



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년02월15일
(11) 등록번호 10-0804335
(24) 등록일자 2008년02월11일

- (51) Int. Cl.
H04N 7/32 (2006.01) *H04N 7/24* (2006.01)
H04N 7/12 (2006.01)
- (21) 출원번호 10-2005-7010219
 (22) 출원일자 2005년06월03일
 심사청구일자 2005년07월19일
 번역문제출일자 2005년06월03일
- (65) 공개번호 10-2005-0085392
 (43) 공개일자 2005년08월29일
 (86) 국제출원번호 PCT/US2003/024953
 국제출원일자 2003년08월07일
 (87) 국제공개번호 WO 2004/054257
 국제공개일자 2004년06월24일
- (30) 우선권주장
 10/313,773 2002년12월06일 미국(US)
- (56) 선행기술조사문헌
 US 5,737,023
 US 6,072,834
 US 6,108,047
 US 6,297,852

- (73) 특허권자
애플 인크.
 미합중국 95014 캘리포니아 쿠퍼티노 인피니트 루프 1
- (72) 발명자
하스켈, 바린, 쥐.
 미국, 캘리포니아 94040-3960, 마운틴 뷰, 페어브룩 드라이브1190
신저, 데이비드, 더블유.
 미국, 캘리포니아 94127-1328, 샌 프란시스코, 와워나 스트리트268
(뒷면에 계속)
- (74) 대리인
백만기, 이중희, 주성민

전체 청구항 수 : 총 71 항

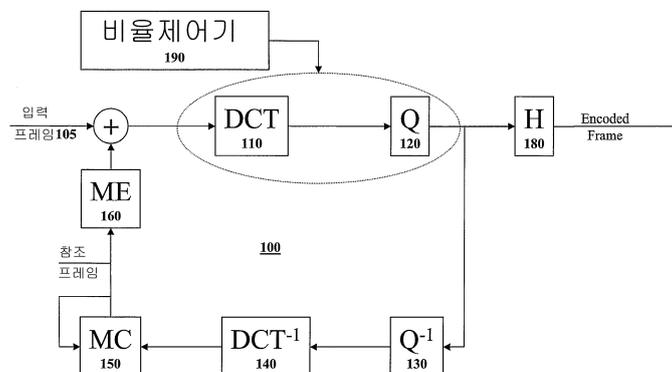
심사관 : 조우연

(54) 분할 연산의 필요성을 감소시킨 디지털 비디오 엔코딩을 위해 가변 정확도 화상간 타이밍 특정을 위한 방법 및 장치

(57) 요약

본 발명은 디지털 비디오 시스템에 있어서 모션 추정을 실행하는 방법 및 장치에 관한 것이다. 특히 효과적인 방법으로 추정 모션 벡터를 신속하게 계산하는 시스템을 제공하는 관한 것이다. 하나의 실시예로, 제1의 피승수는 제1의 비디오 화상과 제2의 비디오 화상 사이의 제1의 디스플레이 시간 차이와 2의 멱승 형태의 스케일 값과의 곱에 의해 결정된다. 이러한 과정은 비율에서 분자를 확대시킨다. 다음으로, 상기의 시스템은 스케일 분자를 제2의 비디오 화상과 제3의 비디오 화상 사이의 제2의 처음 디스플레이 시간 차이로 나눈 스케일 비를 정한다. 스케일 비는 계산용의 모션 벡터 추정으로 저장된다. 상기의 스케일 비를 저장함으로써, 스케일 비가 주요 비트를 세이브하고, 스케일 축소가 나누기 연산에 소요되는 필요 시간을 제거하는 방식의 단순 시프트로 실행되기 때문에, 모든 추정된 모션 벡터는 좋은 정밀도로 신속하게 계산될 수 있다.

대표도 - 도1



(72) 발명자

더미트라스, 애드리아나

미국, 캘리포니아 94087, 씨니배일, 아파트 2403,
더블유. 이엘카미노 리얼 250

파리, 아틀

미국, 캘리포니아 95014-0632, 쿠퍼티노, #4203,
프루네리지애비뉴 19500

특허청구의 범위

청구항 1

디지털 비디오 시스템에서 모션 추정을 실행하기 위한 방법으로서, 상기 방법은:

제1의 비디오 화상과 제2의 비디오 화상 사이의 제1의 디스플레이 시간 차이에 2의 멱승 값을 곱함으로써 제1의 피승수를 결정하는 단계; 및

상기 피승수를 상기 제2의 비디오 화상과 제3의 비디오 화상 사이의 제2의 디스플레이 시간 차이로 나눔으로써 스케일 비를 결정하는 단계 - 상기 스케일 비는 하나 이상의 모션 벡터들을 계산하기 위한 것임 - 를 포함하는, 디지털 비디오 시스템에서의 모션 추정 실행 방법

청구항 2

제1항에 있어서, 상기 방법은:

상기의 스케일 비에 상기 제1의 비디오 화상과 관련된 모션 벡터를 곱하여 스케일 모션 벡터를 결정하는 단계를 더 포함하는, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 3

제2항에 있어서, 상기 방법은:

상기 스케일 모션 벡터를 상기 2의 멱승 값으로 나누는 단계를 더 포함하는, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 4

제3항에 있어서,

상기 나누는 단계는 상기 스케일 모션 벡터의 비트들을 시프팅(shifting) 함으로써 실행되는, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 5

제3항에 있어서,

상기 2의 멱승 값은 256을 포함하는, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 6

제1항에 있어서,

상기 2의 멱승 값은 통신 채널을 통해 주기적으로 전송되는, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 7

제1항에 있어서,

상기 나눗셈에 의해 상기 스케일 비를 결정하는 단계는 비디오 화상마다 한 번씩만 실행되는, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 8

제1항에 있어서,

상기 스케일 비는 끝수가 버려지는 것인, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 9

제1항에 있어서,

상기 스케일 비는 가장 근접한 정수로 반올림 되는, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 10

제1항에 있어서,

상기 2의 먹승 값은 상수를 포함하는, 디지털 비디오 시스템에서의 모션 추정 실행 방법.

청구항 11

명령어 세트를 저장하는 컴퓨터 판독 가능한 매체로서, 상기 명령어 세트는 하나 이상의 프로세서들에 의해 실행될 때 상기 하나 이상의 프로세서들이:

제1의 비디오 화상과 제2의 비디오 화상 사이의 제1의 디스플레이 시간 차이에 2의 먹승 값을 곱함으로써 제1의 피승수를 결정하는 단계;

상기 피승수를 상기 제2의 비디오 화상과 제3의 비디오 화상 사이의 제2의 디스플레이 시간 차이로 나누어 스케일 비를 결정하는 단계 - 상기 스케일 비는 하나 이상의 모션 벡터들을 계산하기 위한 것임- 를 실행하도록 하는 컴퓨터 판독 가능한 매체.

청구항 12

제11항에 있어서,

상기 명령어 세트는 하나 이상의 프로세서들에 의해 실행될 때, 상기 하나 이상의 프로세서들이:

상기 스케일 비에 상기 제1의 비디오 화상과 관련된 모션 벡터를 곱하여 스케일 모션 벡터를 결정하는 단계를 더 실행하도록 하는 컴퓨터 판독 가능한 매체.

청구항 13

제12항에 있어서,

상기 명령어 세트는 하나 이상의 프로세서들에 의해 실행되었을 때, 상기 하나 이상의 프로세서들이:

상기 스케일 모션 벡터를 상기 2의 먹승 값으로 나누는 단계를 더 실행하도록 하는, 컴퓨터 판독 가능한 매체.

청구항 14

제13항에 있어서,

상기 나누는 단계는 상기 스케일 모션 벡터의 비트들을 시프팅(shifting) 함으로써 실행되는, 컴퓨터 판독 가능한 매체.

청구항 15

제13항에 있어서,

상기 2의 먹승 값은 256을 포함하는, 컴퓨터 판독 가능한 매체.

청구항 16

제11항에 있어서,

상기 2의 먹승 값은 통신 채널을 통해 주기적으로 전송되는, 컴퓨터 판독 가능한 매체.

청구항 17

제11항에 있어서,

상기 나눗셈에 의해 상기 스케일 비를 결정하는 단계는 비디오 화상마다 한번 씩 만 실행되는, 컴퓨터 판독 가능한 매체.

청구항 18

제11항에 있어서,

상기 스케일 비는 끝수가 버려지는 것인, 컴퓨터 판독 가능한 매체.

청구항 19

제11항에 있어서,

상기 스케일 비는 가장 근접한 정수로 반올림 되는 것인, 컴퓨터 판독 가능한 매체.

청구항 20

제11항에 있어서,

상기 2의 멱승 값을 상수를 포함하는 컴퓨터 판독 가능한 매체.

청구항 21

제1 비디오 화상, 제2 비디오 화상 및 제3 비디오 화상을 포함하는 스트림에 대한 방법으로서, 상기 방법은:

- a. (i) 상기 제3 비디오 화상에 대한 순서 값과 상기 제1 비디오 화상에 대한 순서 값 사이의 제1 순서 차이 값 및 (ii) 상기 제2 비디오 화상에 대한 순서 값과 상기 제1 비디오 화상에 대한 순서 값 사이의 제2 순서 차이 값에 기초한 스케일 값을 계산하는 단계; 및
- b. 상기 스케일 값 및 상기 제3 비디오 화상에 대한 모션 벡터에 기초해 상기 제2 비디오 화상에 대한 특정 모션 벡터를 계산하는 단계 - 상기 특정 모션 벡터를 계산하는 단계는 비트 시프트 연산을 포함함 - 를 포함하는, 방법.

청구항 22

제21항에 있어서,

특정 비디오 화상에 대한 순서 값은 다른 비디오 화상에 대한 상기 특정 비디오 화상의 시간상의 관계를 나타내는, 방법.

청구항 23

제21항에 있어서,

특정 비디오 화상에 대한 순서 값은 비디오 화상들의 시퀀스에 있어서 상기 특정 비디오 화상에 대한 위치를 나타내는, 방법.

청구항 24

제23항에 있어서,

상기 특정 비디오 화상에 대한 상기 위치는 출력 위치인, 방법.

청구항 25

제21항에 있어서,

비트 시프트 연산을 수행하는 단계는 8 비트씩 비트 값을 시프트하는 단계를 포함하는, 방법.

청구항 26

제21항에 있어서,

상기 계산된 모션 벡터를 이용해서 상기 제2 비디오 화상을 디코딩하는 단계를 더 포함하는, 방법.

청구항 27

제21항에 있어서,

상기 순서 값은 시간 값을 나타내는, 방법.

청구항 28

제21항에 있어서,
상기 순서 값은 시간 값을 나타내지 않는, 방법.

청구항 29

제21항에 있어서,
상기 순서 값은 상기 제2 비디오 화상에 연관된 슬라이스 헤더 내 저장된 값으로부터 도출되는, 방법.

청구항 30

제21항에 있어서,
상기 스케일 값은 2의 멱승 값에 기초하는, 방법.

청구항 31

제21항에 있어서,
상기 제1 비디오 화상은 I 비디오 화상인, 방법.

청구항 32

제21항에 있어서,
상기 제2 비디오 화상은 B 비디오 화상인, 방법.

청구항 33

제21항에 있어서,
상기 모션 벡터를 계산하는 단계는 보간법(interpolation) 연산을 수행하는 단계를 포함하는, 방법.

청구항 34

제21항에 있어서,
상기 모션 벡터를 계산하는 단계는 외삽법(extrapolation) 연산을 수행하는 단계를 포함하는, 방법.

청구항 35

제21항에 있어서,
상기 순서 값의 시퀀스는 비균일(non-uniform)한, 방법.

청구항 36

제21항에 있어서,
특정 순서 값은 압축된 포맷 형태인, 방법.

청구항 37

제36항에 있어서,
상기 특정 순서 값은 가변 길이 코딩을 이용해 압축된, 방법.

청구항 38

제36항에 있어서,
상기 특정 순서 값은 산술 코딩을 이용해 압축된, 방법.

청구항 39

제21항에 있어서,

특정 순서 값은 한번 이상 비트스트림 내 저장된 값으로부터 도출되는, 방법.

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

청구항 43

삭제

청구항 44

삭제

청구항 45

삭제

청구항 46

삭제

청구항 47

삭제

청구항 48

삭제

청구항 49

삭제

청구항 50

삭제

청구항 51

모션 벡터를 계산하기 위한 방법으로서, 상기 방법은:

- a. 2의 먹승 값에 기초하는 스케일 비를 계산하는 단계; 및
- b. 제1의 비디오 화상에 대한 제1의 모션 벡터를 계산하는 단계 - 상기 제1의 비디오 화상에 대해 계산된 제1의 모션 벡터는 상기 스케일 비와 제2의 비디오 화상에 대한 제2의 모션 벡터에 기초함 - 를 포함하는 모션 벡터 계산 방법.

청구항 52

제51항에 있어서,

상기 스케일 비는 제1 순서 차이 값과 제2 순서 차이 값 사이의 비율의 근사인, 모션 벡터 계산 방법.

청구항 53

제52항에 있어서,

상기 제1 순서 차이 값은 상기 제1 비디오 화상에 대한 제1 순서 값과 제3 비디오 화상에 대한 제3 순서 값 사이의 차이인, 모션 벡터 계산 방법.

청구항 54

삭제

청구항 55

명령어 세트를 저장하는 컴퓨터 판독 가능한 매체로서, 상기 명령어 세트는 하나 이상의 프로세서들에 의해 실행될 때 상기 하나 이상의 프로세서들이:

- a. 2의 멱승 값에 기초하는 스케일 비를 계산하는 단계; 및
- b. 제1의 비디오 화상에 대한 제1의 모션 벡터를 계산하는 단계 - 상기 제1의 비디오 화상에 대해 계산된 제1의 모션 벡터는 상기 스케일 비와 제2의 비디오 화상에 대한 제2의 모션 벡터에 기초함 - 를 실행하도록 하는, 컴퓨터 판독 가능한 매체.

청구항 56

명령어 세트를 저장하는 컴퓨터 판독 가능한 매체로서, 상기 명령어 세트는 하나 이상의 프로세서들에 의해 실행될 때 상기 하나 이상의 프로세서들이:

- a. 복수의 비디오 화상들 및 적어도 하나의 순서 값을 포함하는 비트스트림을 수신하는 단계; 및
- b. 제1 비디오 화상에 대한 제1 모션 벡터를 계산하는 단계 - 상기 제1 비디오 화상에 대해 계산된 상기 제1 모션 벡터는 스케일 비와 제2 비디오 화상에 대한 제2 모션 벡터에 기초하고, 상기 스케일 비는 하나 이상의 순서 값 및 2의 멱승 값으로부터 도출됨 - 를 실행하도록 하는, 컴퓨터 판독 가능한 매체.

청구항 57

제56항에 있어서,

상기 스케일 비는 제1 순서 차이 값과 제2 순서 차이 값 사이의 비율의 근사인, 컴퓨터 판독 가능한 매체.

청구항 58

제57항에 있어서,

상기 제1 순서 차이 값은 상기 제1 비디오 화상에 대한 제1 순서 값과 제3 비디오 화상에 대한 제3 순서 값 사이의 차이인, 컴퓨터 판독 가능한 매체.

청구항 59

제58항에 있어서,

상기 제2 순서 차이 값은 상기 제2 비디오 화상에 대한 제2 순서 값과 상기 제3 비디오 화상에 대한 상기 제3 순서 값 사이의 차이인, 컴퓨터 판독 가능한 매체.

청구항 60

삭제

청구항 61

제1 비디오 화상, 제2 비디오 화상 및 제3 비디오 화상을 포함하는 스트림에 대한 방법으로서, 상기 방법은:

- (i) 2의 멱승 값, (ii) 상기 제3 비디오 화상에 대한 순서 값과 상기 제1 비디오 화상에 대한 순서 값 사이의

제1 순서 차이 값, 및 (iii) 상기 제2 비디오 화상에 대한 순서 값과 상기 제1 비디오 화상에 대한 순서 값 사이의 제2 순서 차이 값에 기초한 스케일 값을 계산하는 단계; 및

상기 스케일 값 및 상기 제3 비디오 화상에 대한 모션 벡터에 기초해 상기 제2 비디오 화상에 대한 특정 모션 벡터를 계산하는 단계를 포함하는, 방법.

청구항 62

제61항에 있어서,

상기 계산된 모션 벡터를 이용해 상기 제2 비디오 화상을 디코딩하는 단계를 더 포함하는 방법.

청구항 63

제61항에 있어서,

상기 스케일 값은 상기 제2 순서 차이 값에 비례하는, 방법.

청구항 64

제61항에 있어서,

상기 스케일 값은 상기 제1 순서 차이 값에 역비례하는, 방법.

청구항 65

제61항에 있어서,

상기 특정 모션 벡터를 계산하는 단계는 2의 특정 멱승 값으로 나눗셈을 수행하는 것과 동등한 수학적 연산을 수행하는 단계를 포함하는, 방법.

청구항 66

제61항에 있어서,

상기 특정 모션 벡터를 계산하는 단계는 비트 시프트 연산을 수행하는 것과 동등한 수학적 연산을 수행하는 단계를 더 포함하는, 방법.

청구항 67

제1 비디오 화상, 제2 비디오 화상 및 제3 비디오 화상을 포함하는 스트림에 대한 방법으로서, 상기 방법은:

- (i) 상기 제3 비디오 화상에 대한 순서 값과 상기 제1 비디오 화상에 대한 순서 값 사이의 제1 순서 차이 값 및
- (ii) 상기 제2 비디오 화상에 대한 순서 값과 상기 제1 비디오 화상에 대한 순서 값 사이의 제2 순서 차이 값에 기초한 스케일 값을 계산하는 단계; 및

상기 스케일 값 및 상기 제3 비디오 화상에 대한 모션 벡터에 기초해 상기 제2 비디오 화상에 대한 특정 모션 벡터를 계산하는 단계 - 상기 특정 모션 벡터를 계산하는 단계는 2의 멱승 값에 의한 나눗셈을 포함함 - 를 포함하는, 방법.

청구항 68

제67항에 있어서,

특정 비디오 화상에 대한 순서 값은 다른 비디오 화상에 대한 상기 특정 비디오 화상의 시간상의 관계를 나타내는, 방법.

청구항 69

제67항에 있어서,

특정 비디오 화상에 대한 순서 값은 비디오 화상들의 시퀀스에 있어서 상기 특정 비디오 화상에 대한 위치를 특정하는, 방법.

청구항 70

제69항에 있어서,
상기 특정 비디오 화상에 대한 상기 위치는 출력 위치인, 방법.

청구항 71

제67항에 있어서,
상기 계산된 모션 벡터를 이용해서 상기 제2 비디오 화상을 디코딩하는 단계를 더 포함하는, 방법.

청구항 72

제67항에 있어서,
상기 스케일 값은 2의 멱승 값에 기초하는, 방법.

청구항 73

i) 제1 비디오 화상에 대한 순서 값과 제2 비디오 화상에 대한 순서 값 사이의 제1 순서 차이 값 및 (ii) 제3 비디오 화상에 대한 순서 값과 상기 제2 비디오 화상에 대한 순서 값 사이의 제2 순서 차이 값과 2의 멱승 값에 기초한 특정 값에 기초해 스케일 값을 계산하는 단계; 및
모션 벡터를 계산하기 위해 상기 스케일 값을 이용하는 단계를 포함하는, 방법.

청구항 74

제73항에 있어서,
특정 비디오 화상에 대한 순서 값은 다른 비디오 화상에 대한 상기 특정 비디오 화상의 시간상의 관계를 나타내는, 방법.

청구항 75

제73항에 있어서,
상기 스케일 값을 이용하는 단계는 상기 스케일 값에 상기 제3 비디오 화상과 연관된 모션 벡터를 곱함으로써 스케일된 모션 벡터를 계산하는 단계를 포함하는, 방법.

청구항 76

제75항에 있어서,
상기 스케일된 모션 벡터를 계산하는 단계는 특정한 2의 멱승 값에 의한 나눗셈을 수행하는 단계를 포함하는, 방법.

청구항 77

제76항에 있어서,
상기 특정한 2의 멱승 값은 256인, 방법.

청구항 78

제75항에 있어서,
상기 스케일 된 모션 벡터를 계산하는 단계는 비트 시프트 연산을 수행하는 단계를 포함하는, 방법.

청구항 79

a. 복수의 비디오 화상 및 적어도 하나의 순서 값을 수신하는 단계; 및
b. 상기 적어도 하나의 순서 값에 기초해서 비디오 화상에 대한 암시적 B 예측 블록 웨이팅을 계산하는 단계를

포함하는 방법.

청구항 80

제79항에 있어서,

상기 순서 값은 다른 비디오 화상에 대한 상기 비디오 화상의 시간상의 위치를 나타내는, 방법.

청구항 81

a. 적어도 하나의 순서 값을 이용해서 복수의 비디오 화상을 디코딩하는 단계 - 상기 순서 값은 참조 비디오 화상 선택을 위한 명령을 확립하기 위한 것임 -; 및

b. 상기 순서 값에 기초해서 디코딩된 비디오 화상들을 출력하는 단계를 포함하는, 방법.

청구항 82

제81항에 있어서,

상기 순서 값은 다른 비디오 화상에 대한 비디오 화상의 시간상의 위치를 나타내는, 방법.

청구항 83

적어도 하나의 프로세서들에 의해 실행 가능한 컴퓨터 프로그램을 저장한 컴퓨터 판독 가능한 매체로서, 상기 컴퓨터 프로그램은 청구항 21 내지 39 및 청구항 61 내지 82 중 어느 한 항에 따른 방법을 수행하기 위한 명령어 세트를 포함하는 컴퓨터 판독 가능한 매체.

청구항 84

청구항 21 내지 39 및 청구항 61 내지 82 중 어느 한 항에 따른 방법을 수행하기 위한 수단을 포함하는 컴퓨터 시스템.

명세서

기술 분야

<1> 본 발명은 멀티미디어 압축 시스템에 관한 것으로서, 특히 프로세서 집약적인 분할 연산의 필요성을 감소시키고 가변 정확도 화상간 타이밍을 특정하기 위한 방법 및 시스템에 관한 것이다.

배경 기술

<2> 디지털이 기반을 이루는 되는 전자 미디어 포맷은 아날로그 전자 매체 포맷을 광범위 하게 대체하고 있다. 디지털 콤팩트 디스크(CDs)는 오래전에 아날로그 비닐 레코드를 대체했으며, 아날로그 마그네틱 카세트 테이프는 거의 찾아볼 수 없게 되었다. 미니디스크와 MP3(MPEG Audio - layer 3)과 같은 2, 3 세대 디지털 오디오 시스템은 현재 콤팩트 디스크의 1세대 디지털 오디오 포맷으로부터 시장을 선점하고 있다.

<3> 비디오 미디어 포맷은 오디오 미디어 보다는 점진적으로 디지털 저장과 디지털 전송 포맷으로 바뀌고 있다. 이러한 디지털 채택이 점진적으로 일어나는 이유는 대개 디지털 형태로 적합한 품질의 비디오를 정밀하게 나타내는데 필요한 엄청난 양의 디지털 정보와, 압축된 비디오를 엔코드 하는데 빠른 프로세싱 능력이 필요하기 때문이다. 비디오를 정밀하게 나타내는데 필요한 엄청난 양의 디지털 정보는 고용량 디지털 저장 시스템과 고 밴드폭 전송 시스템이 필요하다.

<4> 그러나, 비디오는 현재 디지털 저장과 전송 포맷으로 급속히 바뀌고 있다. 빠른 컴퓨터 프로세서, 고밀도 저장 시스템, 새로운 효과적인 압축과 엔코딩 알고리즘은 디지털 비디오 전송과 저장을 소비자의 가격관점에서 볼때 매우 실용적으로 만들고 있다. DVD(Digital Versatile Disc), 즉 디지털 비디오 시스템은 해마다 최다 판매되는 가전제품중의 하나가 되고 있다. DVD는 고품질 비디오, 고성능 오디오, 편리함 등 여러가지 이유때문에 선택에 따른 선녹화 후재생 시스템의 비디오 카세트 레코더(VCRs)를 빠른속도로 대신하고 있다. 구식의 아날로그 NTSC(National Television Standards Committee) 비디오 전송 시스템은 최근 디지털 ATSC(Advanced Television Standards Committee) 비디오 전송 시스템으로 대체되는 과정에 있다.

- <5> 컴퓨터 시스템은 몇년 동안 다양하고 상이한 디지털 비디오 엔코딩 포맷을 사용해왔다. 특히, 컴퓨터 시스템은 디지털 비디오의 압축, 엔코딩 혹은 복원, 디코딩을 각각 실행하기 위해서 상이한 비디오 코더/디코더 방식을 사용해 왔다. 하드웨어 혹은 소프트웨어 구현에 있어서 비디오 코더/디코더 방식은 공통적으로 "코덱(CODEC)"으로 불리워 진다.
- <6> 컴퓨터 시스템에 의해 사용되는 최대의 디지털 비디오 압축 및 엔코딩 시스템들은 중에는, 그 두문자어인 MPEG이라고 일반적으로 알려진 모션 픽처스 엑스퍼트 그룹(Motion Pictures Expert Group)에 의해 지지된 디지털 비디오 시스템들이 있다. 가장 유명하고 많이 사용되고 있는 MPEG의 디지털 비디오 포맷은 MPEG-1, MPEG-2, MPEG-4 이다. VCDs(Video CDs)와 초기 소비자 등급 디지털 비디오 편집 시스템은 초기 MPEG-1 디지털 비디오 엔코딩 포맷을 사용한다. DVDs(Digital Versatile Discs)와 DBS(Dish Network brand Direct Broadcast Satellite) 텔레비전 방송 시스템은 고품질의 MPEG-2 디지털 비디오 압축과 엔코딩 시스템을 사용한다. MPEG-4 엔코딩 시스템은 디지털 비디오 엔코더와 관련 디지털 비디오 플레이어를 기초로 하는 최신 컴퓨터에 의해 빠르게 채택되고 있다.
- <7> MPEG-2와 MPEG-4 기준은 일련의 비디오 프레임 또는 비디오 필드를 압축한 후 압축된 프레임 혹은 필드를 디지털 비트스트림으로 엔코딩한다. 비디오 프레임 또는 필드를 MPEG-2와 MPEG-4 시스템으로 압축할 때, 비디오 프레임 또는 필드는 화소블록의 직각 그리드로 분할된다. 각각의 화소블록은 독립적으로 압축되고 엔코딩된다.
- <8> 비디오 프레임 혹은 필드를 압축할 때, MPEG-4 기준은 프레임 혹은 필드를 인트라-프레임 (I-프레임), 단방향 예측 프레임 (P-프레임), 양방향성 예측 프레임 (B-프레임) 세가지 압축된 프레임 혹은 필드 중 하나의 타입으로 압축할 것이다. 인트라 프레임은 다른 비디오 프레임과 관계 없이 완전히 독립적으로 독립적 비디오 프레임을 엔코딩한다. P-프레임은 하나의 미리 디스플레이된 비디오 프레임과 연관된 비디오 프레임을 형성한다. B-프레임은 현재 프레임 이전에 디스플레이된 비디오 프레임과 현재 프레임 이후에 디스플레이 된 비디오 프레임 모두와 연관된 비디오 프레임을 형성한다. 이들 용장 비디오 정보의 효과적인 사용법 때문에 P-프레임과 B-프레임은 일반적으로 최상의 압축을 제공한다.
- <9> 본 발명의 목적은 비디오 코덱에서 모션 추정(motion estimation)을 구현하기 위한 방법과 장치를 제공하는 것이다. 특히 분할 연산의 방대한 수를 사용하지 않고 매우 효과적인 방법으로 추정 모션 벡터(estimated motion vectors)를 신속하게 계산하는 시스템을 제공하는 것이다.
- <10> 하나의 실시예로, 제1의 피승수는 제1의 비디오 화상과 제2의 비디오 화상 사이의 제1의 디스플레이 시간 차이에 2의 멱승 형태의 스케일 값을 곱함으로써 결정된다. 이러한 과정은 비율에서 분자를 확대시킨다. 다음으로, 상기의 시스템은 스케일 분자를 제2의 비디오 화상과 제3의 비디오 화상 사이의 제2의 최초 디스플레이 시간(second first display time)으로 나누어 스케일 비를 정한다. 그리고, 스케일 비는 모션 벡터 추정을 계산하기 위해 후에 사용되도록 저장된다. 상기 스케일 비를 저장함으로써, 스케일 비가 주요 비트를 세이브하고, 스케일의 축소가 나누기 연산에 소요되는 필요 시간을 제거하는 방식의 단순 시프트로 실행되기 때문에, 모든 추정된 모션 벡터는 좋은 정밀도로 신속하게 계산될 수 있다.
- <11> 본 발명의 목적, 특징, 장점들은 도면 및 실시예에서 명확하게 설명될 것이다.

실시예

- <15> 본 발명은 멀티미디어 압축과 엔코딩 시스템에 있어서, 분할 연산의 필요성을 줄이고, 가변 정확도 화상간 타이밍의 특징을 위한 방법과 시스템에 관한 것이다. 보다 자세한 설명을 위해, 다음의 설명에서 본 발명의 이해를 위한 특정 명칭이 먼저 설명되고 있다. 그러나, 당업자 입장에서 볼 때 이들의 상세한 설명이 본 발명의 실행을 위해 불필요하다는 사실은 명백하다. 예를 들어, 본 발명은 MPEG 멀티미디어 압축 및 엔코딩 시스템에 관하여 상술하고 있다. 하지만, 동일 기술들이 다른 타입의 압축 및 엔코딩 시스템에 손쉽게 적용될 수 있다.
- <16> **멀티미디어 압축, 엔코딩 개요**
- <17> 도 1은 업계에 잘 알려진 전형적인 디지털 비디오 엔코더(100)의 고 레벨 블록도를 나타내고 있다. 디지털 비디오 엔코더(100)는 블록도의 왼쪽에서 비디오 프레임(105)의 입력 비디오 스트림을 수신한다. 디지털 비디오 엔코더(100)는 각 비디오 프레임을 화소블록의 그리드로 분해한다. 이 화소블록은 독립적으로 압축된다. 화소블록의 가변적이고 다양한 사이즈는 다른 비디오 엔코딩 시스템에 의해 사용될 수 있을 것이다. 예를 들어, 다른 화소 블록 해상도는 8x8, 8x4, 16x8, 4x4 등을 포함한다. 또한, 화소블록은 때때로 '매크로블록(macroblocks)'

으로 불리기도 한다. 본 명세서에서는 모든 사이즈의 화소들의 블록을 언급할 때, 화소블록이란 단어를 사용할 것이다.

<18> 이산 코사인 변환(DCT) 유닛(110)은 비디오 프레임에 있는 각 화소블록을 처리한다. 상기 프레임은 독립적으로 처리되거나(인트라-프레임), 모션 보상 유닛(인터-프레임)에서 수신된 다른 프레임들의 정보와 같이 처리될 것이다. 다음으로, 양자화(Q) 유닛(120)은 이산 코사인 변환 유닛(110)의 정보를 양자화 한다. 마지막으로, 양자화된 비디오 프레임은 엔코딩된 비트스트림(bitstream)을 산출하기 위해 엔트로피 엔코더(H) 유닛(180)에서 엔코딩된다. 엔트로피 엔코더(H) 유닛(180)은 가변 길이 코딩(VLC) 시스템을 사용할 것이다.

<19> 비디오 프레임이 엔코딩된 인터 프레임은 근처의 다른 비디오 프레임과 관련하여 규정되므로, 디지털 비디오 엔코더(100)는 각각 디코딩된 프레임이 인터 프레임들이 엔코딩된 디지털 비디오 디코더 내에서 어떻게 나타날 것인지의 카피(copy)를 생성해야 한다. 따라서, 디지털 비디오 엔코더(100)의 아랫 부분은 사실상 디지털 비디오 디코더 시스템이다. 특히, 역 양자화(Q⁻¹) 유닛(130)은 비디오 프레임 정보의 양자화를 역전시키고 역 이산 코사인 변환(DCT⁻¹) 유닛(140)은 비디오 프레임 정보의 이산 코사인 변환을 역전한다. 역 이산 코사인 변환(DCT⁻¹) 유닛(140)으로부터 모든 DCT 계수가 재구성 되면, 모션 보상 유닛은 엔코딩된 비디오 프레임을 재구성하기 위해 모션 벡터에 따른 정보를 사용할 것이다. 재구성된 비디오 프레임은 나중 프레임의 모션 추정을 위한 참조 프레임으로 사용되어 진다.

<20> 이후, 디코딩된 비디오 프레임은 그 비디오 프레임에 있는 정보와 관련되어 형성되는 인터 프레임(P 프레임 혹은 B 프레임)을 엔코딩 하는데 사용되며, 특히 모션 보상(MC) 유닛(150)과 모션 추정(ME) 유닛(160)은 모션 벡터를 결정하는데 사용되며 인터 프레임을 엔코딩 하는데 사용되는 미분 값을 형성한다.

<21> 비율 제어기(190)는 디지털 비디오 엔코더(100)의 다양한 소자로부터 정보를 수신하며, 이 정보를 사용해서 각 비디오 프레임의 비트 예산을 배정한다. 비율 제어기(190)는 제한의 특정 세트에 따르는 최고 품질의 디지털 비디오 비트 스트림을 발생시키도록 하는 방법으로 비트 예산을 배정해야 한다. 특히, 비율 제어기(190)는 버퍼들의 오버플로잉(overflowing)(저장될 수 있는 것보다 더 많은 양의 정보를 보냄으로써 비디오 디코더 내의 수용 가능한 메모리 양을 초과함) 또는 버퍼들의 언더플로잉(underflowing)(비디오 디코더가 디스플레이할 비디오 프레임들을 충분히 소진할 정도의 빠르기로 비디오 프레임들을 보내지 않음) 없이 최상의 품질로 압축된 비디오를 생성하도록 한다.

<22> **화소 블록을 갖는 디지털 비디오 엔코딩**

<23> 몇몇의 비디오 신호에서는 성공적인 비디오 화상들(프레임 혹은 필드) 사이의 시간이 상수가 아닐 수도 있다. (주의 : 이 명세서에서는 일반적으로 비디오 프레임이나 비디오 필드를 언급할 때 비디오 화상이란 용어를 사용한다.) 예를 들어, 어떤 비디오 화상은 아마도 밴드폭 제약 전송으로 인해 저하될 것이다. 또한, 비디오 타이밍은 카메라 불규칙성 혹은 슬로우 모션이나 빠른 모션과 같은 특정한 효과들 때문에 변할 수 있다. 어떤 비디오 스트림에서는, 오리지널 비디오 소스(original video source)가 단순하게 디자인에 의해 획일적이지 않은 화상 간 타이밍을 가질 수도 있다. 예를 들어, 컴퓨터 그래픽 애니메이션과 같이 합성된 비디오는 비디오 카메라 시스템과 같은 획일적인 타이밍 비디오 캡처 시스템에 의한 어떠한 임의의 비디오 타이밍도 부과되지 않기 때문에 획일적이지 않은 타이밍을 가질 수 있다. 유연한 디지털 비디오 엔코딩 시스템은 획일적이지 않은 비디오 화상 타이밍을 조절할 수 있어야 한다.

<24> 앞서 언급한 바와 같이, 대부분의 디지털 비디오 엔코딩 시스템은 비디오 화상들을 화소블록의 직각 그리드로 분해한다. 비디오 화상의 각 개별적 화소들은 독립적으로 압축되고 엔코딩된다. 예를 들어, ISO MPEG 혹은 ITU H.264와 같은 비디오 코딩 기준은 비디오 화상을 엔코딩 하기 위해서 예측되는 화소블록들의 다른 종류들을 사용한다. 그 한 예로, 화소블록은 다음의 세가지 종류중에서 하나가 될 것이다.

<25> 1. I-화소블록 - 인트라 (I) 화소 블록, 완전하게 자기규정 되어있으며, 코딩에 있어서 다른 비디오 화상으로부터 정보를 사용하지 않는다.

<26> 2. P-화소블록 - 단방향성 예측 (P) 화소블록은 하나의 이전 화상으로부터 화상 정보를 참조한다.

<27> 3. B-화소블록 - 양방향성 예측 (B) 화소블록은 하나의 이전 화상과 하나의 이후 비디오 화상으로부터 정보를 사용한다.

<28> 만약 비디오 화상의 모든 화소 블록이 인트라 화소블록이라면, 비디오 화상은 인트라 프레임이다. 만약 비디오

화상이 단방향 예측 매크로블록이나 인트라 화소블록만을 포함하고 있다면, 비디오 화상은 P 프레임이다. 만약 비디오 화상이 어떠한 양방향성 예측 화소블록을 포함하고 있다면, 비디오 화상은 B-프레임이다. 단순화를 위해, 본 명세서에서는 주어진 화상 내에서 모든 화소 블록들이 동일한 종류로하는 경우를 고려할 것이다.

<29> 엔코드 될 비디오 화상들의 예시 시퀀스는 다음과 같이 나타날 것이다. :

<30> $I_1B_2B_3B_4P_5B_6B_7B_8B_9P_{10}B_{11}P_{12}B_{13}I_{14}\dots$

<31> 여기서, 문자 (I, P 혹은 B)는 비디오 화상이 I-프레임, P-프레임 혹은 B-프레임인 것을 나타내며 숫자는 비디오 화상의 시퀀스에서 비디오 화상의 카메라 순서를 나타낸 것이다. 카메라 순서는 비디오 화상이 기록된 카메라의 순서이며, 또한 비디오 화상들이 디스플레이 되는(디스플레이 순서) 비디오 순서 이다.

<32> 앞에서 언급된 비디오 화상의 일련의 예시들은 도 2에 도식적으로 나타나있다. 도 2에서, 화살표들은 저장된 화상(이 케이스의 I-프레임 혹은 P-프레임)의 화소블록들이 다른 화상들의 모션 추정 예측에 사용되는 것을 나타낸다.

<33> 도 2에서는 다른 화상들로부터의 정보가 인트라 프레임 비디오 화상 I_1 의 엔코딩에 사용되지 않음을 알 수 있다. 비디오 화상 P_5 는 비디오 화상 I_1 에서 비디오 화상 P_5 로 연결된 화살표와 같이 코딩에 있어서 이전 비디오 화상 I_1 으로부터 비디오 정보를 사용하는 P 프레임이다. 비디오 화상 B_2 , 비디오 화상 B_3 , 비디오 화상 B_4 모두는 비디오 화상 I_1 와 비디오 화상 P_5 가 비디오 화상 B_2 , 비디오 화상 B_3 , 비디오 화상 B_4 로 연결되는 화살표와 같이 코딩에 있어서 비디오 화상 I_1 과 비디오 화상 P_5 의 둘 모두의 정보를 사용한다. 일반적으로, 상기 언급된 바와 같이 화상간 타임은 동일하지 않다.

<34> B 화상들이 이후 화상(future pictures ; 이후에 디스플레이 될 화상들)으로부터의 정보를 사용하기 때문에, 전송 순서는 대부분 디스플레이 순서와는 다르다. 특히, 다른 비디오 화상을 구성하는 것이 필요한 비디오 화상들은 첫 번째로 전송되어야한다. 상기의 시퀀스에 따르면, 전송 순서는 다음과 같이 되어야 할 것이다.

<35> $I_1P_5B_2B_3B_4P_{10}B_6B_7B_8B_9P_{12}B_{11}I_{14}B_{13}\dots$

<36> 도 3은 도 2에서 비디오 화상들의 이전 전송 순서를 도식적으로 나타내고 있다. 다시 말해서, 상기 도면에 있는 화살표들은 저장된 비디오 화상 (본 케이스에서 I 혹은 P)의 화소블록들이 다른 비디오 화상들의 모션 추정 예측에 사용되는 것을 나타낸다.

<37> 도 3을 참조해 보면, 시스템은 다른 프레임에 의존하지 않는 I 프레임 I_1 을 첫번째로 전송한다. 다음에 시스템은 비디오 화상 I_1 에 의존하는 P프레임 비디오 화상 P_5 를 전송시킨다. 다음으로 시스템은 비디오 화상 B_2 가 비디오 화상 P_5 이전에 디스플레이 되더라도 비디오 화상 P_5 이후에 B 프레임 비디오 화상 B_2 를 전송시킨다. 그 이유는 비디오 화상 B_2 를 디코드 하는데 있어서, 디코더는 비디오 화상 B_2 를 디코드 하는데 필요한 비디오 화상 I_1 과 P_5 에서 정보를 수신하고 저장할 것이기 때문이다. 마찬가지로 비디오 화상 I_1 과 P_5 는 연속 비디오 화상 B_3 과 비디오 화상 B_4 를 디코드 하도록 사용될 것이다. 수신기/디코더는 적절한 디스플레이를 위해 비디오 화상 시퀀스를 재명령한다. 이러한 작동원리에서 I와 P 화상은 때때로 저장된 화상으로 언급된다.

<38> P 프레임 화상의 코딩은 일반적으로 모션 추정을 이용하며, 모션 벡터는 화상의 각 화소블록으로 계산된다. 계산된 모션 벡터를 사용하는 것은 예측 화소블록(P-화소블록)이 전술한 이전 화상에서 화소들의 번역에 의해 형성된다. P 프레임 화상의 실제 화소 블록과 예측 화소 블록 사이의 차는 전송을 위해 코드화 된다.

<39>

<40> P-화상

<41> P 화상 코딩은 일반적으로 모션 추정(MC)을 사용하며, 이전 화상에서 위치를 지정하는 모션 벡터(MV)는 현재 화상에서 각 화소블록으로 계산된다. 모션벡터를 사용하면서, 예측 화소블록은 앞서 언급한 이전 화상에서 화소의 번역에 의해 형성될 수 있다. P 화상에서의 실제 화소블록과 예상 화소 블록 사이의 차는 전송을 위해 코드되어 진다.

<42> 각 모션 벡터는 또한 예측된 코딩을 통해서 전송되어 진다. 예를 들어, 모션 벡터 예측은 근처에 있는 모션 벡

터를 사용해서 형성된다. 이러한 케이스에서, 실제 모션 벡터와 모션 벡터 예측 사이의 차는 전송을 위해 코드되어 진다.

<43> B - 화상

<44> 각각의 B 화소블록은 앞서 언급된 이전 비디오 화상을 참조하는 첫번째 모션 벡터와 이후 비디오 화상을 참조하는 두번째 모션 벡터의 두가지 모션 벡터를 사용한다. 이 두개의 모션 벡터들로부터 두개의 예측 화소들이 계산된다. 두개의 예측 화소블록은 최종 예측 화소블록을 형성하기 위해서 어떤 기능을 사용하여 결합된다. 상기와 같이, B 프레임 화상의 실제 화소블록과 최종 예측 화소 블록 사이의 차는 전송을 위해 코드되어 진다.

<45> P 화소블록들과 같이, B 화소블록의 각 모션 벡터(MV)는 예측된 코딩을 통해서 전송될 것이다. 특히, 예측 모션 벡터는 근처의 모션 벡터 사용으로 형성된다. 그 후에, 실제 모션 벡터와 예측 사이의 차는 전송을 위해 코드되어 진다.

<46> 하지만, B 화소블록을 갖는 가장 가까운 곳에 저장된 화상 화소블록의 모션 벡터로부터 모션 벡터를 보간 할 수 있는 기회도 존재한다. 이러한 모션 벡터 보간은 디지털 비디오 엔코더와 디지털 비디오 디코더 모두에서 실행된다.

<47> 모션 벡터 보간은 특히 카메라가 움직임이 없는 배경을 지나 천천히 패닝(panning) 하는 곳의 비디오 시퀀스로부터 비디오 화상에 잘 작용한다. 사실, 이러한 모션 벡터 보간은 단독으로 사용되기에도 충분하며, 특히 이는 미분 정보가 보간을 사용하여 엔코딩된 B 화소 블록 모션 벡터로 계산되거나 전송되지 않아도 됨을 의미한다.

<48> 더욱 자세한 설명을 위해, 상기의 시나리오는 화상 i와 j의 화상간 디스플레이 시간을 $D_{i,j}$ 로 나타내며, 만약 화상의 디스플레이 시간이 각각 T_i 와 T_j 이라면,

<49> $D_{i,j} = T_i - T_j$ 이며,

<50> 이는 $D_{i,k} = D_{i,j} + D_{j,k}$

<51> $D_{j,k} = -D_{k,j}$

<52> 임을 알 수 있다.

<53> $D_{i,j}$ 는 어떤 케이스에서는 음수가 될 수 있음을 주의해야 한다.

<54> 따라서, 만약 $MV_{5,1}$ 이 I_1 과 관련된 P_5 화소 블록의 모션 벡터라면, 각 I_1 , P_5 와 관련된 B_2 , B_3 , B_4 모션 벡터에 상응하는 화소 블록은 각각 다음과 같이 보간될 것이다.

$$\begin{aligned} MV_{2,1} &= MV_{5,1} * D_{2,1} / D_{5,1} \\ MV_{5,2} &= MV_{5,1} * D_{5,2} / D_{5,1} \end{aligned}$$

$$\begin{aligned} MV_{3,1} &= MV_{5,1} * D_{3,1} / D_{5,1} \\ MV_{5,3} &= MV_{5,1} * D_{5,3} / D_{5,1} \end{aligned}$$

$$\begin{aligned} MV_{4,1} &= MV_{5,1} * D_{4,1} / D_{5,1} \\ MV_{5,4} &= MV_{5,1} * D_{5,4} / D_{5,1} \end{aligned}$$

<55>

<56> 디스플레이 시간의 비율이 모션 벡터 예측을 위해 사용되기 때문에, 절대적인 디스플레이 시간은 필요하지 않을을 알아야 한다. 따라서, 상대적인 디스플레이 시간이 D_j 화상간 디스플레이 시간 값에 사용될 것이다.

<57> 이러한 시나리오는 H.264 기준에서 예시된 것과 같이 일반화 될 것이다. 종합적으로, P 또는 B 화상은 모션 벡터 예측을 위해 이전에 전송된 화상을 사용할 것이다. 따라서, 상기의 케이스에서 B3 화상은 예측(prediction)에 있어서 화상 I_1 과 B_2 를 사용할 것이다. 또한, 모션 벡터는 보간만 되는 것 뿐만 아니라 외삽(extrapolate)되

어질 것이다. 따라서, 이 케이스는 다음과 같은 사실을 도출할 수 있다.

$$MV_{3,1} = MV_{2,1} * D_{3,1} / D_{2,1}$$

- <58>
- <59> 이러한 모션 벡터 외삽(혹은 보간)은 모션 벡터의 예측 코딩을 위한 예측 과정에도 사용될 수도 있다.
- <60> **엔코딩 화상간 디스플레이 시간**
- <61> 비디오 시퀀스의 가변 화상간 디스플레이 시간은 매우 높은 코딩 효율을 얻을수 있고 비디오 디코더의 조건에 맞는 선택가능한 정확도를 가질 수 있는 방법으로 엔코드되고 전송되어야 한다. 이론적으로 말하면, 인코딩 시스템은 비교적 단순한 컴퓨터 시스템이 디지털 비디오를 디코드 할 수 있는 것과 같이 디코더를 위한 작업을 단순화 해야 한다.
- <62> 가변 화상간 디스플레이 시간은 차등 모션 벡터, 다이렉트 모드 모션 벡터(Direct Mode motion vectors), 혹은 암시적 B 예측 블록 웨이팅(Implicit B Prediction Block Weighting)을 계산하기 위해서 다수의 다른 비디오 엔코딩 시스템이 잠재적으로 필요하다.
- <63> 비디오 시퀀스에 있어서 가변 화상간 디스플레이 시간의 문제점은 시간상 참조의 사용이 뒤엉키는데 있다. 이론적으로 말하면, 비디오 코덱(video CODEC)의 출력 화상에서 정확한 화소 값의 차등은 화상이 디코드되거나 디스플레이 되는 시간에서 독립되어야 한다. 그러므로, 타이밍 발행과 시간 참조는 코덱 레이어의 바깥에서 해결되어야 한다.
- <64> 요구되는 시간 독립성의 근본에는 코딩 및 시스템 모두에 관계된 이유가 있다. 비디오 코덱에서는 시간 참조가 (1)참조 화상 선택을 위한 명령을 확립하기 위해, (2)화상 사이의 모션 벡터를 보간하기 위한, 두가지 목적으로 사용되어 진다.
- <65> 참조 화상 선택을 위한 명령을 확립하기 위해서, 상대적 위치 값을 단순히 보낸다. 예를 들면, 디코드 명령에 프레임 위치 N과 디스플레이 위치 M사이의 차이, 즉 N-M 에서는 시간 스탬프(time stamps) 혹은 다른 시간 참조들이 필요하지 않다. 모션 벡터를 보간하기 위해서, 시간상 간격이 보간 간격과 관련 있을 수 있다면, 시간상 간격은 유용할 것이다. 하지만, 모션이 비선형이라면 그렇지 않을 것이다. 따라서, 모션 벡터 보간을 위해서는 시간상 정보보다는 파라미터를 전송하는 것이 더 적절하다.
- <66> 시스템에 있어서, 전형적인 비디오 코덱(CODEC)은 비디오 코덱이 다른 비디오 (그리고 오디오) 코덱들과 공존하는 넓은 시스템의 한 부분으로 생각 될 수 있다. 이러한 멀티 코덱 시스템에 있어서, 좋은 시스템 레이어링과 디자인은, 타이밍과 같이 논리적으로 코덱 독립인 일반적인 기능들이 코덱의 바깥쪽에서 조절되는 것이 필요하다. 시스템에 의한 타이밍의 운용과 독립적으로 각 코덱에 의하지 않은 타이밍의 운용은 동기화와 같이 공통적인 기능들의 모순 없는 운용을 이루는데 있어서 매우 중요하다. 예를 들어, 비디오/오디오 표현과 같이 하나의 스트림 이상을 동시에 다루는 시스템에서, 타이밍 조정은 상이한 스트림 동기화를 유지하기 위해서 때때로 스트림 내부에 필요하다. 이와 유사하게, 상이한 클럭 타이밍 조정의 원격 시스템으로부터 스트림을 다루는 시스템에서는 리모트 시스템의 동기화를 유지하는 것이 필요하다. 이러한 타이밍 조정은 시간 스탬프 사용을 이룰 수 있게 한다. 예를 들면, 송신장치로부터 "송신기 리포트(Sender Reports)" 의 의미로 링크되고 각 스트림의 RTP 레이어에서 RTP로 공급되는 타임 스탬프는 동기화로 사용되어 질 것이다. 이러한 송신기 리포트는 다음과 같은 형태를 갖는다.
- <67> 비디오 RTP 타임스탬프 X는 타임스탬프 Y와 관련하여 조정되어 진다.
- <68> 오디오 RTP 타임스탬프 W는 타임스탬프 Z와 관련하여 조정되어 진다.
- <69> 두개의 스트림이 조정되도록 참조 타임스탬프의 벽 시계 비율(wall-clock rate)이 알려져 있다. 하지만, 이 타임스탬프 참조들은 모두 주기적 및 개별적으로 두 스트림들에 도달하며, 두 스트림을 재조정하게 만드는 원인이 된다. 이는 일반적으로 오디오를 매치하기 위한 비디오 스트림 조절에 의해 이루어 지거나, 혹은 그 반대의 경우를 들 수 있다. 타임 스탬프의 시스템 핸들링은 디스플레이 되어지는 화상 값에 영향을 주어서는 안된다. 다시 말하면, 시간상 정보의 시스템 핸들링은 코덱 바깥에서 형성되어야 한다.

<70> 삭제

<71> 실시예

<72> 앞서 언급된 바와 같이, 정형화되지 않은 화상간 타임의 케이스에서의 문제점은, 화상간 디스플레이 시간 값 $D_{i,j}$ 을 효과적인 방법으로 디지털 비디오 전송기로 전송하는 것이다. 상기 목표를 달성하기 위한 한가지 방법은 현재 화상과 첫번째 화상 이후의 각 화상의 가장 최근에 전송되어 저장된 화상 사이의 디스플레이 시간 차를 전송시키는 시스템을 가지는 것이다. 에러 내성(error resilience)을 위해, 전송은 상기 화상 내부에서 몇번 동안 반복될 수 있다. 예를 들면, 디스플레이 시간 차는 MPEG나 H.264기준의 슬라이스 헤더(slice header)에서 반복될 수 있다. 만약 모든 슬라이스 헤더가 손실된다면, 정보 디코딩을 위해 손실된 화상에 의지된 다른 화상들 또한 디코드 될 수 없을 것이다.

<73> 따라서, 앞에서 언급된 실시예와 관련하여, 시스템은 다음의 화상간 디스플레이 값들을 전송할 것이다.

$$D_{5,1} D_{2,5} D_{3,5} D_{4,5} D_{10,5} D_{6,10} D_{7,10} D_{8,10} D_{9,10} D_{12,10} D_{11,12} D_{14,12} D_{13,14} \dots$$

<74>

<75> 모션 벡터 추정을 위해서, 화상간 디스플레이 시간 $D_{i,j}$ 에 필요한 정확도는 화상에 따라 매우 다양하다. 예를 들어, 만약 두개의 P 프레임 화상 B_6 와 P_7 사이의 단일 B 프레임 화상 B_7 이 충분하지 않다면,

$$D_{7,5} = 2 \text{ and } D_{6,7} = -1$$

<76>

<77> 을 보내는 것으로도 충분하다.

<78> 여기서 $D_{i,j}$ 화상간 디스플레이 시간 값은 상대 시간 값이다.

<79> 대신에, 만약 비디오 화상 B_6 가 비디오 화상 P_5 와 비디오 화상 P_7 사이의 거리의 1/4 이라면, 전송을 위한 적절한 $D_{i,j}$ 보조 화상 디스플레이 시간 값은,

$$D_{7,5} = 4 \text{ and } D_{6,7} = -1$$

<80>

<81> 이 된다.

<82> 앞서 언급된 두가지 실시예에서, 비디오 화상 B_6 와 비디오 화상 P_7 (보조 화상 디스플레이 시간 $D_{6,7}$)사이의 디스플레이 시간은 디스플레이 시간 "유닛(units)" 값으로 사용된 것에 유의해야 한다. 가장 최근의 실시예에서, 비디오 화상 P_5 와 비디오 화상 P_7 (보조 화상 디스플레이 시간 $D_{6,7}$)사이의 디스플레이 시간 차는 네개의 디스플레이 시간 "유닛"(4* $D_{6,7}$)이다.

<83> 개선된 디코딩 효율

<84> 일반적으로, 나눗수가 2의 멱승이면, 모션 벡터 추정 계산값은 매우 단순해진다. 이는 두개의 저장된 화상 사이의 $D_{i,j}$ (화상간 시간)이 도 4에 도시된 바와 같이 2의 멱승일 경우에는, 실시예에서 쉽게 달성될 수 있다. 선택적으로, 상기 추정 과정은 2의 멱승으로 모든 나눗수의 뒷자리를 버리거나 끝수를 반올림하여 규정될 수 있다.

<85> 보조 화상 시간이 2의 멱승일 경우에는, (2의) 정수 멱승 형태의 수가 보조 화상 시간의 전체 값 대신에 전송되었을 때, 데이터 비트 숫자는 줄어 든다. 도 4는 화상 사이의 거리가 2의 멱승일 경우를 나타내고 있다. 이러한 케이스에서는, 비디오 화상 P_1 과 비디오 화상 P_3 사이의 $D_{3,1}$ 디스플레이 시간 값이 $1(2^1=2)$ 이기 때문에)로 전송되며, 비디오 화상 P_7 과 비디오 화상 P_3 사이의 디스플레이 시간 값, 4는 $2(2^2=4)$ 이기 때문에) 전송된다.

<86> 선택적으로, 외삽(extrapolation) 작용의 모션 벡터 보간은 분모가 2의 멱승인 방법의 스케일링(scaling)에 의

해서, 원하는 정확도로 접근할 수 있다(분모 나눗셈에 있어서 2의 멱승은, 나누어질 값에 있는 비트들을 단순히 시프팅(shifting)시킴으로써 실행될 수 있다). 예를 들면,

$$D_{5,4}/D_{5,1} \sim Z_{5,4}/P$$

<87>

<88> 에서, P값은 2의 멱승이고 $Z_{5,4} = P * D_{5,4}/D_{5,1}$ 은 가장 가까운 정수값으로 반올림 하거나 끝수를 버린다. P 값은 주기적으로 전송되거나 시스템의 상수로 고정될 것이다. 하나의 예로, P 값은 $P=2^{256}$ 으로 고정된다.

<89> 이러한 접근법의 이점은, 디코더가 각 화상 마다 한 번 씩 $Z_{5,4}$ 만 계산하거나, 혹은 많은 케이스에서 디코더가 Z 값을 미리 계산하고 저장할 수 있다는 것이다. 이는 디코더가 모션 벡터 보간이 더욱 효율적으로 이루어지는 것과 같이 화상에서 모든 모션 벡터가 $D_{5,1}$ 로 나누어 지는 것을 방지한다. 예를 들면, 정상 모션 벡터 계산은 다음과 같다.

$$MV_{5,4} = MV_{5,1} * D_{5,4}/D_{5,1}$$

<90>

<91> 하지만, 만약 $Z_{5,4}$ 를 $Z_{5,4} = P * D_{5,4}/D_{5,1}$ 로 계산하고 저장한다면, 그 결과는 다음과 같다.

$$MV_{5,4} = MV_{5,1} * Z_{5,4}/P$$

<92>

<93> 하지만 P값이 2의 멱승으로 되었기 때문에, P로 나누는 것은 단지 비트의 단순 시프트이다. 따라서, Z값이 비디오 화상에 대해 일단 계산되면, 단일 곱셈과 단일 시프트가, 다음의 화소블록들에 대한 모션 벡터들을 계산하는데 요구된다. 또한, 시스템은 중요한 비트(significant bit)들이 계산 중에 손실되지 않도록 모든 나눗셈을 수행함으로써 높은 정확도를 유지할 수 있다. 이러한 방법으로, 디코더는 엔코더와 정확히 같은 모션 벡터 보간을 수행하며, 따라서 그렇지 않은 경우에 발생할 수 있는 미스매치 문제들을 회피할 수 있다.

<94> 나누기(2의 멱승으로 나누는 것을 제외하고)는 디지털 컴퓨터 시스템에 있어서 덧셈이나 곱셈보다 더욱 계산적으로 철저한 작업이기 때문에, 이러한 접근법은 모션 벡터 보간이나 외삽을 사용하는 화상의 재구성에 필요한 계산을 크게 줄일 수 있다.

<95> 어떤 케이스에서는, 모션 벡터 보간이 사용되지 않을 수도 있다. 하지만, 모션 벡터 보간은 수신기/플레이어 시스템이 적절한 순서로 비디오 화상을 디스플레이하는 것과 같이 수신기/플레이어 시스템으로 비디오 화상의 디스플레이 순서를 전송하는 데에 필요하다. 이러한 경우에, $D_{i,j}$ 의 단순 부호를 갖는 정수 값은 실제 디스플레이 시간에 관계없이 충분하다. 어떤 응용에서는, 화상 순서를 재구성하는데 오직 부호(양수 혹은 음수)만이 필요함을 언급하고 있다.

<96> 화상간 시간 $D_{i,j}$ 는 단순 부호를 갖는 정수 값으로 전송될 것이다. 하지만, 추가적인 압축을 이루기 위한 $D_{i,j}$ 값의 엔코딩에 많은 방법들이 사용될 수 있다. 예를 들면, 가변 길이 코드 절대값의 뒤에 오는 부호 비트는 코딩 효율의 조건을 충족시키고 제공하는데 있어서 상대적으로 용이하다.

<97> 사용될 수 있는 그러한 가변 길이 코딩 시스템은 UVLC(Universal Variable Length Code)로 알려져 있다. 상기 UVLC 가변 길이 코딩 시스템은 다음의 부호어를 가진다. :

1 = 1
 2 = 0 1 0
 3 = 0 1 1
 4 = 0 0 1 0 0
 5 = 0 0 1 0 1
 6 = 0 0 1 1 0
 7 = 0 0 1 1 1
 8 = 0 0 0 1 0 0 0...

<98>

<99> 화상간 시간을 인코딩하는 다른 방법은 산술 코딩(arithmetic coding)을 사용하는 것이다. 전형적으로, 산술 코딩은 데이터 비트의 초 고압축을 실행하기 위해서 조건 확률을 사용한다.

<100> 따라서, 본 발명에서는 모션 벡터 추정의 사용에 있어서 단순하지만 강력한 화상간 디스플레이 시간의 인코딩 및 전송 방법과 이러한 화상간 디스플레이 시간의 디코딩 방법을 채용한다. 화상간 디스플레이 시간의 인코딩은 가변 길이 코딩이나 산술 코딩을 사용함으로써 매우 효과적으로 만들어 질 수 있다. 또한, 요구되는 가변성은 비디오 코덱의 필요에 일치하도록 선택될 수 있지만 그 이상은 될 수 없다.

<101> 전술된 내용은 멀티미디어 압축과 인코딩 시스템에서 가변 정확도 화상간 타이밍을 특징으로 하는 시스템에 관한 것이다. 이는 통상적인 지식을 가지고, 본 발명의 범위에서 벗어나지 않도록 본 발명 요소들의 조정함으로써 변형, 개량될 수 있다.

산업상 이용 가능성

<102> 따라서, 본 발명에 의하면 디지털 비디오 시스템에 있어서 모션 추정을 효과적으로 실행할 수 있다.

도면의 간단한 설명

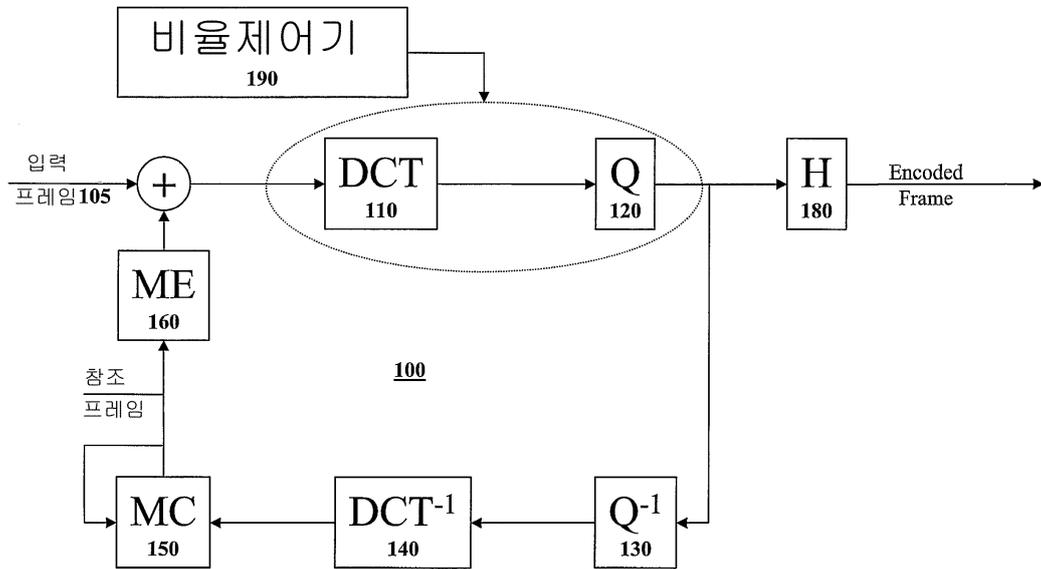
<12> 도 1은 하나의 실현 가능한 디지털 비디오 인코더 시스템의 고 레벨 블록도. 도 2는 다른 화상과 연결되는 화살표들이 모션 보상을 사용하도록 생성된 보조 화상 종속성을 나타내며, 디스플레이 된 화상 순으로 나열한 일련의 비디오 화상을 나타낸 도면.

<13> 도 3은 다른 화상과 연결되는 화살표들이 모션 보상 사용하도록 생성된 보조 화상 종속성을 나타내며, 화상의 실행된 전송 순서로 리스트된 도 2로부터 비디오 화상들을 나타낸 도면.

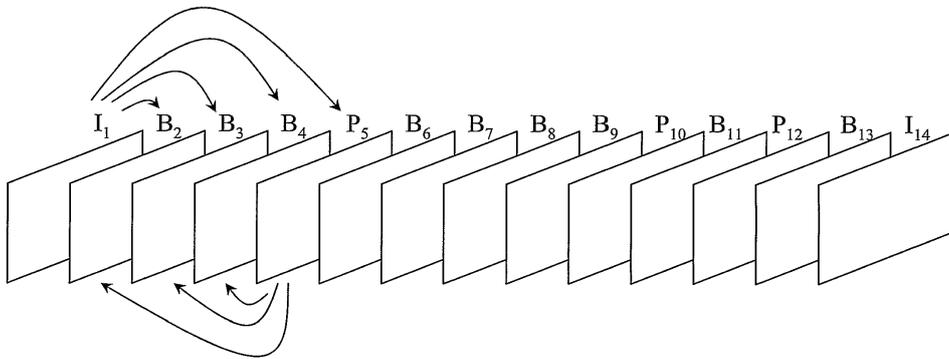
<14> 도 4는 각 참조가 2의 명승으로 선택된 비디오 화상 사이의 거리에서 일련의 비디오 화상들을 그래픽적으로 나타낸 도면.

도면

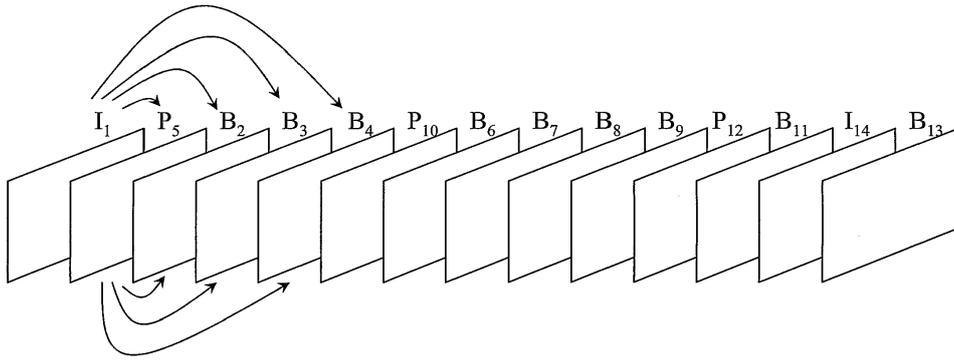
도면1



도면2



도면3



도면4

