



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년09월10일
(11) 등록번호 10-0981147
(24) 등록일자 2010년09월02일

(51) Int. Cl.
G06T 7/00 (2006.01) G06T 1/00 (2006.01)
G06F 19/00 (2006.01)
(21) 출원번호 10-2009-0110178
(22) 출원일자 2009년11월16일
심사청구일자 2009년11월16일
(56) 선행기술조사문헌
Du-Ming Tsai and Chien-Ta Lin, Fast normalized cross correlation for defect detection, Pattern Recognition Letters, vol. 24, no. 15, pp. 2625-2631, November 2003.*
JP2003109009 A
KR20050004044 A
JP1998049681 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
재단법인대구경북과학기술원
대구광역시 달서구 호산동 711
(72) 발명자
권순
대구광역시 달서구 호산동 대구테크노벤처공장 1호관 201호
이중훈
대구광역시 달서구 호산동 대구테크노벤처공장 1호관 201호
(뒷면에 계속)
(74) 대리인
송해모, 김은구

전체 청구항 수 : 총 12 항

심사관 : 김홍수

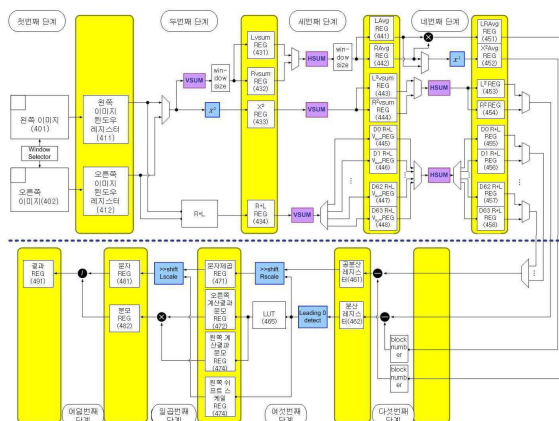
(54) 고속연산으로 이미지의 유사도를 측정하는 방법 및 장치

(57) 요약

본 발명은 고속연산으로 이미지의 유사도를 측정하는 방법 및 장치를 개시하고 있다.

본 발명의 일 실시 예에 따른 고속연산으로 이미지의 유사도를 측정하는 방법은 제 1 이미지 및 제 2 이미지의 유사도를 측정하는 방법에 있어서, 상기 비교 대상인 제 1 이미지의 비교영역의 화소들의 집합인 L 및 제 2 이미지의 비교영역의 화소들의 집합인 R을 저장하는 제 1 단계, 상기 L의 평균값 Lavg, R의 평균값 Ravg을 계산하는 제 2 단계, 상기 L과 R을 곱한 값에서 상기 Lavg, Ravg 및 비례수를 곱한 값을 빼는 제 3단계, 상기 L의 제곱값과 Lavg의 제곱에 비례수를 곱한 값의 차이의 제곱근 및 상기 R의 제곱값과 Ravg의 제곱에 비례수를 곱한 값의 차이의 제곱근을 서로 곱하는 제 4 단계, 및 상기 제 3단계에서 산출된 값에서 상기 제 4 단계에서 산출된 값을 나누어서 유사도를 측정하는 제 5 단계를 포함하며, 상기 제 2 단계, 제 3 단계, 제 4 단계 또는 제 5 단계 중 어느 한 단계의 연산 시간은 다른 한 단계의 연산 시간과 서로 중첩되는 것을 특징으로 한다.

대표도



(72) 발명자

배경렬

대구광역시 동구 신암5동 94-31

문병인

서울특별시 은평구 응암1동 96-28 경향렉스빌 601호

특허청구의 범위

청구항 1

제 1 이미지 및 제 2 이미지의 유사도를 측정하는 방법에 있어서,
 상기 비교 대상인 제 1 이미지의 비교영역의 화소들의 집합인 L 및 제 2 이미지의 비교영역의 화소들의 집합인 R
 을 저장하는 제 1 단계;
 상기 L의 평균값 Lavg, R의 평균값 Ravg을 계산하는 제 2 단계;
 상기 Lavg의 제곱값 및 상기 Ravg의 제곱값을 계산하는 제 2-a 단계;
 상기 L의 제곱값 및 상기 R의 제곱값을 계산하는 제 2-b 단계;
 상기 L과 R을 곱한 값에서 상기 Lavg, Ravg 및 비례수를 곱한 값을 빼는 제 3단계;
 상기 L의 제곱값과 Lavg의 제곱에 비례수를 곱한 값의 차이의 제곱근 및 상기 R의 제곱값과 상기 Ravg의 제곱에
 비례수를 곱한 값의 차이의 제곱근을 서로 곱하는 제 4단계; 및
 상기 제 3단계에서 산출된 값에서 상기 제 4단계에서 산출된 값을 나누어서 유사도를 측정하는 제 5 단계를 포
 함하며,
 상기 제 2-a 단계, 상기 제 2-b단계 및 상기 제 3단계는 병렬로 연산되며,
 상기 제 3 단계와 상기 제 4단계는 병렬로 연산되며,
 상기 L과 상이하하며 상기 제 1 이미지의 비교 영역의 화소들의 집합인 L' 및 상기 R과 상이하하며 상기 제 2 이미
 지의 비교 영역의 화소들의 집합인 R'에 대하여 상기 제 1 단계, 상기 제 2 단계, 상기 제 2-a 단계, 상기 제
 2-b단계, 상기 제 3 단계, 상기 제 4 단계, 상기 제 5 단계를 진행하며, 상기 L, R의 제 1 단계, 상기 제 2 단
 계, 상기 제 2-a 단계, 상기 제 2-b단계, 상기 제 3 단계, 상기 제 4 단계, 상기 제 5 단계들과 병렬로 파이프
 라인을 이루어 연산되는 것을 특징으로 하며,
 상기 제 2 단계는 상기 R 또는 L 영역에서 평균을 구하고자 하는 영역에서 중복 연산 영역을 산출하여, 상기 중
 복 연산 영역에 대해 미리 계산을 수행한 후, 해당 영역의 계산 결과를 반복하여 사용하는 것을 특징으로 하는,
 고속연산으로 이미지의 유사도를 측정하는 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

제 1항에 있어서,
 상기 R 또는 L은 2차원 배열이며, 상기 평균값을 계산하는 제 2 단계는 상기 2차원 배열에서 일방향의 합계를
 구하는 2-1단계; 및
 상기 일방향의 합계를 모두 더하는 2-2단계;를 포함하며,
 상기 2-1 및 상기 2-2는 서로 중첩되어 진행되는 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하
 는 방법.

청구항 5

제 1항에 있어서,

상기 제 3 단계 또는 제 4단계에 있어서, 상기 비례수는 상기 R 또는 L이 2차원 행렬이며, 행렬의 크기가 P, Q 인 경우, P와 Q를 곱한 값에 비례하는 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 방법.

청구항 6

제 1항에 있어서,

상기 제 3단계 또는 제 4단계에 있어서 상기 비례수를 곱하는 것은 비례수를 쉬프트하여 곱할 값에 더하는 과정을 반복하는 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 방법.

청구항 7

제 1항에 있어서,

상기 제 4단계는

상기 제공근을 연산하기 위해 룩업 테이블을 이용하는 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 방법.

청구항 8

제 7항에 있어서,

상기 제 4 단계는

상기 제공근을 구하기 위해 입력되는 인자의 값을 구성하는 일부 비트를 상기 룩업 테이블의 입력값으로 하여 제공근을 연산하며, 상기 일부 비트는 최상위 비트를 포함하여 연속된 비트인 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 방법.

청구항 9

제 1 이미지 및 제 2 이미지의 유사도를 측정하는 장치에 있어서,

상기 비교 대상인 제 1 이미지의 비교영역의 화소들의 집합인 L 및 제 2 이미지의 비교영역의 화소들의 집합인 R을 저장하는 저장부;

상기 L의 평균값 Lavg, R의 평균값 Ravg을 계산하는 평균 연산부;

상기 Lavg의 제공값 및 상기 Ravg의 제공값을 계산하는 제 1 제공 연산부;

상기 L의 제공값 및 상기 R의 제공값을 계산하는 제 2 제공 연산부;

상기 L과 R을 곱한 값에서 상기 Lavg, Ravg 및 비례수를 곱한 값을 빼는 제 1 연산부;

상기 L의 제공값과 Lavg의 제공에 비례수를 곱한 값의 차이의 제공근 및 상기 R의 제공값과 상기 Ravg의 제공에 비례수를 곱한 값의 차이의 제공근을 서로 곱하는 제 2 연산부; 및

상기 제 1 연산부에서 연산한 값에서 상기 제 2 연산부에서 연산한 값을 나누어서 유사도를 측정하는 유사도 측정부를 포함하며,

상기 제 1 제공 연산부, 상기 제 2 제공 연산부, 및 상기 제 1 연산부는 병렬로 연산을 수행하며,

상기 제 1 연산부와 상기 제 2 연산부는 병렬로 연산을 수행하며,

상기 L과 상이하하며 상기 제 1 이미지의 비교 영역의 화소들의 집합인 L' 및 상기 R과 상이하하며 상기 제 2 이미지의 비교 영역의 화소들의 집합인 R'에 대하여 상기 저장부, 상기 평균 연산부, 상기 제 1 제공 연산부, 상기 제 2 제공 연산부, 상기 제 1 연산부, 상기 제 2 연산부, 상기 유사도 측정부는 상기 L, R에 대한 연산을 수행한 후 연속하여 L', R'에 대해 연산을 수행하는 것을 특징으로 하며,

상기 평균 연산부는 R 또는 L 영역에서 평균을 구하고자 하는 영역에서 중복 연산 영역을 산출하여, 상기 중복 연산 영역에 대해 미리 계산을 수행한 후, 해당 영역의 계산 결과를 반복하여 사용하는 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 장치.

청구항 10

삭제

청구항 11

삭제

청구항 12

제 9항에 있어서,

상기 R 또는 L은 2차원 배열이며, 상기 평균값을 계산하는 평균 연산부는 상기 2차원 배열에서 일방향의 합계를 구하는 제 1 합산부; 및

상기 일방향의 합계를 모두 더하는 제 2 합산부;를 포함하며,

상기 제 1 합산부 및 상기 제 2 합산부는 시간을 기준으로 중첩되어 연산하는 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 장치.

청구항 13

제 9항에 있어서,

상기 제 1 연산부 또는 제 2 연산부에 있어서, 상기 비례수는 상기 R 또는 L이 2차원 행렬이며, 행렬의 크기가 P, Q인 경우, P와 Q를 곱한 값에 비례하는 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 장치.

청구항 14

제 9항에 있어서,

상기 제 1 연산부 또는 제 2 연산부에 있어서, 상기 비례수를 곱하는 것은 비례수를 쉬프트하여 곱할 값에 더하는 과정을 반복하는 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 장치.

청구항 15

제 9항에 있어서,

상기 제 2 연산부는

상기 제곱근을 연산하기 위한 룩업 테이블을 포함하는, 고속연산으로 이미지의 유사도를 측정하는 장치.

청구항 16

제 15항에 있어서,

상기 제 2 연산부는

상기 제곱근을 구하기 위해 입력되는 인자의 값을 구성하는 일부 비트를 상기 룩업 테이블의 입력값으로 하여 제곱근을 연산하며, 상기 일부 비트는 최상위 비트를 포함하여 연속된 비트인 것을 특징으로 하는, 고속연산으로 이미지의 유사도를 측정하는 장치.

명세서

발명의 상세한 설명

기술분야

[0001] 본 발명은 고속연산으로 이미지의 유사도를 측정하는 수행하는 방법 및 장치를 개시하고 있다.

배경기술

[0002] 컴퓨터 비전분야에서 정규화 된 상호상관도(Normalized Cross Correlation, 이하 'NCC'라 한다) 기반의 정합도 분석기법은 비전기반 검사장비나 스테레오 정합과 같은 영상 간의 영역단위의 정합 정도를 분석하는데 활용이 되었다. 하지만 고해상도의 영상데이터를 NCC 연산 수식을 통해 처리하는 경우 그 연산량의 복잡도로 인해 실시간 처리가 어려운 한계가 있다.

발명의 내용

해결하고자하는 과제

[0003] 본 발명은 NCC연산구조를 파이프라인화 하여 처리함으로써 연산속도를 개선하였고, 각 파이프라인 단계에서 처리되는 연산구조를 병렬적 하드웨어처리에 적합한 구조로 제안하였다.

[0004] 특히, 본 발명은 중복되는 연산의 횟수를 줄이고 해당 연산 결과를 재차 사용할 수 있도록 하여 연산 속도를 향상시키고자 한다.

과제 해결수단

[0005] 전술한 과제를 달성하기 위해, 본 발명의 일 실시 예에 따른 고속연산으로 이미지의 유사도를 측정하는 방법은 제 1 이미지 및 제 2 이미지의 유사도를 측정하는 방법에 있어서, 상기 비교 대상인 제 1 이미지의 비교영역의 화소들의 집합인 L 및 제 2 이미지의 비교영역의 화소들의 집합인 R을 저장하는 제 1 단계, 상기 L의 평균값 Lavg, R의 평균값 Ravg을 계산하는 제 2 단계, 상기 L과 R을 곱한 값에서 상기 Lavg, Ravg 및 비례수를 곱한 값을 빼는 제 3단계, 상기 L의 제곱값과 Lavg의 제곱에 비례수를 곱한 값의 차이의 제곱근 및 상기 R의 제곱값과 상기 Ravg의 제곱에 비례수를 곱한 값의 차이의 제곱근을 서로 곱하는 제 4단계, 및 상기 제 3단계에서 산출된 값에서 상기 제 4단계에서 산출된 값을 나누어서 유사도를 측정하는 제 5 단계를 포함하며, 상기 제 2 단계, 제 3단계, 제 4단계 또는 제 5단계 중 어느 한 단계의 연산 시간은 다른 한 단계의 연산 시간과 서로 중첩되는 것을 특징으로 한다.

[0006] 본 발명의 다른 실시 예에 따른 고속연산으로 이미지의 유사도를 측정하는 제 1 이미지 및 제 2 이미지의 유사도를 측정하는 장치에 있어서,

[0007] 상기 비교 대상인 제 1 이미지의 비교영역의 화소들의 집합인 L 및 제 2 이미지의 비교영역의 화소들의 집합인 R을 저장하는 저장부, 상기 L의 평균값 Lavg, R의 평균값 Ravg을 계산하는 평균 연산부, 상기 L과 R을 곱한 값에서 상기 Lavg, Ravg 및 비례수를 곱한 값을 빼는 제 1 연산부, 상기 L의 제곱값과 Lavg의 제곱에 비례수를 곱한 값의 차이의 제곱근 및 상기 R의 제곱값과 상기 Ravg의 제곱에 비례수를 곱한 값의 차이의 제곱근을 서로 곱하는 제 2 연산부, 및 상기 제 1 연산부에서 연산한 값에서 상기 제 2 연산부에서 연산한 값을 나누어서 유사도를 측정하는 유사도 측정부를 포함하며, 상기 평균 연산부, 제 1 연산부, 제 2 연산부 또는 유사도 측정부 중 둘 이상이 시간을 기준으로 중첩되어 연산하는 것을 특징으로 한다.

효과

[0008] 본 발명은 파이프라인화에 의한 NCC 연산 수행속도를 개선할 수 있으므로, 보다 큰 영역 간의 비교나 영상 프레임 내 영역 탐색범위가 넓은 경우에도 연산 속도가 향상되어 실시간으로 프로세싱을 가능하게 할 수 있다.

[0009] 또한, 본 발명에서 제안하는 방법과 이를 구현한 하드웨어 기반 NCC연산구조는 컴퓨터 비전분야에서 패턴매칭과 같은 영상비교 분야뿐만 아니라 스테레오비전에서의 좌, 우 시차정합과 같은 다양한 분야에서 과도한 연산 량에 의한 제약사항을 해결해 줄 것으로 기대한다.

[0010] 뿐만 아니라, 본 발명의 영역간 비교를 위해 제시되는 박스 필터링 기법은 블록단위의 화소를 동시에

연산하며, 중복되는 연산은 1회 처리하여 연산 속도를 향상시키는 효과가 있다.

발명의 실시를 위한 구체적인 내용

- [0011] 이하, 본 발명의 일부 실시 예들을 예시적인 도면을 통해 상세하게 설명한다. 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.
- [0012] 또한, 본 발명의 구성 요소를 설명하는 데 있어서, 제 1, 제 2, A, B, (a), (b) 등의 용어를 사용할 수 있다. 이러한 용어는 그 구성 요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 용어에 의해 해당 구성 요소의 본질이나 차례 또는 순서 등이 한정되지 않는다. 어떤 구성 요소가 다른 구성요소에 "연결", "결합" 또는 "접속"된다고 기재된 경우, 그 구성 요소는 그 다른 구성요소에 직접적으로 연결되거나 접속될 수 있지만, 각 구성 요소 사이에 또 다른 구성 요소가 "연결", "결합" 또는 "접속"될 수도 있다고 이해되어야 할 것이다.
- [0013] 도 1은 컴퓨터 비전 어플리케이션과 같은 이차원 영상영역에서 유사도를 분석하기 위해 사용되는 정규상호상관(NCC, Normalized Cross Correlation)의 연산 과정을 나타내는 수식을 포함하는 도면이다.
- [0014] 비교되는 두 영상영역을 $P \times Q$ 의 크기라고 할 때, 도 1의 수식을 적용하여 NCC결과를 계산할 수 있다.
- [0015] 도 1에서 일반적인 응용방법에 있어 L, R은 각각 좌측의 기준 영상과 우측의 비교 영상의 영역을 의미하며, x, y는 영역 내화소의 좌표를 의미한다. d는 좌측 기준영상의 영역에 대한 우측 비교영상의 영역의 상대적인 위치를 의미하며 특히 스테레오 비전에서는 시차(disparity)를 의미한다. i, j는 각각 윈도우 내에서의 위치를 의미하며, μ 는 평균값을 의미한다.
- [0016] 도 1의 NCC 연산구조를 분석하면 상관정도를 연산하는 분자부(numerator) 및 비교하고자 하는 영상 L, R에 대한 제곱의 합(Squared Sum) 및 그것의 곱에 대한 제곱근을 적용한 분모부(denominator)로 이루어져 있다. 그리고 각 분자 및 분모부에는 정규화를 위해 L 및 R에 대한 평균의 차가 적용되어 있다.
- [0017] 연산의 구조에 있어 일반적으로 연산량이 많이 요구되는 부분은 각 L 및 R 영역의 평균을 구하는 연산부, 분자 및 분모의 곱셈기, 분모의 제곱근 연산기 그리고 분자와 분모를 나누는 나눗셈기를 들 수 있다.
- [0018] 가령 도 3의 순서도를 살펴보면, 두 L, R의 영상 내 비교하고자 하는 영역의 크기가 $P \times Q$ 인 사각형 영역이라고 할 때 두 영역 간의 비교에 소모되는 연산 량을 분석하면, 각 영역의 평균을 구하는데 $P*Q$ 번의 덧셈 기 및 한 번의 나눗셈기가 필요하다. 또한 분모의 경우 제곱근 안의 연산의 경우 $2*P*Q$ 번의 각각의 덧셈 기, 뺄셈기 및 제곱기가 필요하고 그 결과에 대해 한 번의 제곱근기 및 곱셈기가 추가로 필요하다. 분자의 경우 $2*P*Q$ 번의 뺄셈기 및 $P*Q$ 번의 덧셈기 및 곱셈기가 필요하다. 그리고 최종적으로 분자와 분모를 나누는 데 필요한 나눗셈기가 필요하다.
- [0019] 본 발명에서의 유사도는 비교 대상인 이미지/영상의 유사 정도를 측정하는 것으로, 유사도의 일 실시예로 정규 상호상관기(NCC)를 사용한 정합도(matching)를 중심으로 설명한다.
- [0020] 또한, 본 발명에서의 이미지는 정지영상, 동영상 등을 포함하며, 영상 내의 작은 비교 영역(작은 영상, 이미지의 일부분 등) 역시 이미지라는 범주 내에 포함된다.
- [0021] 도 2는 본 발명의 일 실시예에 따른 도 1의 수식을 간략화 하여 전개한 도면이다. 도 1의 수식과 달리 도 2의 수식에서는 L, R 이미지의 각 $P \times Q$ 영역의 평균값을 미리 구하지 않고 분모부의 제곱의 합 또는 분자부의 각 L, R 영역 화소의 곱의 합을 구하는 과정에서 평균값을 얻을 수 있으므로 NCC연산 수식의 동작속도를 개선할 수 있다.
- [0022] 본 발명에서는 도 1, 2의 수식과 같이 정규화 된 상호상관(NCC)방법을 빠르게 연산하기 위해 파이프라인화(pipelined) 된 하드웨어 구조를 기술한다. 수식의 간략화 및 하드웨어 비용을 최소화하기 위해 도 1의 수식을 도 2의 수식과 같이 전개하여 적용한다. 도 2의 수식을 적용한 정합 비용 계산을 위한 순서는 도3의 순서를 따라 계산하며 도4는 이를 파이프라인 구조의 하드웨어로 나타낸 구성도이다.
- [0023] 도 3은 본 발명의 일 실시예에 의한 L, R영상의 일치 여부를 검토하는 순서도이다. 비교하고자 하는 영상(이미지

지)에서 비교 및 피비교영역을 선택하여 비교영역 및 피비교영역에 해당하는 화소집단을 불러온다(S301). 위치 카운터가 0이면 처음 시작이므로 피비교영역에 대한 작업을 진행한다. 피비교영역의 화소의 평균값인 L_{avg} , 화소의 제곱인 L^2 를 연산한다(S304). 그리고 화소의 제곱의 합인 $\text{Sum}(L^2)$ 의 결과를 연산한다(S306). 그리고 $\text{Sum}(L^2)$ 에서 $P*Q*L_{avg}^2$ 를 빼는 연산을 한다(S308). 그리고 LZD(leading zero detect)를 하여 비트수를 줄인다(S310). 그리고 룩업테이블을 통해 제곱근을 연산한다(S312). LZD를 수행하여 비트수를 줄이는 것은 제곱근 연산에서 룩업테이블의 크기가 과도하게 크지 않도록 하기 위해서이다. 그리고 피비교영역의 분모값을 저장하고 스케일링 인자(scaling factor)를 저장한다(S314). 그리고 위치 카운터를 증가시키고(S340) 다음 과정을 진행한다.

[0024] 위치 카운터가 0이 아니면 비교 영역에 대해 작업을 진행한다. 비교 영역(d)의 화소의 평균값 R_{avg} , 화소의 제곱 R^2 , 두 비교영역의 곱인 $R*L$ 을 연산한다(S320). 그리고 비교 영역의 화소의 제곱의 합인 $\text{Sum}(R^2)$, 비교 영역의 화소간 곱인 $R*L$ 의 합인 $\text{Sum}(R*L)$ 을 연산한다(S322). 그리고 비교 영역의 분산값을 LZD(Leading Zero Detect) 하여 비트수를 줄이고 이때 줄인 비트수만큼 공분산값도 비트수를 줄인다(S326). 이는 앞서 살펴본 바와 같이 제곱근을 구하기에 너무 넓은 값을 룩업테이블로 만들게 될 경우, 룩업테이블이 과도하게 커질 수 있으므로, LZD를 통해 비트 수를 줄여서 룩업테이블에서 제곱근을 구할 수 있도록 하기 위함이다.

[0025] 그리고 비교 영역의 분산값을 LUT(LookUp Table, 룩업테이블)을 통해 제곱근 연산을 하고(S328), 연산 결과인 비교영역의 분모값을 저장한다(S330). 앞서 S314 단계에서 계산된 값을 토대로, 피비교영역의 분모값*비교영역의 분모값, 저장된 스케일링 팩터(scaling factor)를 이용하여 분자값의 비트 수를 조정하고(S332), 분자/분모 나누기 연산을 수행한다(S334). 그리고 영상의 마지막인지를 확인하여 비교하려는 영상이 존재하면 위치 카운터를 증가시키고(S340), 처음부터 프로세스를 진행한다.

[0026] 이러한 연산은 일반적으로 한번만 이루어지는 것이 아니라 L 및 R 영상 내에서 동일한 크기의 윈도우 영역을 순차적으로 이동해가며 계속 비교가 이루어지는 구조가 되므로, L, R 영상 한 프레임에 대한 정합과정에 소요되는 연산량은 상기에 기술된 연산량*비교횟수 만큼 늘어나게 된다. 더구나 일반적으로 카메라로부터 입력되는 영상은 초당 수십 프레임(15,30,60)이므로 실시간으로 들어오는 영상을 빠짐없이 NCC연산과정을 처리하기 위해서는 상기 수식에 초당 프레임수를 더 곱한 복잡도를 가지게 된다. 이러한 복잡한 구조의 NCC를 실시간 처리하기 위해서 전용 ASIC 칩이나 DSP프로세서와 같은 특화된 하드웨어를 사용할 수 있다.

[0027] 제안된 구조의 특징은 크게 두 가지로 나뉘어 설명할 수 있으며 즉, 파이프라인화 된 연산 동작 구조와 복잡한 처리시간을 요구하는 연산단계에 대한 하드웨어적 고속구조 기법이다.

[0028] 제안된 NCC 하드웨어 구조는 도 1, 2의 복잡한 연산구조를 FPGA와 같은 프로그래밍 가능한 하드웨어 장치에서 동작할 수 있도록 병렬적 파이프라인화 되어 있다.

[0029] 도 4는 본 발명의 일 실시예에 따른 NCC의 각 연산 과정을 파이프라인화 한 하드웨어 구성도이다.

[0030] 도3의 순서도 및 도 4의 하드웨어 구성도를 참고하면, 입력되는 두 비교 영상은 각 한 라인씩 파이프라인 스테이지에 입력이 된다. 각 파이프라인 단계의 역할은 다음과 같다.

[0031] 첫 번째 단계는 외부 메모리에 저장된 영상데이터를 내부 메모리로 불러와 NCC 계산을 위해 필요한 부분을 레지스터(411, 412)로 저장하는 단계이다. NCC 연산을 위해 전체 영상 데이터를 불러오는 것은 적합한 방법이 아니므로 제안된 구조에서는 두 비교영상의 각 라인씩 입력하도록 하였다.

[0032] 두 번째 단계는 VSUM을 이용하여 블록의 세로 평균을 구하여 저장하고 블록전체의 제곱, 왼쪽, 오른쪽 이미지의 블록에서 각각의 화소값을 곱하여 저장한다.

[0033] 세 번째 단계는 앞 단계에서 구한 세로로의 평균을 HSUM을 이용하여 가로로 다시 더하여 평균을 구하며, VSUM을 이용하여 앞 단계에서의 제곱 값과 $R*L$ 값의 세로 합을 구하는 부분을 진행한다.

[0034] 네 번째 단계는 앞 단계에서 구한 평균값을 제공하는 부분과 왼쪽 이미지의 평균값과 오른쪽 이미지의 평균값을 곱하여 그 값을 저장하는 부분, 앞에서 구한 제곱과 $R*L$ 의 세로 합을 가로로 더하여 전체 합을 구하는 부분을 진행한다.

- [0035] 다섯 번째 단계는 분모 값 계산을 위해 평균의 제곱 값에 윈도우 마스크 사이즈만큼 곱한 다음 제곱의 합에서 빼는 부분과 분자 값 계산을 위해 두 이미지 평균값을 곱한 값에 윈도우 마스크 사이즈만큼 곱한 다음 $R*L$ 의 합에서 빼는 부분을 진행한다.
- [0036] 여섯 번째 단계에서는 제곱근을 위한 모듈로 룩업 LUT(look-up table, 룩업테이블)을 사용하며 LUT의 크기가 커지지 않도록 입력 비트수를 LZD(Leading Zero Detect)과정을 통해 보정한다.
- [0037] 일곱 번째 단계에서는 앞 단계에서 저장한 왼쪽 이미지의 스케일 인자(scale factor) 사용하여 분자를 오른쪽으로 쉬프트(right shift)하는 부분과 두 분모 값을 곱하는 부분을 포함한다.
- [0038] 여덟 번째 단계: 분자를 분모로 나누는 부분이며 나눗셈기에서 소요되는 동작 클럭이 길어질 경우 해당 단계를 다시 여러 단계로 세분화 할 수 있다.
- [0039] 앞서 설명된 파이프라인화 된 NCC 처리구조에서 각 단계에서는 연산복잡도가 높은 모듈들이 포함이 되어 있다.
- [0040] 따라서, 본 발명에서는 다음과 같은 하드웨어적 기법을 적용하여 각 단계에서 소모되는 연산시간을 최소화 하였다.
- [0041] 도 2의 수식을 참조하면 NCC연산과정에는 평균값 및 제곱의 합 등 윈도우 마스크 내 화소에 기반한 합을 구하는 과정이 존재한다.
- [0042] 본 발명에서는 박스 필터링(box-filtering) 방법을 적용하여 이차원 윈도우 마스크영역에서의 합 연산을 구한다. 제안된 구조에서 윈도우 마스크 내 화소들의 합을 구하는 과정을 설명하면, 이미지 값을 읽을 때 한 줄씩 읽어 오도록 하여, 세로의 평균을 먼저 구한 후 그것의 평균을 다시 구하면 윈도우 마스크 전체의 평균을 구할 수 있게 된다. 이때 각 줄의 세로의 평균과 전체의 평균을 따로 저장 해두도록 하여 블록을 이동할 때 블록간에 중복되는 부분을 다시 계산할 필요가 없게 되며 연산 시간이 단축 된다.
- [0043] 본 발명에 의한 박스 필터링 과정을 간략히 살펴보면 다음과 같다.
- [0044] 도 5는 본 발명의 일 실시예에 의한 윈도우 마스크와 블록간의 상관 관계를 보여주는 도면이다. 인접한 $K*L$ 개의 화소들(511)을 중심으로 한 4 개의 윈도우 마스크들(510, 520, 530, 540)을 하나의 블록(550)으로 묶을 수 있다. 4개의 윈도우 마스크들의 크기는 $9*9$ (길이 및 높이 각각 9) 이며, 이들 윈도우 마스크에 대해 연산한 블록 연산값이 포함되는 배열의 크기는 $2*2$ 이다. 이하 윈도우 마스크는 마스크로 설명하고자 한다.
- [0045] 도 6은 본 발명의 일 실시예에 의한 이차원 데이터 배열의 구성을 보여주는 도면이다.
- [0046] 도 6은 $M*N$ 의 크기를 가지는 화소 블록(630)에 대해 병렬로 동시에 연산을 수행한다. 화소 블록(630) 중 제 1화소(631)에 대해 주변 화소들을 계산하게 되는데, 주변 화소들의 범위, 즉 제 1 화소에 대한 마스크(611)의 범위는 도 6에서 파란 색으로 표시한 바와 같다. 마스크(611)의 크기는 $K*L$ 이 된다. 앞서 살펴본 도 1, 2, 3 에서의 비교 영역에 대한 크기인 P, Q 는 본 도면에서의 마스크의 크기에 해당하며, 따라서 도 1, 2, 3의 P, Q 의 실시예로 마스크의 크기인 K, L 이 될 수 있다. 그리고 제 1화소(631)에 대한 마스크(611) 내에 포함된 화소들을 계산한 연산 결과(가산 결과)는 620의 첫번째 영역인 621에 저장된다. 화소 블록(630)의 크기가 $M*N$ 이므로, 연산을 수행한 결과 역시 $M*N$ 이 된다. 또한, 화소 블록(630)의 $M*N$ 개의 화소들에 대하여 각각 마스크 영역이 존재하므로, 해당 화소 블록(630)의 마스크는 모두 $M*N$ 개이며, 이들 마스크들의 영역을 모두 합치면 610과 같이 된다. 마스크들은 서로 중첩되므로, 병렬 화소 블록(630)에 대한 모든 마스크들의 영역을 합친 영역(610)의 크기는 $(K+M-1)*(L+N-1)$ 이 된다.
- [0047] 다시 정리하면, 가산 연산에서 계산되는 결과 값(620)들은 $M*N$ 개가 있으며 620과 같이 저장되어 있다. 병렬 화소 블록의 크기(610) $M*N$ 과 같이 각각의 마스크들의 갯수가 $M*N$ 개이며, 마스크의 크기는 $K*L$ 이다. 따라서, $M*N$ 개의 마스크로 구성되는 마스크 영역의 크기는 $(M + K - 1)*(N + L - 1)$ 이 된다.

- [0048] 이하 본 발명의 일 실시예가 적용되는 과정을 보여준다. 설명의 편의를 위하여 각각의 화소들은 화소에 할당된 색과 숫자로 구별된다. 화소로 표현되는 사각형이 연산의 인자로 포함되어 가산/감산 등의 연산에 적용된다.
- [0049] 본 발명에서 포함하는 박스필터링을 구현하기 위해 이차원 데이터배열 영역에 적용하는 과정은 다음과 같은 세 부류의 서로 다른 연산과정을 포함한다. 그 과정은 처음 블록을 연산하는 과정, 이차원 데이터배열의 영역에서 처음 열(가로) 진행방향의 블록 연산과정 또는 처음 행(세로) 진행방향의 블록 연산과정 그리고 나머지 영역에서의 블록 연산과정으로 구분할 수 있다.
- [0050] 도 7a, 7b, 7c, 7d, 7e는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0051] 도 7a는 본 발명의 일 실시예에 의한 연산이 적용될 화소들의 배열을 보여주는 도면이다. 처음 연산이 수행되는 병렬 화소 블록(705)은 이전에 계산해 둔 중간 값, 전체 합이 없으므로 블록에 해당하는 마스크 전체 영역(710)에 대해 연산을 수행해야 한다. 병렬 화소 블록의 크기가 3x3이며, 연산값에 대한 행렬(411)이 3x3이며, 마스크의 개수도 9(3*3)개가 된다. 이하 설명에서 병렬 화소 블록의 크기가 3x3인 경우를 예로 들어 설명하나, 이는 본 발명의 일 실시예로 본 발명이 이에 국한되는 것은 아니다. 또한 반드시 동일한 행과 열의 수로 이루어질 필요가 없으며, 2x3, 5x9와 같이 구성될 수 있다. 이에 따른 마스크의 크기 역시 다양하게 변화될 수 있다.
- [0052] 제안된 구조에서는 행별로 가로줄의 합을 먼저 구한 후 다시 구해진 가로줄의 합을 더하여 전체 합을 구한다. 본 발명의 설명을 위해 가로줄의 합을 더한 후, 세로 방향으로 합하여 전체 합을 구하는 방식을 제시하고 있으나, 이는 실시예에 불과하며, 세로줄의 합을 모두 더한 후, 가로 방향으로 합하여 전체 합을 구하는 방식 역시 본 발명의 실시예에 포함된다.
- [0053] 도 7b는 본 발명의 일 실시예에 의한 중복하여 연산을 진행하는 행들에 대하여 미리 연산을 수행하는 과정을 보여주는 도면이다.
- [0054] 중복되는 연산을 제거하기 위해 K x L 마스크 내 동일 열에 속한 화소들의 가로 합(행가산값)을 구하는 과정으로, 반복되는 동일 열(M, M+1, ..., K-1, K)에 속한 화소들의 행 가산값을 S에 저장한다.
- [0055] 도 7c는 본 발명의 일 실시예에 의한 중복 연산을 진행하는 행의 행 가산값을 이용하여 연산을 수행하는 과정을 보여주는 도면이다.
- [0056] 도 7a의 전체 행 가산값을 구하는 과정이 도시되어 있다. 731과 같이 도 7b에서 미리 연산한 S 및 S에 포함되지 않은 화소들(1, 2, ..., M-1)들을 더하여 구한다. 732, 733, 735, 736, 737 역시 같은 방식으로 하여 이전에 계산하였던 S에 포함되지 않은 화소들의 값을 더하는 방식으로 계산된다. . 도 7c에서 a, b, c는 각각 마스크 전체 영역(710)에서 첫번째 행, 두번째 행, 세번째 행에 대해 연산 과정을 보여주고 있으나, 도 7a에서 알 수 있듯이, 마스크 전체 영역(710)에는 많은 행이 포함되어 있으므로, 도면 및 설명에서 생략하였으나, 네번째 행, 다섯번째 행 등 L+N+1행(738, 739, 740)까지 동일한 과정으로 연산이 진행될 수 있다.
- [0057] 도 7d 및 도 7e는 블록 가산값을 연산하는 과정을 보여주는 도면이다.
- [0058] 도 7d는 본 발명의 일 실시예에 의한 중복 연산을 진행하는 열의 열 가산값을 이용하여 연산을 수행하는 과정을 보여주는 도면이다.
- [0059] 도 7b의 가로 행의 값을 구하는 과정과 같이 동일 행에 속한 화소들의 세로 합(열가산값)을 구하는 과정에서 반복되는 동일 행(N, N, N+1, ..., L-1, L)에 속한 화소들의 열 가산값을 S에 저장한다.
- [0060] 도 7e는 중복되는 열 가산값인 S를 사용하는 과정을 보여주는 도면이다. 751은 도 7d의 미리 연산한 S 및 S에 포함되지 않는 화소들(a, a+1, ..., N-1)을 가산한다. 752, 753, 755, 756, 757, 758, 759, 760 모두 미리 연산한 S 및 이에 포함되지 않는 화소들을 가산하는 과정을 보여주고 있다.
- [0061] 도 7a, 7b, 7c, 7d, 7e에서 S로 표시된 화소들의 합은 반복하여 적용이 되므로 한 번의 연산을 통해 구한 후 동일 열에 속한 화소들의 가로 합을 구하는 데 재사용할 수 있다. 이렇게 할 경우 S를 구하는데 소요되는 덧셈연

산이 (K-M)번이고 이 값을 한 번만 구해놓게 되면 나머지 각 가로 합 의 연산에서는 M개의 덧셈연산으로 가로 합 을 연산할 수 있게 된다.

- [0062] 마찬가지로 세로 합을 연산하는 데도 (L-N)개의 덧셈을 통해 한번만 S를 미리 구해놓으면 나머지 각 세로합의 연산에서는 N번의 덧셈만으로도 가능하다.
- [0063] 도 8a, 8b, 8c, 8d는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다. 도 7a의 병렬 화소 블록에 대한 연산이 완료된 후 가산결과(711)를 이용하여 열(가로) 진행 방향의 다음 블록을 연산하기 위한 과정이 제시되어 있다.
- [0064] 다음 병렬 화소 블록을 계산하기 위해서는 가로로 M 화소만큼 블록을 이동 시킨다. 이렇게 이동시킨 블록(805)에 의한 마스크 전체 영역(810, 검은 색으로 경계지어짐)은 이동하기 전 블록의 마스크 전체 영역(710, 녹색으로 경계지어짐)과 중복되는 구간이 생기며, 추가된 열은 이전 블록의 마스크 전체 영역에서 1열~M열을 제외한 부분이 중복된다. 이 중복되는 구간을 다시 계산하지 않도록 하기 위해 앞의 블록 연산 과정에서 계산해 놓은 가로줄의 합에서 추가로 들어온 만큼의 화소를 더해주고 빠져나간 만큼의 화소를 빼주면 적은 연산 량으로도 마스크 전체의 합을 구할 수 있다.
- [0065] 도 8b는 본 발명의 일 실시예에 의한 상세한 계산 과정을 보여주는 도면이다.
- [0066] 821은 도 7c에서 계산하였던 값을 이용해서 계산하는 과정을 나타내고 있다. M은 마스크의 이동으로 제외된 화소의 값이며 K+M은 마스크의 이동으로 추가된 화소이다. 7c에서 계산한 값에서 제외된 화소의 값을 빼고, 추가된 화소의 값을 더하는 과정이 제시되어 있다. 822, 823, 825, 826, 827 모두 같은 방식으로 제외된 화소의 값을 빼고, 추가된 화소의 값을 더하는 과정이 제시되어 있다.
- [0067] 도 8c는 본 발명의 일 실시예에 의한 중복 연산을 진행하는 열의 열 가산값을 이용하여 블록 연산을 수행하는 과정을 보여주는 도면이다. 앞서 살펴본 바와 같이, 도 7d에서 진행된 과정과 동일하게 진행된다. 동일 행에 속한 화소들의 세로 합(열가산값, 즉 블록 가산값)을 구하는 과정에서 반복되는 동일 행(N, N+1, ..., L-1, L)에 속한 화소들의 열 가산값을 S에 저장한다.
- [0068] 도 8d는 중복되는 열 가산값인 S를 사용하여 블록 가산값을 산출하는 과정을 보여주는 도면이다. 도 7e에서 설명한 과정과 동일하므로 도 7e의 설명으로 대체하고자 한다.
- [0069] 블록을 하단으로 이동하는 경우에도 동일하게 중복되는 값을 계산할 수 있다. 즉 블록을 상하좌우로 이동할 경우, 인접한 블록의 계산 결과를 이용하면 마스크 내에서 추가된 부분에 대해서만 연산을 수행하게 되므로 연산 속도를 높일 수 있다.
- [0070] 앞서 살펴본 바와 같이, 중복되는 영역에 대한 계산은 미리 설정된 마스크의 크기에 따라, 혹은 이전에 계산한 정보를 가지고 연산하게 되는 횟수를 줄일 수 있다. 한편, 마스크 내 화소들의 합을 구한 후 평균값을 얻기 위해서는 구해진 합에 마스크 내 화소수를 나누어 주어야 하는데 이를 위해 나누기 연산이 필요하다. 본 발명의 구조에서는 나눗셈기의 연산시간이 길다는 단점을 보완하기 위하여 참조표인 룩업테이블(LUT, look-up table)를 사용한다. 룩업 테이블은 입력된 값에 대한 나눗셈 값을 저장하고 있어, 별도의 연산 과정이 아니라 매칭되는 값을 가져오는 방식이므로 연산 횟수를 줄일 수 있다.
- [0071] 도 2의 수식에 표시된 분모에 있는 분산(variance)을 구하기 위해 마찬가지로 윈도우 마스크 전체의 화소 값의 제곱을 다 더해야 하는데 이 또한 각 화소값을 제공하는 것만 제외하면 앞서 평균값을 구하는 과정과 동일하게 박스 필터링 방법을 사용한다.
- [0072] 윈도우 마스크 전체크기를 곱하는 과정은 다음과 같이 진행된다.
- [0073] 도 2의 수식을 참조하면 각 분자, 분모의 평균값에 윈도우 마스크 크기인 P*Q를 곱하여야 한다. 곱하기 연산은 나누기 연산만큼은 아니지만 그래도 상당한 규모의 하드웨어가 필요하고 시간도 많이 소요된다. 하지만 곱하는

숫자가 윈도우 마스크 전체크기로 정해져 있으면 몇 번의 쉬프트 동작과 몇 개의 덧셈기로 구현이 가능하다. 예를 들어 곱하기 441의 경우 441은 $256 + 128 + 32 + 16 + 8 + 1$ 로 나타낼 수 있다. 이는 각각 2의 8승, 7승, 5승, 4승, 3승, 0승인데 이를 약간 수정하여 $256 + 128 + 32 + 16 + 8 - 4 - 2 + 1$ 로도 나타낼 수 있다. 이를 2에 관해 풀어 쓰면 $2^8 + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 - 2^1 - 2^0$ 가 되는데 $(2^1 + 2^0)$ 로 묶어 내면 $2^7(2^1 + 2^0) + 2^4(2^1 + 2^0) + 2^2(2^1 + 2^0) - 2^0(2^1 + 2^0)$ 가 된다. 그러므로 제안된 구조에서는 곱할 값을 1비트 왼쪽 쉬프트 시킨 뒤 원래 값과 더하고 다시 이 값을 몇 번 쉬프트 하여 더하고 빼는 과정을 통해 곱셈을 대신한다.

[0074] 제곱근(square root)을 연산하는 과정은 다음과 같이 진행된다.

[0075] 도 2에 표시된 수식의 분모에 있는 분산(variance)의 제곱근(square root)연산은 하드웨어로 구현하기도 어렵고, 구현한다고 해도 연산하는데 시간이 너무 많이 소요된다. 그래서 제안된 구조에서는 참조표(look-up table)를 만들어 모든 입력 값에 따른 출력 값을 기술해 놓고 참조하는 형식을 취한다. 하지만 이때 입력되는 값의 범위가 너무 넓어져 버리면 참조표(look-up table)의 크기가 아주 커져버리게 된다. 그래서 바로 앞에 첫 비트가 0인지 체크하는 "leading 0 detect" 모듈을 두어 입력으로 들어가는 데이터의 전체 비트 중 가장 큰 유효 값인 1을 찾아 거기서부터 몇 비트를 잘라 참조표(look-up table)의 입력으로 들어가게 하여 참조표(look-up table)의 크기도 적당하게 조절하고 어느 정도의 정확성도 확보하도록 하였다. 그리고 추가적으로 "leading 0 detect"를 하면서 뒤에 잘린 값들을 보정해 주는 작업이 필요한데 참조표(look-up table)을 통해 출력되는 값을 다시 늘려주는 방법 대신 분자 값인 공분산(covariance)값의 뒷부분을 똑같이 자르는 방법을 사용한다. 이는 도 3에서 살펴본 S216 단계의 공분산 스케일링 과정에서 진행된다.

[0076] 마지막으로 앞서 구한 계산 결과를 도 2의 분자와 분모로 하여 나누기 연산을 진행하는 것이 필요하다.

[0077] 앞에서 설명한 바와 같이 나누기 연산은 하드웨어로 구현 했을 때 그 크기도 크고 연산하는데 시간이 많이 걸린다. 이를 극복하기 위해 나누기 연산을 여러 단계로 나누고 각각의 단계를 한 클럭 사이클에 계산 할 수 있도록 하며, 한 단계에 소요되는 클럭을 최대한 활용하여 나누기 모듈의 개수를 최소화 한다. 예를 들어 결과로 출력되는 화소 값이 8-bit이고 나누기 모듈의 입력으로 들어가는 값의 크기가 16-bit 이며 한번에 16개(4 x 4)의 화소에 대해 연산을 하고 한 단계에 4 clock이 소요된다고 했을 때, 16-bit / 16-bit를 한 클럭(clock)에 실행 할 수 없으므로 2단계, 혹은 4단계로 나눠 high-radix SRT 알고리즘을 사용하여 총 4 클럭, 혹은 2 클럭에 걸쳐 계산을 완료할 수 있도록 한다. 그리고 나누기 모듈을 2개(2 클럭 소요될 경우), 혹은 4개(4 클럭 소요될 경우)를 두어 동시에 여러 화소 단위로 계산하되 단계별 소요 시간을 최대한 활용하여 나누기 모듈을 최소화 한다.

[0078] 본 발명의 일 실시예에 의한 도 4의 각각의 단계들은 파이프라인 과정에서 병렬로 실행되는 단위가 되며, 둘 이상의 단위를 결합하여 하나의 파이프라인 과정으로 병렬 실행을 할 수도 있다. 즉, 본 발명은 ZNCC를 구하기 위해 병렬의 구조에서 파이프라인으로 연산을 수행하여 연산 속도를 향상시키고자 하는 것으로, 도 4에 도시된 다양한 단계들이 둘 이상 결합하여 하나의 단계가 될 수 있으며, 도 4의 각각의 단계가 다시 세분화되어 병렬로 실행될 수 있다.

[0079] 도 9는 본 발명의 다른 실시예에 의한 파이프라인 구조를 보여주는 도면이다. 도 9는 도 4의 첫번째 단계, 두번째 단계, 세번째 단계, 네번째 단계가 새로이 구성되어 파이프라인으로 구성되는 경우를 보여주는 도면이다.

[0080] 파이프라인은 앞서 설명한 바와 같이 하나의 단계/구성요소가 동작하는 동안, 시간적으로 이전에 동작하였던 단계/구성요소가 그 다음 처리할 값을 연산하는 구성으로 되어있다. 예를 들어, 3단계의 작업이 진행되는 프로세스에서, 각각의 단계에 대해 3 msec이 소요된다면, 통상의 프로세싱에서는 2개의 단위 업무를 진행하는데 있어서 소요되는 시간을 계산해보면 18msec(2개의 단위업무 * 3개의 단계 * 3msec)되지만, 파이프라인을 적용할 경우, 1단계, 2단계, 3단계가 동시에 진행되므로, 각각의 단계가 연속되는 단위 업무를 연속하여 진행하게 되므로, 실제 소요되는 시간은 하나의 단위 업무가 진행되면서 다른 단위 업무 역시 진행되는 바, 제 1 단위 업무에 대해서는 1, 2, 3단계 모두 진행되지만, 제 2 단위 업무는 제 1 단위 업무가 진행되는 과정과 중첩되므로 마지막 3단계만 단독으로 진행되어 총 소요 시간은 4단계 * 3msec 이 되어서 12msec이 소요된다.

[0081] 도 9는 이러한 파이프라인에 근거하여 본 발명을 구성하는 과정이다. 앞서 살펴본 바와 같이 ZNCC를 계산하는 과정에서 필요한 인자로는 왼쪽 이미지(L)에 대한 평균값(Lavg), 오른쪽 이미지(R)에 대한 평균값(Ravg) 및 이

들 각각의 제공의 값(Lavg², Ravg²) 그리고 이들을 곱한 값((Lavg * Ravg), 그리고 L, R 각각을 제공한 값(L², R²) 이다. 이들을 구하기 위해, 본 발명의 일 실시예인 도 9에서는 2개의 단계로 구성되어 있다. 따라서, R, L 이미지가 저장된 레지스터 또는 메모리(902, 904)에서 각각의 화소값이 제 1단계에서 시행되는데, 제 1 단계에서는 R, L에 대해 평균을 구하는 평균 연산부(912, 916)과 제공을 구하는 제공 연산부(914, 918), 그리고 R, L을 곱하는 곱셈부(Multiplier)(920)이 입력된 값을 연산하는 과정을 수행한다. 그리고 이 단계에서 산출된 값은 다시 제 2 단계의 제공 연산부(932, 936)와 곱셈부(Multiplier)(934)에 입력되어 결과를 산출하게 된다.

- [0082] 제 2단계에서 연산을 수행하는 동안, 제 1 단계 역시 연산을 수행하게 되어, 연산 속도를 줄일 수 있다.
- [0083] 2단계에서 산출된 값은 도 4의 다섯번째 단계에서 연산 인자로 입력되어 연산된다. 따라서, 도 4의 첫번째 단계, 두번째 단계, 세번째 단계, 네번째 단계는 도 9의 제 1, 제 2 단계로 대체되어 적용될 수 있다.
- [0084] 도 9의 평균 연산부(912, 916)은 앞서 살펴본 박스 필터링 기법을 적용하여 연산의 횟수를 줄일 수 있다. 만약, L, R의 평균을 구하는 과정에서 전체 합을 구하는 경우, 영역 내의 합을 구하기 위해 중복되는 영역에 대한 합을 미리 계산한 후, 해당 계산 값을 이용할 경우, 이전 계산결과를 참조하여 계산이 이루어지므로 연산 횟수가 줄어든다.
- [0085] 한편, 평균을 연산하거나 제공을 수행하는 과정 역시 파이프라인화 시켜서 연산 횟수를 줄일 수 있다.
- [0086] 도 10은 본 발명의 일 실시예에 의한 양 이미지의 유사도를 측정하는 과정을 보여주는 도면이다.
- [0087] 도 10의 도면은 두 개의 이미지, 즉, 제 1 이미지 및 제 2 이미지의 유사도를 측정하는 방법에 관한 것이다.
- [0088] 도 10의 유사도는 앞서 살펴본 정규상호상관기를 일 실시예로 하는 것으로 양 이미지의 유사 여부를 판단하는 것을 의미한다. 또한 유사도는 양 이미지간의 정합도(matching)를 의미할 수 있다. 또한, 본 발명에서 영상, 이미지 등으로 호칭되는 것은 동영상의 스틸 컷 또는 정지된 영상(사진 등)의 전체 또는 일부를 의미하는 것으로, 이미지 전체를 비교할 수도 있고, 이미지 내의 일부 영역에 대해서 비교를 수행할 수 있다. 본 발명은 이미지의 전체 혹은 부분에 모두 적용할 수 있는 것이며 본 발명의 이미지는 이러한 발명의 사상을 모두 포함하고 있다. 또한 도 3에서 피비교영역과 비교영역을 나누어서 설명하였는데, 피비교영역과 비교영역은 미리 정하여질 수도 있고, 비교 과정에서 양 이미지간에 피비교영역과 비교영역이 바뀔 수도 있다. 본 발명은 이러한 내용을 모두 포함한다.
- [0089] 상기 비교 대상인 제 1 이미지의 비교영역의 화소들의 집합인 L 및 제 2 이미지의 비교영역의 화소들의 집합인 R을 저장한다(제 1 단계, S1010). L, R은 앞서 오른쪽 이미지와 왼쪽 이미지에 대한 예로 제시되었으나, 반드시 왼쪽/오른쪽과 같은 이미지로 나뉘어지는 것은 아니며, 본 발명은 두 개의 이미지에 대한 비교를 수행하는 영역에 적용될 수 있다. 따라서, 본 발명에서 L, R은 비교할 대상의 두 이미지를 각각 지칭하는 것으로 볼 수 있다. 다음으로, 상기 L의 평균값 Lavg, R의 평균값 Ravg를 계산한다(제 2 단계, S1020). 그리고 상기 L과 R을 곱한 값에서 상기 Lavg, Ravg 및 비례수를 곱한 값을 빼는 연산을 수행한다(제 3단계, S1030). 이는 앞서 살펴본 도 2의 수식의 분자의 예가 될 수 있다. 그리고 상기 L의 제공값과 Lavg의 제공에 비례수를 곱한 값의 차이의 제곱근 및 상기 R의 제공값과 Ravg의 제공에 비례수를 곱한 값의 차이의 제곱근을 서로 곱하는 연산을 수행한다(제 4단계, S1040). 이는 앞서 살펴본 수식에서의 분모에 해당할 수 있다. 그리고 마지막으로 상기 제 3단계에서 산출된 값에서 상기 제 4단계에서 산출된 값을 나누어서 유사도를 측정한다(제 5 단계, S1050). 한편, 앞서 살펴본 바와 같이 각각의 단계는 파이프라인 방식으로 연결되어 병렬로 수행할 수 있다. 따라서, 제 2 단계, 제 3단계, 제 4단계 또는 제 5단계 중 어느 한 단계의 연산 시간은 다른 한 단계의 연산 시간과 서로 중첩될 수 있다. 연산 시간이 중첩된다는 것은 어느 단계를 수행하면서 다음 단계를 수행할 수 있는 것을 의미하며, 앞서 살펴본 도 4, 9에서 보여준 파이프라인 방식의 하드웨어에서 구현가능하다.
- [0090] 본 발명의 상세한 실시예를 살펴보면, 상기 제 2 단계는 L 또는 R 영역에서 평균을 구하고자 하는 영역에서 중복 연산 영역을 산출하여, 상기 중복 연산 영역에 대해 미리 계산을 수행한 후, 해당 영역의 계산 결과를 반복하여 사용할 수 있는데, 이는 박스 필터링을 적용하여 중복되는 연산을 미리 저장하고 차후 합을 구하거나 평균을 구하는 과정에서 연산을 줄일 수 있도록 하는 과정을 의미한다.
- [0091] 한편 파이프라인 방식으로 구현되는 또다른 예는 연산시간을 중첩시킴에 있어서, 상기 Ravg 또는 Lavg를 계산하는 제 1 시간단위, 상기 R의 제공값 또는 L의 제공값을 계산하는 제 2 시간단위, 그리고 상기 R과 L을 서로 곱

하는 연산을 수행하는 제 3 시간단위 중 둘 이상이 서로 중첩되도록 구현할 수 있다.

- [0092] 본발명의 일 실시예로 평균을 구하는 과정에서 L, R이 2차원 배열일 경우, 세로합을 구하고 세로합의 가로합을 구하여 전체합을 구하는 방식을 적용할 수 있다. 물론 가로합을 먼저 구한 후 이들의 세로 합을 구하여 전체 합을 구할 수 있다. 이를 보다 일반적으로 구현하기 위해 2차원 배열에서 일방향의 합계를 구하고 그 일방향의 합계를 모두 더하여 전체 합을 구하여 평균을 구할 수 있다. 물론, 이 평균을 구하기 위해 전체 합 또는 세로합/가로합을 구하는 과정에서 앞서 살펴본 박스 필터링 기법을 적용할 수 있다. 그리고, 합을 구하는 과정은 앞서 살펴본 바와 같이, 서로 중첩되어 병렬로 처리할 수 있다. 따라서, 세로합을 구한 결과에 대해 가로합을 구하면서, 다음 단계에 처리할 세로합을 구할 수 있다.
- [0093] 또한, 제3, 4단계에 있어서, 상기 비례수를 구함에 있어서, 상기 L 또는 R이 2차원 행렬이며, 행렬의 크기가 P, Q인 경우, P와 Q를 곱한 값에 비례하도록 할 수 있다. 비례하도록 연산하는 방법 중 일 실시예는 P*Q값을 그대로 곱한 것을 포함한다.
- [0094] 또한, 앞서 곱셈 연산을 쉬프트 연산을 통해 수행함을 설명한 바 있는데, 제 3단계 또는 제 4단계에서 이를 적용하여, 비례수를 곱하기 위해 비례수를 쉬프트하여 곱할 값에 더하는 과정을 반복할 수 있다. 비례수를 곱하는 과정 역시 파이프라인화 하여 병렬로 시간적으로 중첩하여 실행할 수 있다.
- [0095] 한편, 제공근을 구하는 것은 많은 연산을 필요로 하는 것이므로, 제 4단계를 구현함에 있어서, 상기 제공근을 연산하기 위해 룩업 테이블을 이용할 수 있다. 룩업테이블의 구현 예로, 상기 제공근을 구하기 위해 입력되는 인자의 값을 구성하는 일부 비트를 상기 룩업 테이블의 입력값으로 하여 제공근을 연산하되, 상기 일부 비트는 상기 인자의 값을 구성하는 비트 중 유효도가 높은 비트가 되도록 할 수 있다.
- [0096] 비트의 유효도가 높다는 것은 해당 비트가 1에서 0, 또는 0에서 1로 바뀔 경우 그 값의 변화가 큰 것을 의미한다. 예를 들어, 001001(십진수로 9)에서 MSG(Most Significant Bit)를 왼쪽에 있는 비트라고 할 때, 왼쪽에서 세번째 비트의 1은 8의 값을 가지므로 세번째 비트의 값이 바뀌면 전체 값이 크게 바뀐다. 세번째 비트가 1에서 0으로 변하면 9에서 1로 값이 바뀌게 된다. 한편, 마지막 비트(가장 오른쪽에 있는 비트)의 1은 1의 값을 가진다. 따라서, 그 값이 바뀌어도 큰 변화가 없다. 마지막 비트인 1이 0으로 바뀌어도 값은 9에서 8로 바뀌는 것에 불과하다. 따라서 룩업 테이블을 사용하는 경우, 모든 비트에 대한 제공근을 구할 수 없는 경우, 상위 몇비트에 대해서만 룩업 테이블을 작성하여 이에 대한 제공근을 구할 수 있다. 그리고 그 결과값인 공분산의 뒷부분을 제거하는 방식으로 구할 수 있다.
- [0097] 본 발명은 컴퓨터 비전 기반 영상처리 시스템의 응용에 있어 적용 가능한 정규화된 상호상관도(Normalized Cross Corellation)기반 정합(matching)도 분석 알고리즘의 빠른 연산을 위한 방법 및 구조에 대해 제시하고 있다.
- [0098] 도 11은 본 발명의 또다른 실시예에 의한 양 이미지의 유사도를 측정하는 파이프라인으로 구성된 하드웨어 구성을 보여주는 도면이다.
- [0099] 도 11은 제 1 이미지 및 제 2 이미지의 유사도를 측정하는 장치의 전부 혹은 일부를 표현하고 있다. 상기 비교대상인 제 1 이미지의 비교영역의 화소들의 집합인 L 및 제 2 이미지의 비교영역의 화소들의 집합인 R을 저장하는 저장부(1102, 1104)는 레지스터, RAM 등 다양한 구현예를 가진다. 평균 연산부(1112, 1116)은 상기 L의 평균값 Lavg, R의 평균값 Ravg을 계산하며, 제 1 연산부(1150)는 상기 L과 R을 곱한 값에서 상기 Lavg, Ravg 및 비례수를 곱한 값을 빼는 연산을 수행한다. 그리고, 제 2 연산부(1160)은 상기 L의 제공값과 Lavg의 제공에 비례수를 곱한 값의 차이의 제공근 및 상기 R의 제공값과 Ravg의 제공에 비례수를 곱한 값의 차이의 제공근을 서로 곱하는 연산을 한다.
- [0100] 그리고 유사도 측정부(1170)는 제 1 연산부(1150)에서 연산한 값에서 상기 제 2 연산부(1160)에서 연산한 값을 나누어서 유사도를 측정하게 된다.
- [0101] 이때, 평균 연산부(1112, 1116), 제 1 연산부(1150), 제 2 연산부(1160) 그리고, 유사도 측정부(1170)은 파이프 라인 방식으로 동작한다. 이들 구성요소 중 일부일 수 있고, 이들 전체 구성요소가 모두 파이프라인 방식으로 동작할 수 있다. 따라서, 연산을 수행하는 구성요소 둘 이상이 시간을 기준으로 중첩되어 연산하게 된다. 앞서 살펴본 바와 같이 평균 연산을 수행하는 과정에서 이미지 화소들에 대한 제공을 함께 계산할 수 있다. 평균 연산부(1112, 1116)는 L 또는 R 영역에서 평균을 구하고자 하는 영역에서 중복 연산 영역을 산출하여, 상기 중

복 연산 영역에 대해 미리 계산을 수행한 후, 해당 영역의 계산 결과를 반복하여 사용할 수 있다. 앞서 살펴본 바와 같이 박스 필터링을 수행하여 연산 횟수를 줄일 수 있다.

- [0102] 파이프라인을 구현하기 위해 각각의 구성 요소들이 시간을 기준으로 중첩되어 연산하는 일 실시예로, 상기 Ravg 또는 Lavg를 계산하는 제 1 시간단위, 상기 R의 제공값 또는 L의 제공값을 계산하는 제 2 시간단위, 그리고 상기 R과 L을 서로 곱하는 연산을 수행하는 제 3 시간단위 중 둘 이상이 서로 중첩되도록 할 수 있다.
- [0103] 상기 L 또는 R은 2차원 배열이며, 상기 평균값을 계산하는 평균 연산부(1112, 1116)는 상기 2차원 배열에서 일 방향의 합계를 구하는 제 1 합산부와 상기 일방향의 합계를 모두 더하는 제 2 합산부를 더 포함할 수 있고, 상기 제 1 합산부 및 상기 제 2 합산부는 시간을 기준으로 중첩되어 연산할 수 있다.
- [0104] 상기 제 1 연산부(1150), 제 2 연산부(1160)에서 상기 비례수는 상기 L 또는 R이 2차원 행렬이며, 행렬의 크기가 P, Q인 경우, P와 Q를 곱한 값에 비례하도록 설정할 수 있다.
- [0105] 상기 제 1 연산부(1150), 제 2 연산부(1160)에서 상기 비례수를 곱하는 것은 비례수를 쉬프트하여 곱할 값에 더하는 과정을 반복할 수 있다.
- [0106] 한편, 제곱근을 구하기 위해 룩업테이블과 같이 참조표를 사용할 수 있다. 상기 제 2 연산부(1160)는 제곱근을 연산하기 위한 룩업 테이블을 사용하는데, 이때, 룩업 테이블을 구현하는 일 실시예로 상기 제곱근을 구하기 위해 입력되는 인자의 값을 구성하는 일부 비트를 상기 룩업 테이블의 입력값으로 하여 제곱근을 연산할 수 있다. 이때, 상기 일부 비트는 상기 인자의 값을 구성하는 비트 중 유효도가 높은 비트가 된다. 유효도가 높은 비트에 대해서는 도 10에서의 설명으로 대신하고자 한다.
- [0107] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 개시된 실시 예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시 예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면의 간단한 설명

- [0108] 도 1은 컴퓨터 비전 어플리케이션에서 템플릿 매칭시의 연산 과정을 나타내는 수식을 포함하는 도면이다.
- [0109] 도 2은 본 발명의 일 실시예에 따른 도 1의 수식을 간략화 하여 전개한 도면이다.
- [0110] 도 3은 본 발명의 일 실시예에 의한 L, R영상의 유사 여부를 검토하는 순서도이다.
- [0111] 도 4는 본 발명의 일 실시예에 따른 NCC의 각 연산 과정을 파이프라인화 한 하드웨어 구성도이다.
- [0112] 도 5는 본 발명의 일 실시예에 의한 마스크와 블록간의 상관 관계를 보여주는 도면이다.
- [0113] 도 6은 본 발명의 일 실시예에 의한 이차원 데이터 배열의 구성을 보여주는 도면이다.
- [0114] 도 7a, 7b, 7c, 7d, 7e는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0115] 도 8a, 8b, 8c, 8d는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0116] 도 9는 본 발명의 다른 실시예에 의한 파이프라인 구조를 보여주는 도면이다.
- [0117] 도 10은 본 발명의 일 실시예에 의한 양 이미지의 유사도를 측정하는 과정을 보여주는 도면이다.
- [0118] 도 11은 본 발명의 또다른 실시예에 의한 양 이미지의 유사도를 측정하는 파이프라인으로 구성된 하드웨어 구성을 보여주는 도면이다.

도면

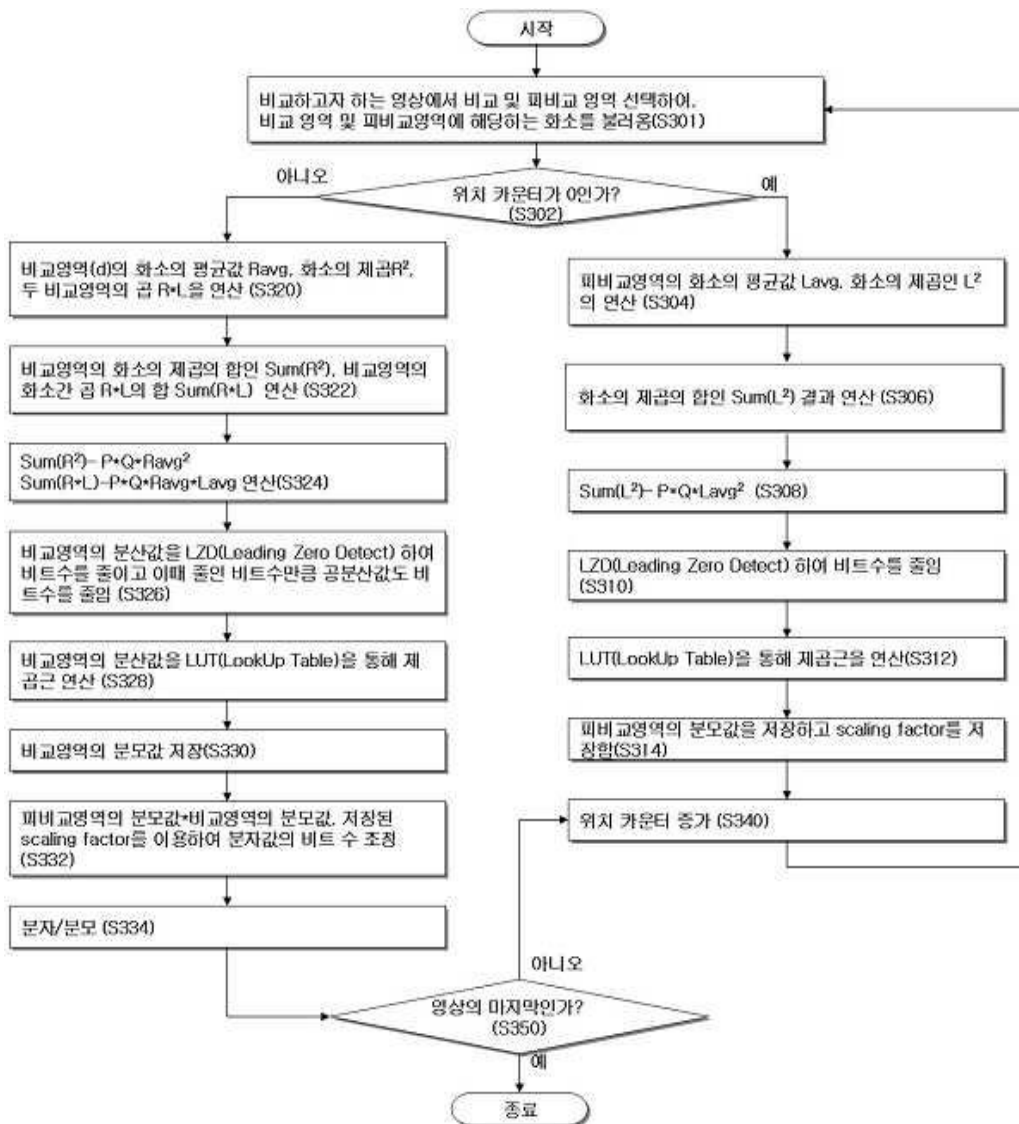
도면1

$$ZNCC(x,y,d) = \frac{\sum_{j=1}^Q \sum_{i=1}^P [L(x+i,y+j) - \mu(L(x,y))] \cdot [R(x+d+i,y+j) - \mu(R(x+d,y))]}{\sqrt{\sum_{j=1}^Q \sum_{i=1}^P [L(x+i,y+j) - \mu(L(x,y))]^2} \cdot \sqrt{\sum_{j=1}^Q \sum_{i=1}^P [R(x+d+i,y+j) - \mu(R(x+d,y))]^2}}$$

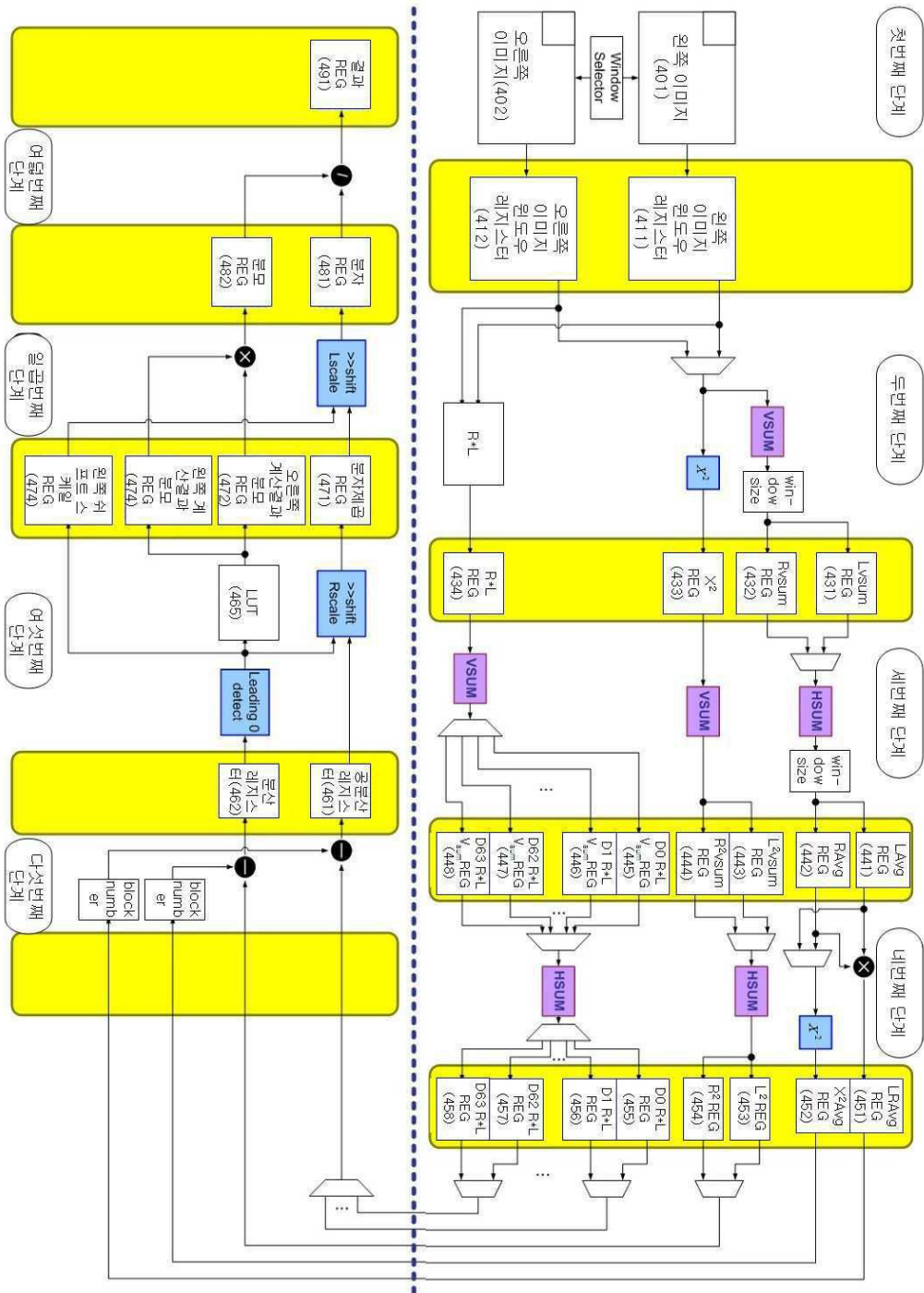
도면2

$$ZNCC(x,y,d) = \frac{\sum_{j=1}^Q \sum_{i=1}^P L(x+i,y+j) \cdot R(x+d+i,y+j) - P \cdot Q \cdot \mu(L(x,y)) \cdot \mu(R(x+d,y))}{\sqrt{\sum_{j=1}^Q \sum_{i=1}^P L^2(x+i,y+j) - P \cdot Q \cdot \mu(L(x,y))^2} \cdot \sqrt{\sum_{j=1}^Q \sum_{i=1}^P R^2(x+d+i,y+j) - P \cdot Q \cdot \mu(R(x+d,y))^2}}$$

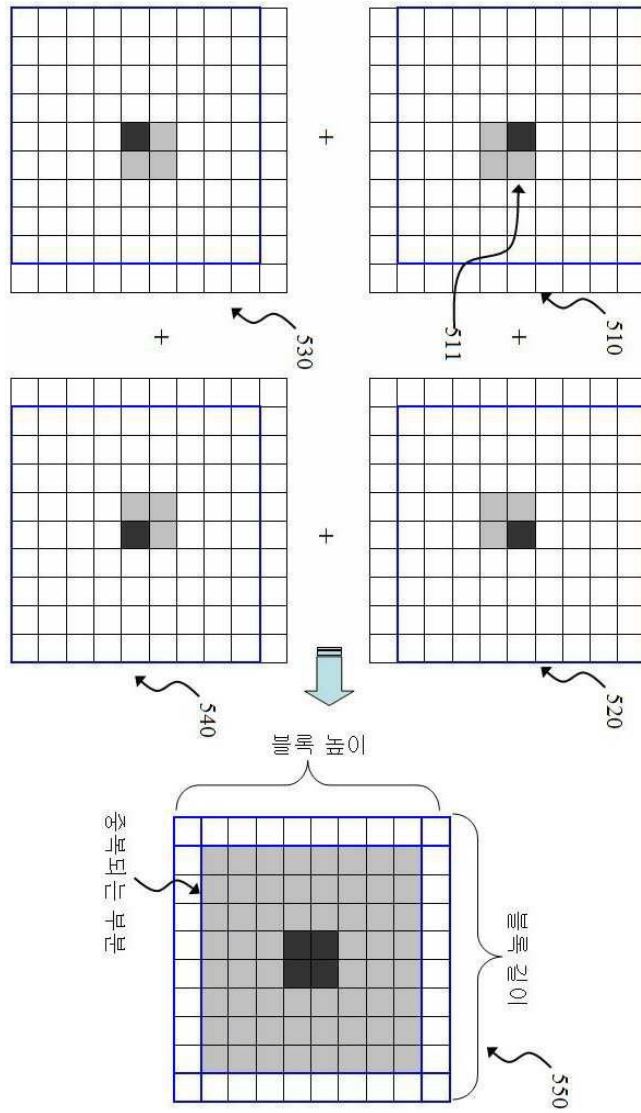
도면3



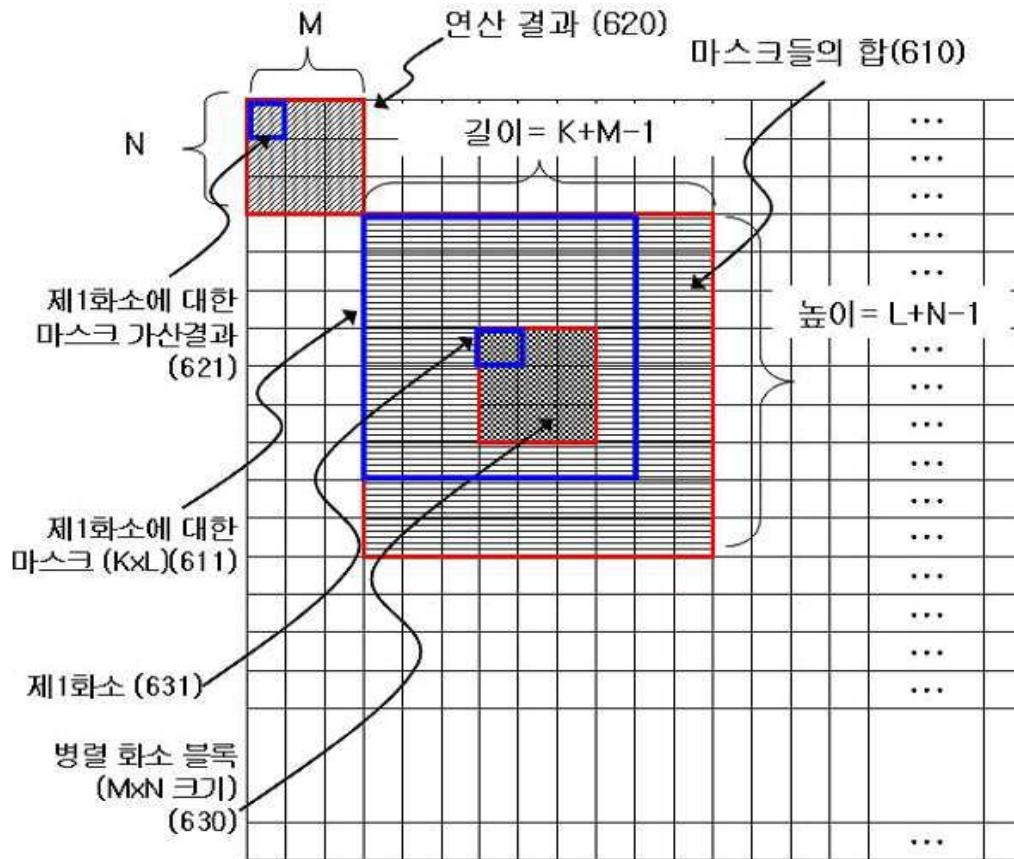
도면4



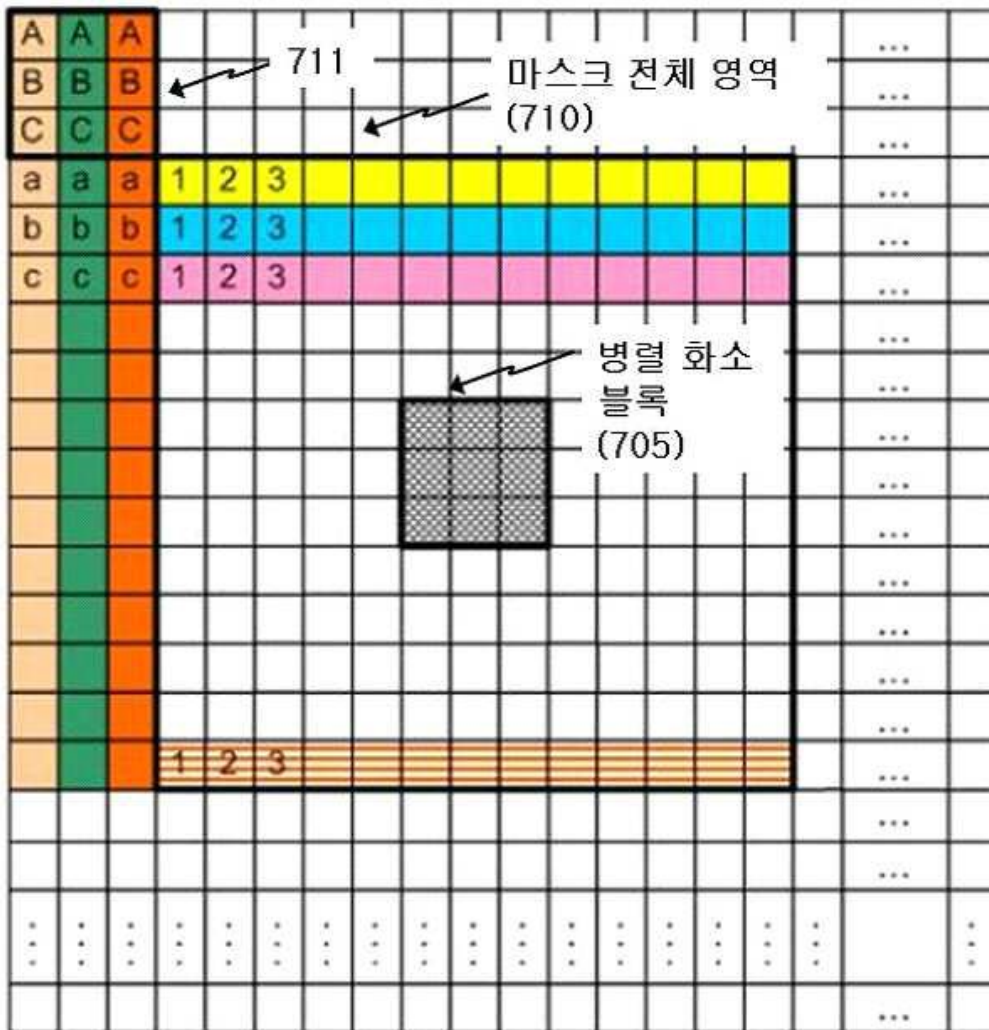
도면5



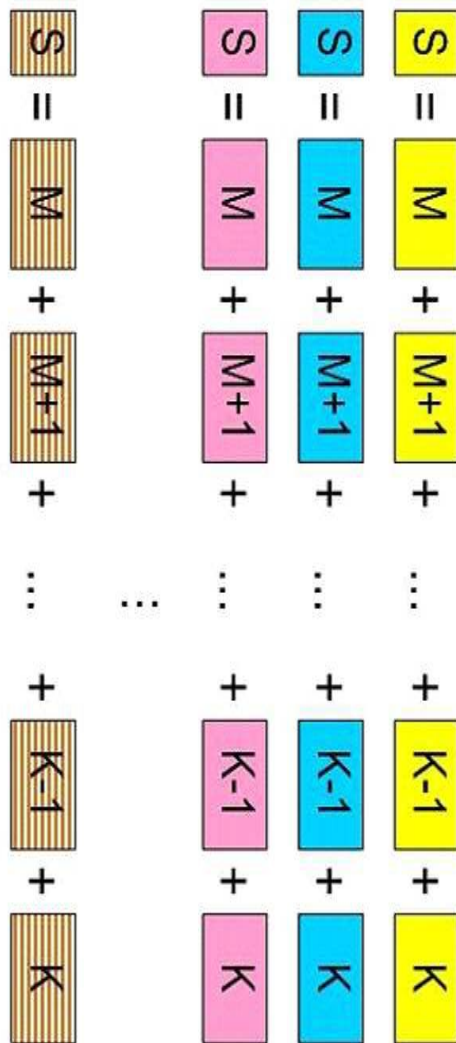
도면6



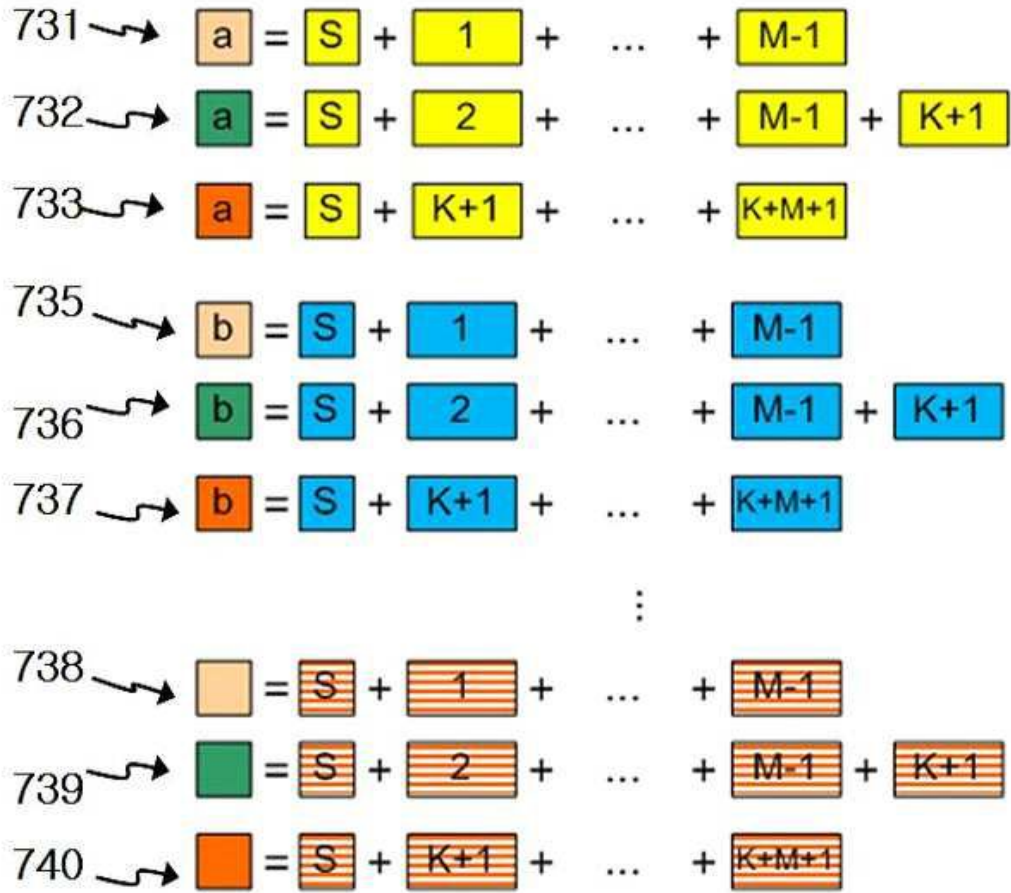
도면7a



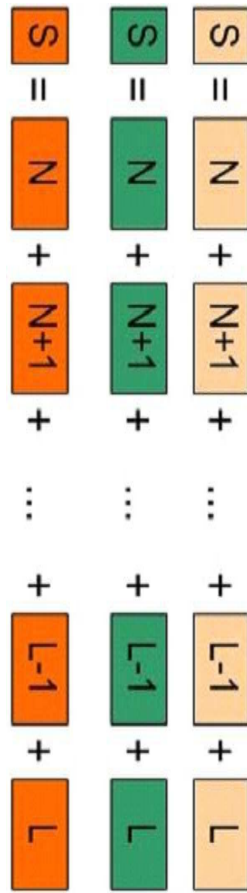
도면7b



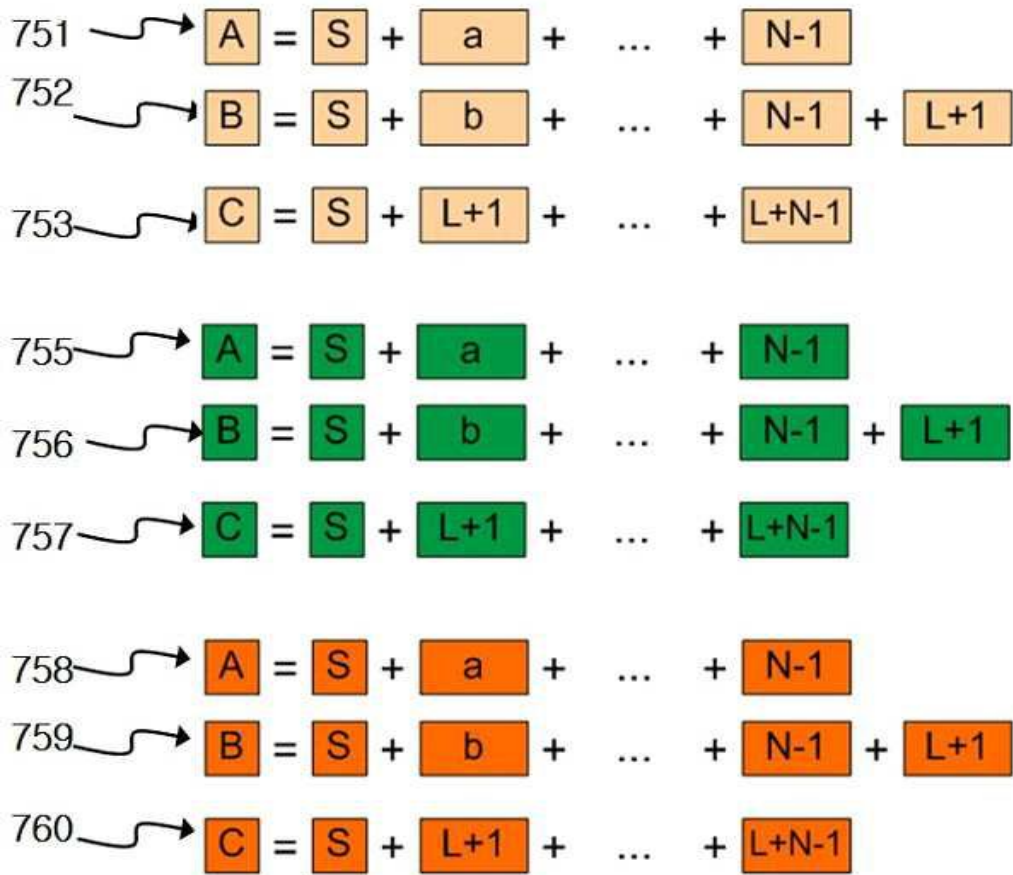
도면7c



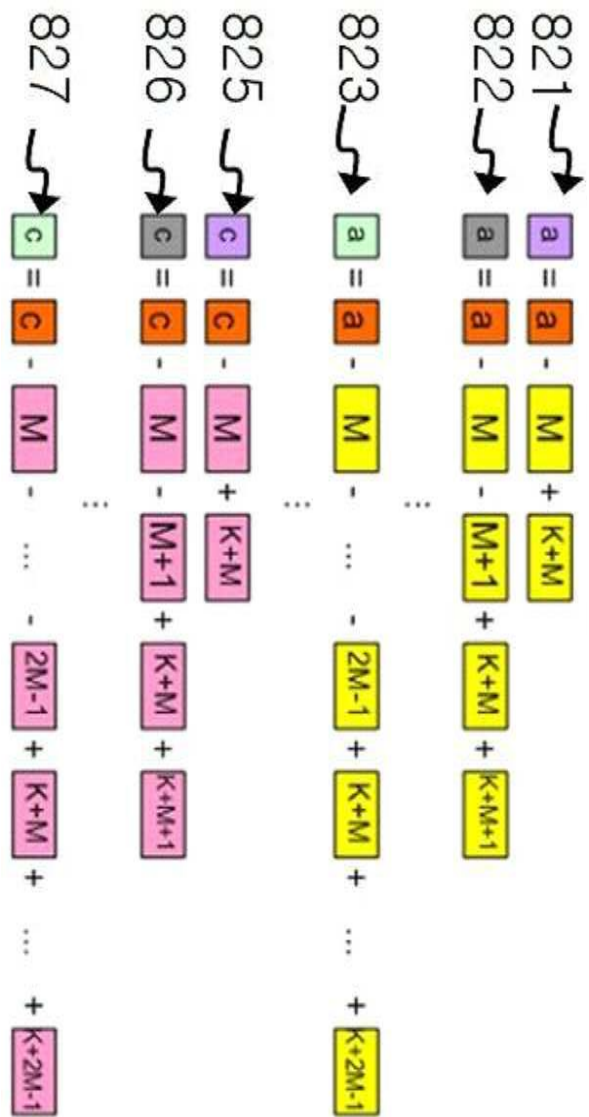
도면7d



도면7e



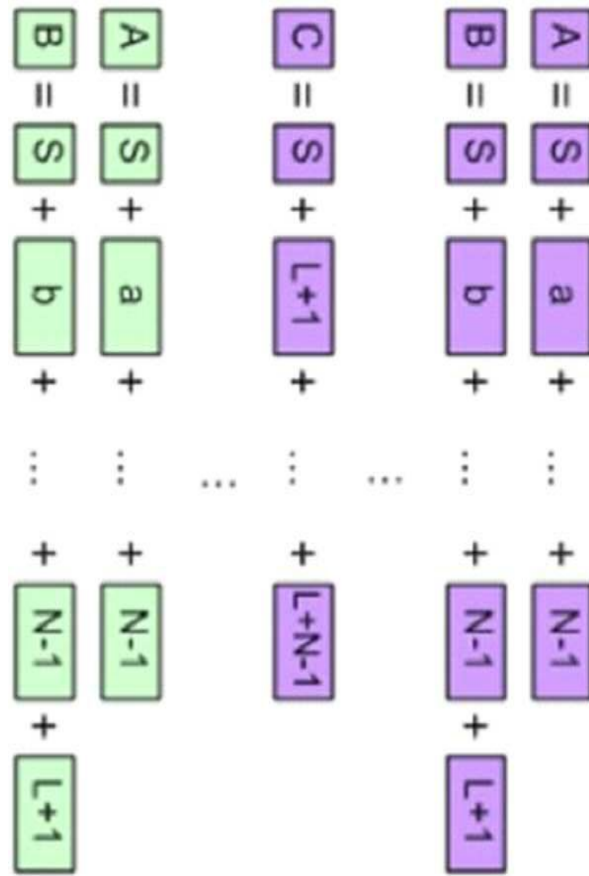
도면8b



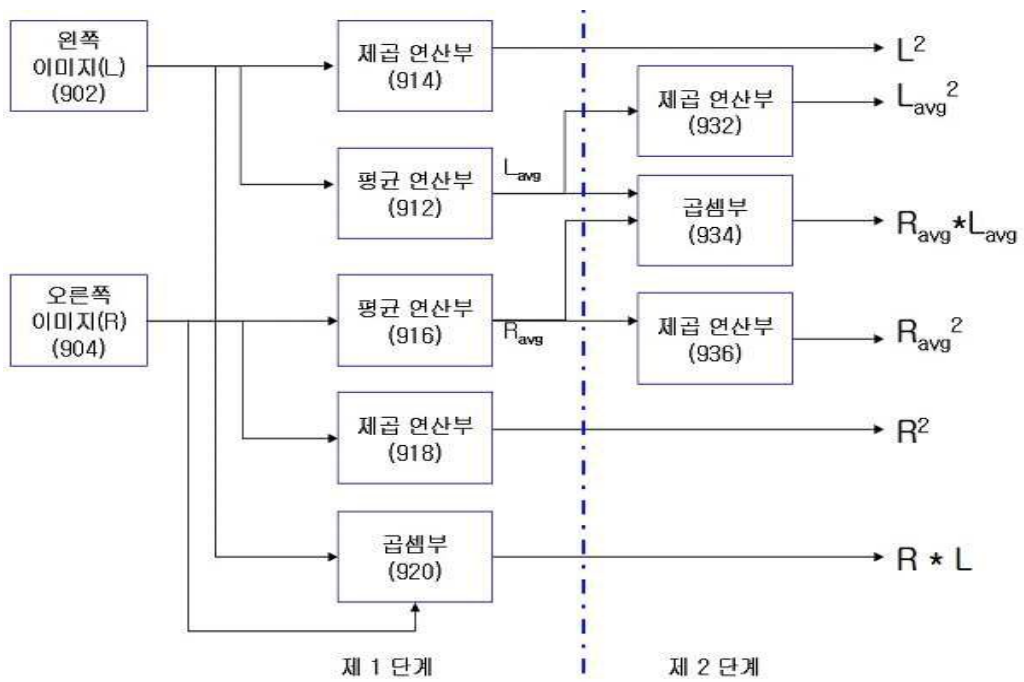
도면8c



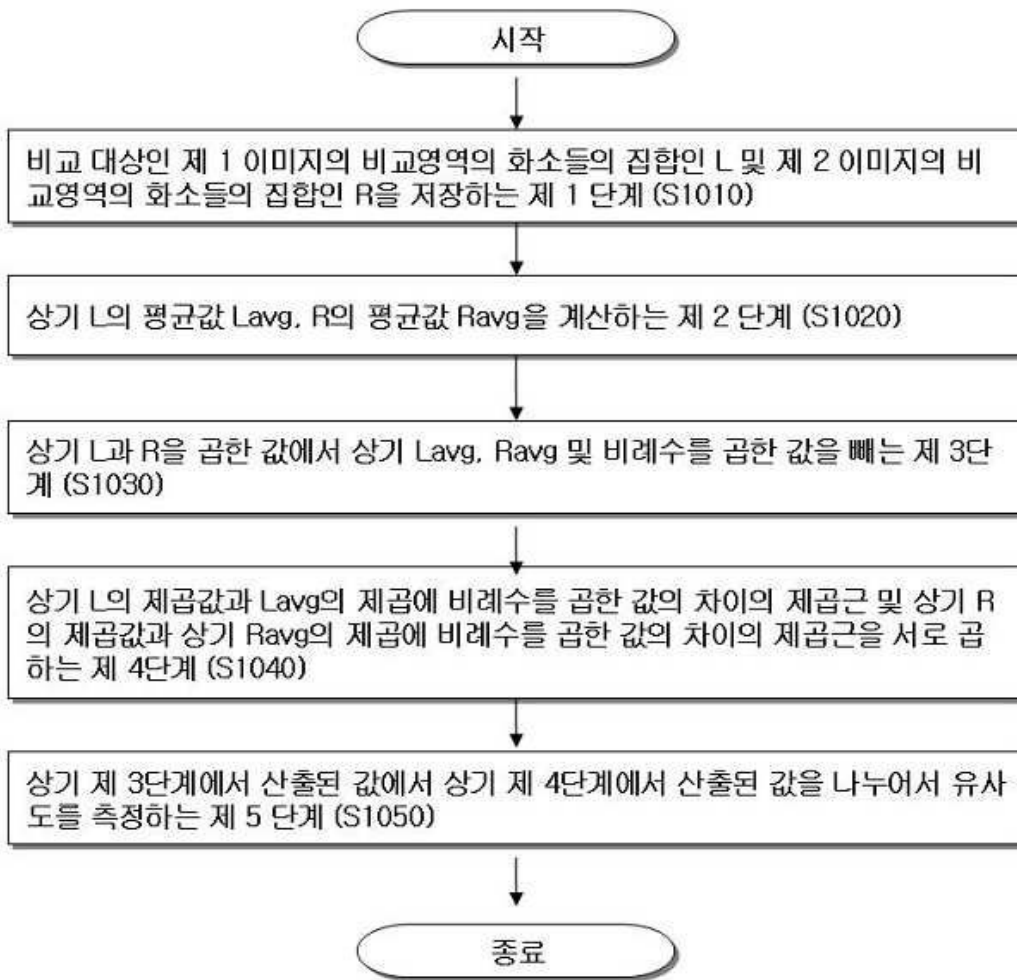
도면8d



도면9



도면10



도면 11

