d+f$, $d+f$ 또는 f 이
다.
- [0069] 제3가산기(224)는 제1가산기(220)와 제2가산기(222)에 연결되고 또한 이들 가산기(220, 222)로부터 출력들을
가산하도록 구성된다. 그러므로, 이 제3가산기(224)는 결과 $3d+f$, $2d+f$ 또는 $d+f$ 를 제공한다. 제1우측 이동
연산자(230)가 제3가산기(224)에 연결되고 또한 제3가산기(224)로부터의 비트열을 우측으로 2비트 위치 이동
시켜 이중선형 패치에서 제1행 또는 제1열 내 두 텍셀들 중 하측 또는 우측 텍셀의 텍셀값을 얻도록
구성된다. 제2우측 이동 연산자(232)는 제2가산기(222)에 연결되고 또한 가산기 출력을 2비트 위치 이동시켜
제1행 또는 제1열에서 두 텍셀드 중 상측 또는 좌측 텍셀의 텍셀값을 형성한다.
- [0070] 입력 색상 코드워드(X_0 , X_H , X_V)들은 바람직하게 도 7의 제3복호기와 유사하게 6 또는 7비트에서 8비트로 확장
된다. 따라서, 선택적이지만 바람직한 확장기(280, 282, 284)들은 바람직하게 입력 단자들에 배열되어, 하위
2비트로서 10_{bin} 의 가산에 앞서 제1색상 코드워드(X_0)를 확장시켜, 이 경우에 10-비트열을 형성하도록 구성되
는 제1확장기(280)를 가진다. 제1확장기(280)는 또한 부정 연산자(240)에 연결된다. 제2확장기(282) 및 제3확
장기(284)들은 제1멀티플렉서(250)에 연결되고 또한 제2색상 코드워드(X_H)와 제3색상 코드워드(X_V)를 각각 확
장한다.
- [0071] 선택적인 구현에서, 우측 이동 연산자(230, 232)들은 바람직하게 각각의 클램퍼 또는 클램프 연산자(290,
292)에 연결된다. 이들 클램퍼(290, 292)들은 우측 이동 연산자(230, 232)들로부터의 출력을 최소 텍셀값과
최대 텍셀값들 사이에서 고정시키도록 구성된다. 따라서, 클램퍼(290, 292)들은, 제2복호기로부터의 출력 비
트열이 허용된 값 간격 내에 있도록 보장한다. 이러한 허용된 값 간격의 예는, 만일 제2복호기(200)로부터의
출력열들이 8비트이어야 한다면 0 내지 255이다.
- [0072] 도 11은 도 8의 제2복호기 또는 도 9의 제1복호기에 채용되는 것과 같은, 클램퍼(190 내지 196, 290, 292)의
하드웨어 구현예를 도시하고 있다. 이 클램퍼 실시예는 [-1024, 1023]의 간격에서 값을 잠재적으로 나타내는
11-비트열을 [0, 255]의 간격에서 값을 나타내는 8-비트열로 고정하도록 디자인된다. 클램퍼(190 내지 196,
290, 292)는, 각각 입력 비트열의 덜 8개의 덜 중요한 비트들 중 하나를 수신하는 8개의 OR 연산자(701 내지
708)들을 포함한다. OR 연산자(701 내지 708)에 대한 다른 각 입력들은 다른 OR 연산자(700)의 출력에 연결된

다. 이 OR 연산자(700)는 입력 비트열 중 2개의 최상위 비트를 수신한다. 8개의 OR 연산자(701 내지 708)들의 출력들은 각각 AND 연산자(711 내지 718)들 각각에 연결된다. AND 연산자(711 내지 718)들의 다른 입력들은 입력 비트열 중 부호 비트(sign bit)를 수신한다.

[0073] 클램퍼(190 내지 196, 290, 292)들에 대한 대안 구현 실시예가 도 12에 도시되어 있다. 이 실시예에서, 8개의 OR 연산자들은 8개의 NOR 연산자(721 내지 724)로 대체된다. 따라서, 도 11의 8개의 AND 연산자들 또한 8개의 NOR 연산자(731 내지 738)들로 대체된다. 이외에도, NOT 연산자(740)가 8개의 NOR 연산자(731 내지 738)와 입력 비트열의 부호 비트에 대한 입력 사이에 연결된다.

[0074] 도 12에 도시된 하드웨어 실시예를 구현하기 위한 희생(cost)은 다음과 같다:

[0075] 16 NOR → 16x1=16 게이트

[0076] 1 OR → 1x1.5=1.5 게이트

[0077] 1 NOT → 1x0.5=0.5 게이트

[0078] 전체: 16+1.5+0.5=18 게이트

[0079] 클램퍼를 제외하고, 도 8의 제2복호기(200)의 구현예에 대한 게이트들의 숫자는 다음과 같다:

[0080] R/G/B: 1 8-비트 가산기 → 8x4.4=35.2 게이트

[0081] 1 11-비트 가산기 → 11x4.4=48.4 게이트

[0082] 1 12-비트 가산기 → 12x4.4=52.8 게이트

[0083] 1 8-비트 믹스(mux) → 8x2.2=17.6 게이트

[0084] 1 10-비트 믹스 → 10x2.2=22 게이트

[0085] 1 2-비트 믹스 → 2x2.2=4.4 게이트

[0086] 1 1-비트 OR → 1x1.5=1.5 게이트

[0087] 1 10-비트 AND → 10x1.5=15 게이트

[0088] 1 8-비트 NOT → 8x0.5=4 게이트

[0089] 전체: 3x200.9=602.7 게이트

[0090] 마지막으로, 도 9는 제1복호기(100)의 하드웨어 구현의 예이다. 제1복호기(100)는 제1색상 코드워드(X_0)를 부정하도록 구성되는 부정 연산자(140)를 포함한다. 제1가산기(120)는 부정 연산자(140)에 연결되고 또한 부정된 제1색상 코드워드를 제2색상 코드워드(X_H)에 가산하여 $g=X_H-X_0$ 를 얻는다. 제2가산기(120)는 또한 부정 연산자(140)에 연결되고 또한 부정된 제1색상 코드워드를 제3색상 코드워드(X_V)에 가산하여 $h=X_V-X_0$ 를 얻는다. 대안 구현에서, 부정 연산자(140)는 NOT 연산자로 대체된다. 그런 다음 제1가산기(120)와 제2가산기(122) 둘 다는 1에 설정된 캐리 입력을 가진다.

[0091] 제1좌측 이동 연산자(110)는 제1가산기(120)에 연결되고 또한 가산기 출력을 좌측으로 1비트 위치 이동시켜 $2g$ 에 대응하는 좌측-이동된 비트열을 얻는다. 제2좌측 이동 연산자(112)는 제2가산기(122)에 연결되고 또한 가산기 출력을 좌측으로 1비트 위치 이동시켜 $2h$ 의 비트열을 얻는다. 제1멀티플렉서(150)는 제1좌측 이동 연산자(110)와 제1가산기(120)에 연결되고 또한 열좌표(S_X)의 최상위 비트를 기반으로 g 와 $2g$ 사이를 선택하도록 구성된다. 따라서, 제2멀티플렉서(152)는 제2좌측 이동 연산자(112)와 제2가산기(122)에 연결되고 또한 행좌표(S_Y)의 최상위 비트를 기반으로 h 와 $2h$ 사이를 선택한다. 열과 행좌표들은 $2x2$ 이중선형 패치에서 상부 좌측 텍셀, 즉 도 1a 내지 1d에서 텍셀(30)의 좌표이다.

- [0092] 제1 OR 연산자(160)는 열좌표(S_x)의 최상위 비트와 하위비트에 대해 OR 연산을 수행한다. 제2 OR 연산자(162)는 행좌표(S_y)의 최상위 비트와 하위비트에 대해 OR 연산을 수행한다. 제1 OR 연산자(160)와 제1멀티플렉서(150)들은, 제1 OR 연산자(160)로부터의 단일 비트로 멀티플렉서 출력에 대해 비트에 관한 AND 연산을 수행하는 제1 AND 연산자(170)에 연결된다. 따라서, 제1 AND 연산자(170)로부터의 출력은 $2g$, g 또는 0 이다. 제2 AND 연산자(172)는 제2멀티플렉서(152)와 제2 OR 연산자(162)에 연결된다. 제2 AND 연산자(172)는 제2 OR 연산자(162)로부터의 비트로 멀티플렉서 입력에 대해 비트에 관한 AND 연산을 수행하여 $2h$, h 또는 0 의 출력을 얻는다.
- [0093] 제3가산기(124)는 제1 AND 연산자(124)에 연결되고 또한 제1색상 코드워드에 후행하는 두 개의 덜 중요한 비트들로서 10_{bin} 을 가지는 비트열에 가산기 출력을 가산한다. 이는, 앞서 설명하였듯이, $f=4X_0+2$ 에 대응한다. 따라서, 가산기 출력은 $2g+f$, $g+f$ 또는 f 이다. 제4가산기(126)는 제1가산기(120)와 제3가산기(124)에 연결되고 그리고 이들 가산기(120, 124)들로부터의 출력들을 가산하여 $3g+f$, $2g+f$ 또는 $g+f$ 를 얻도록 구성된다. 제5가산기(128)는 제2가산기(122)와 제2 AND 연산자(172)에 연결되고 그리고 이들 두 유닛(122, 172)로부터의 출력들을 가산하여 $3h$, $2h$ 또는 h 를 얻는다.
- [0094] 제6가산기(121)는 제3가산기(124)와 제2 AND 연산자(172)에 연결되고 그리고 이들 유닛(124, 172)들로부터의 출력들을 가산하도록 구성된다. 제1 우측 이동 연산자(130)는 제6가산기(121)에 연결되고 그리고 가산기 출력의 비트열을 우측으로 2비트 위치 이동시켜 도 1a의 텍셀(30)의 텍셀값을 얻도록 구성된다.
- [0095] 제7가산기(123)는 제4가산기(126)와 제2 AND 연산자(172)에 연결되고 그리고 이들 유닛(126, 172)로부터의 출력들을 가산하도록 구성된다. 제2 우측 이동 연산자(132)는 제7가산기(123)에 연결되고 그리고 가산기 출력의 비트열을 우측으로 2비트 위치 이동시켜 도 1a의 텍셀(32)의 텍셀값을 얻도록 구성된다.
- [0096] 제8가산기(125)는 제3가산기(124)와 제5가산기(128)에 연결되고 그리고 이들 가산기(124, 28)들로부터의 출력들을 가산하도록 구성된다. 제3 우측 이동 연산자(134)는 제8가산기(125)에 연결되고 그리고 가산기 출력을 우측으로 2비트 위치 이동시켜 도 1a의 텍셀(34)의 텍셀값을 구하도록 구성된다.
- [0097] 제9가산기(127)는 제4가산기(126)와 제5가산기(128)에 연결되고 그리고 이들 가산기(126, 128)들로부터의 출력들을 가산하도록 구성된다. 제4 우측 이동 연산자(136)는 제4가산기(127)에 연결되고 그리고 가산기 출력의 비트열을 우측으로 2비트 위치 이동시켜 도 1a의 텍셀(36)의 텍셀값을 얻도록 구성된다.
- [0098] 선택적이지만 바람직한 구현에서, 입력 색상 코드워드(X_0 , X_H , X_V)들은 먼저 확장되거나 6 또는 7비트에서 8비트로 확장된다. 따라서, 제1비트 확장기(180)는 바람직하게 부정 연산자(140)에 연결되도록 배열되고 또한 덜 중요한 비트로서 10_{bin} 이 가산되기 전에 제1색상 코드워드(X_0)를 선-규정된 비트열, 바람직하게 8비트로 신장 또는 확장하여 10-비트열을 얻도록 구성된다. 따라서, 제2확장기(182)는 제1가산기(120)에 연결되도록 배열되고 또한 제2색상 코드워드(X_H)를 확장하도록 구성된다. 마지막으로, 제3확장기(184)는 제2가산기(122)에 연결되고 또한 제3색상 코드워드(X_V)를 확장하도록 구성된다.
- [0099] 선택적 구현에서, 우측 이동 연산자(130 내지 136)들 각각은 클램퍼 또는 클램프 연산자(190 내지 196)에 연결된다. 이들 클램퍼(190 내지 196)들은 우측 이동 연산자(130 내지 136)로부터의 출력을 최소 텍셀값과 최대 텍셀값 사이에서 고정시키도록 구성된다. 그러므로, 클램퍼(190 내지 196)들은, 제1복호기(100)로부터의 출력 비트열이 허용된 값 간격 내에 있도록 한다. 이러한 허용된 값 간격의 예는, 만일 제1복호기(100)로부터의 출력열들이 8비트가 되어야만 한다면 0 내지 255이다.
- [0100] 선택적 클램퍼(190 내지 196)를 배제한 도 8의 구현예에 대한 게이트들의 숫자는 다음과 같다:
- [0101] R/G/B: 2 8-비트 가산기 → 16x4.4=70.4 게이트
- [0102] 1 10-비트 가산기 → 10x4.4=44 게이트
- [0103] 1 11-비트 가산기 → 11x4.4=게이트
- [0104] 5 12-비트 가산기 → 60x4.4=264 게이트
- [0105] 2 10-비트 믹스 → 20x2.2=44 게이트

- [0106] 2 1-비트 OR → 2x1.5=3 게이트
- [0107] 2 10-비트 AND → 20x1.5=30 게이트
- [0108] 1 8-비트 NOT → 8x0.5=4 게이트

- [0109] 전체: 3x507.8+1523.4 게이트

- [0110] 따라서, 복호시스템의 네 개의 복호기들은 클램퍼들을 배제하고서 단지 0+250.8+602.7+1523.4=2376.9 게이트만으로 구현될 수 있다.
- [0111] 선행기술 [3]에서 제안한 바와 같이 네 개의 동일한 복호기들을 가지는 통상적인 구현은 도 9의 제1복호기와 매우 닮아 보인다. 단지 차이점은, 도면에서와 KX이 네 개 대신에 단지 하나의 텍셀만이 출력에서 필요할 수 있다는 것이다. 그러므로, 네 개의 가산기(121, 123, 125, 127)들은 가산기에 선행하는 두 개의 4방향(four-way) 멀티플렉서들로 대체할 수 있다. 이는 색상 채널당 6개의 가산기들과, 4개의 멀티플렉서와 그리고 두 개의 AND 연산자들을 필요로 한다. 이는, 네 개의 복호기들과 세 개의 상이한 색상 요소들을 가지는 선행기술에 따른 하드웨어 희생 또는 복잡도는 다음과 같았다:
- [0112] 6x3x4=72가산기
- [0113] 4x3x4=48멀티플렉서
- [0114] 2x3x2=24AND 연산자

- [0115] 이는 다음과 같은 것을 필요로 하는 도 6 내지 8의 구현 실시예들에 필적할 수 있다:
- [0116] (9+3+2+0)x3=42가산기
- [0117] (2+3+0+0)x3=15멀티플렉서
- [0118] (2+1+0+0)x3=9AND 연산자

- [0119] 따라서, 가산기들의 숫자는 72에서 42로, 42% 줄어들었다. 멀티플렉서는 훨씬 더 48개에서 15개로, 69% 감소하였다.
- [0120] 도 13은 값 선택기(500)의 하드웨어 구현의 실시예를 도시하고 있다. 이 실시예에서, 값 선택기(500)는 네 개의 멀티플렉서(51)를 포함하고, 멀티플렉서 각각은 이중선형 패치의 텍셀(30 내지 36)의 텍셀값을 출력한다. 제1멀티플렉서(510)는 제1복호기(100)의 네 개의 출력들에 연결되고 그리고 이중선형 패치에서 제1텍셀(30)의 텍셀값으로서 네 개의 비트열들 중 하나를 선택한다. 제2멀티플렉서(520)는 제1복호기(100)의 제2 및 제4출력과 제2복호기(200)의 두 출력들에 연결된다. 제2멀티플렉서(520)는 이중선형 패치에서 제2텍셀(32)의 텍셀값을 출력한다. 제3멀티플렉서(530)는 제1복호기(100)의 제3 및 제4출력들과, 제2복호기(200)의 제1출력과 그리고 제3복호기(300)의 단일 출력에 연결된다. 제3멀티플렉서(530)는 이중선형 패치에서 제3텍셀(34)의 텍셀값을 출력한다. 마지막으로, 제4멀티플렉서(540)는 제1복호기(100)의 제4출력과, 제2복호기(200)의 제2출력과 그리고 제4복호기(400)의 단일 출력에 연결된다. 제4멀티플렉서(540)는 이중선형 패치에서 제4텍셀(36)의 텍셀값을 출력한다.
- [0121] 텍셀블록 경계에 관한 이중선형 패치의 위치는, 어느 비트열들이 멀티플렉서(510 내지 540)들로부터 출력되는지를 나타낸다. 그러므로, 도 1a에 도시된 상황에서, 제1복호기(100)의 네 출력들로부터의 비트열들은 멀티플렉서(510 내지 540)들에 의해 선택된다. 상세히 설명하면, 제1멀티플렉서(510)는 제1복호기(100)의 제1출력으로부터 비트열을 출력하고, 제2멀티플렉서(520)는 제2출력으로부터 비트열을 출력하고, 제3(530) 및 제4멀티플렉서(540)는 각각 제3 및 제4복호기 출력들로부터 비트열을 출력한다. 도 1b에 도시된 바와 같이 위치한 이중선형 패치는, 제1멀티플렉서(510)가 제1복호기(100)의 제2출력의 비트열을 선택한다는 것을 의미한다. 제2멀티플렉서(520)는 제2복호기(200)의 제1출력의 비트열을 선택한다. 제3복호기(300)는 제1복호기(100)로부터

제4출력의 비트열을 선택하고 또한 제4복호기(400)는 제2복호기(200)의 제2출력으로부터 비트열을 선택한다. 이 개념은 도 1c와 1d에 도시된 바와 같은 다른 두 패치 위치들에도 비슷한 방식으로 적용될 수 있다.

[0122] 도 13에 도시된 것과 같은 값 선택기(500)의 실시예를 구현하기 위한 게이트들의 숫자의 면에서 희생은 다음과 같다:

[0123] 3 8-비트 4-방향 믹스 → 24x6.6=158.4 게이트

[0124] 1 8-비트 3-방향 믹스 → 8x4.4=35.2 게이트

[0125] 전체: 158.4+35.2=193.6 게이트

[0126] 클램퍼를 포함한, 복호기들과 값 선택기에 대한 전체 희생은 2376.9+6x18+193.6=2678.5 게이트

[0127] 만일 도 8 및 9에 도시된 것과 같이 제1 및 제2복호기들의 클램퍼들이 생략되고 또한 대신에 각각이 값 선택기(500)의 멀티플렉서(510 내지 540)들 중 하나에 연결되는 네 개의 클램퍼들로 대체된다면, 값 선택기(500)를 구현하기 위한 희생이 증가하는데, 이는 193.6 게이트 대신에 266.2 게이트의 희생에, 세 개의 11-11xm 4-방향 멀티플렉서들과 하나의 11-비트 3-방향 멀티플렉서를 필요로 할 수 있기 때문이다.

[0128] 복호 시스템의 화소 계산기는 기술분야에서 잘 공지된 기술에 따라 하드웨어에서 구현될 수 있다. 상세히 설명하면, 이중선형 필터링의 경우에, 값 선택기로부터의 네 개의 텍셀값들이 A, B, C 및 D로 표시된다고 가정한다. 그러면, 화소 계산기는 화소값을 다음과 같이 계산한다:

[0129]
$$\text{화소값} = A(1-dx)(1-dy) + Bdx(1-dy) + C(1-dx)dy + Ddxdy$$

[0130] 여기서, dx는 x-축을 따라 텍셀(A)의 중간 지점과 화소에 대한 샘플 지점 간의 간격이고, dy는 y-축을 따라 텍셀(A)의 중간 지점과 화소에 대한 샘플 지점 간의 대응하는 간격이다.

[0131] 상기에 기술한 식은 적, 녹 및 청색 요소들에 대해 동시에 수행된다. 하드웨어 구현은 식을 기반으로 규정될 수 있다. 예컨대, 샘플 지점의 위치는 세 개의 이진화 십진수(three binary decimal)로 나타낼 수 있다. 예컨대 dx 위치는 다음의 값 [0.000, 0.125, 0.250, 0.375, 0.500, 0.625, 0.750, 0.875]를 추정할 수 있다. 이 경우에, 위치는 여덟 개로 규정될 수 있어서, 상기 값들은 [0,1,2,3,4,5,6,7]로 나타낼 수 있다. 그러면, 상기 식은 다음과 같이 다시 적을 수 있다:

[0132]
$$\text{화소값} = (A(8-DX)(8-DY) + BDX(8-DY) + C(8-DX)DY + DDXY) \gg 6$$

[0133] 여기서, DX=8 x dx 그리고 DY=8 x dy 그리고 DX, DY는 0과 7 사이의 정수들이다.

[0134] 두 번째 항 8DX(8-DY)를 조사하면, DX와 (8-DY) 둘 다는 0과 8 사이의 정수라는 것을 이해할 것이다. 8을 나타내기 위하여, 네 개의 비트들이 필요하다. 각 입력에서 네 개의 비트들을 처리하고 또한 8-비트열을 출력하는 승산 연산자를 사용할 수 있다. 그러나, 승산 연산자가 출력할 수 있는 가장 큰 숫자는 7x8=56 이어서, 단지 여섯 개의 하위비트들이 출력에서부터 필요하다.

[0135] 그런 다음, 승산 연산자로부터의 출력은 텍셀값 B, 바람직하게는 0과 255 사이의 숫자로 승산되어야만 한다. 이 승산은, 두 개의 8-비트 입력들을 취하는 승산 연산자에 의해 처리되어 16-비트 비트열을 출력한다. 그러나, 승산 연산자로부터 단지 14개의 하위 비트들만이 필요한데, 이는 가능한 가장 큰 수가 56x266<2¹⁴ 이기 때문이다.

[0136] 그런 다음, 식에서 다른 항들에 대해 대응하는 구현이 수행되고 또한 각 최종 승산 연산자들로부터의 출력들이 가산되고 그리고 최종 8-비트 화소값을 얻기 위하여 우측으로 6단계 이동된다.

[0137] 상기에서 기술한 구현 예들은, 복호 시스템들의 유닛들을 하드웨어에서 어떻게 구현할 수 있는지에 대한 바람직하지만 비-제한적인 실시예들이다. 이의 다양한 변형들이 가능하고 또한 본 발명의 범위 내에 들어간다. 중요한 특징은, 복호기들의 회로는 복잡도가 증가함에 따라 다르고 또한 각 복호기는 패치 내 규정된 위치들에서 하나 이상의 텍셀들의 텍셀값(들)을 처리하고 출력하는 것이라 생각된다.

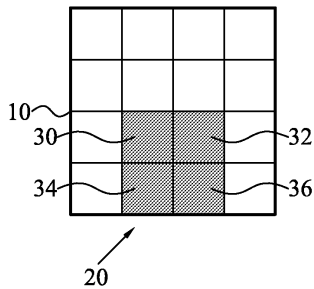
[0138] 도 14는 적어도 하나의 암호화된 텍셀블록을 복호함으로써 복호된 화소를 생성하는 방법을 흐름도로 도시하고

있다. 방법은 단계(S1)에서 시작하는데, 텍셀블록 경계에 관해 적어도 N개 텍셀값들의 위치, 즉 패치가 결정된다. 암호화된 각 텍셀블록, 즉 결정된 상대 패치 위치에 따라 동일하게 암호화된 텍셀블록 또는 상이하게 암호화된 텍셀블록이 N개 복호기들에 입력되어 복호된 텍셀값들을 얻는다. 다음 단계(S2)는 N개 복호기들 중 적어도 하나로부터 적어도 N개 텍셀값들을 선택적으로 출력한다. 단계(S2)에서 선택하고자 하는 특정 N개 텍셀값들과 이들은 어느 복호기(들)로부터 패치되는지가 단계(S1)에서 얻은 상대 패치 위치를 기반으로 결정된다. 이외에도, N개 복호기들 각각은 N-1개 다른 복호기들의 회로와는 상이한 각 회로를 가진다. 최종단계(S3)는 단계(S2)에서 선택된 적어도 N개 텍셀값들을 기반으로, 바람직하게는 이들 적어도 N개 텍셀값들의 선형 조합으로서 복호된 화소값을 계산한다. 그런 다음, 방법은 종료하거나 또는 복호된 새로운 화소를 생성할 목적으로 새롭게 다시 시작한다.

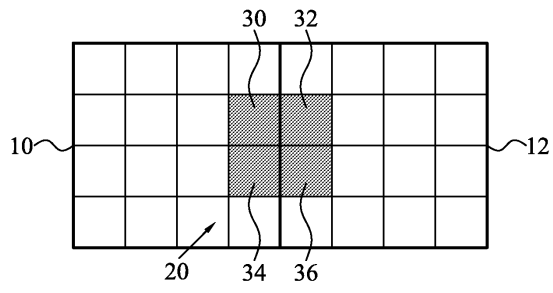
- [0139] 단계(S2)는 바람직하게, 만일 N개 텍셀들이 텍셀블록의 경계 내측에 위치한다면 N개 복호기들 중 제1복호기로부터 모든 N개 텍셀값들을 선택적으로 출력하는 것을 수반한다. 그러나, 만일 N개 텍셀들 중 반이 한 텍셀블록 내측에 위치하고 또한 텍셀들의 나머지 반이 이웃하는 텍셀블록 내측에서 발견된다면, NRO 텍셀값들 중 반은 바람직하게 제1복호기의 출력으로부터 선택되고 또한 나머지 반은 제2복호기의 출력으로부터 선택된다.
- [0140] 바람직한 실시예에서 N=4이다. 만일 각각의 텍셀이 개별적인 텍셀블록 내에서 발견되도록 네 개의 텍셀들이 위치한다면, 단계(S2)는 네 개의 복호기들 각각으로부터 각각 하나의 텍셀값을 선택하는 것을 수반한다.
- [0141] 상기에서 기술한 실시예들은 본 발명의 몇몇 실시적인 예들로서 이해하여야 한다. 본 기술분야의 당업자라면, 다양한 수정과, 조합과 변경안들이 본 발명의 범위를 이탈하는 일이 없이 실시예들에 대해 이루어질 수 있다는 것을 알게 될 것이다. 특히, 상이한 실시예들에서 상이한 부분 해결책들은, 기술적으로 가능한 다른 구성들로 결합될 수 있다. 그러나, 본 발명의 범위는 첨부한 청구항들에 의해 규정된다.
- [0142] 참조문헌
- [0143] [1] 미국특허 제5,956,431호
- [0144] [2] Strom 및 Moller, "iPACKMAN:High-Quality, Low-Complexity Texture Compression for Mobile Phones", Graphics Hardware, 2005, ACM Press, pp.63-70
- [0145] [3] Strom and Pettersson, "ETC2:Texture Compression using Invalid Combinations", Graphics Hardware, 2007, ACM Press, pp.49-54
- [0146] [4] 국제출원 제WO 2006/126949호

도면

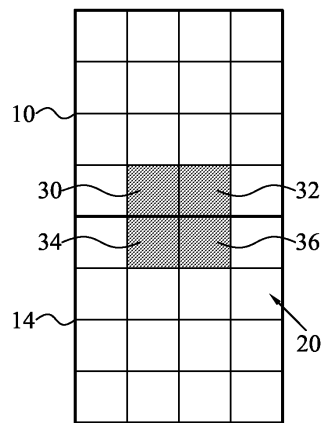
도면1a



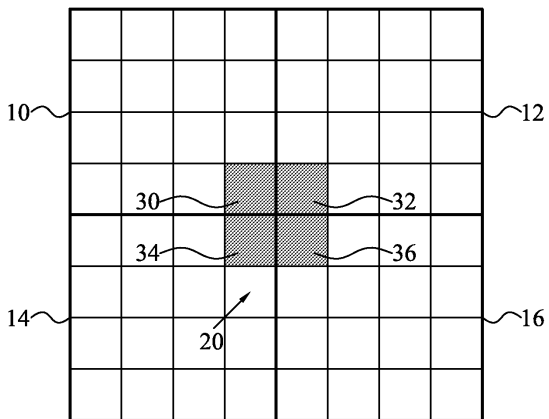
도면1b



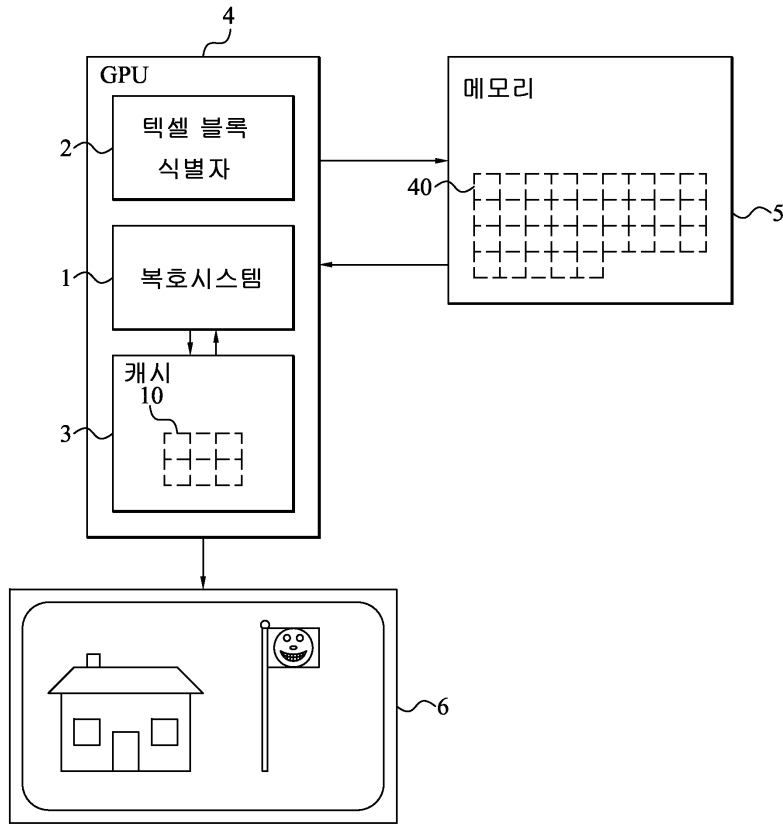
도면1c



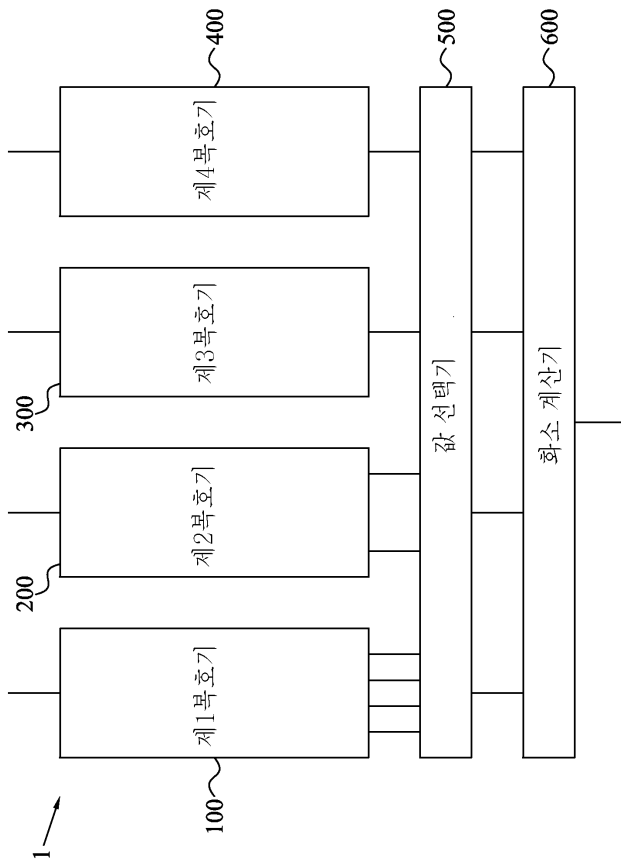
도면1d



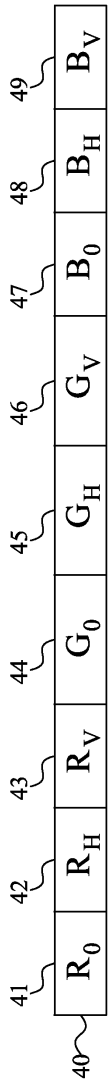
도면2



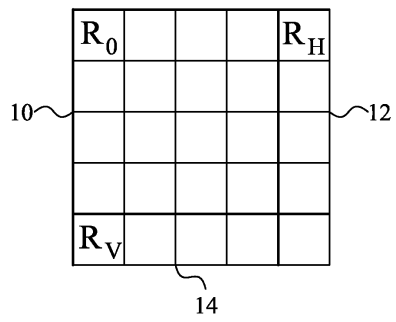
도면3



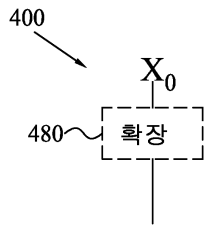
도면4



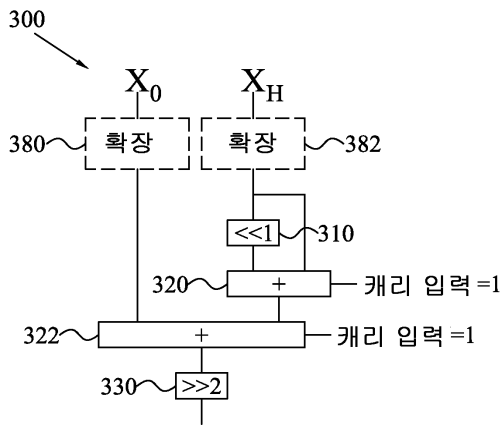
도면5



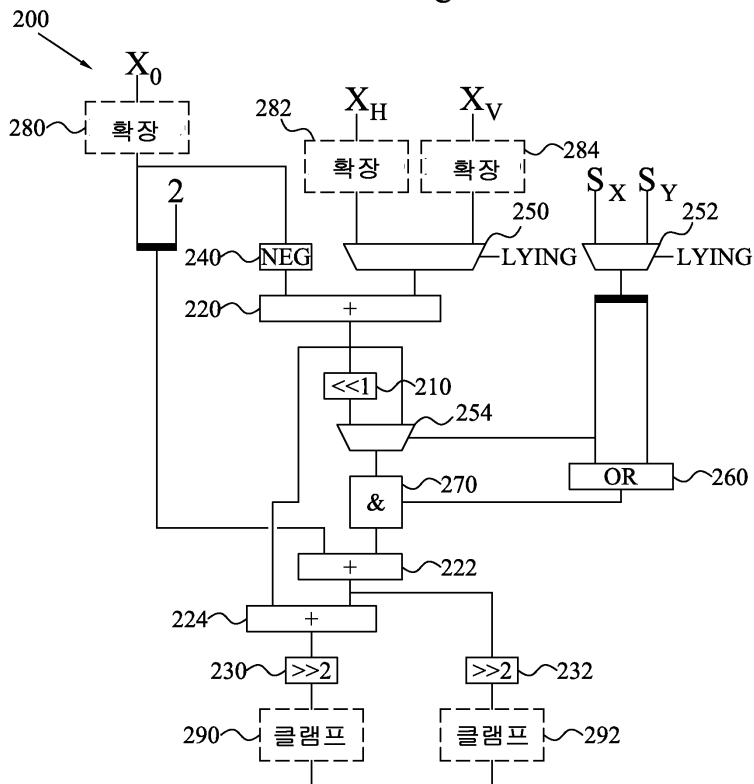
도면6



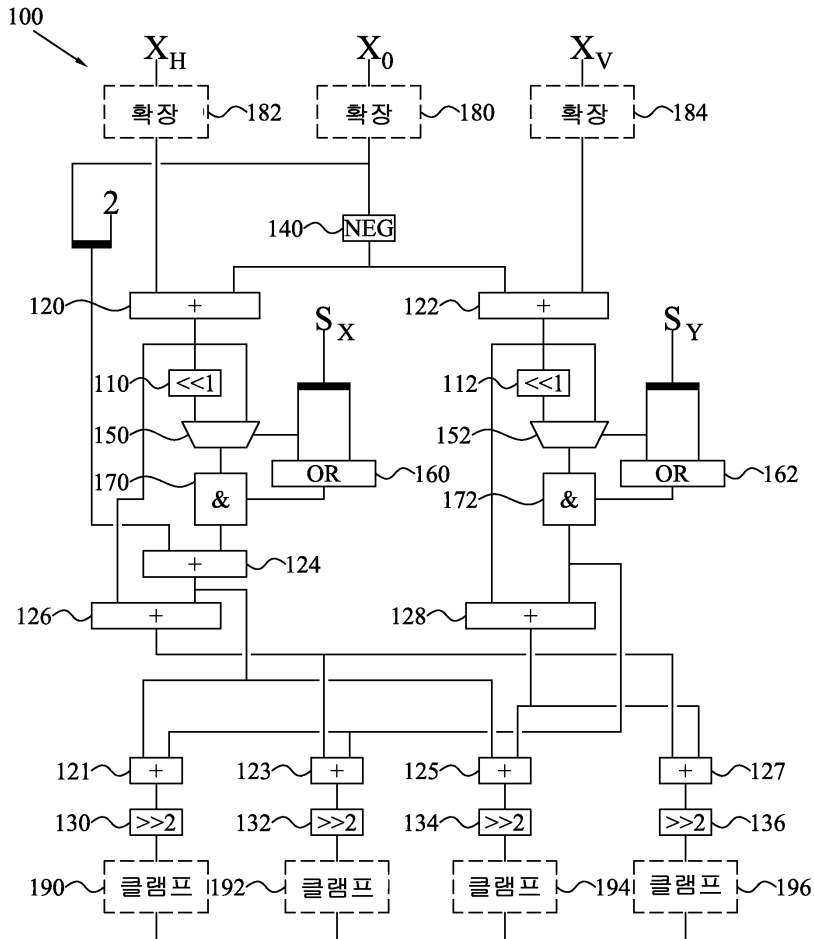
도면7



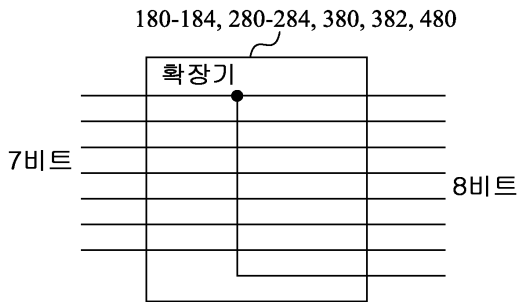
도면8



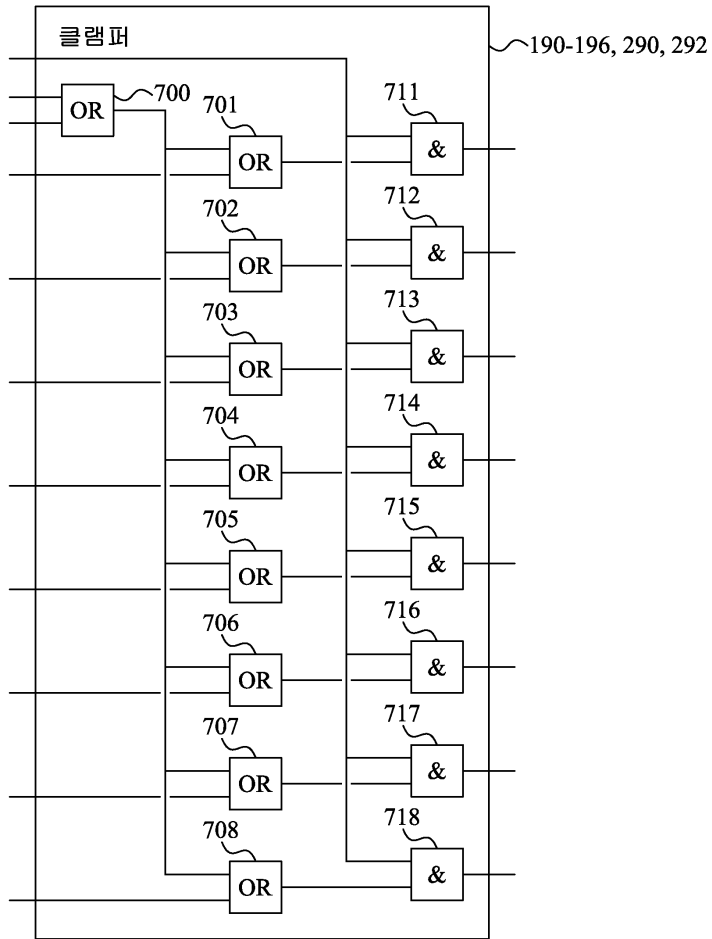
도면9



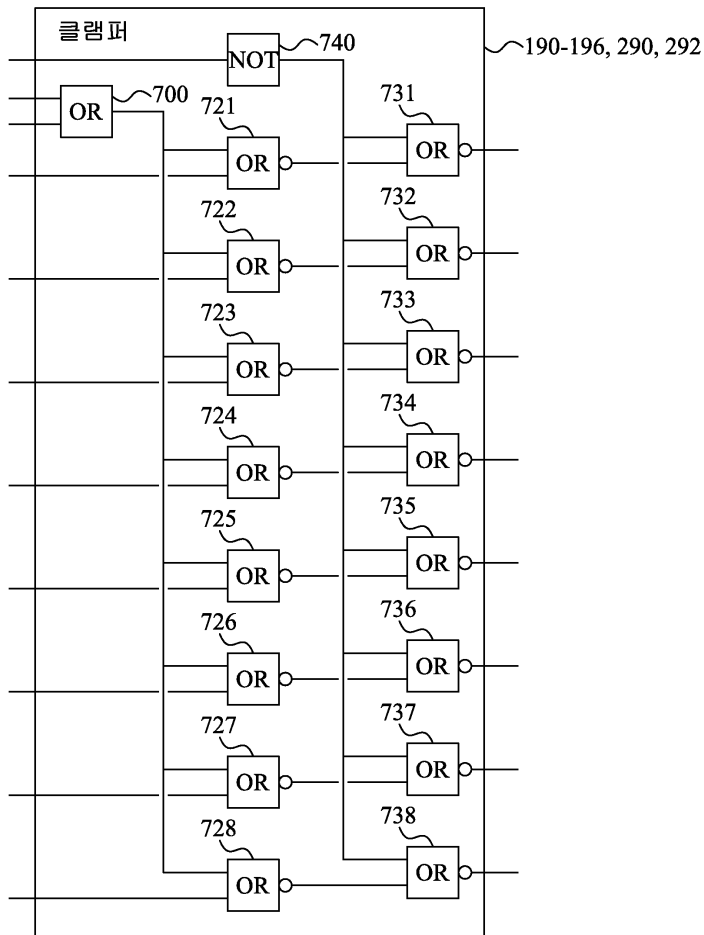
도면10



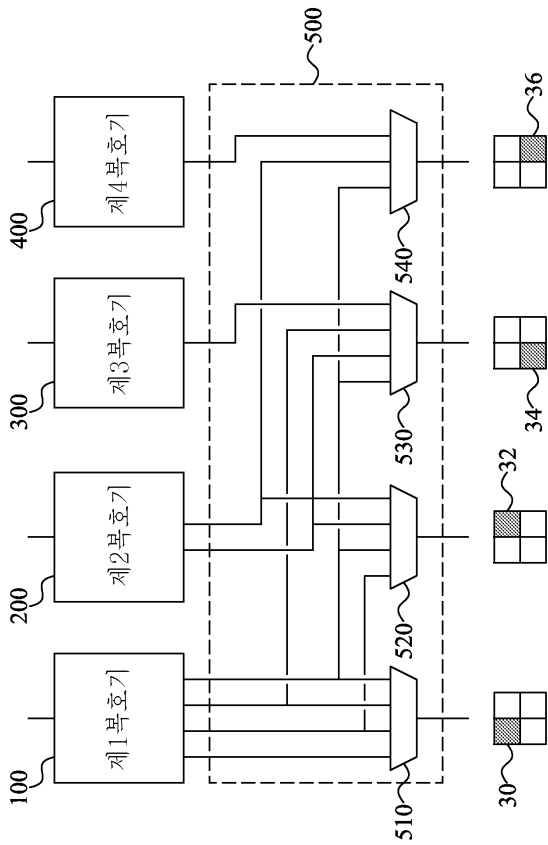
도면11



도면12



도면13



도면14

