



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2014년10월13일  
(11) 등록번호 10-1448440  
(24) 등록일자 2014년09월30일

(51) 국제특허분류(Int. Cl.)  
H04N 19/50 (2014.01)  
(21) 출원번호 10-2014-7002920(분할)  
(22) 출원일자(국제) 2009년05월05일  
심사청구일자 2014년02월04일  
(85) 번역문제출일자 2014년02월04일  
(65) 공개번호 10-2014-0026652  
(43) 공개일자 2014년03월05일  
(62) 원출원 특허 10-2010-7028344  
원출원일자(국제) 2009년05월05일  
심사청구일자 2012년03월08일  
(86) 국제출원번호 PCT/US2009/042785  
(87) 국제공개번호 WO 2010/008654  
국제공개일자 2010년01월21일  
(30) 우선권주장  
12/218,656 2008년07월16일 미국(US)  
(56) 선행기술조사문헌  
JP2007279800 A  
전체 청구항 수 : 총 8 항

(73) 특허권자  
소니 주식회사  
일본국 도쿄도 미나토쿠 코난 1-7-1  
소니 일렉트로닉스 인코포레이티드  
미국, 뉴저지 07656, 파크 리지, 원 소니 드라이브  
(72) 발명자  
나카자토 무네히로  
미국 94085 캘리포니아주 쉐니베일 산타 페 테라스 #113 250  
구 성율  
일본 141-0032 도쿄도 시나가와구 오사끼 아파트 #602 3-1-1  
(74) 대리인  
장수길, 이중희

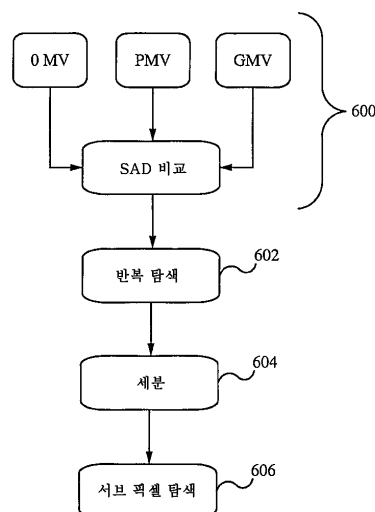
심사관 : 박상철

(54) 발명의 명칭 움직임 추정을 위해 반복 탐색 결과를 세분화하는 시스템 및 방법, 움직임 추정 시스템 및 기록 매체

(57) 요약

움직임 추정 반복 탐색을 위한 추론적 시작점 선택은 반복 시작 위치를 추론적으로 선택함으로써 정수-픽셀 움직임 추정 반복 탐색의 효율 및 품질을 개선한다. 시작 위치는 0 움직임 벡터, 예측된 움직임 벡터 및 전역 움직임 벡터(GMV)의 절대 차분합(SAD) 값을 비교하고, 최소 SAD 값을 갖는 위치를 선택함으로써 선택된다. 임계값을 갖는 세분화 스킵은 적합한 벡터 선택을 보장하도록 수차례의 비교를 행함으로써 움직임 추정 반복 탐색의 효율 및 품질을 개선한다. 이와 같이 개선된 움직임 추정 탐색의 응용은 화상의 손떨림 보정뿐 아니라, 움직임 벡터를 이용하는 기타 많은 응용을 포함한다.

대 표 도 - 도6



## 특허청구의 범위

### 청구항 1

움직임을 추정하는 시스템으로서, 상기 시스템은,

a. 애플리케이션을 저장하는 메모리로서, 상기 애플리케이션은 i. (1) 복수의 위치 각각마다 거리 값을 계산하고, (2) 복수의 위치 각각의 거리 값을 비교하고, (3) 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 것을 포함하여 시작 위치를 결정하고; ii. 시작 위치부터 시작하여 최소 거리 위치를 반복적으로 탐색하도록 구성되는, 메모리와;

b. 상기 애플리케이션을 처리하는 프로세서를 포함하고,

상기 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함하고,

상기 제1 위치는 제로(0) 움직임 벡터이고, 상기 제2 위치는 예측된 움직임 벡터이고, 상기 제3 위치는 전역 움직임 벡터인, 움직임 추정 시스템.

### 청구항 2

제1항에 있어서,

상기 거리 값은 절대 차분합 값인, 움직임 추정 시스템.

### 청구항 3

제1항에 있어서,

상기 반복적으로 탐색하는 것은,

- a. 카운트를 지정된 값으로 설정하는 것,
- b. 서브-영역의 서브-영역 거리 값을 계산하는 것,
- c. 상기 서브-영역에서 최소 영역 거리 값을 선택하는 것,
- d. 상기 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 것,
- e. 상기 최소 영역 거리 값과 상기 이전의 최소 영역 거리 값 중 더 작은 영역 거리 값을 보존하는 것,
- f. 상기 카운트를 감소시키는 것, 및
- g. 상기 카운트가 0이 될 때까지 상기 b 내지 f를 반복하는 것을 더 포함하는, 움직임 추정 시스템.

### 청구항 4

제1항에 있어서,

상기 프로세서 및 상기 메모리는 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된 장치 내에 포함되는, 움직임 추정 시스템.

### 청구항 5

제1항에 있어서,

상기 최소 거리 위치를 이용하여 화상의 손떨림을 보정하는 데 이용되는 움직임 벡터를 결정하는, 움직임 추정 시스템.

### 청구항 6

프로세서에 의해 처리되는 움직임을 추정하는 애플리케이션을 저장한 컴퓨터 판독가능 기록 매체로서, 상기 애플

플리케이션은,

- a. 반복 탐색을 시작하는 시작 위치를 결정하는 시작 위치 컴포넌트 및
  - b. 최소 거리 위치를 반복적으로 탐색하는 반복 탐색 컴포넌트를 포함하고,
- 상기 시작 위치를 결정하는 것은,

- a. 복수의 위치 각각 마다 거리 값을 계산하고,
- b. 상기 복수의 위치 각각의 거리 값을 비교하고,
- c. 상기 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 것을 포함하고,

상기 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함하고,

상기 제1 위치는 제로(0) 움직임 벡터이고, 상기 제2 위치는 예측된 움직임 벡터이고, 상기 제3 위치는 전역 움직임 벡터인, 컴퓨터 판독가능 기록 매체.

## 청구항 7

제6항에 있어서,

상기 반복적으로 탐색하는 것은,

- a. 카운트를 지정된 값으로 설정하는 것,
- b. 서브-영역의 서브-영역 거리 값을 계산하는 것,
- c. 상기 서브-영역에서 최소 영역 거리 값을 선택하는 것,
- d. 상기 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 것,
- e. 상기 최소 영역 거리 값과 상기 이전의 최소 영역 거리 값 중 더 작은 영역 거리 값을 보존하는 것,
- f. 상기 카운트를 감소시키는 것, 및
- g. 상기 카운트가 0이 될 때까지 상기 b 내지 f를 반복하는 것을 더 포함하는, 컴퓨터 판독가능 기록 매체.

## 청구항 8

제6항에 있어서,

상기 최소 거리 위치를 이용하여 화상의 손떨림을 보정하는 데 이용되는 움직임 벡터를 결정하는, 컴퓨터 판독가능 기록 매체.

## 명세서

### 기술 분야

[0001] 본 발명은 비디오 압축 분야에 관한 것이다. 보다 구체적으로는, 본 발명은 디지털 비디오 인코더에서의 움직임 추정(motion estimation) 개선에 관한 것이다.

### 배경 기술

[0002] 비디오 시퀀스(video sequence)는 통상 프레임이라 일컫는 다수의 픽처들(pictures)로 이루어진다. 연속되는 프레임들은 매우 유사하여, 프레임마다 많은 중복성(redundancy)을 포함한다. 비디오 데이터를 채널을 통해 효율적으로 전송하거나 메모리에 저장하기 전에, 대역폭 및 메모리 모두를 절약하기 위해 비디오 데이터를 압축시킨다. 이런 목적은 중복성을 제거하여 양호한 압축률을 얻는 데 있다. 첫 번째 비디오 압축 접근법은 특정 프레임에서 참조(reference) 프레임을 감산하여 상대적 차분(relative difference)을 발생시키는 것이다. 압축된 프레임은 참조 프레임보다 적은 정보를 포함한다. 상대적 차분은 동일한 품질을 유지하면서 낮은 비트-레이트로 부호화시킬 수 있다. 디코더는 상대적 차분을 참조 프레임에 부가시킴으로써 원래(original) 프레임을 재구성한다.

[0003] 더 정교한 접근법은, 비디오 시퀀스의 물체들(objects) 및 전체 장면(scene)의 움직임의 근사치를 계산하는 것

이다. 움직임은 비트-스트림에서 부호화되는 파라미터들로 묘사된다. 예측된 프레임의 픽셀들은 적절하게 변환된 참조 프레임의 픽셀들에 의해 근사치가 계산된다. 이런 접근법은 단순 감산법보다 개선된 예측 능력을 제공한다. 그러나, 움직임 모델의 파라미터들이 차지하는 비트-레이트가 너무 크게 되어선 안 된다.

[0004] 일반적으로, 비디오 압축은 Moving Picture Experts Group(MPEG) - 예를 들어, MPEG-1, MPEG-2 및 MPEG-4 - 중 오디오 및 비디오 압축을 위한 하나 이상의 표준을 비롯하여, 다수의 표준에 따라 수행된다. H.264로도 알려진 MPEG-4 파트 10 표준, 또는 AVC(advanced Video Coding)의 일부로서 추가적 향상(additional enhancements)이 행해졌다. MPEG 표준 하에서는, 비디오 데이터를 우선 부호화(예를 들어, 압축)시킨 후, 비디오 시스템의 인코더 측 상의 인코더 버퍼에 저장시킨다. 나중에, 부호화된 데이터가 비디오 시스템의 디코더 측으로 전송되며, 여기서 부호화된 데이터는 디코더 버퍼에 저장된 후에, 대응하는 픽처들을 볼 수 있도록 복호화된다.

[0005] H.264/AVC 프로젝트의 의도는, 이전의 표준들(예컨대, MPEG-2, H.263 또는 MPEG-4 파트 2)이 필요로 할 수 있는 것보다 상당히 낮은 비트-레이트로 양호한 비디오 품질을 제공할 수 있는 표준을 개발하려는 것이었다. 더욱이, 설계를 구현하기에 비현실적인 정도로 복잡성을 상당히 증가시키지 않고 이들 개선을 행하려는 것이 바람직하였다. 추가의 목적은, 낮은 비트-레이트 및 높은 비트-레이트와 저 해상도 비디오 및 고 해상도 비디오 모두에 사용될 수 있을 정도로, 표준을 광범위한 응용에 적용할 수 있게끔 할 수 있는 유연한 방식으로 이들을 변경하기 위한 것이었다. 또 다른 목적은 매우 광범위한 네트워크 및 시스템에 양호하게 동작할 수 있게 하는 것이었다.

[0006] H.264/AVC/MPEG-4 파트 10은 이전 표준들보다 훨씬 더 효과적으로 비디오를 압축가능하게 하며 광범위한 네트워크 환경에 적용하도록 더 유연성을 제공할 수 있게 하는 다수의 새로운 기능들(features)을 포함한다. 일부 기능들은, 이전에 부호화된 픽처를 참조로 이용하는 멀티-픽처 움직임 보상(motion compensation), 16×16만 큼 큰 블록 크기와 4×4만 큼 작은 블록 크기를 갖는 가변 블록-크기 움직임 보상(VBSMC), 1/2-픽셀 루마(half-pel luma) 샘플 예측을 이끌어 내기 위한 6-탭(tap) 필터링, 매크로블록 쌍(macroblock pair) 구조, 움직임 보상을 위한 1/4-픽셀 정도(precision), 가중 예측(weighted prediction), 루프 내 블록킹 제거 필터(in-loop deblocking filter), 정확히 매칭되는(exact-match) 정수 4×4 공간 블록 변환, 일차적(primary) 공간 변환의 "DC" 계수에 대해 수행되는 부차적 아다마르(secondary Hadamard) 변환 - 여기서 아다마르 변환은 고속 푸리에 변환과 유사함 -, "인트라(intra)" 코딩을 위해 이웃 블록의 에지로부터의 공간 예측, 문맥-적응적(context-adaptive) 2진 산술 코딩(CABAC), 문맥-적응적 가변-길이 코딩(CAVLC), CABAC 또는 CAVLC에 의해 코딩되지 않은 많은 수의 구문(syntax) 엘리먼트(element)를 위한 단순하고 고도로 구조화된 가변 길이 코딩(VLC) 기법 - 지수-골롬(Exponential-Golomb) 코딩으로 참조됨 -, 네트워크 추상화 계층(NAL) 정의, 스위칭 슬라이스(switching slices), 유연성 매크로블록 순서화(ordering), 중복성 슬라이스(RS), 보충 증강 정보(SEI) 및 비디오 유용성 정보(VUI), 보조 픽처들, 프레임 번호 매김 및 픽처 순서 카운트를 포함한다. 이들 기술 및 여러 다른 기술들은 H.264가 종래 표준들보다, 그리고 더 많은 상황과 더 많은 환경에서 상당히 양호하게 수행할 수 있도록 해준다. H.264는 통상적으로 비트-레이트의 절반 또는 훨씬 더 적은 비트-레이트로도 동일한 품질을 얻음으로써 MPEG-2 비디오보다 양호하게 수행된다.

[0007] MPEG는 동영상 및 연관된 오디오의 일반적인 코딩 시에 사용되며, 3 가지 유형의 일련의 부호화된 데이터 프레임으로 이루어진 압축된 비디오 비트-스트림을 생성한다. 이들 3 가지 유형의 데이터 프레임은 인트라 프레임(I-프레임 또는 I-픽처라 함), 양방향성 예측된 프레임(B-프레임 또는 B-픽처라 함) 및 순방향 예측된 프레임(P-프레임 또는 P-픽처라 함)이다. 이들 3 가지 유형의 프레임은 GOP(Group Of Pictures) 구조라 칭하는 특정 순서로 배열될 수 있다. I-프레임은 픽처를 재구성하는 데 필요한 모든 정보를 포함한다. I-프레임은 움직임 보상없이 보통 화상으로 부호화된다. 반면에, P-프레임은 이전 프레임으로부터의 정보를 이용하며, B-프레임은 이전 프레임, 후속 프레임 또는 이들 모두로부터의 정보를 이용하여 픽처를 재구성한다. 구체적으로, P-프레임은 선행하는 I-프레임 또는 바로 선행하는 P-프레임으로부터 예측된다.

[0008] 또한, 바로 후속되는 프레임으로부터 프레임들을 예측할 수 있다. 이와 같이 후속되는 프레임을 이용하기 위해, 예측된 프레임에 앞서 후속되는 프레임을 부호화해야 한다. 따라서, 부호화 순서가 실제 프레임 순서와 반드시 일치할 필요는 없다. 이러한 프레임들은 통상적으로 두 방향, 예를 들어, 예측된 프레임에 바로 선행하는 I-프레임 또는 P-프레임, 또는 예측된 프레임에 바로 후속되는 P-프레임으로부터 예측된다. 이러한 양방향으로 예측되는 프레임을 B-프레임이라 한다.

[0009] 가능성 있는 많은 GOP 구조가 있다. 일반적인 GOP 구조는 15 프레임 길이로서, 시퀀스 I\_BB\_P\_BB\_P\_BB\_P\_BB\_P\_BB\_를 갖는다. 유사한 12-프레임 시퀀스 또한 일반적이다. I-프레임은 공간적 중복성

에 대해 부호화하며, P-프레임 및 B-프레임은 시간적 중복성 및 공간적 중복성 모두에 대해 부호화한다. 비디오 스트림에서 인접한 프레임들은 흔히 상당 부분 상관되어 있으므로, P-프레임 및 B-프레임의 크기는 I-프레임 크기에 비해 아주 작은 비율이다. 그러나, 압축시킬 수 있는 프레임의 크기와 이러한 압축된 프레임을 부호화하는 데 필요한 처리 시간 및 자원 사이에는 상반 관계가 존재한다. GOP 구조에서 I, P 및 B-프레임의 비는 비디오 스트림의 본질(nature)과 출력 스트림에 대한 대역폭 제한에 의해 결정되지만, 부호화 시간이 또한 문제로 될 수 있다. 이것은 특히 컴퓨팅(computing) 자원이 제한된 라이브(live) 전송 및 실시간 환경에서 그러한데, 그 이유는 많은 B-프레임을 포함하는 스트림이 I-프레임-전용 파일(only file)보다 부호화하는데 훨씬 더 오래 걸리기 때문이다.

[0010] B-프레임 및 P-프레임은 일반적으로 현재 프레임과 이전 프레임, 후속되는 프레임 또는 이들 둘 모두 사이의 차분에 대한 차분 비트를 포함하여, 픽처 데이터를 저장하는 데 더 적은 비트를 필요로 한다. 그래서, B-프레임 및 P-프레임은 프레임 전체에 걸쳐 포함된 중복성 정보를 줄이는데 이용된다. 동작(operation) 시에, 디코더는 부호화된 B-프레임 또는 부호화된 P-프레임을 수신하고 이전 또는 후속되는 프레임을 이용하여 원래 프레임을 재구성한다. 순차적인 프레임들이 상당히 유사할 경우, 이 프로세스는 훨씬 더 용이하여 더 매끄러운 장면 전환(scene transition)이 이루어지는데, 이는 프레임들 간의 차분이 작기 때문이다.

[0011] 각 비디오 화상(image)은 하나의 휘도(Y)와 두 개의 색도(chrominance) 채널(색차 신호 Cb 및 Cr로도 칭함)로 분리된다. 휘도 및 색도 어레이의 블록들은 "매크로블록들"로 체계화되어 있으며, 이들 매크로블록이 프레임 내에서의 코딩의 기본 단위이다.

[0012] I-프레임의 경우, 실제 화상 데이터가 부호화 프로세스를 거친다. 그러나, P-프레임 및 B-프레임은 먼저 "움직임 보상" 프로세스를 거친다. 움직임 보상은, 이전 프레임의 각 매크로블록이 움직인 경우의 겹지에서 연속되는 프레임들 간의 차분을 묘사하는 방법이다. 이러한 기술은 흔히 비디오 압축을 위해 비디오 시퀀스의 시간적 중복성을 줄이는데 채용된다. P-프레임 및 B-프레임에서의 각 매크로블록은 "움직임 벡터(motion vector)"를 이용하여 인코더에 의해 선택되고 충분한 상관 관계를 갖는 이전 또는 다음 화상 내의 영역에 연관된다. 매크로블록을 그에 상관된 영역에 매핑시키는 움직임 벡터가 부호화되고, 이어서 두 영역 간의 차분이 부호화 프로세스를 거친다.

[0013] 통상의 비디오 코덱(codecs)은 움직임 보상된 예측을 이용하여 미처리된(raw) 입력 비디오 스트림을 효율적으로 부호화한다. 현재 프레임 내의 매크로블록은 이전 프레임 내에서의 변위된 매크로블록으로부터 예측된다. 원래의 매크로블록과 그 예측된 것 간의 차분을 압축하여 변위(움직임) 벡터와 함께 전송한다. 이 기술을 MPEG 표준에서 이용되는 접근법인, 인터-코딩(inter-coding)이라 한다.

[0014] 부호화 프로세스 내에서 가장 시간이 많이 걸리는 컴포넌트들 중 하나가 움직임 추정이다. 움직임 추정은 움직임 보상된 예측을 예측 오류의 변환 코딩과 결합하여 구현함으로써 비디오 신호의 비트-레이트를 감소시키는데 이용된다. 픽셀간 움직임 추정을 이용함에 의해 움직임 추정-관련된 에일리어싱(aliasing)을 피할 수 없으며, 이 에일리어싱이 예측 효율을 떨어뜨린다. 효율성 저하 문제를 해결하기 위해, 에일리어싱의 영향을 감소시키는 1/2-픽셀 보간법(interpolation) 및 1/4-픽셀 보간법을 적용시킨다. 1/4-픽셀 정확도(accuracy)를 갖는 움직임 벡터를 추정하기 위해, 일반적으로 3-단계 탐색(a three step search)을 이용한다. 제1 단계에서, 지정된 탐색 범위 내에서 각각의 정수 픽셀에 대해 움직임 추정을 적용시켜 최적의 매칭(best match)을 찾는다. 그 다음, 제2 단계에서, 선택된 정수-픽셀 움직임 벡터 주변의 8개 절반-픽셀점을 검사하여 최적의 절반-픽셀 매칭점을 찾는다. 최종적으로, 제3 단계에서, 선택된 절반-픽셀 움직임 벡터 주변의 8개 1/4-픽셀 지점을 검사하여, 최적의 매칭점을 최종 움직임 벡터로서 선택한다. 움직임 추정의 복잡성을 고려해 볼 때, 정수-픽셀 움직임 추정 시에 완전(full)-탐색을 이용할 경우, 정수-픽셀 움직임 추정은 움직임 추정의 대부분을 차지한다. 그러나, 고속(fast) 정수 움직임 추정 알고리즘을 이용할 경우에는, 10개 미만의 탐색점을 검사하는 것으로 정수-픽셀 움직임 벡터를 찾을 수 있다. 결과적으로, 절반-픽셀 움직임 벡터 및 1/4-픽셀 움직임 벡터를 탐색하는 계산 복잡성이 지배적이 된다.

## 발명의 내용

### 해결하려는 과제

[0015] 움직임 추정 반복 탐색을 위한 추론적(speculative) 시작점 선택은, 반복(iteration)의 시작 위치를 선택하는 것에 의해 정수-픽셀 움직임 추정 반복 탐색의 효율성 및 품질을 개선한다. 시작 위치는 제로(0) 움직임 벡터, 예측된 움직임 벡터 및 전역(global) 움직임 벡터(GMV)의 절대 차분합(Sum of Absolute Differences; SAD) 값



을 비교하여 최소 SAD 값을 갖는 위치를 선택함에 의해 선택된다. 적합한 움직임 벡터의 선택을 보장하기 위해 수차례의 비교를 수행하는 것에 의해 임계값을 갖는 세분화 스킴(refinement scheme)은 움직임 추정 반복 탐색의 효율성 및 품질을 개선한다. 이러한 개선된 움직임 추정 탐색의 응용으로는, 화상의 손떨림 보정(stabilizing)뿐만 아니라, 움직임 벡터를 이용하는 다른 많은 응용이 포함된다.

### 과제의 해결 수단

[0016] 본 발명의 한 측면에서, 움직임 추정을 위해 컴퓨팅 장치를 이용하여 반복 탐색 결과를 세분화하는 방법은, 복수의 위치 각각마다 거리 값(distance value)을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 최소 거리 위치의 최소 거리 값에 움직임 벡터 비용(cost)을 가산하여 총 비용을 형성하는 단계와; 총 비용과 예측 움직임 벡터 거리 값을 비교하여 더 작은 값을 결정하는 단계와; 총 비용과 예측 움직임 벡터 거리 값 중 더 작은 값을 선택하는 단계를 포함한다. 일부 실시예에서, 거리 값은 절대 차분합 값이다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 움직임 벡터 비용은 예측된 움직임 벡터로부터의 거리에 기초하여 계산된다. 더 작은 값은 더 세분화되어 화상의 손떨림 보정을 행한다. 반복적으로 탐색하는 단계는 카운트를 지정된 값으로 설정하는 단계와, 서브-영역의 서브-영역 거리 값을 계산하는 단계와, 서브-영역에서 최소 영역 거리 값을 선택하는 단계와, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계와, 최소 영역 거리 값과 이전의 최소 영역 거리 값 중 더 작은 영역 거리 값을 보존하는 단계와, 카운트를 감소시키는 단계와, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 컴퓨팅 장치는 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된다.

[0017] 본 발명의 다른 측면에서, 움직임 추정을 위해 반복 탐색 결과를 세분화하는 시스템은 애플리케이션을 저장하는 메모리와, 애플리케이션을 처리하는 프로세서를 포함하며, 상기 애플리케이션은 복수의 위치 각각마다 거리 값을 계산하고, 복수의 위치 각각의 거리 값을 비교하고, 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 것을 포함하여 시작 위치를 결정하고; 시작 위치부터 시작하여 최소 거리 위치를 반복적으로 탐색하고; 최소 거리 위치의 최소 거리 값에 움직임 벡터 비용(cost)을 가산하여 총 비용을 형성하고; 총 비용과 예측 움직임 벡터 거리 값을 비교하여 더 작은 값을 결정하고; 총 비용과 예측 움직임 벡터 거리 값 중 더 작은 값을 선택하도록 구성된다. 일부 실시예에서, 거리 값은 절대 차분합 값이다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 움직임 벡터 비용은 예측된 움직임 벡터로부터의 거리에 기초하여 계산된다. 반복적으로 탐색하는 것은 카운트를 지정된 값으로 설정하는 단계, 서브-영역의 서브-영역 거리 값을 계산하는 단계, 서브-영역에서 최소 영역 거리 값을 선택하는 단계, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계, 최소 영역 거리 값과 이전의 최소 영역 거리 값 중 더 작은 영역 거리 값을 보존하는 단계, 카운트를 감소시키는 단계, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 더 작은 값을 더 세분화하여 이용함으로써 화상의 손떨림 보정을 행한다. 프로세서 및 메모리는 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된 장치 내에 포함된다.

[0018] 본 발명의 또 다른 측면에서, 움직임 추정을 위해 반복 탐색 결과를 세분화하도록 프로세서에 의해 처리되는 애플리케이션은, 반복 탐색을 시작하기 위해 최소 시작 위치를 결정하는 시작 위치 컴포넌트와, 최소 거리 위치를 반복적으로 탐색하는 반복 탐색 컴포넌트와, 총 비용을 예측된 움직임 벡터 거리 값과 비교하는 비교 컴포넌트를 포함하며, 여기서 총 비용은 최소 거리 위치의 최소 거리 값 및 움직임 벡터 비용을 포함한다. 최소 시작 위치를 결정하는 것은, 복수의 위치 각각마다 거리 값을 계산하고, 복수의 위치 각각의 거리 값을 비교하고, 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 것을 포함한다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 반복적으로 탐색하는 것은 카운트를 지정된 값으로 설정하는 단계, 서브-영역의 서브-영역 거리 값을 계산하는 단계, 서브-영역에서 최소 영역 거리 값을 선택하는 단계, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계, 최소 영역 거리 값과 이전의 최소 영역 거리 값 중

더 작은 영역 거리 값을 보존하는 단계, 카운트를 감소시키는 단계, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 움직임 벡터 비용은 예측된 움직임 벡터로부터의 거리에 기초하여 계산된다. 최적의 거리 위치를 이용하여 화상의 손떨림 보정을 행하는 데 이용되는 적절한 움직임 벡터를 결정한다. 상기 애플리케이션은 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된 장치 내에 포함된다.

[0019] 본 발명의 또 다른 측면에서, 움직임 추정을 위해 컴퓨팅 장치를 이용하여 반복 탐색 결과를 세분화하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 최소 거리 위치의 최소 거리 값에 움직임 벡터 비용을 가산하여 총 비용을 형성하는 단계와; 전역 움직임 벡터가 임계값 미만인지를 판단하는 단계를 포함하며, 여기서, 전역 움직임 벡터가 임계값 미만이면, 총 비용을 예측 움직임 벡터 거리 값과 비교하여 더 작은 값을 결정하는 단계와 세분화를 위해 총 비용과 예측 움직임 벡터 거리 값 중 더 작은 값을 선택하는 단계를 포함한, 추가의 단계를 택하고, 전역 움직임 벡터가 임계값 이상이면, 최소 거리 위치의 최소 거리 값이 세분화용으로 선택된다. 일부 실시예에서, 거리 값은 절대 차분합 값이다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 움직임 벡터 비용은 예측된 움직임 벡터로부터의 거리에 기초하여 계산된다. 이 방법은 화상의 손떨림 보정을 행한다. 반복적으로 탐색하는 단계는 카운트를 지정된 값으로 설정하는 단계와, 서브-영역의 서브-영역 거리 값을 계산하는 단계와, 서브-영역에서 최소 영역 거리 값을 선택하는 단계와, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계와, 최소 영역 거리 값과 이전의 최소 영역 거리 값 중 더 작은 거리 값을 보존하는 단계와, 카운트를 감소시키는 단계와, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 컴퓨팅 장치는 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된다.

[0020] 본 발명의 다른 측면에서, 움직임 추정을 위해 반복 탐색 결과를 세분화하는 시스템은 애플리케이션을 저장하는 메모리와, 애플리케이션을 처리하는 프로세서를 포함하며, 상기 애플리케이션은 복수의 위치 각각마다 거리 값을 계산하고, 복수의 위치 각각의 거리 값을 비교하고, 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 것을 포함하여 시작 위치를 결정하고; 시작 위치부터 시작하여 최소 거리 위치를 반복적으로 탐색하고; 최소 거리 위치의 최소 거리 값에 움직임 벡터 비용(cost)을 부가하여 총 비용을 형성하고; 전역 움직임 벡터가 임계값 미만인지를 판단하도록 구성되며, 여기서, 전역 움직임 벡터가 임계값 미만이면, 총 비용을 예측 움직임 벡터 거리 값과 비교하여 더 작은 값을 결정하는 단계와 세분화를 위해 총 비용과 예측 움직임 벡터 거리 값 중 더 작은 값을 선택하는 단계를 포함한, 추가의 단계를 택하고, 전역 움직임 벡터가 임계값 이상이면, 최소 거리 위치의 최소 거리 값이 세분화용으로 선택된다. 일부 실시예에서, 거리 값은 절대 차분합 값이다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 움직임 벡터 비용은 예측된 움직임 벡터로부터의 거리에 기초하여 계산된다. 반복적으로 탐색하는 것은 카운트를 지정된 값으로 설정하는 단계, 서브-영역의 서브-영역 거리 값을 계산하는 단계, 서브-영역에서 최소 영역 거리 값을 선택하는 단계, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계, 최소 영역 거리 값과 이전의 최소 영역 거리 값을 보유하는 단계, 카운트를 감소시키는 단계, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 이 시스템은 화상의 손떨림 보정을 행한다. 프로세서 및 메모리는 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된 장치 내에 포함된다.

[0021] 본 발명의 또 다른 측면에서, 움직임 추정을 위해 반복 탐색 결과를 세분화하도록 프로세서에 의해 처리되는 애플리케이션은, 반복 탐색을 시작하기 위해 최소 시작 위치를 결정하는 시작 위치 컴포넌트와, 최소 거리 위치를 반복적으로 탐색하는 반복 탐색 컴포넌트와, 총 비용을 예측된 움직임 벡터 거리 값과 비교하는 비교 컴포넌트-상기 총 비용은 최소 거리 위치의 최소 거리 값 및 움직임 벡터 비용을 포함함-와, 전역 움직임 벡터를 임계값과 비교하여 비교 컴포넌트를 실행할지 또는 반복 탐색 컴포넌트로부터의 결과를 이용할지를 결정하는 임계값 컴포넌트를 포함한다. 최소 시작 위치를 결정하는 것은, 복수의 위치 각각마다 거리 값을 계산하고, 복수의

위치 각각의 거리 값을 비교하고, 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 것을 포함한다. 일부 실시예에서, 거리 값은 절대 차분합 값이다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 반복적으로 탐색하는 것은 카운트를 지정된 값으로 설정하는 단계, 서브-영역의 서브-영역 거리 값을 계산하는 단계, 서브-영역에서 최소 영역 거리 값을 선택하는 단계, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계, 최소 영역 거리 값과 이전의 최소 영역 거리 값 중 더 작은 영역 거리 값을 보존하는 단계, 카운트를 감소시키는 단계, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 움직임 벡터 비용은 예측된 움직임 벡터로부터의 거리에 기초하여 계산된다. 상기 애플리케이션은 화상의 손떨림 보정을 행한다. 상기 애플리케이션은 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된 장치 내에 포함된다.

[0022] 본 발명의 또 다른 측면에서, 움직임 추정 반복 탐색을 위해 컴퓨팅 장치를 이용하여 시작 위치를 선택하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계와, 복수의 위치 각각의 거리 값을 비교하는 단계와, 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함한다. 일부 실시예에서, 거리 값은 절대 차분합 값이다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 컴퓨팅 장치는 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된다. 최소 시작 거리 위치를 화상의 손떨림 보정을 행하는 움직임 추정 반복 탐색을 위한 시작 위치로서 이용한다.

[0023] 본 발명의 또 다른 측면에서, 컴퓨팅 장치를 이용하여 움직임을 추정하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계를 포함한다. 반복적으로 탐색하는 단계는 카운트를 지정된 값으로 설정하는 단계, 서브-영역의 서브-영역 거리 값을 계산하는 단계, 서브-영역에서 최소 영역 거리 값을 선택하는 단계, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계, 최소 영역 거리 값과 이전의 최소 영역 거리 값 중 더 작은 영역 거리 값을 보존하는 단계, 카운트를 감소시키는 단계, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 일부 실시예에서, 거리 값은 절대 차분합 값이다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 컴퓨팅 장치는 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된다. 최소 거리 위치를 이용하여 화상의 손떨림 보정을 행하는 데 이용되는 적절한 움직임 벡터를 결정한다.

[0024] 본 발명의 또 다른 측면에서, 움직임을 추정하는 시스템은 애플리케이션을 저장하는 메모리 및 애플리케이션을 처리하는 프로세서를 포함하며, 상기 애플리케이션은 복수의 위치 각각마다 거리 값을 계산하고, 복수의 위치 각각의 거리 값을 비교하고, 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 것을 포함하여 시작 위치를 결정하고; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하도록 구성된다. 반복적으로 탐색하는 것은 카운트를 지정된 값으로 설정하는 단계, 서브-영역의 서브-영역 거리 값을 계산하는 단계, 서브-영역에서 최소 영역 거리 값을 선택하는 단계, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계, 최소 영역 거리 값과 이전의 최소 영역 거리 값 중 더 작은 영역 거리 값을 보존하는 단계, 카운트를 감소시키는 단계, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 일부 실시예에서, 거리 값은 절대 차분합 값이다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 프로세서 및 메모리는 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된 장치 내에 포함된다. 최소 거리 위치를 이용하여 화상의 손떨림 보정을 행하는 데 이용되는 적절한 움직임 벡터를 결정한다.



- [0025] 본 발명의 또 다른 측면에서, 프로세서에 의해 처리되는 움직임을 추정하는 애플리케이션은, 반복 탐색을 시작하는 최적의 시작 위치를 결정하는 시작 위치 컴포넌트 및 최소 거리 위치를 반복적으로 탐색하는 반복 탐색 컴포넌트를 포함한다. 반복적으로 탐색하는 것은 카운트를 지정된 값으로 설정하는 단계, 서브-영역의 서브-영역 거리 값을 계산하는 단계, 서브-영역에서 최소 영역 거리 값을 선택하는 단계, 최소 영역 거리 값을 이전의 최소 영역 거리 값과 비교하는 단계, 최소 영역 거리 값과 이전의 최소 영역 거리 값 중 더 작은 영역 거리 값을 보존하는 단계, 카운트를 감소시키는 단계, 카운트가 0이 될 때까지 상기 단계들을 반복하는 단계를 더 포함한다. 최적의 시작 위치를 결정하는 것은, 복수의 위치 각각마다 거리 값을 계산하고, 복수의 위치 각각의 거리 값을 비교하고, 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 것을 포함한다. 복수의 위치는 제1 위치, 제2 위치 및 제3 위치를 포함한다. 제1 위치는 제로(0) 움직임 벡터이고, 제2 위치는 예측된 움직임 벡터이고, 제3 위치는 전역 움직임 벡터이다. 상기 애플리케이션은 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod® 및 가정용 오락 시스템으로 이루어진 그룹에서 선택된 장치 내에 포함된다. 최소 거리 위치를 이용하여 화상의 손떨림을 보정하는 데 이용되는 적절한 움직임 벡터를 결정한다.
- [0026] 본 발명의 또 다른 측면에서, 컴퓨팅 장치를 이용하여 움직임 추정을 개선하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 지역적 최소값(local minimum)을 피하는 단계를 포함하며, 상기 지역적 최소값을 피하는 단계는 이전 위치를 결정하는 단계와, 새로운 위치를 결정하는 단계와, 이전 위치와 새로운 위치를 비교하는 단계와, 이전 위치와 새로운 위치 비교에 기초하여, 새로운 탐색 중심점(center)을 선택하는 단계를 포함하며, 여기서 새로운 탐색 중심점은 새로운 위치가 이전 위치와 반대 방향인 경우에는 이전 위치에 기초하고, 새로운 위치가 이전 위치와 반대 방향이 아닌 경우에는 새로운 위치에 기초한다.
- [0027] 본 발명의 또 다른 측면에서, 컴퓨팅 장치를 이용하여 움직임 추정을 개선하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 다음 탐색 영역의 중심점을 결정하는 단계를 포함하며, 다음 탐색 영역의 중심점을 결정하는 단계는 최소 거리 탐색 값을 갖는 탐색 영역의 위치를 결정하는 단계와, 최소 거리 탐색 값을 갖는 위치에 기초하여 다음 탐색 영역의 중심점에 대한 오프셋을 결정하는 단계와, 상기 오프셋에 기초하여 다음 탐색 영역의 중심점을 선택하는 단계를 포함한다.
- [0028] 본 발명의 또 다른 측면에서, 컴퓨팅 장치를 이용하여 움직임 추정을 개선하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 다음 탐색 영역의 중심점을 결정하는 단계- 상기 다음 탐색 영역의 중심점을 결정하는 단계는 최소 거리 탐색 값을 갖는 탐색 영역의 위치를 결정하는 단계와, 최소 거리 탐색 값을 갖는 위치에 기초하여 다음 탐색 영역의 중심점에 대한 오프셋을 결정하는 단계와, 상기 오프셋에 기초하여 다음 탐색 영역의 중심점을 선택하는 단계를 포함함 -와; 지역적 최소값을 피하는 단계를 포함하며, 상기 지역적 최소값을 피하는 단계는 이전 위치를 결정하는 단계와, 새로운 위치를 결정하는 단계와, 이전 위치와 새로운 위치를 비교하는 단계와, 이전 위치와 새로운 위치 비교에 기초하여, 새로운 탐색 중심점을 선택하는 단계를 포함하며, 여기서 새로운 탐색 중심점은 새로운 위치가 이전 위치와 반대 방향인 경우에는 이전 위치에 기초하고, 새로운 위치가 이전 위치와 반대 방향이 아닌 경우에는 새로운 위치에 기초한다.
- [0029] 본 발명의 또 다른 측면에서, 컴퓨팅 장치를 이용하여 움직임 추정을 개선하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 최소 거리 위치의 최소 거리 값에 움직임 벡터 비용을 가산하여 총 비용을 형성하는 단계와; 전역 움직임 벡터가 임계값 미만인지를 판단하는 단계- 여기서, 전역 움직임 벡터가 임계값 미만이면, 총 비용을 예측 움직임 벡터 거리 값과 비교하여 더 작은 값을 결정하는 단계와 세분화용으로 총 비용과 예측 움직임 벡터 거리 값 중 더 작은 값을 선택하는 단계를 포함함, 추가의 단계를 택하고, 전역 움직임 벡터가 임계값 이상이면, 최소 거리 위치의 최소 거리 값이 세분화용으로 선택됨-와; 다음 탐색 영역의 중심점을 결정하는 단계를 포함하며, 다음 탐색 영역의 중심점을 결정하는 단계는 최소 거리 탐색 값을 갖는

탐색 영역의 위치를 결정하는 단계와, 최소 거리 탐색 값을 갖는 위치에 기초하여 다음 탐색 영역의 중심점에 대한 오프셋을 결정하는 단계와, 상기 오프셋에 기초하여 다음 탐색 영역의 중심점을 선택하는 단계를 포함한다.

[0030] 본 발명의 또 다른 측면에서, 컴퓨팅 장치를 이용하여 움직임 추정을 개선하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 최소 거리 위치의 최소 거리 값에 움직임 벡터 비용을 가산하여 총 비용을 형성하는 단계와; 전역 움직임 벡터가 임계값 미만인지를 판단하는 단계- 여기서, 전역 움직임 벡터가 임계값 미만이면, 총 비용을 예측 움직임 벡터 거리 값과 비교하여 더 작은 값을 결정하는 단계와 세분화용으로 총 비용과 예측 움직임 벡터 거리 값 중 더 작은 값을 선택하는 단계를 포함한, 추가의 단계를 택하고, 전역 움직임 벡터가 임계값 이상이면, 최소 거리 위치의 최소 거리 값이 세분화용으로 선택됨-와; 지역적 최소값을 피하는 단계를 포함하며, 상기 지역적 최소값을 피하는 단계는 이전 위치를 결정하는 단계와, 새로운 위치를 결정하는 단계와, 이전 위치와 새로운 위치를 비교하는 단계와, 이전 위치와 새로운 위치 비교에 기초하여, 새로운 탐색 중심점을 선택하는 단계를 포함하며, 여기서 새로운 탐색 중심점은 새로운 위치가 이전 위치와 반대 방향인 경우에는 이전 위치에 기초하고, 새로운 위치가 이전 위치와 반대 방향이 아닌 경우에는 새로운 위치에 기초한다.

[0031] 본 발명의 또 다른 측면에서, 컴퓨팅 장치를 이용하여 움직임 추정을 개선하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 최소 거리 위치의 최소 거리 값에 움직임 벡터 비용을 가산하여 총 비용을 형성하는 단계와; 전역 움직임 벡터가 임계값 미만인지를 판단하는 단계- 여기서, 전역 움직임 벡터가 임계값 미만이면, 총 비용을 예측 움직임 벡터 거리 값과 비교하여 더 작은 값을 결정하는 단계와 세분화용으로 총 비용과 예측 움직임 벡터 거리 값 중 더 작은 값을 선택하는 단계를 포함한, 추가의 단계를 택하고, 전역 움직임 벡터가 임계값 이상이면, 최소 거리 위치의 최소 거리 값이 세분화용으로 선택됨-와; 다음 탐색 영역의 중심점을 결정하는 단계- 다음 탐색 영역의 중심점을 결정하는 단계는 최소 거리 탐색 값을 갖는 탐색 영역의 위치를 결정하는 단계와, 최소 거리 탐색 값을 갖는 위치에 기초하여 다음 탐색 영역의 중심점에 대한 오프셋을 결정하는 단계와, 상기 오프셋에 기초하여 다음 탐색 영역의 중심점을 선택하는 단계를 포함함-와; 지역적 최소값을 피하는 단계를 포함하며, 상기 지역적 최소값을 피하는 단계는 이전 위치를 결정하는 단계와, 새로운 위치를 결정하는 단계와, 이전 위치와 새로운 위치를 비교하는 단계와, 이전 위치와 새로운 위치 비교에 기초하여, 새로운 탐색 중심점을 선택하는 단계를 포함하며, 여기서 새로운 탐색 중심점은 새로운 위치가 이전 위치와 반대 방향인 경우에는 이전 위치에 기초하고, 새로운 위치가 이전 위치와 반대 방향이 아닌 경우에는 새로운 위치에 기초한다.

[0032] 본 발명의 또 다른 측면에서, 컴퓨팅 장치를 이용하여 움직임 추정을 개선하는 방법은, 복수의 위치 각각마다 거리 값을 계산하는 단계, 복수의 위치 각각의 거리 값을 비교하는 단계 및 복수의 위치 각각의 거리 값으로부터 최소 시작 거리 위치를 선택하는 단계를 포함하여 시작 위치를 결정하는 단계와; 시작 위치로부터 시작하여 최소 거리 위치를 반복적으로 탐색하는 단계와; 최소 거리 위치의 최소 거리 값에 움직임 벡터 비용을 가산하여 총 비용을 형성하는 단계와; 전역 움직임 벡터가 임계값 미만인지를 판단하는 단계- 여기서, 전역 움직임 벡터가 임계값 미만이면, 총 비용을 예측 움직임 벡터 거리 값과 비교하여 더 작은 값을 결정하는 단계와 세분화용으로 총 비용과 예측 움직임 벡터 거리 값 중 더 작은 값을 선택하는 단계를 포함한, 추가의 단계를 택하고, 전역 움직임 벡터가 임계값 이상이면, 최소 거리 위치의 최소 거리 값이 세분화용으로 선택됨-와; 다음 탐색 영역의 중심점을 결정하는 단계- 다음 탐색 영역의 중심점을 결정하는 단계는 최소 거리 탐색 값을 갖는 탐색 영역의 위치를 결정하는 단계와, 최소 거리 탐색 값을 갖는 위치에 기초하여 다음 탐색 영역의 중심점에 대한 오프셋을 결정하는 단계와, 상기 오프셋에 기초하여 다음 탐색 영역의 중심점을 선택하는 단계를 포함함-와; 지역적 최소값을 피하는 단계와; 반복적으로 탐색하는 단계를 조기에(early) 종료하는 단계를 포함하며, 상기 지역적 최소값을 피하는 단계는 이전 위치를 결정하는 단계와, 새로운 위치를 결정하는 단계와, 이전 위치와 새로운 위치를 비교하는 단계와, 이전 위치와 새로운 위치 비교에 기초하여, 새로운 탐색 중심점을 선택하는 단계를 포함하며, 여기서 새로운 탐색 중심점은 새로운 위치가 이전 위치와 반대 방향인 경우에는 이전 위치에 기초하고, 새로운 위치가 이전 위치와 반대 방향이 아닌 경우에는 새로운 위치에 기초하며, 상기 반복적으로 탐색하는 단계를 조기에 종료하는 단계는 서브-영역의 서브-영역 거리 값을 계산하는 단계와, 서브-영역의 최소 거리 값을 결정하는 단계와, 최소 거리 값을 임계값과 비교하는 단계와, 최소 거리 값이 임계값 미만이면 조기에 종료하는

단계와, 최소 거리 값이 임계값 이상이면 카운트가 제로가 될 때까지 상기 단계들을 반복하는 단계를 포함한다.

### 도면의 간단한 설명

[0033] 도 1은 본 발명에 따른 비디오 코딩 계층(coding layer)의 블록도.

도 2는 본 발명에 따른 움직임 추정 흐름을 예시한 도면.

도 3은 본 발명에 따른 예시적인 탐색을 도시한 도면.

도 4는 본 발명에 따른 반복 탐색의 흐름도.

도 5는 본 발명에 따라 예시적인 탐색 위치 및 절대 차분 합(SAD)을 계산하는 데 이용되는 픽셀들을 도시한 도면.

도 6은 본 발명에 따른 움직임을 추정하기 위해 추론적 시작점을 구현하는 흐름도.

도 7은 본 발명에 따른 예시적인 컴퓨팅 장치의 블록도.

도 8은 본 발명에 따른 나이브(naive) 다음 단계 선택을 예시한 도면.

도 9는 본 발명에 따른 지역적 최소값으로 고정되어 있는 탐색을 예시한 도면.

도 10은 본 발명에 따른 반대 방향으로 되돌아가는 나이브 다음 단계 선택을 예시한 도면.

도 11은 본 발명에 따른 지역적 최소값을 벗어나 이동하는 탐색을 예시하는 도면.

도 12는 본 발명에 따른 위치 행렬을 도시한 도면.

도 13은 본 발명에 따른 반대 위치를 음영으로 나타낸 위치 행렬을 도시한 도면.

도 14는 본 발명에 따른 새로운 위치 및 금지된 반대 위치를 도시한 도면.

도 15는 본 발명에 따른 나이브 탐색의 흐름도.

도 16은 본 발명에 따른 지역적 최소값 회피 탐색의 흐름도.

도 17은 본 발명에 따른 다음 위치 선택기의 흐름도.

도 18 내지 도 20은 본 발명에 따른 3×3 탐색 영역에 대한 현재 탐색 영역의 최소 SAD를 갖는 위치에 기초한 다음 탐색 영역을 예시하는 도면.

도 21은 본 발명에 따른 다음 위치를 선택하는 방법의 흐름도.

도 22는 본 발명에 따른 최소 SAD를 갖는 위치에 기초하여 다음 탐색 영역의 중심점을 결정하는 예시적인 참조표를 예시한 도면.

도 23은 본 발명에 따른 예측된 움직임 벡터의 SAD 및 움직임 벡터 비용에 기초한 세분화를 이용한 반복 탐색의 흐름도.

도 24는 본 발명에 따른 인접 매크로블록들 및 그들의 움직임 벡터들을 갖는 현재 매크로블록을 예시한 도면.

도 25는 본 발명에 따른 예측된 움직임 벡터의 SAD, 임계값 및 움직임 벡터 비용에 기초한 세분화를 이용한 반복 탐색의 흐름도.

도 26은 본 발명에 따른 조기 종료 스킵에 의한 반복 탐색의 흐름도.

### 발명을 실시하기 위한 구체적인 내용

[0034] 도 1은 매크로블록의 비디오 코딩 계층(100)의 블록도이다. 비디오 코딩 계층(100)(예컨대, 인코더)은 변환 코딩과 함께 시간적 및 공간적 예측의 결합을 포함한다. 입력 비디오(102)가 수신되어 복수의 블록으로 분할된다. 시퀀스의 제1 픽처는 통상적으로 자체에 포함된 정보만을 이용하여 "인트라(intra)" 코딩된다. 다음에, 인트라 프레임 내 블록의 각 부분(part)은 인트라 예측 모듈(110)에서 이전에 코딩된 블록의 공간적으로 이웃하는 샘플들을 이용하여 예측된다. 부호화 프로세스는 인트라 예측에 이웃하는 어느 샘플들을 활용할 것인지 또한 그들의 이용 방법을 선택한다. 이 프로세스는 인코더(100)에서뿐 아니라, 로컬 디코더(118)에서도 행해진다. 시퀀스의 나머지 픽처의 경우에는, 통상적으로 "인터(inter)" 코딩이 이용된다. 인터 코딩은 이전에

복호화된 다른 픽처로부터 움직임 보상(112)을 구현한다. 움직임 추정 모듈(114)에서 인터 예측/움직임 추정을 위한 부호화 프로세스는 움직임 데이터를 선택하는 것과, 기준(reference) 픽처 및 블록의 모든 샘플에 적용되는 공간적 변위를 결정하는 것을 포함한다. 움직임 데이터는 인코더(100) 및 로컬 디코더(118)에 의해 이용되는 부차적인(side) 정보로서 전달된다.

[0035] 원래 블록과 예측된 블록 간의 차이는 예측 잔차(residual of prediction)로 참조된다. 이 잔차를 변환시키고, 변환 및 스케일링 양자화 모듈(104)에서 변환 계수를 스케일링하고 양자화시킨다. 변환 계수의 양자화를 위해, 스칼라 양자화를 이용한다. 정수 변환을 이용하여 각각의 블록을 변환하고, 엔트로피-코딩(entropy-coding) 방법을 이용하여 변환 계수를 양자화 및 전달한다. 엔트로피 인코더(116)는 양자화된 변환 계수를 제외한 모든 엘리먼트에 대해 설정된 코드워드(codeword)를 이용한다. 양자화된 변환 계수의 경우, 문맥 적응 가변 길이 코딩(CAVLC) 또는 문맥 적응 2진 산술 코딩(CABAC)이 이용된다. 필터링의 강도를 화상의 블록 현상(blockiness)을 감소시키도록 제어하는 블록킹 제거 필터(108)가 구현되어 있다.

[0036] 인코더(100)는 또한 로컬 디코더(118)를 포함하여 다음 블록에 대한 예측 기준을 생성한다. 양자화된 변환 계수는 디코딩된 예측 잔차를 제공하는 인코더 측과 동일한 방법으로 역스케일링 및 역변환된다(106). 디코딩된 예측 잔차가 예측값에 가산되고, 디코딩된 비디오를 출력으로서 제공하는 블록킹 제거 필터(108)에 결합물(combination)이 전달된다. 최종적으로, 엔트로피 인코더(116)는 원래 입력된 비디오(102)의 압축된 비디오 비트(120)를 생성한다.

[0037] 비디오 인코더의 움직임 추정(ME)에서, 가장 소모적인 계산은 예측된 픽처와 원래 픽처 간의 절대 차분합(SAD)이다. 특히, 정수 픽셀 탐색의 SAD 계산이 두드러진다. 그러므로, 정수 픽셀 탐색에서 SAD 계산의 수를 줄이는 것이 하드웨어 크기를 줄이는데, 이로써 비용을 줄이는데 상당한 역할을 한다. 도 2는 픽처를 취해 정수 픽셀 탐색, 1/2 픽셀 탐색 및 1/4 픽셀 탐색을 수행하여 움직임 벡터를 결정하는 움직임 추정 흐름(200)을 도시한다.

[0038] SAD 계산을 줄이기 위한 일반적인 접근법은 SAD 계산 전에 기준 픽처와 원래 픽처를 서브-샘플링하여, ME 하드웨어가 적은 수의 SAD 계산으로 동일한 탐색 범위를 탐색할 수 있게 하는 것이다. 서브-샘플링된 탐색 범위 내에서도, 탐색은 여전히 소모적이다. 탐색이 대강 서브-샘플링된 도메인(crude sub-sampled domain)에서 행해지므로, 더 정교한 움직임 벡터를 얻기 위해서는 세분화(refinement)가 필요로 된다.

[0039] Yamauchi, H 및 그 외 공동인에 의한 "An 81 MHz, 1280×720 픽셀 ×30 프레임/초 MPEG-4 비디오/오디오 CODEC 프로세서"에 개시되어 있으며, 본원에 인용되는, 반복 움직임 추정 탐색 방법에서는, 탐색 범위 중 단지 일부의 SAD만을 계산함으로써 SAD 계산 수를 더 줄인다. 이 방법을 이용하여, 시스템은 작은 영역의 SAD를 한 번에 계산한다. 이어서, 시스템은 그 영역의 SAD를 비교하여 최소 SAD의 위치를 가려낸다(pick). 다음에, 시스템은 이전의 탐색 결과들에 기초하여 또 다른 작은 영역을 선택한다. 시스템은 그러한 프로세스를 일정 횟수 반복한다. 모든 반복 동안 최소 SAD 값을 갖는 탐색 위치가 최적의 매칭 위치로서 선택된다. 반복 횟수는 하드웨어 성능에 따라 좌우된다. 도 3은 시작 위치(300) 및 결정된 최소 SAD(302)를 나타내는 예시적인 탐색을 예시하며, 여기서 각 단계는 한 번에 9 탐색점 ( $3 \times 3$ )을 탐색하며, 5회 반복한다.

[0040] 도 4는 반복 탐색의 흐름도를 예시한 것이다. 단계(400)에서, 프로세스는 N과 동일한 카운트에서 시작되며, 여기서 N은 탐색 횟수이다. 단계(402)에서, 영역의 SAD가 계산된다(예를 들어,  $3 \times 3$  영역). 단계(404)에서, 단계(402)에서의 계산에 기초한 영역에서 최소 SAD 위치가 선택된다. 단계(406)에서, 최소 SAD 위치를 이전의 최적(예컨대, 최소) SAD와 비교하여 둘 중 더 작은 것을 보존한다. 단계(408)에서, 반복을 처리하기 위해 카운트를 1만큼 감소시킨다. 단계(410)에서, 카운트가 0인지를 판단한다. 카운트가 0이면, 프로세스는 종료된다. 카운트가 0이 아니면, 단계(412)에서 다음 탐색 위치를 선택한다. 그 후, 프로세스는 단계(402)에서의 시작을 반복한다. 일부 실시예에서는, 카운팅의 다른 구현 또는 카운팅의 다른 순서를 이용할 수 있다.

[0041] 탐색 범위 중 일부만을 탐색하므로, SAD 계산 수는 소모적인 탐색에 비해 급격하게 줄어든다. 서브-샘플링된 탐색에서는, 적은 수의 탐색점(예를 들어,  $3 \times 3$  격자의 9)을 한 번에 탐색하고, 그 탐색을 N회(예컨대, 5회) 반복한다. " $3 \times 3$  탐색"은 "9 SAD"가 아닌 "9 위치"에 대한 SAD 계산을 의미한다. 따라서, 매크로블록( $16 \times 16$  픽셀)을 1/4 크기(예컨대,  $8 \times 8$  픽셀)로 서브-샘플링하면, "1 위치"에 대한 SAD의 계산 수는  $8 \times 8 = 64$ 이다. 하드웨어에서, 64 SAD 계산은 동시에 계산될 수 있다. 도 5는 예시적인  $3 \times 3$  탐색 위치(500) 및  $8 \times 8$  픽셀의 매크로블록의 서브-샘플(502)을 나타낸다.  $3 \times 3$  서브-샘플 탐색을 기술하였지만, 탐색의 크기는 그러한 크기로부터 제한되는 것은 아니다.



- [0042] 그러나, 서브-샘플링된 반복 탐색에 의하면, 탐색 범위 내의 모든 위치를 검사하지 않기 때문에 문제가 된다. 그러므로, 탐색은 최적의 가능한 해를 구하는 것을 보장해 주지 않는다. 구체적으로 설명하자면, 탐색이 틀린 방향으로 진행할 수 있고/있거나 최적의 탐색 위치가 시작 위치로부터 너무 멀리에 있을 수 있다.
- [0043] 추론적 시작점(Speculative Starting Point)
- [0044] 상기한 문제들을 해결하기 위해, 추론적 시작점 반복 탐색 방식이 이용된다. 추론적 탐색 스킴을 이용하면, 반복을 시작하기 전에, 시스템은 여러 개의 "가능성 있는(promising)" 위치에 대한 SAD를 비교한다. 이로써, 반복 탐색의 시작점으로서 "가능성 있는" 위치들 중 최적의 것을 이용한다. H.264 인코더의 구현에서, 탐색은 두 단계로 수행되지만, 이와는 다르게, 탐색은 임의의 수의 단계로 수행될 수 있다.
- [0045] 도 6은 움직임 추정을 위한 추론적 시작점을 구현하는 흐름도를 예시한다. 단계(600)에서, 세 위치, 즉 전역 움직임 벡터(Global Motion Vector; GMV), 제로(0) 움직임 벡터(zero Motion Vector; OMV) 및 예측된 움직임 벡터(Predicted Motion Vector; PMV)에 대한 SAD를 계산한다. GMV는 이전에 부호화된 프레임의 통계(statistics)로부터 프레임당 1회 계산되고, PMV는 현재 프레임 내의 인접한 매크로블록으로부터 각 매크로블록마다 추정된다. 최소 SAD를 갖는 위치를 이어지는 반복 탐색의 시작점으로서 이용한다. 상술한 것에만 한정되지 않고, 코더/디코더(코덱)에 따라, "가능성 있는" 위치로서 여러 위치가 선택될 수 있다. 위에서는 세 위치를 기술하였지만, 선택에 이용가능한 위치의 수는 세 개에 한정되지 않고, 더 많거나 더 적은 수의 위치가 선택될 수 있다.
- [0046] 단계(602)에서, 시스템은 도 4에서 기술된 반복 탐색을 행한다. 그러나, 반복의 시작 위치는 단계(600)에서의 최적의 시작 위치이다. 일부 실시예에서, 단계(600) 후에, 선택된 SAD를 반복 탐색의 다른 구현을 위한 시작 위치로서 이용한다. 단계(604)에서, 움직임 벡터를 세분화한다. 단계(606)에서, 서브 픽셀 탐색을 실행한다. 결과물은 보다 단시간에 처리되는 평탄한 화상/비디오이다.
- [0047] 도 7은 본 발명에 따른 예시적인 컴퓨팅 장치(700)의 블록도이다. 컴퓨팅 장치(700)는 화상 및 비디오 등과 같은 정보를 획득하고, 저장하고, 계산하고, 전달하고/하거나 디스플레이하는데 이용될 수 있다. 예를 들어, 컴퓨팅 장치(700)는 비디오를 획득하며, 비디오를 획득하는 동안, 개선된 움직임 추정 프로세스가 발생한다. 일반적으로, 컴퓨팅 장치(700)를 구현하기에 적합한 하드웨어 구조는 네트워크 인터페이스(702), 메모리(704), 프로세서(706), I/O 장치(들)(708), 버스(710) 및 저장 장치(712)를 포함한다. 프로세서의 선택은 충분한 속도의 적절한 프로세서를 선택하기만 하면 중요치 않다. 메모리(704)는 본 기술 분야에 공지된 임의의 통상의 컴퓨터 메모리일 수 있다. 저장 장치(712)는 하드 드라이브, CDROM, CDRW, DVD, DVDRW, 플래시 메모리 카드 또는 기타 임의의 저장 장치를 포함할 수 있다. 컴퓨팅 장치(700)는 하나 이상의 네트워크 인터페이스(702)를 포함할 수 있다. 네트워크 인터페이스의 일례로는, 이더넷 또는 다른 유형의 LAN에 연결되는 네트워크 카드가 포함된다. I/O 장치(들)(708)는 키보드, 마우스, 모니터, 디스플레이, 프린터, 모뎀, 터치스크린, 버튼 인터페이스 및 기타 장치 중 하나 이상을 포함할 수 있다. 본 발명의 방법을 수행하는데 이용되는 애플리케이션(들)(730)은 저장 장치(712) 및 메모리(704)에 저장되는 것으로 예상되며, 애플리케이션들이 전형적으로 처리되는 대로 처리될 것으로 예상된다. 애플리케이션들(730)은 시작점을 결정하는 시작점 컴포넌트(730') 및 반복 탐색을 수행하는 반복 탐색 컴포넌트(730'') 뿐 아니라, 기타 소망하거나 필요한 임의의 컴포넌트를 포함한다. 컴퓨팅 장치(700)에는 도 7에 도시된 컴포넌트들보다 더 많거나 더 적은 컴포넌트들이 포함될 수 있다. 일부 실시예에서, 움직임 추정 정보를 처리하기 위한 움직임 추정 하드웨어(720)가 포함된다. 비록, 도 7에 도시된 컴퓨팅 장치(700)가 애플리케이션들(730) 및 움직임 추정 처리용 하드웨어(720)를 포함하지만, 컴퓨팅 장치 상에서 하드웨어, 펌웨어, 소프트웨어 또는 그들의 임의의 결합으로 개선된 움직임 추정 프로세스를 구현할 수 있다.
- [0048] 적합한 컴퓨팅 장치의 예로는, 개인용 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 서버, 메인프레임 컴퓨터, 핸드헬드형 컴퓨터, 개인 휴대 단말기(PDA), 셀룰러/이동(모바일) 전화, 스마트 기기, 게이밍 콘솔, 디지털 카메라, 디지털 캠코더, 카메라 전화, iPod®, 가정용 오락 시스템 또는 기타 임의의 적합한 컴퓨팅 장치를 들 수 있다.
- [0049] 추론적 시작점 반복 탐색 방식을 이용함에 있어, 컴퓨팅 장치는 어느 때와 마찬가지로 동작하지만, 움직임 추정 프로세스는 더 효율적이고 더 정확하다는 점에서 개선된다. 사용자의 관점에서 본 컴퓨팅 장치의 이용은, 표준 움직임 추정을 이용하는 것과 유사하거나 동일하다. 예를 들어, 사용자는 여전히 디지털 캠코더의 전원을 켜서 캠코더를 이용하여 비디오를 기록하기만 하면 된다. 개선된 움직임 추정 프로세스는 비디오 품질을 자동으로

개선시킬 수 있다. 예를 들어, 추론적 시작점 반복 탐색 스킴은 손떨림 보정 기능(image stabilizer) 등의 움직임 추정을 필요로 하는 어느 곳에서도 이용될 수 있다. 기타 많은 용도에서 개선된 움직임 추정 프로세스가 이용될 수 있다.

[0050] 동작 중, 추론적 시작점 반복 탐색 스킴을 이용함에 의해, 반복 탐색은 여러 방법으로 개선된다. 개선된 반복 탐색은 완전히 틀린 위치로 가는 것을 방지하는데, 이는 이러한 반복 탐색이 최적의 가능한 시작점부터 시작하기 때문이다. 총 반복 횟수는 줄어들 수 있는데, 이는 반복 탐색이 "정적 위치(static location)" (OMV), 카메라 팬 등의 "전역 움직임(global movement)" (GMV) 및 "물체의 움직임(motion of object)" (PMV) 중에서 선택된 시작 MV를 이용하여 탐색 범위에서 최적의 가능한 위치에 가까운 위치부터 시작할 공산이 크기 때문이다. 최적의 위치(예컨대, MV)가 크더라도, 반복을 전역 움직임부터 시작하면 탐색 위치에 도달할 수 있다. H.264 인코더의 추론적 시작점 반복 탐색 스킴은 SAD 계산 시간을 극단적으로 줄여준다.

[0051] 지역적 최소값 회피(local minimum avoidance)

[0052] 반복 탐색의 중요한 양상은 다음 중심점을 선택하는 방법이다. 시스템이 단순히 최소 SAD를 갖는 방향을 선택한다면, 시스템은 지역적 최소값에서 왔다갔다하는 위치에 있을 수 있다. 예를 들어, 나이트 시스템은 최소 SAD 점 방향을 선택할 수 있다. 도 8에 도시된 예에서, 위치(802)가  $5 \times 5$  영역(800) 중 최소 SAD 값을 가질 경우에는 우측으로 진행하는 것이 합당하다. 다음 반복은 우측의  $5 \times 5$  영역에 대한 탐색이다. 그러나, 이런 탐색은 실제 최소값(902)이 아니라 도 9에 도시된 지역적 최소값(900)일 수 있다. 이 경우에, 다음 반복의 최소 SAD는 도 10에 도시된 바와 같이,  $5 \times 5$  영역에 있는 좌측 중심부(1002)이다. 그때, 나이트 선택 알고리즘은 다음 선택이 이미 탐색한 좌측의  $5 \times 5$  영역으로 진행할지를 결정한다. 이어지는 탐색에서, 최소 SAD는 다시 중심의 우측이 되고, 탐색은 계속 왔다갔다하여, 지역적 최소값에서 머무른다.

[0053] 지역적 최소값에서 머무르는 것을 회피하기 위해, 지역적 최소값 회피 탐색은 분석해야 할 다음 방향이 이전에 분석된 방향의 반대인지를 판단한다. 탐색이 다음 탐색의 중심부로 진행하기 전에, 탐색은 현재 탐색의 최소 SAD 위치를 보존한다. 도 8의 경우, 시스템은 그것이 동쪽으로 진행한 것을 기억한다.

[0054] 다음 탐색 위치의 SAD 계산을 완료하여 최적의 SAD를 골라낸 후, 시스템은 이전 반복의 "반대 방향"으로 진행하려고 시도하는가를 검사한다. 도 10의 경우에, 새로운 위치는 서쪽이며, 이 방향은 이전 방향과 정확히 반대이다. 이것이 발생하면, 시스템은 이전에 선택된 방향(예를 들어, 동쪽)으로 진행하는 것으로 결정한다. 반대 방향을 두어번 회피한 후, 탐색은 도 11에 도시된 바와 같이 지역적 최소값(1100)을 벗어나 실제 최소값(1102)을 향해 이동한다.

[0055] 도 12 내지 도 14는 지역적 최소값 회피 탐색의 예시적인 구현을 도시한 것이다. H.264 비디오 인코더의 구현에서, 시스템은 단지 이전 탐색의 최소 SAD 위치만을 기억한다.  $5 \times 5$  탐색(1200)의 경우, 0 내지 24의 번호가 보존된다. 도 12는 최소 SAD가 동쪽 중심 위치에서 위치(14)인 것으로 판단됨을 보여준다.

[0056] 반복 시 다음 단계에서, 새로운 최소 SAD 위치를 계산한 후, 새로운 위치를 이전 최적의 위치와 비교한다. 참조표 또는 다른 매핑 방식에 근거하여, 방법/시스템은 어느 위치를 반대 방향이 되도록 고려할지를 알 수 있다. 예를 들어, 이전 단계에서 최적의 위치는 도 12에서 번호 14이므로, 도 13의 음영 위치들(1300)은 "이전에 최적인 것에 반대"인 것으로 여겨진다.

[0057] 새로운 위치가 음영 위치들(1300) 중 임의의 것이면, 시스템은 반복을 이전의 탐색 위치에 대한 "후퇴(retreating)"로서 고려한다. 그래서, 시스템은 기본 반복 규칙을 무시하고, 다음 방향으로 동쪽을 택한다.

[0058]  $5 \times 5$  탐색의 경우, 일부 적용가능한 규칙들이 도 14에 도시된 패턴으로 줄어든다. 다른 위치들은 도 14를 회전시킴에 의해 적용될 수 있다. 이전의 탐색 위치에 대한 음영 위치(1400)가 도시되며, 음영 위치(1400) 반대쪽에 대응하는 금지된 위치(1402)가 도시된다. 일반적으로, 되도록이면 연장부(extension)가 반대쪽인 것으로 하여, 모서리 위치는 반대쪽 모서리 위치를 생성하며, 측면/상부/하부 위치는 각각의 반대되는 측면/하부/상부를 반대쪽 위치로서 갖는다.

[0059] 도 15는 간략화된 반복 탐색의 흐름도이다. 단계(1500)에서,  $5 \times 5$  탐색의 SAD를 계산한다. 단계(1502)에서, 최소 SAD 위치를 선택한다. 그 후, 단계(1504)에서, 현재 탐색 결과에 기초하여 새로운 탐색 중심점을 선택한다. 임계값에 도달하거나 또는 또 다른 기준을 충족할 때까지 프로세스는 정해진 횟수 반복된다.

[0060] 도 16에서, 지역적 최소값 회피 탐색에 대한 흐름을 예시한다. 새로운 탐색 중심점을 선택하기 전에, 새로운

방향이 후퇴하고 있는지에 대해 검사를 행한다. 그렇다면, 이전 방향으로 대체한다. 방향은 최소 SAD의 위치에 의해 결정된다. 예를 들어, 최소 SAD가 중심-동쪽 위치인 경우, 방향은 동쪽이다. 방향이 위쪽-동쪽 위치인 경우, 방향은 북-동쪽, 등등이다.

[0061] 구체적으로, 단계(1600)에서,  $5 \times 5$  영역의 각 탐색 위치의 SAD를 계산한다. 단계(1602)에서, 최소 SAD를 갖는 위치를 선택한다. 단계(1604)에서, 새로운 위치의 방향을 결정한다. 단계(1606)에서는, 방향이 이전 방향과 반대인지를 판단한다. 방향이 반대가 아닌 경우, 단계(1608)에서 새로운 위치를 보존하고, 단계(1610)를 건너뛰는다. 방향이 반대인 경우, 단계(1610)에서 새로운 위치를 이전 위치로 설정한다. 이어서, 단계(1612)에서, 단계(1608) 또는 단계(1610)에서 결정된 위치에 기초하여 새로운 탐색 중심점을 선택한다. 단계(1614)에서 임계값에 도달하거나 또는 또 다른 기준을 충족할 때까지, 프로세스는 정해진 횟수를 반복한다.

[0062] 일부 실시예에서, "후퇴"가 검출되면, 동일 방향(예를 들어, 이전 방향)을 따른다. 일부 실시예에서, "후퇴"가 검출되면, 조정된 방향을 택하며, 여기서 조정된 방향은 이전 방향과 새로운 방향 사이의 어느 곳이다.

[0063] 예시적 목적으로  $5 \times 5$  탐색을 이용하였지만, 임의 적절하게 크기로 분류된 탐색도 가능하다.

[0064] 위에서 기술한 바와 같이, 도 7은 컴퓨팅 장치(700)를 예시한다. 컴퓨팅 장치가 추론적 시작점 방법을 실행할 수 있는 것 이외에도, 컴퓨팅 장치는 다른 애플리케이션에 더하여 또는 다른 애플리케이션 대신 지역적 최소값 회피를 실행할 수 있다. 일부 실시예에서, 컴퓨팅 장치(700)는 지역적 최소값들을 회피하기 위해 추가의 애플리케이션(732)을 포함한다. 일부 실시예에서, 지역적 최소값 회피 방법은 추론적 시작점 선택 애플리케이션(730)과 동일한 애플리케이션으로 구현된다. 일부 실시예에서, 컴퓨팅 장치의 ME HW(720)는 지역적 최소값들을 회피하도록 구성된다. 지역적 최소값 회피는 하드웨어, 펌웨어, 소프트웨어 또는 이들의 임의 결합으로 구현될 수 있다.

[0065] 지역적 최소값 회피 탐색을 이용함에 있어, 컴퓨팅 장치는 통상적으로 동작하지만, 지역적 최소값들을 회피할 수 있기 때문에 더 효율적이며 더 정확하다는 점에서 움직임 추정 프로세스는 개선되었다. 사용자의 관점에서 컴퓨팅 장치의 활용도는 표준 움직임 추정을 이용하는 것과 유사하거나 동일하다. 예를 들어, 사용자는 여전히 디지털 캠코더의 전원을 켜고, 그 캠코더를 사용하여 비디오를 기록하기만 하면 된다. 지역적 최소값 회피 탐색은 비디오 품질을 자동으로 개선할 수 있다. 예를 들어, 지역적 최소값 회피 탐색은 손떨림 보정 기능 등의 움직임 추정을 필요로 하는 그 어느 곳에서도 이용될 수 있다. 기타 여러 용도에서 지역적 최소값 회피 탐색이 이용될 수 있다.

[0066] 동작 중, 지역적 최소값 회피 탐색은 반복 탐색이 동일 위치를 왔다갔다 연속으로 탐색하는 것을 방지함으로써 지역적 최소값을 회피한다.

[0067] 다음 탐색 위치 스킴(Next Search Position Scheme)

[0068] 반복 탐색은 탐색 범위 중 제한된 탐색점들만을 탐색할 수 있으므로, 탐색점들이 효과적으로 선택되지 않을 경우, 화질은 상당히 떨어진다. 또한, 탐색점 선택은 반복에서 매 단계마다 수행된다. 그러므로, 판정(decision-making) 알고리즘은 신속한 것이 바람직하다.

[0069] 다음 탐색 위치 스킴은 최소 SAD의 위치에만 기초하여 다음 탐색 중심부를 신속하게 결정할 수 있다.

[0070] 도 17은 동작 중의 다음 위치 선택기(1702)의 흐름도이다. 다음 위치 선택기(1702)의 입력(1700)은 현재 탐색에서 최소 SAD의 위치이다.  $3 \times 3$  스텝의 경우, 위치는 도 18에 도시된 바와 같이 번호 0 내지 8이다. 출력(1704)은 현재 탐색 중심부로부터 새로운 탐색 중심부까지의 오프셋이다. 이로써, 반복 탐색은 출력에 기초하여 이동한다.

[0071] 예를 들어, 최소 SAD의 위치가 "1"로 분류되거나 도 18의 위치 중 하나이면, 다음 탐색 중심부는 현재 탐색 중심부의 북쪽으로 3 픽셀이 될 것이다. 따라서, 현재 탐색의 탐색 중심부가 위치(10, 10)에 있고,  $3 \times 3$  탐색 영역에서 최적의/최소 SAD 위치가 위치 1인 경우, 다음 탐색 중심부는 위치(10, 7)에 있을 것이다. 그러나, 최소 SAD가 위치 5에 있으면, 다음 탐색 중심부는 동쪽으로 3 픽셀이 될 것이다. 따라서, 현재 탐색의 탐색 중심부가 위치(10, 10)에 있는 경우, 새로운 탐색 중심부는 위치(13, 10)에 있게 될 것이다.

[0072] 도 18 내지 도 20은  $3 \times 3$  탐색 영역에 대한 현재 탐색 영역의 최소 SAD를 갖는 위치에 기초한 다음 탐색 영역을 예시한 것이다. 도 18은 각 위치를 번호로 나타낸  $3 \times 3$  탐색을 예시한다. 패턴은 수평, 수직 및 대각선

위치만을 포함한다. 도 19는 대응하는 다음 탐색 영역 오프셋 - 위치 1: 북쪽 (0, -3), 위치 3: 서쪽 (-3, 0), 위치 7: 남쪽 (0, 3) 및 위치 5: 동쪽 (3, 0) - 을 갖는 수평 위치(3, 5) 및 수직 위치(1, 7)를 도시한다. 도 20은 대응하는 다음 탐색 영역 오프셋 - 위치 0: 북서쪽 (-2, -2), 위치 6: 남서쪽 (-2, 2), 위치 8: 남동쪽 (2, 2) 및 위치 2: 북동쪽 (2, -2) - 을 갖는 대각선 위치(0, 2, 6, 8)를 도시한다.

[0073] 비록  $3 \times 3$  탐색 영역에 대해 기술하였지만, 임의의 적합한 탐색 영역이 가능하다.

[0074] 도 21은 다음 위치를 선택하는 방법에 대한 흐름을 도시한다. 단계(3100)에서, 탐색 영역에서 최소 SAD를 갖는 위치가 결정된다. 일부 실시예에서, 탐색 영역은  $3 \times 3$  픽셀이며; 일부 실시예에서, 탐색 영역은 크기가 다르게 되어 있다. 단계(3102)에서, 최소 SAD 위치에 기초한 다음 탐색 영역마다 다음 탐색 영역 중심점에 대한 오프셋을 자동으로 계산/조회(look-up)한다. 일부 실시예에서, 다음 중심점의 실제 위치는 현재 중심점에 기초하여 계산된다. 단계(3104)에서, 다음 탐색 영역에 대한 중심점이 선택된다.

[0075] 다음 위치 선택기에서 이용되는 규칙들(rules)은 참조표에 의해 구현될 수 있다. 일부 실시예에서, 규칙들은 다른 수단에 의해 구현된다. 일부 실시예에서, SAD 자체의 값은 전혀 필요로 되지 않는다. 규칙들은 다음 탐색 영역의 중심점을 결정하기 위해 최소 SAD의 위치를 이용하기만 하면 된다. 또한, 규칙들은 참조표로 구현되므로, 그들은 필요에 따라 쉽사리 변경가능하다.

[0076] 도 22는 최소 SAD를 갖는 위치에 기초하여 다음 탐색 영역의 중심점을 결정하는 예시적인 참조표(3200)를 도시한다. 위에서 기술한 바와 같이, 다음 탐색 영역의 중심점에 대한 오프셋은 최소 SAD 위치에 기초하여 결정된다. 도 22에 도시된 바와 같이, 최소 SAD가 위치 0에 있을 경우, 다음 탐색 영역의 중심점은 (-2, -2)의 오프셋이다. 다른 참조 값들 역시 나타나 있다. 도 22는  $3 \times 3$  픽셀의 탐색 영역을 취한 것이다. 다른 탐색 영역의 경우, 값들은 적절하게 변경될 것이다.

[0077] 단순한 지역 최소 회피 탐색 스킴과 결합한 다음 탐색 영역 선택 스킴은 H.264 인코더에서 적은 프로세서 사이클로 고품질의 부호화를 이룰 수 있다.

[0078] 위에서 기술한 바와 같이, 도 7은 컴퓨팅 장치(700)를 예시한다. 컴퓨팅 장치(700)가 추론적 시작점 반복 탐색 스킴 및 지역적 최소값 회피 방법을 실행할 수 있는 것 이외에도, 컴퓨팅 장치(700)는 다른 애플리케이션 대신 또는 다른 애플리케이션에 더하여 다음 탐색 위치 선택 방법을 실행할 수 있다. 일부 실시예에서, 컴퓨팅 장치(700)는 다음 탐색 위치를 선택하는 추가 애플리케이션(734)을 포함한다. 일부 실시예에서, 다음 탐색 위치 선택 방법은 애플리케이션(732) 또는 추론적 시작점 애플리케이션(730) 등과 동일한 애플리케이션으로 구현된다. 일부 실시예에서, 다음 탐색 위치를 선택하는 애플리케이션(734) 또는 하드웨어(720)는 탐색 영역의 최소 SAD 위치를 받아들이는 입력 컴포넌트(734'), 최소 SAD 위치에 기초한 오프셋을 조회하는 참조표 등의 매핑 컴포넌트(734'') 및 다음 탐색 영역의 중심점의 오프셋을 출력하는 출력 컴포넌트(734''')를 포함한다. 일부 실시예에서는, 오프셋 출력 대신 또는 오프셋 출력에 더하여, 다음 탐색 영역의 중심점의 실제 위치가 결정된다. 일부 실시예에서, 컴퓨팅 장치 ME HW(720)는 다음 탐색 위치를 선택하도록 구성된다. 다음 탐색 위치 선택은 하드웨어, 펌웨어, 소프트웨어 또는 이들의 임의의 결합으로 구현될 수 있다.

[0079] 다음 위치 선택기를 이용함에 있어, 컴퓨팅 장치는 평소와 같이 동작하지만, 탐색해야 할 더 양호한 다음 위치를 선택할 수 있으므로 움직임 추정 프로세스는 더 효율적이고 더 정확하다는 점에서 개선된다. 사용자의 관점에서 본 컴퓨팅 장치의 이용은 표준 움직임 추정을 이용하는 것과 유사하거나 동일하다. 예를 들어, 사용자는 여전히 디지털 캠코더의 전원을 켜서 캠코더를 이용하여 비디오를 기록하기만 하면 된다. 다음 위치 선택기는 비디오 품질을 자동으로 개선시킬 수 있다. 예를 들어, 다음 위치 선택기는 손떨림 보정 기능 등의 움직임 추정을 필요로 하는 어느 곳에서도 이용될 수 있다. 기타 많은 용도에서 다음 위치 선택기가 이용될 수 있다.

[0080] 동작 중, 다음 위치 선택기는 움직임 추정의 효율을 개선하기 위해 다음 탐색 영역의 중심점을 결정한다. 또한, 다음 위치 선택기는 단순한 스킴을 이용하므로, 쉽사리 구현된다.

# [0081] 세분화 스킴(Refinement Scheme)

[0082] 매끄럽고, 평탄한 영역에서, 대부분의 위치는 유사한 SAD 값들을 갖는다. 그러므로, 어떠한 움직임 벡터도 양호한 후보인 것으로 보이므로, 탐색 알고리즘은 쉽게 무력화된다(fool). 그러나, ME가 임의의 움직임 벡터를 선택하면, 계수보다 움직임 벡터 헤더용으로 더 많은 비트를 소비하여 주관적(subjective) 화질을 악화시키게 된다. 또한, 신호 대 잡음비에 많은 영향을 주지 않더라도, 매끄러운 면에서 일관성 없는 움직임은 부자연스러운



화상을 생성하여 주관적 화질을 떨어뜨린다.

- [0083] 원시 화상이 많은 잡음을 가지면, SAD 계산은 잡음으로 인해 쉽게 무력화된다. 때때로, 잡음 때문에 올바르게 많은 움직임 벡터가 최소 SAD 값을 갖는다. 그러나, 이러한 올바르게 많은 움직임들은 사람의 눈에 쉽게 발견된다.
- [0084] 예측 움직임 벡터(PMV)를 이용한 반복 탐색을 세분화하는 스킴에 대해 기술하기로 한다. PMV는 인접한 매크로블록으로부터 예측되는 현재 매크로블록의 가장 가능성이 큰 움직임 벡터이다.
- [0085] 주변 매크로블록의 움직임 벡터를 이용한 반복 탐색에 대한 흐름이 도 23에 도시된다. PMV 세분화는 PMV 위치에 두 기회를 제공한다. 단계(3300)에서, 세 위치: 전역 움직임 벡터(GMV), 제로(0) 움직임 벡터(OMV) 및 예측된 움직임 벡터(PMV)에 대한 SAD가 계산된다. GMV는 이전에 부호화된 프레임의 통계로부터 프레임당 한 번 계산되고, PMV는 현재 프레임 내의 인접한 매크로블록으로부터 각 매크로블록마다 추정된다. 최소 SAD를 갖는 위치는 이어지는 반복 탐색의 시작 위치로 이용된다. 코더/디코더(코덱)에 따라, 여러 위치가 "유력한(promising)" 위치로 선택될 수 있지만, 위에서 기재한 것에만 한정되는 것은 아니다. 비록 위에서는 세 위치를 기재하고 있지만, 선택할 수 있는 위치의 수는 3개로만 한정되는 것이 아니라, 더 많거나 더 적은 위치가 선택될 수 있다.
- [0086] 단계(3302)에서, 시스템은 도 4에 기술된 반복 탐색을 수행한다. 그러나, 반복 시작 위치는 단계(3300)에서의 최적의 시작 위치가 된다. PMV 위치의 SAD 값을 계산하여 OMV 및 전역 MV 등의 다른 위치와 비교하여 위에서 기술한 반복 시작 위치를 결정한다. 일부 실시예에서, 단계(3300) 후에, 선택된 SAD를 반복 탐색의 다른 구현을 위한 시작 위치로서 이용한다. 반복 탐색 결과, 최소 SAD 위치가 생겨난다. 단계(3304)에서, 반복 탐색 중 승자(winner)의 SAD에 움직임 벡터 비용(cost)이 가산되는 한편, PMV의 SAD에는 MV 비용이 가산되지 않는다. 그 후, 단계(3306)에서 움직임 벡터 비용이 가산된 반복 탐색 중 승자(최소 SAD 위치)를, PMV 위치의 SAD와 다시 비교한다. 움직임 벡터 비용을 갖는 반복 탐색 결과와 PMV의 SAD 중 작은 것이 선택된다. 단계(3308)에서, 움직임 벡터가 세분화된다. 그 결과, 평탄한 화상/비디오가 더 줄어드는 시간으로 처리된다.
- [0087] MV 비용을 계산하는 여러 방법이 존재한다. 일반적으로, 움직임 벡터 비용은 PMV로부터의 거리에 기초하여 계산된다. 따라서, PMV 위치에는 MV 비용이 가산되지 않는다.
- [0088] 도 24는 현재 매크로블록(3400)을 인접한 매크로블록(3402) 및 그들의 움직임 벡터와 함께 예시한 것이다. PMV는 인접한 매크로블록(3402)으로부터 계산된다.
- [0089] 세분화 스킴은 반복 탐색이 독단적인 움직임 벡터를 선택하는 것을 방지한다. 이것은 또한, 움직임 벡터들이 매끄러운 면에서도 일관성 있게 되는 것을 도와주어, 움직임 벡터 헤더용으로 쓰이는 비트들을 절약하게 된다. 이로 인해, 주관적인 화질이 개선된다. 더욱이, 잡음이 있는 비디오 시퀀스에서 조차, 시스템이 무력화될 가능성은 적다.
- [0090] 세분화 스킴은 PMV를 계산하는 방법에만 제한되는 것은 아니다. 또한, 세분화 스킴은 어떠한 종류의 ME 응용에도 적용가능하다.
- [0091] 위에서 기술한 바와 같이, 도 7은 컴퓨팅 장치(700)를 예시한다. 컴퓨팅 장치(700)는 또한 세분화 스킴을 실행할 수 있다. 일부 실시예에서, 컴퓨팅 장치(700)는 세분화 스킴을 실행하기 위한 추가의 애플리케이션(736)을 포함한다. 일부 실시예에서, 애플리케이션(736)은 시작 위치 컴포넌트(736'), 반복 탐색 컴포넌트(736'') 및 추가의 비교 컴포넌트(736''')를 갖는다. 일부 실시예에서, 세분화 스킴은 이전에 기술한 애플리케이션들 중 하나와 동일한 애플리케이션으로 구현된다. 일부 실시예에서, 컴퓨팅 장치 ME HW(720)는 세분화 스킴을 구현하도록 구성된다. 세분화 스킴은 하드웨어, 펌웨어, 소프트웨어 또는 그들의 임의 결합으로 구현될 수 있다.
- [0092] 세분화 스킴을 활용함에 있어, 컴퓨팅 장치는 통상적으로 동작하지만, 움직임 추정 프로세스는 개선되는데, 그 이유는 적합한 움직임 벡터가 선택될 것이므로, 특히 매끄럽고 평탄한 영역인 경우에 더 효율적이고 정확하기 때문이다. 사용자 관점에서 본 컴퓨팅 장치의 활용은 표준 움직임 추정을 이용하는 것과 유사하거나 동일하다. 예를 들어, 사용자는 여전히 디지털 캠코더의 전원을 켜서 캠코더를 이용하여 비디오를 기록하기만 하면 된다. 세분화 스킴은 비디오 품질을 자동으로 개선시킬 수 있다. 예를 들어, 세분화 스킴은 손떨림 보정 기능 등의 움직임 추정을 필요로 하는 그 어느 곳에서도 이용될 수 있다. 기타 많은 용도에서 세분화 스킴이 이용될 수 있다.
- [0093] 동작 시, 세분화 스킴은 움직임 벡터의 SAD에 움직임 벡터 비용을 가산하여 그 값을 PMV의 SAD와 비교함으로써

적합한 움직임 벡터를 결정한다. 그래서, 둘 중 낮은 것을 움직임 벡터의 세분화용으로 이용한다. 이런 추가의 비교는 화상의 매끄럽고 평편한 영역이나 잡음이 있는 화상으로부터 기인하는 문제들을 회피하는 것을 도와준다.

[0094]

임계값을 갖는 세분화 스킴

[0095]

픽처의 움직임이 매우 큰 경우, 위에서 기술한 세분화 스킴은 일부 부정적인 효과를 갖는다. 구체적으로 기술하자면, 시스템은 반복 탐색 중 승자와 PMV 위치를 비교한다. 그러나, 총 비용은 PMV 위치와 비교되며, 여기서 총 비용은 "SAD + MV 비용"인 반면, PMV의 총 비용은 단지 SAD이다. 따라서, 픽처에서의 움직임이 클 경우, MV 비용 또한 상대적으로 커지려는 경향이 있다. 이로 인해, 세분화 스킴은 하여금 움직임 추정 시스템이 큰 움직임 벡터를 생성하는 것을 강력하게 하지 못하게 한다.

[0096]

임계값이 추가된 주변 매크로블록의 움직임 벡터를 이용한 반복 탐색의 흐름이 도 25에 도시되어 있다. PMV 세분화는 PMV 위치에 두 기회를 제공하지만, 임계값은 두 번째 기회가 도움이 되는 것인지를 판단한다. 단계(3500)에서, SAD는 세 위치: 전역 움직임 벡터(GMV), 제로(0) 움직임 벡터(OMV) 및 예측된 움직임 벡터(PMV)에 대해 계산된다. GMV는 이전에 부호화된 프레임의 통계로부터 프레임당 한 번 계산되며, PMV는 현재 프레임 내의 인접한 매크로블록으로부터 각 매크로블록마다 추정된다. 최소 SAD를 갖는 위치가 후속되는 반복 탐색의 시작점으로서 이용된다. 코더/디코더(CODEC)에 따라, 위에서 기술된 것에 한하지 않고, 여러 위치들이 "유망한" 위치로서 선택될 수 있다. 위에서는 세 위치를 기술하였지만, 선택할 수 있는 위치의 개수는 3개에만 한정되는 것이 아니라, 더 많거나 더 적은 개수의 위치가 선택될 수 있다.

[0097]

단계(3502)에서, 시스템은 도 4에 기술된 반복 탐색을 수행한다. 그러나, 반복 시작 위치는 단계(3500)로부터의 최적의 시작 위치이다. PMV 위치의 SAD가 계산되고 OMV 및 전역 MV(GMV) 등의 다른 위치와 비교되어 위에서 기술한 반복 시작 위치를 결정한다. 일부 실시예에서, 단계(3500) 이후에, 선택된 SAD는 반복 탐색의 다른 구현을 위한 시작점으로서 이용된다. 반복 탐색의 결과, 최소 SAD 위치가 산출된다. 단계(3504)에서, 반복 탐색 중 승자의 SAD에 움직임 벡터 비용이 가산된다. 일부 실시예에서, 단계(3504) 및 단계(3506)의 순서가 바뀐다. 단계(3506)에서 GMV가 임계값과 비교된다. GMV가 임계값보다 작은 경우, 단계(3508)에서 움직임 벡터 비용이 가산된 반복 탐색 중 승자(예를 들어, 최소 SAD 위치)를 PMV 위치의 SAD와 비교한다. 반복 탐색 결과와 PMV의 SAD 중 작은 것이 선택된다. 단계(3510)에서, 선택된 움직임 벡터가 세분화된다. 단계(3506)에서, GMV가 임계값 이상이면, 단계(3508)에서의 비교 단계는 건너뛰고, 반복 탐색의 결과는 단계(3510)에서 세분화된 움직임 벡터가 된다. 그 결과, 매끄러운 화상/비디오가 단축된 시간으로 처리된다.

[0098]

임계값을 갖는 세분화 스킴은 반복 탐색이 독단적인 움직임 벡터를 선택하는 것을 방지하고, 또한 PMV 우선(precedence)이 더 큰 움직임 벡터를 막으려는 것을 못하게 한다. 이것은 또한, 움직임 벡터들이 매끄러운 면에서도 일관성 있게 되는 것을 도와주어, 움직임 벡터 헤더에 쓰이는 비트들이 절약된다. 이는, 픽처에서의 움직임이 클 경우 주관적인 화상 품질 개선으로 이어진다.

[0099]

임계값을 갖는 세분화 스킴은 PMV를 계산하는 방법에만 제한되지 않는다. 또한, 세분화 스킴은 어떤 종류의 ME 응용에도 적용가능하다.

[0100]

위에서 기술한 바와 같이, 도 7은 컴퓨팅 장치(700)를 예시한다. 컴퓨팅 장치(700)는 또한 임계값을 갖는 세분화 스킴을 실행할 수 있다. 일부 실시예에서, 컴퓨팅 장치(700)는 임계값을 갖는 세분화 스킴을 실행하기 위한 추가의 애플리케이션(738)을 포함한다. 일부 실시예에서, 애플리케이션(738)은 시작 위치 컴포넌트(738'), 반복 탐색 컴포넌트(738''), 임계값 컴포넌트(738''') 및 추가의 비교 컴포넌트(738''')를 갖는다. 일부 실시예에서, 임계값을 갖는 세분화 스킴은 이전에 기술한 애플리케이션들 중 하나와 동일한 애플리케이션으로 구현된다. 일부 실시예에서, 컴퓨팅 장치 ME HW(720)는 임계값을 갖는 세분화 스킴을 구현하도록 구성된다. 임계값을 갖는 세분화 스킴은 또한 하드웨어, 펌웨어, 소프트웨어 또는 그들의 임의 결합으로 구현될 수 있다.

[0101]

임계값을 갖는 세분화 스킴을 활용함에 있어, 컴퓨팅 장치는 통상적으로 동작하지만, 움직임 추정 프로세스는 개선되는데, 그 이유는 적합한 움직임 벡터가 선택될 것이므로, 특히 매끄럽고 평탄한 영역인 경우에 더 효율적이고 정확하기 때문이다. 사용자 관점에서 본 컴퓨팅 장치의 활용은 표준 움직임 추정을 이용하는 것과 유사하거나 동일하다. 예를 들어, 사용자는 여전히 디지털 캠코더의 전원을 켜서 캠코더를 이용하여 비디오를 기록하 기만 하면 된다. 임계값을 갖는 세분화 스킴은 비디오 품질을 자동으로 개선시킬 수 있다. 예를 들어, 임계값을 갖는 세분화 스킴은 손떨림 보정 기능 등의 움직임 추정을 필요로 하는 그 어느 곳에서도 이용될 수 있다.

기타 많은 용도에서 임계값을 갖는 세분화 스킴이 이용될 수 있다.

- [0102] 동작 중, 임계값을 갖는 세분화 스킴은 PMV를 갖는 반복 탐색 결과와의 비교가 GMV와 임계값의 비교에 기초하여 적절한지를 판단한다. 적절할 경우, 임계값을 갖는 세분화 스킴은 움직임 벡터의 SAD에 움직임 벡터 비용을 추가하고 그 값을 PMV의 SAD와 비교함으로써 적절한 움직임을 결정한다. 이로써, 둘 중에 낮은 것이 움직임 벡터의 세분화용으로 이용된다. 적절하지 않을 경우에는, 반복 탐색 결과로부터의 움직임 벡터가 세분화용으로 이용된다.
- [0103] 조기 종료 스킴(Early Termination Scheme)
- [0104] 반복 탐색 알고리즘이 SAD 계산 횟수를 줄여주지만, 시스템에서의 SAD의 전력 소모는 여전히 크다. 그러므로, 반복 횟수를 줄이는 방법이 결국에는 전력 소모를 줄이게 될 것이다.
- [0105] 반복 탐색의 전력 소모를 줄이기 위해, "양호한" 탐색 후보가 발견되면 반복 탐색은 종료된다. 시스템은 1 스텝(one step)의 최적 SAD를 임계값과 비교한다. 최적의 SAD가 임계값보다 작을 경우에는, 그것을 "양호한" 탐색 후보인 것으로 고려하여 반복을 종료한다. 그렇지 않을 경우에는, 탐색을 통상대로 계속한다.
- [0106] 도 26은 임계값을 갖는 반복 탐색의 흐름을 도시한다. 단계(3600)에서, 프로세스는 N과 동일한 카운트로 시작되며, 여기서, N은 탐색할 횟수이다. 단계(3602)에서, 소정 영역(region)에서의 SAD가 계산된다(예를 들어,  $3 \times 3$  영역). 단계(3604)에서, 단계(3602)에서의 계산에 기초하여 그 영역에서 최소 SAD 위치가 선택된다. 단계(3606)에서, 최소 SAD 위치가 이전 최적의(예를 들어, 최소) SAD와 비교하여, 둘 중 작은 것을 보존한다. 단계(3608)에서, 단계(3606)에서의 비교를 설명하기 위해 카운트를 1만큼 감소시킨다. 단계(3610)에서, 카운트가 0인지를 판단한다. 카운트가 0이면, 프로세스가 종료된다. 카운트가 0이 아니면, 단계(3612)에서 보존된 SAD가 임계값 미만인지를 판단한다. SAD가 임계값 미만이면, 프로세스가 종료된다. SAD가 임계값 미만인 경우, 단계(3614)에서 그 다음 탐색 위치가 선택된다. 그 후, 프로세스는 단계(3602)에서의 시작을 반복한다. 일부 실시예에서, 카운팅의 다른 구현 또는 카운팅의 다른 순서를 이용할 수 있다.
- [0107] \*위에서 기술한 바와 같이, 도 7은 컴퓨팅 장치(700)를 예시한다. 컴퓨팅 장치(700)는 또한 조기 종료 스킴을 실행할 수 있다. 일부 실시예에서, 컴퓨팅 장치(700)는 조기 종료 스킴을 실행하는 추가의 애플리케이션(740)을 포함한다. 일부 실시예에서, 조기 종료 스킴은 이전에 기술한 애플리케이션들 중 하나와 동일한 애플리케이션으로 구현된다. 일부 실시예에서, 애플리케이션(740)은 반복적으로 탐색하는 반복 탐색 컴포넌트(740') 및 임계값 조건을 충족한 경우 조기 종료를 위한 임계값 컴포넌트(740'')를 포함한다. 일부 실시예에서, 컴퓨팅 장치 ME HW(720)는 조기 종료 스킴을 구현하도록 구성된다. 조기 종료 스킴은 하드웨어, 소프트웨어 또는 그들의 임의의 결합으로 구현될 수 있다.
- [0108] 조기 종료 스킴을 활용함에 있어, 컴퓨팅 장치는 어느 때와 마찬가지로 동작하지만, 움직임 추정 프로세스는 더 효율적인 점에서 개선된다. 사용자의 관점에서 본 컴퓨팅 장치의 이용은 표준 움직임 추정을 이용하는 것과 유사하거나 동일하다. 예를 들어, 사용자는 여전히 디지털 캠코더의 전원을 켜서 캠코더를 이용하여 비디오를 기록하기만 하면 된다. 조기 종료 스킴은 비디오 품질을 자동으로 개선시킬 수 있다. 예를 들어, 조기 종료 스킴은 손떨림 보정 기능 등의 움직임 추정을 필요로 하는 그 어느 곳에서도 이용될 수 있다. 기타 많은 용도에서 조기 종료 스킴이 이용될 수 있다.
- [0109] 동작 중, 조기 종료 스킴은 최적의 SAD가 임계값 미만인지를 판단한다. 최적의 SAD가 임계값 미만인 경우, 조기 종료 스킴은 나머지 카운트를 완료하지 않고 종료한다. 움직임 추정을 하드웨어로 구현할 경우, 조기 종료 스킴은 반복 탐색의 전력 소모를 줄여준다. 움직임 추정을 소프트웨어로 구현할 경우, 조기 종료 스킴은 프로세서 사이클을 줄여주어 부호화 속도가 더 빠르게 된다.
- [0110] 상술한 방법들 중 임의 및/또는 전부 필요에 따라 독립된 장치 또는 단일 장치 상에서 구현될 수 있다. 예를 들어, 디지털 캠코더가 추론적 시작점 방법, 지역적 최소값 회피(local minimum avoidance), 다음 탐색 위치 스킴, 조기 종료 스킴, 임계값을 갖는 세분화 스킴 및 조기 종료 스킴을 포함할 수 있다.
- [0111] 본원에서는 SAD가 패턴 매칭 정도를 계산하기 위해 반복 탐색에서 구현되는 거리 메트릭으로서 기술되었지만, Sum of Absolute Transferred Distances 및 Mean Squared Error 등(단, 이것에만 한정되는 것은 아님)의 오차

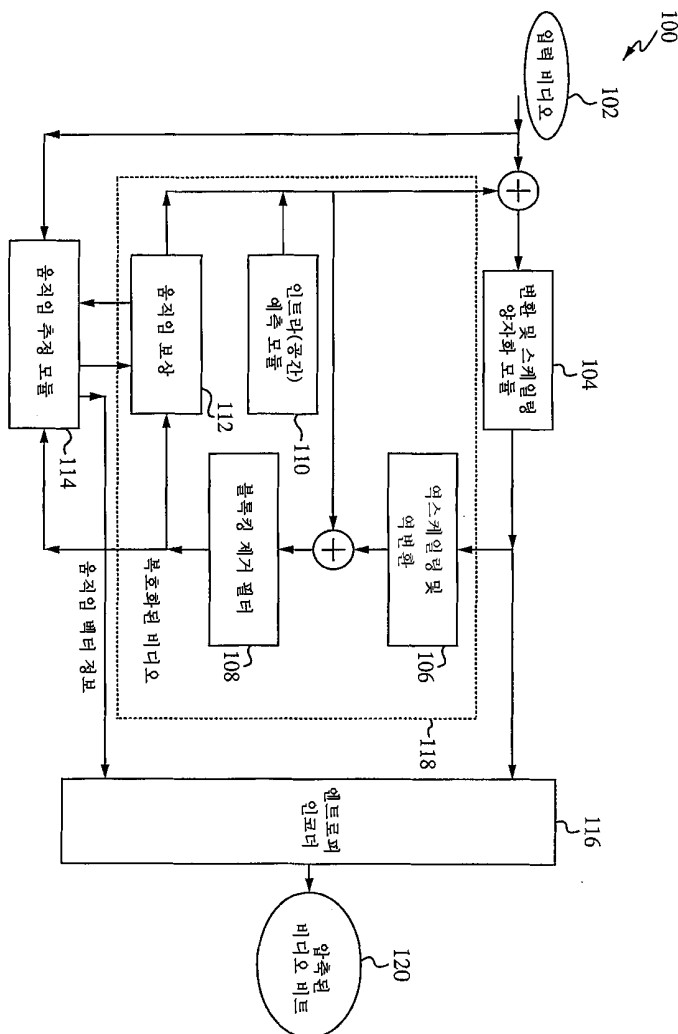
값 또는 다른 거리와 같이 사용될 수 있는 다수의 다른 거리 또는 오차 메트릭이 있다.

[0112]

본 발명은 본 발명의 구성 및 동작에 대한 원리의 이해를 돕기 위해 상세한 설명을 포함한 특정 실시예들에 대해 기술하였다. 특정 실시예 및 그에 대한 상세한 설명에 대한 본원에서의 그러한 참조는 첨부된 청구범위의 범주를 한정하려는 의도는 아니다. 당업자라면, 청구범위에 의해 한정되는 본 발명의 사상 및 범주를 벗어나지 않는 한, 설명을 위해 선택된 실시예들에 대한 여러 수정 및 변경 실시예가 가능하다는 것을 쉽게 알 수 있을 것이다.

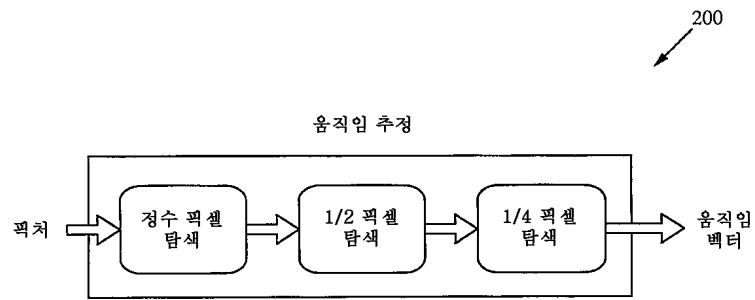
## 도면

### 도면1

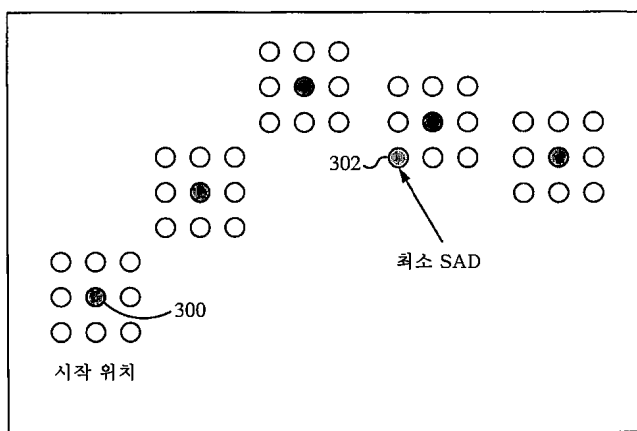




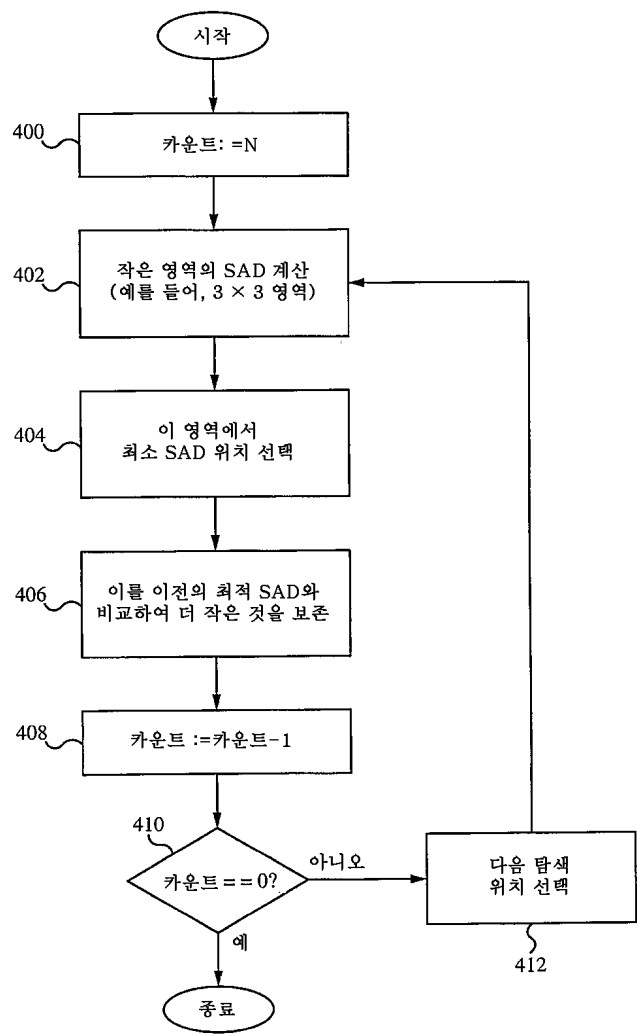
도면2



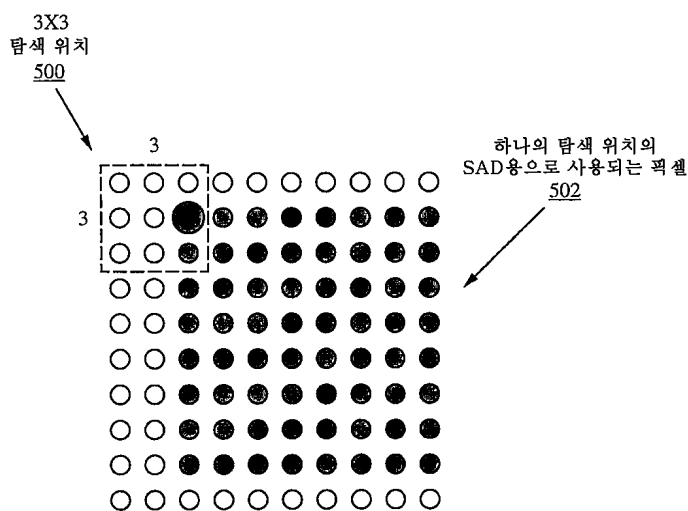
도면3



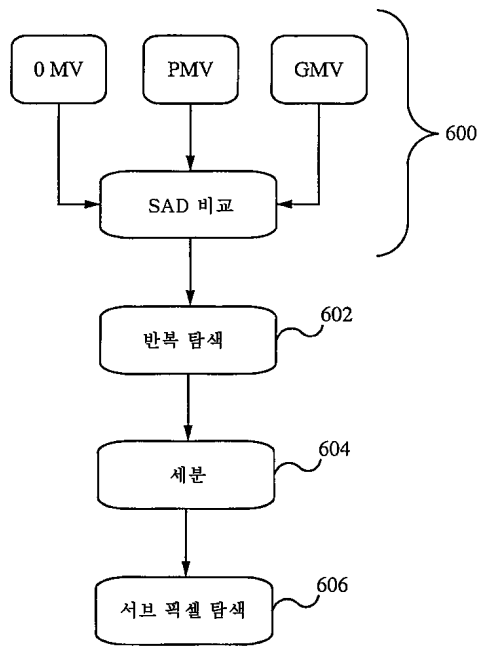
도면4



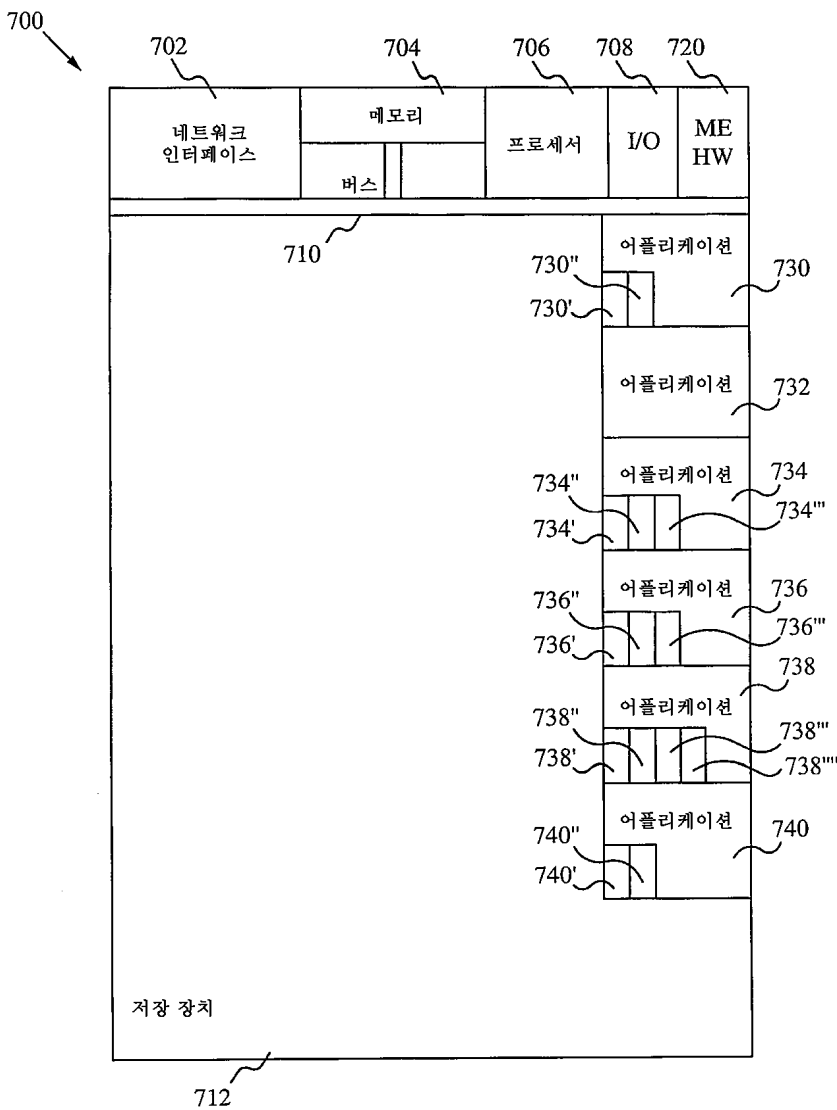
도면5



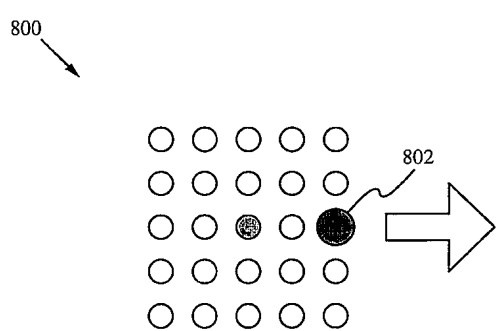
도면6



도면7

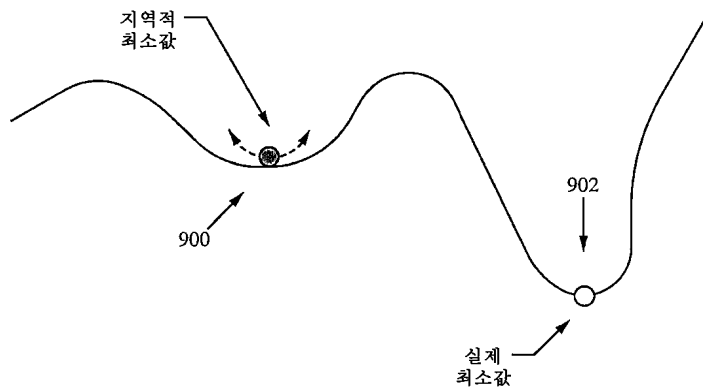


도면8

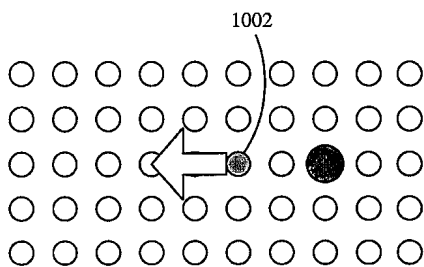




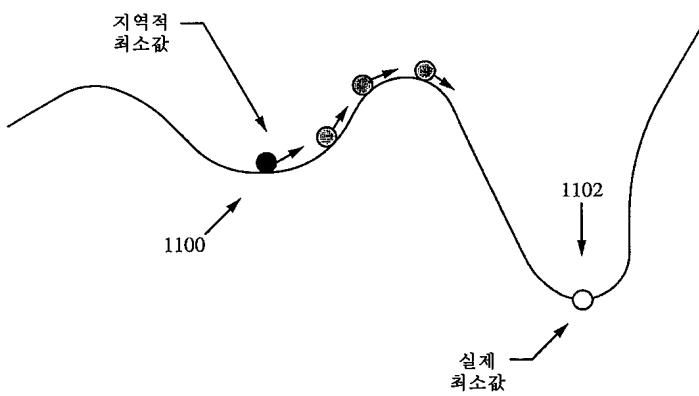
도면9



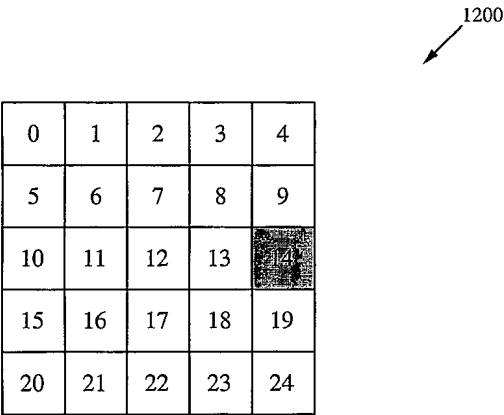
도면10



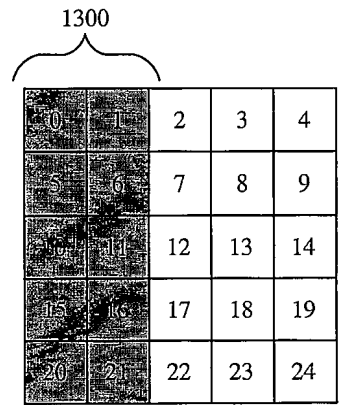
도면11



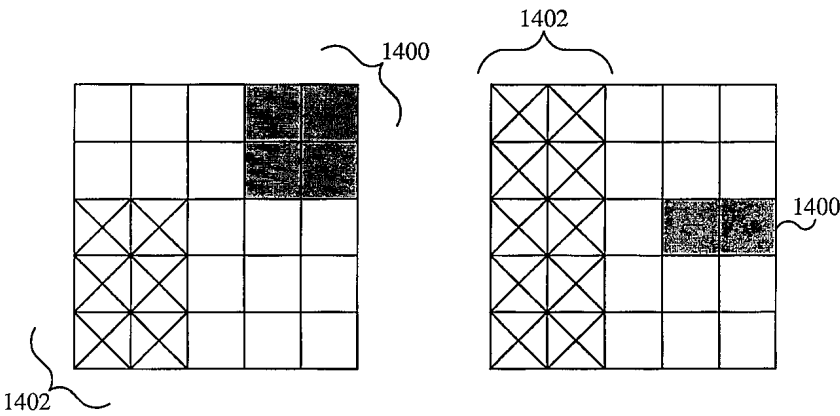
도면12



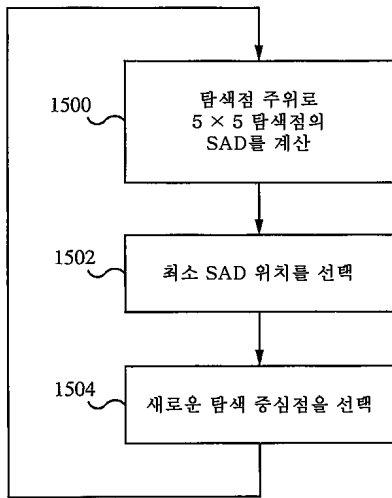
도면13



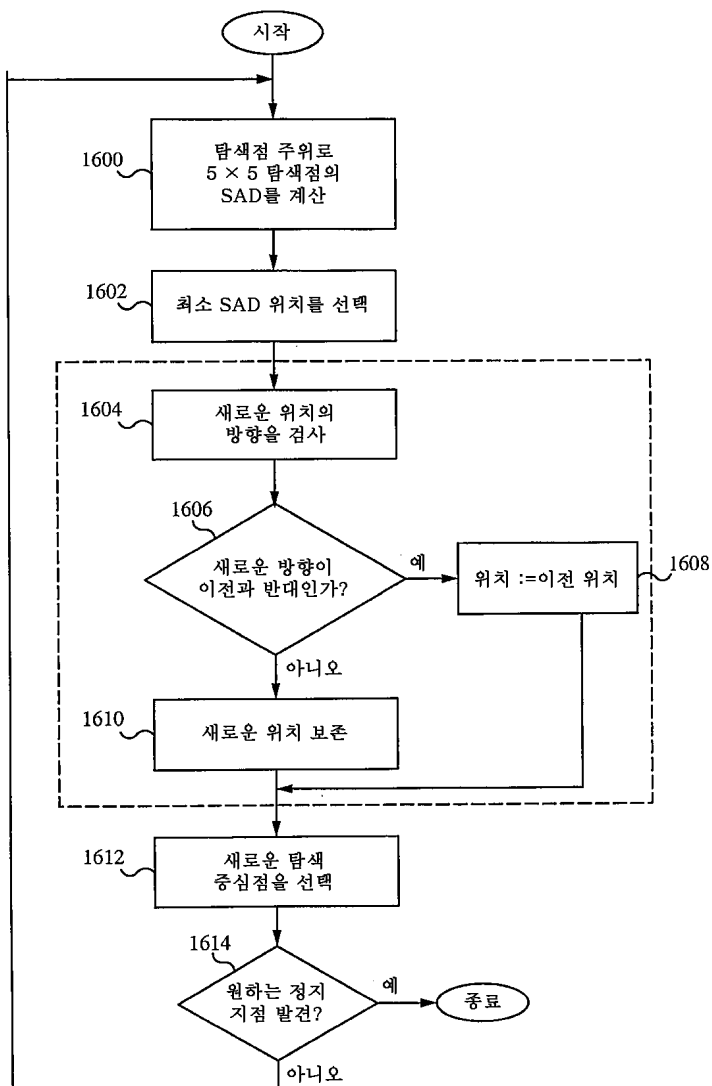
도면14



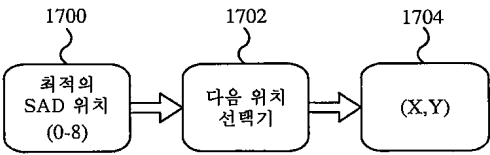
도면15



도면16



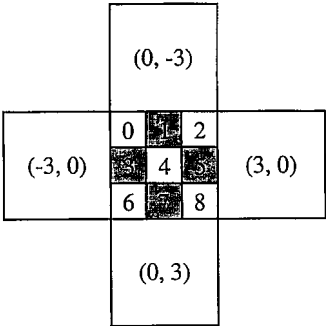
도면17



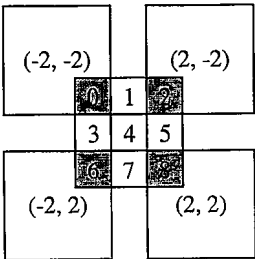
도면18

0	1	2
3	4	5
6	7	8

도면19

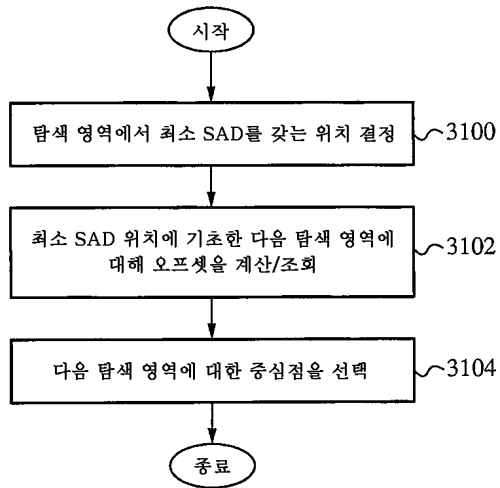


도면20





도면21

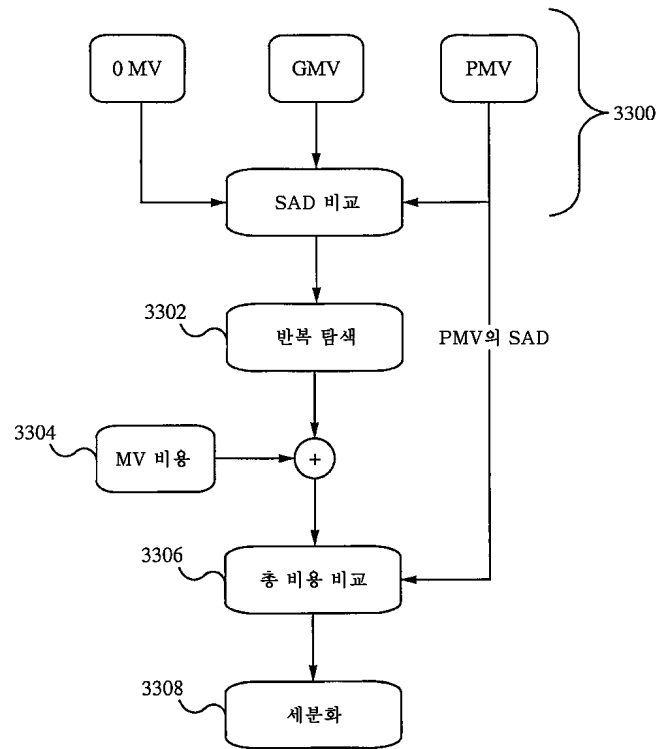


도면22

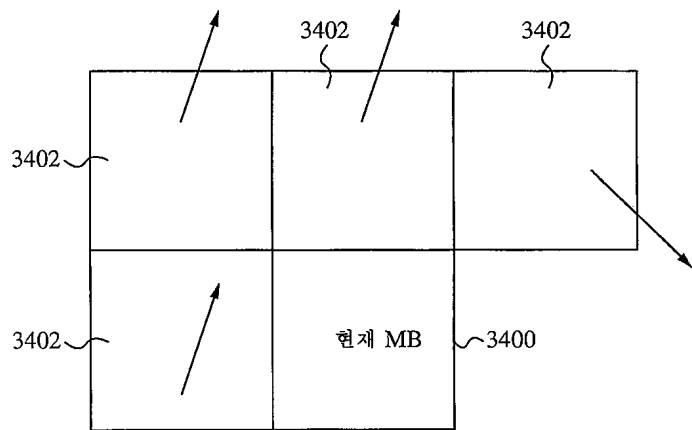
3200

최소 SAD 위치	다음 탐색 영역의 중심점으로 오프셋
0	(-2, -2)
1	(0, -3)
2	(2, -2)
3	(-3, 0)
4	-
5	(3, 0)
6	(-2, 2)
7	(0, 3)
8	(2, 2)

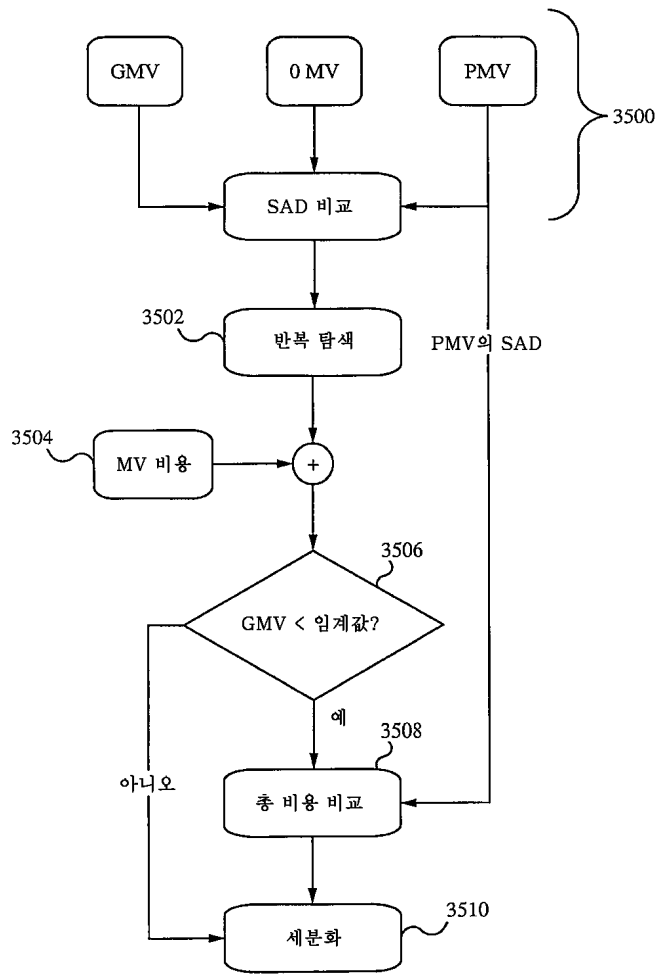
도면23



도면24



도면25



도면26

