



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년01월09일
(11) 등록번호 10-1350193
(24) 등록일자 2014년01월02일

- (51) 국제특허분류(Int. Cl.)
H04N 5/232 (2006.01) *H04N 5/225* (2006.01)
H04N 5/21 (2006.01)
- (21) 출원번호 10-2007-0010471
(22) 출원일자 2007년02월01일
심사청구일자 2012년01월17일
- (65) 공개번호 10-2007-0079324
(43) 공개일자 2007년08월06일
(30) 우선권주장
JP-P-2006-00024065 2006년02월01일 일본(JP)

(56) 선행기술조사문헌

- JP4389779 B2
JP10322585 A
JP2005229169 A
JP2006186481 A

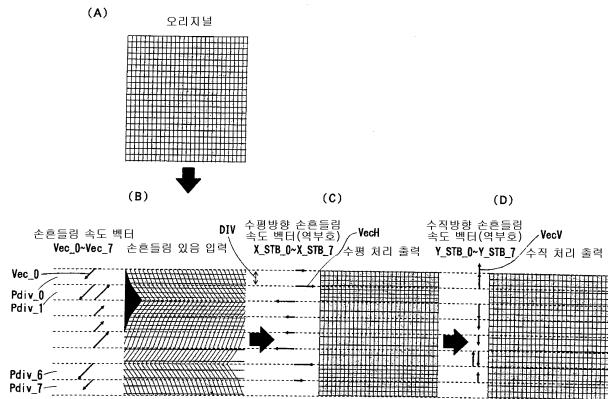
전체 청구항 수 : 총 29 항

심사관 : 김평수

(54) 발명의 명칭 **촬상 화상 신호의 일그러짐 보정 방법, 촬상 화상 신호의 일그러짐 보정 장치, 촬상 방법 및 촬상 장치****(57) 요약**

자이로 센서 등의 기구계(機構系) 부품을 사용하는 일없이, 디지털 신호 처리에 의해 고정밀도(高精度)이고 또한 저(低)코스트이며, 손흔들림(手振; hand movement) 및 CMOS 이미지 등에서 생기는 포커스 플레인(focal plane) 현상(現象)에 의한 촬상 화상의 일그러짐(歪; distortions)을 경감하는 방법을 제공하는 것을 과제로 한다.

본 발명에서는, 화상의 1화면 구간을 복수개(複數個)의 분할 화상 구간으로 분할한다. 분할 화상 구간의 각각에서의 화상의 움직임 벡터를, 주목 화면(注目畫面; observed screen)과 해당 주목 화면보다도 앞(前)의 원화면(元畫面; original screen)과의 사이의 분할 화상 구간의 화상 정보로부터 검출한다. 검출된 분할 화상 구간의 움직임 벡터의, 인접(隣接; adjacent)하는 분할 화상 구간의 것의 차분(差分; difference)을 구하는 것에 의해, 분할 화상 구간의 각각마다, 해당(當該) 분할 화상 구간의 화상의 일그러짐 변화의 속도를 검출한다. 복수개의 분할 화상 구간의 각각 내(内)에서는, 각 분할 화상 구간에 대해서 검출된 화상의 일그러짐 변화의 속도에 의거해서, 화상의 일그러짐을 보정한다.

대 표 도 - 도1

특허청구의 범위

청구항 1

1화면의 화상의 수평 방향 또는 수직 방향의 일그러짐(歪; distortions)을 보정하는 방법으로서, 상기 화상의 1화면 구간을 복수개(複數個)의 분할 화상 구간으로 분할하고, 상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를, 2화면 사이의 상기 분할 화상 구간의 화상 정보로부터 검출하고, 검출된 상기 복수개의 분할 화상 구간의 상기 움직임 벡터 중의, 상기 1화면 내에서 인접(隣接; adjacent)하는 분할 화상 구간의 것의 차분(差分; difference) 벡터를 구하고, 구해진 상기 차분 벡터에 의거해서, 상기 분할 화상 구간의 각각 내에서의 화상의 일그러짐을 보정하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 2

제1항에 있어서, 상기 차분 벡터를 구하는 것에 의해, 상기 분할 화상 구간의 각각마다, 해당 분할 화상 구간의 화상의 일그러짐 속도를 검출하고, 상기 각 분할 화상 구간 내에서는, 해당 각 분할 화상 구간에 대해서 검출된 상기 화상의 일그러짐 속도의 시간 적분값(積分值)을 상기 화상의 일그러짐 보정 대상 부위(補正對象部位)의 일그러짐 변위량(變位量)으로 하고, 상기 시간 적분값을 이용해서 상기 화상의 일그러짐을 보정하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 3

제2항에 있어서, 상기 화상의 일그러짐은, 촬영시의 활상 소자(素子)의, 활상 화상의 수평 방향 또는 수직 방향의 위치적 변화에 의한 활상 화상의 일그러짐인 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 4

제3항에 있어서, 상기 활상 소자는, 상기 활상 화상의 데이터를 화소 단위로 판독출력(讀出; read out)이 가능한 것으로서, 상기 각 분할 화상 구간 내에서는, 해당 각 분할 화상 구간의 선두 부위(先頭部位; head member)에 대한 일그러짐 보정 대상 부위의 판독출력 지연 시간을 이용해서 상기 화상의 일그러짐 속도를 시간 적분해서 얻어지는 변위량을, 상기 일그러짐 보정 대상 부위의 변위량으로 해서 상기 활상 화상의 일그러짐을 보정하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 5

제3항에 있어서, 상기 활상 소자로부터는 상기 활상 화상의 데이터를 라인 단위로 판독출력하는 것임과 동시에, 상기 분할은, 상기 각 분할 화상 구간이 복수 라인으로 이루어지도록 행하고, 상기 각 분할 화상 구간 내에서는, 각각의 상기 분할 화상 구간에 대해서 검출된 상기 화상 일그러짐 속도를 상기 라인 단위로 시간 적분해서 얻어지는 변위량을 각 라인 위치의 변위량으로 해서 상기 활상 화상의 일그러짐을 보정하는 것을 특징으로 하는 활상 화상 신호의 일그러짐 보정 방법.

청구항 6

제3항에 있어서,

상기 화상의 일그러짐 속도를, 수평 방향의 성분과 수직 방향의 성분으로 나누고,

상기 화상의 일그러짐 속도의 수평 방향 성분을 이용해서, 상기 화상의 수평 방향의 일그러짐을 보정하고, 상기 화상의 일그러짐 속도의 수직 방향 성분을 이용해서, 상기 화상의 수직 방향의 일그러짐을 보정하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 7

제5항에 있어서,

상기 화상의 일그러짐 속도를, 수평 방향의 성분과 수직 방향의 성분으로 나누고,

상기 각 분할 화상 구간 내에서는, 상기 화상의 일그러짐 속도의 수평 방향의 성분을, 상기 라인 단위로 시간 적분해서 얻어지는 상기 수평 방향의 변위량을 각 라인의 수평 방향의 변위량으로 해서 상기 화상의 수평 방향의 일그러짐을 보정함과 동시에, 상기 화상의 일그러짐 속도의 수직 방향의 성분을, 상기 라인 단위로 시간 적분해서 얻어지는 상기 수직 방향의 변위량을 각 라인의 수직 방향의 변위량으로 해서 상기 화상의 수직 방향의 일그러짐을 보정하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 8

제7항에 있어서,

상기 화상의 수평 방향의 일그러짐의 보정 처리와, 상기 화상의 수직 방향의 일그러짐의 보정 처리를 병행(並行)해서 행하도록 함과 동시에, 상기 화상의 수평 방향의 일그러짐의 보정 처리의 진척 정도(度合; degree)를 참조(參照; check)하면서 상기 수직 방향의 변위량의 시간 적분 처리를 하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 9

제1항에 있어서,

상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를 검출하는 방법은,

상기 움직임 벡터를 검출하는 2화면은, 주목 화면(注目畫面; observed screen)인 참조 화면과 해당 참조 화면보다도 앞(前)의 원화면(元畫面; original screen)이고,

상기 원화면 중(中)의 상기 각 분할 화상 구간에서, 소정의 위치에 복수의 화소로 이루어지는 소정의 크기의 적어도 1개(個)의 타겟 블록을 설정하고,

상기 타겟 블록과 같은(同) 크기의 참조 블록을, 상기 참조 화면에 설정된 서치 범위에서 복수개 설정하고,

상기 복수개의 참조 블록 내로부터, 상기 타겟 블록과 상관(相關)이 강한 상기 참조 블록을 검출하고,

해당 검출한 참조 블록의 상기 타겟 블록에 대한 화면 위(上)의 위치 어긋남량(shift over)에 의거해서, 상기 움직임 벡터를 검출하는

것인 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 10

제9항에 있어서,

상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를 검출하는 방법은,

상기 참조 블록의 각각에서, 해당 참조 블록 내의 각 화소의 화소값과, 상기 타겟 블록 내에서 대응하는 위치의 각 화소의 화소값과의 차분의 절대값(絕對值)의 총합(總和; total of sum)을 구하는 차분 절대값합 산출(算出)

공정과,

상기 참조 블록의 각각의 상기 참조 화면 위의 위치의, 상기 타겟 블록의 화면 위의 위치와의 위치 어긋남량을, 방향 성분도 포함하는 참조 벡터로 하고, 해당 참조 벡터를 소정의 축소율로 축소한 참조 축소 벡터를 얻는 참조 축소 벡터 취득 공정과,

상기 참조 축소 벡터에 따른 크기의 상기 참조 벡터를 상기 위치 어긋남량으로 하는, 상기 소정의 축소율에 따라서 삽입된 수의 복수개의 상기 참조 블록의 각각에 대한 차분 절대값합을 기억하는 축소 차분 절대값합 테이블을 생성하는 테이블 생성 공정과,

상기 축소 차분 절대값합 테이블에서의 상기 차분 절대값합의 최소값(最小值)에 대응하는 상기 참조 벡터를 적어도 이용해서, 상기 참조 화면과 상기 원화면과의 사이의 상기 분할 화상 구간의 각각에 대한 움직임 벡터를 산출하는 움직임 벡터 산출 공정

을 구비하고,

상기 테이블 생성 공정은,

상기 참조 축소 벡터 취득 수단에서 취득된 상기 참조 축소 벡터의 근방값(近傍值; neighborhood value)으로 되는 복수의 상기 참조 벡터를 검출하는 근방 참조 벡터 검출 공정과,

상기 차분 절대값합 산출 공정에서 산출된 상기 참조 블록의 각각에 대한 상기 차분 절대값의 총합으로부터, 상기 근방 참조 벡터 검출 공정에서 검출된 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 차분 절대값합의 각각을 산출하는 분산(分散) 차분 절대값합 산출 공정과,

상기 분산 차분 절대값합 산출 공정에서 산출된 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합을, 그때까지의 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합에 가산(加算; add)하는 분산 가산 공정

을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 11

제9항 또는 제10항에 있어서,

상기 각 분할 화상 구간은, 또 재분할(再分割)됨과 동시에, 해당 재분할 화상 구간마다 상기 타겟 블록이 설정되어, 각각의 타겟 블록에 대한 상기 움직임 벡터가 검출되고,

상기 각 분할 화상 구간에 대해서 얻어지는 복수의 타겟 블록에 대한 복수의 상기 움직임 벡터로부터, 상기 각 분할 화상 구간에 대한 상기 움직임 벡터를 검출하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 12

제11항에 있어서,

상기 각 분할 화상 구간에 대한 상기 복수의 타겟 블록에 대한 복수의 상기 움직임 벡터의 평균값(平均值)을, 상기 각 분할 화상 구간에 대한 상기 움직임 벡터로서 검출하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 13

제12항에 있어서,

상기 복수의 타겟 블록에 대한 상기 움직임 벡터의 평균값을 구할 때에, 각각의 상기 타겟 블록에 대한 상기 움직임 벡터가 예외(例外)로 되는 움직임 벡터인지 여부를 판정하고, 예외라고 판정된 상기 타겟 블록에 대한 상기 움직임 벡터는, 상기 평균값을 구하는 연산으로부터 배제(排除)하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 14

주목 화면의 화상의 1화면 구간을 복수개의 분할 화상 구간으로 분할하고,

상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를, 상기 주목 화면과 해당 주목 화면보다도 앞의 원화면과의 사이의 상기 분할 화상 구간의 화상 정보로부터 검출하고,

검출된 상기 복수개의 분할 화상 구간의 상기 움직임 벡터 중의, 상기 1화면내에서 인접하는 분할 화상 구간의 것의 차분 벡터를 구하고, 구해진 상기 차분 벡터에 의거해서, 상기 주목 화면의 상기 분할 화상 구간의 각각 내에서의 수평 방향 또는 수직 방향의 화상의 일그러짐을 보정하는 방법에 있어서,

상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를 검출하는 방법은,

상기 원화면 중의 상기 각 분할 화상 구간에서, 소정의 위치에 복수의 화소로 이루어지는 소정의 크기의 적어도 1개의 타겟 블록을 설정하고,

상기 타겟 블록과 같은 크기의 참조 블록을, 상기 주목 화면에 설정된 서치 범위에서 복수개 설정하고,

상기 복수개의 참조 블록 내로부터, 상기 타겟 블록과 상관이 강한 상기 참조 블록을 검출하고,

해당 검출한 참조 블록의 상기 타겟 블록에 대한 화면 위의 위치 어긋남량에 의거해서, 상기 움직임 벡터를 검출하는

것으로서,

상기 참조 블록의 각각에서, 해당 참조 블록 내의 각 화소의 화소값과, 상기 타겟 블록 내에서 대응하는 위치의 각 화소의 화소값과의 차분의 절대값의 총합을 구하는 차분 절대값합 산출 공정과,

상기 참조 블록의 각각의 상기 참조 화면 위의 위치의, 상기 타겟 블록의 화면 위의 위치와의 위치 어긋남량을, 방향 성분도 포함하는 참조 벡터로 하고, 해당 참조 벡터를 소정의 축소율로 축소한 참조 축소 벡터를 얻는 참조 축소 벡터 취득 공정과,

상기 참조 축소 벡터에 따른 크기의 상기 참조 벡터를 상기 위치 어긋남량으로 하는, 상기 소정의 축소율에 따라서 삽감된 수의 복수개의 상기 참조 블록의 각각에 대한 차분 절대값합을 기억하는 축소 차분 절대값합 테이블을 생성하는 테이블 생성 공정과,

상기 축소 차분 절대값합 테이블에서의 상기 차분 절대값합의 최소값에 대응하는 상기 참조 벡터를 적어도 이용해서, 상기 참조 화면과 상기 원화면과의 사이의 상기 분할 화상 구간의 각각에 대한 움직임 벡터를 산출하는 움직임 벡터 산출 공정

을 구비하고,

상기 테이블 생성 공정은,

상기 참조 축소 벡터 취득 공정에서 취득된 상기 참조 축소 벡터의 근방값으로 되는 복수의 상기 참조 벡터를 검출하는 근방 참조 벡터 검출 공정과,

상기 차분 절대값합 산출 공정에서 산출된 상기 참조 블록의 각각에 대한 상기 차분 절대값의 총합으로부터, 상기 근방 참조 벡터 검출 공정에서 검출된 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 차분 절대값합의 각각을 산출하는 분산 차분 절대값합 산출 공정과,

상기 분산 차분 절대값합 산출 공정에서 산출된 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합을, 그때까지의 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합에 가산하는 분산 가산 공정

을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법.

청구항 15

촬영시의 손흔들림(手振; hand movement)에 의한 활상 소자의 위치적 변화에 의한 활상 화상의 일그러짐을 보정하고, 보정 후의 활상 화상의 정보를 기록하도록 하는 활상 방법으로서,

상기 활상 화상의 1화면 구간을 복수개의 분할 화상 구간으로 분할하고,

상기 분할 화상 구간의 각각에서의 상기 활상 화상의 움직임 벡터를, 2화면 사이의 상기 분할 화상 구간의 활상

화상 정보로부터 검출하고,

검출된 상기 분할 화상 구간의 상기 움직임 벡터 중의, 상기 1화면 내에서 인접하는 분할 화상 구간의 것의 차분 벡터를 구하고, 구해진 상기 차분 벡터에 의해, 상기 분할 화상 구간의 각각 내에서의 활상 화상의 일그러짐을 보정하고,

상기 보정한 상기 활상 화상의 화상 정보를 기록 매체에 기록하는

것을 특징으로 하는 활상 방법.

청구항 16

1화면의 화상의 수평 방향 또는 수직 방향의 일그러짐을 보정하는 장치로서,

상기 1화면의 화상 영역이 복수개로 분할된 각 분할 화상 구간마다, 주목 화면과 해당 주목 화면보다도 앞의 원화면과의 사이의 상기 분할 화상 구간의 화상 정보로부터 움직임 벡터를 검출하는 움직임 벡터 검출 수단과,

상기 움직임 벡터 검출 수단에서 검출된 상기 분할 화상 구간의 상기 움직임 벡터 중의, 상기 1화면 내에서 인접하는 분할 화상 구간의 것의 차분 벡터를 구하는 차분 벡터 검출 수단과,

상기 각 분할 화상 구간 내에서, 상기 각 분할 화상 구간에 대해서 상기 차분 벡터 검출 수단에서 검출된 상기 차분 벡터에 의거해서, 상기 화상의 일그러짐을 보정하는 일그러짐 보정 처리 수단

을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 17

제16항에 있어서,

상기 일그러짐 보정 처리 수단은,

상기 차분 벡터에 의해, 상기 분할 화상 구간의 각각마다, 해당 분할 화상 구간의 화상의 일그러짐 속도를 검출하고,

상기 각 분할 화상 구간 내에서는, 상기 각 분할 화상 구간에 대해서 검출된 상기 화상의 일그러짐 속도의 시간 적분값을 상기 화상의 일그러짐 보정 대상 부위의 일그러짐 변화량으로서 구하는 변위량 적분 수단과,

상기 변위량 적분 수단에서 구해진 상기 시간 적분값을 이용해서 상기 화상의 일그러짐을 보정하는 보정 수단을 구비하는 것을 특징으로 하는 활상 화상 신호의 일그러짐 보정 장치.

청구항 18

제17항에 있어서,

상기 화상의 일그러짐은, 활영시의 활상 소자의, 활상 화상의 수평 방향 또는 수직 방향의 위치적 변화에 의한 활상 화상의 일그러짐인

것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 19

제18항에 있어서,

상기 활상 소자는, 상기 활상 화상의 데이터를 화소 단위로 판독출력이 가능한 것으로서,

상기 일그러짐 보정 처리 수단은,

상기 각 분할 화상 구간 내에서는, 해당 각 분할 화상 구간의 선두 부위에 대한 일그러짐 보정 대상 부위의 판독출력 지연 시간을 이용해서, 상기 화상의 일그러짐 속도를 시간 적분하고, 그 시간 적분값을 상기 일그러짐 보정 대상 부위의 변위량으로서 구하는 변위량 적분 수단과,

상기 변위량 적분 수단에서 얻은 상기 변위량을 이용해서 상기 활상 화상의 일그러짐을 보정하는 보정 수단을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 20

제18항에 있어서,

상기 활상 소자로부터는 상기 활상 화상의 데이터를 라인 단위로 판독출력하는 것임과 동시에, 상기 분할은, 상기 각 분할 화상 구간이 복수 라인으로 이루어지도록 행하는 것으로서,

상기 일그러짐 보정 처리 수단은,

상기 각 분할 화상 구간 내에서, 상기 각 분할 화상 구간에 대해서 검출된 상기 화상의 일그러짐 속도를, 상기 라인 단위로 시간 적분하고, 그 시간 적분값을 상기 일그러짐 보정 대상 라인의 변위량으로서 구하는 변위량 적분 수단과,

상기 변위량 적분 수단에서 얻은 상기 변위량을 각 라인의 변위량으로 해서 상기 활상 화상의 일그러짐을 보정하는 보정 수단

을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 21

제18항에 있어서,

상기 일그러짐 보정 처리 수단은,

상기 화상의 일그러짐 속도 검출 수단에서 검출된 상기 화상의 일그러짐 속도의 수평 방향 성분을 이용해서, 상기 화상의 수평 방향의 일그러짐을 보정하는 수평 보정 처리 수단과,

상기 화상의 일그러짐 속도의 수직 방향 성분을 이용해서, 상기 화상의 수직 방향의 일그러짐을 보정하는 수직 보정 처리 수단

을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 22

제20항에 있어서,

상기 일그러짐 보정 처리 수단의 상기 변위량 적분 수단은,

상기 각 분할 화상 구간 내에서, 상기 화상의 일그러짐 속도의 수평 방향의 성분을, 상기 라인 단위로 시간 적분해서, 그 시간 적분값을 상기 일그러짐 보정 대상 라인의 수평 방향의 변위량으로서 구하는 수평 방향 변위량 산출 수단과,

상기 각 분할 화상 구간 내에서, 상기 화상의 일그러짐 속도의 수직 방향의 성분을, 상기 라인 단위로 시간 적분해서, 그 시간 적분값을 상기 일그러짐 보정 대상 라인의 수직 방향의 변위량으로서 구하는 수직 방향 변위량 산출 수단

을 구비하고,

상기 일그러짐 보정 처리 수단의 상기 보정 수단은,

상기 각 분할 화상 구간 내에서, 상기 수평 방향 변위량 산출 수단에서 얻은 상기 수평 방향의 변위량을 각 라인의 수평 방향의 변위량으로 해서 상기 화상의 수평 방향의 일그러짐을 보정하는 수평 보정 처리 수단과,

상기 각 분할 화상 구간 내에서, 상기 수직 방향 변위량 산출 수단에서 얻은 상기 수직 방향의 변위량을 각 라인의 수직 방향의 변위량으로 해서 상기 화상의 수직 방향의 일그러짐을 보정하는 수직 보정 처리 수단

을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 23

제16항에 있어서,

상기 움직임 벡터 검출 수단은, 상기 분할 화상 구간의 각각에서, 주목 화면인 참조 화면과, 해당 참조 화면보다도 앞의 원화면과의 사이에서의 움직임 벡터를 산출하는 수단으로서,

상기 원화면 중의 상기 각 분할 화상 구간에서, 소정의 위치에 복수의 화소로 이루어지는 소정의 크기의 적어도 1개의 타겟 블록이 설정됨과 동시에, 상기 타겟 블록과 같은 크기의 참조 블록이, 상기 주목 화면에 설정된 서치 범위에서 복수개 설정되고,

상기 복수개의 참조 블록 내로부터, 상기 타겟 블록과 상관이 강한 상기 참조 블록을 검출하고,

해당 검출한 참조 블록의 상기 타겟 블록에 대한 화면 위의 위치 어긋남량에 의거해서, 상기 움직임 벡터를 검출하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 24

제23항에 있어서,

상기 움직임 벡터 검출 수단은,

상기 참조 블록의 각각에서, 해당 참조 블록 내의 각 화소의 화소값과, 상기 타겟 블록 내에서 대응하는 위치의 각 화소의 화소값과의 차분의 절대값의 총합을 구하는 차분 절대값합 산출 수단과,

상기 참조 블록의 각각의 상기 참조 화면 위의 위치의, 상기 타겟 블록의 화면 위의 위치와의 위치 어긋남량을, 방향 성분도 포함하는 참조 벡터로 하고, 해당 참조 벡터를 소정의 축소율로 축소한 참조 축소 벡터를 얻는 참조 축소 벡터 취득 수단과,

상기 참조 축소 벡터에 따른 크기의 상기 참조 벡터를 상기 위치 어긋남량으로 하는, 상기 소정의 축소율에 따라서 삽감된 수의 복수개의 상기 참조 블록의 각각에 대한 차분 절대값합을 기억하는 축소 차분 절대값합 테이블을 생성하는 테이블 생성 수단과,

상기 축소 차분 절대값합 테이블에서의 상기 차분 절대값합의 최소값에 대응하는 상기 참조 벡터를 적어도 이용해서, 상기 참조 화면과 상기 원화면과의 사이의 상기 분할 화상 구간의 각각에 대한 움직임 벡터를 산출하는 움직임 벡터 산출 수단

을 구비하고,

상기 테이블 생성 수단은,

상기 참조 축소 벡터 취득 수단에서 취득된 상기 참조 축소 벡터의 근방값으로 되는 복수의 상기 참조 벡터를 검출하는 근방 참조 벡터 검출 수단과,

상기 차분 절대값합 산출 수단에서 산출된 상기 참조 블록의 각각에 대한 상기 차분 절대값의 총합으로부터, 상기 근방 참조 벡터 검출 공정에서 검출된 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 차분 절대값합의 각각을 산출하는 분산 차분 절대값합 산출 수단과,

상기 분산 차분 절대값합 산출 수단에서 산출한 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합을, 그때까지의 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합에 가산하는 분산 가산 수단

을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 25

제22항 또는 제23항에 있어서,

상기 움직임 벡터 검출 수단은,

상기 각 분할 화상 구간을, 또 재분할함과 동시에, 해당 재분할 화상 구간마다 상기 타겟 블록을 설정해서, 각각의 타겟 블록에 대한 상기 움직임 벡터를 검출하고,

상기 각 분할 화상 구간에 대해서 얻어지는 복수의 타겟 블록에 대한 복수의 상기 움직임 벡터로부터, 상기 각 분할 화상 구간에 대한 상기 움직임 벡터를 검출하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 26

제25항에 있어서,

상기 움직임 벡터 검출 수단은,

상기 각 분할 화상 구간에 대한 상기 복수의 타겟 블록에 대한 복수의 상기 움직임 벡터의 평균값을, 상기 각 분할 화상 구간에 대한 상기 움직임 벡터로서 검출하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 27

제26항에 있어서,

상기 움직임 벡터 검출 수단은,

상기 복수의 타겟 블록에 대한 상기 움직임 벡터의 평균값을 구할 때에, 각각의 상기 타겟 블록에 대한 상기 움직임 벡터가 예외로 되는 움직임 벡터인지 여부를 판정하고, 예외라고 판정된 상기 타겟 블록에 대한 상기 움직임 벡터는, 상기 평균값을 구하는 연산으로부터 배제하는

것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 28

1화면의 화상의 1화면 구간을 복수개의 분할 화상 구간으로 분할하고, 상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를, 주목 화면인 참조 화면과 해당 참조 화면보다도 앞의 원화면과의 사이의 상기 분할 화상 구간의 화상 정보로부터 검출하고, 검출된 상기 분할 화상 구간의 상기 움직임 벡터 중의, 상기 1화면 내에서 인접하는 분할 화상 구간의 것의 차분 벡터를 구하고, 구해진 상기 차분 벡터에 의거해서, 상기 주목 화면의 상기 분할 화상 구간의 각각 내에서의 수평 방향 또는 수직 방향의 화상의 일그러짐을 보정하는 장치로서,

상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를 검출하는 움직임 벡터 검출 수단은, 상기 원화면 중의 상기 각 분할 화상 구간에서, 소정의 위치에 복수의 화소로 이루어지는 소정의 크기의 적어도 1개의 타겟 블록을 설정함과 동시에, 상기 타겟 블록과 같은 크기의 참조 블록을, 상기 주목 화면에 설정된 서치 범위에서 복수개 설정하고, 상기 복수개의 참조 블록 내로부터, 상기 타겟 블록과 상관이 강한 상기 참조 블록을 검출하고, 해당 검출한 참조 블록의 상기 타겟 블록에 대한 화면 위의 위치 어긋남량에 의거해서, 상기 움직임 벡터를 검출하는 것인 화상 신호의 일그러짐 보정 장치에 있어서,

상기 움직임 벡터 검출 수단은,

상기 참조 블록의 각각에서, 해당 참조 블록 내의 각 화소의 화소값과, 상기 타겟 블록 내에서 대응하는 위치의 각 화소의 화소값과의 차분의 절대값의 총합을 구하는 차분 절대값합 산출 수단과,

상기 참조 블록의 각각의 상기 참조 화면 위의 위치의, 상기 타겟 블록의 화면 위의 위치와의 위치 어긋남량을, 방향 성분도 포함하는 참조 벡터로 하고, 해당 참조 벡터를 소정의 축소율로 축소한 참조 축소 벡터를 얻는 참조 축소 벡터 취득 수단과,

상기 참조 축소 벡터에 따른 크기의 상기 참조 벡터를 상기 위치 어긋남량으로 하는, 상기 소정의 축소율에 따라서 삽감된 수의 복수개의 상기 참조 블록의 각각에 대한 차분 절대값합을 기억하는 축소 차분 절대값합 테이블을 생성하는 테이블 생성 수단과,

상기 축소 차분 절대값합 테이블에서의 상기 차분 절대값합의 최소값에 대응하는 상기 참조 벡터를 적어도 이용해서, 상기 참조 화면과 상기 원화면과의 사이의 상기 분할 화상 구간의 각각에 대한 움직임 벡터를 산출하는 움직임 벡터 산출 수단

을 구비하고,

상기 테이블 생성 수단은,

상기 참조 축소 벡터 취득 수단에서 취득된 상기 참조 축소 벡터의 근방값으로 되는 복수의 상기 참조 벡터를 검출하는 근방 참조 벡터 검출 수단과,

상기 차분 절대값합 산출 공정에서 산출된 상기 참조 블록의 각각에 대한 상기 차분 절대값의 총합으로부터, 상기 근방 참조 벡터 검출 공정에서 검출된 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 차분 절대값합의 각각을 산출하는 분산 차분 절대값합 산출 수단과,

상기 분산 차분 절대값합 산출 수단에서 산출한 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합을, 그때까지의 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합에 가산하는 분산 가산 수단

을 구비하는 것을 특징으로 하는 화상 신호의 일그러짐 보정 장치.

청구항 29

촬영시의 손흔들림에 의한 활상 소자의 위치적 변화에 의한 활상 화상의 일그러짐을 보정하고, 보정 후의 활상 화상의 정보를 기록하도록 하는 활상 장치로서,

상기 활상 화상의 1화면 영역이 복수개로 분할된 각 분할 화상 구간마다, 2화면 사이의 상기 분할 화상 구간의 활상 화상 정보로부터 움직임 벡터를 검출하는 움직임 벡터 검출 수단과,

상기 움직임 벡터 검출 수단에서 검출된 상기 분할 화상 구간의 상기 움직임 벡터 중의, 상기 1화면 내에서 인접하는 분할 화상 구간의 것의 차분 벡터를 구하는 차분 벡터 검출 수단과,

상기 각 분할 화상 구간 내에서, 상기 각 분할 화상 구간에 대해서 상기 차분 벡터 검출 수단에서 검출된 상기 차분 벡터에 의거해서, 상기 활상 화상의 일그러짐을 보정하는 일그러짐 보정 처리 수단과,

상기 일그러짐 보정 수단에서 보정된 상기 활상 화상의 화상 정보를 기록 매체에 기록하는 기록 수단을 구비하는 것을 특징으로 하는 활상 장치.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0064] 본 발명은, 활상 장치에서의 피사체 화상의 활상시의 손흔들림(手振; hand movement) 등에 의한 활상 화상에 발생하는 일그러짐(歪; distortions)을 보정하는 방법 및 해당(當該) 일그러짐을 보정하는 장치에 관한 것으로서, 특히 X-Y 어드레스형(型)의 고체 활상 소자(素子), 예를 들면 CMOS(Complementary Metal Oxide Semiconductor)형의 고체 활상 소자(이하, CMOS 이미저라고 한다)를 이용하는 비디오 카메라, 디지털 카메라 등의 활상 장치나, 활상 기능을 가지는(機能付) 기록 재생 장치에 적용해서 매우 적합(好適)한 것이다.

[0065] 전자식(電子式)으로 활상 화상을 활상 소자에 기억하고, 판독출력(讀出; read out)하도록 하는 전자식 활상 장치에서는, 활영자에 의한 이른바 손흔들림이나, 배(船) 위(上) 등 불안정한 장소에서의 활영 등의 이유에 의해, 활영시에, 활상 소자가, 물리적으로, 활상 화상의 수평 방향 및/또는 수직 방향으로 비교적 고속의 위치적 변화를 일으키면, 그 활상 화상에 일그러짐을 일으킨다.

[0066] 그러나, 전자식 활상 장치에서는, 활상 소자로부터 판독출력한 활상 화상 데이터에 대해서 디지털 처리(일그러짐 보정 처리)하는 것에 의해, 이 손흔들림 등에 기인(起因)하는 활상 화상의 일그러짐을 보정하는 것이 가능하다.

[0067] 종래의 비디오 카메라나 디지털 카메라 등의 활상 장치나, 활상 기능을 가지는 기록 재생 장치(카메라 달린(付; having, including) 휴대 전화 단말이나 카메라 달린 퍼스널 컴퓨터 등)에서는, 활상 소자로서, 일반적으로 CCD(Charge Coupled Device)를 이용한 고체 활상 소자(이하, CCD를 이용한 고정 활상 소자를 CCD 이미저라고 한다)가 널리(廣) 이용되고 있다.

[0068] 그래서, 종래에는, 예를 들면 특허 문헌 1(미국 특허 제3, 384, 459호 공보) 등과 같이, 활상 소자로서 CCD 이미저를 이용하는 것을 전제로 한 손흔들림 보정 기술이 수많이 제안되어 있으며, 이미 해당 손흔들림 보정 기술을 탑재(搭載; adopting)한 제품으로서도 세상에 널리 나돌고(出回; used) 있다.

- [0069] 이 종래의 손흔들림 보정 기술은, CCD 이미저의 구조상, CCD 이미저의 전화소(全畫素)에 축적된 광량의 샘플링 시각이, 동일(1프레임에 1회(回))인 특성을 이용하고 있다.
- [0070] 즉, CCD 이미저에서는, 전화소를 같은(同) 시기에 노광(露光)하고, 1프레임의 화상 데이터에 대해서는, 데이터 취출(取出) 타이밍이 완전히 동일(同一)하므로, 손흔들림 변위량(變位量)으로서는, 도 58에서 화살표로 나타내는 바와 같이, 1프레임분(分)의 전화소에 대해서, 하나의 손흔들림 변위량 V_{cs} 를 생각하면 좋다. 다시말해, 도 58에서, 본래(本來), 피사체는 실선으로 나타내는 영역 FLa 에 축적되는 것이었는데, 손흔들림에 의해 점선으로 나타내는 영역 FLb 로 이동한 경우에는, 해당 프레임에서 하나의 손흔들림 변위량 V_{cs} 를 검출하고, 해당 손흔들림량 V_{cs} 만큼, 판독출력 화소 위치(샘플링 화소 위치)를 보정하는 것에 의해, 손흔들림에 기인하는 활상 화상 일그러짐을 보정할 수가 있다.
- [0071] 또한, 도 58에 도시하는 바와 같이, 일반적으로, 활상 소자에서는, 그의 전화소를 유효 화소로서 취급하는 것이 아니라, 전화소로 이루어지는 영역(이하, 실효(實效) 화상 영역이라고 한다) AFL 중의, 주변(周邊)의 영역을 제외한, 수평 유효 영역 및 수직 유효 영역으로 정해지는 중앙부를 유효 화상 영역 EFL로서 이용하고 있는 것이 많다.
- [0072] 이와 같은 이미저를 이용한 경우, 손흔들림 보정에 의해 판독출력 화소 위치가 변화해도, 손흔들림량이, 실효 화상 영역 AFL과 유효 화상 영역 EFL과의 차분(差分; difference)보다도 작은 범위에서는, 이미저가 원래(元元) 가지고 있는 화소의 데이터를 이용해서 일그러짐 보정을 할 수 있으므로, 보간(補間; interpolation) 처리 등에 의해, 손흔들림 보정에 필요한 데이터를 생성하는 경우에 비해서, 화상의 열화(劣化; deterioration)는 적어진다.
- [0073] 그런데, 최근에는, 상술한 바와 같은 전자식 활상 장치의 활상 소자로서, 화면의 수평 방향(X방향)의 위치와, 수직 방향(Y방향) 위치를 지정하는 것에 의해, 화소 단위로 활상 데이터를 판독출력하는 것이 가능한 X-Y 어드레스형 고체 활상 소자, 예를 들면 CMOS형 고체 활상 소자(이하, CMOS 이미저라고 한다)가 이용되도록 되어 있다.
- [0074] 이 CMOS 이미저는,
- [0075] (a) 증폭형(增幅型)이며, 증폭한 신호를 판독출력하므로, 고감도(高感度)이다.
- [0076] (b) CMOS 회로를 이용하므로 저소비 전력이다,
- [0077] (c) 저(低)코스트이다,
- [0078] (d) 원리적으로, 1화소 단위로 랜덤하게 액세스하는(판독출력하는) 것이 가능하다,
- [0079] 는 등의 특장(特長)을 가진다.
- [0080] CMOS 이미저는, 1화소 단위로도 활상 화상 데이터의 판독출력은 가능하지만, 실용상은, 1수평 라인분(分)의 화소군(畫素群) 단위로 판독출력(샘플링)해서, 활상 화상 데이터를 출력하는 것이 일반적이다.
- [0081] 이와 같이, CMOS 이미저로부터 활상 데이터를 수평 라인 단위로 판독출력하는 경우에는, 도 59에 도시하는 바와 같이, 각 수평 라인에 대한 노광 기간은, 수평 라인 단위의 판독출력 시간차에 따른 Δt 만큼, 시간적인 어긋남(shift)을 일으킨다. 또한, 1화소 단위로 활상 화상 데이터의 판독출력이 이루어지는 경우이더라도, 라인 사이의 판독출력 시간차에 비해서, 화소 사이의 판독출력 시간차는 무시할 수 있을 정도로 작다. 그래서, 1화소 단위로 활상 화상 데이터의 판독출력이 이루어지는 경우이더라도, 마찬가지 노광 기간의 시간적인 어긋남을 일으키는 것으로 할 수가 있다.
- [0082] 이 때문에, CMOS 이미저를 이용한 활상 장치에 의해, 예를 들면 달리는 전차(電車) 안에서 바깥의 경치(景色)를 촬영하면, 본래는 도 60의 (A)와 같은 화상이 얻어져야 하는 것인데, 도 60의 (B)에 도시하는 바와 같이, 본래 연직(鉛直; 수직 방향)으로 서 있는 집이나 나무가 기울어진(傾; inclined) 화상이 출력 활상 화상으로 되어 버린다. 이것이 CMOS 이미저 특유의 포컬 플레인 (focal plane) 현상(現象)이다.
- [0083] 도 60의 (B)의 화상에는, 수평 방향으로 이동하면서 촬영한 것이기 때문에, 피사체가 기울어져서 활상되지만, 예를 들면 수직 방향으로 이동하면서 촬영한 경우에는, 도시는 생략하지만, 피사체가 수직 방향으로 줄어들거나, 늘어나거나 하는 바와 같은 화상으로 된다.
- [0084] 이 현상이 발현(發現; 발생)하는 것은, CMOS 이미저를 사용하는 활상 장치를 들고 있는(持; holding) 촬영자가

고속으로 이동하면서 촬영하거나, 역(逆)으로, 고정 위치에 있는 촬영자가 고속으로 이동하는 피사체를 촬영하거나 한 경우이며, 촬상 장치와 피사체의 상대 속도가 빠른 경우에 현저하게 된다. 단, 일반적인 촬영에서, 이와 같은 상황은 드물다(稀)고 할 수 있다.

[0085] 그렇지만, 촬영자가 촬상 장치를 손으로 들고서 촬영하고 있는 경우에, 촬상 장치를 들고 있는 촬영자의 손이, 미소(微小)하게 또한 고속으로 진동(振動; vibration)한 경우, 즉 손흔들림이 발생한 경우, 상술한 포컬 플레인 현상이 발현하게 된다.

[0086] 왜냐하면, CMOS 이미저인 경우의 손흔들림(이하, CMOS 손흔들림이라고 한다)의 값(值)은, CCD 이미저와 같이 1프레임 내(内)에서 1개(個)의 값이 아니라, 전술한 바와 같이, 1프레임 내에서의 화소나 라인마다의 샘플링 시각의 틀림(違; difference; 차이)에 의해, 화소나 수평 라인마다 다른(異) 것으로 된다. 이 때문에, CMOS 이미저를 이용한 촬상 장치에서는, 단지 1프레임 단위로의 손흔들림량을 이용한 보정을 행해도, 전술한 포컬 플레인 현상에 의한 일그러짐은, 보정되지 않고 남아 버린다.

[0087] 이 때, 손흔들림의 방향, 크기, 속도가, 1프레임 내(1매(枚))의 촬상 화상 내)에서 일률적(一律; uniform)이지 않은 것이 원인으로 되어, CMOS 이미저를 이용한 촬상 장치에서, CMOS 손흔들림이 생겼을 때의 피사체의 촬상 화상 출력은, 도 60의 (C)에 도시하는 바와 같이, 는적은적한(squishy; 흐물흐물한) 기묘(奇妙)한 화상 일그러짐이 생긴 것으로 된다.

[0088] 그런데, 디지털 카메라와 같이 정지화상(靜止畫; still picture)을 촬영하는 장치에서는, 원래 정지화상 촬영의 전제로서 손흔들림량이 한정되어 있는 것과, 메카 셔터(mechanical shutter)와 병용(併用)함으로써, CMOS 손흔들림에 기인하는 포컬 플레인 현상을 억제하는 것이, 비교적 용이하다.

[0089] 한편, 비디오 카메라와 같은 동화상(動畫; moving picture) 촬영을 전제로 한 촬상 장치에서는, 업무용이나 고급 기종에서는, 1프레임에서의 최대 샘플링 시간차(맨(一番) 위의 라인과 맨 아래의 라인의 샘플링 시각의 차(差))를 짧게 하기 위해서, 초고속의 판독출력을 행함으로써, 실질적으로 CMOS 손흔들림에 따르는 포컬 플레인의 발생을 방지하는 방법이 채용되고 있는 일이 있다.

[0090] 또, 촬상 화상에 대한 상대적인 손흔들림량의 크기는, 광학(光學) 줌의 배율이 커짐에 따라서 증대하기 때문에, 동화상 촬영 용도이더라도, 광학 줌을 가지지 않거나, 배율이 작은 기종이면, CMOS 손흔들림은 큰 문제로 되고 있지 않다. 애초부터, 종래의 CCD 이미저에서 행하고 있는 바와 같은, 가속도 센서를 이용한 손흔들림 보정조차 탑재하고 있지 않은 저렴(安價)한 촬상 장치에서는, CMOS 손흔들림의 악영향은 상대적으로 작게 되고, 문제로 조차 되고 있지 않은 경우가 많다.

[0091] 상기의 특허 문헌은, 다음과 같다.

[특허 문헌 1] 미국 특허 제3, 384, 459호 공보

발명이 이루고자 하는 기술적 과제

[0093] 상술한 상황을 감안하면, 정지화상 촬영은 할 수 있지만 동화상 촬영을 주용도(主用途)로 하고, 고배율(高倍率)의 광학 줌을 탑재한 촬상 장치에서는, 메카 셔터와 병용하는 등의 특수한 장치나, 고속 샘플링 클럭을 이용하는 것이 필요하게 된다. 그러나, 그와 같은 구성으로 한 경우에는, 매우 고정밀도이고 고(高)코스트로 되는 문제가 있다.

[0094] 또, 종래, 손흔들림을 검출하는 방법으로서는, 자이로(각속도(角速度)) 센서 등의 기계적인 부품을 이용하는 방법이 일반적이지만, 자이로 센서를 이용한 경우에는, 촬상 장치의 소형, 경량화, 저코스트화를 실현하는 경우의 지장(支障)으로 된다고 하는 문제가 있다.

[0095] 또, 종래는, 자이로 센서를 이용한 손흔들림 보정의 약점인, 자이로 센서 자체(自體)의 검출 정밀도(精度)의 낮음(甘; low precision)이, 그때까지의 주요한 응용처(應用先)인 동화상 촬영에서는, 문제로 되지 않았지만, 요 몇년(數年)에 있어서는, 디지털 스틸 카메라의 급속한 보급과, 그것과 기(機)를 같이 한 급속한 고화소화(高畫素化)의 흐름이, 새로운 문제를 낳기 시작하고 있다. 그것은, 저조도(低照度)(노광 시간이 길다) 시의 정지화상에서도, 손흔들림 보정이 강하게 요구되고 있지만, 해답(解; solution)이 자이로 센서 등의 센서를 이용한 것밖에 존재하지 않고, 앞서(先) 든 자이로 센서의 약점이나 그 밖의 문제가 드러나고 있는 점이다.

[0096] 현재 시장에 나돌고 있는 민생기(民生機)에서의, 정지화상 용도의 손흔들림 보정은, 모두 보편적으로(遍; 널리)

자이로 센서 혹은 가속도 센서를 사용해서 손흔들림 벡터를 계측하고, 그것을 기구계(機構系)에 피드백해서, CCD(Charge Coupled Device) 이미저나 CMOS(Complementary Metal Oxide Semiconductor) 이미저 등의 이미지 센서에 사영(寫影; project; 투영, 투사)되는 상(像)이, 흔들림을 일으키지 않도록 고속으로 제어한다고 하는 것이다.

- [0097] 여기에서 말하는 기구계로서는, 렌즈, 프리즘, 이미저(혹은, 이미저와 일체화한 모듈)이 제안되어 있으며, 각각 렌즈 시프트, 프리즘 시프트, 이미저 시프트라고 불리고 있다.
- [0098] 이와 같은 방법으로 손흔들림 보정이 이루어지고 있는 이상, 앞서 든 자이로 센서 자체의 정밀도 오차(誤差)에 더하여, 기구계에의 피드백 지연, 혹은 피드백 지연을 회피하기 위한 예측 오차, 그리고 기구계의 제어 오차도 중첩(重疊)되고, 대단히(몹시) 픽셀 정밀도로 보정을 가하는 것은 불가능하다.
- [0099] 이상에서 든 바와 같이, 현상태(現狀)의 센서를 사용한 손흔들림 보정에는, 원리적으로, 정밀도를 추구할 수 없다고 하는 큰 문제가 있음에도 불구하고, 시장에서 높은 평가를 얻고 있는 것은, 손흔들림을 보정할 수는 없어도(없을지언정) 저감할 수 있기 때문이다.
- [0100] 그렇지만, 앞으로(今後) 점점(益益) 고품질화가 예상되는 가운데, 픽셀 사이즈가 작아짐에 따라서, 보정 한계가 픽셀 정밀도로 점점 되어(開; become) 가지 아니할 수 없다고 하는 사실에 시장이 깨닫는(氣付; 알아차리는) 것도 시간 문제이다.
- [0101] 본 발명은, 이상의 점을 감안하고, 상술한 바와 같은 문제를 회피하면서, CMOS 이미저 등의 X-Y 어드레스형 활상 소자를 이용한 활상 장치에서도, 손흔들림 등이 원인으로 생기는 포컬 플레인 현상에 의한 활상 화상의 일그러짐을 경감할 수 있는 방법 및 장치를, 자이로 센서 등의 기구계 부품을 사용하는 일없이, 디지털 신호 처리에 의해 고정밀도로 또한 저코스트로 실현하는 것을 목적으로 한다.

발명의 구성 및 작용

- [0102] 상기의 과제를 해결하기 위해서, 청구항 1의 발명은,
- [0103] 1화면의 화상의 수평 방향 및/또는 수직 방향의 일그러짐을 보정하는 방법으로서,
- [0104] 상기 화상의 1화면 구간을 복수개의 분할 화상 구간으로 분할하고,
- [0105] 상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를, 2화면 사이의 상기 분할 화상 구간의 화상 정보로부터 검출하고,
- [0106] 검출된 상기 분할 화상 구간의 상기 움직임 벡터의 인접(隣接; adjacent)하는 분할 화상 구간의 것의 차분을 구하는 것에 의해, 상기 분할 화상 구간의 각각마다, 해당(當該) 분할 화상 구간의 화상의 일그러짐 변화의 속도를 검출하고,
- [0107] 상기 복수개의 분할 화상 구간의 각각 내에서는, 각 분할 화상 구간에 대해서 검출된 상기 화상의 일그러짐 변화의 속도에 의거해서, 상기 화상의 일그러짐을 보정하는
- [0108] 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법을 제공한다.
- [0109] 상술한 구성의 청구항 1의 발명에 따르면, 1화면 구간분(區間分)이 복수로 분할된 각각의 분할 화상 구간에서, 자이로 센서 등의 기구 부품을 이용하는 일없이, 해당 분할 화상 구간의 화상 정보가 이용되어, 해당 분할 화상 구간에서의 화상의 움직임 벡터가 검출된다.
- [0110] 그리고, 검출된 움직임 벡터의, 인접하는 분할 화상 구간의 것의 차분을 구하는 것에 의해, 분할 화상 구간의 각각마다의 화상의 일그러짐 변화의 속도가 검출된다. 그리고, 해당 분할 화상 구간에 대해서 검출된 화상 일그러짐 변화의 속도에 의거해서, 화상의 일그러짐이 보정되고, 예를 들면 포컬 플레인 현상을 포함하는 화상의 일그러짐을 보정할 수가 있다.
- [0111] 또, 청구항 2의 발명은, 청구항 1에 기재된 화상 신호의 일그러짐 보정 방법에 있어서,
- [0112] 상기 각 분할 화상 구간 내에서는, 해당 각 분할 화상 구간에 대해서 검출된 상기 화상의 일그러짐 변화의 속도의 시간 적분값(積分值)을 상기 화상의 일그러짐 보정 대상 부위(補正對象部位)의 일그러짐 변위량(變位量; displacement quantity)으로 하고, 상기 시간 적분값을 이용해서 상기 화상의 일그러짐을 보정하는

- [0113] 것을 특징으로 하는 화상 신호의 일그러짐 보정 방법을 제공한다.
- [0114] 상술한 구성의 청구항 2의 발명에 따르면, 분할 화상 구간의 각각 내에서는, 해당 분할 화상 구간에 대해서 검출된 화상 일그러짐 변화의 속도의 시간 적분값을 화상의 일그러짐 보정 대상 부위의 변위량으로서 구하고, 해당 구한 시간 적분값을 이용해서 상기 화상의 일그러짐을 보정하므로, 적은 처리 지연으로, 예를 들면 포컬 플레인 현상을 포함하는 화상의 일그러짐을 보정할 수 있음과 동시에, 시간 적분값을 이용하므로, 화상 일그러짐 변화의 속도가 바뀌는 다른(他) 분할 화상 구간의 화상과의 경계(境界)에서도, 화상에 어긋남을 일으키는 일은 없다.
- [0115] 또, 청구항 14의 발명은,
- [0116] 주목 화면(注目畫面; observed screen)의 화상의 1화면 구간을 복수개의 분할 화상 구간으로 분할하고,
- [0117] 상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를, 상기 주목 화면인 참조 화면과 해당 참조 화면보다도 앞(前)의 원화면(元畫面; original screen)과의 사이의 상기 분할 화상 구간의 화상 정보로부터 검출하고,
- [0118] 검출된 상기 분할 화상 구간의 상기 움직임 벡터에 의거해서, 상기 참조 화면의 수평 방향 및/또는 수직 방향의 화상의 일그러짐을 보정하는 방법에 있어서,
- [0119] 상기 분할 화상 구간의 각각에서의 화상의 움직임 벡터를 검출하는 방법은,
- [0120] 상기 원화면 중의 상기 각 분할 화상 구간에서, 소정의 위치에 복수의 화소로 이루어지는 소정의 크기의 적어도 1개의 타겟 블록을 설정하고,
- [0121] 상기 타겟 블록과 같은 크기의 참조 블록을, 상기 참조 화면에 설정된 서치 범위에서 복수개 설정하고,
- [0122] 상기 복수개의 참조 블록 내로부터, 상기 타겟 블록과 상관(相關)이 강한 상기 참조 블록을 검출하고,
- [0123] 해당 검출한 참조 블록의 상기 타겟 블록에 대한 화면 위의 위치 어긋남량(shift over)에 의거해서, 상기 움직임 벡터를 검출하는
- [0124] 것으로서,
- [0125] 상기 참조 블록의 각각에서, 해당 참조 블록 내의 각 화소의 화소값과, 상기 타겟 블록 내에서 대응하는 위치의 각 화소의 화소값과의 차분의 절대값(絕對值)의 총합(總和; total of sum)을 구하는 차분 절대값합 산출(算出) 공정과,
- [0126] 상기 참조 블록의 각각의 상기 참조 화면 위의 위치의, 상기 타겟 블록의 화면 위의 위치와의 위치 어긋남량을, 방향 성분도 포함하는 참조 벡터로 하고, 해당 참조 벡터를 소정의 축소율로 축소한 참조 축소 벡터를 얻는 참조 축소 벡터 취득 공정과,
- [0127] 상기 참조 축소 벡터에 따른 크기의 상기 참조 벡터를 상기 위치 어긋남량으로 하는, 상기 소정의 축소율에 따라서 삼각형 수의 복수개의 상기 참조 블록의 각각에 대한 차분 절대값합을 기억하는 축소 차분 절대값합 테이블을 생성하는 테이블 생성 공정과,
- [0128] 상기 축소 차분 절대값합 테이블에서의 상기 차분 절대값합의 최소값(最小值)에 대응하는 상기 참조 벡터를 적어도 이용해서, 상기 참조 화면과 상기 원화면과의 사이의 상기 분할 화상 구간의 각각에 대한 움직임 벡터를 산출하는 움직임 벡터 산출 공정
- [0129] 을 구비하고,
- [0130] 상기 테이블 생성 공정은,
- [0131] 상기 참조 축소 벡터 취득 공정에서 취득된 상기 참조 축소 벡터의 근방값(近傍值; neighborhood value)으로 되는 복수의 상기 참조 벡터를 검출하는 근방 참조 벡터 검출 공정과,
- [0132] 상기 차분 절대값합 산출 공정에서 산출된 상기 참조 블록의 각각에 대한 상기 차분 절대값의 총합으로부터, 상기 근방 참조 벡터 검출 공정에서 검출된 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 차분 절대값합의 각각을 산출하는 분산(分散) 차분 절대값합 산출 공정과,
- [0133] 상기 분산 차분 절대값합 산출 공정에서 산출된 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합을, 그때까지의 상기 근방의 복수의 상기 참조 벡터의 각각에 대응하는 상기 차분 절대값합에 가산

(加算; add)하는 분산 가산 공정

[0134] 을 구비하는 것을 특징으로 한다.

[0135] 상술한 구성의 청구항 14의 발명에 따른 화상 신호의 일그러짐 보정 방법에서의 움직임 벡터의 검출 방법에 있어서는, 주목 화면의 서치 범위 내의 참조 블록의 각각에서, 해당 참조 블록 내의 각 화소의 화소값과, 상기 타겟 블록 내에서 대응하는 위치의 각 화소의 화소값과의 차분의 절대값의 총합(이 차분의 절대값의 총합을 차분 절대값합이라고 부른다. 이하, 이 차분 절대값합을 SAD(Sum of Absolute Difference)라고 기재하기로 한다)을 구한다.

[0136] 이 청구항 14의 발명에서는, 구한 참조 블록의 SAD의 값(SAD값)을, 그 참조 블록의 참조 벡터에 대응해서 기억하는 것이 아니라, 그 참조 벡터를 축소한 참조 축소 벡터에 대응해서 기억한다.

[0137] 이 경우에, 참조 축소 벡터는, 참조 블록의 각각이 대응하는 참조 벡터와는 일치하고 있지 않으므로, 테이블 생성 공정에서는, 우선, 참조 축소 벡터 근방의 복수개의 참조 벡터를 검출한다. 그리고, 그들 근방의 복수의 참조 벡터의 각각 대응하는 SAD값을, SAD 산출 공정에서 산출한 SAD값으로부터 산출한다.

[0138] 그리고, 산출한 근방의 복수의 참조 벡터의 각각에 대응하는 SAD값을, 그때까지의 해당 근방의 복수의 참조 벡터의 각각에 대응하는 SAD값에 가산한다.

[0139] 이렇게 해서 얻어진 SAD 테이블은, 참조 축소 벡터 각각의 근방의 복수개의 참조벡터만에 대한 SAD값의 테이블로 되어 있고, 참조 벡터의 축소율에 따라서 축소된 것으로 되어 있다.

[0140] 다시말해, 바꾸어말하면, 생성된 SAD 테이블은, 참조 벡터의 축소율에 따라서 축소된 프레임 화상에 대한 SAD 테이블에 대응한다. 이 경우에 있어서, 타겟 블록 및 참조 블록의 크기는 축소되어 있지 않으므로, 생성된 SAD 테이블 사이즈는, 작아진다.

[0141] 이 축소된 SAD 테이블에서, SAD값이 최소값으로 되어 있는 참조 벡터를 검출하고, 그 검출한 참조 벡터로부터 구하는 움직임 벡터를 산출한다. 예를 들면, 구한 SAD값이 최소값으로 되어 있는 참조 벡터를, 참조 벡터의 축소율에 따라서 확대해서, 움직임 벡터를 산출할 수가 있다.

[0142] 그리고, 이 청구항 14의 발명에 있어서는, 화상을 축소하는 처리는 불필요하므로, 화상의 축소 변환에 따르는 처리 시간의 증대나 메모리 용량(容量)의 소비는 없고, 또 SAD값은, 원래(元) 프레임의 모든 화소를 이용해서 구하고, 해당 구한 SAD값을 참조 축소 벡터 근방의 복수개의 참조 벡터에 대응해서 분산시키면서 구하는 것이므로, SAD값을 분산시키는 시점에서 스스로 축소 배율에 따른 적절한 필터 연산을 하고 있는 것과 등가(等價)이며, 화상을 축소한 경우와 같은 로우패스 필터의 실장(實裝)은, 불필요해지는 것이다.

[발명을 실시하기 위한 조건의 형태]

[0144] 이하, 본 발명에 따른 화상 신호의 일그러짐 보정 장치 및 방법의 실시형태를, CMOS 이미저를 활용 소자에 이용한 활용 장치에 적용한 경우를 예로, 도면을 참조하면서 설명한다.

[0145] [실시형태에서의 CMOS 손흔들림에 기인하는 포컬 플레이인 현상의 일그러짐 보정에 대해서]

[0146] CMOS 손흔들림에 기인하는 포컬 플레이인 현상의 일그러짐 보정 및 억제 기술로서는, 앞서 특히 문헌 2(일본 특개(特開) 제2004-266322호 공보)에 기재된 것이 있다.

[0147] 이 특히 문헌 2에 기재되는 보정 방법은, 1수평 라인 단위로 활용 화상 데이터가 CMOS 이미저로부터 판독출력될 때, 각 수평 라인에 대한 손흔들림에 의한 위치 변위량(이하, 손흔들림량이라고 한다)을 검출하고, 해당 검출한 손흔들림량 분만큼, 발생한 손흔들림과는 역방향으로 어긋난 위치로부터 해당 수평 라인의 데이터를 판독출력하도록 보정하는 것이다.

[0148] 그러나, 손흔들림량을 각 수평 라인에 대해서 얻는 것은, 손흔들림을 검출하는 센서의 샘플링 주파수 등의 조건으로 인해 곤란하다는 것을 감안하고, 이 특히 문헌 2에서는, 도 61의 (A)에 도시하는 바와 같이, 화면의 수직 방향의 복수 라인분 걸러(置; at intervals)의 위치에서의 손흔들림량을, 이산적(離散的; discretely)으로 검출하도록 하고 있다. 도 61에서는, 50 라인 걸러의 라인 위치에서의 손흔들림량 Q1, Q2, Q3…(도 61의 (B) 참조; 또한, 도면에서는 수평 방향의 손흔들림량만을 도시하고 있다)을 검출하고 있다.

[0149] 그리고, 손흔들림 검출을 행하는 라인 이외의, 손흔들림량을 직접적으로는 검출하지 않는 49 라인에 대한 손흔들림량은, 검출된 손흔들림량 Q1, Q2, Q3…을 이용해서, 보간에 의해 구하도록 한다. 이 보간 방법으로서는,

예를 들면 도 61의 (C)에 예를 나타내는 바와 같이, 몇 가지(幾通; some)의 방법이 있다. 도 61의 (C)의 보간 방법은, 기본적으로는 손흔들림량을 직접적으로는 검출하지 않는 49라인의 직전(直前) 라인 및 직후(直後) 라인의 2위치에서의 2개의 손흔들림량 Q_n 및 Q_{n+1} (n 은 1이상의 정수(整數))을 이용해서 행한다.

[0150] 예를 들면, 보간(1)의 방법에서는, 직전 라인에서 검출된 손흔들림량 Q_n 을, 상기 49라인 중의 전반(前半) 라인에서는 그대로 이용하고, 상기 49라인 중의 후반(後半) 라인에서는, 직후 라인에서 검출된 손흔들림량 Q_{n+1} 을 이용하도록 한다. 또, 보간(2)의 방법은, 직전 라인에서 검출된 손흔들림량 Q_n 과 직후 라인에서 검출된 손흔들림량 Q_{n+1} 을 직선으로 연결했을 때의 각 라인에서의 값을, 해당 라인의 손흔들림량으로 하는, 즉 이른바 평균값(平均值) 보간을 행하는 방법이다.

[0151] 이 특허 문헌 2의 방법에 따르면, 손흔들림이 원인으로 생기는 CMOS 이미저에서의 포컬 플레인 현상을 포함하는 일그러짐을 보정하는 것이 가능하다.

[0152] 그렇지만, 이 특허 문헌 2에서 채용되고 있는 방법의 경우에는, 이산적인 샘플링 위치에서 손흔들림을 검출하고, 그 검출 결과로부터 상기 이산적인 샘플링 위치에서의 손흔들림량을 구하고, 해당 이산적인 샘플링 위치에서의 손흔들림량을 이용해서 보간을 하는 것에 의해, 각 라인 위치에서의 손흔들림량을 추정하도록 하고 있다.

[0153] 이 때문에, 특허 문헌 2의 경우에는, 상기 직전 라인의 손흔들림량 Q_n 과, 상기 직후 라인의 손흔들림량 Q_{n+1} 과의 양쪽(兩方; both)이 얹어진 후가 아니면, 상기 직전 라인 뒤(後)의 각 라인의 손흔들림량을, 보간 처리에 의해 얻도록 할 수가 없다. 이 때문에, 상기 직전 라인 뒤의 각 라인에 대한 손흔들림 보정 처리가, 최대로, 이산적인 샘플링 주기(周期) 구간인 복수 라인 구간 분만큼 지연되게 된다.

[0154] 또, 상술한 보간(1)의 방법을 이용한 경우에는, 보간값(補間值)으로서의 손흔들림량의 변화 시간에서, 화상 위치가 어긋날 가능성이 있다.

[0155] 또, 보간(2)의 방법의 경우에는, 직전 라인의 손흔들림량 Q_n 과 직후 라인의 손흔들림량 Q_{n+1} 로부터, 손흔들림량의 변화의 기울기(傾; gradient)를 구하고, 그 기울기에, 직전 라인으로부터의 대상 라인까지의 라인수를 승산(乘算)하는 것에 의해, 해당 대상 라인의 손흔들림량을 구하도록 하지 않으면 안된다. 이 때문에, 승산기가 필요하게 됨과 동시에, 승산 파라미터를 별도(別途) 설정하기 위한 레지스터를 설치(設; provide)할 필요가 있고, 하드웨어가 복잡화됨과 동시에, 회로 규모가 커진다. 또, 승산 오차로 인해, 손흔들림량의 샘플링값(值)이 바뀌는 경계 위치에서, 도 61의 (C)의 보간(2)의 경우의 도면과 같이 연속적으로는 되지 않고, 화상 위치가 어긋날 가능성이 있다.

[0156] 이하에 설명하는 실시형태에서는, 상술한 바와 같은 문제를 회피하면서, CMOS 이미저 등의 X-Y 어드레스형 활상 소자를 이용한 활상 장치에서도, 손흔들림 등이 원인으로 생기는 포컬 플레인 현상에 의한 활상 화상의 일그러짐을 경감할 수 있도록 하고 있다.

[0157] 우선, 도 1을 이용해서, 본 발명에 따른 화상 신호의 일그러짐 보정 방법의 실시형태에서의 포컬 플레인 현상에 의한 활상 화상의 일그러짐을 경감하는 방법의 개요를 설명한다. 이하에 설명하는 실시형태는, 활상 소자로서 X-Y 어드레스형 고체 활상 소자의 대표예인 CMOS 이미저를 이용해서 활상한 활상 화상에 대해서, 전술한 CMOS 손흔들림을 보정하는 경우의 예이다. 또한, 본 발명은, 활상 화상이 동화상인 경우와 정지화상인 경우의, 어느 경우에도 적용할 수 있는 것이다.

[0158] 도 1의 개요 설명도에서는, 도 1의 (A)에 도시하는 바와 같이, 화상에 생기는 CMOS 손흔들림에 의한 일그러짐을 알기 쉽게 나타내기 위해서, 일그러짐이 없는 오리지널 화상으로서 다수개(多數個)의 장방형(長方形; rectangles)으로 이루어지는 격자 무늬 모양(格子縞模樣; lattice pattern)의 화상을 상정(想定)하고 있다. 따라서, CMOS 손흔들림에 의한 화상 일그러짐은, 해당 격자 무늬 모양을 구성하는 장방형의 변형 일그러짐으로서 나타나는 것으로 된다.

[0159] 또한, 이 실시형태에서도, CMOS 이미저는, 도 58에 도시한 바와 같은 1화면으로 하는 수평 유효 영역 및 수직 유효 영역으로 정해지는 유효 화상 영역 EFL(1수직 주기분)보다도 넓은 범위의 실효 화상 영역 AFL을 가지는 것이 이용되는 것이다. 또한, 도 1의 (A)~(D)에 도시한 1화면분의 화상은, 도 58의 유효 화상 영역 EFL의 크기의 화상이다.

[0160] 이 실시형태에서는, 수평 방향의 화소 클럭의 속도는, 손흔들림의 속도보다도 충분히 빠르게 하고, 또 활상 화상 데이터의 판독출력은, 1수평 라인 단위로 순차(順次)로 행하는 것으로서, 손흔들림 보정은, 1수평 라인 단위

로 행하도록 한다.

[0161] 또, 이 실시형태에서는, 도 1의 (B), (C), (D)에 도시하는 바와 같이, CMOS 이미저의 유효 화상 영역 EFL 분의 화상 구간은, 수직 방향으로 복수로 등분할(等分割)하고, 각 분할 화상 구간 Pdiv에서는, 같은 손흔들림 속도의 손흔들림 변위를 받는 것으로 가정(假定)하고 있다. 여기에서, 분할수는, 손흔들림 보정했을 때에, 필요 충분한 보정 효과가 얻어지는 분할 화상 구간 Pdiv의 크기를 고려해서 정해진다. 이하에 설명하는 예에서는, 유효 화상 영역 EFL의 화상 구간을 수직 방향으로 8분할하도록 한다. 즉, 이 예에서는, 분할 화상 구간 Pdiv, 수직 동기(同期) 신호의 1주기를 1/8로 분할한 구간으로 된다.

[0162] 하나의 분할 화상 구간 Pdiv는, 복수 수평 라인분으로 이루어지지만, 이 실시형태에서는, 같은 분할 화상 구간 Pdiv에 포함되는 복수 수평 라인은 같은 손흔들림 속도의 손흔들림 변위를 받는 것으로 가정해서 손흔들림 보정 처리를 행하는 것이다.

[0163] 그리고, 이 실시형태에서는, 8개의 분할 화상 구간 Pdiv_0~Pdiv_7의 각각의 선두의 수평 라인에 대한 손흔들림 속도 벡터를, 도 1의 (B)의 좌측의 화살표로 나타내는 바와 같은 것으로 해서, 손흔들림 속도 검출 수단에 의해 검출한다. 즉, 이 예에서는, 손흔들림 속도 벡터는, 1화면분의 활상 화상에 대해서, 8개소(箇所)에서, 이산적 으로 검출하도록 한다.

[0164] 도 1의 (B)의 화상은, 손흔들림 속도 벡터가, 각 분할 화상 구간 Pdiv에서, 좌측의 화살표와 같은 것으로 했을 때에 생기는 손흔들림에 의한 활상 화상의 일그러짐을 나타내고 있다. 이 실시형태에서는, 이 손흔들림에 의한 활상 화상의 일그러짐을, 수평 방향 처리와 수직 방향 처리로 나누어, 각각의 방향의 일그러짐을 따로따로(別別) 보정하도록 한다.

[0165] 보정 처리의 상세(詳細)에 대해서는 후술하겠지만, 이 실시형태에서는 우선, 수평 방향의 일그러짐에 대해서 보정 처리를 행하고, 그 다음에, 수직 방향의 일그러짐에 대한 보정 처리를 행한다. 이 경우에, 1화면분의 활상 화상의 모든 데이터에 대한 수평 방향의 일그러짐 보정 처리가 완료하기 전이더라도, 수직 방향의 일그러짐 보정 처리가 가능하게 된 단계(段階)에서, 수직 방향의 일그러짐 보정 처리를 개시(開始; start)해서, 수평 방향의 일그러짐 보정 처리와 병행(並行)해서 행하도록 하는 것에 의해, 효율좋게 일그러짐 보정 처리를 행하도록 한다.

[0166] 그래서, 이 실시형태에서는, 우선, 손흔들림에 의한 수평 방향의 화상 일그러짐을 보정하기 위해서, 도 1의 (C)의 좌측의 화살표 및 도 2의 좌측의 화살표로 나타내는 바와 같이, 각 분할 화상 구간 Pdiv_0~Pdiv_7에 대해서 검출한 손흔들림 속도 벡터 Vec(Vec_0~Vec_7)의 수평 방향 성분의 역부호(逆符號) 성분(이하, 이 손흔들림 속도 벡터의 수평 방향 성분의 역부호 성분을, 수평 보정 속도 성분이라고 한다) X_STB(X_STB_0~X_STB_7)를 구한다.

[0167] 그리고, 각 분할 화상 구간 Pdiv_0~Pdiv_7 내에서는, 각 수평 라인의 각각에 대한 수평 방향의 손흔들림량을 보정하기 위한 보정량으로서, 상술한 바와 같이 해서 구한 수평 보정 속도 성분 X_STB_0~X_STB_7을, 각각의 수평 라인에 대해서 시간 적분(각 분할 화상 구간 Pdiv_0~Pdiv_7의 선두의 수평 라인 시점을 기준점으로 한 시간으로 적분)해서 산출한다. 도 2의 우측에, 그 적분 결과의 값으로서의 수평 방향의 손흔들림 보정량(이하, 수평 손흔들림 보정량이라고 한다) SX_ADD를 도시한다.

[0168] 이 도면으로부터 알 수 있는 바와 같이, 각 수평 라인의 각각에 대한 수평 손흔들림 보정량 SX_ADD는, 각 분할 화상 구간 Pdiv_0~Pdiv_7에서는, 일정(一定; fixed)한 기울기로 변화하고, 또한, 적분이므로, 분할 화상 구간의 경계선(境目; boundary point)에서는 꺾은선(折線; broken curve)으로는 되지만, 연속적인 것으로 되고, 어긋남은 생기지 않는다.

[0169] 수평 방향의 일그러짐 보정 처리에서는, 이상과 같이 해서 구해진 각 라인에 대한 손흔들림 수평 보정량 SX_ADD를 이용해서, CMOS 이미저 위에서의 각 수평 라인의 수평 방향의 판독출력 개시 위치를 보정하는 것에 의해, 활상 화상을 보정 처리한다. 즉, 각 수평 라인의 판독출력 개시 위치를, 도 1의 (B)에서의 손흔들림에 따라서 수평 방향으로 어긋난 위치로부터로 하는 것에 의해, 도 1의 (C)의 화상에 나타내는 바와 같이, 수평 방향의 일그러짐이 보정되게 된다.

[0170] 이 도 1의 (C)의 화상은, 도 1의 (B)의 화상 일그러짐 내의 수평 방향의 일그러짐은 보정되어 제거되어 있지만, 수직 방향의 일그러짐이 남아 있다.

[0171] 그래서, 이 실시형태에서는, 손흔들림에 의한 수직 방향의 화상 일그러짐을 보정하기 위해서, 도 1의 (D)의 좌

측의 화살표로 나타내는 바와 같이, 각 분할 화상 구간 Pdiv_0~Pdiv_7에 대해서 검출한 손흔들림 속도 벡터의 수직 방향 성분의 역부호 성분(이하, 이 손흔들림 속도 벡터의 수직 방향 성분의 역부호(逆符號) 성분을, 수직 보정 속도 성분이라고 한다) Y_STB_0~Y_STB_7을 구한다.

[0172] 그리고, 각 분할 화상 구간 Pdiv_0~Pdiv_7 내에서는, 각 수평 라인의 각각에 대한 수직 방향의 손흔들림량을 보정하기 위한 보정량으로서, 상술한 바와 같이 해서 구한 수직 보정 속도 성분 Y_STB_0~Y_STB_7을, 각각의 수평 라인에 대해서 시간 적분(각 분할 화상 구간 Pdiv_0~Pdiv_7의 선두의 수평 라인 시점을 기준점으로 한 시간으로 적분)해서 산출한다.

[0173] 이 적분 결과로서의 수직 방향의 손흔들림 보정량(이하, 수직 손흔들림 보정량이라고 한다) SY_ADD는, 여기에서는, 도시를 생략하지만, 각 분할 화상 구간 Pdiv_0~Pdiv_7에서는, 일정한 기울기로 변화하고, 또한 적분이므로, 분할 화상 구간의 경계선에서는 꺾은선으로는 되지만, 연속적인 것으로 되고, 어긋남은 생기지 않는다.

[0174] 수직 방향의 일그러짐 보정 처리에서는, 이상과 같이 해서 구해진 각 라인에 대한 수직 손흔들림 보정량 SY_ADD를 이용해서, CMOS 이미저 위에서의 각 수평 라인의 수직 방향의 판독출력 개시 위치를 보정하는 것에 의해, 활상 화상을 보정 처리한다. 이것에 의해, 도 1의 (D)의 화상에 나타내는 바와 같이, 수평 방향의 경우와 마찬가지로 해서 수직 방향의 일그러짐이 보정되고, 일그러짐이 제거된 활상 화상 데이터 출력이 얻어진다.

[0175] 수평 손흔들림 보정량 및 수직 손흔들림 보정량을 이용해서, 손흔들림에 의한 CMOS 이미저에 대한 포컬 플레인 현상에 의한 화상 일그러짐이 보정되는 모습(樣子; state)은, 전술한 특허 문헌 2와 마찬가지이다.

[0176] 그러나, 특허 문헌 2에서는, 손흔들림량을, 수직 방향의 이산적인 수평 라인에 대해서 검출함과 동시에, 직접적으로 손흔들림량을 검출하지 않는, 이산적인 수평 라인 사이의 각 수평 라인에 대해서는, 상기 이산적으로 검출한 손흔들림량을 이용한 보간 처리에 의해 얻도록 하고 있는데 대해서, 이 실시형태에서는, 수직 방향의 이산적인 수평 라인에 대해서는 손흔들림 속도를 검출하고, 해당 검출한 손흔들림 속도를, 각 수평 라인에 대해서 순차로 적분해 가는 것에 의해, 각 수평 라인의 손흔들림량(손흔들림 보정량)을 검출하는 점이 크게 다르다.

[0177] 이와 같이, 이 실시형태에서는, 각 수평 라인의 손흔들림량(손흔들림 보정량)은, 손흔들림 속도를 적분하는 것에 의해 산출하는 것이므로, 분할 화상 구간의 경계선에서는 꺾은선으로는 되지만, 연속적인 것으로 되고, 어긋남은 생기지 않는다.

[0178] 또한, 이 실시형태에서는, 손흔들림 속도 벡터로부터 수평 성분 및 수직 성분을 추출할 때에, 그 역부호 성분을 산출해서, 보정 속도 성분으로 하도록 했지만, 속도 성분 혹은 손흔들림량을 역부호로 해서, 보정 속도 성분 혹은 손흔들림 보정량으로 하는 것은, 활상 화상 데이터에 대한 손흔들림 보정 처리의 실행 개시전이면, 어느 시점이더라도 좋다.

[0179] [손흔들림 움직임 벡터 및 손흔들림 속도 벡터의 산출]

[0180] 이 실시형태에서는, 활상 화상의 손흔들림 움직임 벡터를 검출하는 방법으로서, 2화면 사이의 상관을 구하는 블록 매칭을 이용한다. 이 블록 매칭을 이용하는 방법은, 자이로(각속도) 센서 등의 기계적인 부품이 불필요하므로, 활상 장치의 소형, 경량화를 실현할 수 있다고 하는 점에서 유리하다.

[0181] 도 3 및 도 4는, 블록 매칭의 개요를 도시한 것이다. 또, 도 5에는, 그의 처리 플로차트의 일반예를 도시한다.

[0182] 블록 매칭은, 활상 장치부로부터의 활상 화상에 대해, 주목 화면인 참조 화면과, 해당 참조 화면보다도 1화면분 앞의 활상 화면인 원화면과의 사이의 1화면분 단위로의 움직임 벡터를, 소정 크기의 직사각형(矩形; rectangular) 영역의 블록에 대해서, 참조 화면과 원화면과의 상관을 산출하는 것에 의해 산출하는 방법이다.

[0183] 또한, 여기에서 화면이라 함은, 1프레임 또는 1필드의 화상 데이터로 이루어지는 화상을 의미하고 있지만, 이 명세서에서는, 설명의 편의상, 화면은 1프레임으로 이루어지는 것으로 해서, 화면을 프레임이라고 칭(稱)하기로 한다. 따라서, 참조 화면은 참조 프레임, 원화면은 원(元)프레임이라고 칭한다.

[0184] 예를 들면, 참조 프레임의 화상 데이터는, 활상 장치부로부터의 현(現)프레임의 화상 데이터, 또는 현프레임의 화상 데이터가 프레임 메모리에 격납(格納; store)되어 1프레임분 지연된 것으로 된다. 원프레임의 화상 데이터는, 참조 프레임의 화상 데이터가 또 프레임 메모리에 격납되어 1프레임분 지연된 것으로 된다.

- [0185] 도 3 및 도 4는, 종래의 블록 매칭의 개요를 설명하기 위한 도면이다. 또, 도 5는, 종래의 블록 매칭 처리의 플로차트의 예이다.
- [0186] 블록 매칭에서는, 도 3에 도시하는 바와 같이, 원프레임(101)의 임의(任意)의 소정의 위치에서, 수평 방향의 복수 화소 및 수직 방향의 복수 라인으로 이루어지는 소정 크기의 직사각형 영역으로 이루어지는 타겟 블록(103)이 설정된다.
- [0187] 이것에 대해서, 참조 프레임(102)에서, 원프레임의 타겟 블록(103)의 위치와 같은 위치에, 타겟 블록의 사영 이미지 블록(104)(도 3의 점선 참조)을 상정하고, 이 타겟 블록의 사영 이미지 블록(104)을 중심으로 한 서치 범위(105)(도 3의 1점 쇄선 참조)를 설정함과 동시에, 타겟 블록(103)과 같은 크기의 참조 블록(106)을 생각한다.
- [0188] 그리고, 이 참조 블록(106)의 위치를 참조 프레임(102)의 서치 범위(105) 내에서 이동시키고, 각 위치에서 참조 블록(106)에 포함되는 화상 내용과, 타겟 블록(103)의 화상 내용과의 상관을 구하고, 가장(最) 상관이 강하다고 해서 검출된 참조 블록(106)의 위치를, 원프레임의 타겟 블록(103)이, 참조 프레임(102)에서 이동한 위치로서 검출하도록 한다. 그리고, 그 검출한 참조 프레임(106)의 위치와, 타겟 블록의 위치와의 사이의 위치 어긋남량을, 방향 성분을 포함하는 양으로서의 움직임 벡터로서 검출하도록 한다.
- [0189] 이 경우, 참조 블록(106)은, 서치 범위(105)를, 예를 들면 수평 방향 및 수직 방향으로, 1화소 또는 복수 화소 단위로 이동시키도록 한다. 따라서, 서치 범위(105) 내에는, 복수개의 참조 블록이 설정되게 된다.
- [0190] 여기에서, 타겟 블록(103)과, 서치 범위(105) 내를 이동하는 각 참조 블록(106)과의 상관은, 타겟 블록(103) 내의 각 화소의 휘도값(輝度值)과, 참조 블록(106) 내의 대응하는 각 화소의 휘도값과의 차분의 절대값의, 블록 내의 전화소에 대한 총합(이 차분의 절대값의 총합을 차분 절대값합이라고 부른다. 이하, 이 차분 절대값합을 SAD(Sum of Absolute Difference)라고 기재하기로 한다)를 구하는 것에 의해 검출한다. 즉, SAD값이 최소로 되는 위치의 참조 블록(106)이 가장 상관이 강한 참조 블록으로서 검출되고, 그 검출된 참조 블록(106)의 타겟 블록(103)의 위치에 대한 위치 어긋남량이 움직임 벡터로서 검출된다.
- [0191] 블록 매칭에서는, 서치 범위(105) 내에 설정되는 복수개의 참조 블록(106)의 각각의, 타겟 블록(103)의 위치에 대한 위치 어긋남량은, 방향 성분을 포함하는 양으로서의 참조 벡터(107)(도 3 참조)로 표현된다. 각 참조 블록(106)의 참조 벡터(107)는, 참조 블록(106)의 참조 프레임(102) 위의 위치에 따른 값으로 되지만, 종래의 블록 매칭에서는, SAD값이 최소값으로 되는 참조 블록(106)의 참조 벡터를, 움직임 벡터로서 검출하는 것이다.
- [0192] 그래서, 블록 매칭에서는, 일반적으로, 도 4에 도시하는 바와 같이, 서치 범위(105) 내에서 설정되는 복수개의 참조 블록(106)의 각각과 타겟 블록(103)과의 사이에서의 SAD값(이하, 설명의 간단을 위해서 참조 블록에 대한 SAD값이라고 한다)을, 각각의 참조 블록(106)의 위치에 따른 참조 벡터(107)의 각각에 대응시켜서, 메모리에 기억해 두고, 그 메모리에 기억된 모든 참조 블록(106)에 대한 SAD값 중에서, 최소의 SAD값의 참조 블록(106)을 검출함으로써, 움직임 벡터(110)를 검출하도록 하고 있다.
- [0193] 서치 범위(105) 내에 설정된 복수개의 참조 블록(106)의 위치에 따른 참조 벡터(107)의 각각에 대응시켜서, 각각의 참조 블록(106)에 대한 SAD값을 기억한 것을, 차분 절대값합 테이블(이하, SAD 테이블이라고 한다)이라고 부른다. 도 4의 SAD 테이블(108)이, 이것을 나타내고 있으며, 이 SAD 테이블(108)에서, 각각의 참조 블록(106)에 대한 SAD값을 SAD 테이블 요소(109)라고 한다.
- [0194] 또한, 상술한 설명에서, 타겟 블록(103) 및 참조 블록(106)의 위치라 함은, 그들 블록의 임의의 특정의 위치, 예를 들면 중심 위치를 의미하는 것이고, 참조 벡터(107)는, 참조 프레임(102)에서의 타겟 블록(103)의 사영 이미지 블록(104)의 위치와, 참조 블록(106)의 위치와의 사이의 어긋남량(방향을 포함한다)을 나타내는 것이다. 도 3 및 도 4의 예에서는, 타겟 블록(103)은, 프레임의 중심 위치에 있다고 하고 있는 경우이다.
- [0195] 그리고, 각 참조 블록(106)에 대응하는 참조 벡터(107)는, 참조 프레임(102) 위의 타겟 블록(103)에 대응하는 위치에 대한 해당 각 참조 블록(106)의 위치 어긋남으로 되고 있으므로, 참조 블록(106)의 위치가 특정되면, 그 위치에 대응해서 참조 벡터의 값도 특정된다. 따라서, SAD 테이블(108)의 메모리에서의 참조 블록의 SAD 테이블 요소의 어드레스가 특정되면, 대응하는 참조 벡터는 특정되게 된다.
- [0196] 이상 설명한 종래의 블록 매칭의 처리를, 도 5의 플로차트를 참조해서 설명하면, 다음과 같이 된다.
- [0197] 먼저, 서치 범위(105) 내의 하나의 참조 블록 I_i 를 지정하지만, 이것은, 해당 참조 블록 I_i 에 대응하는 참조 벡터를 지정하는 것과 똑같다(等)(스텝 S1). 여기에서, 도 5에서, (vx, vy) 는, 타겟 블록의 프레임 위의 위치를

기준 위치 (0, 0)으로 했을 때에, 지정된 참조 벡터에 의해 나타내어지는 위치를 나타내고, vx는 지정된 참조 벡터에 의한, 기준 위치로부터의 수평 방향의 어긋남량 성분이며, 또 vy는 지정된 참조 벡터에 의한, 기준 위치로부터의 수직 방향 성분의 어긋남량 성분이다.

[0198] 여기에서는, 어긋남량 vx, vy는, 화소를 단위로 한 값으로 되고, 예를 들면 vx=+1은, 기준 위치 (0, 0)에 대해서, 수평 방향의 오른쪽(右) 방향으로 1화소분 어긋난 위치를 나타내고, 또 vx=-1은, 기준 위치 (0, 0)에 대해서, 수평 방향의 왼쪽(左) 방향으로 1화소분 어긋난 위치를 나타내고 있다. 또, 예를 들면 vy=+1은, 기준 위치 (0, 0)에 대해서, 수직 방향의 아래(下) 방향으로 1화소분 어긋난 위치를 나타내고, 또 vy=-1은, 기준 위치 (0, 0)에 대해서, 수직 방향의 위(上) 방향으로 1화소분 어긋난 위치를 나타내고 있다.

[0199] 이상과 같이, (vx, vy)는, 참조 벡터로 나타내어지는 기준 위치에 대한 위치(이하, 간단을 위해서, 참조 벡터로 나타내어지는 위치라고 한다)를 나타내고 있고, 참조 벡터의 각각에 대응하고 있다. 다시말해, vx 및 vy를 정수로 했을 때, (vx, vy)는 참조 벡터의 각각을 나타내게 된다. 따라서, 이하의 설명에서는, (vx, vy)의 위치를 나타내는 참조 벡터를, 참조 벡터 (vx, vy)라고 기재하는 일이 있다.

[0200] 여기에서, 서치 범위의 중심 위치를 타겟 블록의 위치, 다시말해 상기 기준 위치 (0, 0)으로 하고, 서치 범위를, 수평 방향으로는 $\pm Rx$, 수직 방향으로는 $\pm Ry$ 로 했을 때,

$$-Rx \leq vx \leq +Rx, -Ry \leq vy \leq +Ry$$

[0202] 로 되는 것이다.

[0203] 다음에, 타겟 블록 Io 내의 하나의 화소의 좌표 (x, y)를 지정한다(스텝 S2). 다음에, 타겟 블록 Io 내의 지정된 하나의 좌표 (x, y)의 화소값 $Io(x, y)$ 와, 참조 블록 Ii 내의 대응하는 화소 위치의 화소값 $Ii(x+vx, y+vy)$ 와의 차분의 절대값 a 를 산출한다(스텝 S3). 즉, 차분 절대값 a 는,

$$a = | Io(x, y) - Ii(x+vx, y+vy) | \quad \dots(\text{식 1})$$

[0205]로서 산출된다.

[0206] 그리고, 산출한 차분 절대값 a 를, 해당 참조 블록 Ii의 참조 벡터(vx, vy)가 가리키는(지시하는) 어드레스(태이블 요소)의, 그때까지의 SAD값에 가산하고, 그 가산인 SAD값을, 해당 어드레스에 다시쓰기(書戻; write-back; 라이트백)하도록 한다(스텝 S4). 즉, 참조 벡터(vx, vy)에 대응하는 SAD값을, SAD(vx, vy)로 나타내면,

$$SAD(vx, vy) = \sum a = \sum | Io(x, y) - Ii(x+vx, y+vy) | \quad \dots(\text{식 2})$$

[0208]로서 산출하고, 해당 참조 벡터 (vx, vy)가 가리키는 어드레스에 기입(書入; ; write; 써넣기)하도록 한다.

[0209] 다음에, 타겟 블록 Io 내의 모든 좌표 (x, y)의 화소에 대해서, 상기의 연산을 행했는지 여부를 판별하고(스텝 S5), 타겟 블록 Io 내의 모든 좌표 (x, y)의 화소에 대해서는, 아직 연산은 종료하고 있지 않다고 판별했을 때에는, 스텝 S2로 되돌아가고, 타겟 블록 Io 내의 다음의 좌표(x, y)의 화소 위치를 지정하고, 이 스텝 S2 이후(以降)의 처리를 되풀이(繰返; 반복)한다.

[0210] 또, 스텝 S5에서, 타겟 블록 Io 내의 모든 좌표 (x, y)의 화소에 대해서, 상기의 연산을 행했다고 판별했을 때에는, 해당 참조 블록에 대한 SAD값의 산출이 종료했다고 판별해서, 서치 범위 내의 모든 참조 블록, 즉 모든 참조 벡터 (vx, vy)에 대한 상기의 연산 처리를 완료했는지 여부를 판별한다(스텝 S6).

[0211] 스텝 S6에서, 아직 상기의 연산 처리를 완료하고 있지 않은 참조 벡터 (vx, vy)가 있다고 판별하면, 스텝 S1로 되돌아가고, 상기의 연산 처리를 완료하고 있지 않은 다음의 참조 벡터 (vx, vy)를 설정하고, 이 스텝 S1 이후의 처리를 되풀이한다.

[0212] 그리고, 스텝 S6에서, 상기의 연산 처리를 완료하고 있지 않은 참조 벡터 (vx, vy)는 서치 범위 내에 없게 되었다고 판별하면, SAD 테이블이 완성했다고 해서, 해당 완성된 SAD 테이블에서, 최소값으로 되어 있는 SAD값을 검출한다(스텝 S7). 그리고, 해당 최소값으로 되어 있는 SAD값의 어드레스에 대응하는 참조 벡터를 움직임 벡터로서 검출한다(스텝 S8). 여기에서, SAD의 최소값을 SAD(mx, my)라고 써서 나타내면(書表), 목적으로 하는 움직임 벡터는, 위치 (mx, my)를 나타내는 벡터 (mx, my)로서 산출된다.

[0213] 이상으로, 하나의 타겟 블록에 대한 블록 매칭에 의한 움직임 벡터의 검출 처리는, 종료로 된다

[0214] 이상과 같은 블록 매칭을 이용한 센서리스(sensorless)의 손흔들림 움직임 벡터의 검출 방법의 경우, 원리적으로, 픽셀 정밀도의 손흔들림 벡터 검출이 실현가능하고, 또 센서나 렌즈 시프트 등의 기구를 삭감할 수 있기 때

문에, 코스트적으로도 상당히 우위(優位; excellent)이다.

[0215] 그러나, 상술한 종래의 블록 매칭에 의존하는 기술의 연장(延長)에서는, 1화면분의 화소수에 비례해서 SAD 테이블의 규모가 증가하기 때문에, 현재의 500만 화소 오버(over; 이상)의 정지화상 사이즈의 움직임 벡터 검출을, 현실적인 회로 규모로 실현하는 것은 매우 곤란하다.

[0216] 과거, 각 사(各社) 여러가지로 머리를 짜내면서, NTSC(National Television System Committee) 동화상의 고작(高高) 17만 화소의 손흔들림 벡터 검출의 회로 규모 삽감에 고심(苦慮)하고 있던 배경이 있는데다(있을 뿐만 아니라), NTSC 동화상의 경우, 60fps(frame per second; 프레임/초(秒))이기 때문에 손흔들림의 서치 범위는 좁게 끝나지만(좁아도 되지만), 정지화상의 경우, 3fps 정도가 전제로 되고, 서치 범위가 극단적(極端)으로 커지는 것도 문제를 곤란하게 하고 있는 한 요인(一因)이다. 화소수와 같이(同), 서치 범위에도 비례해서 SAD 테이블의 테이블 요소수가 증가하기 때문이다.

[0217] 센서리스 손흔들림 보정을 정지화상으로 실현하는 수법(手法)으로서는, 특히 문헌 3(일본 특개평(特開平) 제7-283999호 공보)을 비롯해서, 몇개 제안은 되어 있다. 특히 문헌 3은, 손흔들림이 발생하지 않을 정도의 짧은 노광 시간 동안에 몇 매인가의 정지화상을 연사(連射; consecutively) 촬영하고, 그 정지화상 사이의 손흔들림 벡터를 구하고, 그 손흔들림 벡터에 따라서, 상기 연사 촬영한 복수매의 정지화상을 평행 이동(및 롤 축방향으로 회전)시키면서 가산(혹은, 평균화)해 감으로써, 최종적으로 손흔들림과 저조도 노이즈가 없는 고화질의 정지화상을 얻는다고 하는 알고리즘이다.

[0218] 실현할 수 있는 레벨의 현실적인 제안으로서는, 특히 문헌 4(일본 특개 제2005-38396호 공보)를 들 수가 있다. 이 특허 문헌 4에 나타내어진(개시된) 것은, 화상을 축소 변환한 사이즈로 움직임 벡터를 구하는 수단과, 동일한 SAD 테이블을 복수의 블록에서 공유(共有)하는 수단으로 구성된다. 화상의 축소 변환과, SAD 테이블의 복수 블록에서의 공유화는, SAD 테이블 사이즈의 삽감을 실현하기 위한, 매우 좋은 수법이며, MPEG(Moving Picture Experts Group) 화상 압축 방식에서의 움직임 벡터 검출이나 신 체인지(scene change; 장면 바뀜) 검출 등, 다른 분야에서도 사용되고 있다.

[0219] 그러나, 이 특허 문헌 4의 알고리즘의 문제점으로서, 화상의 축소 변환과, 그 때의 메모리(DRAM(Dynamic RAM(Random Access Memory))) 액세스에, 시간과 메모리 용량을 소비하는 것과, SAD 테이블을 복수 블록에서 시분할(時分割) 액세스하는 수법이기 때문에, 메모리 액세스가 매우 증가하고, 이 처리에도 시간을 요(要)해 버린다고 하는 문제가 있다. 동화상의 손흔들림 보정에서는, 리얼타임성(性)과 동시에 시스템 지연 시간의 단축이 요구되기 때문에, 이 처리 시간의 문제가 과제로 되어 버린다는 것이다.

[0220] 또, 원화상을 축소 변환할 때에는, 에일리어싱(aliasing)(반환되는 일그러짐(折返歪))이나, 저조도 노이즈 제거를 위한 로우패스 필터를, 축소 처리의 전처리(前處理)로서 실장하는 것이 필요하다. 그러나, 축소 배율에 따라서, 로우패스 필터의 특성이 변화하는데다, 특히, 수직 방향의 로우패스 필터의 경우, 다(多)탭의 디지털 필터로 한 경우에, 많은 라인 메모리와 연산 로직을 필요로 하지 않으면 안되고, 회로 규모 증가의 문제가 생긴다.

[0221] 다른 한편, 블록 매칭을 사용하지 않는 알고리즘도 제안되어 있다(예를 들면, 특히 문헌 5(일본 특개평 제6-86149호 공보), 특허 문헌 6(일본 특개 제2004-343483 공보) 등). 이들은, 2개의 프레임 화상내 각각에서, 어느것인가의 특징점을 복수점 검출하고, 그들 2개의 프레임 사이의 대응지점(對應付; 대응시킴)을 취함으로써 전체의 손흔들림 벡터(글로벌 벡터)를 구하는 수법이다. 혹은, 한쪽편(片方)의 프레임에서만 특징점을 검출하고, 그 특징점 주위에 관해서만, 또다른(나머지) 한쪽의 프레임에 대해서 블록 매칭을 행해도 좋다.

[0222] 이 특허 문헌 5나 특허 문헌 6의 알고리즘은, 회로 규모를 삽감하는데 있어서 매우 유효하며, 이상적(理想的)이기는 하지만, 현실(現實)로는, 2 프레임에 공통해서 존재하는, 화상 전체의 진짜로 특징이 있는 특징점을, 어디까지 효율적으로 수를 줄여서(絞; reduce) 특정할 수 있는지에, 그 실효성이 걸려 있다. 따라서, 이 특허 문헌 5나 특허 문헌 6의 알고리즘에 비하면, 민생기에서는 삼라만상(森羅萬象; all things)이 활상 대상인 한(限), 로버스트성(性)(robustness)에서 블록 매칭 수법이 조금 우위에 있다고 생각된다.

[0223] 상술한 대로, 디지털 카메라 등의 활상 장치에서는, 앞으로 점점 고밀도 화소화가 진행되고, 고성능화가 요구되리라고 예상되지만, 이와 같은 상황에서는, 정지화상의 촬영시의 손흔들림 보정을, 차이로(각속도) 센서를 이용하지 않는 센서리스로 실현하는 의의(意義)는 매우 크다.

[0224] 그래서, 상술한 바와 같이, 블록 매칭을 이용해서, 센서리스로, 손흔들림 움직임 벡터를 검출하고, 그 검출한 움직임 벡터를 이용해서 손흔들림 보정을 행하는 것이 유망하지만, 블록 매칭의 현실 해답으로서, 회로 규모,

처리 속도, 로버스트성을 모두 만족시키는 제안은 이루어져 있지 않은 것이 현상태이다.

[0225] 그리고, 블록 매칭 기법에서, 최대 문제로 되고 있는 것이, SAD 테이블의 증대이다. 이미 기술한 바와 같이, 디지털 카메라에서는, 그 이미지가 500만 화소 이상이 전제인 요즈음(昨今)에 있어서, 화소수에 비례해서 SAD 테이블 사이즈가 커지지 아니할 수 없는 데다(없을 뿐만 아니라), 정지화상의 경우, 3fps 정도이기 때문에, 동화상의 60fps의 손흔들림 범위와 비교해서, 몇 10배(倍)나 넓은 서치 범위가 필요하고, 해당 서치 범위의 확대는, 즉 SAD 테이블의 증대와 똑같기 때문이다.

[0226] 많은 사람(多人數)의 평가 결과, 3fps의 정지화상인 경우에서의 손흔들림 범위는, 전(全)프레임을 100으로 해서 $\pm 10\%$ 정도인 것이 판명되어 있다. 이미 고급기에서는 세상에 나와 있는 1200만 화소를 가정하고, 현상태에서 제안되어 있는 기술 그대로, 필요한 SAD 테이블 사이즈를 견적(見積; estimate; 예측, 예상)하면, 약 80메가비트이다. 게다가, 현실적인 처리 속도를 만족시키려고 하면, 이 SAD 테이블 정보를 격납하는 메모리는, 내장(內藏) SRAM(Static RAM(Random Access Memory))인 것이 요구된다. 반도체 프로세스 레이아웃이 진행(발달)되었다고는 할 수 있으며, 이 사이즈는, 거의(약) 3자리수(桁; digits) 정도, 현실적인 레벨로부터는 동떨어져(far away) 있다.

[0227] 이 실시형태에서는, 이상의 점을 감안하고, 블록 매칭을 이용해서 2 프레임 사이의 움직임 벡터를 검출하는 경우에 있어서, SAD 테이블 사이즈의 대폭적인 축소가 가능한 화상 처리 방법 및 장치를 제공하도록 한다.

[0228] 또, 전술한 블록 매칭에서의 종래의 제안 수법 중, 특히 문헌 4에 기재된 화상의 축소 변환에 의한 SAD 테이블의 축소 수법에 관해서, 2개의 문제를 제기했다. 화상의 축소 변환에 따르는 처리 시간의 증대 및 메모리 용량의 소비와, 화상의 축소 변환에 따르는 에일리어싱 회피를 위한 적절한 로우패스 필터의 실장에 따르는 회로 증대의 문제이다. 이하에 설명하는 실시형태에서는, 이러한 문제점도 해결한 것이다.

[0229] [실시형태에서 이용하는 신규(新規)한 블록 매칭 수법의 개요]

[0230] 이 실시형태에서도, 상술한 블록 매칭을 이용해서, 2프레임 사이의 움직임 벡터를 검출하는 것이지만, 타겟 블록과 참조 블록 사이에서 구해지는 SAD값을, 참조 블록의 참조 벡터에 대응해서 기억하는 것이 아니라, 해당 참조 벡터를 축소하고, 그 축소한 참조 축소 벡터에 대응하는, 해당 참조 축소 벡터 근방의 복수의 참조 벡터에 분산 가산해서 기억하도록 한다.

[0231] 이것에 의해, 종래의 SAD 테이블과 비교해서, SAD 테이블의 사이즈를 대폭(大幅) 축소하도록 하는 것이다.

[0232] 도 6~도 8은, 실시형태에서 이용하는 신규한 블록 매칭 수법의 개요를 설명하기 위한 도면이다. 도 6은, 종래의 SAD 테이블 TBLo와, 실시형태에서 이용하는 신규한 블록 매칭 수법에서 생성되는 축소 SAD 테이블 TBLs와의 관계를 도시하는 것이다.

[0233] 이 실시형태에서도, 도 3에 도시한 바와 같이, 종래와 마찬가지로 참조 프레임에서, 원(元)프레임에 설정된 타겟 블록의 위치를 중심으로 해서 서치 범위가 설정된다. 그리고, 이 서치 범위에서, 전술한 바와 같은 복수의 참조 블록이 설정되고, 각 참조 블록 내의 화소와 타겟 블록 내의 대응하는 화소의 휘도값의 차분의 절대값의 총합, 다시 말해, SAD값이 구해진다.

[0234] 종래에는, 구해진 SAD값은, 도 6에 도시하는 바와 같이, 대상으로 되어 있는 참조 블록의 참조 벡터 RV에 대응하는 어드레스의 테이블 요소 tbl로서 SAD 테이블 TBLo에 기입된다.

[0235] 따라서, 종래의 블록 매칭에서는, 타겟 블록과 참조 블록과의 프레임 화상 위에서의 위치 어긋남량을 나타내는 참조 벡터 RV와, SAD 테이블 TBLo의 각 테이블 요소인 참조 블록의 SAD값과는, 1대 1로 대응하고 있다. 즉, 종래의 SAD 테이블 TBLo에서는, 서치 범위에서 취할 수 있는 참조 벡터 RV와 똑같은 수의 SAD값의 테이블 요소수를 구비하는 것으로 되어 있다.

[0236] 이것에 대해서, 이 실시형태에서의 블록 매칭에서는, 도 6 및 도 7의 (A), (B)에 도시하는 바와 같이, 대상으로 되어 있는 참조 블록의 참조 벡터 RV는, 축소율 $1/n$ (n은 자연수)로 축소되어 참조 축소 벡터 CV로 된다.

[0237] 여기에서, 이하의 설명에서는, 설명의 편의상, 수평 방향 축소 배율과 수직 방향의 축소 배율을 같다고 하고 있지만, 수평 방향 축소 배율과 수직 방향의 축소 배율과는 독립(獨立)된 다른 값이라도 좋다. 또, 나중(後)에 기술하겠지만, 수평 방향 축소 배율과 수직 방향의 축소 배율을 독립적으로(독립해서) 임의의 자연수분의 1로서 설정할 수 있도록 해 두는 쪽이, 유연성(柔軟性)도 높고 안성맞춤(好都合)이다.

[0238] 이 실시형태에서도, 전술한 종래의 블록 매칭 수법에 대해서 설명한 것과 마찬가지로, 서치 범위의 중심으로 되

는 타겟 블록의 위치를 기준 위치 (0, 0)로 하고, 참조 벡터는, 해당 기준 위치로부터의 화소 단위의 수평 방향 및 수직 방향의 어긋남량 (vx, vy) (vx, vy 는, 정수)을 가리키는 것으로 되고, 참조 벡터 RV의 각각은, 참조 벡터 (vx, vy)로 나타내어진다.

[0239] 참조 벡터 (vx, vy)가, 수평 방향 및 수직 방향의 각각에 대해서 $1/n$ 로 축소된 참조 축소 벡터 ($vx/n, vy/n$)로 나타내어지는 위치 ($vx/n, vy/n$)는, 정수가 아니라, 소수(小數; fraction) 성분이 발생하는 일이 있다. 이 때문에, 이 실시형태에서는, 축소전의 원래의 참조 벡터 RV에 대응해서 구해진 SAD값을, 참조 축소 벡터 CV에 가장 가까운 하나의 참조 벡터에 대응하는 테이블 요소로서 기억해 버리면 오차가 생기게 된다.

[0240] 그래서, 이 실시형태에서는, 우선, 참조 축소 벡터 CV로 나타내어지는 위치 ($vx/n, vy/n$) 근방의 복수의 참조 벡터로 나타내어지는 복수의 위치(테이블 요소)를 검출한다. 그리고, 참조 벡터 RV의 참조 블록에 대해서 구해진 SAD값은, 그 검출한 근방의 복수의 참조 벡터에 대응하는 SAD값에 분산해서 가산하도록 한다.

[0241] 이 경우에, 이 실시형태에서는, 참조 축소 벡터 CV로 나타내어지는 위치 근방의 주위의 복수개의 참조 벡터로 나타내어지는 위치에 대응하는 테이블 요소 tb1에 기입해야 할 성분으로서 분산 가산하는 값은, 축소전의 원래의 참조 벡터 RV에 대응해서 구해진 SAD값으로부터, 참조 축소 벡터와 그 근방의 참조 벡터와의 각각이 나타내는 위치의 관계를 이용해서, 상기 근방의 참조 벡터의 각각 대응해서 분산 가산할 SAD값을 산출하고, 산출한 SAD값의 각각을, 대응하는 참조 벡터의 테이블 요소 성분으로서 가산하도록 한다.

[0242] 여기에서, 분산할 뿐만 아니라 가산한다는 것은, 참조 축소 벡터 근방의 복수의 참조 벡터는, 다른(異) 복수의 참조 축소 벡터에 대해서 중복(重複)해서 검출되게 되므로, 하나의 참조 벡터에 대해서, 중복한 SAD값은 가산한다고 하는 의미이다.

[0243] 또한, 참조 축소 벡터 CV가 가리키는 위치 ($vx/n, vy/n$)가, 참조 벡터가 나타내는 위치와 일치하는 경우, 다시 말해, vx/n 및 vy/n 의 값이 정수일 때에는, 주변의 복수의 참조 벡터를 검출할 필요는 없고, 해당 위치 ($vx/n, vy/n$)를 나타내는 참조 벡터에 대응해서, 축소전의 원래의 참조 벡터 RV에 대응해서 구해진 SAD값을 기억하도록 하는 것이다.

[0244] 다음에, 구체예를 들어서, 이상의 처리를 설명한다. 예를 들면, 타겟 블록의 위치를 기준 (0, 0)으로 했을 때에, 도 7의 (A)에 도시하는 바와 같이, (-3, -5)의 위치를 나타내는 참조 블록 RV를, 수평 방향 및 수직 방향으로, $1/n=1/4$ 배로 축소하면, 그 참조 축소 벡터 CV로 나타내어지는 위치는, 도 7의 (B)에 도시하는 바와 같이, (-0.75, -1.25)로 된다.

[0245] 따라서, 참조 축소 벡터 CV로 나타내어지는 위치는 소수 성분이 발생하고, 참조 벡터로 나타내어지는 위치와는 일치하지 않는다.

[0246] 그래서, 이 경우에는, 도 8에 도시하는 바와 같이, 해당 참조 축소 벡터 CV가 나타내는 위치의 근방 위치를 나타내는 복수개의 근방 참조 벡터가 검출된다. 도 8의 예에서는, 하나의 참조 축소 벡터 CV에 대해서, 4개의 근방 참조 벡터 NV1, NV2, NV3, NV4가 검출된다.

[0247] 그리고, 전술한 바와 같이, 이 실시형태에서는, 참조 벡터 RV의 참조 블록에 대해서 구해진 SAD값은, 이들 4개의 근방 참조 벡터 NV1, NV2, NV3, NV4에 대응하는 SAD값으로서 분산 가산된다.

[0248] 이 경우에, 이 실시형태에서는, 4개의 근방 참조 벡터 NV1, NV2, NV3, NV4의 각각에 분산 가산하는 SAD값은, 참조 축소 벡터 CV로 나타내어지는 위치 P0(도 8에서 ×표로서 나타낸다)과, 4개의 근방 참조 벡터 NV1, NV2, NV3, NV4의 각각으로 나타내어지는 위치 P1, P2, P3, P4(도 8에서 ○표시로서 나타낸다)와의 위치 관계를 이용해서 선형 가중 분산값(線形加重分散值)으로서 산출한다.

[0249] 도 8의 예의 경우에는, 참조 축소 벡터 CV로 나타내어지는 위치 P0은, 주변 근방의 4개의 참조 벡터 NV1, NV2, NV3, NV4의 각각으로 나타내어지는 위치 P1, P2, P3, P4를, 수평 방향으로 1:3, 수직 방향으로 3:1로 내분(內分)하는 위치에 있다.

[0250] 그래서, 축소전의 원래의 참조 벡터 RV에 대응해서 구해진 SAD값을 S_a 로 했을 때, 주변 근방의 4개의 참조 벡터 NV1, NV2, NV3, NV4의 각각으로 나타내어지는 위치 P1, P2, P3, P4에 대응하는 SAD 테이블 요소에 분산 가산하는 값 $SADp1, SADp2, SADp3, SADp4$ 의 각각은,

[0251] $SADp1=S_a \times 9/16$

[0252] $SADp2=S_a \times 3/16$

- [0253] $SADp3=S \alpha \times 3/16$
- [0254] $SADp4=S \alpha \times 1/16$
- [0255] 으로 된다.
- [0256] 그리고, 이 실시형태에서는, 구해진 $SADp1$, $SADp2$, $SADp3$, $SADp4$ 의 각각을, 근방의 4개의 참조 벡터 $NV1$, $NV2$, $NV3$, $NV4$ 의 각각으로 나타내어지는 위치 $P1$, $P2$, $P3$, $P4$ 에 대응하는 SAD 테이블 요소에 각각 가산한다.
- [0257] 이 실시형태에서는, 이상의 처리를, 서치 범위 내의 모든 참조 블록에 대해서 행한다.
- [0258] 이상의 것으로 인해, 이 실시형태에서는, 참조 벡터 RV 를 $1/n$ 로 축소하는 경우에는, 모든 참조 벡터에 1대 1로 대응하는 종래 사이즈의 SAD 테이블 $TBLo$ 에 대해서, 수평 방향으로 $1/n$, 또 수직 방향으로 $1/n$ 로 축소한 축소 SAD 테이블 $TBLo$ 를 준비해서, 이 축소 SAD 테이블 $TBLo$ 의 테이블 요소로서, 참조 축소 벡터 CV 근방의 참조 벡터에 대응하는 SAD값을 구하도록 하면 좋다(도 6 참조).
- [0259] 따라서, 이 실시형태의 경우에는, 축소 SAD 테이블 $TBLo$ 의 테이블 요소의 수는, 종래의 SAD 테이블 $TBLo$ 의 테이블 요소수의 $1/n^2$ 로 되고, 테이블 사이즈를 대폭 작게 하는 것이 가능하다.
- [0260] 또한, 상술한 실시형태의 설명에서는, 참조 축소 벡터 CV 근방의 4개의 참조 벡터를 검출하고, 해당 4개의 근방 참조 벡터에 대응하는 SAD 테이블 요소에 대해서, 대조(對照; contracted)의 참조 블록(참조 벡터 RV)에 대해서 산출한 SAD값을 선형 가중 분산 가산하도록 했지만, 참조 축소 벡터 CV 근방의 복수의 참조 벡터의 선택 방법(選方) 및 그 근방 참조 벡터에 대응하는 SAD 테이블 요소에 대한 분산 가산의 방법은, 상술한 예에 한정되는 것은 아니다.
- [0261] 예를 들면, 참조 축소 벡터 CV 근방의 9개 혹은 16개의 참조 벡터를 검출하고, 해당 9개 혹은 16개의 근방 참조 벡터에 대응하는 SAD 테이블 요소에 대해서, 이른바 큐빅(Cubic) 보간에 의한 분산 가산을 행하도록 하면, 보다 정밀도는 높아진다. 그러나, 리얼타임성과 연산 회로의 삭감을 중시하면, 상술한 근방 4개의 참조 벡터에 대응하는 테이블 요소에의 선형 가중 분산 가산이, 보다 유효하다.
- [0262] 이 실시형태에서도, 참조 블록을 서치 범위 내에서 골고루(遍) 이동시키고, 모든 참조 블록의 SAD값에 대해서 SAD 테이블(이 실시형태에서는 축소 SAD 테이블)에의 대입(代入)을 행하는 점은, 종래 수법과 같다.
- [0263] 다만, 종래에는, 참조 벡터와 SAD 테이블 요소의 어드레스가 1대 1로 대응하고 있었기 때문에, SAD 테이블에는 단순한 대입으로 끝났지만(해결되었지만), 이 실시형태에 따른 수법에서는, 참조 블록에 대해서 산출한 SAD값을 분산 가산시키기 때문에, 축소 SAD 테이블에서, 참조 벡터와 테이블 어드레스는 1대 1은 아니다. 따라서, 이 실시형태의 수법의 경우에는, SAD값의 테이블 어드레스에의 단순한 대입이 아니라, 가산해서 대입하는, 이른바 대입 가산일 필요가 있다. 또, 그 때문에, SAD 테이블(축소 SAD 테이블)의 각 테이블 요소는, 최초에 초기화(0 클리어)해 두지 않으면 안 된다.
- [0264] 그런데, 종래의 블록 매칭 수법에서는, 이상과 같이 해서 완성시킨 SAD 테이블에서, SAD값이 최소값으로 되는 테이블 요소를 탐색(探索)하고, 그 최소값으로 되는 테이블 요소의 테이블 어드레스를, 참조 벡터로 변환하면, 움직임 벡터의 검출을 완료했다.
- [0265] 이것에 대해서, 이 실시형태에 의한 수법에서는, SAD 테이블은, 참조 벡터를 축소한 참조 축소 벡터에 대응한 축소 SAD 테이블이기 때문에, 해당 축소 SAD 테이블의 최소값이 그대로 정확한 움직임 벡터에는 대응하고 있지 않다.
- [0266] 무엇보다, 어느 정도의 오차를 허용하는 장치인 경우에는, 축소 SAD 테이블의 최소값으로 되는 테이블 요소의 테이블 어드레스를, 참조 벡터로 변환한 것을, 또 축소율 $1/n$ 의 역수배, 다시 말해 n 배함으로써, 움직임 벡터를 검출하도록 할 수도 있다.
- [0267] 그러나, 보다 정확한 움직임 벡터를 검출하도록 하는 경우에는, 이하에 설명하는 바와 같이, 축소 SAD 테이블의 테이블 요소값에 대해서 보간 처리를 행함으로써, 원래의 벡터 정밀도로, 정확한 움직임 벡터를 검출하도록 한다.
- [0268] [보다 정확한 움직임 벡터를 검출하기 위한 보간 처리의 제1 예]
- [0269] 보다 정확한 움직임 벡터를 검출하기 위한 보간 처리의 제1 예는, 축소 SAD 테이블에서의 복수개의 SAD 테이블 요소값(SAD값)을, 하나의 2차 곡면(曲面)으로 근사(近似)시키는 수법이다. 이 수법은, 전술한 특허 문헌 1에

기재되어 있는 수법을 축소 SAD 테이블에 대해서 적용한 수법이다.

[0270] 즉, 축소 SAD 테이블에서, SAD값이 최소값으로 되는 테이블 요소(정수 정밀도 최소값 테이블 요소(정수 정밀도 테이블 어드레스))와, 이 정수 정밀도 최소값 테이블 요소를 중심으로 하는 복수의 정수 정밀도 테이블 요소를 구하고, 그들 테이블 요소의 SAD값을 이용해서, 최소 제곱법(最小自乘法; least squares method)에 의해 SAD값의 2차 곡면을 결정하고, 이 2차 곡면의 최소값으로 되는 SAD값을 검출하고, 해당 검출한 최소값으로 되는 SAD값에 대응하는 위치(참조 프레임 위에서, 기준 위치에 대해서 어긋난 위치)를 검출하고, 해당 검출한 위치를 소수 정밀도의 최소값 테이블 어드레스(축소 SAD 테이블에서 SAD값이 최소값으로 되는 벡터(최소값 벡터라고 한다)에 대응)로 한다.

[0271] 이 경우, 일의(一意; 오직 하나)의 2차 곡면을 정하기 위해서는, 도 9의 (A) 또는 (B)에 도시하는 바와 같이, 정수 정밀도 최소값 테이블 요소 t_m 과, 해당 테이블 요소 t_m 을 그의 양측 사이에 끼우는(挟; sandwiching) 위치의, 해당 테이블 요소 t_m 근방의 4개의 정수 정밀도 테이블 요소 t_1, t_2, t_3, t_4 가 최저한 필요하다.

[0272] 그리고, 도 10에 도시하는 바와 같이, 참조 프레임의 서치 범위 내의 축소 SAD 테이블에 대응하는 참조 축소 벡터의 범위 내에서, 타겟 프레임의 위치를 기준 위치 (0,0)으로 해서, 수평 방향 및 수직 방향의 어긋남량(참조 축소 벡터에 대응)의 축 vx/n 및 축 vy/n 를 생각함과 동시에, 이를 축 vx/n 및 축 vy/n 에 수직인 축으로서, SAD값의 축을 생각하고, 이를 3축으로 이루어지는 좌표 공간을 상정한다.

[0273] 그리고, 예를 들면 정수 정밀도 최소값 테이블 요소 t_m 의 SAD값과, 해당 정수 정밀도 최소값 테이블 요소 t_m 을 사이에 끼우는 2개의 테이블 요소 t_1, t_3 의 SAD값으로부터, 도 10의 좌표 공간에서 2차 곡선을 생성한다. 또, 정수 정밀도 최소값 테이블 요소 t_m 의 SAD값과, 해당 최소값 테이블 요소 t_m 을 사이에 끼우는 다른 2개의 테이블 요소 t_2, t_4 의 SAD값으로부터, 도 10의 좌표 공간에서, 다른 2차 곡선을 생성한다. 그리고, 이를 2개의 2차 곡선을 포함하는 2차 곡면(201)을, 최소 제곱법에 의해 구하고, 그 2차 곡면(201)을, 도 10에 도시하는 바와 같이, 좌표 공간에서 생성한다.

[0274] 그리고, 생성된 SAD값의 2차 곡면(201)의 최소값(202)을 검출하고, 그 최소값을 취하는 SAD값에 대응하는 위치 ($vx/n, vy/n$)(도 10의 위치(203))를 검출하고, 해당 검출한 위치($vx/n, vy/n$)를, 소수 정밀도의 테이블 요소(테이블 어드레스)로서 검출한다. 그리고, 검출한 소수 정밀도 테이블 요소에 대응하는 벡터(최소값 벡터)(204)를, 도 11에 도시하는 바와 같이 n 배해서, 원래의 크기 정밀도의 움직임 벡터(205)를 얻는다.

[0275] 예를 들면, 도 12에 도시하는 바와 같이, 참조 벡터를 1/4로 축소한 경우에서의 축소 SAD 테이블 TBLs의, 소수 정밀도 테이블 요소의 최소값 어드레스로부터 구해지는 최소값 벡터(204)가, (-0. 777, -1. 492)인 경우에, 이들을 4배했지만(-3. 108, -5. 968), 움직임 벡터(205)로 된다. 이 움직임 벡터(205)는, 원화상의 스케일에서의 움직임 벡터를 재현한 것으로 되고 있다.

[0276] 이상의 설명은, 정수 정밀도 최소값 테이블 요소 t_m 과, 그 근방의 4테이블 요소를 이용한 경우로서 설명했지만, SAD값의 2차 곡면을 최소 제곱법에 의해 구하기 위해서는, 보다 많은 복수 근방 테이블 요소를 이용하는 쪽이 좋다. 그래서, 일반적으로는, 정수 정밀도 최소값 테이블 요소 t_m 을 중심으로 해서, 그 주위의 수평 방향×수직 방향= $m \times m$ 개(m 은 3이상의 정수)의 직사각형 영역의 테이블 요소를 이용하도록 한다.

[0277] 그러나, 이 복수 근방 테이블 요소의 수는, 많으면 좋다고 하는 것이 아니라, 넓은 범위의 테이블 요소를 이용하면, 연산량의 증가를 초래하는데다, 화상 패턴에 의존하는 로컬 미니멈(local minimum)의 위치를 사용해 버릴 가능성도 높아지므로, 적절한 복수 근방 테이블 요소의 수로 이루어지는 직사각형 영역의 테이블 요소를 이용하도록 한다.

[0278] 적절한 복수 근방 테이블 요소의 수로 이루어지는 직사각형 영역의 테이블 요소의 예로서, 이 실시형태에서는, 정수 정밀도 최소값 테이블 요소 t_m 을 중심으로 해서, 그 주위의 수평 방향×수직 방향= 3×3 개의 직사각형 영역의 테이블 요소를 이용하도록 하는 예와, 정수 정밀도 최소값 테이블 요소 t_m 을 중심으로 해서, 그 주위의 수평 방향×수직 방향= 3×3 개의 직사각형 영역의 테이블 요소를 이용하도록 하는 예에 대해서 설명한다.

[0279] [3×3개의 직사각형 영역의 테이블 요소를 이용하는 예]

[0280] 도 13에, 정수 정밀도 최소값 테이블 요소 t_m 을 중심으로 해서, 그 주위의 수평 방향×수직 방향= 3×3 개의 직사각형 영역(도 13에서 칠(塗)을 부가해서(회색 블록으로) 나타내고 있다)의 테이블 요소를 이용하도록 하는 예를 도시한다.

[0281] 이 도 13의 예의 경우에는, 도 13의 (A)에 도시하는 바와 같이, 정수 정밀도 최소값 테이블 요소 t_m 과, 그 근방

의 8개의 근방 테이블 요소의 SAD값을 이용해서, 도 13의 (B)에 도시하는 바와 같은 2차 곡면(201)을, 최소 제곱법에 의해 생성한다. 그리고, 생성된 SAD값의 2차 곡면(201)의 최소값(202)을 검출하고, 그 최소값을 취하는 SAD값에 대응하는 위치 $(vx/n, vy/n)$ (도 13의 (B)의 위치(203))를 검출하고, 해당 검출한 위치(203)를, 소수정밀도의 최소값 테이블 요소 위치(소수 정밀도 최소값 테이블 어드레스)로서 검출한다.

[0282] 그리고, 검출한 소수 정밀도 테이블 요소 위치(203)에 대응하는 벡터(최소값 벡터)(204)를, 전술한 도 11에 도시하는 바와 같이 n 배해서, 원래의 크기 정밀도의 움직임 벡터(205)를 얻는다.

[0283] 여기에서, SAD값의 2차 곡면(201)의 최소값(202)에 대응하는 위치(203)의 산출 방법은, 다음과 같이 된다. 즉, 도 14에 도시하는 바와 같이, 정수 정밀도 최소값 테이블 요소 t_m 의 위치를 원점 $(0, 0)$ 으로 하는 (x, y) 좌표를 생각한다. 이 경우, 주변의 8개의 테이블 요소의 위치는, 3개의 x축 방향의 위치, 즉 $x=-1, x=0, x=1$ 과, 3개의 Y축 방향의 위치, 즉 $y=-1, y=0, y=1$ 과의 배합(繰合; arrange)으로 나타내어지고, $(-1, -1), (0, -1), (1, -1), (-1, 0), (0, 0), (1, 0), (-1, 1), (0, 1), (1, 1)$ 의 8위치로 된다.

[0284] 그리고, 도 14의 테이블에서의 각 테이블 요소의 SAD값을, S_{xy} 로 한다. 따라서, 예를 들면 정수 정밀도 최소값 테이블 요소 t_m (위치 $(0, 0)$)의 SAD값은 S_{00} 으로 나타내어지고, 또 오른쪽 아래(右下)의 위치 $(1, 1)$ 의 테이블 요소값의 SAD값은 S_{11} 로 나타내진다.

[0285] 그러자, 정수 정밀도 최소값 테이블 요소 t_m 의 위치를 원점 $(0, 0)$ 으로 하는 (x, y) 좌표에서의 소수 정밀도의 위치 (dx, dy) 는, 도 15에 도시하는 (식 A) 및 (식 B)에 의해, 구할 수가 있다.

[0286] 도 15의 (식 A) 및 (식 B)에서,

[0287] $x=-1$ 일 때, $Kx=-1$

[0288] $x=0$ 일 때, $Kx=0$

[0289] $x=1$ 일 때, $Kx=1$

[0290] 로 된다. 또,

[0291] $y=-1$ 일 때, $Ky=-1$

[0292] $y=0$ 일 때, $Ky=0$

[0293] $y=1$ 일 때, $Ky=1$

[0294] 로 된다.

[0295] 이렇게 해서 구해진 소수 정밀도의 위치 (dx, dy) 는, 정수 정밀도 최소값 테이블 요소 t_m 의 위치를 원점 $(0, 0)$ 으로 하는 위치이므로, 이 소수 정밀도의 위치 (dx, dy) 와 정수 정밀도 최소값 테이블 요소 t_m 의 위치로부터, 구하는 서치 범위의 중심 위치로부터의 위치(203)를 검출할 수가 있다.

[0296] [4×4개의 직사각형 영역의 테이블 요소를 이용하는 예]

[0297] 도 16에, 정수 정밀도 최소값 테이블 요소 t_m 을 거의 중심으로 해서, 그 주위의 수평 방향×수직 방향=4×4개의 직사각형 영역의 테이블 요소(도 16에서 칠을 부가해서 나타내고 있다)를 사용하도록 하는 예를 도시한다.

[0298] 이 경우에, 정수 정밀도 최소값 테이블 요소 t_m 과, 그 근방의 8테이블 요소(3×3)나, 그 근방의 24테이블 요소(5×5)와 같이, 상기 m 의 값이 홀수(奇數; odd integer)인 경우에는, 정수 정밀도 최소값 테이블 요소 t_m 은, 항상, 사용하는 직사각형 영역의 복수의 테이블 요소의 중심으로 되기 때문에, 사용하는 테이블 범위는 단순하게 결정한다.

[0299] 이것에 대해서, 근방의 15테이블 요소 (4×4)와 같이, m 이 짝수인 경우에는, 정수 정밀도 최소값 테이블 요소 t_m 은, 사용하는 직사각형 영역의 복수의 테이블 요소의 중심 위치로는 되지 않으므로, 약간(若干)의 연구(工夫; device)가 필요하게 된다.

[0300] 다시말해, 정수 정밀도 최소값 테이블 요소 t_m 에서 보아, 수평 방향으로 좌우의 인접 테이블 요소의 SAD값을 비교하고, 작은 값으로 된 측의 방향의, 해당 방향의 인접 테이블 요소에 인접하는 테이블 요소를 근방 테이블 요소의 4열째(列目)로서 채용한다. 마찬가지로, 수직 방향으로 상하의 인접 테이블 요소의 SAD값을 비교하고, 작은 값으로 된 측의 방향의, 해당 방향의 인접 테이블 요소에 인접하는 테이블 요소를 근방 테이블 요소의 4행째(行目)로서 채용한다.

- [0301] 도 16의 예에서는, 정수 정밀도 최소값 테이블 요소 t_m 의 수평 방향으로 좌우의 인접 테이블 요소의 SAD값은, 「177」과 「173」이므로, SAD값이 작은 오른쪽측(右隣; right side)의 값 「173」의 테이블 요소의 더욱 오른쪽 옆의 열을 제4열째로서 채용한다. 또, 정수 정밀도 최소값 테이블 요소 t_m 의 수직 방향으로 상하의 인접 테이블 요소의 SAD값은, 「168」과 「182」이므로, SAD값이 작은 위쪽측(上隣; above)의 값 「168」의 테이블 요소의 더욱 위쪽측의 행을 제4행째로서 채용한다.
- [0302] 그리고, 도 16의 예의 경우에는, 정수 정밀도 최소값 테이블 요소 t_m 과, 그 근방의 15개의 근방 테이블 요소의 SAD값을 이용해서, 2차 곡면(201)을, 최소 제곱법에 의해 생성한다. 그리고, 생성된 SAD값의 2차 곡면(201)의 최소값(202)을 검출하고, 그 최소값을 취하는 SAD값에 대응하는 위치 $(vx/n, vy/n)$ (도 16의 위치(203))를 검출하고, 해당 검출한 위치(203)를, 소수 정밀도의 최소값 테이블 요소 위치(소수 정밀도 최소값 테이블 어드레스)로서 검출한다.
- [0303] 그리고, 검출한 소수 정밀도 테이블 요소 위치(203)에 대응하는 벡터(최소값 벡터)(204)를, 전술한 도 11에 도시하는 바와 같이 n 배해서, 원래의 크기 정밀도의 움직임 벡터(205)를 얻는다.
- [0304] 여기에서, 이 예의 경우에는 SAD값의 2차 곡면(201)의 최소값(202)에 대응하는 위치(203)의 산출 방법은, 다음과 같이 된다. 즉, 도 17에 도시하는 바와 같이, 정수 정밀도 최소값 테이블 요소 t_m 의 위치를 원점 $(0, 0)$ 으로 하는 (x, y) 좌표를 생각한다.
- [0305] 이 예의 경우에는, 16테이블 요소로 이루어지는 직사각형 영역 중에서의 정수 정밀도 최소값 테이블 요소 t_m 의 위치에 따라서, 도 17의 (A), (B), (C), (D)와 같은 4가지의 테이블 요소 배치(配置; layouts)를 생각할 필요가 있다.
- [0306] 이 경우, 주변의 15개의 테이블 요소의 위치는, 도 17의 (A), (B), (C), (D)로부터 알 수 있는 바와 같이, 4개의 x축 방향의 위치, 즉, $x=-1, x=0, x=1, x=2$ 또는 $x=-2$ 와, 4개의 Y축 방향의 위치, 즉, $y=-1, y=0, y=1, y=2$ 또는 $y=-2$ 와의 배합으로 나타내어지는 15위치로 된다.
- [0307] 그리고, 도 17의 테이블에서의 각 테이블 요소의 SAD값을, S_{xy} 로 한다. 따라서, 예를 들면 정수 정밀도 최소값 테이블 요소 t_m (위치 $(0, 0)$)의 SAD값은 S_{00} 으로 나타내어지고, 또 위치 $(1, 1)$ 의 테이블 요소값의 SAD값은 S_{11} 로 나타내어진다.
- [0308] 그러면, 정수 정밀도 최소값 테이블 요소 t_m 및 그 주변의 16테이블 요소로 이루어지는 직사각형 영역 중의 중심 위치를 원점 $(0, 0)$ 으로 하는 (x, y) 좌표에서의 소수 정밀도의 위치 (dx, dy) 는, 도 18에 도시하는 (식 C) 및 (식 D)에 의해, 구할 수가 있다.
- [0309] 여기에서, 도 18의 (식 C) 및 (식 D)에서, K_x 및 K_y 는, 정수 정밀도 최소값 테이블 요소 t_m 및 그 주변의 16테이블 요소로 이루어지는 직사각형 영역 중의 중심 위치를 원점 $(0, 0)$ 으로 하는 (K_x, K_y) 좌표를 생각했을 때의, 상기 도 17의 (A), (B), (C), (D)에 도시한 4가지의 테이블 요소 배치에 따른 값으로 된다. 이 때의 (K_x, K_y) 좌표를 도 19에 도시한다.
- [0310] 즉, 도 17의 (A)의 경우에는,
- [0311] $x=-2$ 일 때, $K_x=-1.5$
- [0312] $x=-1$ 일 때, $K_x=-0.5$
- [0313] $x=0$ 일 때, $K_x=0.5$
- [0314] $x=1$ 일 때, $K_x=1.5$
- [0315] 로 된다. 또,
- [0316] $y=-2$ 일 때, $K_y=-1.5$
- [0317] $y=-1$ 일 때, $K_y=-0.5$
- [0318] $y=0$ 일 때, $K_y=0.5$
- [0319] $y=1$ 일 때, $K_y=1.5$
- [0320] 로 된다.

- [0321] 또, 도 17의 (B)의 경우에는,
 [0322] $x=-2$ 일 때, $Kx=-1.5$
 [0323] $x=-1$ 일 때, $Kx=-0.5$
 [0324] $x=0$ 일 때, $Kx=0.5$
 [0325] $x=1$ 일 때, $Kx=1.5$
 [0326] 로 된다. 또,
 [0327] $y=-1$ 일 때, $Ky=-1.5$
 [0328] $y=0$ 일 때, $Ky=-0.5$
 [0329] $y=1$ 일 때, $Ky=0.5$
 [0330] $y=2$ 일 때, $Ky=1.5$
 [0331] 로 된다.
 [0332] 또, 도 17의 (C)의 경우에는,
 [0333] $x=-1$ 일 때, $Kx=-1.5$
 [0334] $x=0$ 일 때, $Kx=-0.5$
 [0335] $x=1$ 일 때, $Kx=0.5$
 [0336] $x=2$ 일 때, $Kx=1.5$
 [0337] 로 된다. 또,
 [0338] $y=-2$ 일 때, $Ky=-1.5$
 [0339] $y=-1$ 일 때, $Ky=-0.5$
 [0340] $y=0$ 일 때, $Ky=0.5$
 [0341] $y=1$ 일 때, $Ky=1.5$
 [0342] 로 된다.
 [0343] 또, 도 17의 (D)의 경우에는,
 [0344] $x=-1$ 일 때, $Kx=-1.5$
 [0345] $x=0$ 일 때, $Kx=-0.5$
 [0346] $x=1$ 일 때, $Kx=0.5$
 [0347] $x=2$ 일 때, $Kx=1.5$
 [0348] 로 된다. 또,
 [0349] $y=-1$ 일 때, $Ky=-1.5$
 [0350] $y=0$ 일 때, $Ky=-0.5$
 [0351] $y=1$ 일 때, $Ky=0.5$
 [0352] $y=2$ 일 때, $Ky=1.5$
 [0353] 로 된다.
 [0354] 또, 도 18의 (식 C) 및 (식 D)에서의 Δx 및 Δy 는, (Kx, Ky) 좌표에 대한 도 17의 (A), (B), (C), (D)의 각 테이블 요소 배치에서의 (x, y) 좌표와의 어긋남량을 나타내고 있으며,
 [0355] 도 17의 (A)의 경우에는, $\Delta x=-0.5$, $\Delta y=-0.5$,

[0356] 도 17의 (B)의 경우에는, $\Delta x=-0.5$, $\Delta y=0.5$,

[0357] 도 17의 (C)의 경우에는, $\Delta x=0.5$, $\Delta y=-0.5$,

[0358] 도 17의 (D)의 경우에는, $\Delta x=0.5$, $\Delta y=0.5$,

[0359] 로 된다.

[0360] 이렇게 해서 구해진 소수 정밀도의 위치 (dx , dy)는, 정수 정밀도 최소값 테이블 요소 t_m 의 위치를 원점 (0, 0)으로 하는 위치이므로, 이 소수 정밀도의 위치 (dx , dy)와 정수 정밀도 최소값 테이블 요소 t_m 의 위치로부터, 구하는 서치 범위의 중심 위치로부터의 위치(203)를 검출할 수가 있다.

[0361] [보다 정확한 움직임 벡터를 검출하기 위한 보간 처리의 제2 예]

[0362] 보다 정확한 움직임 벡터를 검출하기 위한 보간 처리의 제2 예는, 축소 SAD 테이블에서의 정수 정밀도 최소값 테이블 요소를 포함하는 복수개의 수평 방향의 테이블 요소의 SAD값을 이용해서 수평 방향의 3차 곡선을 생성함과 동시에, 정수 정밀도 최소값 테이블 요소를 포함하는 복수개의 수직 방향의 테이블 요소의 SAD값을 이용해서 수직 방향의 3차 곡선을 생성하고, 각각의 3차 곡선의 극소값(極小值)으로 되는 위치 (vx , vy)를 검출해서, 검출한 위치를 소수 정밀도의 최소값 어드레스로 하는 것이다.

[0363] 도 20은, 이 제2 예를 설명하기 위한 도면이다. 전술한 제1 예와 마찬가지로 해서, 정수 정밀도 최소값 테이블 요소 t_m 과, 이 정수 정밀도 최소값 테이블 요소를 중심으로 하는 복수의 정수 정밀도 테이블 요소, 도 20의 예에서는, $4 \times 4 = 16$ 개의 테이블 요소를 구한다(도 20의 (A)에서 칠을 부가한 부분 참조).

[0364] 다음에, 제1 예와 마찬가지로 해서, 도 20의 (B)에 도시하는 바와 같이, 참조 프레임의 서치 범위 내의 축소 SAD 테이블에 대응하는 참조 축소 벡터의 범위 내에서, 타겟 프레임의 위치를 기준 위치 (0, 0)로 해서, 수평 방향 및 수직 방향의 어긋남량(참조 축소 벡터에 대응)의 축 vx/n 및 축 vy/n 을 생각함과 동시에, 이들 축 vx/n 및 축 vy/n 에 수직인 축으로서, SAD값의 축을 생각하고, 이들 3축으로 이루어지는 좌표 공간을 상정한다.

[0365] 다음에, 정수 정밀도 최소값 테이블 요소 t_m 주위의 16개의 테이블 요소 중, 정수 정밀도 최소값 테이블 요소 t_m 을 포함하는 4개의 수평 방향의 테이블 요소의 SAD값을 이용해서, 상기 좌표 공간에 수평 방향의 3차 곡선 (206)을 생성한다. 이 수평 방향의 3차 곡선(206)의 극소값에 대응하는 수평 방향의 위치 vx/n 로서, 소수 정밀도 최소값 테이블 요소 위치의 수평 방향 위치를 검출한다.

[0366] 다음에, 정수 정밀도 최소값 테이블 요소 t_m 주위의 16개의 테이블 요소 중, 정수 정밀도 최소값 테이블 요소 t_m 을 포함하는 4개의 수직 방향의 테이블 요소의 SAD값을 이용해서, 상기 좌표 공간에 수직 방향의 3차 곡선 (207)을 생성한다. 이 수직 방향의 3차 곡선(207)의 극소값에 대응하는 수직 방향의 위치 vy/n 로서, 소수 정밀도 최소값 테이블 요소 위치의 수직 방향 위치를 검출한다.

[0367] 이상에 의해 구한 소수 정밀도 최소값 테이블 요소 위치의 수평 방향의 위치와, 수직 방향의 위치로부터, 소수 정밀도 최소값 테이블 요소 위치(소수 정밀도 최소값 테이블 어드레스)(208)를 검출한다. 그리고, 해당 검출한 소수 정밀도 테이블 요소 위치(208)에 대응하는 벡터(최소값 벡터)(209)를, 전술한 도 11에 도시하는 바와 같이 n 배해서, 원래의 크기 정밀도의 움직임 벡터를 얻는다.

[0368] 즉, 제2 예는, 제1 예에서 설명한 방법에 의해, 수평 방향, 수직 방향의 각각의 4개의 테이블 요소를 확정하고, 도 20의 (B)에 도시하는 바와 같이, 수평 방향, 수직 방향의 각각에서, 3차 곡선을 일의적(一意)으로 정하는 수법이다.

[0369] 여기에서, SAD값의 3차 곡선(206 및 209)의 최소값(202)에 대응하는 위치(208)의 산출 방법은, 다음과 같이 된다. 즉, 수평 방향 또는 수직 방향의 어느것인가의 방향에서의 3차 곡선에서, 최소값 근방의 4점의 SAD값을, 상기 수평 방향 또는 수직 방향의 어느것인가의 방향을 따른 순번(順番)으로, S_0 , S_1 , S_2 , S_3 으로 했을 때, 소수 정밀도의 최소값이, 도 21에 도시하는 3개의 구간 R_a , R_b , R_c 의 어느것인가에 있는지에 따라, 최소값을 취하는 소수 성분 u 를 산출하는 식이 다르다.

[0370] 여기에서, 구간 R_a 는 SAD값 S_0 으로 되는 위치와 SAD값 S_1 로 되는 위치와의 사이의 구간, R_b 는 SAD값 S_1 로 되는 위치와 SAD값 S_2 로 되는 위치와의 사이의 구간, R_c 는 SAD값 S_2 로 되는 위치와 SAD값 S_3 으로 되는 위치와의 사이의 구간이다.

[0371] 그리고, 소수 정밀도의 최소값이, 도 21에 도시하는 구간 R_a 에 있을 때에는, 도 22의 (식 E)에 의해, 정수 정밀

도의 최소값의 위치에 대한 최소값을 취하는 위치까지의 어긋남의 소수 성분 u 가 산출된다.

[0372] 또, 마찬가지로, 소수 정밀도의 최소값이, 도 21에 도시하는 구간 Rb 에 있을 때에는, 도 22의 (식 F)에 의해, 정수 정밀도의 최소값의 위치에 대한 최소값을 취하는 위치까지의 어긋남의 소수 성분 u 가 산출된다.

[0373] 또, 소수 정밀도의 최소값이, 도 21에 도시하는 구간 Rc 에 있을 때에는, 도 22의 (식 G)에 의해, 정수 정밀도의 최소값의 위치에 대한 최소값을 취하는 위치까지의 어긋남의 소수 성분 u 가 산출된다.

[0374] 그리고, 소수 정밀도의 최소값이, 도 21에 도시하는 3개의 구간 Ra , Rb , Rc 의 어느것에 있는지의 판별은, 다음과 같이 해서 행한다.

[0375] 즉, 도 23은, 그 판별을 설명하기 위한 도면이다. 도 23의 (A), (B), (C)에 도시하는 바와 같이, 우선, 정수 정밀도의 SAD값의 최소값 S_{min} 과, 2번째로 작은 정수 정밀도의 SAD값 S_{n2} 를 검출하고, 소수 정밀도의 최소값은, 검출된 정수 정밀도의 SAD값의 최소값 S_{min} 의 위치와, 2번째로 작은 정수 정밀도의 SAD값 S_{n2} 의 위치와의 사이의 구간에 존재한다고 해서 검출한다. 다음에, 정수 정밀도의 SAD값의 최소값 S_{min} 과, 2번째로 작은 정수 정밀도의 SAD값 S_{n2} 가, 도 21에 도시한 SAD값 S_0 , S_1 , S_2 , S_3 의 어느 위치로 되어 있는지에 따라, 검출한 구간이 구간 Ra , Rb , Rc 의 어느것인지의 판별을 행한다.

[0376] 또한, 도 23의 (D)에 도시하는 바와 같이, 정수 정밀도의 SAD값의 최소값 S_{min} 이 SAD값의 또는 위치에 있어서, 4개의 테이블 요소값의 끝(端; edge; 가장자리)에 위치하는 경우에는, 최소 위치를 추정할 수 없다고 해서, 이 실시형태에서는, 에러(error)로서 취급하고, 최소값 위치의 산출은 행하지 않도록 한다. 무엇보다, 이 도 23의 (D)과 같은 경우에 있어서도, 최소값 위치를 산출하도록 해도 좋다.

[0377] 이상과 같이 해서, 이 실시형태에 따르면, $1/n^2$ 로 스케일다운한 작은 사이즈의 축소 SAD 테이블을 이용해서, 원래의 화상 스케일에서의 움직임 벡터를 검출할 수가 있다. 이 경우에, $1/n^2$ 로 스케일다운한 작은 사이즈의 축소 SAD 테이블을 이용하고 있음에도 불구하고, 종래와 거의 마찬가지 벡터 검출 결과가 얻어지는 것을 도 24에 도시한다.

[0378] 도 24의 횡축(橫軸)은, 수평 방향 또는 수직 방향의 한쪽에 대한 1차원 방향의 축소 배율 n 이며, 또 종축(縱軸)은, 검출되는 움직임 벡터에 대한 오차(벡터 오차)를 나타내고 있다. 도 24의 벡터 오차의 수치(數值)는 화소수로 나타내어지고 있다.

[0379] 도 24에서, 곡선 (301)은, 축소 배율에 대한 벡터 오차의 평균값이다. 또, 곡선 (302)는, 축소 배율에 대한 벡터 오차의 분산 σ 의 3배값(3σ (99.7%))을 나타내고 있다. 곡선 (303)은, 곡선 (302)의 근사 곡선을 나타내고 있다.

[0380] 도 24는, 1차원 방향의 축소 배율 n 에 대한 벡터 오차를 도시하고 있지만, SAD 테이블은 2차원이기 때문에, 도 24에 도시되는 것의 2승(乘)의 비율로 테이블 사이즈(테이블 요소수)가 삭감되는데 대해, 벡터 오차는, 선형 정도로밖에 증가하지 않기 때문에, 이 실시형태에 따른 수법의 유용성을 알 수 있다.

[0381] 또, $n=64$ (축소율 1/64)배의 축소 배율에서도, 벡터 오차는 작고, 전혀 다른 움직임 벡터를 검출 출력으로 하는 바와 같은 파탄(破綻; failure)은 보이지 않기 때문에, 실질적으로, 1/4096로, SAD 테이블의 사이즈를 삭감 가능하다고 할 수 있다.

[0382] 또, 전술한 바와 같이, 동화상의 손흔들림 보정에서는, 리얼타임성과 시스템 지연의 삭감이 강하게 요구되는데 대해, 정밀도에 대해서는, 파탄한 전혀 다른 움직임 벡터가 검출되는 경우를 제외하고, 어느 정도의 벡터 검출 오차에 대해서 관용적이다(寬容; 너그롭다). 따라서, 파탄하지 않는 채 SAD 테이블의 사이즈를 크게 삭감할 수 있는, 이 실시형태는 유용성이 높다고 할 수 있다.

[0383] 이상 설명한 실시형태의 화상 처리 방법은, 종래 수법으로서 설명한 특허 문헌 4에 기재된 화상을 축소 변환한 사이즈로 움직임 벡터를 검출하는 수법에 비해서, 다음에 드는 2개의 점(点)에 대해서, 크게 다른 메리트(이점)를 가지는 것이다.

[0384] 우선, 첫째로, 이 실시형태에 따른 수법은, 특허 문헌 4에 기재된 종래 수법과 달리, 화상을 축소 변환하는 프로세스를 전혀 필요로 하지 않는다. 이 실시형태에 따른 수법에서는, 참조 블록에 대해서 산출한 SAD값을, SAD 테이블(축소 SAD 테이블)에 대입 가산할 때에, 동시에 축소 배율에 상당(相當)하는 어드레스 변환을 행하기 때문이다.

- [0385] 이것에 의해, 이 실시형태에 따른 수법에서는, 특히 문헌 4에 기재된 종래 수법과 같은 화상의 축소 변환을 위한 로직도, 축소한 화상을 메모리에 격납하는 시간 및 뱀드폭의 낭비(浪費)도, 축소 화상을 메모리에 붙이는(貼) 영역 확보도 필요없다고 하는 메리트를 가진다.
- [0386] 특히 문헌 4에 기재된 종래 수법의 또다른 하나의 중요한 문제점으로서, 전술하면서도 사용한 바와 같이, 화상을 축소 변환할 때의 에일리어싱(반환 일그리짐)이나, 저조도 노이즈 제거를 위한 로우패스 필터의 존재의 문제가 있다. 즉, 화상 축소할 때에는, 적절한 로우패스 필터를 통과시키고 나서 리샘플링을 행하지 않으면 안되고, 그렇지 않으면, 불요한 에일리어싱이 발생하고, 그 축소 화상을 이용한 움직임 벡터의 정밀도가 현저하게 손상되기 때문이다.
- [0387] 축소 변환시의 이상적인 로우패스 필터의 특성으로서는, sinc 함수와 유사(類似)한 함수인 것이, 이론적으로 증명되어 있다. sinc 함수 자체는, $\sin(x\pi)/(x\pi)$ 의 형태로 나타내어지는 컷오프 주파수 $f/2$ 의 무한 텁의 FIR(Finite Impulse Response) 필터이지만, 축소 배율 $1/n$ 시의 이상적인 컷오프 주파수 $f/(2n)$ 의 로우패스 필터로서는, $\sin(x\pi/n)/(x\pi/n)$ 로 나타내어진다. 그러나, 이것도 sinc 함수의 한 형태로서 좋다.
- [0388] 도 25~도 27의 상측(上側)에는, 각각 축소 배율이 $1/2$ 배, $1/4$ 배, $1/8$ 배일 때의 sinc 함수(로우패스 필터의 이상적인 특성)의 형상을 도시한다. 이 도 25~도 27로부터, 축소 배율이 커지면 커질수록, 함수가 텁 축방향으로 확대해 가는 것을 알 수 있다. 다시말해, 무한 텁의 sinc 함수를 주요한 계수만으로 근사하는 경우에도, FIR 필터의 텁수(數)를 증가시키지 않으면 안 된다고 할 수 있다.
- [0389] 또, 일반적으로, 보다 낮은 대역의 컷오프 주파수를 실현하는 필터는, 필터 형상보다도 텁수가, 그 성능에 대해서 지배적으로 되어 가는 것이 알려져 있다.
- [0390] 따라서, 특히 문헌 4에 기재된 종래 수법의 축소 화상을 이용하는 움직임 벡터 연산 수법의 경우, 화상의 축소 배율이 커지면 커질수록, 그 SAD 테이블 색감 효과가 큼에도 불구하고, 화상을 생성할 때의 전처리용 필터로서의 로우패스 필터는, 축소 배율이 커지면 커질수록, 코스트가 증가해 버린다고 하는 모순(矛盾)을 아울러 가지는 것이다.
- [0391] 일반적으로, 고차(高次) 텁의 FIR 필터를 실현하는 경우, 연산 로직의 코스트가 텁수의 2승에 비례해서 증가하기 때문에, 문제로 되지만, 보다 큰 문제는, 수직 필터 실현을 위한 라인 메모리수의 증가이다. 근년(近年; 최근)의 디지털 스틸 카메라에서는, 화소수 향상에 따르는 라인 메모리의 사이즈 색감을 위해서, 이른바 스트랩 처리(短冊處理; strap processing)를 행하고 있지만, 예를 들면 1라인당(當)의 사이즈를 색감했다고 해도, 라인 메모리 그 자체의 개수(本數)가 증가하는 것은, 물리 레이아웃 에리어에서 환산되는 토탈 코스트를 현저하게 밀어올린다(상승시킨다).
- [0392] 이상, 기술한 바와 같이, 특히 문헌 4에 기재된 종래 수법의 화상 축소에 의한 어프로치는, 특히 수직 로우패스 필터의 실현에서, 큰 벽이 가로막혀 있는 것을 알 수 있다. 그에 대해서, 본 발명의 수법은, 전혀 다른 형(形)으로 간결하게 이 문제를 해결하고 있다.
- [0393] 도 25~도 27의 하측(下側)에, 이 실시형태에서 이용하는 신규한 블록 매칭 수법에서의 로우패스 필터의 이미지를 도시한다. 이 실시형태에서 이용하는 신규한 블록 매칭 수법에서는, 화상 축소 처리를 수반(伴; 실행)하고 있지 않지만, 축소 SAD 테이블의 생성 연산 과정에서의 로우패스 필터의 이미지를 도시한 것이다.
- [0394] 도 25~도 27의 하측에 도시되는 바와 같이, 이 로우패스 필터의 특성은, sinc 함수의 주요 계수 부분을 선형으로 근사시킨, 심플한 필터 특성이지만, 축소 배율에 연동(運動)해서 텁수가 증가하고 있는 것을 알 수 있다. 이것은, 앞서 기술한, 컷오프 주파수가 낮아질 수록, 로우패스 필터의 성능은 텁수가 지배적으로 된다고 하는 사실에 매우 적합하다. 즉, 실시형태의 선형 가중 분산 가산을 행하는 처리와 같은, 본 발명에 있어서의 SAD값의 분산 가산을 행하는 처리 그 자체가, 배율 연동의 고성능 로우패스 필터를, 심플한 회로로 실현하고 있는 것과 등가인 것이다.
- [0395] 이 로우패스 필터에 관련해서, 그 밖에도 메리트가 있다. 특히 문헌 4에 기재된 종래 수법에서는, 로우패스 필터를 통과시킨 후, 리샘플링함으로써 화상을 축소하지만, 이 시점에서 상당수의 화상 정보가 없어진다. 다시말해, 로우패스 필터의 연산에서, 화상 정보의 휘도값의 워드길이(語長; word length)는 대폭 반올림(丸; rounding)되어 메모리에 격납되고, 대부분의 화소 정보의 하위 비트는, 축소 후의 화상에 영향을 미치지 않는 것이다.
- [0396] 한편, 이 실시형태에서 이용하는 신규한 블록 매칭 수법에서는, 모든 화소의 휘도값의 전(全)비트 정보를, 골고

루 평등하게 사용해서 SAD값을 연산하고, 그 분산 가산값을 구해서 축소 SAD 테이블에 가산한다. 축소 SAD 테이블의 각 테이블 요소값의 워드길이만 늘리면, 최종적인 SAD값의 출력까지, 일체(一切; 모든)의 반올림 오차(丸誤差; rounding-process error)를 포함하지 않는 형태로 연산가능하다. 축소 SAD 테이블의 면적은 프레임 메모리와 비교해서 작기 때문에, 축소 SAD 테이블의 워드길이 확장은 큰 문제로 되지 않는다. 그 결과로서, 축소 SAD 테이블 및 움직임 벡터 검출을, 고정밀도로 실현할 수 있는 것이다.

[0397]

이상의 설명은, 주목 화면인 참조 프레임(102)에 대해서, 하나의 타겟 블록을 생각하고, 해당 타겟 블록을 중심으로 한 서치 범위를 설정하고, 그 서치 범위에서의 움직임 벡터를 검출하는 경우의 설명이지만, 실제의 손흔들림 보정 시스템에서는, 원프레임(101)을 복수의 영역으로 분할하고, 각각의 분할 영역에서, 각각 타겟 블록을 설정한다. 그리고, 참조 프레임(102)에서는, 각각의 타겟 블록의 사영 이미지를 중심으로 해서 서치 범위를 설정해서, 각각의 서치 범위에서 움직임 벡터(205)를 검출하도록 한다.

[0398]

그리고, CCD 이미지를 활상 소자로서 사용하는 손흔들림 보정 시스템에서는, 참조 프레임에서 취득한 그들 복수개의 타겟 블록에 대해서 얻어진 복수개의 움직임 벡터를 이용해서, 과거의 프레임에서의 그들 움직임 벡터로부터의 추이(推移)도 가미(加味)하면서 통계적으로 처리함으로써, 1프레임에 대해서 하나의 글로벌 벡터, 즉 프레임의 손흔들림 벡터를 확정하도록 한다.

[0399]

이것에 대해서, 이 실시형태에서는, 하나의 프레임에 대해서, 복수개의 타겟 블록 및 서치 범위를 설정하고, 그들 복수개의 타겟 블록에 대해서 얻어진 복수개의 움직임 벡터를 이용해서, 전술한 분할 화상 구간의 각각에 대한, 손흔들림 속도 벡터를 산출하도록 한다.

[0400]

즉, 이 실시형태에서는, 도 28에 도시하는 바와 같이, 1프레임의 화상 영역을, 전술한 분할 화상 구간에 대응해서, 수직 방향으로 8분할함과 동시에, 이 예에서는, 수평 방향으로도 8분할하고, 합계 64개의 분할 영역으로 한다.

[0401]

그리고, 도 28에 도시하는 바와 같이, 64개의 분할 영역의 각각에서 검출하고 싶은 64개의 움직임 벡터 205의 기준 위치 P01~P064의 각각을 중심으로 하는 서치 범위 SR1, SR2, …, SR64를 정하고, 각 서치 범위에서, 타겟 블록의 사영 이미지 블록 IB1, IB2, …, IB64를 상정한다. 이 경우, 타겟 블록의 사영 이미지 블록 IB1, IB2, …, IB64는, 서로 인접(隣接; adjacent)하는 것끼리 서로 겹치는(重合; overlap) 일은 없지만, 서치 범위 SR1, SR2, …, SR64는, 도28로부터 명확한 바와 같이, 서로 인접하는 것끼리에서는, 서로 영역적으로 겹치는 것으로 되어도 좋다.

[0402]

그리고, 이 사영 이미지 블록 IB1, IB2, …, IB64와 같은 크기의 참조 블록을 설정하고, 각 서치 범위 SR1, SR2, …, SR64 내를, 설정한 참조 블록을 이동시켜서, 상술한 바와 마찬가지로 해서, 축소 SAD 테이블을 생성하고, 각 서치 범위 SR1, SR2, …, SR64의 타겟 블록 TG1~TG64(타겟 블록 TG1~TG64의 도시는 생략)에 대한 합계 64개의 움직임 벡터 205V01~205V64를 검출하도록 한다.

[0403]

그리고, 이 실시형태에서는, 도 29에 도시하는 바와 같이, 전술한 분할 화상 영역 Pdiv_0~Pdiv_7의 각각에 대응하는 같은 수직 방향의 영역에 포함되는 8개의 타겟 블록에 대한 8개의 움직임 손흔들림 벡터를 평균화하는 것에 의해, 각각의 분할 화상 영역 Pdiv_0~Pdiv_7에 대한 손흔들림 움직임 벡터 VCdiv_0~VCdiv_7을 산출하도록 한다.

[0404]

즉, 검출된 움직임 벡터 205V1~205V8에 대해서, 평균화 처리를 하는 것에 의해, 분할 화상 구간 Pdiv_0에 대한 손흔들림 움직임 벡터 VCdiv_0을 산출하고, 검출된 움직임 벡터 205V9~205V16에 대해서, 평균화 처리를 하는 것에 의해, 분할 화상 구간 Pdiv_1에 대한 손흔들림 움직임 벡터 VCdiv_1을 산출하고, 검출된 움직임 벡터 205V17~205V24에 대해서, 평균화 처리를 하는 것에 의해, 분할 화상 구간 Pdiv_2에 대한 손흔들림 움직임 벡터 VCdiv_2를 산출하고, 검출된 움직임 벡터 205V25~205V32에 대해서, 평균화 처리를 하는 것에 의해, 분할 화상 구간 Pdiv_3에 대한 손흔들림 움직임 벡터 VCdiv_3을 산출하고, 검출된 움직임 벡터 205V33~205V40에 대해서, 평균화 처리를 하는 것에 의해, 분할 화상 구간 Pdiv_4에 대한 손흔들림 움직임 벡터 VCdiv_4를 산출하고, 검출된 움직임 벡터 205V41~205V48에 대해서, 평균화 처리를 하는 것에 의해, 분할 화상 구간 Pdiv_5에 대한 손흔들림 움직임 벡터 VCdiv_5를 산출하고, 검출된 움직임 벡터 205V49~205V56에 대해서, 평균화 처리를 하는 것에 의해, 분할 화상 구간 Pdiv_6에 대한 손흔들림 움직임 벡터 VCdiv_6을 산출하고, 검출된 움직임 벡터 205V57~205V64에 대해서, 평균화 처리를 하는 것에 의해, 분할 화상 구간 Pdiv_7에 대한 손흔들림 움직임 벡터 VCdiv_7을 산출한다.

[0405]

여기에서, 평균화 처리를 할 때에는, 각 분할 화상 구간에 포함되는 8개의 움직임 벡터 205Vi~205V(i+7)(단,

$i=1, 9, 17, 25, 33, 41, 49, 57$ 이다)에 대해서, 예외적인 움직임 벡터가 있으면, 해당 예외적인 움직임 벡터는 평균화 처리에서 제거하도록 한다.

[0406] 예외적인 움직임 벡터인지 여부의 판별은, 이 실시형태에서는, 각 움직임 벡터에 대해서, 그것과 수평 방향 및 수직 방향으로 인접하는 타겟 블록에 대해서 검출된 움직임 벡터와 비교해서, 그 비교 결과의 벡터차가, 미리 정한 임계값(threshold value)보다도 큰지 여부에 따라 행하고, 상기 비교 결과의 벡터차가, 미리 정한 임계값보다도 크다고 판별했을 때에는, 그 움직임 벡터는 평균화 처리에서 제거하는 움직임 벡터라고 판별하도록 한다.

[0407] 또한, 예외적인 움직임 벡터인지 여부의 판별은, 상술한 바와 같은 판별 방법에 한정되는 것은 아니다. 예를 들면, 하나의 분할 화상 구간의 움직임 벡터에 대한 예외적인 움직임 벡터의 판별시에, 해당 분할 화상 구간 및 그의 상하에 (수직 방향으로) 인접하는 분할 화상 구간의 모든 움직임 벡터의 평균값 및 분산을 구하고, 구한 평균값과 판별 대상인 움직임 벡터와의 차가, 구한 분산의 범위 내인지 여부에 따라 판별하도록 해도 좋다.

[0408] 그리고, 이 실시형태에서는, 분할 화상 구간 $Pdiv_0 \sim Pdiv_7$ 의 각각의 손흔들림 속도 벡터 $Vec_0 \sim Vec_7$ 은, 자(自) 분할 화상 구간의 손흔들림 움직임 벡터와 하나 뒤의 인접하는 분할 화상 구간의 손흔들림 움직임 벡터와의 차분(변화분)으로서 검출한다.

[0409] 즉,

[0410] $Vec_0 = VCdiv_1 - VCdiv_0$

[0411] $Vec_1 = VCdiv_2 - VCdiv_1$

[0412] $Vec_2 = VCdiv_3 - VCdiv_2$

[0413] $Vec_3 = VCdiv_4 - VCdiv_3$

[0414] $Vec_4 = VCdiv_5 - VCdiv_4$

[0415] $Vec_6 = VCdiv_7 - VCdiv_6$

[0416] $\dots (식 3)$

[0417]으로서 손흔들림 속도 벡터 $Vec_0 \sim Vec_7$ 을 산출한다.

[0418] 1프레임의 화상의 수직 방향의 최후의 분할 화상 구간 $Pdiv_7$ 뒤의 수직 방향의 분할 화상 구간은 존재하지 않으므로, 해당 분할 화상 구간 $Pdiv_7$ 에 대한 손흔들림 속도 벡터는, 상술한 바와 같은 차분 연산에서는 산출할 수가 없다.

[0419] 이 실시형태에서는, 이 분할 화상 구간 $Pdiv_7$ 에 대한 손흔들림 움직임 벡터는, 해당 분할 화상 구간 $Pdiv_7$ 보다도 수직 방향으로 앞(前)의 1개~복수의 분할 화상 구간의 손흔들림 속도 벡터로부터 예측한 것을 이용하도록 한다. 또한, 수직 방향으로 하나 앞의 분할 화상 구간의 손흔들림 속도 벡터를, 해당 분할 화상 구간 $Pdiv_7$ 의 손흔들림 속도 벡터로서 이용하도록 해도 좋다.

[0420] 또한, 구한 각 분할 화상 구간 $Pdiv_0 \sim Pdiv_7$ 에 대한 손흔들림 속도 벡터 $Vec_0 \sim Vec_7$ 의 역부호 성분이, 손흔들림 보정 및 포컬 플레이인 현상에 의한 화상 일그러짐을 보정에 이용하는 것은 전술한 대로이다.

[0421] 그리고, 이 실시형태에서는, 1프레임의 수직 방향의 선두의 분할 화상 구간 $Pdiv_0$ 에 대한 손흔들림 움직임 벡터 $VCdiv_0$ 은, 해당 1프레임의 손흔들림 벡터로 한다. 그리고, 해당 프레임의 화상 데이터의 화상 메모리로부터의 판독출력을, 앞(前)프레임의 화상 데이터에 대해서, 이 분할 화상 구간 $Pdiv_0$ 에 대한 손흔들림 움직임 벡터 $VCdiv_0$ 에 의거해서 산출된 변위량만큼 시프트한 화상 위치(대응하는 메모리 어드레스 위치)로부터 개시하도록 한다.

[0422] 또, 이 실시형태에서는, 각 분할 화상 구간 $Pdiv_0 \sim Pdiv_7$ 에 대한 손흔들림 속도 벡터 $Vec_0 \sim Vec_7$ 의 역부호 성분을 시간 적분하는 것에 의해 산출한 화상 일그러짐 변위량 분만큼, 화상 메모리로부터의 각 화상 데이터의 수평 방향 및 수직 방향의 판독출력 위치를 시프트하도록 하는 것에 의해, 손흔들림 및 포컬 플레이인 현상에 의거하는 화상 일그러짐을 캔슬하도록(제거하도록) 보정한다.

[0423] [본 발명에 따른 화상 신호의 일그러짐 보정 장치의 제1 실시형태]

[0424] 이상과 같은 특징을 구비하는 화상 신호의 일그러짐 보정 장치의 제1 실시형태로서, 촬상 장치의 경우를 예를

들어, 도면을 참조하면서 설명한다. 도 30은, 본 발명의 화상 신호의 일그러짐 보정 장치의 실시형태로서의 활상 장치의 1예의 블록도를 도시하는 것이다.

[0425] 도 30에 도시하는 바와 같이, 이 실시형태의 활상 장치는, 시스템 버스(2)에 CPU(Central Processing Unit)(1)가 접속됨과 동시에, 시스템 버스(2)에, 활상 신호 처리계(10)나, 유저 조작 입력부(3), 화상 메모리부(4), 기록 재생 장치부(5) 등이 접속되어 구성되어 있다. 또한, 이 명세서에서는, CPU(1)는, 여러 가지 소프트웨어 처리를 행하는 프로그램을 기억하는 ROM(Read Only Memory)나 워크에리어용 RAM 등을 포함하는 것으로 하고 있다.

[0426] 유저 조작 입력부(3)를 통한 활상 기록 개시 조작을 받아서, 도 30의 활상 장치는, 후술하는 바와 같은 활상 화상 데이터의 기록 처리를 행한다. 또, 유저 조작 입력부(3)를 통한 활상 기록 화상의 재생 개시 조작을 받아서, 도 30의 활상 장치는, 기록 재생 장치부(5)의 기록 매체에 기록된 활상 화상 데이터의 재생 처리를 행한다.

[0427] 그리고, 이 실시형태에서는, 유저 조작 입력부(3)를 통해서, 유저로부터의 화상 확대·축소 지시 입력이나, 해상도 지시 입력, 또 줌 배율의 값의 지시 입력이 제어부로서의 CPU(1)에 부여(與)된다. CPU(1)는, 그들 지시 입력에 따른 제어 신호를 형성하고, 타이밍 신호 발생부(12)나 각 처리부에 공급한다.

[0428] 도 30에 도시하는 바와 같이, 활상 렌즈(10L)를 구비하는 카메라 광학계(光學系)(도시는 생략)를 통한 피사체로부터의 입사광(入射光)은, 활상 소자(11)에 조사(照射)되어 활상된다. 이 예에서는, 활상 소자(11)는, CMOS(Complementary Metal Oxide Semiconductor) 이미저로 구성되어 있다.

[0429] 이 예의 활상 장치에서는, 활상 기록 개시 조작이 이루어지면, 활상 소자(11)로부터는, 타이밍 신호 발생부(12)로부터의 타이밍 신호에 의해 샘플링되는 것에 의해, 적(赤)(R), 녹(綠)(G), 청(青)(B)의 3원색으로 구성되는 베이어 배열(Bayer array)의 RAW(Read After Write) 신호인 아날로그 활상 신호가 출력된다.

[0430] 이 예에서는, 활상 소자(11)로부터는, 타이밍 신호 발생부(12)로부터의 타이밍 신호에 의해 1라인 단위가 동시에 취출되는(샘플링되는) 것에 의해 아날로그 활상 신호로 되어 출력된다.

[0431] 출력된 아날로그 활상 신호는, 전처리부(13)에 공급되고, 결합 보정이나 γ 보정 등의 전처리가 행해지고, 데이터 변환부(14)에 공급된다.

[0432] 데이터 변환부(14)는, 이것에 입력된 아날로그 활상 신호로부터, 휘도 신호 성분 Y와, 색차 신호 성분 Cb/Cr에 의해 구성되는 디지털 활상 신호(YC 데이터)로 변환하고, 그 디지털 활상 신호를 시스템 버스를 거쳐서, 화상 메모리부(4)에 공급한다.

[0433] 화상 메모리부(4)는, 이 도 30의 예에서는, 2개의 프레임 메모리(41, 42)로 이루어지고, 데이터 변환부(14)로부터의 디지털 활상 신호는, 우선, 프레임 메모리(41)에 격납된다. 그리고, 1프레임 경과하면, 프레임 메모리(41)에 기억되어 있는 디지털 활상 신호가, 프레임 메모리(42)에 전송(轉送; transfer)됨과 동시에, 프레임 메모리(41)에는, 데이터 변환부(14)로부터의 새로운 프레임의 디지털 활상 신호가 기입된다. 따라서, 프레임 메모리(42)에는, 프레임 메모리(41)에 격납되어 있는 프레임 화상보다도 1프레임분 앞의 프레임 화상이 격납되어 있다.

[0434] 그리고, 손흔들림 움직임 벡터 검출부(15)는, 시스템 버스(2)를 거쳐서, 이들 2개의 프레임 메모리(41) 및 프레임 메모리(42)를 액세스해서, 그 입력 데이터를 판독 출력하고, 전술한 바와 같은 분할 화상 구간마다의 움직임 벡터 검출 처리를 실행한다. 이 경우, 프레임 메모리(42)에 격납되어 있는 프레임 화상은, 원프레임의 화상으로 되고, 또 프레임 메모리(41)에 격납되어 있는 프레임 화상은, 참조 프레임의 화상으로 된다.

[0435] 그리고, 손흔들림 움직임 벡터 검출부(15)는, 그 검출 결과인 전술한 바와 같은 분할 화상 구간마다의 움직임 벡터를, 그 후단(後段)의 일그러짐 보정 해상도 변환부(16)에 제어 신호로서 전달(傳達)한다.

[0436] 일그러짐 보정 해상도 변환부(16)는, 손흔들림 움직임 벡터 검출부(15)로부터 수취한 분할 화상 구간마다의 움직임 벡터로부터 손흔들림 속도 벡터를 산출하고, 산출한 손흔들림 속도 벡터에 의거해서 화상 일그러짐 보정량을 산출한다. 그리고, 산출한 화상 일그러짐 보정량에 따라서, 프레임 메모리(42)에 격납되어 있는 지연 프레임의 화상 데이터를 절출(切出; cut out; 잘라냄)하면서, 필요한 해상도 및 화상 사이즈로 변환하는 처리를 한다. 이 프레임 메모리(42)로부터의 산출한 화상 일그러짐 보정량에 따른 절출에 의해, 변환후의 화상은, 손흔들림 및 포컬 플레인 현상에 의거하는 화상 일그러짐이 제거된 화상으로 된다.

- [0437] 이 일그러짐 보정 해상도 변환부(16)로부터의 손흔들림 및 포컬 플레인 현상에 의거하는 화상 일그러짐이 제거된 화상 데이터는, NTSC(National Television System Committee) 인코더(18)에 의해 NTSC 방식의 표준 컬러 영상 신호에 변환되고, 전자식 뷰파인더를 구성하는 모니터 디스플레이(6)에 공급되고, 촬영시의 화상이 그 표시 화면에 모니터 표시된다.
- [0438] 이 모니터 표시와 병행해서, 일그러짐 보정 해상도 변환부(16)로부터의 손흔들림이 제거된 화상 데이터는 코덱부(17)에서 기록 변조 등의 코딩 처리된 후, 기록 재생 장치부(5)에 공급되어, DVD(Digital Versatile Disc) 등의 광 디스크나 하드디스크 등의 기록 매체에 기록된다.
- [0439] 이 기록 재생 장치부(5)의 기록 매체에 기록된 활상 화상 데이터는, 유저 조작 입력부(3)를 통한 재생 개시 조작에 따라서 판독출력되고, 코덱부(17)에 공급되어, 재생 디코드된다. 그리고, 재생 디코드된 화상 데이터는 NTSC 인코더(18)를 통해서 모니터 디스플레이(6)에 공급되고, 재생 화상이 그 표시 화면에 표시된다. 또한, 도 30에서는, 도시를 생략했지만, NTSC 인코더(18)로부터의 출력 영상 신호는, 영상 출력 단자를 통해서 외부에 도출(導出)하는 것이 가능하게 되어 있다.
- [0440] 상술한 손흔들림 움직임 벡터 검출부(15)는, 하드웨어에 의해 구성할 수 있으며, 또 DSP(Digital Signal Processor)를 이용해서 구성할 수도 있다. 또, CPU(1)에 의해 소프트웨어 처리로 할 수도 있다. 마찬가지로, 일그러짐 보정 해상도 변환부(16)도, 하드웨어에 의해 구성할 수 있으며, 또 DSP(Digital Signal Processor)를 이용해서 구성할 수도 있다. 나아가서는, CPU(1)에 의해 소프트웨어 처리로 할 수도 있다.
- [0441] [손흔들림 움직임 벡터 검출부(15)에서의 처리 동작]
- [0442] <제1 예>
- [0443] 이 손흔들림 움직임 벡터 검출부(15)에서의 처리 동작의 제1 예의 흐름을, 도 31 및 도 32의 플로차트를 참조해서, 이하에 설명한다.
- [0444] 우선, 전술한 도 3 혹은 도 28에 도시한 바와 같은 서치 범위(105) 혹은 서치 범위 SR1~SR64의 각각 내의 하나의 참조 블록 Ii에 대응하는 참조 벡터 (vx, vy)를 지정한다(스텝 S101). 전술한 바와 같이, (vx, vy)는, 타겟 블록의 프레임 위의 위치(서치 범위의 중심 위치이다)를 기준 위치 (0, 0)으로 했을 때에, 지정된 참조 벡터에 의해 나타내어지는 위치를 나타내고, vx는 지정된 참조 벡터에 의한, 기준 위치로부터의 수평 방향의 어긋남량 성분이며, 또 vy는 지정된 참조 벡터에 의한, 기준 위치로부터의 수직 방향 성분의 어긋남량 성분이다. 그리고, 전술한 종래예에서 기술한 바와 마찬가지로, 어긋남량 vx, vy는, 화소를 단위로 한 값으로 되어 있다.
- [0445] 여기에서, 서치 범위의 중심 위치를 상기 기준 위치 (0, 0)로 하고, 서치 범위를, 수평 방향으로는 $\pm Rx$, 수직 방향으로는 $\pm Ry$ 로 했을 때,
- [0446] $-Rx \leq vx \leq +Rx, -Ry \leq vy \leq +Ry$
- [0447] 로 되는 것이다.
- [0448] 다음에, 타겟 블록 Io 내의 하나의 화소의 좌표 (x, y)를 지정한다(스텝 S102). 다음에, 타겟 블록 Io 내의 지정된 하나의 좌표 (x, y)의 화소값 $Io(x, y)$ 와, 참조 블록 Ii 내의 대응하는 화소 위치의 화소값 $Ii(x+vx, y+vy)$ 와의 차분 절대값 a 를, 전술하는 (식 1)에 나타낸 바와 같이 해서 산출한다(스텝 S103).
- [0449] 그리고, 산출한 차분 절대값 a 를, 해당 참조 블록 Ii의 참조 벡터 (vx, vy)가 가리키는 어드레스(테이블 요소)의, 그때까지의 SAD값에 가산하고, 그 가산인 SAD값을, 해당 어드레스에 다시쓰기하도록 한다(스텝 S104). 즉, 참조 벡터 (vx, vy)에 대응하는 SAD값을, $SAD(vx, vy)$ 로 나타냈을 때, 이것을, 전술한 (식 2), 즉,
- [0450] $SAD(vx, vy) = \sum a = \sum |Io(x, y) - Ii(x+vx, y+vy)| \dots (식 2)$
- [0451]로서 산출하고, 해당 참조 벡터 (vx, vy)가 가리키는 어드레스에 기입하도록 한다.
- [0452] 다음에, 타겟 블록 Io 내의 모든 좌표 (x, y)의 화소에 대해서, 상기의 스텝 S102~스텝 S104의 연산을 행했는지 여부를 판별하고(스텝 S105), 타겟 블록 Io 내의 모든 좌표 (x, y)의 화소에 대해서는, 아직 그 연산은 종료하고 있지 않다고 판별했을 때에는, 스텝 S102로 되돌아가고, 타겟 블록 Io 내의 다음의 좌표 (x, y)의 화소 위치를 지정하고, 이 스텝 S102 이후의 처리를 되풀이한다.
- [0453] 이상의 스텝 S101~스텝 S105까지의 처리는, 도 5에 도시한 플로차트의 스텝 S1~스텝 S5와 완전히 마찬가지이

다.

[0454] 이 실시형태에서는, 스텝 S105에서, 타겟 블록 Io 내의 모든 좌표 (x, y)의 화소에 대해서, 상기의 연산을 행했다고 판별했을 때에는, 축소 배율을 1/n로 해서, 참조 벡터 (vx, vy)를 1/n로 축소한 참조 축소 벡터 (vx/n, vy/n)를 산출한다(스텝 S106).

[0455] 그 다음에, 참조 축소 벡터 (vx/n, vy/n) 근방의 복수의 참조 벡터, 이 예에서는, 상술한 바와 같이 4개의 근방 참조 벡터를 검지(檢知)한다(스텝 S107). 그리고, 검지한 4개의 근방 참조 벡터의 각각에 대응하는 테이블 요소로서 분산 가산해야 할 값을, 전술한 바와 같이, 참조 축소 벡터와 근방 참조 벡터가 각각 나타내는 위치의 관계에 의거해서, 스텝 S104에서 구한 SAD값으로부터, 선형 가중 분산값으로서 구한다(스텝 S108). 그리고, 구한 4개의 선형 가중 분산값을, 근방 참조 벡터의 각각에 대응하는 SAD 테이블 요소값에 가산한다(스텝 S109).

[0456] 이 스텝 S109가 종료하면, 주목 중인 참조 블록에 대한 SAD값의 산출이 종료했다고 판별해서, 서치 범위 내의 모든 참조 블록, 즉 모든 참조 벡터 (vx, vy)에 대한 상기의 스텝 S101부터 스텝 S109까지의 연산 처리를 완료했는지 여부를 판별한다(도 32의 스텝 S111).

[0457] 스텝 S111에서, 아직 상기의 연산 처리를 완료하고 있지 않은 참조 벡터 (vx, vy)가 있다고 판별하면, 스텝 S101로 되돌아가고, 상기의 연산 처리를 완료하고 있지 않은 다음의 참조 벡터 (vx, vy)를 설정하고, 이 스텝 S101 이후의 처리를 되풀이한다.

[0458] 그리고, 스텝 S111에서, 상기의 연산 처리를 완료하고 있지 않은 참조 벡터 (vx, vy)는 서치 범위 내에 없게 되었다고 판별하면, 축소 SAD 테이블이 완성했다고 해서, 해당 완성한 축소 SAD 테이블에서, 최소값으로 되어 있는 SAD값을 검출한다(스텝 S112).

[0459] 다음에, 해당 최소값으로 되어 있는 테이블 요소 어드레스 (mx, my)의 SAD값(최소값)과, 그 근방의 복수개, 이 예에서는, 상술한 바와 같이 15개의 근방 테이블 요소의 SAD값을 이용해서 2차 곡면을 생성하고(스텝 S113), 그 2차 곡면의 최소값의 SAD값이 대응하는 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py)를 산출한다(스텝 S114). 이 최소값 벡터 (px, py)는, 소수 정밀도의 최소 테이블 요소 어드레스에 대응하고 있다.

[0460] 그리고, 산출한 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py)를 n배하는 것에 의해, 구해야 할 움직임 벡터 (px×n, py×n)를 산출한다(스텝 S115).

[0461] 이상에서, 하나의 타겟 블록에 대한, 이 실시형태에서의 블록 매칭에 의한 움직임 벡터의 검출 처리는, 종료된다. 도 20에 도시한 바와 같은, 1프레임에 대해서 분할한 영역에서, 복수개의 움직임 벡터를 검출하는 경우에는, 서치 범위 및 축소 배율 1/n을 재설정하고, 상술한 도 31 및 도 32에 도시한 처리를, 각 분할 영역의 타겟 블록에 대해서 되풀이하는 것이다.

[0462] 또한, 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py)를 산출하는 방법으로서는, 전술한 수평 방향 및 수직 방향의 3차 곡선을 이용하는 방법을 이용해도 좋은 것은 말할 필요도 없다(물론이다).

[0463] <제2 예>

[0464] 상술한 제1 예에서는, 하나의 참조 블록(참조 벡터)에 대해서, 그 SAD값을 구한 후, 그 SAD값으로부터, 참조 축소 벡터 근방의 복수 참조 벡터에 대한 분산 가산값을 구하고, 분산 가산 처리를 행하도록 했다.

[0465] 이것에 대해서, 이 제2 예에서는, 참조 블록 내의 각 화소의, 타겟 블록의 화소와의 차분을 검출했을 때에, 그 차분값으로부터, 참조 축소 벡터 근방의 복수 참조 벡터에 대한 분산 가산값(SAD값이 아닌 차분값)을 구하고, 구한 차분값을 분산 가산 처리하도록 한다. 이 제2 예에 따르면, 하나의 참조 블록 내의 모든 화소에 대한 차분 연산을 종료했을 때에는, 축소 SAD 테이블이 생성되게 된다.

[0466] 도 33 및 도 34는, 이 제2 예에 따른 움직임 벡터 검출 처리의 플로차트를 도시하는 것이다.

[0467] 도 33의 스텝 S121~스텝 S123까지의 처리는, 도 31의 스텝 S101~스텝 S103까지의 처리와 완전히 마찬가지이므로, 여기에서는, 그 상세한 설명은 생략한다.

[0468] 이 제2 예에서는, 스텝 S123에서, 좌표 (x, y)의 화소에 대한 참조 블록과 타겟 블록 사이에서의 차분값 a가 산출되면, 다음에는, 축소 배율을 1/n로 해서, 참조 벡터 (vx, vy)를 1/n로 축소한 참조 축소 벡터 (vx/n, vy/n)를 산출한다(스텝 S124).

[0469] 다음에, 참조 축소 벡터 (vx/n, vy/n) 근방의 복수의 참조 벡터, 이 예에서는, 상술한 바와 같이 4개의 근방 참

조 벡터를 검지한다(스텝 S125). 그리고, 검지한 4개의 근방 참조 벡터의 각각에 대응하는 테이블 요소로서 분산 가산해야 할 차분값을, 전술한 바와 같이, 스텝 S123에서 구한 차분값 a 로부터, 참조 축소 벡터와 근방 참조 벡터가 각각 나타내는 위치의 관계에 의거해서, 선형 가중 분산값(차분값)으로서 구한다(스텝 S126).

[0470] 그리고, 구한 4개의 선형 가중 분산값을, 근방 참조 벡터의 각각에 대응하는 테이블 요소값에 가산한다(스텝 S127).

[0471] 이 스텝 S127이 종료하면, 타겟 블록 Io 내의 모든 좌표 (x, y) 의 화소에 대해서, 상기의 스텝 S122~스텝 S127의 연산을 행했는지 여부를 판별하고(스텝 S128), 타겟 블록 Io 내의 모든 좌표 (x, y) 의 화소에 대해서는, 아직 그 연산은 종료하고 있지 않다고 판별했을 때에는, 스텝 S122로 되돌아가고, 타겟 블록 Io 내의 다음의 좌표 (x, y) 의 화소 위치를 지정하고, 이 스텝 S122 이후의 처리를 되풀이한다.

[0472] 스텝 S128에서, 타겟 블록 Io 내의 모든 좌표 (x, y) 의 화소에 대해서, 상기의 연산을 행했다고 판별했을 때에는, 주목 중인 참조 블록에 대한 SAD값의 산출이 종료했다고 판별해서, 서치 범위 내의 모든 참조 블록, 즉 모든 참조 벡터 (vx, vy) 에 대한 상기의 스텝 S121부터 스텝 S128까지의 연산 처리를 완료했는지 여부를 판별한다(도 34의 스텝 S131).

[0473] 스텝 S131에서, 아직 상기의 연산 처리를 완료하고 있지 않은 참조 벡터 (vx, vy) 가 있다고 판별하면, 스텝 S121로 되돌아가고, 상기의 연산 처리를 완료하고 있지 않은 다음의 참조 벡터 (vx, vy) 를 설정하고, 이 스텝 S121 이후의 처리를 되풀이한다.

[0474] 그리고, 스텝 S121에서, 상기의 연산 처리를 완료하고 있지 않은 참조 벡터 (vx, vy) 는 서치 범위 내에 없게 되었다고 판별하면, 축소 SAD 테이블이 완성했다고 해서, 해당 완성한 축소 SAD 테이블에서, 최소값으로 되어 있는 SAD값을 검출한다(스텝 S132).

[0475] 다음에, 해당 최소값으로 되어 있는 테이블 요소 어드레스 (mx, my) 의 SAD값(최소값)과, 그 근방의 복수개, 이 예에서는, 전술한 바와 같이 15개의 근방 테이블 요소의 SAD값을 이용해서 2차 곡면을 생성하고(스텝 S133), 그 2차 곡면의 최소값의 SAD값이 대응하는 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py) 를 산출한다(스텝 S134). 이 최소값 벡터 (px, py) 는, 소수 정밀도의 최소 테이블 요소 어드레스에 대응하고 있다.

[0476] 그리고, 산출한 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py) 를 n 배하는 것에 의해, 구해야 할 움직임 벡터 $(px \times n, py \times n)$ 를 산출한다(스텝 S135).

[0477] 이상에서, 하나의 타겟 블록에 대한, 이 제2 예에서의 블록 매칭에 의한 움직임 벡터의 검출 처리는, 종료로 된다. 도 28에 도시한 바와 같이, 1프레임에 대해서 분할한 영역에서, 복수개의 움직임 벡터를 검출하는 경우에는, 서치 범위 및 축소 배율 $1/n$ 을 재설정하고, 전술한 도 33 및 도 34에 도시한 처리를, 각 분할 영역에 대해서 되풀이하는 것이다.

[0478] 또한, 이 제2 예에서도, 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py) 를 산출하는 방법으로서는, 전술한 수평 방향 및 수직 방향의 3차 곡선을 이용하는 방법을 이용해도 좋은 것은 말할 필요도 없다.

[0479] <제3 예>

[0480] 도 24에 도시한 바와 같이, 이 실시형태에 따른 움직임 벡터의 검출 수법을 이용한 경우에는, 참조 벡터의 축소 배율이 $1/64$ 인 경우에서도, 전혀 다른 움직임 벡터를 출력하는 바와 같은 패턴은 보이지 않기 때문에, 실질적으로 $1/4096$ 으로, SAD 테이블을 삭감가능하다.

[0481] 다시말해, $1/4096$ 로 삭감한 축소 SAD 테이블을 준비해 두고, 예를 들면 1회째의 축소 배율 $1/na=1/64$ 로 1회째의 움직임 벡터를 검출한다. 다음에, 1회째에서 검출한 그 움직임 벡터를 중심으로 해서 서치 범위를 좁히고, 2회째의 검출을, 1회째의 축소 배율 $1/na$ 보다도 작은 2회째의 축소 배율 $1/nb$, 예를 들면 $1/nb=1/8$ 로 행하도록 하면 좋다. 즉, 1회째와 2회째에서 축소 배율을 바꾸어, 1회째의 벡터 오차 범위 내로 들어가도록, 2회째의 축소 배율을 설정하면, 상당한 고정밀도로, 움직임 벡터 검출이 가능하다.

[0482] 이 제3 예의 경우에서의 움직임 벡터 검출 처리를, 도 35~도 38의 플로차트를 참조하면서 설명한다.

[0483] 이 도 35~도 38에 도시하는 제3 예는, 기본적인 움직임 검출 처리로서 전술한 제1 예를 이용하고 있다. 따라서, 도 35의 스텝 S141~스텝 S149의 처리 스텝 및 도 36의 스텝 S151~스텝 S155까지의 처리 스텝은, 도 31의 스텝 S101~스텝 S109의 처리 스텝 및 도 32의 스텝 S111~스텝 S115까지의 처리 스텝과 완전히 마찬가지이다.

- [0484] 이 제3 예에서는, 도 36의 스텝 S155에서 움직임 벡터를 산출했으면, 그곳에서 처리를 종료하는 것이 아니라, 해당 스텝 S155에서 산출한 움직임 벡터는, 1회째의 움직임 벡터로서, 다음의 스텝 S156에서, 이 1회째에서 산출한 움직임 벡터에 의거해서, 같은 참조 프레임 내에서, 서치 범위를 줄이고, 또 참조 벡터의 축소 배율을, 1회째의 축소 배율 1/na보다도 작은 축소 배율 1/nb로 변경한다.
- [0485] 즉, 1회째의 처리에서, 움직임 벡터가 산출되면, 그 산출된 움직임 벡터로부터, 참조 프레임과의 사이에서, 상관이 있는 블록 범위를 대량 검출할 수가 있다. 그래서, 그 상관이 있는 블록 범위를 중심으로 한, 줄인 서치 범위(narrower search range)를 설정할 수가 있다. 그리고, 1회째보다도 축소 배율을 작게 함으로써, 보다 오차가 적은 상태에서, 2회째의 움직임 벡터의 산출이 가능하게 된다고 기대할 수 있다.
- [0486] 이렇게 해서, 스텝 S156에서, 서치 범위를 설정하고, 새로운 축소 배율을 설정했으면, 1회째와 완전히 마찬가지로 해서, 2회째의 움직임 벡터의 검출 처리를, 스텝 S157~스텝 S158, 도 37의 스텝 S161~스텝 S168, 또 도 38의 스텝 S171~스텝 S174에 의해 실행한다. 이들 스텝의 처리는, 도 31의 스텝 S101~스텝 S109의 처리 스텝 및 도 32의 스텝 S111~스텝 S115까지의 처리 스텝과 완전히 마찬가지이다.
- [0487] 이렇게 해서, 최종적으로, 스텝 S174에서, 2회째의 움직임 벡터로서, 목적으로 하는 움직임 벡터가 얻어진다.
- [0488] 이상의 예는, 움직임 벡터의 검출 방법으로서, 전술한 제1 예를 이용하고, 그것을 2단계 되풀이한 경우이지만, 서치 범위를 더욱 줄이고, 또한, 필요에 따라서 축소 배율을 변경하면서, 2단계 이상 되풀이하도록 해도, 물론(勿論) 좋다.
- [0489] 또, 움직임 벡터의 검출 방법으로서는, 전술한 제1 예 대신에, 전술한 제2 예를 이용할 수 있는 것은 말할 필요도 없다. 또, 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py)를 산출하는 방법으로서는, 전술한 수평 방향 및 수직 방향의 3차 곡선을 이용하는 방법을 이용해도 좋은 것은 전술한 예와 마찬가지이다.
- [0490] [일그리짐 보정 해상도 변환부(16)의 구성예 및 처리 동작]
- [0491] 도 39는, 이 실시형태에서의 일그리짐 보정 해상도 변환부(16)의 구성예를 도시하는 블록도이다.
- [0492] 이 실시형태에서의 일그리짐 보정 해상도 변환부(16)는, 신호 처리부(50)와, 이 신호 처리부(50)에 접속되는, 화상 메모리(60)와, 초기 좌표 산출부(70)와, 속도 벡터 처리부(80)로 이루어진다.
- [0493] 신호 처리부(50)는, 화상 메모리부(4)의 프레임 메모리(42)로부터의 활상 화상 데이터 Din에 대해서, 이 신호 처리부(50)에 접속되어 있는 화상 메모리(60)에의 기입/판독출력을 제어하면서, 손흔들림 보정 등의 보정 처리나, 활상 화상의 전자 줌 처리(전자적 확대·축소 처리)를 행함과 동시에, 지정된 해상도의 출력 화상 데이터 Dout을 생성해서 출력한다.
- [0494] 신호 처리부(50)에는, 예를 들면 타이밍 신호 발생부(12)로부터, 출력 화상 데이터의 수평 주기와 동기하는 신호, 즉 수평 동기 신호 H-SYNC와, 출력 화상 데이터의 수직 주기와 동기 하는 신호, 즉 수직 동기 신호 V-SYNC와, 1화소 주기의 클록 신호 CLK와, 그들과 동기하는 클럭이, 처리 타이밍 신호 St로서 공급된다(도 30에서, 이들 신호의 도시는 생략).
- [0495] 속도 벡터 처리부(80)에는, 상술한 손흔들림 움직임 벡터 검출부(15)로부터의, 이 예에서는 64개의 움직임 벡터 205V1~205V64가 공급된다. 이 속도 벡터 처리부(80)는, 하드웨어에 의해 구성할 수도 있고, CPU(1)에 의한 소프트웨어 처리에 의해 구성하는 것도 가능하다.
- [0496] 속도 벡터 처리부(80)는, 도 39에 도시하는 바와 같이, 손흔들림 움직임 벡터 검출부(15)로부터의 움직임 벡터 205V1~205V64로부터, 8개의 분할 화상 구간 Pdiv_0~Pdiv_7의 각각에 대한 손흔들림 속도 벡터 Vec_0~Vec_7을 생성하는 속도 벡터 생성부(81)와, 이 속도 벡터 생성부(81)에서 생성된 손흔들림 속도 벡터 Vec_0~Vec_7이 공급되는 수평 보정 속도 성분 검출부(82) 및 수직 보정 속도 성분 검출부(83)로 이루어진다.
- [0497] 도 40에, 속도 벡터 생성부(81)의 하드웨어 구성예를 도시한다. 도 40에 도시하는 바와 같이, 속도 벡터 생성부(81)는, 8개의 분할 화상 구간 Pdiv_0~Pdiv_7의 각각에 대응해서 설치되어 있는 평균화 연산부(810, 811, 812, 813, 814, 815, 816, 817)와, 차분 연산을 행하는 감산부(821, 822, 823, 824, 825, 826, 827)와, 손흔들림 벡터 출력부(830)와, 속도 벡터 출력부(840)으로 이루어진다.
- [0498] 평균화 연산부(810~817)의 각각은, 손흔들림 벡터 검출부(15)로부터의, 각각이 대응하는 분할 화상 구간의 8개의 움직임 벡터 205Vi~205V(i+7)을 취입(取入; retrieve, store)하고, 전술한 예외 제거 처리를 포함하는 평균

화 처리를 행해서, 각 분할 화상 구간 Pdiv_0~Pdiv_7에 대한 손흔들림 움직임 벡터 VCdiv_0~VCdiv_7을 산출해서 출력한다.

[0499] 그리고, 감산부(821~827)는, 평균화 연산부(810~817)로부터의 손흔들림 움직임 벡터 VCdiv_0~VCdiv_7를 받아서, 전술한 (식 3)에 나타낸 차분 연산을 행하고, 분할 화상 구간 Pdiv_0~Pdiv_6에 대한 손흔들림 속도 벡터 Vec_0~Vec_6을 산출하고, 속도 벡터 출력부(840)에 공급한다.

[0500] 속도 벡터 출력부(840)는, 전술한 바와 같이 해서, 수직 방향의 최하부의 분할 화상 구간 Pdiv_7에 대한 손흔들림 속도 벡터 Vec_7을, 이 예에서는, 산출된 다른 손흔들림 속도 벡터로부터 예측해서 구하고, 감산부(821~827)로부터의 손흔들림 속도 벡터 Vec_0~Vec_6과 함께, 수평 보정 속도 성분 검출부(82) 및 수직 보정 속도 성분 검출부(83)에 출력한다.

[0501] 도 41은, 이 속도 벡터 생성부(81)를 소프트웨어로 구성한 경우의 처리의 플로차트를 도시하는 것이다. 또한, 이하의 도 41의 플로차트의 설명에서는, 움직임 벡터 205Vi~205(i+7)은, 벡터 V(x, y)(x=1, 2, ..., 8; y=1, 2, ..., 8)로 나타낸다.

[0502] 우선, 손흔들림 벡터 검출부(15)로부터의 움직임 벡터 V(x, y)를 취입한다(스텝 S181). 다음에, 취입한 움직임 벡터(x, y)를, 해당 취입한 움직임 벡터 (x, y)와 같은 분할 화상 구간에 포함되는 다른 벡터 V(0~7, y)와 비교한다(스텝 S182). 다음에, 취입한 움직임 벡터 (x, y)를, 취입한 움직임 벡터 (x, y)에 대해서 수직 방향으로 인접하는 분할 화상 구간의 다른 벡터 V(0~7, y±1)과 비교한다(스텝 S183).

[0503] 다음에, 스텝 S182 및 스텝 S183의 비교 결과의 벡터차가, 모두 미리 정해진 임계값 이하로서, 취입한 움직임 벡터가 예외적인 움직임 벡터가 아니라고 확인되었는지 여부를 판별한다(스텝 S184).

[0504] 스텝 S184에서, 취입한 움직임 벡터가 예외적인 움직임 벡터라고 판별했을 때에는, 취입한 움직임 벡터는, 평균화 처리에서 제외한다(스텝 S185). 그리고, 스텝 S181로 되돌아간다.

[0505] 또, 스텝 S184에서, 취입한 움직임 벡터가 예외적인 움직임 벡터가 아니라고 판별했을 때에는, 취입한 움직임 벡터를, 평균화 처리에 반영시킬 것을 결정한다(스텝 S186). 그리고, 1화면 내의 모든 움직임 벡터에 대해서, 상기의 스텝 S181~186까지의 처리가 종료했는지 여부를 판별한다(스텝 S187).

[0506] 스텝 S187에서, 1화면 내의 모든 움직임 벡터에 대해서, 상기의 스텝 S181~186까지의 처리가 종료하고는 있지 않다고 판별했을 때에는, 스텝 S181로 되돌아간다.

[0507] 또, 스텝 S187에서, 1화면 내의 모든 움직임 벡터에 대해서, 상기의 스텝 S181~186까지의 처리가 종료했다고 판별했을 때에는, 평균화 처리를 실행한다(스텝 S188).

[0508] 다음에, 평균화 처리해서 얻어진 각 분할 화상 구간마다의 움직임 벡터의 차분 연산을 행해서, 손흔들림 속도 벡터를 산출한다. 다음에, 모든 분할 화상 구간에 대한 평균화 처리를 종료했는지 여부를 판별한다(스텝 S190).

[0509] 스텝 S190에서, 모든 분할 화상 구간에 대한 평균화 처리를 종료하고 있지 않다고 판별했을 때에는, 스텝 S188로 되돌아가고, 이 스텝 S188 이후의 처리를 실행한다. 또, 스텝 S190에서, 모든 분할 화상 구간에 대한 평균화 처리를 종료했다고 판별했을 때에는, 수직 방향의 최후의 분할 화상 구간 Pdiv_7에 대한 움직임 속도 벡터를 전술한 바와 같이 해서 예측해서 산출하고(스텝 S191), 이 처리 루틴을 종료한다.

[0510] 속도 벡터 출력부(840)는, 이상과 같이 해서 생성한 각 분할 화상 구간 Pdiv_0~Pdiv_7에 대한 움직임 속도 벡터 VCdiv_0~VCdiv_7을, 전술한 바와 같이, 수평 보정 속도 성분 검출부(82) 및 수직 보정 속도 성분 검출부(83)에 출력한다.

[0511] 수평 보정 속도 성분 검출부(82)에서는, 분할 화상 구간 Pdiv_0~Pdiv_7의 각각에서의 손흔들림 속도 벡터 Vec_0~Vec_7의 수평 방향 성분을 추출하고, 그 역부호 성분으로서, 수평 보정 속도 성분 X_STB_0~X_STB_7을 구한다. 그리고, 그 산출한 수평 보정 속도 성분 X_STB_0~X_STB_7을 신호 처리부(50)에 공급한다.

[0512] 또, 수직 보정 속도 성분 검출부(83)에서는, 분할 화상 구간 Pdiv_0~Pdiv_7의 각각에서의 손흔들림 속도 벡터 Vec_0~Vec_7의 수직 방향 성분을 추출하고, 그 역부호 성분으로서, 수직 보정 속도 성분 Y_STB_0~Y_STB_7을 구한다. 그리고, 그 산출한 수직 보정 속도 성분 Y_STB_0~Y_STB_7을 신호 처리부(50)에 공급한다.

[0513] 또, 분할 화상 구간 Pdiv_0에 대한 평균화 연산부(810)에서 산출된 손흔들림 움직임 벡터 VCdiv_0은, 손흔들림

벡터 출력부(830)를 통해서 그대로 출력되어, 초기 좌표 산출부(70)에 공급된다.

[0514] 초기 좌표 산출부(70)는, 처리 대상인 프레임의 화상 데이터의 화상 메모리로부터의 판독출력 위치의 초기 좌표 (SX, SY)를, 손흔들림 움직임 벡터 VCdiv_0으로부터 산출해서, 신호 처리부(50)에 공급한다.

[0515] 다음에, 신호 처리부(50)에서는, 화상 메모리(60)에의 화상 데이터의 기입/판독출력을 제어하면서, 나중에 상세하게 기술(詳述)하는 바와 같이 해서, 속도 벡터 처리부(80)로부터의 수평 보정 속도 성분과 수직 보정 속도 성분으로부터, 각 수평 라인의 수평 보정 변위량 SX_ADD 및 수직 보정 변위량 SY_ADD를 산출하고, 이를 산출한 수평 보정 변위량 SX_ADD 및 수직 보정 변위량 SY_ADD를 이용해서, 화상 메모리부(4)로부터의 활상 화상 데이터 Din에 대한 손흔들림에 의한 화상 일그러짐을 보정 처리하고, 출력 화상 데이터 Dout를 얻는다.

[0516] 또한, 전술한 바와 같이, 신호 처리부(50)에서는, 전자 줌(화상 확대·축소) 처리를 행함과 동시에, 표준 정세도(標準精細度; standard fineness), 고정세도(高精細度; high fineness) 등에 따른 해상도 변환 처리 등도 행한다.

[0517] 즉, 손흔들림 보정 처리나 전자 줌 처리, 출력 데이터 생성 처리를 위해서, 신호 처리부(50)는, 수평 처리 블록(51)과, 수직 처리 블록(52)과, 레지스터 블록(53)을 포함하고 있다. 그리고, 수평 처리 블록(51)은, 수평 손흔들림 보정량 적분부(511)와 수평 화상 처리부(512)를 구비하고, 또 수직 처리 블록(52)은, 수직 손흔들림 보정량 적분부(521)와 수직 화상 처리부(522)를 구비한다.

[0518] 수평 처리 블록(51)은, 입력되는 활상 화상 데이터 Din에 대해서 수평 방향의 처리를 행한다. 이 수평 방향의 처리에는, 수평 방향의 손흔들림 보정 처리를 포함한다. 수직 처리 블록(52)은, 입력되는 활상 화상 데이터 Din에 대해서 수직 방향의 처리를 행한다. 이 수직 방향의 처리에는, 수직 방향의 손흔들림 보정 처리를 포함한다. 또, 레지스터 블록(53)은, 속도 벡터 처리부(80)에서 검출된 손흔들림 속도 벡터의 수평 방향 및 수직 방향의 정보를, 수평 처리 블록(51) 및 수직 처리 블록(52)과 주고받는(受渡; delivery, exchange; 교환하는) 처리를 행한다.

[0519] 이 경우에, 이 실시형태에서는, 수평 보정 속도 성분 X_STB_0~X_STB_7 및 수직 보정 속도 성분 Y_STB_0~Y_STB_7의 각각의 값은, 1수평 라인 구간의 시간길이(時間長) 당의 손흔들림 보정량으로서 구해진다.

[0520] 즉, 수평 보정 속도 성분 X_STB(X_STB_0~X_STB_7)은,

[0521] $X_{STB} = (\text{수평 손흔들림 보정량} / \text{1수평 라인 구간의 시간길이})$

[0522] 로 된다.

[0523] 또, 수직 보정 속도 성분 Y_STB(Y_STB_0~Y_STB_7)

[0524] $Y_{STB} = (\text{수평 손흔들림 보정량} / \text{1수평 라인 구간의 시간길이})$

[0525] 로 된다.

[0526] 그리고, 수평 보정 속도 성분 X_STB_0~X_STB_7은, 수평 방향의 화소 피치(화소의 간격) dx의 배수값(倍數值) (소수점 이하의 값을 취할 수 있다. 이하 같다), 바꾸어 말하면, 수평 방향의 화소수(소수점 이하를 취할 수 있다)로 나타내어진다. 또, 수직 보정 속도 성분 Y_STB_0~Y_STB_7은, 수직 방향의 화소 피치(화소의 간격) dy의 배수값(소수점 이하를 취할 수 있다), 바꾸어 말하면, 수직 방향의 화소수(소수점 이하를 취할 수 있다. 이하 같다)로 나타내어진다.

[0527] 이것은, 속도 벡터 생성부(81)로부터의 손흔들림 속도 출력의 수평 방향 성분과, 수직 방향 성분의 각각 대해서, 「속도 성분의 값 → 화소수(소수점 이하의 값을 취할 수 있다. 이하 같다)의 값」의 대응 테이블을 준비하고, 속도 성분의 값을 입력값으로 해서, 화소수의 값을 얻도록 함으로써 실현된다. 그리고, 얻어진 화소수의 값에 대해서, 손흔들림의 속도 성분의 방향과는 역방향에 대응하는 부호를 부가해서, 각각 수평 보정 속도 성분 X_STB_0~X_STB_7 및 수직 보정 속도 성분 Y_STB_0~Y_STB_7로 하는 것이다.

[0528] 이 실시형태에서는, 이와 같이 수평 보정 속도 성분 X_STB_0~X_STB_7 및 수직 보정 속도 성분 Y_STB_0~Y_STB_7의 값을 정하는 것에 의해, 후술하는 라인 단위의 시간 적분 계산에 의한 수평 손흔들림 보정량 SX_ADD 및 수직 손흔들림 보정량 SY_ADD를, 수평 보정 속도 성분 및 수직 보정 속도 성분의 단순 가산에 의해 행할 수 있도록 하고 있다.

[0529] 그리고, 이들 수평 보정 속도 성분 검출부(82) 및 수직 보정 속도 성분 검출부(83)에서 산출된 수평 보정 속도

성분 X_STB_0~X_STB_7 및 수직 보정 속도 성분 Y_STB_0~Y_STB_7은, 예를 들면 타이밍 신호 발생부(12)로부터의 타이밍 신호 St에 따른 타이밍에서, 신호 처리부(50)의 레지스터 블록(53)의 IF(인터페이스) 레지스터(도시는 생략)에, 순차 취입된다.

[0530] 그리고, 취입된 수평 보정 속도 성분 X_STB_0~X_STB_7 및 수직 보정 속도 성분 Y_STB_0~Y_STB_7은, 수평 처리 블록(51) 및 수직 처리 블록(52)에서의 처리 타이밍에 대응해서, 레지스터 블록(53) 내의 수평 처리 블록용 레지스터 및 수직 처리 블록용 레지스터(어느것도 도시는 생략)에, 상기 IF 레지스터로부터, 상기 취입된 타이밍과는 다른 타이밍에서, 전송된다.

[0531] 수평 처리 블록(51) 및 수직 처리 블록(52)의 수평 손흔들림 보정량 적분부(511) 및 수직 손흔들림 보정량 적분부(521)는, 레지스터 블록(53)의 수평 처리 블록용 레지스터 및 수직 처리 블록용 레지스터에 취입되어 있는 각 분할 화상 구간 Pdiv_0~Pdiv_7의 수평 보정 속도 성분 X_STB_0~X_STB_7 및 수직 보정 속도 성분 Y_STB_0~Y_STB_7을 적분하는 것에 의해, 각 분할 화상 구간 Pdiv_0~Pdiv_7에서의 각 수평 라인에 대한 수평 손흔들림 보정량 SX_ADD 및 수직 손흔들림 보정량 SY_ADD를 산출한다.

[0532] 수평 처리 블록(51)의 수평 화상 처리부(512)에서는, 수평 손흔들림 보정량 적분부(511)에서 산출된 수평 손흔들림 보정량 SX_ADD를 이용해서, 수평 방향의 손흔들림 성분을 보정하고, 또 수직 처리 블록(52)의 수직 화상 처리부(522)에서는, 수직 손흔들림 보정량 적분부(521)에서 산출된 수직 손흔들림 보정량 SY_ADD를 이용해서, 수직 방향의 손흔들림 성분을 보정한다. 이 보정 처리에서는, 수평 처리 블록(51)의 수평 화상 처리부(512) 및 수직 처리 블록(52)의 수직 화상 처리부(522)에서는, 각각 화상 메모리(60)에 대해서 화상 데이터의 읽기 쓰기(讀書; read out and write; 판독출력 및 기입)를 하면서 보정 처리를 실행하도록 한다.

[0533] 이 실시형태에서는, 화상 메모리(60)는, 수평 처리용 FIFO 라인메모리(61)와, 수직 처리용 소규모 라인메모리(62)로 이루어진다. 수평 처리용 FIFO 라인메모리(61)는, 이 예에서는, 1라인 분의 용량을 구비한다. 또, 수직 처리용 소규모 라인메모리(62)는, 후술하는 수직 방향의 보간 처리용 FIR(Finite Impulse Response) 필터의 템수 이상의 라인수 분의 용량을 가진다.

[0534] 수평 처리 블록(51)의 수평 화상 처리부(512)에는, 수평 처리용 FIFO 라인메모리(61)에의 화상 데이터의 기입, 판독출력을 제어하는 FIFO 메모리컨트롤러(도 39에서는 도시를 생략)가 설치되어 있음과 동시에, 산출한 수평 손흔들림 보정량 SX_ADD가 소수점 이하(보정 후의 화소 위치가, 수평 방향의 화소 위치로부터 오프셋된 위치로 된다)인 경우를 상정하고, 보간 처리를 행하는 수평 방향 보간 처리부(도 39에서는 도시를 생략)가 설치되어 있다. 이 수평 방향 보간 처리부는, 후술하는 바와 같이, 이 실시형태에서는, 수평방향의 디지털 FIR 필터(이하, 수평 FIR 필터라고 한다)가 이용된다.

[0535] 또, 수직 처리 블록(52)의 수직 화상 처리부(522)에는, 수직 처리용 소규모 라인메모리(62)에의 화상 데이터의 기입, 판독출력을 제어하는 메모리컨트롤러(도 39에서는 도시를 생략)가 설치되어 있음과 동시에, 산출한 수직 손흔들림 보정량 SY_ADD가 소수점 이하(보정 후의 화소 위치가, 수직 방향의 화소 위치로부터 오프셋된 위치로 된다)인 경우를 상정하고, 보간 처리를 행하는 수직 방향 보간 처리부(도 39에서는 도시를 생략)가 설치되어 있다. 이 수직 방향 보간 처리부는, 후술하는 바와 같이, 이 실시형태에서는, 수직 방향의 디지털 FIR 필터(이하, 수직 FIR 필터라고 한다)가 이용된다.

[0536] 여기에서, 산출한 수평 손흔들림 보정량 SX_ADD 및 수직 손흔들림 보정량 SY_ADD이 소수점 이하의 값을 가지는 경우의 보간 처리에 대해서 설명한다.

[0537] 예를 들면, 손흔들림의 수평 방향의 속도 성분에 의해, 수평 방향으로 도 42에 도시하는 바와 같은 화상 일그레짐이 생긴 경우를 상정한다. 즉, 도 42의 예는, 손흔들림이 없으면 파선(破線; dashed lines)으로 나타내는 직선 위에 있어야 할 화소 G11, G21, G31…, G12, G22, G32…, G13, G23, G33…, 이, 손흔들림의 수평 방향의 속도 성분으로 인해 실선으로 나타내는바와 같은 기울어진 위치(傾位置; inclined position)로 되어 있는 경우이다.

[0538] 이 손흔들림을 보정하기 위해서는, 어긋나 있는 화소 위치를 본래의 위치에 있도록 어긋나게 하면 좋고, 그 어긋남량이, 도 42에 도시하는 바와 같이, 전술한 수평 손흔들림 보정량 SX_ADD에 상당한다. 이 수평 손흔들림 보정량 SX_ADD가, 화소 피치 dx의 정수배(整數倍; multiple)이면, 화소 데이터를 해당 수평 손흔들림 보정량 SX_ADD에 따른 화소 피치분만큼 어긋나게 해서 판독출력하도록 함으로써 보정할 수가 있다.

[0539] 그렇지만, 수평 손흔들림 보정량 SX_ADD가, 화소 피치 dx의 정수배가 아니라, 소수점 이하의 어긋남량을 포함하는 경우에는, 그 어긋남량에 대응하는 위치는, 화소 데이터가 존재하는 위치는 아니므로, 그 위치 근방의 복수

개의 화소 데이터를 이용한 보간 처리에 의해, 대응하는 화소 위치의 화소 데이터를 생성할 필요가 있다.

[0540] 도 43은, 그 보간 방법의 1예를 도시하는 것으로, 구하는 화소 데이터가, 화소 G1과 화소 G2와의 사이의 보간 화소 Gs인 경우에, 이 보간 화소 Gs의 위치와 화소 G1, G2의 위치와의 거리 k1, k2에 따른 비율로, 화소 G1의 데이터와 화소 G2의 데이터를 가산하는 것에 의해, 보간 화소 Gs의 화소 데이터를 생성하는 것이다. 이 경우, $k1:k2=W:(1-W)$ 로 했을 때,

[0541] 보간 화소 $Gs=G1 \times W + G2 \times (1-W)$

[0542] 인 보간 연산에 의해, 보간 화소 Gs의 화소 데이터를 얻는다.

[0543] 또, 도 43의 예와 같은 2개의 화소 데이터를 이용하는 것이 아니라, 도 44의 예와 같이 2개 이상(도 44의 예에서는 4개)의 근방 화소 데이터를 이용해서, 보간 연산을 해서 보간 화소를 구할 수도 있다. 이 경우, 보간 화소 Gs의 위치에 대한 화소 G0, G1, G2, G3의 위치와의 거리 k0, k1, k2, k3에 따른 비율로 되는 승산 계수 W0, W1, W2, W3을 정해서, 화소 G0, G1, G2, G3의 데이터를 가산하는 것에 의해, 보간 화소 Gs의 화소 데이터를 얻는다.

[0544] 즉,

[0545] 보간 화소 $Gs=G0 \times W0 + G1 \times W1 + G2 \times W2 + G3 \times W3$

[0546] 인 보간 연산에 의해, 보간 화소 Gs의 화소 데이터를 얻도록 한다.

[0547] 이 도 44의 예의 경우, 승산 계수 W0, W1, W2, W3의 값의 조(組; set)는, 화소 G1 또는 화소 G2와 보간 화소 Gs와의 거리 ds에 대응하는 것으로서 관련지어진 테이블을 준비해 두고, 거리 ds를 검색자(檢索子; search key)로 해서, 상기 승산 계수 W0, W1, W2, W3의 조를 테이블로부터 판독출력하도록 함으로써 얻을 수가 있다. 또한, 거리 ds는, 수평 손흔들림 보정량 SX_ADD의 소수부에 상당하는 것이다.

[0548] 이 보간 처리는, 이 실시형태에서는, 수평 FIR 필터를 이용해서 행하도록 한다. 즉, 수평 손흔들림 보정량 SX_ADD의 소수부에 대응하는 상기 승산 계수의 조를 테이블로부터 판독출력하고, 그 승산 계수를 수평 FIR 필터에 대해서 공급함으로써 상기 보간 처리를 행할 수가 있다.

[0549] 이상은, 수평 방향의 보간 처리이지만, 수직 방향에 대해서도, 보간 방향이 수직 방향으로 될 뿐으로, 완전히 마찬가지이다.

[0550] [수평 처리 블록(51)에서의 처리 동작]

[0551] [수평 손흔들림 보정량 SX_ADD의 적분 처리]

[0552] 도 45에, 수평 손흔들림 보정량 적분부(511)에서의 적분값으로서의 수평 손흔들림 보정량 SX_ADD를 얻기 위한 적분 처리 동작의 플로차트를 도시한다.

[0553] 우선, 초기 y좌표(수평 라인 위치에 대응) SY에, 수평 동기 신호 H-SYNC(여기에서는, 수평 동기 신호 H-SYNC는, 일반적인 수평 주사 신호 중, 유효 화상 영역 이외를 마스크한 타이밍 신호를 의미하는 것으로 한다)마다 「step」이라고 하는 값을 가산해 간다(스텝 S1).

[0554] 초기 y좌표 SY는, 출력 화상의 최초의 1수평 라인째가, CMOS 이미저 위의 총화소의 좌표계(座標系)(이후, 절대 좌표계라고 부른다)의 어디에 위치하는지를 나타내고 있으며, 일반적으로는 「0」은 아니다.

[0555] 이 초기 좌표 (SX, SY)는, 상술한 바와 같은 해당 프레임의 손흔들림에 따른 초기적 변위량을 포함할 뿐만 아니라, 다음과 같은 요인에 의거해서 정해지는 것이다. 즉, 동화상 시에는, 통상, CMOS 이미저 위의 전화소가, 신호 처리부(50)에 입력되는 것이 아니라, CMOS 이미저 내에서 수직 방향으로 평균 가산되고 나서, 후단의 신호 처리부(50)에 입력되기 때문이다. 또, 신호 처리부(50)에 입력되고 나서도, 몇개의 처리를 거치기(經) 때문에, 1수평 라인째의 절대 좌표는, 「0」이 아닌 것 쪽이 많기 때문이다. 또, 입력 화상 데이터에 대해서, 수직 방향의 부분 확대 처리를 행하는 경우에는, 초기 y좌표 SY는, 화면 중간정도(middle)의 수평 라인 위치로 되므로, 매우 큰 값을 받는 일도 있을 수 있기 때문이다.

[0556] 스텝 S201에서, 1수평 동기 신호 H-SYNC마다 가산하는 「step」이라고 하는 값을, 라인 간격의 절대 좌표계에서의 수직 좌표의 증량(增量)을 나타내지만, 상술한 CMOS 이미저 내 혹은 수평 처리 블록의 전단(前段)에서의, 수직 방향의 복수 라인의 화상 데이터의 평균 가산에 의해, 1이외에 2나 4라고 하는 값을 취할 가능성이 있다.

- [0557] 그런데, 스텝 S201에서, 초기 y좌표값을 SY로 해서, 1수평 동기 신호 H-SYNC마다, 값 step을 가산하는 것에 의해, 현재 처리하고 있는 수평 라인의 절대 좌표가 구해지고, 해당 절대 좌표의 정수 성분 vp_i(스텝 S202)가, 수평 보정 속도 성분 X_STB_*(*는 0~7의 어느것인가이다. 즉, 수평 보정 속도 성분 X_STB_*는, X_STB_0~X_STB_7의 어느것인가를 의미한다. 이하, 같다)를, 현시점에까지 적분해야 할 적분 회수(回數)에 대응한다. 여기에서, 이 적분 회수는, 1화면 내에서의 적산값(積算值)이다.
- [0558] 다음에, 현재 처리하고 있는 수평 라인(이하, 현처리 라인이라고 한다)의 절대 좌표의 정수 성분 vp_i와, 현처리 라인의 1수평 라인전에까지 적분한 회수(적분 회수 hstb_cnt)를 비교한다. 이것에 의해, 현처리 라인에서, 이후에(앞으로) 몇회 적분하면 좋은지를 알 수 있으며, 그 회수만큼, 처리 단위 사이클(1수평 주기보다도 충분히 짧다)마다 트리거 TG1이 발생한다(스텝 S203).
- [0559] 이 트리거 TG1이 발생할 때마다, 적분 회수 hstb_cnt는 1씩 인크리먼트(increment)한다(스텝 S204, 스텝 S205).
- [0560] 그리고, 이 트리거 TG1마다, 수평 보정 속도 성분 X_STB_0~X_STB_7 중의, 현처리 라인이 소재(所在)하는 분할 화상 구간에 대한 수평 보정 속도 성분을 가산하는 것에 의해, 적분 처리를 행하고, 수평 손흔들림 보정량 SX_ADD를 얻는다(스텝 S206). 여기에서, 도 45의 스텝 S206에서의 「stb_x」는, 수평 보정 속도 성분 X_STB_0~X_STB_7의 어느것인가의 값이다. 즉, 「stb_x」는, 현처리 라인이 소재하는 분할 화상 구간에 대한 수평 보정 속도 성분의 값이다.
- [0561] 스텝 S201에서, 1수평 동기 신호 H-SYNC마다 가산되는 값 「step」 = 1로서, 화상의 확대 등이 아닌 통상의 화상 출력의 경우에는, 트리거 TG1은, 1수평 동기 신호 H-SYNC마다 발생하기 때문에, 스텝 S206에서는, 1수평 라인마다 「stb_x」를 가산(적분)하는 것이다.
- [0562] 이렇게 해서 얻어지는 수평 손흔들림 보정량 SX_ADD는, 전술도 한 바와 같이, CMOS 이미지 위에서의 각 수평 라인의 수평 방향의 판독출력 개시 위치의 보정량, 즉 수평 처리 블록(51)에서의 수평 방향 화상 일그러짐 보정 처리에서의 각 라인의 수평 초기 위치(수평 오프셋) 그것(그 자체)으로 된다.
- [0563] 상술한 바와 같이, 스텝 S206에서는, 현처리 라인이, 1화면분(1프레임=1수직 동기 주기)을 수직 방향으로 8분할한 분할 화상 구간의 어디에 속하는지에 따라서, 수평 보정 속도 성분 X_STB_0~X_STB_7 중의, 어느것인가 하나를 선택하지 않으면 안된다. 그것을 위한 짜임새(仕組; mechanism; 메카니즘)가, 도 45의 나머지 부분의 처리이다.
- [0564] 이 예에서는, 분할 화상 구간 분의 라인수의 가산값 stb_grid가 정의(定義)된다(스텝 S207). 이 가산값 stb_grid의 초기값(初期值)은, 1분할 화상 구간의 라인수 DIV이다.
- [0565] 그리고, 이 가산값 stb_grid와, 적분 회수 hstb_cnt를 비교하고, 적분 회수 hstb_cnt가 가산값 stb_grid를 넘을 때마다, 트리거 TG2를 발생한다(스텝 S208). 그리고, 트리거 TG2가 발생할 때마다, 가산값 hstb_cnt에, 1분할 화상 구간의 라인수 DIV를 가산한다(스텝 S209).
- [0566] 이상의 처리에 의해, 현처리 라인이 다음의 분할 화상 구간으로 옮길(移; transit) 때마다, 트리거 TG2가 발생한다. 따라서, 현처리 라인이 각 분할 화상 구간 Pdiv_0~Pdiv_7의 어느것에 존재하는지의 분할 구간값 HDIV_CNT를 정의하고(스텝 S210), 이 분할 구간값 HDIV_CNT를, 트리거 TG2가 발생할 때마다, 1씩 인크리먼트한다(스텝 S211). 그러자, 분할 구간값 HDIV_CNT는, 현처리 라인이 소재하는 각 분할 화상 구간 Pdiv_0~Pdiv_7의 각각에 대응해서 「0」 ~ 「7」의 값을 취한다. 즉, 분할 구간값 HDIV_CNT는, 수평 처리의 진척 정도(度合; degree)를 나타내는 것으로 된다. 스텝 S6에서는, 이 분할 구간값 HDIV_CNT를 참조해서, 「stb_x」를 결정한다.
- [0567] 또한, 이 예에서는, 분할 구간값 HDIV_CNT=8일 때에는, 1프레임 분의 화상 데이터에 대한 처리가 종료한 것을 의미한다. 그리고, 분할 구간값 HDIV_CNT=8에, 「1」이 더해지면, 분할 구간값 HDIV_CNT=0으로 되도록 계산된다.
- [0568] 이 분할 구간값 HDIV_CNT는, 수직 처리 블록(52) 및 레지스터 블록(53)에, 수평 처리 블록(51)에서의 현처리 라인이 어느 분할 화상 구간에 소재하고 있는지를 나타내기 위해서, 즉 수평 처리의 진척 상황을 알리기 위해서 공급된다.
- [0569] [수평 화상 처리부(512)의 처리 동작]

- [0570] 도 46은, 수평 화상 처리부(512)의 구성예를 도시하는 것이다. 이 도 46의 예에서는, 전술한 도 45의 플로차트에서의 수평 동기 신호 H-SYNC마다 값 「step」을 초기 y좌표 SY에 가산하는 가산부(5101)가, 이 수평 화상 처리부(512)에 포함되는 것으로 하고 있으며, 이 때문에, 이 수평 화상 처리부(512)에 초기 y좌표 SY가 입력된다. 또, 이 수평 화상 처리부(512)에는 초기 x좌표 SX가 입력된다. 이를 초기 y좌표 SY 및 초기 x좌표 SX는, 1프레임의 선두에서, 예를 들면 제어부를 구성하는 CPU(1)에서 설정되어 입력된다.
- [0571] 그리고, 가산부(5101)로부터의, 수평 동기 신호 H-SYNC마다 값 「step」이 가산된 y좌표값 SY'는, 수평 손흔들림 보정량 적분부(511)에 공급된다. 그러면, 이 수평 손흔들림 보정량 적분부(511)로부터는, 현처리 라인의 선두의 x좌표의 오프셋값으로서, 수평 손흔들림 보정량 SX_ADD가 되돌아온다.
- [0572] 수평 화상 처리부(512)에서는, 가산부(5102)에서, 각 수평 라인의 선두에서 1회만, 수평 손흔들림 보정량 적분부(511)로부터의 수평 손흔들림 보정량 SX_ADD를, 초기 x좌표 SX에 가산한다.
- [0573] 그리고, 이 가산부(5102)의 출력값에 대해서, 화소 처리 클럭 CLK마다, 수평 확대·축소의 파라미터인 값 「hmag」를, 가산부(5103)에 의해 가산한다. 이 가산부(5103)의 출력값 X_ADD는, 비교부(5104 및 5105), 또 소수부 추출부(5106) 및 정수부 추출부(5107)에 공급된다.
- [0574] 비교부(5104)는, 가산부(5103)으로부터의 출력값 X_ADD를, 1화소 처리 클럭 CLK만큼 앞(前)의 출력값 X_ADD의 정수 성분과 비교하고, 차가 1이상일 때, 즉 화소 위치가 1화소분 이상 점프했을 때에, 그 점프를 나타내는 신호 SKIP을 출력한다. 이 예에서는, 이 점프를 나타내는 신호 SKIP을 참조함으로써, 다음에 화소 처리 클럭 CLK의 타이밍에서 처리해야 할 점프 한 화소 위치를 알 수가 있다. 이 신호 SKIP은, FIFO 메모리 컨트롤러(5108)에 공급된다.
- [0575] 또, 비교부(5105)는, 가산부(5103)로부터의 출력값 X_ADD를, 1화소 처리 클럭 CLK만큼 앞의 출력값 X_ADD의 정수 성분과 비교하고, 차가 1보다 작을 때에, 그 취지(旨)를 나타내는 신호 HOLD를 출력한다. 따라서, 이 신호 HOLD를 참조함으로써, 판독출력 화소가, 1화소 처리 클럭 CLK만큼 앞(前)과 같음을 알 수가 있다. 이 신호 HOLD는, FIFO 메모리 컨트롤러(5108)에 공급된다.
- [0576] 소수부 추출부(5106)는, 가산부(5103)로부터의 출력값 X_ADD의 소수부를, 화소 처리 클럭 CLK마다 추출하고, 추출한 소수부의 값 X_PHASE를 출력한다. 이 소수부의 값 X_PHASE는, 수평 FIR 필터(5110)를 구비하는 수평 보간 처리부(5109)에 공급된다. 수평 보간 처리부(5109)에서는, 이 소수부의 값 X_PHASE에 의거해서 승산 계수 테이블을 검색하고, 수평 FIR 필터(5110)에 공급할 승산 계수를 얻도록 한다.
- [0577] 정수부 추출부(5107)는, 가산부(5103)로부터의 출력값 X_ADD의 정수부를, 수평 동기 신호 H-SYNC에 의해 각 수평 라인의 선두에서 추출하고, 추출한 정수부의 값 ST_POS를 출력한다. 이 정수부의 값 ST_POS는, 현처리 라인의 초기 x좌표로서 FIFO 메모리 컨트롤러(5108)에 공급된다.
- [0578] FIFO 메모리 컨트롤러(5108)는, 입력 화상 데이터 Din에 대해서, 수평 처리용 FIFO 라인메모리(61)를 이용해서, 수평 손흔들림 보정량 SX_ADD에 의거한 수평 손흔들림 보정 처리를 행함과 동시에, 수평 확대·축소 처리를 행한다. 여기에서, 수평 처리용 FIFO 라인메모리(61)가, 실제로 1 수평 라인분 상당의 사이즈를 필요로 하는 것은, 후술하는 바와 같이, 화상의 부분 확대 처리를 서포트하는 것을 고려하는 경우뿐이다.
- [0579] 즉, FIFO 메모리 컨트롤러(5108)는, 입력 화상 데이터 Din에 대해서, 상기 정수부의 값 ST_POS를 이용해서, 출력 화상 데이터로서 판독출력하는 각 수평 라인의 화소 데이터의 초기 x좌표를 결정한다. 또, FIFO 메모리 컨트롤러(5108)는, 신호 SKIP 및 신호 HOLD를 화소 처리 클럭 CLK마다 참조하고, 수평 처리용 FIFO 라인메모리(61)로부터 판독출력할 화소 데이터를 결정하도록 한다.
- [0580] 그리고, FIFO 메모리 컨트롤러(5108)는, 이상과 같이 해서 판독출력한 수평 라인의 화소 데이터를 출력 화상 데이터로서, 데이터 인에이블 신호 EN과 함께, 수평 보간 처리부(5109)에 공급한다.
- [0581] 수평 보간 처리부(5109)는, 데이터 인에이블 신호 EN이 인에이블 상태로 되어 있을 때에 입력되는 화상 데이터를 유효 데이터로 해서, 수평 FIR 필터(5110)를 이용해서 수평 방향의 보간 처리를 실행한다. 전술한 바와 같이, 이 때에 수평 보간 처리부(5109)에 입력되는 소수부의 값 X_PHASE를 이용해서, 수평 FIR 필터(5110)에 공급할 승산 계수가 결정되고 있다.
- [0582] 이렇게 해서, 수평 보간 처리부(5109)로부터는, 수평 손흔들림 보정량 SX_ADD에 따라서 수평 손흔들림 보정 처리가 이루어짐과 동시에, 수평 방향의 확대·축소 처리가 이루어진 화상 데이터 DHout가 얻어진다. 이

DHout는, 수직 처리 블록(52)에 공급된다.

[0583] 그런데, 상술한 설명에서는, 수평 손흔들림 보정량 적분부(511)의 수평 손흔들림 보정량 SX_ADD의 적분 처리와, 수평 화상 처리부(511)의 수평 확대·축소 화상 처리의, 서로의 처리 타이밍에 대해서 특별히 언급하지 않았지만, 수평 손흔들림 보정량 SX_ADD의 적분 처리를, 자연 없이(遲延無), 수평 확대·축소 화상 처리와의 단순한 시퀀스 처리로 행하는 것이 안성맞춤인 것은, 도 47의 상측의 도 47의 (A)에 도시하는 바와 같이, 수직 방향의 부분 절출 및 부분 확대를 서포트하지 않는 경우에 거의 한정된다. 도 47에서, V-SYNC는 수직 동기 신호를 나타내고, 또 ST_TG는, 유효 주사 라인의 개시 시점을 나타내고 있다. 또, 도 47에서, 망긋기를 부가해서 나타낸 사각(四角; rectangle)으로 둘러싸인 구간은, 각각의 처리 구간을 나타내고 있다.

[0584] 한편, 수직 방향의 부분 절출, 혹은 부분 확대를 서포트하는 경우, 초기 y좌표 SY가 매우 큰 값을 취할 수 있기 때문에, 프레임 선두의 적분 처리에, 상당히 긴 사이클수(화소 처리 사이클수)를 요(要)해 버릴 가능성이 있다. 도 47 하측의 (B)의 시퀀스에서의 기간 tm이 그것을 위한 적분 기간이다.

[0585] 이와 같이 되면, 도 47 상측의 (A)의 시퀀스에서는, 최초의 확대·축소 처리가 수평 동기 신호 H-SYNC의 1주기 기간을 오버해 버리기 때문에, 다음의 수평 동기 신호 H-SYNC로 입력되어 오는 2라인째의 화상 데이터를 보존유지해 두기 위한 라인메모리가 필요하게 된다.

[0586] 그래서, 이 실시형태에서는, 도 47 하측의 (B)의 시퀀스에 도시하는 바와 같이, 수평 확대·축소 화상 처리가 개시되는 1수평 주기 전부터 적분 처리의 기동(起動)을 행하고, 항상 화상 처리에 대해서, 1수평 주기만큼 선행(先行)해서 적분 처리를 행하는 타이밍 및 시퀀스를 채용한다. 이 때문에, 수평 처리용 FIFO 라인메모리(61)는, 1수평 주기 분의 사이즈의 것이 이 실시형태에서는 이용되고 있다.

[0587] 상술한 문제는, 화상의 부분 확대 등에 관계없이, 수평 처리는 항상 전입력(全入力) 라인에 대해서 처리를 행한다고 하는 아키텍처(architecture)로 하면 회피할 수 있다. 그러나, 어쨌든, 도 47 상측의 시퀀스를 실현하려고 하면, 적분 처리의 종료 신호를 새롭게 만들고(設; 생성하고), 그것을 받아서 화상 처리의 기동을 결지 않으면 안되고, 약간 번잡하다. 다시말해, 바꿔말하면, 상술한 이 실시형태에서 채용한 수법은, 기존의 화상 처리의 회로에 새로운 손을 보지 않아도(구성요소를 더 부가하지 않아도) 된다고 하는 메리트가 있는 것이다.

[0588] 또한, 상술한 실시형태에서 채용한 수법은, 수평 동기 신호 H-SYNC의 1주기 기간 내의 화소 처리 사이클수보다도, 절대 좌표계의 수직 사이즈가 큰 경우에는 채용할 수 없다. 그 경우에는, 수평 처리는 항상 전입력 라인에 대해서 처리를 행한다고 하는, 상술한 아키텍처로 하면 좋다.

[0589] [수직 처리 블록(52)에 있어서의 처리 동작]

[0590] [수직 손흔들림 보정량 SY_ADD의 적분 처리]

[0591] 다음에, 수직 손흔들림 보정량 적분부(521)에서의 적분값으로서의 수직 손흔들림 보정량 SY_ADD를 얻기 위한 적분 처리 동작에 대해서 설명한다. 도 48은, 이 수직 손흔들림 보정량 SY_ADD를 얻기 위한 적분 처리 동작의 플로차트를 도시하는 것이다.

[0592] 이 수직 손흔들림 보정량 SY_ADD를 얻기 위한 적분 처리 동작은, 도 45를 이용해서 상술한 수평 손흔들림 보정량 SX_ADD를 얻기 위한 적분 처리 동작과 유사하지만, 다음의 점이 다르다.

[0593] 즉, 수평 동기 신호 H-SYNC마다 초기 y좌표 SY에 가산하는 값이 「v mag」인 것과, 수평 처리의 진척 정도를 나타내는 HDIV_CNT를 보고, 웨이트(WAIT)를 가(掛; apply)하는 것과, 각 라인의 적분 처리가 종료한 것을 나타내는 적분 완료 신호 STB_RDY를 생성해서, 수직 신호 처리 모듈에 전달하는 것의 이상(以上) 3점이다.

[0594] 이 실시형태에서는, 수직 화상 처리에서 수직 방향의 확대·축소를 상정하고 있고, 값 「v mag」는 수직 확대·축소의 배율 파라미터를 나타내고 있다. 따라서, 수직 처리 블록(52)에서는, 수평 처리 블록(51)으로부터 출력되는 화상 데이터를 모두 처리하는 것이 아니라, 확대·축소 처리 후에 출력하는 라인의 수직 좌표를 기준으로해서 처리를 행한다.

[0595] 그 때문에, 이 수직 방향의 적분 처리에서도, 실제로, 이 수직 방향의 적분 처리 후의 수직 화상 처리의 대상으로 되는 좌표에서의 적분값을 출력하는 것이 요구되기 때문에, 수직 방향의 좌표 증분(增分) 파라미터를 수평 동기 신호 H-SYNC마다 가산하는 것이다. 또한, 화상의 확대·축소를 수반하지 않고, CMOS 손흔들림의 보정만을 행한다면, v mag=1로서 좋다.

[0596] 수평 방향의 적분 처리와는 두번째의, 분할 구간값 HDIV_CNT를 보고, 수평 처리의 진척 정도를 관측하는 필요성

은, 수평 처리를 수직 처리가 추월(追越)하는 일이 있어서는 안되며 때문이다. 일반적인 화상의 수직 확대·축소 처리에서는, 수직 처리가 수평 처리를 추월한다고 하는 것은, 아직 생성되어 있지 않은 화상에 대해서 처리를 행하는 것이기 때문에, 스스로 웨이트를 가하는 짜임새가 필요하게 된다. 마찬가지로, 이 수직 방향의 적분 처리에서도, 화상을 직접 취급하는 것은 아니지만, 수평 처리를 추월하지 않는 메카니즘(仕掛; mechanism)이 필요한 것이다.

[0597] 좀 더 상세하게 설명한다. 수평 및 수직 방향의 적분 파라미터인 수평 보정 속도 성분 X_{STB_*} (*는 0~7의 어느것인가이고, X_{STB_*} 는, $X_{STB_0} \sim X_{STB_7}$ 의 어느것인가를 의미한다. 이하, 같다)와, 수직 보정 속도 성분 Y_{STB_*} (*는 0~7의 어느것인가이고, $Y_{STB_0} \sim Y_{STB_7}$ 의 어느것인가를 의미한다. 이하, 같다)는, 처리를 행하고 있는 수평 라인의 샘플링 시각(엄밀하게 말하면, 노광 기간의 중심 시점인 것은 이미 설명) 부근의 손흔들림 속도 벡터(손흔들림의 속도와 방향)의, 각각 수평 성분, 수직 성분의 역부호이다. 따라서, 노광 시간이 지극히 짧은 시간의 경우까지 고려하면, 수평 보정 속도 성분 X_{STB_*} 와 수직 보정 속도 성분 Y_{STB_*} 는, 대응하는 수평 라인의 화상 데이터가 입력되기 직전까지 확정할 수 없는 경우가 있다.

[0598] 나중에 상세하게 기술하는 레지스터 블록(53)에서는, 이 사정을 고려하고, 수평 방향의 적분 처리에서 현재 처리하고 있는 라인이 속하는 분할 화상 구간을 의미하는 분할 구간값 HDIV_CNT의 값을 보고, 수평 보정 속도 성분 X_{STB_*} 와 수직 보정 속도 성분 Y_{STB_*} 를, 수평 처리 블록용 레지스터와, 수직 처리 블록용 레지스터에 래치(latch)한다. 따라서, 수평 처리 블록(51)에서는, 특별한 짜임새를 설치하지 않아도, 입력된 화상을 순차 처리해 가면 좋은 것이다.

[0599] 한편, 수직 처리 블록(52)에서의 적분 처리에서는, 해당 적분 처리를 행한 결과를, 후단의 수직 방향 화상 처리의 초기 오프셋 좌표로 하기 위해서, 수평 처리로부터의 화상이 아직도 들어와(입력되어) 있지 않은 것을 나중에 깨달아도, 그 시점에서는 이미 그 라인의 적분 처리는 끝나 버려 있다. 다시말해, 그 때에는, 미확정(未確定)의 수직 보정 속도 성분이 이용되어 적분이 이루어져 버린 것으로 된다.

[0600] 그 때문에, 적분 처리 단계에서, 처리하려고 하는 라인에 대응한 적분 파라미터인 수직 보정 속도 성분 Y_{STB_*} 가, 아직 확정 전인 것을, 수평 처리의 진척 정도로부터 판단할 필요가 있는 것이다.

[0601] 그것을 위한 짜임새로서, 도 48과 같이, 수직 처리의 진척 정도를 나타내는 분할 구간값 VDIV_CNT가, 수평 처리의 진척 정도를 나타내는 분할 구간값 HDIV_CNT를 넘어 버리고 있으면, 웨이트를 발행하고, 도 48의 왼쪽 아래(左下)에서, 수직 처리에서의 현처리 라인이 소재하는 분할 화상 구간에 대한 수직 보정 속도 성분의 값인 「 stb_y 」가 미확정인 채의 수직 보정 속도 성분 Y_{STB_*} 가 선택된 상태에서, 수직 손흔들림 보정량 SY_ADD에 가산되는 것을 방지한다.

[0602] 수직 방향의 적분 처리(도 48)가 수평 방향의 적분(도 45)과 다른, 세번째의 점은, 각 라인에서의 수직 적분이 완료한 것을 나타내는 적분 완료 신호 STB_RDY의 생성이다. 각 라인 내에서 적분을 되풀이해서 행한 결과, 그 라인의 절대 좌표의 정수 성분 vp_i 와, 그 프레임 내에서의 누적 적분 회수 $vstb_cnt$ 가 일치했을 때에, 후단의 수직 화상 처리 모듈에 전네주어야(渡; supply) 할 수직 오프셋 SY_ADD가 구해지고, 그것과 동시에, 후단의 수직 화상 처리의 기동을 거는 것이 목적이다.

[0603] 이미 설명한 수평 방향의 적분 처리에서는, 유효 화상이 입력되는 타이밍보다도 1수평 동기 신호 H-SYNC(1수평 주기) 전부터 적분 처리를 개시해 두면, 유효 화상에 대한 수평 화상 처리가 시작되기까지 적분 처리를 완료할 수 있는 것이, 대부분의 경우, 시스템적으로 보증가능하지만, 수직 방향의 적분 처리에서는, 전술한 웨이트가 발생할 가능성이 있기 때문에, 직전 1수평 주기의 시간에서 적분이 완료하는 것을, 모든 경우에 있어서 보증하는 것은 불가능하기 때문이다. 즉, 웨이트가 발생했을 때에는, 수직 적분이 완료한 것을 나타내는 적분 완료 신호 STB_RDY가 발생했을 때에, 수직 손흔들림 보정 및 수직 확대·축소 처리를 포함하는 수직 화상 처리를 실행하도록 한다.

[0604] 이상의 기능을 가지는, 이 수직 방향의 적분 회로를 적용한 수직 방향의 적분 처리를 도 48의 플로차트를 참조하면서 설명한다.

[0605] 우선, 초기 y좌표(수평 라인 위치에 대응) SY에, 수평 동기 신호 H-SYNC마다 전술한 값 「 v_{mag} 」를 가산해 간다(스텝 S301).

[0606] 스텝 S21에서, 초기 y좌표값을 SY로 해서, 1수평 동기 신호 H-SYNC마다, 값 v_{mag} 를 가산하는 것에 의해, 현재 처리하고 있는 수평 라인의 절대 좌표가 구해지고, 해당 절대 좌표의 정수 성분 vp_i (스텝 S302)가, 수직 보정 속도 성분 Y_{STB_*} ($Y_{STB_0} \sim Y_{STB_7}$ 의 어느것인가)를, 현시점에까지 적분해야 할 적분 회수에 대응한다. 여기

에서, 이 적분 회수는, 1화면 내에서의 적산값이다.

[0607] 다음에, 현재 처리하고 있는 수평 라인, 즉, 현처리 라인의 절대 좌표의 정수 성분 v_{p_i} 와 현처리 라인의 1수평 라인전에까지 적분한 회수(적분 회수 v_{stb_cnt})를 비교한다. 이것이 의해, 현처리 라인에서, 이후에(앞으로) 몇회 적분하면 좋은지를 알 수 있으며, 그 회수만큼, 처리 단위 사이클(1수평 주기보다도 충분히 짧다)마다 트리거 TG3이 발생한다(스텝 S303).

[0608] 이 트리거 TG3이 발생할 때마다, 적분 회수 v_{stb_cnt} 는 1씩 인크리먼트한다(스텝 S304, 스텝 S305).

[0609] 그리고, 이 트리거 TG3마다, 수직 보정 속도 성분 $Y_{STB_0} \sim Y_{STB_7}$ 중의, 현처리 라인이 소재하는 분할 화상 구간에 대한 수직 보정 속도 성분을 가산하는 것에 의해, 적분 처리를 행하고, 수직 손흔들림 보정량 SY_ADD 를 얻는다(스텝 S306). 여기에서, 도 48의 스텝 S306에서의 「 stb_y 」는, 수직 보정 속도 성분 $Y_{STB_0} \sim Y_{STB_7}$ 의 어느것인가의 값이다. 즉, 「 stb_y 」는, 현처리 라인이 소재하는 분할 화상 구간에 대한 수직 보정 속도 성분의 값이다.

[0610] 스텝 S301에서, 1수평 동기 신호 H-SYNC마다 가산되는 값 「 v_{mag} 」 = 1로서, 화상의 확대 등이 아닌 통상의 화상 출력의 경우에는, 트리거 TG3은, 1수평 동기 신호 H-SYNC마다 발생하기 때문에, 스텝 S26에서는, 1수평 라인마다 「 stb_y 」를 가산(적분)하는 것이다.

[0611] 이렇게 해서 얻어지는 수직 손흔들림 보정량 SY_ADD 는, 전술도 한 바와 같이, CMOS 이미저 위에서의 각 수평 라인의 수직 방향의 판독출력 개시 위치의 보정량, 즉 수직 처리 블록(52)에서의 수직 방향 화상 일그러짐 보정 처리에서의 각 라인의 수직 위치(수직 오프셋) 그 자체로 된다.

[0612] 상술한 바와 같이, 스텝 S306에서는, 현처리 라인이, 1화면분(1프레임=1수직 동기 주기)을 수직 방향으로 8분할한 분할 화상 구간의 어디에 속하는지에 따라서, 수직 보정 속도 성분 $Y_{STB_0} \sim Y_{STB_7}$ 중의, 어느것인가 하나를 선택하지 않으면 안된다.

[0613] 이 예에서는, 분할 화상 구간 분의 라인수의 가산값 v_{stb_grid} 가 정의된다(스텝 S307). 이 가산값 v_{stb_grid} 의 초기값은, 1분할 화상 구간의 라인수 DIV이다.

[0614] 그리고, 이 가산값 v_{stb_grid} 와, 적분 회수 v_{stb_cnt} 를 비교하고, 적분 회수 v_{stb_cnt} 가 가산값 v_{stb_grid} 를 넘을 때마다, 트리거 TG4를 발생한다(스텝 S308). 그리고, 트리거 TG4가 발생할 때마다, 가산값 v_{stb_cnt} 에, 1분할 화상 구간의 라인수 DIV를 가산한다(스텝 S309).

[0615] 이상의 처리에 의해, 현처리 라인이 다음의 분할 화상 구간으로 옮길 때마다, 트리거 TG4가 발생한다. 따라서, 현처리 라인이 각 분할 화상 구간 $Pdiv_0 \sim Pdiv_7$ 의 어느것에 존재하는지의 분할 구간값 $VDIV_CNT$ 를 정의하고(스텝 S310), 이 분할 구간값 $VDIV_CNT$ 를, 트리거 TG4가 발생할 때마다, 1씩 인크리먼트한다(스텝 S311). 그러면, 분할 구간값 $VDIV_CNT$ 는, 현처리 라인이 소재하는 각 분할 화상 구간 $Pdiv_0 \sim Pdiv_7$ 의 각각에 대응해서 「0」 ~ 「7」의 값을 취한다. 즉, 분할 구간값 $VDIV_CNT$ 는, 수직 처리의 진척 정도를 나타내는 것으로 된다.

[0616] 또한, 이 예에서는, 분할 구간값 $VDIV_CNT=8$ 일 때에는, 1프레임 분의 화상 데이터에 대한 처리가 종료한 것을 의미한다. 그리고, 분할 구간값 $VDIV_CNT=8$ 에, 「1」이 더해지면, 분할 구간값 $VDIV_CNT=0$ 으로 되도록 계산된다.

[0617] 이 분할 구간값 $VDIV_CNT$ 는, 수평 처리 블록(51)으로부터의, 수평 처리의 진척 정도를 나타내는 분할 구간값 $HDIV_CNT$ 와 비교되고, 분할 구간값 $VDIV_CNT$ 가 분할 구간값 $HDIV_CNT$ 보다도 크게 되었을 때에는, 수평 처리를 수직 처리가 추월해 버리는 상태를 의미하므로, 웨이트(WAIT)를 발생해서, 트리거 TG3의 발생을 대기(待機)시킨다. 즉, 트리거 TG3은, 웨이트가 해제된 후에 발생하도록 지연시킨다(스텝 S312).

[0618] 그리고, 이 실시형태에서는, 스텝 S302에서 얻어지는 y좌표의 정수 성분 v_{p_i} 와, 적분 회수 v_{stb_cnt} 가 똑같은지 여부를 체크하고, 똑같으면 현처리 라인에 대한 수직 방향의 적분이 완료한 것을 나타내는 적분 완료 신호 STB_RDY 를 생성해서 출력한다(스텝 S313). 이 적분 완료 신호 STB_RDY 는, 수직 화상 처리부(522)에 공급된다. 수직 화상 처리부(522)에서는, 이 적분 완료 신호 STB_RDY 에 의해, 현처리 라인에 대한 수직 방향의 손흔들림 보정 처리를 개시해도 좋은 것을 인식하고, 다음의 수평 동기 신호 H-SYNC의 타이밍에서 수직 화상 처리를 재개한다.

[0619] [수직 화상 처리부(522)의 처리 동작]

[0620] 도 49는, 수직 화상 처리부(522)의 구성예를 도시하는 것이다. 이 도 49의 예에서는, 전술한 도 48의 플로차트

에서의 수평 동기 신호 H-SYNC마다 값 「v mag」를 가산하는 가산부(5201)가, 이 수직 화상 처리부(522)에 포함되는 것으로 하고 있고, 이 수직 화상 처리부(522)에 초기 y좌표 SY가 입력된다.

[0621] 그리고, 가산부(5201)로부터의, 수직 동기 신호 H-SYNC마다 값 「v mag」가 가산된 y좌표값 SY' 는, 수직 손흔들림 보정량 적분부(521)에 공급된다. 그러면, 이 수직 손흔들림 보정량 적분부(521)로부터는, 현처리 라인의 선두의 y좌표의 오프셋값으로서, 수직 손흔들림 보정량 SY_ADD가 되돌아온다.

[0622] 수직 화상 처리부(522)에서는, 가산부(5202)에서, 각 수평 라인의 선두에서 1회만, 수직 손흔들림 보정량 적분부(521)로부터의 수직 손흔들림 보정량 SY_ADD를, 초기 y좌표 SY에 가산한다. 이 가산부(5202)의 출력값 X_ADD는, 소수부 추출부(5203) 및 정수부 추출부(5204)에 공급된다.

[0623] 소수부 추출부(5203)는, 가산부(5202)로부터의 출력값 Y_ADD의 소수부를, 수평 동기 신호 H-SYNC에 의해 각 수평 라인의 선두에서 추출하고, 추출한 소수부의 값 Y_PHASE를 출력한다. 이 소수부의 값 Y_PHASE는, 수직 FIR 필터(5207)를 구비하는 수직 보간 처리부(5206)에 공급된다.

[0624] 수직 FIR 필터(5207)는, 수직 방향으로 늘어서는(暨; arranged) 복수 라인의 화소 데이터를 이용해서 보간 처리를 행하는 것으로, 수직 보간 처리부(5206)에서는, 이 소수부의 값 Y_PHASE에 의거해서 승산 계수 테이블을 검색하고, 수직 FIR 필터(5207)에 공급할 승산 계수를 얻도록 한다. 이것은, 보간 방향이 수직 방향으로 될 뿐으로, 도 42~도 44를 이용해서 설명한 수평 FIR 필터를 이용하는 수평 방향의 보간 처리와 거의 마찬가지이다.

[0625] 정수부 추출부(5204)는, 가산부(5202)로부터의 출력값 Y_ADD의 정수부를, 수평 동기 신호 H-SYNC에 의해 각 수평 라인의 선두에서 추출하고, 추출한 정수부의 값 Y_POS를 출력한다. 이 정수부의 값 Y_POS는, 현처리 라인의 y좌표로서 메모리 컨트롤러(5205)에 공급된다.

[0626] 메모리 컨트롤러(5205)는, 수평 처리 블록(51)으로부터의 화상 데이터 DHout에 대해서, 수직 처리용 소규모 라인메모리(62)를 이용해서, 수직 손흔들림 보정량 SY_ADD에 의거한 수직 손흔들림 보정 처리를 행함과 동시에, 수직 확대·축소 처리를 행한다. 여기에서, 수직 처리용 소규모 라인메모리(62)는, 적어도 수직 FIR 필터(5207)에서의 보간 처리에 이용하는 라인수분, 즉, 수직 FIR 필터의 템수분의 용량을 가진다.

[0627] 메모리 컨트롤러(5205)는, 그 입력 화상 데이터 DHout를 수직 처리용 소규모 라인 메모리(62)에 일시 보존(保存)함과 동시에, 상기 정수부의 값 Y_POS를 이용해서, 출력 화상 데이터 DYout_0, DYout_1, …, DY_n으로서 출력하는 복수의 수평 라인의 데이터를 결정한다. 그리고, 메모리 컨트롤러(5205)는, 이상과 같이 해서 결정한 복수 수평 라인의 화소 데이터 DYout_0, DYout_1, …, DY_n을, 소규모 라인메모리(62)로부터 판독출력해서 수직 보간 처리부(5206)에 공급한다.

[0628] 수직 보간 처리부(5206)는, 입력되는 화상 데이터 DYout_0, DYout_1, …, DY_n을, 수직 FIR 필터(5207)를 이용해서 수직 방향의 보간 처리를 실행한다. 전술한 바와 같이, 이 때에 수직 보간 처리부(5206)에 입력되는 소수부의 값 Y_PHASE를 이용해서, 수직 FIR 필터(5207)에 공급할 승산 계수가 결정되고 있다.

[0629] 이렇게 해서, 수직 보간 처리부(5206)로부터는, 수직 손흔들림 보정량 SY_ADD에 따라서 수직 손흔들림 보정 처리가 이루어짐과 동시에, 수직 방향의 확대·축소 처리가 이루어진 화상 데이터 Dout가 얻어진다.

[0630] 다음에, 도 50에, 위에서 설명한 수직 방향의 적분 처리와, 그 후단의 수직 화상 처리의 타이밍 관계를 도시한다. 도 50의 위(上)의 도 50의 (A)와 같이, 수직 방향의 부분 절출, 수직 방향의 부분 확대·축소의 어느것도 서포트하지 않는 경우에는, 전술한 웨이트는 발생하지 않고, 현처리 라인에 대한 수직 방향의 적분 회수의 상한(上限)은 고작 수회(數回)이다. 이 때문에, 동일한 라인에 대한 적분 처리와 화상 처리의 양쪽을, 같은 1수평 동기 신호 H-SYNC 기간내에서 시퀀스화 가능하다. 따라서, 이 경우에는, 일부러 적분 완료 신호 STB_RDY를 도입할 필요는 없다.

[0631] 한편, 도 50의 아래(下)의 도 50의 (B)와 같이, 수직 방향의 부분 절출, 혹은 수직 방향의 부분 확대를 서포트하는 경우에는, 1라인째의 적분 처리 도중에서 웨이트가 발생할 가능성이 있고, 축소를 서포트하는 경우에는, 어딘가 도중의 수평 라인에서 웨이트가 발생할 가능성이 있다.

[0632] 어느 경우에는, 동일 라인의 적분 처리가, 웨이트에 의해서 수(數) 수평 동기 신호 H-SYNC 구간 혹은 그 이상에 걸쳐서 중단되기 때문에, 적분 처리가 종료한 것을 후단에 알리기 위한 적분 완료 신호 STB_RDY가 필요하게 된다. 그리고, 도 50의 (B)의 아래 2단(段)에 도시하는 바와 같이, 수직 확대·축소 처리는, 웨이트가 종료하

고, 적분 완료 신호 STB_RDY가 발생한 후의 수평 동기 신호 H-SYNC의 타이밍부터 개시된다.

[0633] 또, 동시에, 어떤 라인의 수직 화상 처리가 완료하고 있지 않은 경우에는, 다음의 수평 동기 신호 H-SYNC를, 도 50의 (B)에서 점선으로 나타내는 바와 같이, 마스크하도록 한다. 즉, 웨이트가 발생하고 있는 동안의 수평 동기 신호 H-SYNC는, 마스크하도록 한다.

[0634] 이상 설명한 수평 처리 블록(51)과 수직 처리 블록(52)을 포함하는 신호 처리부(5)에서의 처리의 흐름은, 도 51에 도시하는 바와 같이 나타낼 수가 있다. 즉, 데이터 변환부(3)로부터의 화상 데이터 Din은, 수평 처리용 FIFO 라인메모리(61)에 일단 기입해서 판독출력하고, 수평 화상 처리부(512) 및 수평 손흔들림 보정량 적분부(511)를 이용해서 수평 손흔들림 보정 및 수평 확대·축소 처리를 행한다. 그리고, 수평 화상 처리부(512)의 처리 결과는, 수직 처리용 소규모 라인메모리(62)에 기입한다.

[0635] 수직 처리용 소규모 라인메모리(62)에 기입된 화상 데이터는, 수직 화상 처리부(522) 및 수직 손흔들림 보정량 적분부(521)를 이용해서 수직 손흔들림 보정 및 수직 확대·축소 처리를 행한다. 그리고, 수직 화상 처리부(522)의 처리 결과는, 수직 처리용 소규모 라인 메모리(62)에 다시쓰기한다. 그리고, 또 이 수직 처리용 소규모 라인메모리(62)로부터, 출력 화상 데이터 Dout를 판독출력해서 출력한다.

[0636] 이상의 처리 결과, 도 52의 오른쪽 아래 도면(右下圖)과 같이, CMOS 손흔들림에 기인하는 일그러짐 보정이 이루어진 출력 화상이 얻어진다.

[0637] 여기에서의 수직 처리용 소규모 라인메모리(62)로서는, 최저한, 수직 FIR 필터(5207)의 템수분의 라인메모리가 필요한 것을 기술했지만, 현실적으로는, 수직 방향의 부분 확대 처리를 행하는 경우에는, 수직 처리의 스피드보다도 수평 처리의 스피드 쪽이 빨라지기 때문에, 확대 대상으로 되는 수평 처리 결과를 모아들(溜; accommodating) 정도의 라인메모리가 필요하게 된다.

[0638] 또, 부분 확대를 하지 않는 경우에서도, 수직 보정 속도 성분 Y_STB_*가 연직 상향(上向)인 경우에는, 부분 확대의 경우와 동작상 같게 되기 때문에, 소규모 라인메모리(62)로서는, 손흔들림이 상하 방향으로 변동하는 분량(分)을 고려한 라인메모리가 필요하다.

[0639] 또한, 상술한 실시형태의 경우, 1수직 주기가 1/60초라고 하면, 분할 화상 구간 Pdiv의 시간 길이 DIV는, $1/60 \div 8 = 1/480$ 초이지만, 실험에 의하면, 이 정도의 시간 길이 DIV에서, 손흔들림 속도 벡터를 구해서 손흔들림 보정을 하는 것에 의해, 필요 충분한 손흔들림 보정 효과가 얻어지고, 보정 후의 출력 화상으로서는, 양호한 것으로 되는 것이 확인되고 있다.

[0640] 다음에, 이상의 수평 처리 및 수직 처리에서의 레지스터 블록(53)에서의 레지스터에의 래치 타이밍을, 도 53에 도시한다.

[0641] 즉, 이 실시형태에서는, 수직 동기 신호 V-SYNC의 1주기에 대해서, 각 분할 화상 구간 Pdiv_0~Pdiv_7의 선두 라인의 시점에서 인터럽트(割入; interrupt)(마이크로컴퓨터 인터럽트)가 발생한다.

[0642] 이 인터럽트에 의해 기동되어, 속도 벡터 처리부(80)에서, 전술한 바와 같이 해서, 수평 보정 속도 성분 X_STB_* 및 수직 보정 속도 성분 Y_STB_*가 생성되고, 그 생성 결과인 수평 보정 속도 성분 X_STB_* 및 수직 보정 속도 성분 Y_STB_*가 레지스터 블록(53)의 IF 레지스터에 취입된다.

[0643] 수평 보정 속도 성분 X_STB_* 및 수직 보정 속도 성분 Y_STB_*의 값을, CPU(1)의 소프트웨어로 계산하는 경우, 도 52에 도시하는 바와 같은 정기적인 인터럽트를 발생시키는 방법이 유효하다.

[0644] 또, 레지스터 블록(53) 내에서는, 분할 구간값 HDIV_CNT의 값의 변화를 보고 래치 펄스를 생성하고, 전술한 바와 같이 CPU(1)가 설정한, 대응하는 수평 보정 속도 성분 X_STB_*와 수직 보정 속도 성분 Y_STB_*를, IF 레지스터로부터, 수평 처리용 레지스터 및 수직 처리용 레지스터에, 이 생성한 래치 펄스의 타이밍에서 래치한다. 다시말해, 수평 처리 블록(51)에서 이용하는 수평 보정 속도 성분 X_STB_*를, 그 사용 직전에서 래치하는 것이다. 이 때, 수직 처리 블록(52)에서 이용하는 수직 보정 속도 성분 Y_STB_*도 함께 래치해 둔다.

[0645] 수직 처리 블록(52)에서 사용하는, 수직 보정 속도 성분 Y_STB_*는, 기본적으로는, 상기의 수평 처리 블록(51)용 수평 보정 속도 성분 X_STB_*와 함께 래치한, Y_STB_*의 값을 항상 계속해서 래치한다.

[0646] 단, 그것은, 수직 처리 블록(52)의 프레임 처리 기동 펄스 VL1이 액티브(하이 상태가 액티브)로 되고 나서, 수평 처리 블록(51)의 프레임 처리 기동 펄스 VL0가 액티브(하이 상태가 액티브)로 될 때까지의 기간(도 52의 RV 래치 인에이블의 액티브(하이 상태가 액티브) 기간)에 한정된다.

- [0647] 이것은, 수직 처리의 시간은 길어지는 일이 있으며, 입력 화상의 수직 동기 신호 V-SYNC 뿐만 아니라, 다음의 프레임의 수평 처리 블록의 프레임 기동 펄스 VL0조차도 걸쳐 있는(跨; straddling) 일이 있기 때문이다.
- [0648] [본 발명에 따른 화상 신호의 일그러짐 보정 장치의 제2 실시형태]
- [0649] 상술한 화상 신호의 일그러짐 보정 장치의 제1 실시형태로서의 활상 장치에서의 손흔들림 움직임 벡터 검출부(15)에서는, 도 30에 도시한 바와 같이, 화상 메모리부(4)는, 2매의 화상, 다시 말해 원프레임의 화상과 참조 프레임의 화상이 양쪽 모두, 프레임 메모리(42, 42)에 격납되어 있는 것을 전제로 하고 있었다. 이 때문에, 움직임 벡터의 검출 타이밍은, 1프레임분 지연 되게 된다.
- [0650] 이것에 대해서, 이 제2 실시형태에서는, 활상 소자(11)로부터의 방류(垂流; output; 출력) 화상 데이터를 참조 프레임으로 하는 구성으로 해서, 러스터 스캔(luster scan)의 스트림 데이터에 대해서, 리얼 타임으로 SAD값을 연산할 수 있도록 하고 있다.
- [0651] 도 53에, 이 제2 실시형태의 경우에서의 활상 장치의 구성 예의 블록도를 도시한다. 이 도 53으로부터 알 수 있는 바와 같이, 활상 신호 처리계(10)의 구성 블록 및 그 밖의 구성 블록은, 도 30에 도시한 제1 실시형태와 완전히 마찬가지이지만, 이 제2 실시형태에서는, 도 53에 도시하는 바와 같이, 화상 메모리부(4)는 하나의 프레임 메모리(43)로 이루어진다.
- [0652] 이 제2 실시형태에서는, 원프레임이 프레임 메모리(43)에 격납되어 있고, 참조 프레임은, 데이터 변환부(14)로부터 스트림으로 입력되어 오는 것으로 된다. 손흔들림 움직임 벡터 검출부(15)는, 제1 실시형태에서는, 2개의 프레임 메모리(41, 42)에 격납된 2매의 화상 데이터를 이용해서, 참조 블록에 대한 SAD값을 구하는 처리를 하도록 했다. 이것에 대해서, 이 제2 실시형태에서는, 도 53에 도시하는 바와 같이, 데이터 변환부(14)로부터의 스트림 화상 데이터를 참조 프레임의 화상 데이터로 함과 동시에, 프레임 메모리(43)에 격납되어 있는 화상 데이터를 원프레임의 화상 데이터로 해서, 참조 블록에 대한 SAD값을 구하도록 한다.
- [0653] 그리고, 일그러짐 보정 해상도 변환부(16)는, 프레임 메모리(43)로부터의 화상 데이터의 절출을, 손흔들림 움직임 벡터 검출부(15)에서 검출된 움직임 벡터에 의거해서 행함으로써, 손흔들림이 없는 화상 데이터를 출력하도록 하고 있다. 그 밖의 구성 및 동작은, 제1 실시형태와 마찬가지이다.
- [0654] 상술한 바와 같이, 이 제2 실시형태에서는, 데이터 변환부(14)로부터의 스트림 화상 데이터를 참조 프레임의 화상 데이터로 한다. 이 때문에, 어떤 입력 화소에 대해서, 이 화소를 요소로 하는 참조 블록이, 참조 프레임 위에 동시에 복수 존재하게 된다. 도 54는, 그것을 설명하기 위한 도면이다.
- [0655] 즉, 참조 프레임(102) 위의 서치 범위(105)에서의 입력 화소 Din은, 예를 들면 참조 벡터(1071)가 대응하는 참조 블록(1061)의 좌측에 위치하는 화소임과 동시에, 참조 벡터(1072)가 대응하는 참조 블록(1062)의 오른쪽 위에 위치하는 화소로 되어 있는 것을, 이 도 54로부터 알 수 있다.
- [0656] 따라서, 입력 화소 Din이 참조 블록(1061)에 속한다고 한 경우에는, 타겟 블록(103)의 화소 D1을 판독 출력해서, 그 차분을 산출할 필요가 있다. 또, 입력 화소 Din이 참조 블록(1062)에 속한다고 한 경우에는, 타겟 블록(103)의 화소 D2를 판독 출력해서, 그 차분을 산출할 필요가 있다.
- [0657] 도 54 및 후술하는 도 55에서는 간단하게 하기 위해서, 2개의 참조 블록만을 도시하고 있지만, 실제상(實際上)은, 입력 화소 Din을, 그 참조 블록 내의 화소로 하는 참조 블록은 다수로 된다.
- [0658] 이 제2 실시형태의 경우의 SAD 연산은, 입력 화소 Din의 휘도값 Y와, 각각의 참조 블록 내의 입력 화소 Din의 위치에 대응한, 타겟 블록 내의 화소의 휘도값 Y와의 차분 절대값을 산출하고, 그 산출한 차분 절대값을, 각각의 참조 블록에 대응한 참조 벡터에 따라서, SAD 테이블에 가산해 가도록 해서 행한다.
- [0659] 예를 들면, 입력 화소 Din이 참조 블록(1061)에 속한다고 한 경우에서의, 타겟 블록(103)의 화소 D1과 입력 화소 Din과의 차분 절대값은, 도 55에 도시하는 바와 같이, SAD 테이블(108)의 참조 벡터(1071)이 대응하는 SAD 테이블 요소(1092)의 SAD값에 가산해서 기입하도록 한다. 또, 입력 화소 Din이 참조 블록(1062)에 속한다고 한 경우에서의, 타겟 블록(103)의 화소 D2와 입력 화소 Din과의 차분 절대값은, 도 55에 도시하는 바와 같이, SAD 테이블(108)의 참조 벡터(1072)가 대응하는 SAD 테이블 요소(1092)의 SAD값에 가산해서 기입하도록 한다.
- [0660] 따라서, 서치 범위 내의 모든 영역의 입력 화소가 입력되어 처리가 종료했을 때에는, SAD 테이블이 완성하게 된다.
- [0661] 도 55의 설명은, 종래 수법에, 리얼타임 SAD 산출 처리를 적용한 경우이다. 이 제2 실시형태에서는, 도

55에서, SAD 테이블(108)의 참조 벡터(1071 또는 1072)가 대응하는 SAD 테이블 요소(1091 또는 1092)의 SAD값으로서 산출한 차분 절대값의 각각을, 가산해서 기입하는 것이 아니라, 전술한 제1 실시형태와 같이, 참조 벡터(1071, 1072)를 축소 배율 $1/n$ 로 축소한 참조 축소 벡터를 산출하고, 그 참조 축소 벡터 근방의 복수의 참조 벡터에, 상기 산출한 차분 절대값으로부터, 각각을 분산 가산하기 위한 분산 가산값을 구하고, 구한 분산 가산값을, 상기 근방의 복수의 참조 벡터에 대응하는 SAD값에 가산하도록 하는 것이다.

[0662] SAD 테이블(축소 SAD 테이블)이 완성된 후의 정확한 움직임 벡터를 검출하기 위한 처리는, 이 제2 실시형태에서도, 전술한 제1 실시형태에서 기술한 수법과 완전히 마찬가지로 해서, 2차 곡면이나, 수평 방향 및 수직 방향의 3차 곡선을 이용한 수법을 이용할 수가 있다.

[0663] 이 제2 실시형태에서의 순흔들림 움직임 벡터 검출부(15)에서의 움직임 벡터의 검출 처리 동작의 플로차트를 도 56 및 도 57에 도시한다.

[0664] 우선, 순흔들림 움직임 벡터 검출부(15)에서는, 입력 화상의 프레임(참조 프레임)의 임의의 위치 (x, y) 의 화소 데이터 $Din(x, y)$ 를 수취한다(스텝 S401). 다음에, 해당 화소의 위치 (x, y) 를 포함하는 복수의 참조 블록의 하나에 대응하는 참조 벡터 (vx, vy) 를 설정한다(스텝 S402).

[0665] 다음에, 설정된 참조 벡터 (vx, vy) 의 참조 블록 Ii 의 해당 화소값 $Ii(x, y)$ 와, 이것에 대응하는 타겟 블록 Io 내의 화소값 $Io(x-vx, y-vy)$ 와의 차분의 절대값 a 를 산출한다(스텝 S403). 즉, 차분 절대값 a 는,

$$a = | Io(x-vx, y-vy) - Ii(x, y) | \quad \dots(\text{식 } 3)$$

[0666]으로서 산출된다.

[0667] 다음에, 축소 배율을 $1/n$ 로 해서, 참조 벡터 (vx, vy) 를 $1/n$ 로 축소한 참조 축소 벡터 $(vx/n, vy/n)$ 를 산출한다(스텝 S404).

[0668] 그 다음에, 참조 축소 벡터 $(vx/n, vy/n)$ 근방의 복수의 참조 벡터, 이 예에서는, 상술한 바와 같이 4개의 근방 참조 벡터를 검지한다(스텝 S405). 그리고, 검지한 4개의 근방 참조 벡터의 각각에 대응하는 테이블 요소로서 분산 가산해야 할 값(차분 절대값)을, 전술한 바와 같이, 참조 축소 벡터와 근방 참조 벡터가 각각 나타내는 위치의 관계에 의거해서, 스텝 S403에서 구한 차분 절대값 a 로부터, 선형 가중 분산값으로서 구한다(스텝 S406). 그리고, 구한 4개의 선형 가중 분산값을, 근방 참조 벡터의 각각에 대응하는 SAD 테이블 요소값에 가산한다(스텝 S407).

[0669] 다음에, 입력 화소 $Din(x, y)$ 를 포함하는 참조 블록의 모두에 대한 상기 스텝 S402~스텝 S407의 연산을 행했는지 여부 판별하고(스텝 S408), 해당 입력 화소 $Din(x, y)$ 를 포함하는 다른 참조 블록이 있다고 판별했을 때에는, 스텝 S402로 되돌아가고, 해당 입력 화소 Din 을 포함하는 다른 참조 블록 (vx, vy) 를 설정하고, 이 스텝 S402~스텝 S407의 처리를 되풀이한다.

[0670] 또, 스텝 S408에서, 입력 화소 $Din(x, y)$ 를 포함하는 참조 블록 모두에 대한 상기 스텝 S402~스텝 S407의 연산을 행했다고 판별했을 때에는, 서치 범위 내의 모든 입력 화소 Din 에 대해서, 상기의 연산 스텝의 처리를 종료했는지 여부를 판별하고(도 57의 스텝 S411), 종료하고 있지 않다고 판별했을 때에는, 스텝 S401로 되돌아가고, 서치 범위 내의 다음의 입력 화소 Din 을 취입하고, 이 스텝 S401 이후의 처리를 되풀이한다.

[0671] 그리고, 스텝 S411에서, 서치 범위 내의 모든 입력 화소 Din 에 대해서, 상기의 연산 스텝의 처리를 종료했다고 판별하면, 축소 SAD 테이블이 완성했다고 해서, 해당 완성한 축소 SAD 테이블에서, 최소값으로 되어 있는 SAD값을 검출한다(스텝 S412).

[0672] 다음에, 해당 최소값으로 되어 있는 테이블 요소 어드레스 (mx, my) 의 SAD값(최소값)과, 그 근방의 복수개, 이 예에서는, 상술한 바와 같이 15개의 근방 테이블 요소의 SAD값을 이용해서 2차 곡면을 생성하고(스텝 S413), 그 2차 곡면의 최소값의 SAD값이 대응하는 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py) 를 산출한다(스텝 S414). 이 최소값 벡터 (px, py) 는, 소수 정밀도의 최소 테이블 요소 어드레스에 대응하고 있다.

[0673] 그리고, 산출한 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py) 를 n 배하는 것에 의해, 구해야 할 움직임 벡터 $(px \times n, py \times n)$ 을 산출한다(스텝 S415).

[0674] 또한, 이 예에서도, 소수 정밀도의 위치를 나타내는 최소값 벡터 (px, py) 를 산출하는 방법으로서는, 전술한 수평 방향 및 수직 방향의 3차 곡선을 이용하는 방법을 이용해도 좋은 것은 전술한 예와 마찬가지이다.

[0676] 또, 전술한 제1 실시형태의 제3 예와 마찬가지로 해서, 이 제2 실시형태에서도, 서치 범위를 줄이면서(絞; 좁히면서), 또한 필요에 따라서 축소 배율을 변경하면서, 2 단계 이상, 축소 SAD 테이블을 이용한 움직임 벡터 검출 처리를 되풀이하도록 해도, 물론 좋다.

[0677] 이 제2 실시형태의 메리트는, 프레임 메모리를, 제1 실시형태에 비해서 1매분 삭감할 수 있는 것과, 프레임 메모리에 입력 화상을 격납하는 시간을 단축할 수 있는 것이다. 메모리 삭감의 효과는 말할 필요도 없지만, 처리 시간의 단축도, 근년에, 중요시되어 오고 있다. 특히, 동화상을 취급하는 경우, 그대로 시스템 지연의 단축으로 이어지기 때문에, 시스템 지연이 원인으로 생기는, 실제의 피사체와 패널 표시 화상 사이에 생기는 위화감을 가능한 없애는 것은, 유저에의 소구(訴求; solicit; 손님끌기, 선전) 효과가 높다.

[0678] [그 밖의 실시형태 및 변형예]

[0679] 상술한 실시형태에서는, 분할 화상 구간의 각각에서는, 수평 방향으로 영역을 복수개로 분할해서 각각의 분할 영역에 타겟 블록을 설정하고, 각각의 타겟 블록에 대해서 검출한 움직임 벡터의 평균화 처리에 의해, 각각의 분할 화상 구간의 손흔들림 움직임 벡터를 구하도록 했지만, 분할 화상 구간의 각각에 대해서 하나의 타겟 블록을 설정해서, 하나의 움직임 벡터를 산출하도록 해도 물론 좋다.

[0680] 또, 상술한 실시형태에서는, 수평 보정 속도 성분 X_{STB_*} 및 수직 보정 속도 성분 Y_{STB_*} 는, 1수평 라인 구간의 시간 길이당의 손흔들림 보정량으로서, 수평 손흔들림 보정량 SX_ADD 및 수직 손흔들림 보정량 SY_ADD 의 시간 적분 연산을, 수평 보정 속도 성분 X_{STB_*} 및 수직 보정 속도 성분 Y_{STB_*} 의 단순 가산으로 하도록 했지만, 각각의 검출한 속도 성분과 각 라인의 시간을 승산하도록 해도 좋은 것은 말할 필요도 없다.

[0681] 또, 상술한 실시형태의 설명은, 활상 장치를 조작하는 유저에 의해 일어나는(생기는) 손흔들림에 의한 화상 일그러짐을 보정하는 경우였지만, 유저에 의해 일어나는 손흔들림만이 아니라, 촬영시에, 활상 소자에 대해서, 그 활상 화상의 수평 방향 및/또는 수직 방향의 위치적 변화를 일으키도록 가해진 진동 등의 편의력(偏倚力; bias force)에 기인해서 생기는 화상 일그러짐도 보정할 수 있는 것은 말할 필요도 없다.

[0682] 또, 상술한 실시형태는, X-Y 어드레스형 고체 활상 소자로서 CMOS 이미저를 이용한 경우이지만, 활상 소자로서는 CMOS 이미저에 한정되는 것이 아닌 것은 물론이다.

[0683] 또, 본 발명은, 활상 장치(카메라)에만 적용되는 것은 아니며, 예를 들면 휴대 전화 단말, 정보 휴대 단말 등에 활상 소자가 부착(取付; provide)되어, 화상을 촬영하는 경우에도 적용할 수 있는 것이다. 또, 유저가 손으로 들고서 촬영을 행하는 장치에만 적용되는 것은 아니며, 퍼스널 컴퓨터나 화상 전화 장치 등의 고정적으로 설치되는 장치에 대해서 외력이 가해져서 진동 등이 생기는 경우, 또 자동차 등에 활상 소자가 부착되어, 화상을 촬영하는 경우에도 적용가능하다.

발명의 효과

[0684] 본 발명에 따르면, CMOS 이미저 등의 X-Y 어드레스형 활상 소자를 이용한 활상 장치에 있어서, 자이로 센서 등의 기구계 부품을 사용하는 일없이, 디지털 신호 처리에 의해 고정밀도로 또한 저코스트로, 손흔들림 등이 원인으로 생기는 포컬 플레인 현상에 의한 활상 화상의 일그러짐을 경감할 수가 있다.

도면의 간단한 설명

[0001] 도 1은 본 발명에 따른 화상 일그러짐 보정 방법의 실시형태의 개요를 설명하기 위한 도면,

[0002] 도 2는 본 발명에 따른 화상 일그러짐 보정 방법의 실시형태의 개요의 요부(要部)를 설명하기 위한 도면,

[0003] 도 3은 블록 매칭에 의해 움직임 벡터를 검출하는 처리를 설명하기 위한 도면,

[0004] 도 4는 블록 매칭에 의해 움직임 벡터를 검출하는 처리를 설명하기 위한 도면,

[0005] 도 5는 블록 매칭에 의해 움직임 벡터를 검출하는 처리를 설명하기 위한 도면,

[0006] 도 6은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 처리의 개요를 설명하기 위한 도면,

[0007] 도 7은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 처리의 개요를 설명하기 위한 도면,

[0008] 도 8은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 처리의 개요를 설명하기 위한 도면,

- [0009] 도 9는 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0010] 도 10은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0011] 도 11은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 처리의 개요를 설명하기 위한 도면,
- [0012] 도 12는 본 발명의 실시형태에서의 움직임 벡터를 검출하는 처리의 개요를 설명하기 위한 도면,
- [0013] 도 13은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0014] 도 14는 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0015] 도 15는 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0016] 도 16은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0017] 도 17은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0018] 도 18은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0019] 도 19는 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제1 예를 설명하기 위한 도면,
- [0020] 도 20은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제2 예를 설명하기 위한 도면,
- [0021] 도 21은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제2 예를 설명하기 위한 도면,
- [0022] 도 22는 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제2 예를 설명하기 위한 도면,
- [0023] 도 23은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법에서, 정확한 움직임 벡터를 검출하기 위한 처리의 제2 예를 설명하기 위한 도면,
- [0024] 도 24는 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법의 처리 성능을 설명하기 위한 도면,
- [0025] 도 25는 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법의 특징을, 종래의 수법과 비교하여 설명하기 위해서 이용하는 도면,
- [0026] 도 26은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법의 특징을, 종래의 수법과 비교하여 설명하기 위해서 이용하는 도면,
- [0027] 도 27은 본 발명의 실시형태에서의 움직임 벡터를 검출하는 방법의 특징을, 종래의 수법과 비교하여 설명하기 위해서 이용하는 도면,
- [0028] 도 28은 본 발명의 실시형태에서의 움직임 벡터의 검출 방법을 설명하기 위한 도면,
- [0029] 도 29는 본 발명의 실시형태에서의 움직임 벡터의 검출 처리의 요부를 설명하기 위한 도면,
- [0030] 도 30은 본 발명에 따른 화상 신호의 일그러짐 보정 방법의 실시형태가 적용된 활상 장치의 제1 실시형태의 구성을 도시하는 블록도,
- [0031] 도 31은 본 발명에 따른 활상 장치의 제1 실시형태에서의 움직임 벡터 검출 처리의 제1 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,

- [0032] 도 32는 본 발명에 따른 활상 장치의 제1 실시형태에서의 움직임 벡터 검출 처리의 제1 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,
- [0033] 도 33은 본 발명에 따른 활상 장치의 제1 실시형태에서의 움직임 벡터 검출 처리의 제2 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,
- [0034] 도 34는 본 발명에 따른 활상 장치의 제1 실시형태에서의 움직임 벡터 검출 처리의 제2 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,
- [0035] 도 35는 본 발명에 따른 활상 장치의 제1 실시형태에서의 움직임 벡터 검출 처리의 제2 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,
- [0036] 도 36은 본 발명에 따른 활상 장치의 제1 실시형태에서의 움직임 벡터 검출 처리의 제3 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,
- [0037] 도 37은 본 발명에 따른 활상 장치의 제1 실시형태에서의 움직임 벡터 검출 처리의 제3 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,
- [0038] 도 38은 본 발명에 따른 활상 장치의 제1 실시형태에서의 움직임 벡터 검출 처리의 제3 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,
- [0039] 도 39는 도 30의 활상 장치의 제1 실시형태의 일부의 블록의, 보다 상세 구성예의 블록도,
- [0040] 도 40은 도 30의 활상 장치의 제1 실시형태의 일부의 블록의, 보다 상세 구성예의 블록도,
- [0041] 도 41은 도 30의 활상 장치의 제1 실시형태의 일부의 블록의, 처리 동작예를 설명하기 위한 플로차트,
- [0042] 도 42는 본 발명에 따른 화상 일그러짐 보정 방법의 실시형태에서, 화상 일그러짐을 보정하기 위해서 이용하는 보간 처리를 설명하기 위한 도면,
- [0043] 도 43은 본 발명에 따른 화상 일그러짐 보정 방법의 실시형태에서, 화상 일그러짐을 보정하기 위해서 이용하는 보간 처리를 설명하기 위한 도면,
- [0044] 도 44는 본 발명에 따른 화상 일그러짐 보정 방법의 실시형태에서, 화상 일그러짐을 보정하기 위해서 보간 처리를 설명하기 위한 도면,
- [0045] 도 45는 실시형태의 활상 장치에서의 수평 손흔들림 보정량 적분부의 동작의 흐름을 설명하기 위한 플로차트,
- [0046] 도 46은 실시형태의 활상 장치에서의 수평 화상 처리부의 상세 구성예를 도시하는 블록도,
- [0047] 도 47은 도 46의 수평 화상 처리부의 동작 타이밍을 설명하기 위한 도면,
- [0048] 도 48은 실시형태의 활상 장치에서의 수직 손흔들림 보정량 적분부의 동작의 흐름을 설명하기 위한 플로차트,
- [0049] 도 49는 실시형태의 활상 장치에서의 수직 화상 처리부의 상세 구성예를 도시하는 블록도,
- [0050] 도 50은 도 49의 수직 화상 처리부의 동작 타이밍을 설명하기 위한 도면,
- [0051] 도 51은 도 30의 화상 일그러짐 보정 장치의 실시형태에서의 요부의 처리 동작을 설명하기 위한 블록도,
- [0052] 도 52는 도 30의 화상 일그러짐 보정 장치의 실시형태에서의 신호 처리부 내의 레지스터 블록에서의 레지스터 동작의 타이밍을 설명하기 위한 도면,
- [0053] 도 53은 본 발명에 따른 화상 일그러짐 보정 방법의 실시형태가 적용된 활상 장치의 제2 실시형태의 구성예를 도시하는 블록도,
- [0054] 도 54는 활상 장치의 제2 실시형태에서의 움직임 벡터 검출 처리를 설명하기 위한 도면,
- [0055] 도 55는 활상 장치의 제2 실시형태에서의 움직임 벡터 검출 처리를 설명하기 위한 도면,
- [0056] 도 56은 활상 장치의 제2 실시형태에서의 움직임 벡터 검출 처리의 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,
- [0057] 도 57은 활상 장치의 제2 실시형태에서의 움직임 벡터 검출 처리의 예를 설명하기 위한 플로차트의 일부를 도시하는 도면,

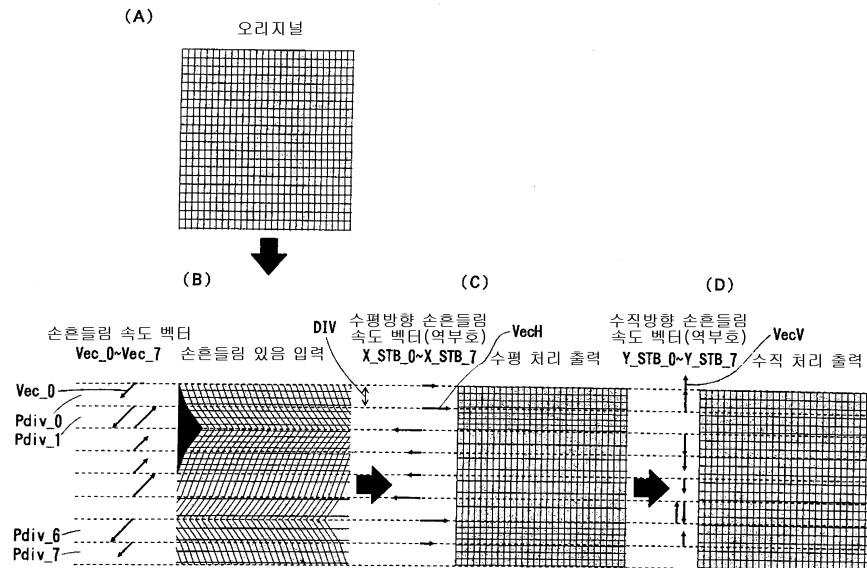
- [0058] 도 58은 종래의 손흔들림 보정을 설명하기 위해서 이용하는 도면,
- [0059] 도 59는 CMOS 이미저에서의, 손흔들림에 의한 프레임내 화상 일그러짐의 원인을 설명하기 위한 도면,
- [0060] 도 60은 CMOS 이미저에서의, 손흔들림에 의한 프레임내 화상 일그러짐을 설명하기 위해서 이용하는 도면,
- [0061] 도 61은 종래의 CMOS 이미저에 대한, 손흔들림에 의한 프레임내 화상 일그러짐의 보정 방법을 설명하기 위한 도면.

[부호의 설명]

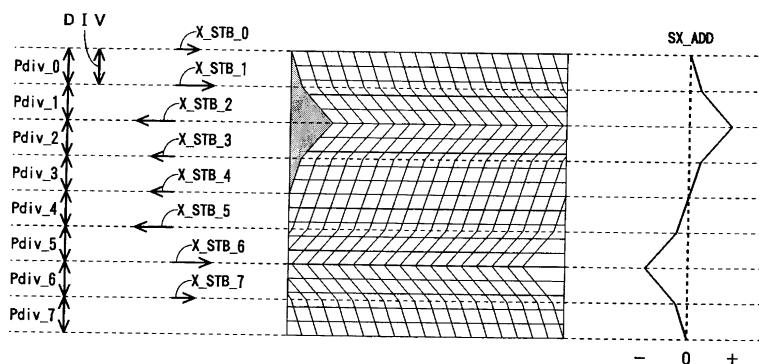
- [0063] 101…원(元)프레임, 102…참조 프레임, 103…타겟 블록, 105…서치 범위, 106…참조 블록, 107…참조 벡터, 15…손흔들림 움직임 벡터 검출부, 16…일그러짐 보정 해상도 변환부, 41~43…프레임 메모리, TBLs…축소 SAD 테이블, TBL0…종래의 SAD 테이블, RV…참조 벡터, CV…참조 축소 벡터, NV1~NV4…근방 참조 벡터.

도면

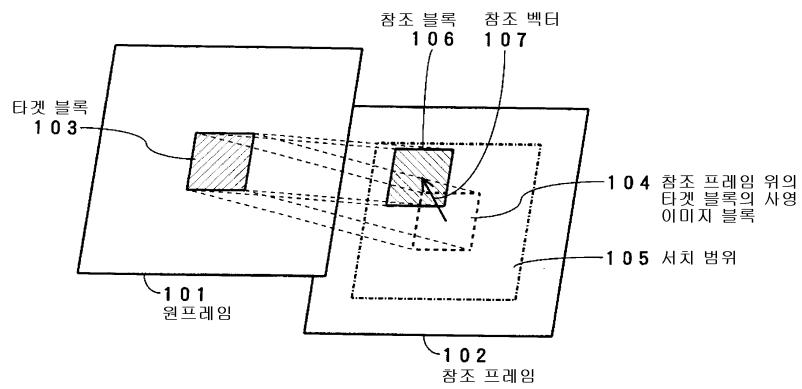
도면1



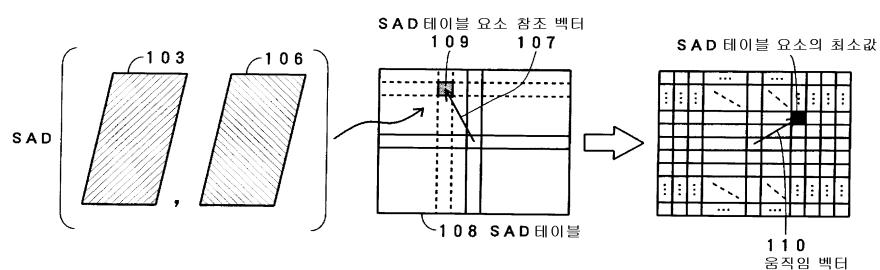
도면2



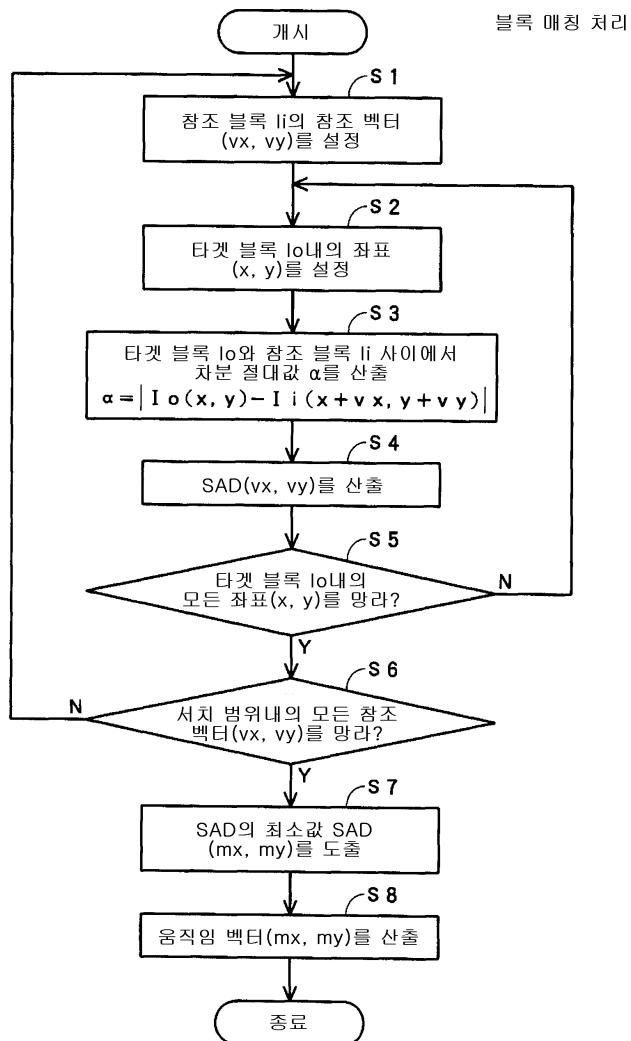
도면3



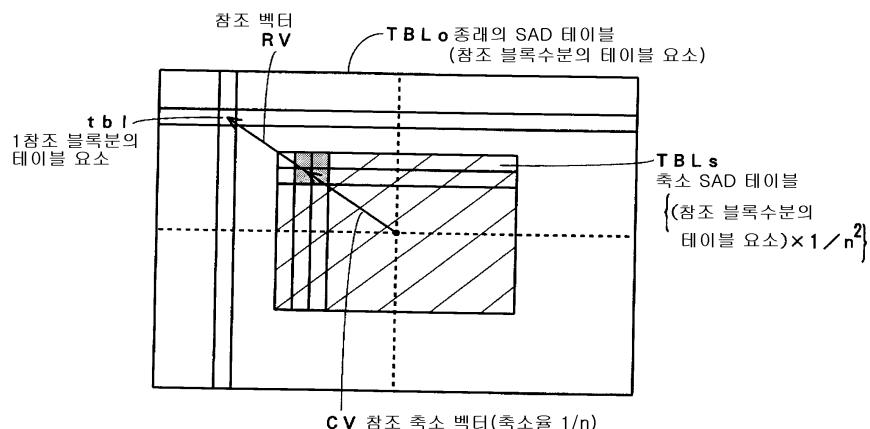
도면4



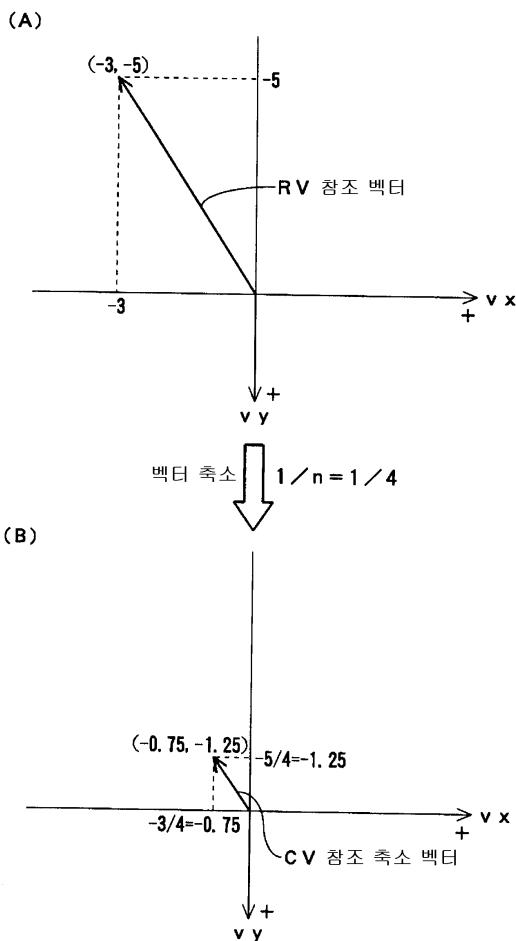
도면5



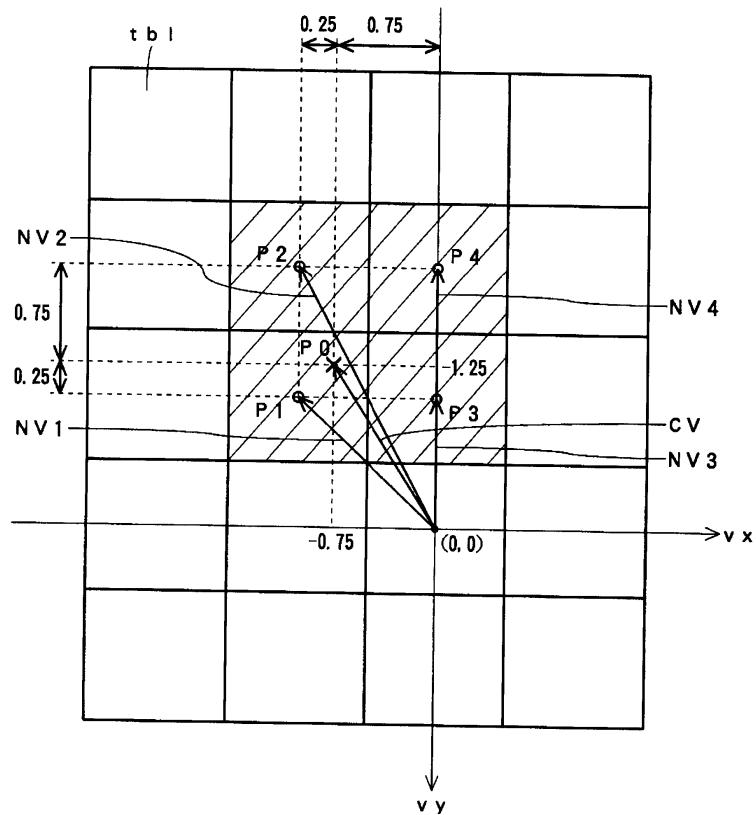
도면6



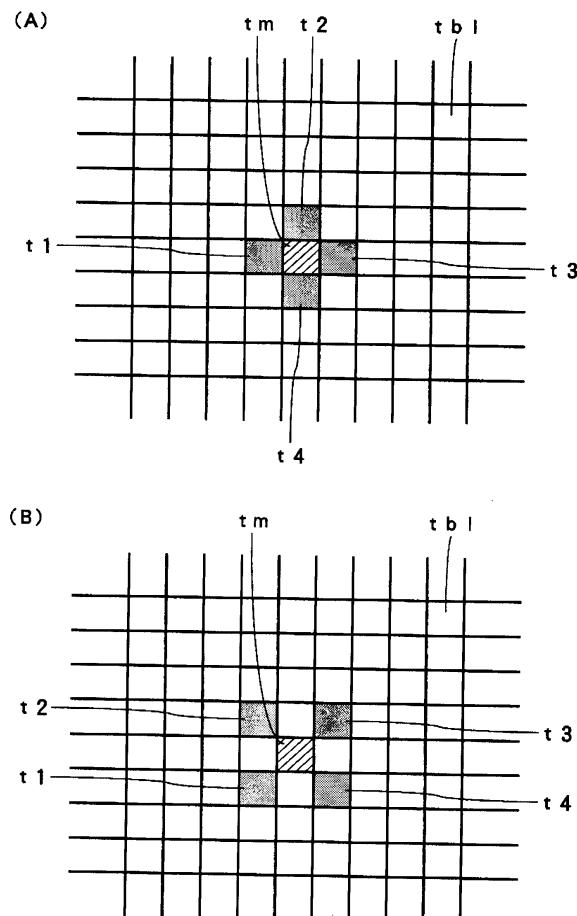
도면7



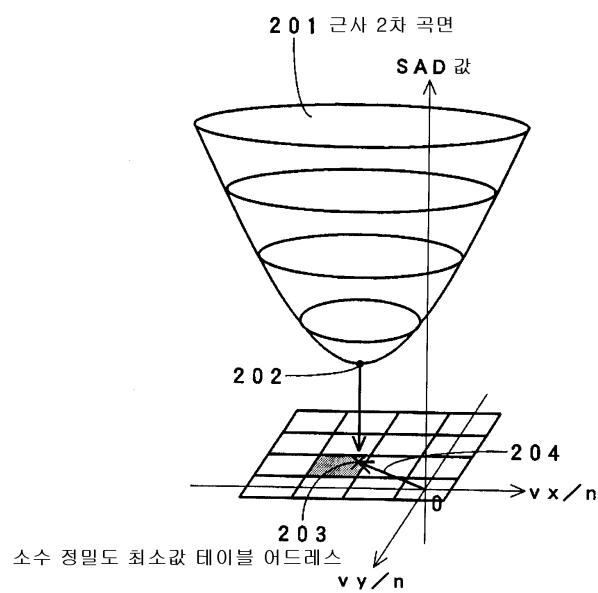
도면8



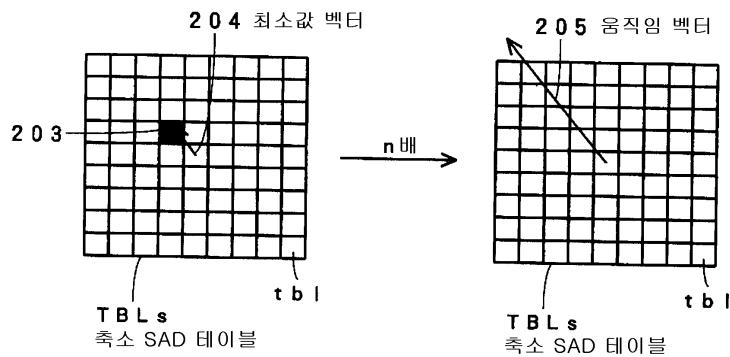
도면9



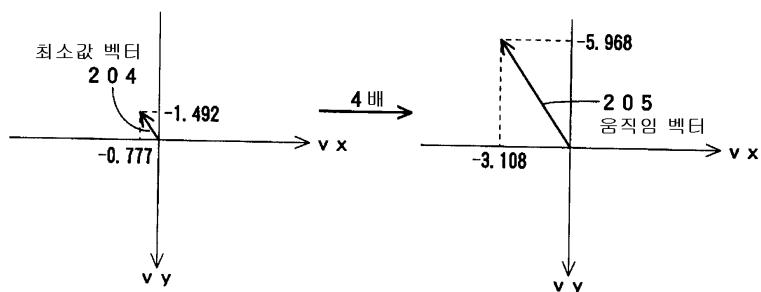
도면10



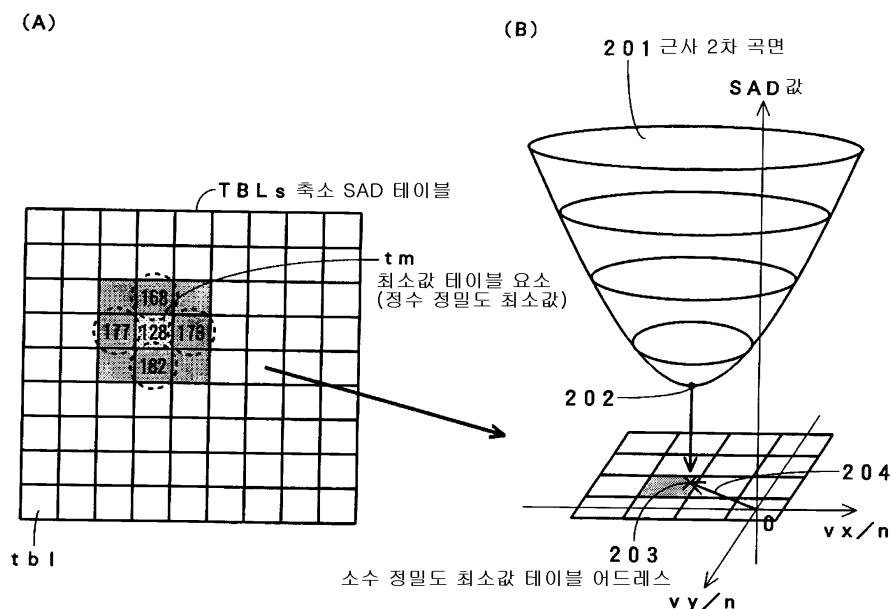
도면11



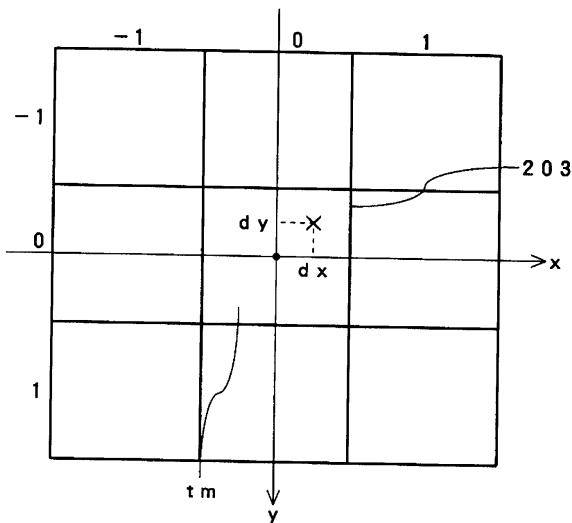
도면12



도면13



도면14



도면15

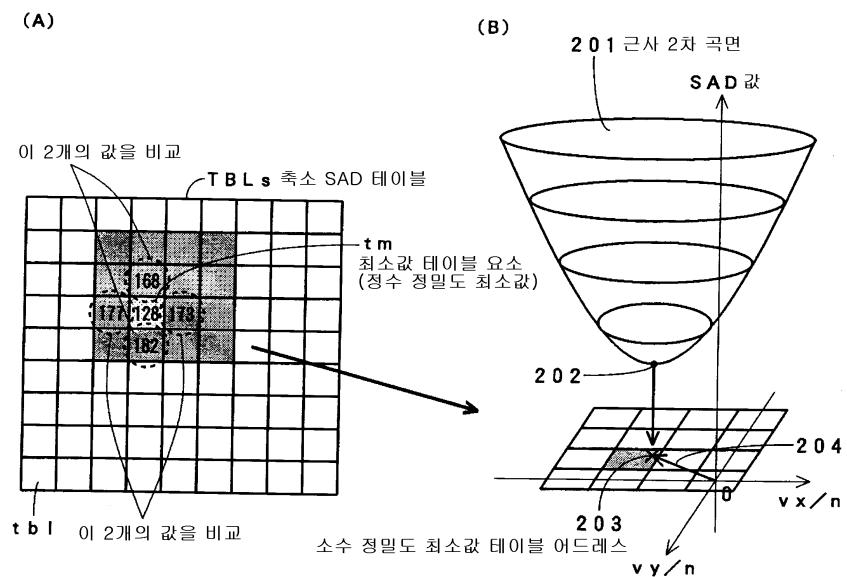
$$dx = \frac{\sum (Sxy + Kx)}{4 \sum Sxy - 6 \sum (Sxy + Kx^2)} \quad \dots \text{식 A}$$

$$dy = \frac{\sum (Sxy + Ky)}{4 \sum Sxy - 6 \sum (Sxy + Ky^2)} \quad \dots \text{식 B}$$

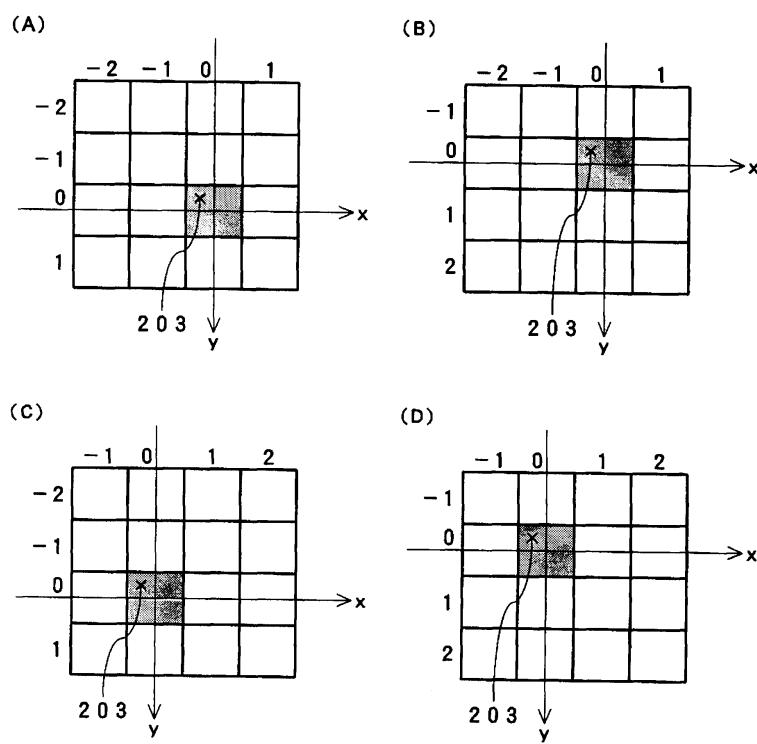
단,

| | |
|------------------------|------------------------|
| $x = -1$ 일 때 $Kx = -1$ | $y = -1$ 일 때 $Ky = -1$ |
| $x = 0$ 일 때 $Kx = 0$ | $y = 0$ 일 때 $Ky = 0$ |
| $x = 1$ 일 때 $Kx = 1$ | $y = 1$ 일 때 $Ky = 1$ |

도면16



도면17

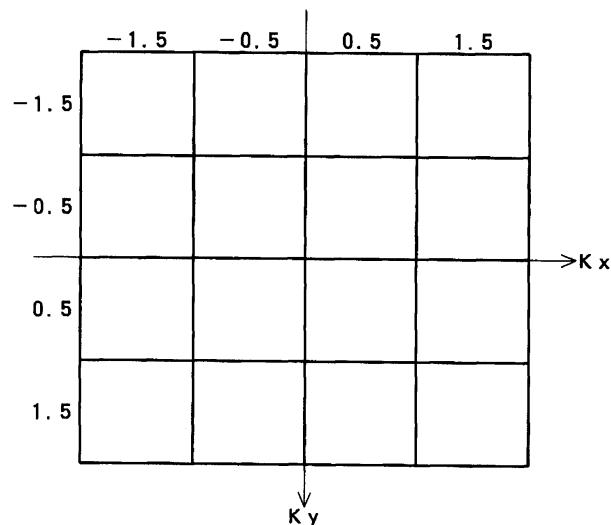


도면18

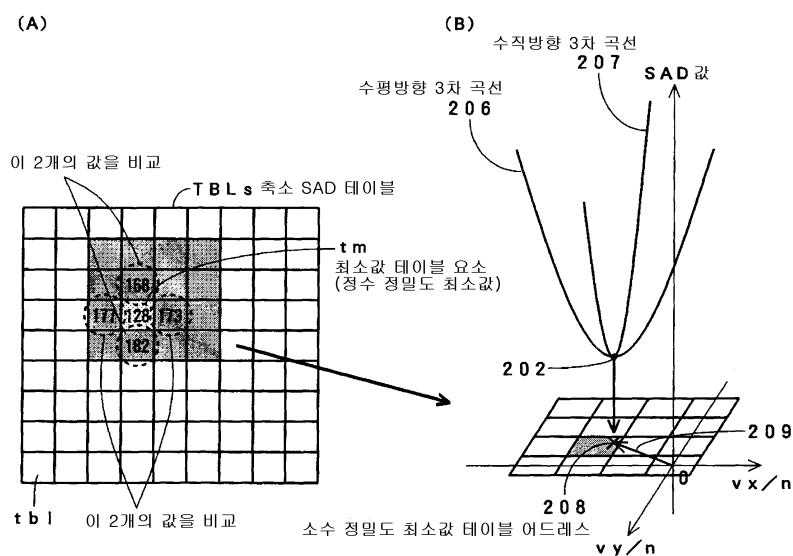
$$dx = \frac{8 \sum (Sxy + Kx)}{25 \sum Sxy - 20 \sum (Sxy + Kx^2)} + \Delta x \quad \dots \text{(식C)}$$

$$dy = \frac{8 \sum (Sxy + Ky)}{25 \sum Sxy - 20 \sum (Sxy + Ky^2)} + \Delta y \quad \dots \text{(식D)}$$

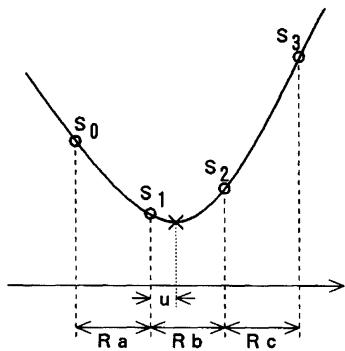
도면19



도면20



도면21



도면22

구간 R a

$$u = \frac{2s_3 - 11s_2 + 16s_1 - 7s_0 + \sqrt{(4s_3 - 7s_2 + 2s_1 + s_0)^2 + 15(s_3 - 3s_2 + 3s_1 - s_0)(s_1 - s_3)}}{3(s_3 - 3s_2 + 3s_1 - s_0)} \dots (\text{식 E})$$

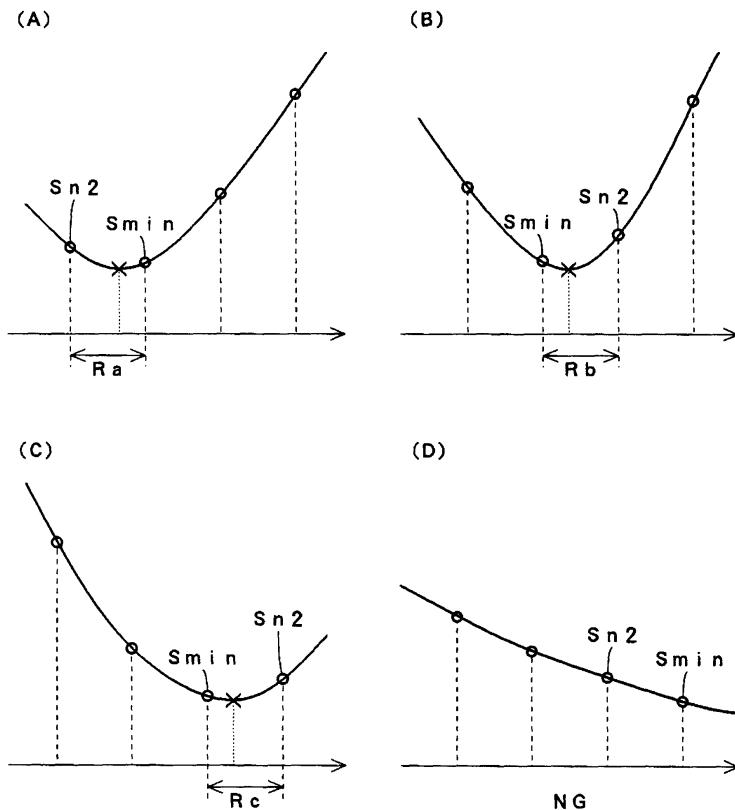
구간 R b

$$u = \frac{s_3 - 4s_2 + 5s_1 - 2s_0 + \sqrt{(s_3 - 5s_2 + 5s_1 - s_0)^2 + (s_2 - s_0)(s_1 - s_3)}}{3(s_3 - 3s_2 + 3s_1 - s_0)} \dots (\text{식 F})$$

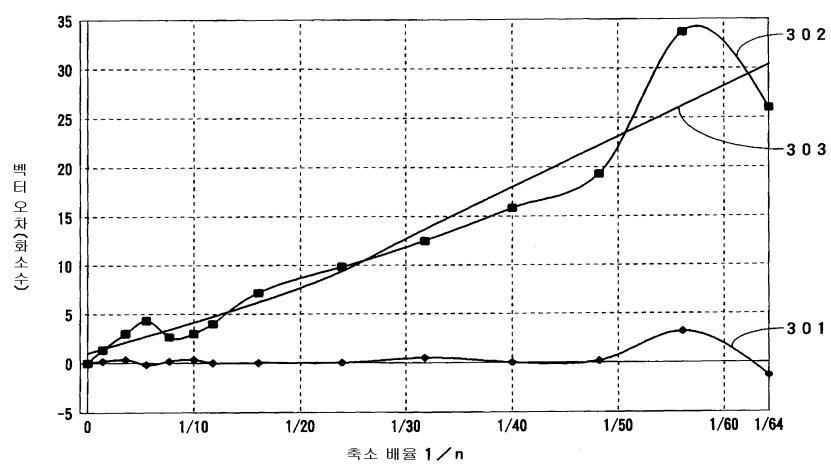
구간 R c

$$u = \frac{-4s_3 + 7s_2 - 2s_1 - s_0 + \sqrt{(4s_3 - 7s_2 + 2s_1 + s_0)^2 + 15(s_3 - 3s_2 + 3s_1 - s_0)(s_1 - s_3)}}{3(s_3 - 3s_2 + 3s_1 - s_0)} \dots (\text{식 G})$$

도면23



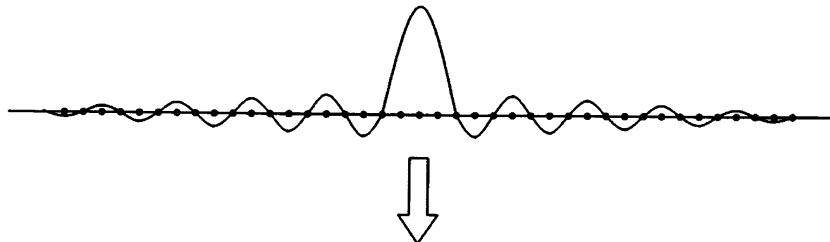
도면24



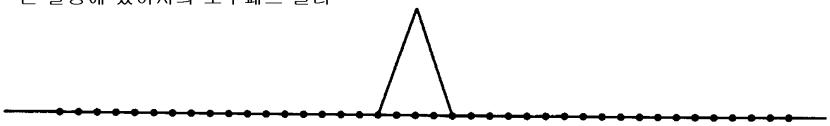
도면25

축소 배율 1/2

이상적인 로우패스 필터(sinc 함수)



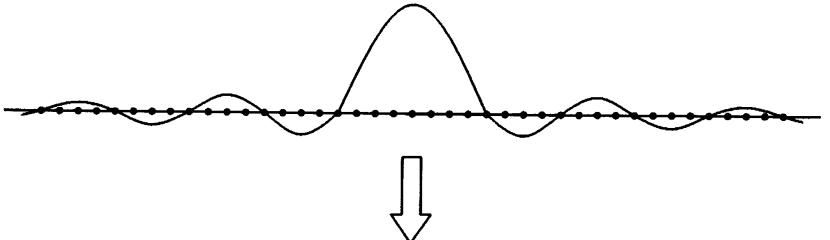
본 발명에 있어서의 로우패스 필터



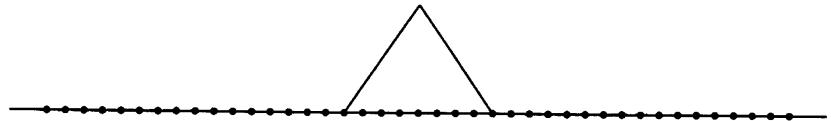
도면26

축소 배율 1/4

이상적인 로우패스 필터(sinc 함수)

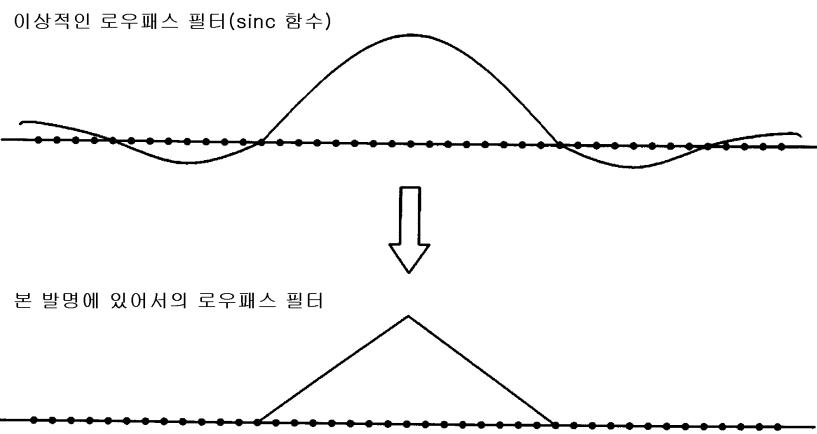


본 발명에 있어서의 로우패스 필터

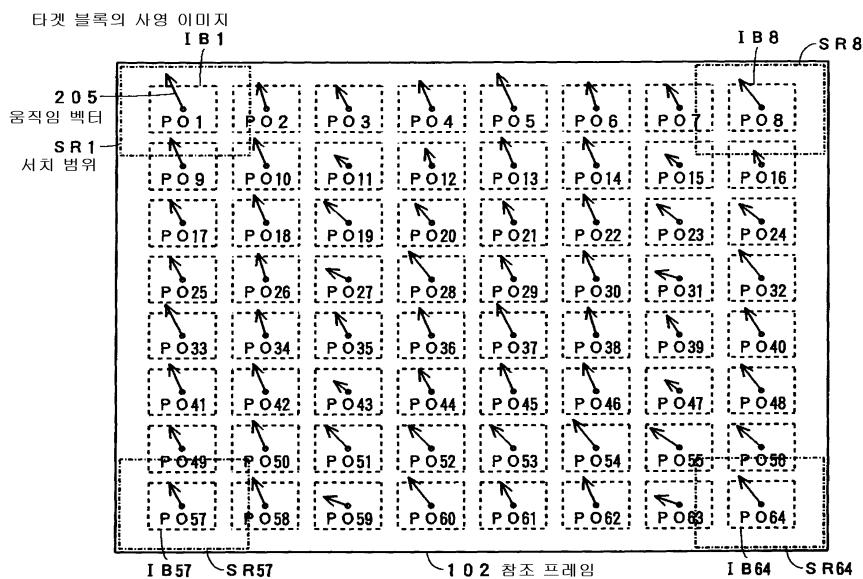


도면27

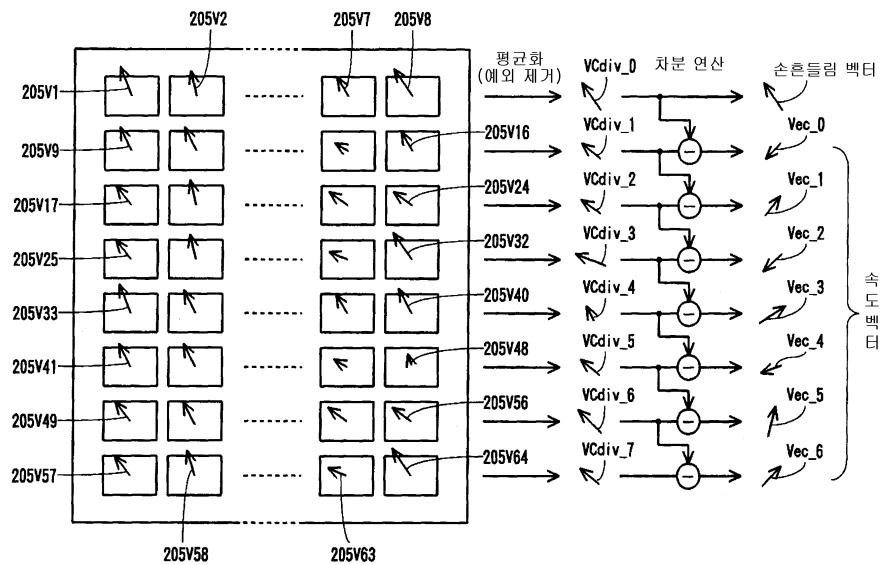
축소 배율 1/8



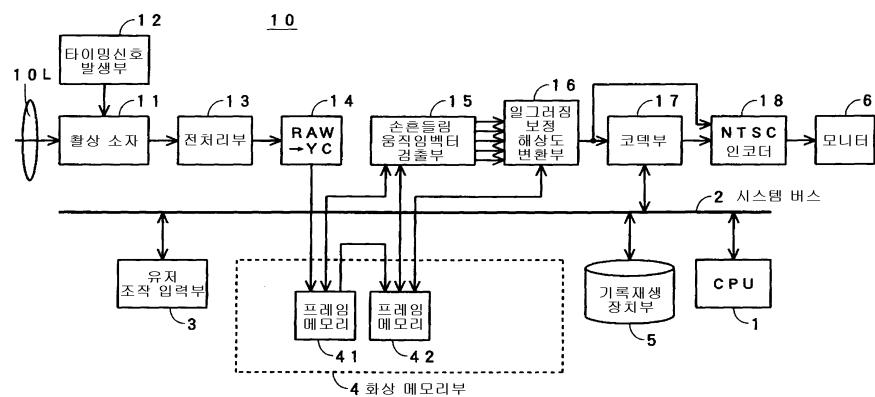
도면28



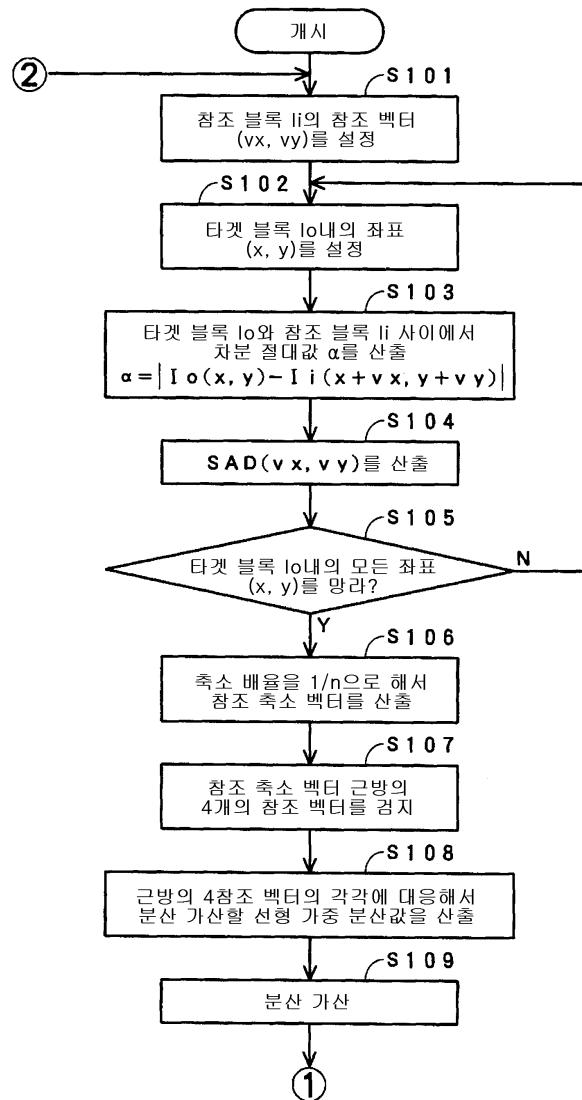
도면29



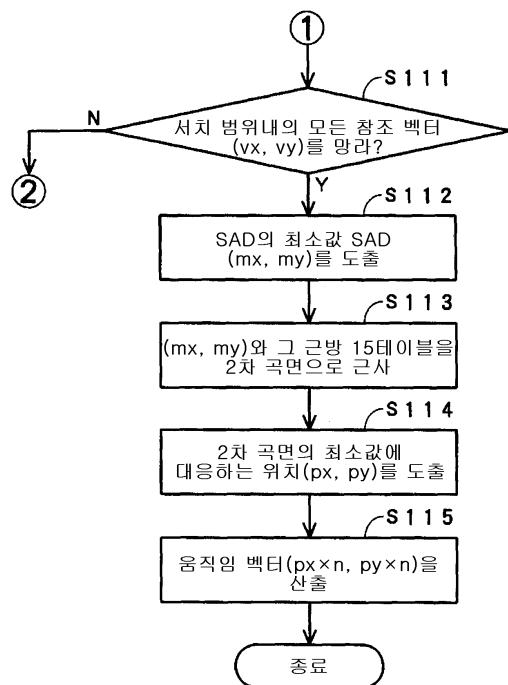
도면30



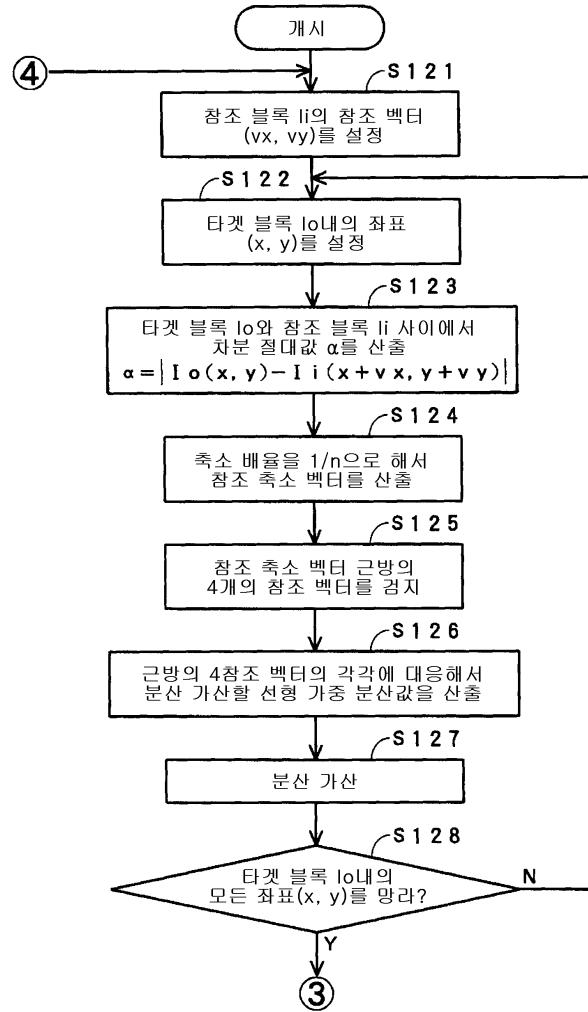
도면31



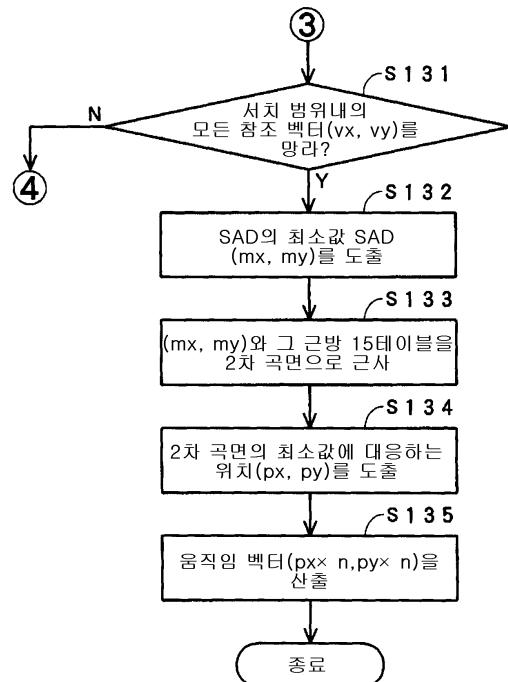
도면32



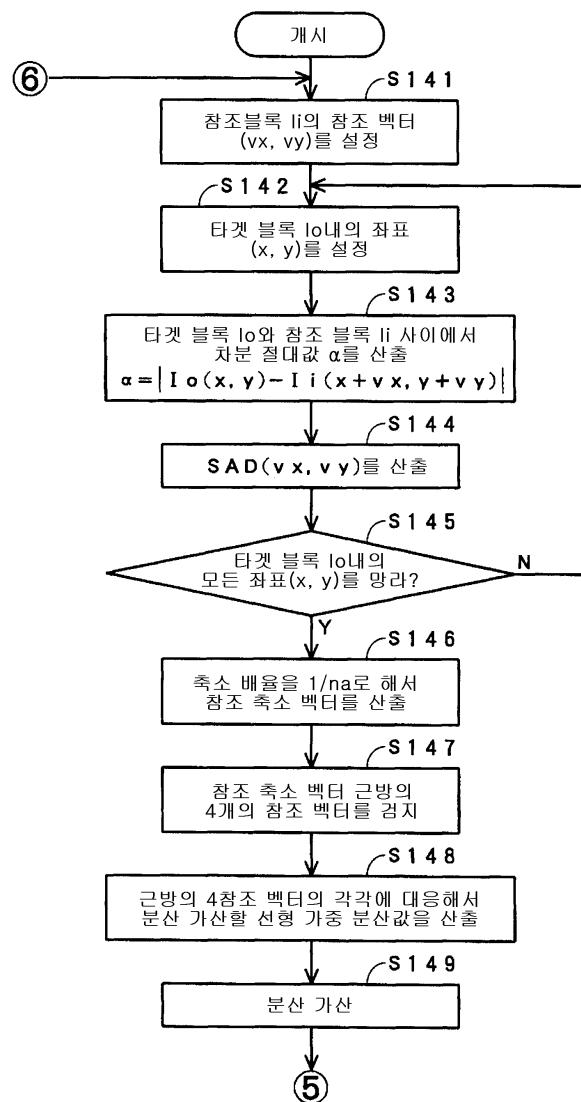
도면33



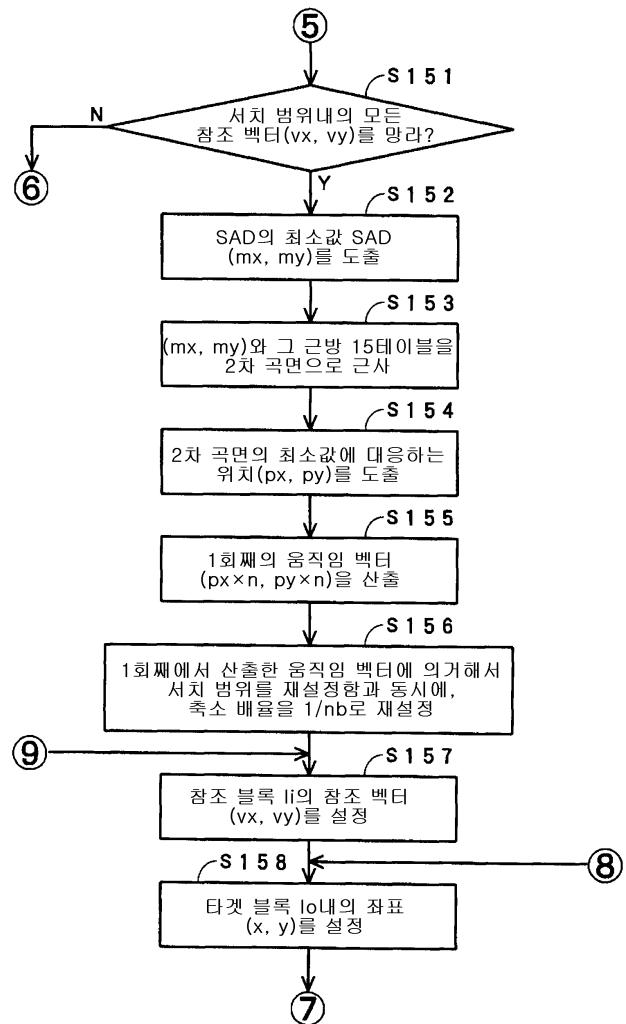
도면34



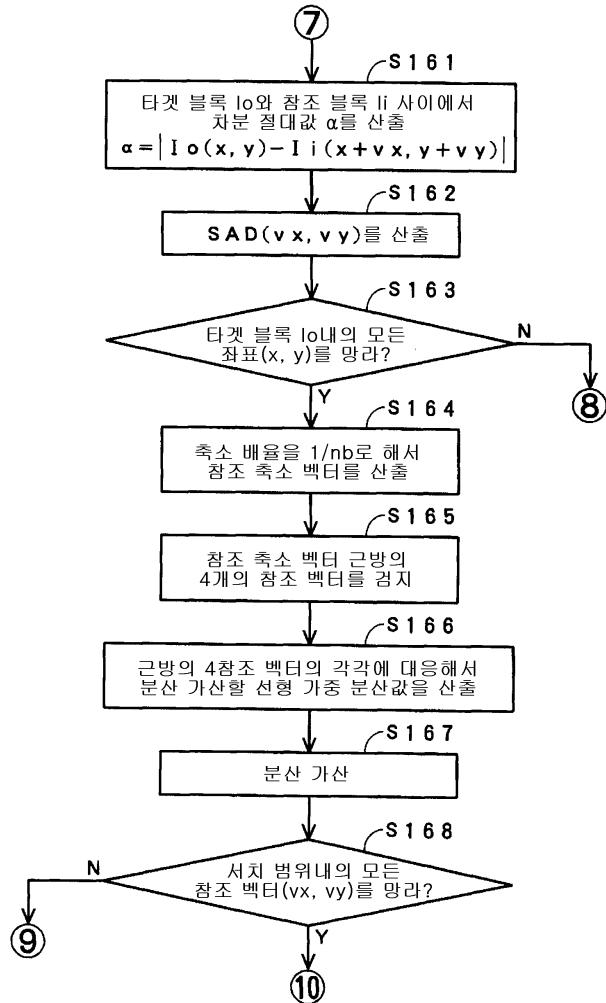
도면35



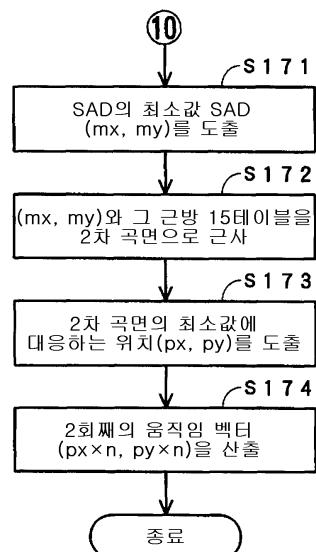
도면36



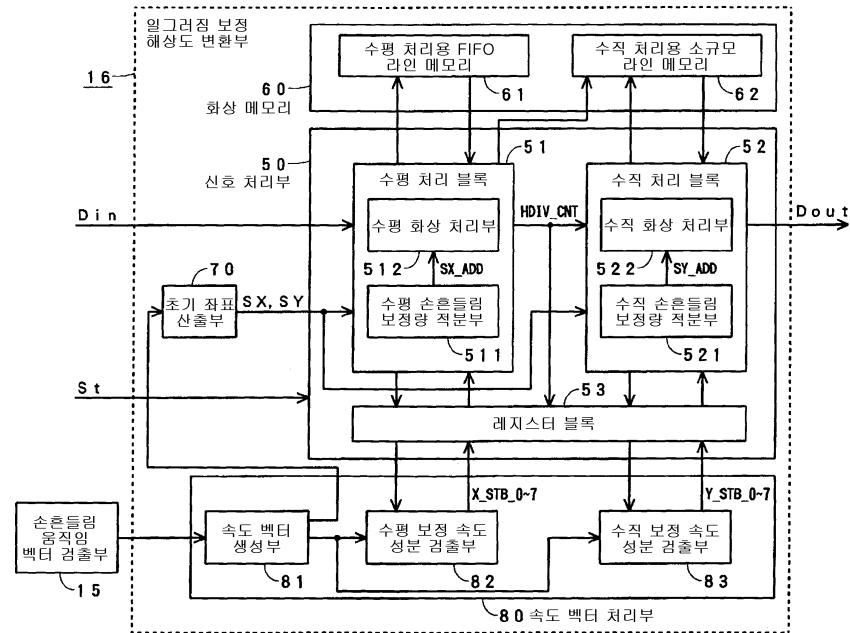
도면37



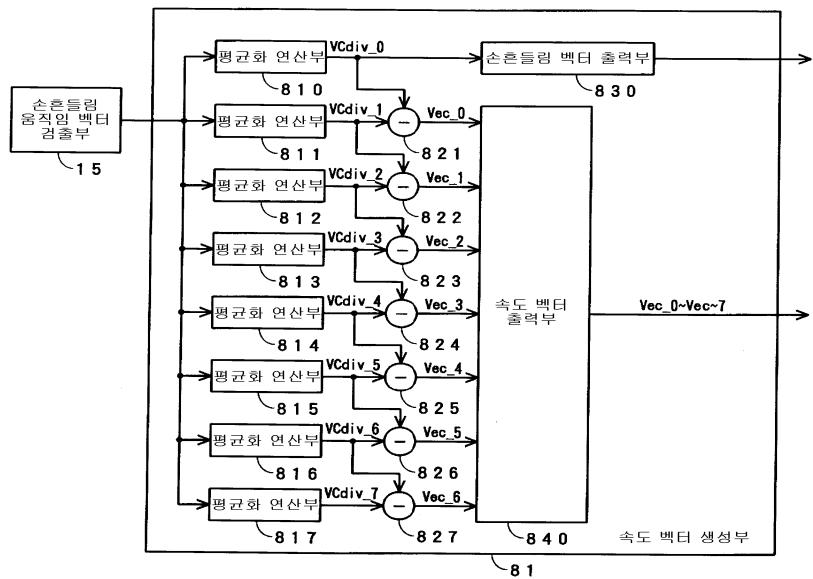
도면38



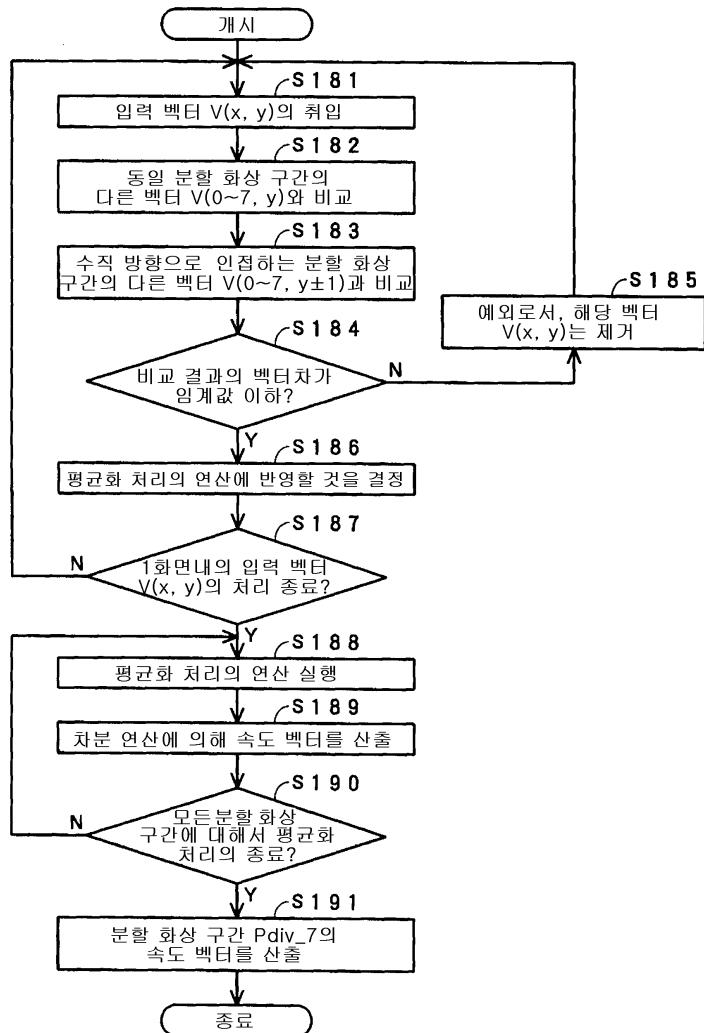
도면39



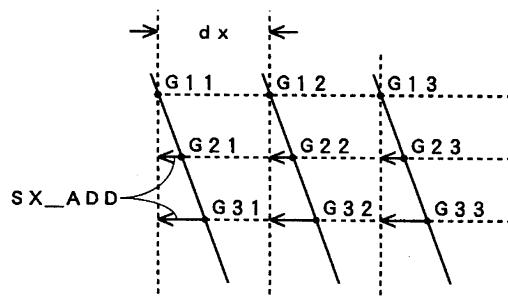
도면40



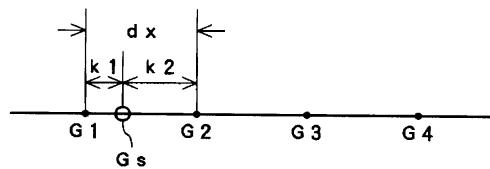
도면41



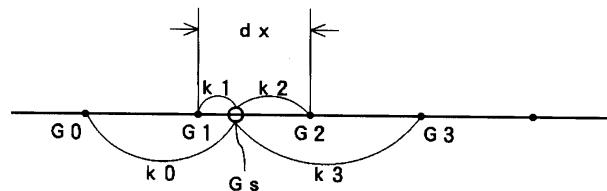
도면42



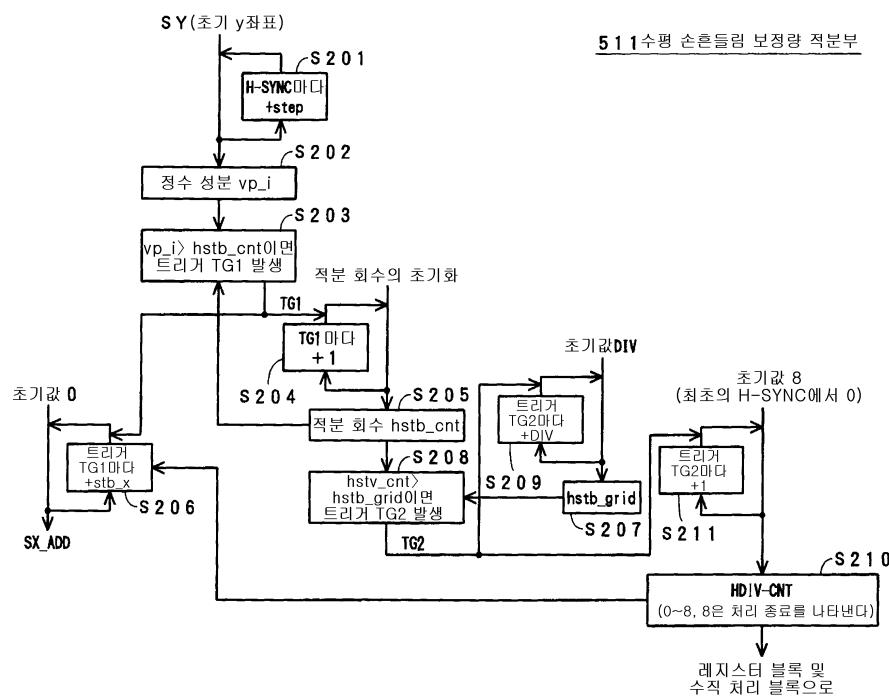
도면43



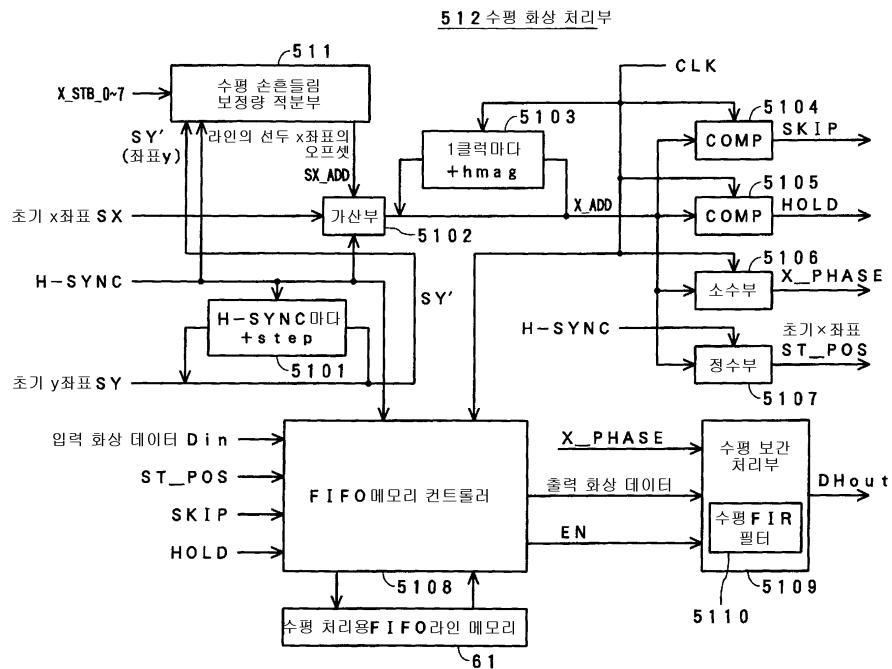
도면44



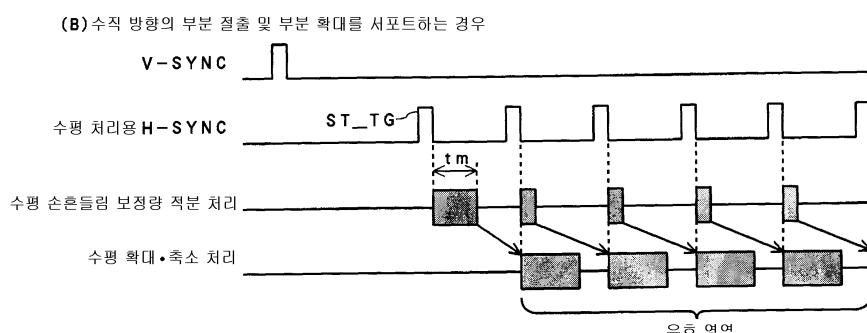
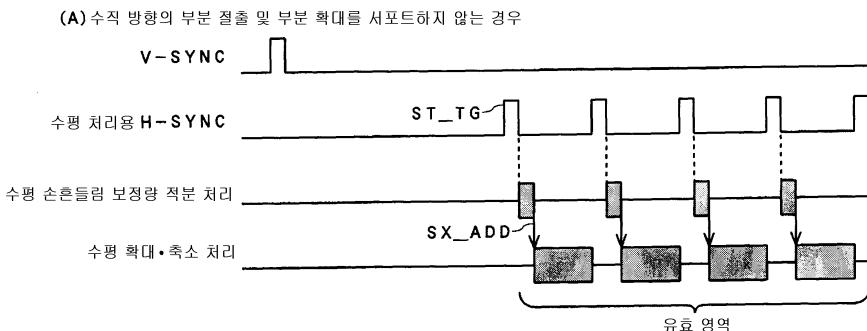
도면45



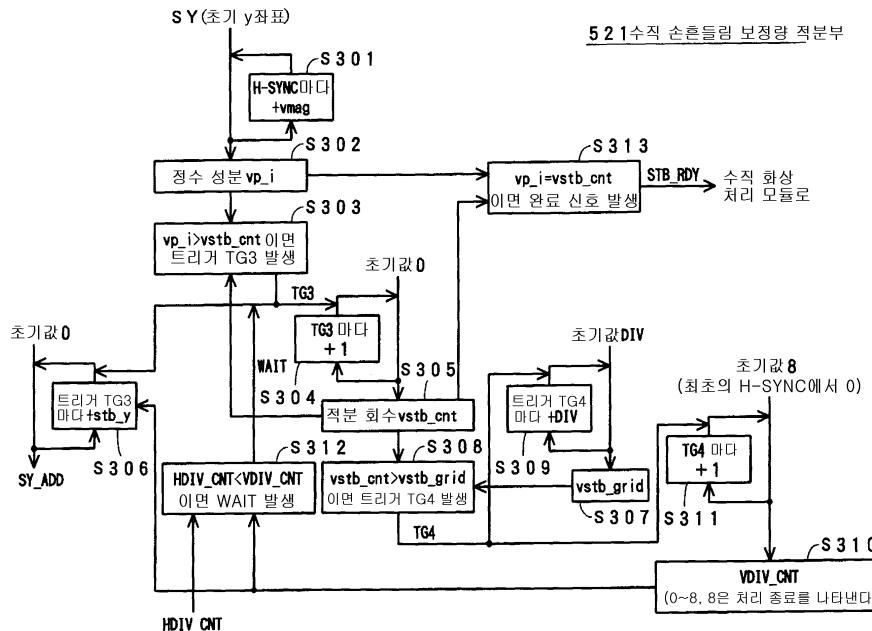
도면46



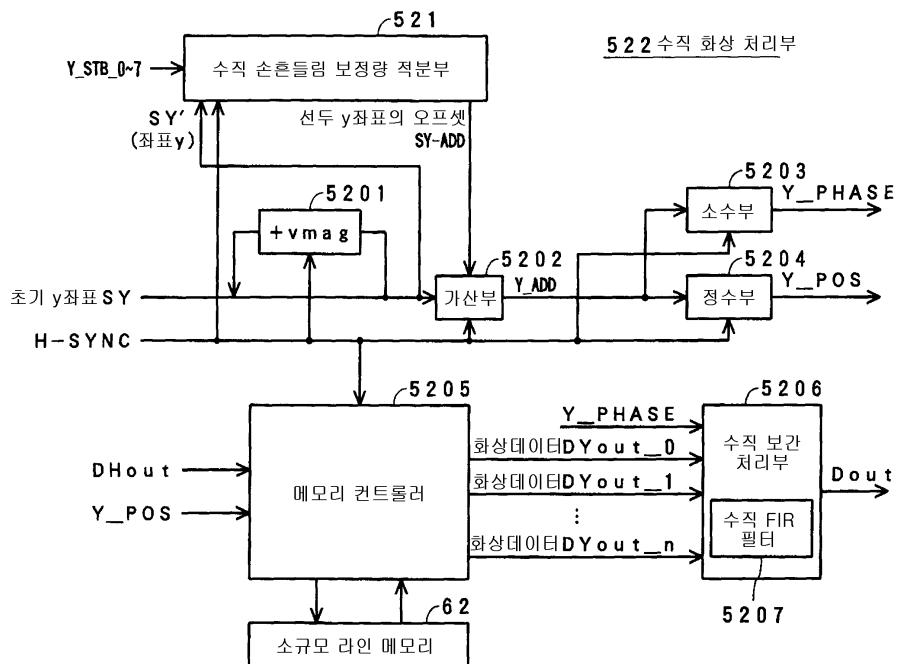
도면47



도면48

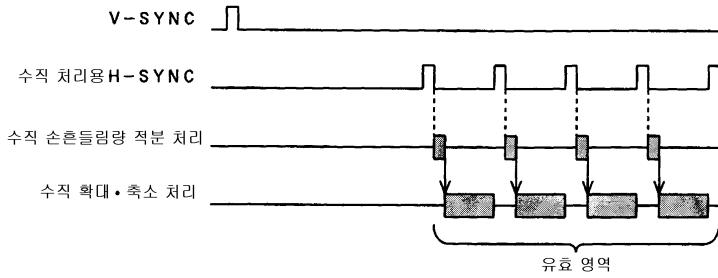


도면49

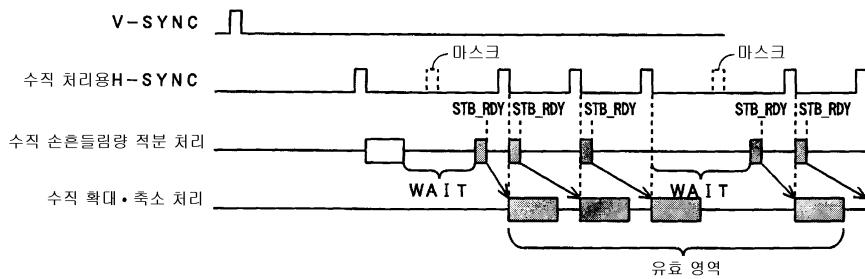


도면50

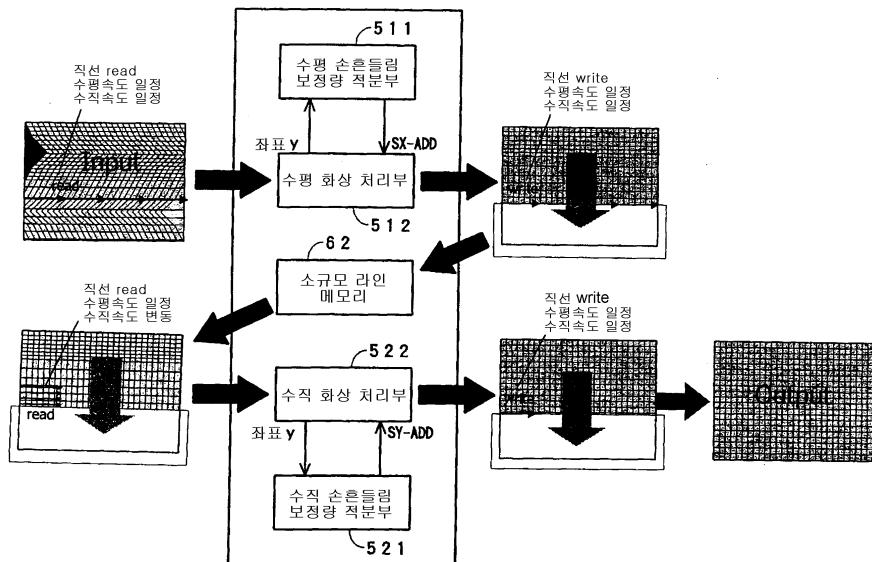
(A) 수직 방향의 부분 절출, 부분 확대, 축소의 어느것도 서포트하지 않는 경우



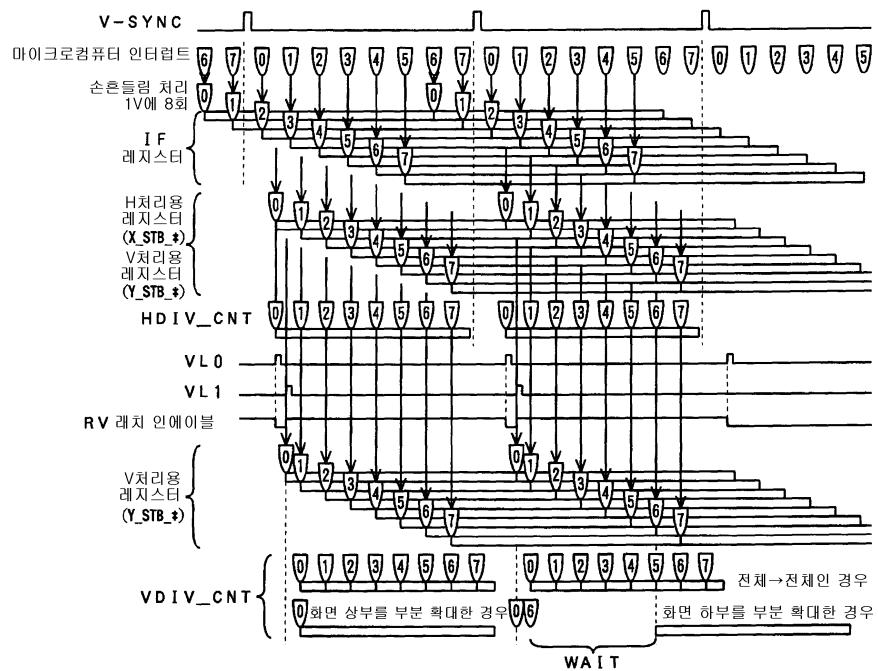
(B) 수직 방향의 부분 절출, 부분 확대, 축소의 어느것이나 서포트하는 경우



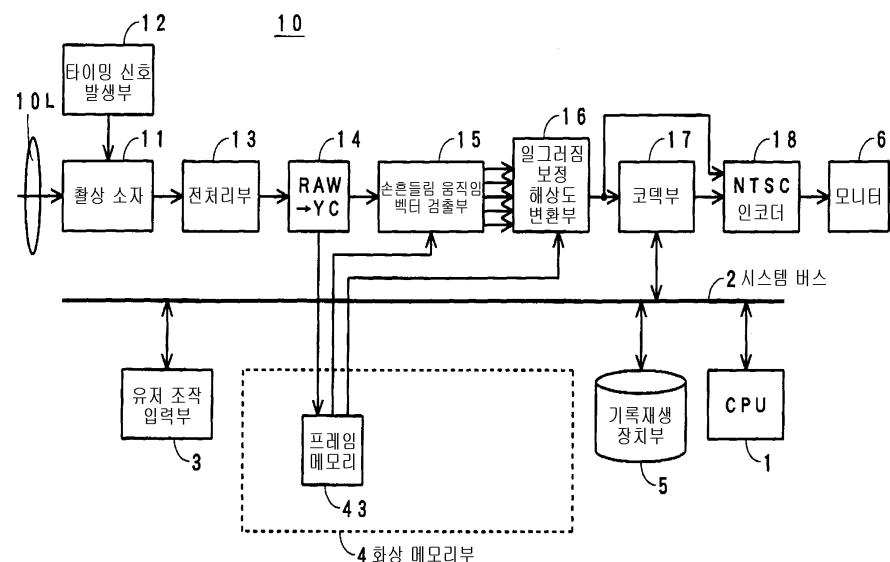
도면51



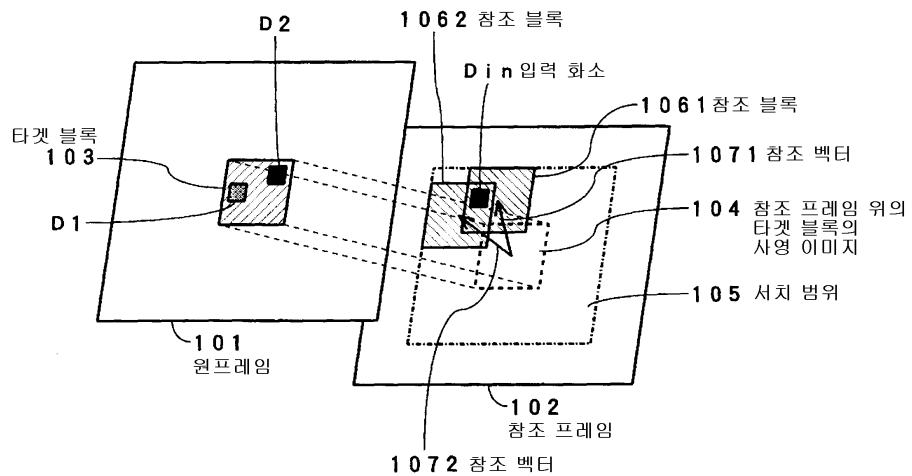
도면52



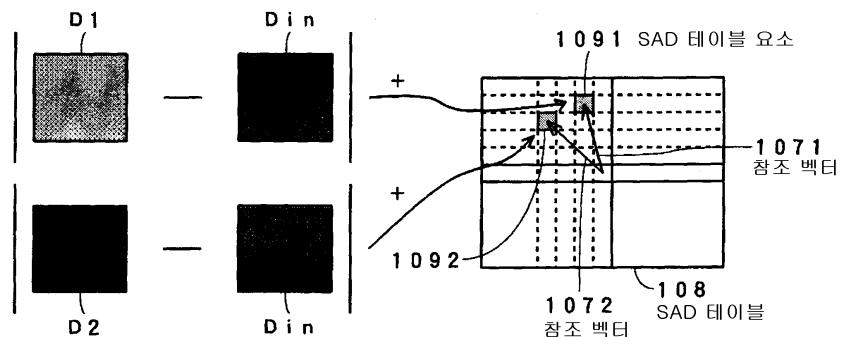
도면53



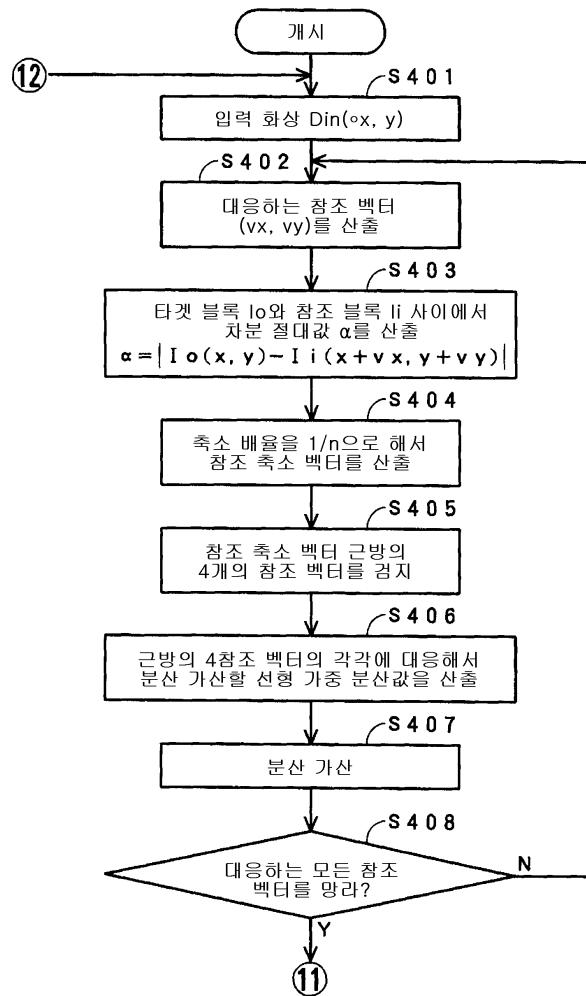
도면54



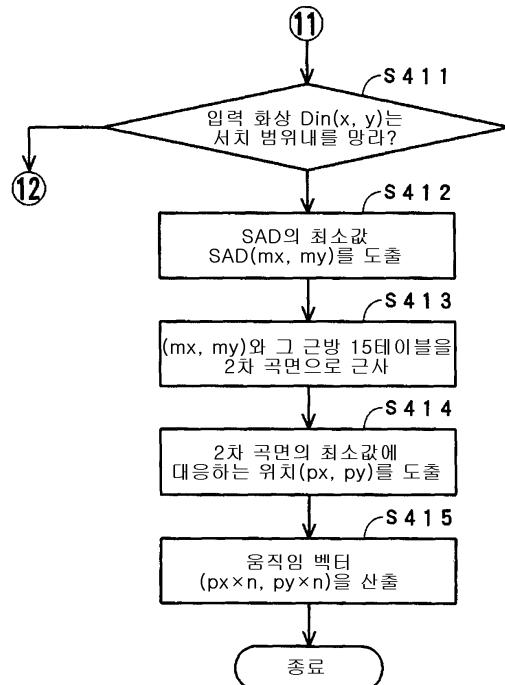
도면55



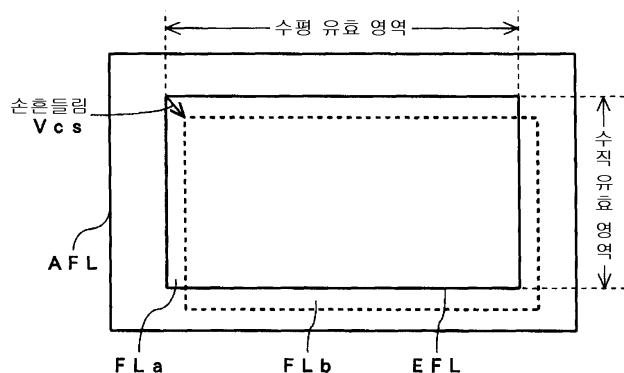
도면56



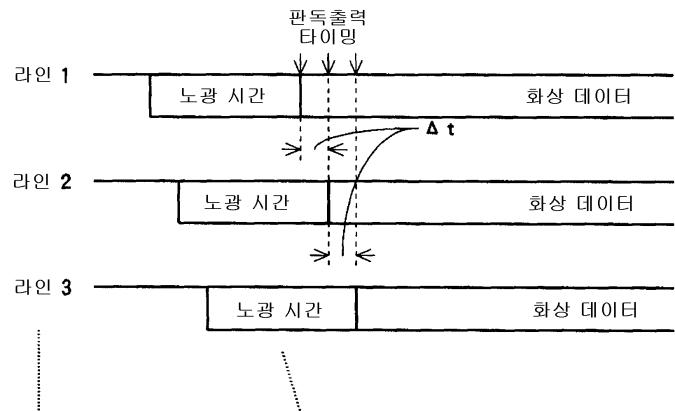
도면57



도면58



도면59

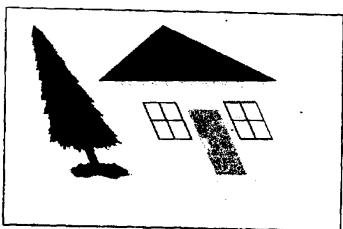


도면60

(A) 본래의 활상 이미지



(B) 달리는 전차 안에서 바깥의 경치를 활영한 경우



(C) (CMOS 이미저에서의) 손흔들림이 발생한 경우



도면61

