



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년08월05일
(11) 등록번호 10-0973964
(24) 등록일자 2010년07월29일

(51) Int. Cl.

G06T 1/00 (2006.01)

(21) 출원번호 10-2009-0109477

(22) 출원일자 2009년11월13일

심사청구일자 2009년11월13일

(56) 선행기술조사문헌

JP11168725 A

KR100305190 B1

JP2005092362 A

KR100803045 B1

(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

김은구, 송해모

전체 청구항 수 : 총 8 항

심사관 : 권성호

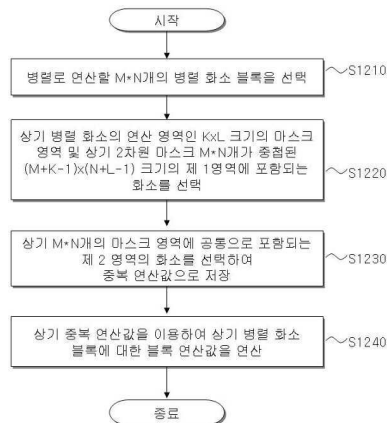
(54) 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 방법 및 장치

(57) 요약

본 발명은 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 방법 및 장치를 개시하고 있다.

본 발명의 일 실시 예에 따른 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 방법은 병렬로 연산할 $M \times N$ 크기의 병렬 화소 블록을 선택하는 제 1 단계, 상기 병렬 화소의 연산 영역인 $K \times L$ 크기의 마스크 영역 및 상기 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 1 영역에 포함되는 화소를 선택하는 제 2 단계, 상기 $M \times N$ 개의 마스크 영역에 공통으로 포함되는 제 2 영역의 화소를 선택하여 제 2 영역의 화소들에 대하여 연산한 결과를 중복 연산값으로 저장하는 제 3 단계, 및 상기 중복 연산값을 이용하여 상기 병렬 화소 블록에 대한 블록 연산값을 연산하는 제 4 단계를 포함한다.

대표도 - 도12



(72) 발명자

배경렬

대구광역시 동구 신암5동 94-31

문병인

서울특별시 은평구 응암1동 96-28 경향렉스빌 601
호

특허청구의 범위

청구항 1

병렬로 연산할 $M \times N$ 크기의 병렬 화소 블록을 선택하는 제 1 단계;

상기 병렬 화소의 연산 영역인 $K \times L$ 크기의 마스크 영역 및 상기 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 1 영역에 포함되는 화소를 선택하는 제 2 단계;

상기 $M \times N$ 개의 마스크 영역에 공통으로 포함되는 제 2 영역의 화소를 선택하여 제 2 영역의 화소들에 대하여 연산한 결과를 중복 연산값으로 저장하는 제 3 단계; 및

상기 중복 연산값을 이용하여 상기 병렬 화소 블록에 대한 블록 연산값을 연산하는 제 4 단계를 포함하는, 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 방법.

청구항 2

제 1항에 있어서,

상기 M 은 상기 병렬 화소 블록의 가로의 크기이며,

상기 N 이 상기 병렬 화소 블록의 세로의 크기이며,

상기 K 가 상기 마스크의 가로의 크기이며,

상기 L 이 상기 마스크의 세로의 크기이며,

상기 중복 연산값은 상기 제 1 영역의 좌측 경계 또는 우측 경계에서 가로로 M 번째 화소에서 K 번째 화소 또는 상기 제 1 영역의 상측 경계 또는 하측 경계에서 세로로 N 번째 화소에서 L 번째 화소를 포함하며,

상기 가로 또는 세로는 상기 제 1 영역의 경계를 기준으로 하는 것을 특징으로 하는, 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 4 단계는

상기 병렬 화소 블록 중 제 1 화소에 대응하는 제 1 마스크 영역을 선택하여 상기 제 1 마스크 영역에서 상기 제 2 영역에 포함되지 않는 행 방향의 $M-1$ 개의 화소 또는 열방향으로의 $N-1$ 개 화소를 상기 중복 연산값과 연산하는 단계를 더 포함하는, 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 방법.

청구항 4

제 1항에 있어서,

상기 제 4 단계 이후,

상기 제 1 영역 내에서 상기 제 1 단계의 병렬 화소 블록에 평행하며 가로로 M 또는 세로로 N 화소의 차이가 발생하는 병렬 화소 블록을 선택하는 제 5단계;

상기 제 5단계의 병렬 화소의 연산 영역인 $K \times L$ 크기의 마스크 영역 및 상기 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 3 영역에 포함되는 화소를 선택하는 제 6단계; 및

상기 제 1 영역과 중복되지 않으면서 상기 제 3 영역에 포함된 화소의 값을 상기 블록 연산값 또는 상기 블록 연산값을 산출하는 과정에서 발생하는 중간 연산값에 포함시키며,

상기 제 1 영역에 포함되며 상기 제 3 영역에 포함되지 않는 화소의 값을 상기 블록 연산값 또는 상기 블록 연산값을 산출하는 과정에서 발생하는 중간 연산값에서 제외시키는 제 7단계를 더 포함하는, 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 방법.

청구항 5

병렬로 연산할 $M \times N$ 크기의 병렬 화소 블록을 선택하여, 상기 병렬 화소의 연산영역인 $K \times L$ 크기의 마스크 영역 및 상기 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 1영역에 포함되는 화소를 선택하는 연산부;

상기 $M \times N$ 개의 마스크 영역에 공통으로 포함되는 제 2 영역의 화소를 선택하여 제 2영역의 화소들에 대하여 연산한 결과를 중복 연산값으로 저장하는 중복 연산 저장부; 및

상기 병렬 화소 블록에 대한 블록 연산값의 결과를 저장하는 블록 연산 저장부를 포함하며,

상기 연산부는 상기 중복 연산 저장부에 저장된 중복 연산값을 이용하여 상기 병렬 화소 블록에 대한 블록 연산값을 연산하여 상기 블록 연산저장부에 저장하는, 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 장치.

청구항 6

제 5항에 있어서,

상기 M 은 상기 병렬 화소 블록의 가로 크기이며,

상기 N 이 상기 병렬 화소 블록의 세로 크기이며,

상기 K 가 상기 마스크의 가로 크기이며,

상기 L 이 상기 마스크의 세로 크기이며,

상기 중복 연산값은 상기 제 1 영역의 좌측 경계 또는 우측 경계에서 가로로 M 번째 화소에서 K 번째 화소 또는 상기 제 1 영역의 상측 경계 또는 하측 경계에서 세로로 N 번째 화소에서 L 번째 화소를 포함하며,

상기 가로 또는 세로는 상기 제 1 영역의 경계를 기준으로 하는 것을 특징으로 하는, 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 장치.

청구항 7

제 5 항에 있어서,

상기 연산부는

상기 병렬 화소 블록 중 제 1 화소에 대응하는 제 1 마스크 영역을 선택하여 상기 제 1 마스크 영역에서 상기 제 2 영역에 포함되지 않는 행 방향의 $M-1$ 개의 화소 또는 열방향으로의 $N-1$ 개 화소를 상기 중복 연산 저장부에 저장된 중복 연산값과 연산하는 것을 특징으로 하는, 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 장치.

청구항 8

제 5항에 있어서,

상기 연산부는

상기 제 1 영역 내에서 상기 제 1 단계의 병렬 화소 블록에 평행하며 가로로 M 또는 세로로 N 화소의 차이가 발생하는 병렬 화소 블록을 선택하고, 상기 선택된 병렬 화소의 연산 영역인 $K \times L$ 크기의 마스크 영역 및 상기 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 3영역에 포함되는 화소를 선택하며,

상기 제 1 영역과 중복되지 않으면서 상기 제 3 영역에 포함된 화소의 값을

상기 블록 연산값 또는 상기 블록 연산값을 산출하는 과정에서 발생하는 중간 연산값에 포함시키며,

상기 제 1 영역에 포함되며 상기 제 3영역에 포함되지 않는 화소의 값을 상기 블록 연산값 또는 상기 블록 연산값을 산출하는 과정에서 발생하는 중간 연산값에서 제외시키는, 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 장치.

명세서

발명의 상세한 설명

기술분야

[0001] 본 발명은 영역 기반으로 영상을 처리하는 병렬 박스 필터링을 수행하는 방법 및 장치를 개시하고 있다.

배경기술

[0002] 영상처리에서 화소 단위의 데이터 처리 방법의 단점을 보완하기 위해 영역기반의 필터링과 관련된 방법들이 많이 제시되고 있다. 일례로 평균 필터의 경우 주어진 영역마스크 내 화소 데이터들의 평균값을 영역의 중심에 해당하는 화소의 값으로 대체를 하여 처리를 하는 방법이다. 또한 정규화된 상호상관기(Normalized Cross Correlation, 이하 NCC) 역시 영역 간의 상관도를 분석하기 위해 두 영역간의 화소에 대해 연산을 수행한다.

[0003] 이러한 영역기반의 데이터 처리 방법에 있어 영역내의 합 또는 평균을 구하는 과정이 필요한데, 영역의 크기가 크면 해당 영역에 포함되는 화소의 수가 증가하고, 그 결과 처리해야하는 화소의 수가 많아진다. 그 결과, 영역의 합 또는 평균을 구하기 위해 수행하는 연산 량이 영역 내의 화소의 수에 비례하여 증가하게 되어 연산에 많은 자원(시간, 프로세서 등)을 필요로 하는 문제가 발생하고 있다.

발명의 내용

해결하고자하는 과제

[0004] 본 발명은 영상정보와 같은 이차원 데이터 배열에서 일정한 크기의 마스크의 합을 마스크 크기에 상관없이 고속으로 구할 수 있는 방법인 박스 필터링 기법을 제안하여 병렬수행 구조를 통해 더욱 빠른 연산속도와 중복되는 연산 량을 최소화 하고자 한다.

[0005] 특히, 본 발명은 중복되는 연산은 별도로 저장하여, 해당 연산 결과를 재차 사용할 수 있도록 하여 연산 속도를 향상시키고자 한다.

과제 해결수단

[0006] 전술한 과제를 달성하기 위해, 본 발명의 일 실시 예에 따른 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 방법은 병렬로 연산할 $M \times N$ 크기의 병렬 화소 블록을 선택하는 제 1 단계, 상기 병렬 화소의 연산 영역인 $K \times L$ 크기의 마스크 영역 및 상기 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 1영역에 포함되는 화소를 선택하는 제 2 단계, 상기 $M \times N$ 개의 마스크 영역에 공통으로 포함되는 제 2 영역의 화소를 선택하여 제 2영역의 화소들에 대하여 연산한 결과를 중복 연산값으로 저장하는 제 3 단계, 및 상기 중복 연산값을 이용하여 상기 병렬 화소 블록에 대한 블록 연산값을 연산하는 제 4 단계를 포함한다.

본 발명의 다른 실시 예에 따른 영역 기반 영상 처리에서 병렬적인 박스 필터링을 수행하는 장치는 병렬로 연산할 $M \times N$ 크기의 병렬 화소 블록을 선택하여, 상기 병렬 화소의 연산영역인 $K \times L$ 크기의 마스크 영역 및 상기 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 1영역에 포함되는 화소를 선택하는 연산부, 상기 $M \times N$ 개의 마스크 영역에 공통으로 포함되는 제 2 영역의 화소를 선택하여 제 2영역의 화소들에 대하여 연산한 결과를 중복 연산값으로 저장하는 중복 연산 저장부, 및 상기 병렬 화소 블록에 대한 블록 연산값의 결과를 저장하는 블록 연산 저장부를 포함하며, 상기 연산부는 상기 중복 연산 저장부에 저장된 중복 연산값을 이용하여 상기 병렬 화소 블록에 대한 블록 연산값을 연산하여 상기 블록 연산저장부에 저장한다.

[0007] 삭제

효과

[0008] 본 발명은 컴퓨터 비전 기반 영상처리 시스템의 응용에 있어 적용 가능한 영역기반 데이터 처리 또는 필터링 방법에 있어 마스크 영역 내의 화소들의 합 또는 평균을 박스 필터링(box-filtering)기반의 방법을 통해 빠른 속

도로 연산하는 효과가 있다.

[0009] 특히, 블록단위의 화소를 동시에 박스 필터링(box-filtering)하며, 중복되는 연산은 1회 처리하여 연산 속도를 향상시키는 효과가 있다.

발명의 실시를 위한 구체적인 내용

[0010] 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.

[0011] 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 구성 요소에 "연결", "결합" 또는 "접속"된다고 기재된 경우, 그 구성 요소는 그 다른 구성 요소에 직접적으로 연결되거나 접속될 수 있지만, 각 구성 요소 사이에 또 다른 구성 요소가 "연결", "결합" 또는 "접속"될 수도 있다고 이해되어야 할 것이다.

[0012] 도 1a, 1b, 1c 및 1d는 종래에 영역 기반 데이터 처리 과정을 보여주는 도면이다.

[0013] 설명에 앞서 본 발명에서 사용하는 용어에 대하여 다음과 같이 정의를 내리고자 한다.

[0014] 데이터 처리 과정에서 더하기 연산(가산 연산)에 참여하는 화소를 '가산 화소'이라고 하고, 빼기 연산(감산 연산)에 참여하는 화소를 '감산 화소'이라 한다. 한편, 행 방향의 화소값을 더한 값은 '행 가산값'이라 하고, 열 방향의 화소값을 더한 값을 '열 가산값'이라고 한다. 한편, $A \times B$ 크기의 화소들의 모든 화소값을 더한 값을 '블록 가산값'이라고 한다. 본 발명의 일 실시예를 설명하는 과정에서 연산의 일 실시예로 가산 및 감산 연산을 제시하고 있으나, 본 발명이 이에 한정되는 것은 아니며, 가산/감산 및 평균, 편차 등 다양한 연산에 적용될 수 있다.

[0015] 도 1a에서 연산을 수행하는 블록의 크기는 3×3 이며, 3×3 의 범위 내에 존재하는 화소들 중에서 3개 행의 화소들에 대해 행 가산값을 구하는 과정이 도시되어 있다. 연산이 이루어지는 블록은 윈도우 마스크(112)로 표시되어 있다. 3개의 행 가산값의 합은 블록 가산값(110)에 저장된다.

[0016] 도 1b는 블록 연산을 수행하는 윈도우 마스크를 오른쪽으로 한 화소 크기만큼 이동시킨 경우의 연산 과정을 보여주는 도면이다.

[0017] 도 1b에서는 도 1a와 구별되도록 윈도우 마스크(128)가 오른쪽으로 한 화소 이동하여 연산을 수행하고 있다. 행 가산값을 구하는 과정은 마스크 내의 행 방향의 화소값을 모두 더할 수도 있으나, 도 1a 과정에서 연산한 계산값을 이용하기 위해 도 1a의 행 가산값을 참조한다.

[0018] 즉, 행 가산값을 구하는 과정(121)은 도 1a에서 행 가산을 수행하여 얻은 결과(124)에서 새로 추가된 화소의 값(132)을 더하고, 윈도우 마스크(128)에 포함되지 않는 화소의 값(133)을 빼는 과정으로 구성된다. 이 과정은 122, 123에서도 동일하게 진행된다. 그리고, 새로운 블록 가산값(120)은 121, 122, 123 과정에서 산출된 값을 모두 더하여 구할 수 있다.

[0019] 도 1c는 블록 연산을 수행하는 윈도우 마스크가 아래 쪽으로 이동을 완료한 경우, 아래 방향으로 한 화소 크기만큼 이동시킨 경우의 연산 과정을 보여주는 도면이다. 윈도우 마스크(141)가 도 1a의 윈도우 마스크(112)보다 아래 방향으로 한 화소 크기만큼 이동하였으므로, 하나의 행이 추가된 상황이다. 따라서, 첫번째 행과 두번째 행의 행 가산값은 도 1a의 114, 115 값을 그대로 이용하게 되며, 별도의 연산이 필요없으며, 새롭게 추가된 행에 대한 행 가산값(142)이 143과 같이 계산된다. 블록 가산값(145)은 144 과정과 같이 도 1a에서 계산되었던 블

록 가산값(146)을 이용하여 산출할 수 있다.

- [0020] 도 1d는 도 1c의 윈도우 마스크가 오른쪽으로 이동을 완료한 경우의 연산 과정을 보여주는 도면이다. 윈도우 마스크(151)가 도 1c의 윈도우 마스크(141)보다 오른쪽으로 한 화소 크기만큼 이동하였다. 도 1a, 1b, 1c의 과정에서 구한 결과를 사용하고, 새로이 추가된 열의 마지막 화소(151)만 계산하여 블록 가산값을 계산한다.
- [0021] 도 1a, 1b, 1c, 1d에서 영역의 크기를 결정하는 윈도우 마스크 크기가 클 경우 적용하고자 하는 필터의 처리속도 저하로 인해 실시간 성능을 요구하는 응용시스템에서 적용하기가 어려워진다.
- [0022] 이러한 윈도우 마스크 기반의 데이터 처리기법의 경우 비교하고자 하는 영역이 전체 이차원 영상영역에서 순차적으로 이동하는 경우 박스 필터링(Box-filtering) 기법을 적용하면 영역의 합을 연산하기 위해 연산 횟수를 줄일 수 있다.
- [0023] 도 1a, 1b, 1c, 1d는 이차원(2D) 데이터구조를 가지는 영상의 윈도우 마스크에 대해 박스 필터링(box-filtering)에 의해 연산을 수행하는 과정을 나타낸다. $K \times L$ 윈도우 마스크 내의 화소값의 합을 연산하는 데 있어 $K \times L$ 번의 덧셈이 필요하였으나 앞서 살펴본 방식으로 기술된 2차원 박스 필터링(2D box-filtering)을 적용하게 되면 대부분의 마스크에서는 4번의 연산(2번의 덧셈, 2번의 뺄셈)만으로 합(또는 평균)을 구할 수 있으므로 상당량의 연산비용 및 속도를 개선할 수 있음을 알 수 있다.
- [0024] 그러나, 여전히 초당 수십 프레임의 고해상도 영상이 출력되는 시스템 및 윈도우 마스크 탐색범위가 큰 경우에 있어서 순차적인 연산방식으로는 처리요구속도를 감당하기가 쉽지 않다. 이를 극복하기 위해 본 발명에서는 박스 필터링(Box-filtering)방법의 적용에 있어 동시에 여러 화소를 블록단위로 그룹핑하여 병렬 적용하는 방법을 제안하고 그 동작 구조를 설명하고자 한다.
- [0025] 도 2는 본 발명의 일 실시예에 의한 마스크와 블록간의 상관 관계를 보여주는 도면이다.
- [0026] 인접한 $K \times L$ 개의 화소들(211)을 중심으로 한 4 개의 윈도우 마스크들(210, 220, 230, 240)을 하나의 블록(250)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.
- [0027] 도 3은 본 발명의 일 실시예에 의한 이차원 데이터 배열의 구성을 보여주는 도면이다.
- [0028] 도 3은 $M \times N$ 의 크기를 가지는 화소 블록(330)에 대해 병렬로 동시에 연산을 수행한다. 화소 블록(330) 중 제 1화소(331)에 대해 주변 화소들을 계산하게 되는데, 주변 화소들의 범위, 즉 제 1 화소에 대한 마스크(311)의 범위는 도 3에서 파란 색으로 표시한 바와 같다. 마스크(311)의 크기는 $K \times L$ 이 된다. 그리고 제 1화소(331)에 대한 마스크(311) 내에 포함된 화소들을 계산한 연산 결과(가산 결과)는 320의 첫번째 영역인 321에 저장된다. 화소 블록(330)의 크기가 $M \times N$ 이므로, 연산을 수행한 결과 역시 $M \times N$ 이 된다. 또한, 화소 블록(330)의 $M \times N$ 개의 화소들에 대하여 각각 마스크 영역이 존재하므로, 해당 화소 블록(330)의 마스크는 모두 $M \times N$ 개이며, 이들 마스크들의 영역을 모두 합치면 310과 같이 된다. 마스크들은 서로 중첩되므로, 병렬 화소 블록(330)에 대한 모든 마스크들의 영역을 합친 영역(310)의 크기는 $(K+M-1) \times (L+N-1)$ 이 된다.
- [0029] 다시 정리하면, 가산 연산에서 계산되는 결과 값(320)들은 $M \times N$ 개가 있으며 320과 같이 저장되어 있다. 병렬 화소 블록의 크기(310) $M \times N$ 과 같이 각각의 마스크들의 갯수가 $M \times N$ 개이며, 마스크의 크기는 $K \times L$ 이다. 따라서, $M \times N$ 개의 $K \times L$ 크기 마스크로 구성되는 마스크 영역의 크기는 $(M + K - 1) \times (N + L - 1)$ 이 된다.
- [0030] 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.
- [0031] 본 발명을 도3에서 정의된 이차원 데이터배열 영역에 적용하는 과정은 다음과 같은 세 부류의 서로 다른 연산과

정을 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.

- [0032] 도 4a, 4b, 4c, 4d, 4e는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0033] 도 4a는 본 발명의 일 실시예에 의한 연산이 적용될 화소들의 배열을 보여주는 도면이다. 처음 연산이 수행되는 병렬 화소 블록(405)은 이전에 계산해 둔 중간 값, 전체 합이 없으므로 블록에 해당하는 마스크 전체 영역(410)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.
- [0034] 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.
- [0035] 도 4b는 본 발명의 일 실시예에 의한 중복하여 연산을 진행하는 행들에 대하여 미리 연산을 수행하는 과정을 보여주는 도면이다.
- [0036] 중복되는 연산을 제거하기 위해 $K \times L$ 마스크 내 동일 열에 속한 화소들의 가로 합(행가산값)을 구하는 과정으로, 반복되는 동일 열($M, M+1, \dots, K-1, K$)에 속한 화소들의 행 가산값을 S 에 저장한다.
- [0037] 도 4c는 본 발명의 일 실시예에 의한 중복 연산을 진행하는 행의 행 가산값을 이용하여 연산을 수행하는 과정을 보여주는 도면이다.
- [0038] 도 4a의 전체 행 가산값을 구하는 과정이 도시되어 있다. 431과 같이 도 4b에서 미리 연산한 S 및 S 에 포함되지 않는 화소들(1, 2, ..., $M-1$)들을 더하여 구한다. 432, 433, 435, 436, 437 역시 같은 방식으로 하여 이전에 계산하였던 S 에 포함되지 않는 화소들의 값을 더하는 방식으로 계산된다. 도 4c에서 a, b, c는 각각 마스크 전체 영역(410)에서 첫번째 행, 두번째 행, 세번째 행에 대해 연산 과정을 보여주고 있으나, 도 4a에서 알 수 있듯이, 마스크 전체 영역(410)에는 많은 행이 포함되어 있으므로, 도면 및 설명에서 생략하였으나, 네번째 행, 다섯번째 행 등 $L+N+1$ 행(438, 439, 440)까지 동일한 과정으로 연산이 진행될 수 있다.
- [0039] 도 4d 및 도 4e는 블록 가산값을 연산하는 과정을 보여주는 도면이다.
- [0040] 도 4d는 본 발명의 일 실시예에 의한 중복 연산을 진행하는 열의 열 가산값을 이용하여 연산을 수행하는 과정을 보여주는 도면이다.
- [0041] 도 4b의 가로 행의 값을 구하는 과정과 같이 동일 행에 속한 화소들의 세로 합(열가산값)을 구하는 과정에서 반복되는 동일 행($N, N, N+1, \dots, L-1, L$)에 속한 화소들의 열 가산값을 S 에 저장한다.
- [0042] 도 4e는 중복되는 열 가산값인 S 를 사용하는 과정을 보여주는 도면이다. 451은 도 4d의 미리 연산한 S 및 S 에 포함되지 않는 화소들(a, a+1, ..., $N-1$)을 가산한다. 452, 453, 455, 456, 457, 458, 459, 460 모두 미리 연산한 S 및 이에 포함되지 않는 화소들을 가산하는 과정을 보여주고 있다.
- [0043] 도 4a, 4b, 4c, 4d, 4e에서 S 로 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 S 를 구하는데 소요되는 덧셈연산이 $(K-M)$ 번이고 이 값을 한 번만 구해놓게 되면 나머지 각 가로 합의 연산에서는 M 개의 덧셈연산으로 가로 합을 연산할 수 있게 된다.
- [0044] 마찬가지로 세로 합을 연산하는 데도 $(L-N)$ 개의 덧셈을 통해 한번만 S 를 미리 구해놓으면 나머지 각 세로합의 연산에서는 N 번의 덧셈만으로도 가능하다.
- [0045] 도 5a, 5b, 5c, 5d는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다. 도 4a의 병

렬 화소 블록에 대한 연산이 완료된 후 가산결과(411)를 이용하여 열(가로) 진행 방향의 다음 블록을 연산하기 위한 과정이 제시되어 있다.

[0046] 다음 병렬 화소 블록을 계산하기 위해서는 가로로 M 화소만큼 블록을 이동 시킨다. 이렇게 이동시킨 블록(505)에 의한 마스크 전체 영역(510, 검은 색으로 경계지어짐)은 이동하기 전 블록의 마스크 전체 영역(410, 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.

[0047] 도 5b는 본 발명의 일 실시예에 의한 상세한 계산 과정을 보여주는 도면이다.

[0048] 521은 도 4c에서 계산하였던 값을 이용해서 계산하는 과정을 나타내고 있다. M은 마스크의 이동으로 제외된 화소의 값이며 K+M은 마스크의 이동으로 추가된 화소이다. 4c에서 계산한 값에서 제외된 화소의 값을 빼고, 추가된 화소의 값을 더하는 과정이 제시되어 있다. 522, 523, 525, 526, 527 모두 같은 방식으로 제외된 화소의 값을 빼고, 추가된 화소의 값을 더하는 과정이 제시되어 있다.

[0049] 도 5c는 본 발명의 일 실시예에 의한 중복 연산을 진행하는 열의 열 가산값을 이용하여 블록 연산을 수행하는 과정을 보여주는 도면이다. 앞서 살펴본 바와 같이, 도 4d에서 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.

[0050] 도 5d는 중복되는 열 가산값인 S를 사용하여 블록 가산값을 산출하는 과정을 보여주는 도면이다. 도 4e에서 설명한 과정과 동일하므로 도 4e의 설명으로 대체하고자 한다.

[0051] 도 6a, 6b, 6c, 6d는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다. 도 5a, 5b, 5c, 5d와 마찬가지로 도 4a의 가산결과(411)로부터 세로(행) 진행 방향에 대한 병렬 화소 블록의 연산을 진행하는 과정을 나타낸다. 병렬 화소 블록의 열(가로)진행방향 이동과 같이 행(세로) 진행방향으로 이동시킬 때에도 마찬가지로 방향만 다를 뿐, 가로로 적용한 방법을 똑같이 적용할 수 있다. 이전에 계산해 놓은 중간 값과 전체 합을 이용하여 추가로 들어오는 만큼의 화소를 더해주고 빠져나간 만큼의 화소를 빼주면 전체 합을 구할 수 있다. 세로로 블록을 N만큼 이동 시키면 그에 의한 마스크 전체 영역은 이동하기 전 블록의 마스크 전체 영역과 중복되는 구간이 생기며 그 결과, 추가로 마스크 내 N열만큼 새로운 구간이 늘어나고 N열만큼 빠져나가는 구간이 생긴다. 새로 들어온 가로줄의 합을 구하고 바로 윗 블록의 중간 값과 전체 합을 이용하여 새로운 블록의 전체 합을 구할 수 있다.

[0052] 도 6b는 중복되는 화소들의 값을 미리 계산하는 과정과 도 6c의 중복된 화소들의 값을 사용하여 추가되는 화소의 값을 가산하여 계산하는 과정은 도 4b, 4c와 동일하므로 도 4b, 4c의 설명으로 대체하고자 한다.

[0053] 도 6d는 본 발명의 일 실시예에 의한 블록 가산값을 산출하는 과정을 보여주는 도면이다. D는 모두 도 4e 과정에서 산출한 C를 참조하여 제외되는 화소(N)의 값을 빼고, 추가되는 화소(L+N)의 값을 더하는 과정을 진행하고 있다. E, F 역시 C 값을 참조하여 제외되는 화소의 값을 빼고, 추가되는 화소의 값을 더하는 과정을 통해 산출되고 있다.

[0054] 도 7a, 7b, 7c는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다. 새롭게 추가된 M개의 열의 화소들을 더하고, 역시 제외되는 M개 열의 화소의 값을 제외시키는 과정을 진행한다. 이 때 도 5a, 5b, 5c, 5d 및 도 6a, 6b, 6c, 6d 진행과정과의 차이점은 새로운 블록을 연산할 때 이전의 행 진행방향 가산결과 및 열 진행방향 가산결과를 모두 사용할 수 있다.

[0055] 도 7b는 본 발명의 일 실시예에 의한 행가산값을 구하는 과정을 보여주는 도면이다. 721은 이전에 구한 행가산

값에서 제외되는 화소(M)의 값을 빼고, 추가되는 화소(K+M)의 값을 더하는 과정이 제시되어 있다. 722, 723 역시 이전에 계산하였던 행 가산값을 참조하여 제외되는 화소의 값을 빼고, 추가되는 화소의 값을 더하는 과정을 통해 산출되고 있다.

- [0056] 도 7c는 본 발명의 일 실시예에 의한 블록 가산값을 연산하는 과정을 보여주는 도면이다. D는 도 5d에서 산출한 값인 C를 바탕으로 추가되는 화소(N)와 제외되는 화소(L+N)의 값을 더하는 과정을 진행한다. E, F 역시 C 값을 참조하여 제외되는 화소의 값을 빼고, 추가되는 화소의 값을 더하는 과정을 통해 산출되고 있다.
- [0057] 도7a, 7b, 7c를 참고하면 마스크의 이동에 따른 변화는 하단에 새로이 추가되었던 행에서 새로이 들어오는 열의 화소들을 더하고 역시 빠져나가는 열의 화소를 빼면 된다.
- [0058] 도 7b, 7c, 7d의 과정을 통해 병렬 화소 블록들 중에서 연산 대상이 되는 병렬 화소 블록들이 가로에 배열되면서 이전의 블록에 대한 마스크 전체 영역과도 중복되는 마스크 전체 영역이 생기지만 세로로 볼 때 위에 위치한 병렬 화소 블록의 마스크 전체 영역과도 중복되는 구간이 생기는 것을 알 수 있다. 결과적으로 M*N의 화소들이 추가되고 M*N의 화소를 제외함으로써 새로운 블록의 전체 연산을 구할 수 있다.
- [0059] 다음은 본 발명의 일 실시예에 의한 블록 연산이 수행되는 과정을 상세히 살펴보고자 한다.
- [0060] 도 8a, 8b, 8c는 도 4a, 4b, 4c, 4d, 4e를 구현한 것으로 첫번째 블록의 가산값을 연산하는 과정을 보여주는 도면이다.
- [0061] 도 8a는 첫번째 블록의 가산을 수행하기 위한 화소의 배치를 보여주는 도면이다. 앞서 살펴본 M, N, K, L의 값은 각각 3, 3, 11, 11이 된다. 병렬로 연산하고자 하는 병렬 화소 블록의 크기(MxN)는 3x3이고 병렬 화소 1개에 대한 마스크의 크기(KxL)는 11x11이다. 그 결과, 하나의 병렬 화소 블록(9개의 화소)에 대해 병렬로 연산되는 마스크들 역시 9개가 되며, 마스크 전체 영역의 크기((M+K-1)x(N+L-1))는 13x13이 된다.
- [0062] 도 8b는 도 4b, 4c에서 살펴본 바와 같이 본 발명의 일 실시예에 의해 중복되는 S값을 먼저 구한 후, 이를 이용하여 행가산값을 구하는 과정을 보여주는 도면이다.
- [0063] S는 총 9개(M-K)인 3, 4, ..., 11 화소의 값을 더한다. 그리고 이 값은 행 가산값 a, b, ..., m을 구하는데 반복하여 이용된다. 행 가산값 a, b, ..., m은 각각 새로이 추가되는 화소의 값을 S에 더하는 방식으로 구할 수 있다.
- [0064] 도 8c는 도 4d, 4e에서 살펴본 바와 같이 본 발명의 일 실시예에 의해 중복되는 S값을 먼저 구한 후, 이를 이용하여 블록 가산값을 구하는 과정을 보여주는 도면이다.
- [0065] 반복되는 행 가산값인 c, d, ..., k 까지의 값을 S에 저장한다. 그리고 블록 가산값 A, B, C는 S를 이용하여 추가되는 값을 각각 더하는 방식으로 산출한다. 예를 들어, A는 S에 a, b를 더하고, B는 S에 b, l을 더하며, C는 S에 l, m을 더하는 과정을 통해 구할 수 있다.
- [0066] 도 9a, 9b, 9c는 도 5a, 5b, 5c, 5d를 구현한 것으로 블록의 가로방향 진행블록 가산값을 연산하는 과정을 보여주는 도면이다.
- [0067] 도 9a는 도 8a에서 연산 대상인 병렬 화소 블록이 우측으로 이동한 형태이다. 따라서, 이에 따른 도 9a에서 연산할 병렬 화소 블록의 마스크 전체 영역은 도 8a의 마스크 전체 영역을 기준으로 1, 2, 3열의 화소들이 제외되고, 14, 15, 16열의 화소들이 추가된 상태이다. 따라서, 각 열의 가산값을 구하는 과정은 배제된 열의 화소의 값을 빼고, 추가된 열의 화소의 값을 더하는 과정으로 이루어진다.
- [0068] 도 9b는 도 5b, 5c 를 구현한 것으로 두번째 블록의 행 가산값을 산출하는 과정을 보여주는 도면이다.
- [0069] 도 8b에서 산출한 행 가산값(910)에서 제외되는 열(912)과 추가되는 열(914))의 화소 값을 각각 빼고 더하는 과정이 진행된다.
- [0070] 도 9c는 도 5d, 5e를 구현한 것으로 두번째 블록의 블록 가산값을 산출하는 과정을 보여주는 도면이다.
- [0071] 중복되는 S를 산출하기 위해 c, d, ..., j, k의 화소의 값을 더한다. 그리고 블록 가산값 A, B, C를 S를 이용하여 추가되는 값을 각각 더하는 방식으로 산출한다. 예를 들어, A는 S에 a, b를 더하고, B는 S에 b, l을 더하며,

C는 S에 1, m을 더하는 과정을 통해 구할 수 있다.

- [0072] 도 10a, 10b, 10c는 도 6a, 6b, 6c, 6d를 구현한 것으로 세 번째 블록의 블록 가산값을 연산하는 과정을 보여주는 도면이다.
- [0073] 도 10a는 도 8a와 비교하여 연산 대상인 병렬 화소 블록이 아래 방향으로 이동한 형태이다. 따라서, 이에 따른 도 10a에서 연산할 병렬 화소 블록의 마스크 전체 영역은 도 8a의 마스크 전체 영역을 기준으로 아래의 3개의 행이 추가되었으며, 위의 3개 행이 제외되었다. 따라서, 이를 참조하여 추가된 3개의 행에 대해서만 계산을 수행하여 행 가산값을 계산할 수 있다.
- [0074] 도 10b는 본 발명의 일 실시예에 의해 새롭게 추가된 행에 대하여 행 가산값을 구하는 과정을 보여주는 도면이다.
- [0075] 도 10b는 도 6b, 6c를 구현한 것으로 새롭게 추가된 3개 행의 행 가산값을 구하는 과정을 보여주고 있다. 중복되는 계산 부분 S를 별도로 계산하여 이용하는 것은 전술한 바와 같다.
- [0076] 도 10c는 도 6d를 구현한 것으로, 본 발명의 앞서 계산된 8c의 블록 가산값 C(1010)을 이용하며, 배제되는 행(1012)의 화소 값을 빼고, 추가되는 행(1014)의 화소 값을 더하여 계산할 수 있다.
- [0077] 도 11a, 11b, 11c는 도 7a, 7b, 7c를 구현한 것으로 네 번째 블록의 블록 가산값을 연산하는 과정을 보여주는 도면이다. 마스크의 이동에 따른 변화는 하단에 새로이 추가되었던 3개의 열 중에서 새로이 들어오는 3개의 행의 화소들(14, 15, 16)을 더하고 역시 빠져나가는 3개의 행의 화소(3, 4, 5)를 빼면 된다. 이는 블록이 가로로 이동하면서 이전에 연산한 블록의 마스크 전체 영역과도 중복되는 구간이 생기지만 세로로 볼 때 위의 블록의 마스크 전체 영역과도 중복되는 구간이 생기기 때문이다.
- [0078] 도 11b는 도 7b의 과정을 구현한 도면으로, 본 발명의 일 실시예에 따른 행 가산값을 구하는 과정을 보여주는 도면이다.
- [0079] 전술한 바와 같이, 행 가산값을 구하기 위해 도 10b에서 계산한 행 가산값(1110)에서 마스크의 이동으로 인해 제외되는 열(1112)의 화소 값을 빼고, 추가되는 열(1114)의 화소 값을 더한다.
- [0080] 도 11c는 도 7c를 구현한 것으로 본 발명의 일 실시예에 의한 블록 가산값을 구하는 과정을 보여주고 있다.
- [0081] 도 10c에서 계산한 블록 가산값(1120)에서 마스크의 이동으로 인해 제외되는 행(1122)의 화소 값을 빼고, 추가되는 행(1124)의 화소 값을 더한다.
- [0082] 앞서 살펴본 바와 같이 병렬 화소 블록에 대해 소정 크기의 마스크를 적용하여 블록 가산값을 계산하는 과정을 보았다. 블록 가산값은 블록 연산값의 일종으로 블록 가산값을 구하기 위해 마스크 범위 내에서의 행 가산, 열 가산 등은 모두 일 실시예에 해당하며, 가산/감산/평균 등 다양한 연산이 적용될 수 있다.
- [0083] 지금까지 살펴본 본 발명의 실시예에 의하여, 종래에는 $K \times L$ 마스크 내의 화소 값의 합을 $M \times N$ 개 블록으로 동시에 연산하는 데 $MNKL$ 번의 덧셈기가 필요하였으나 기술된 블록기반의 2차원 박스 필터링(2D box-filtering)을 적용하게 되면 대부분의 마스크에서는 $((M^2(M+1) + N^2(N+1))/2)$ 번의 덧셈기 및 뺄셈기가 필요하므로, 블록의 크기에 비해 윈도우 마스크가 클 경우 그 연산량 감소의 이득이 K 와 L 에 비례하여 개선됨을 알 수 있다.
- [0084] 더구나 한번에 $M \times N$ 화소의 합을 동시에 구할 수 있으므로 $M \times N$ 배의 처리속도 증가 이득을 얻을 수 있다.
- [0085] 도 12는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0086] 앞서 살펴본 바와 같이, $M \times N$ 크기의 2차원 배열에 블록 연산값을 저장하게 된다. 따라서 $M \times N$ 개의 마스크를 통해 계산되며, 이 마스크의 크기는 $K \times L$ 이다. 병렬로 연산할 $M \times N$ 개의 병렬 화소 블록을 선택한다(S1210). 그리

고, 상기 병렬 화소의 연산 영역인 $K \times L$ 크기의 마스크 영역 및 상기 2차원 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 1영역에 포함되는 화소를 선택한다(S1220). 이는 앞서 살펴본 바와 같이 병렬 화소 블록, 병렬 화소 각각의 화소에 해당하는 마스크 및 마스크 전체 영역을 설정하는 과정을 의미한다.

[0087] 그리고, 앞서 마스크 전체 영역에서 중복되는 영역인 "S" 영역을 계산하기 위해 상기 $M \times N$ 개의 마스크 영역에 공통으로 포함되는 제 2 영역의 화소를 선택하여 중복 연산값으로 저장한다(S1230).

[0088] 그리고 중복 연산값을 이용하여 상기 병렬 화소 블록에 대한 블록 연산값을 연산한다(S1240).

[0089] 상기 중복 연산값은 상기 제 1 영역에서 M번째 화소에서 K번째 화소 또는 N번째 화소에서 L 번째 화소를 포함하는데, 이 과정은 앞서 살펴본 바와 같다.

[0090] 블록 연산값을 계산하는 과정에 있어, 마스크들이 중첩되는 부분에 대해 연산을 줄이기 위하여, 상기 병렬 화소 블록 중 제 1 화소에 대응하는 제 1 마스크 영역을 선택하여 상기 제 1 마스크 영역에서 상기 제 2 영역(즉 중복 연산을 수행한 영역)에 포함되지 않는 행 방향의 M-1개의 화소 또는 열방향으로의 N-1개 화소를 상기 중복 연산값과 연산할 수 있다.

[0091] 하나의 병렬 화소 블록에 대해 연산이 완료되면, 인접한 다른 병렬 화소 블록에 대해서도 12의 과정을 반복한다. 단, 인접한 다른 병렬 화소 블록에서 계산한 값을 사용하므로, 도 12의 과정에서 일부 변경되는 과정이 존재할 수 있다.

[0092] 예를 들어, 상기 제 1 영역 내에서 상기 제 1 단계의 병렬 화소 블록에 평행하며 행방향 M 또는 열 방향으로 N 화소의 차이가 발생하는 병렬 화소 블록을 선택하여, 상기 제 5단계의 병렬 화소의 연산 영역인 $K \times L$ 크기의 마스크 영역 및 상기 2차원 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 3영역에 포함되는 화소를 선택한다. 새로운 병렬 화소 블록에 대해 새로운 마스크 전체 영역을 구할 수 있다. 그리고 상기 제 1 영역(이전 병렬 화소 블록의 마스크 전체 영역)과 상기 제 3 영역(새로운 병렬 화소 블록의 마스크 전체 영역)의 차이에 해당하는 화소들의 값을 상기 블록 연산값 또는 상기 블록 연산값을 산출하는 과정에서 포함시키거나 제외시킬 수 있다.

[0093] 보다 상세하게 살펴보면, 상기 제 1 영역과 중복되지 않으면서 상기 제 3 영역에 포함된 화소의 값을 상기 블록 연산값 또는 상기 블록 연산값을 산출하는 과정에서 발생하는 중간 연산값에서 제외시키며,

[0094] 상기 제 1 영역에 포함되며 상기 제 3영역에 포함되지 않는 화소의 값을 상기 블록 연산값 또는 상기 블록 연산값을 산출하는 과정에서 발생하는 중간 연산값에 포함시킬 수 있다. 본 발명의 일 실시예에 의한 병렬 박스 필터링을 위해서는 중복 가산값(중복 연산값)을 저장하는 중복 연산 저장부, 각각의 행별 또는 열별 연산 결과를 저장하는 중간 연산 저장부, 및 행별/열별 연산 결과를 취합한 블록 연산 결과를 저장하는 블록 연산 저장부가 필요하다.

[0095] 또한, 연산하고자 하는 영역 내의 중복 연산, 행별/열별 연산 및 블록 연산을 수행하는 연산부가 필요하다.

[0096] 보다 상세하게 살펴볼 때, 병렬로 연산할 $M \times N$ 개의 병렬 화소 블록을 선택하여, 상기 병렬 화소의 연산 영역인 $K \times L$ 크기의 마스크 영역 및 상기 2차원 마스크 $M \times N$ 개가 중첩된 $(M+K-1) \times (N+L-1)$ 크기의 제 1영역에 포함되는 화소를 선택하는 연산부와, 상기 $M \times N$ 개의 마스크 영역에 공통으로 포함되는 제 2 영역의 화소의 연산 결과를 중복 연산값으로 저장하는 중복 연산 저장부, 그리고 상기 병렬 화소 블록에 대한 블록 연산값의 결과를 저장하는 블록 연산 저장부로 구성된다. 또한 연산부는 상기 중복 연산 저장부에 저장된 중복 연산값을 이용하여 상기 병렬 화소 블록에 대한 블록 연산값을 연산하여 상기 블록 연산저장부에 저장한다. 중간 연산 저장부는 블록 연산값을 저장하는 과정에서 필요로 하는 저장부이다.

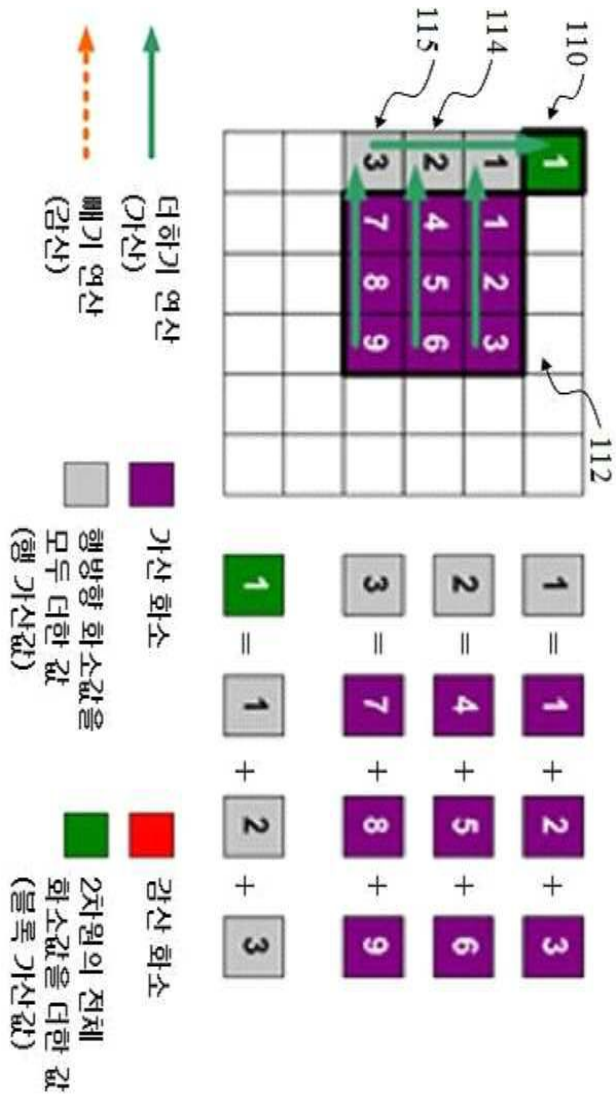
[0097] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 개시된 실시 예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시 예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면의 간단한 설명

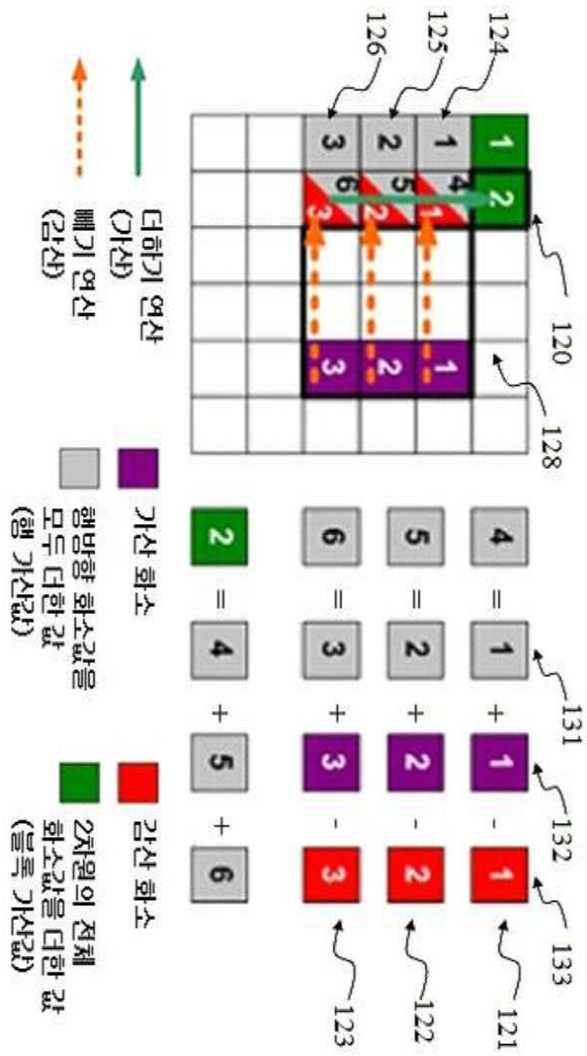
- [0098] 도 1a, 1b, 1c 및 1d는 종래에 영역 기반 데이터 처리 과정을 보여주는 도면이다.
- [0099] 도 2는 본 발명의 일 실시예에 의한 마스크와 블록간의 상관 관계를 보여주는 도면이다.
- [0100] 도 3은 본 발명의 일 실시예에 의한 이차원 데이터 배열의 구성을 보여주는 도면이다.
- [0101] 도 4a, 4b, 4c, 4d, 4e는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0102] 도 5a, 5b, 5c, 5d는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0103] 도 6a, 6b, 6c, 6d는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0104] 도 7a, 7b, 7c는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.
- [0105] 도 8a, 8b, 8c는 도 4a, 4b, 4c, 4d, 4e를 구현한 것으로 첫번째 블록의 가산값을 연산하는 과정을 보여주는 도면이다.
- [0106] 도 9a, 9b, 9c는 도 5a, 5b, 5c, 5d를 구현한 것으로 두 번째 블록의 블록 가산값을 연산하는 과정을 보여주는 도면이다.
- [0107] 도 10a, 10b, 10c는 도 6a, 6b, 6c, 6d를 구현한 것으로 세 번째 블록의 블록 가산값을 연산하는 과정을 보여주는 도면이다.
- [0108] 도 11a, 11b, 11c는 도 7a, 7b, 7c를 구현한 것으로 네번째 블록의 블록 가산값을 연산하는 과정을 보여주는 도면이다.
- [0109] 도 12는 본 발명의 일 실시예에 의한 블록 연산을 수행하는 과정을 보여주는 도면이다.

도면

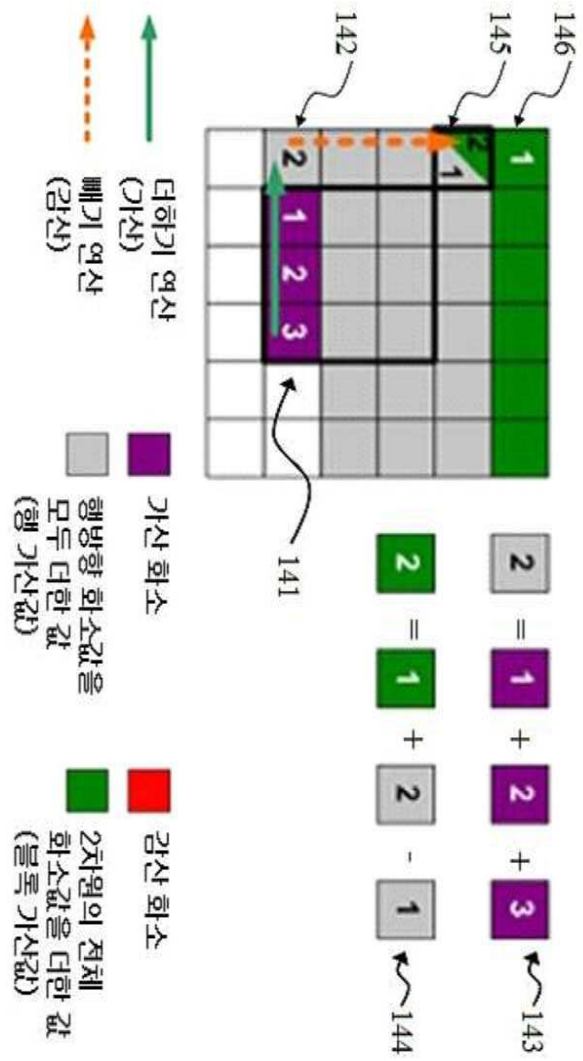
도면1a



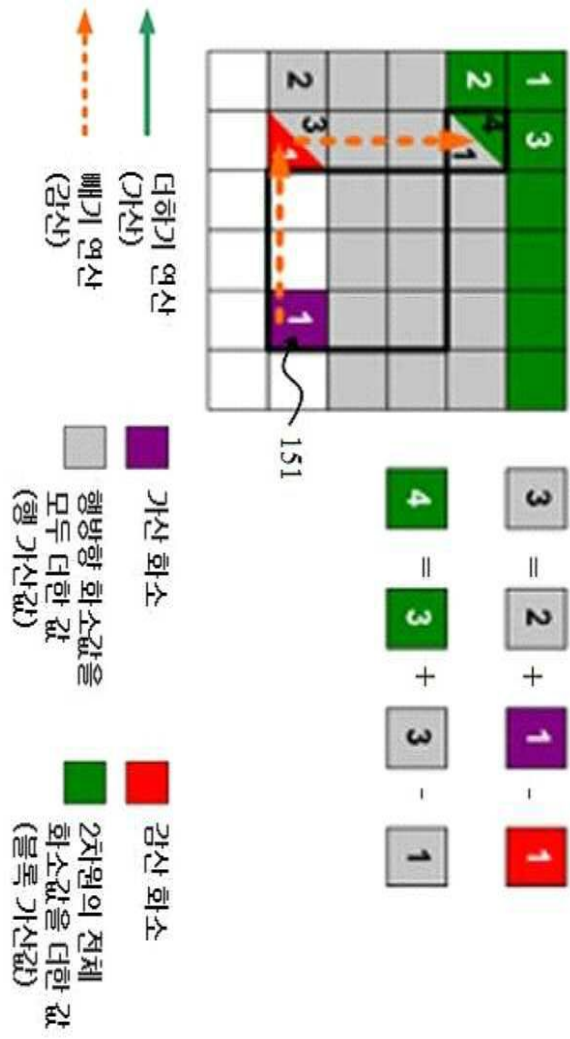
도면1b



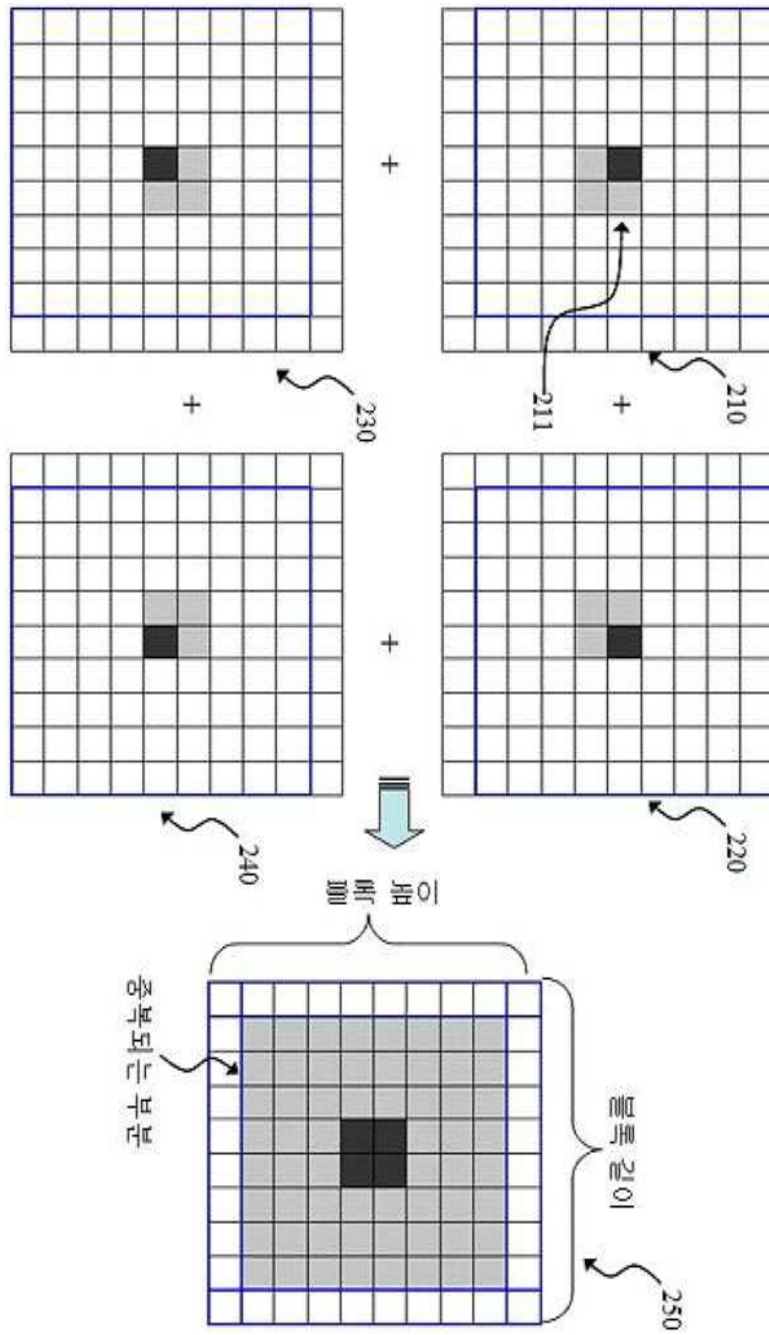
도면1c



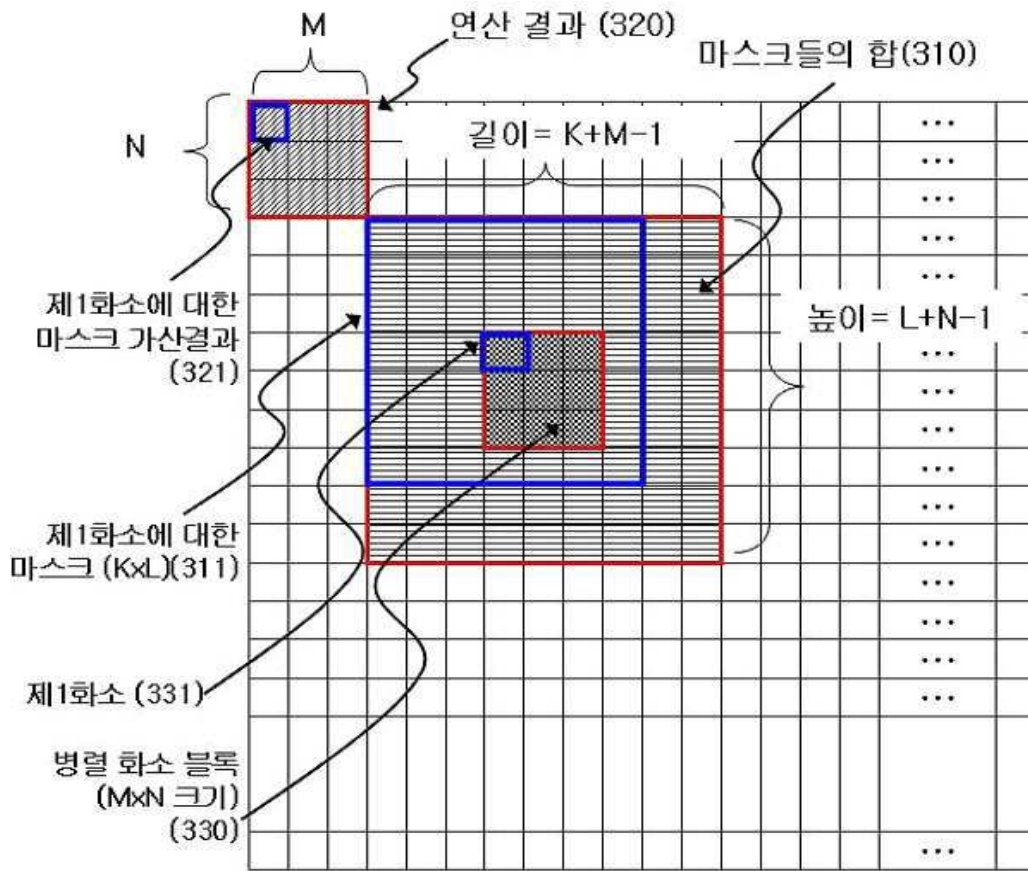
도면1d



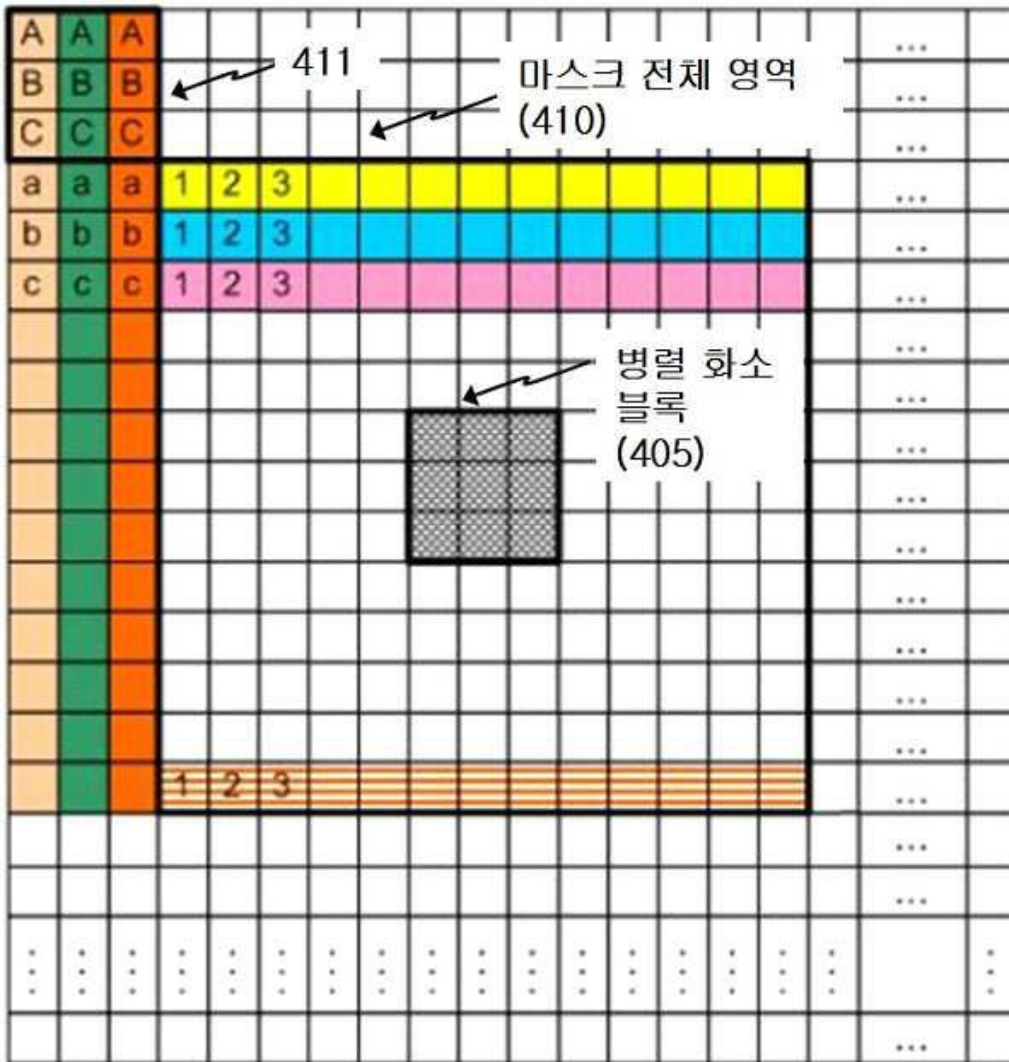
도면2



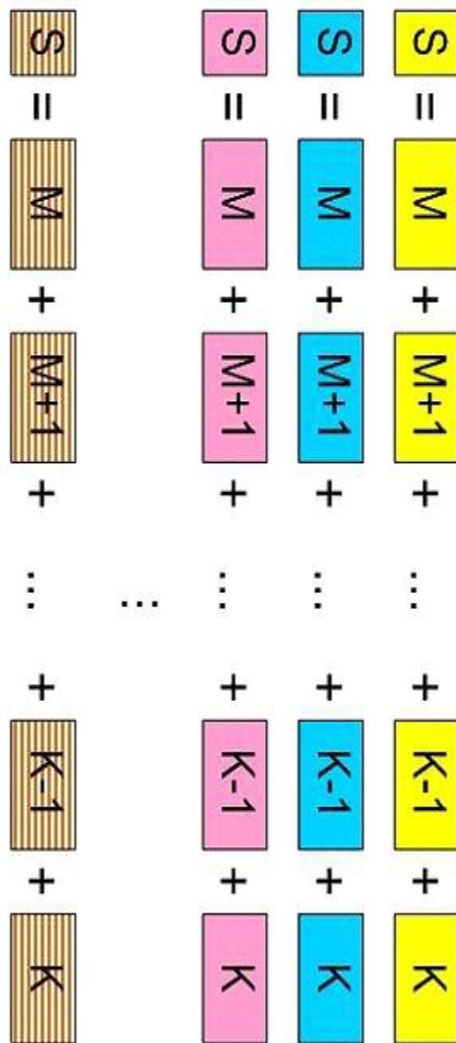
도면3



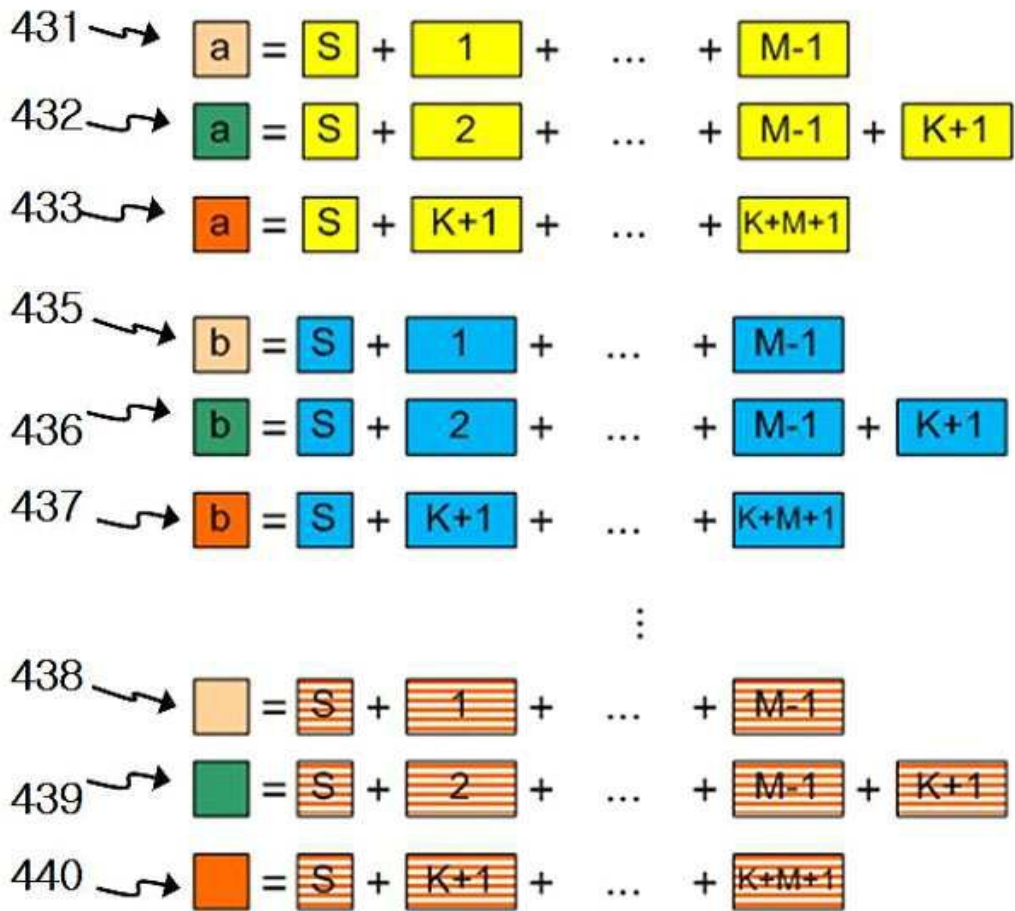
도면4a



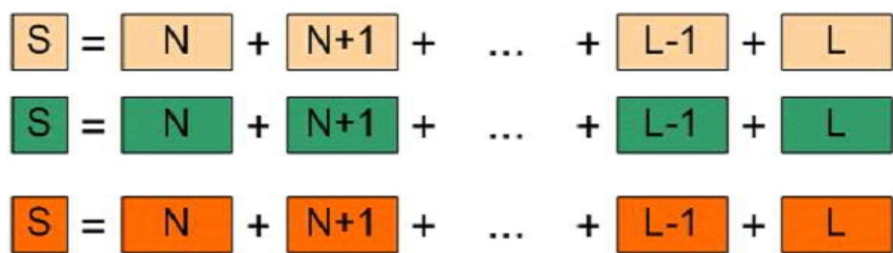
도면4b



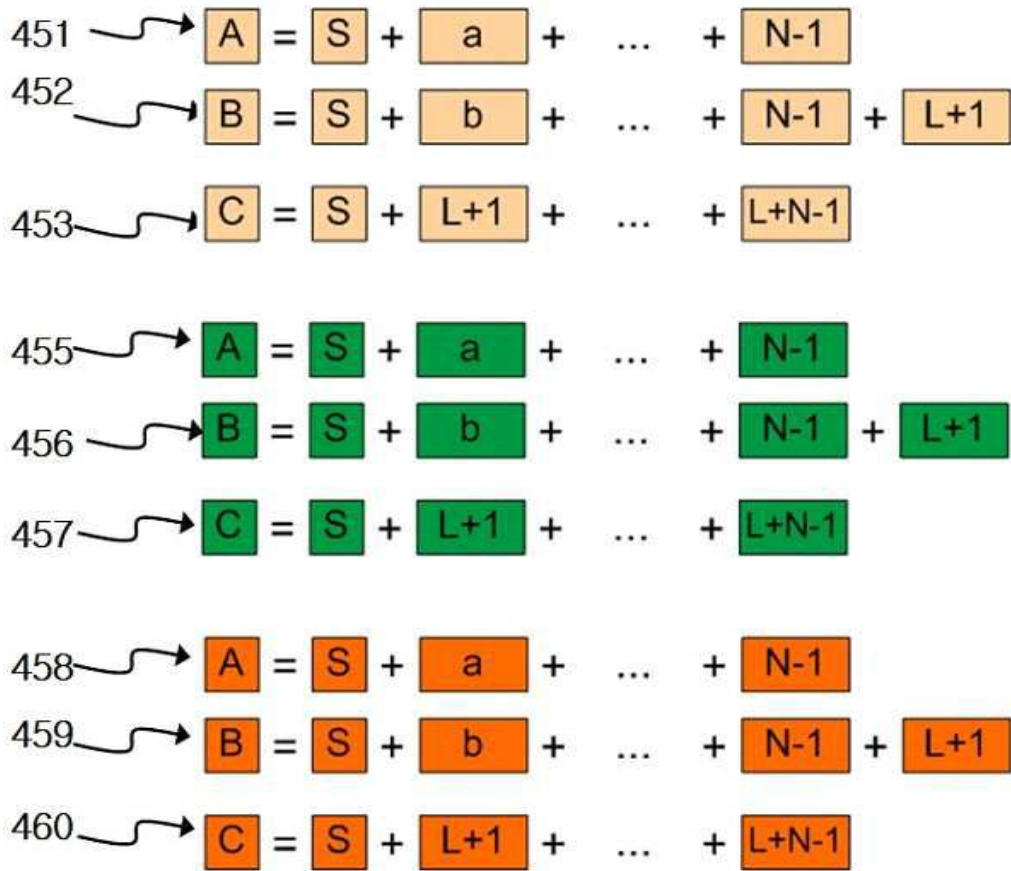
도면4c



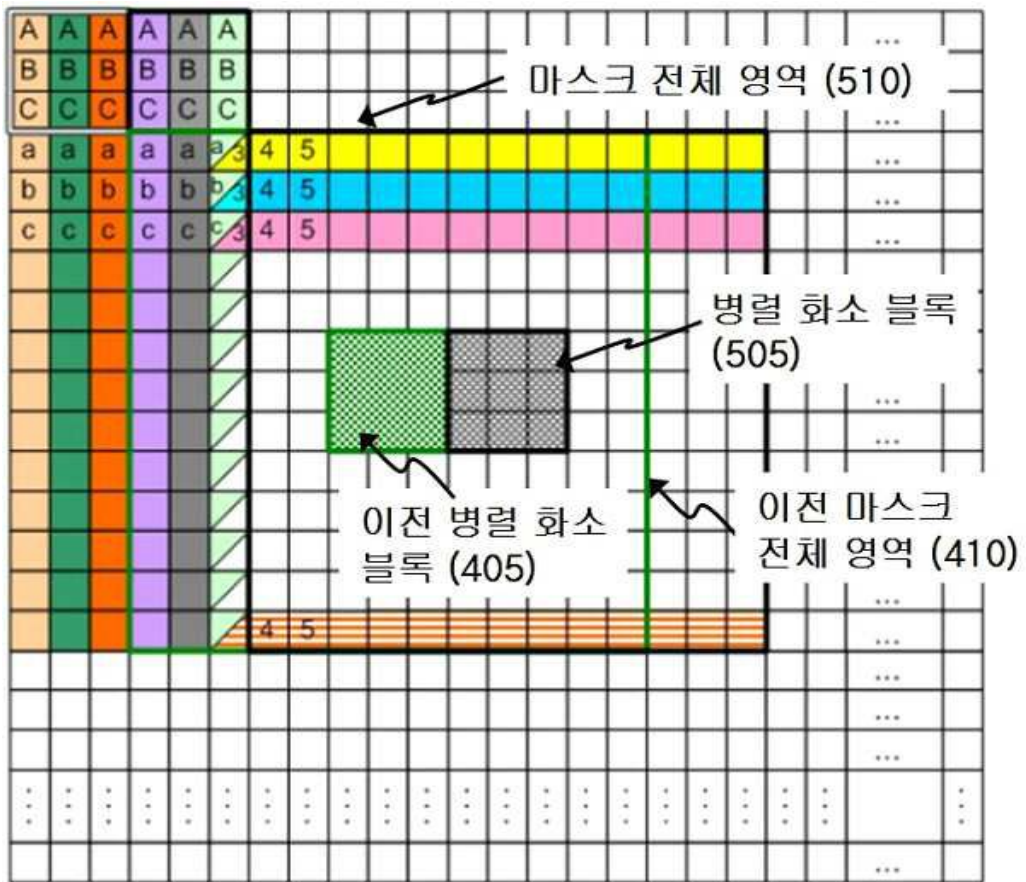
도면4d



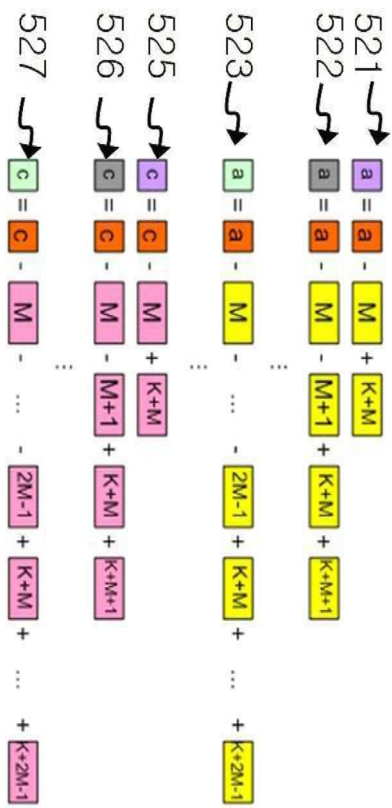
도면4e



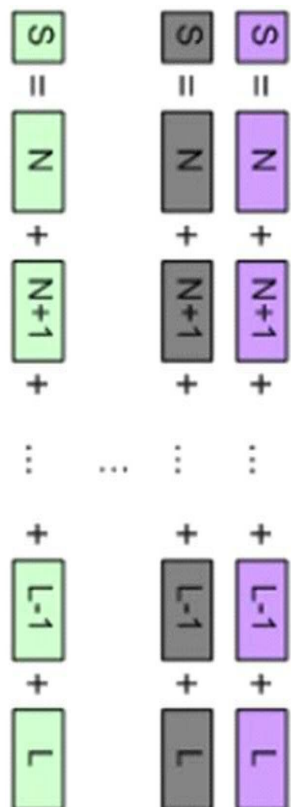
도면5a



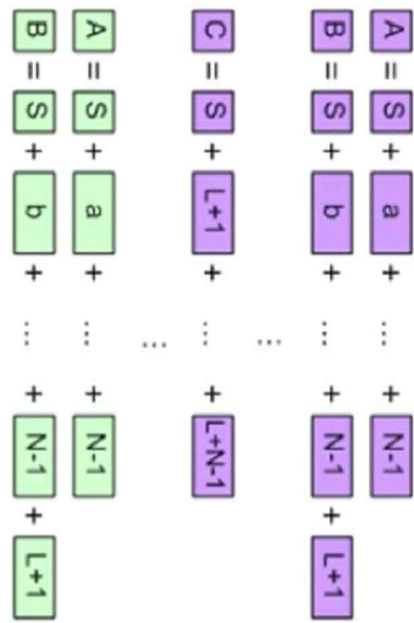
도면5b



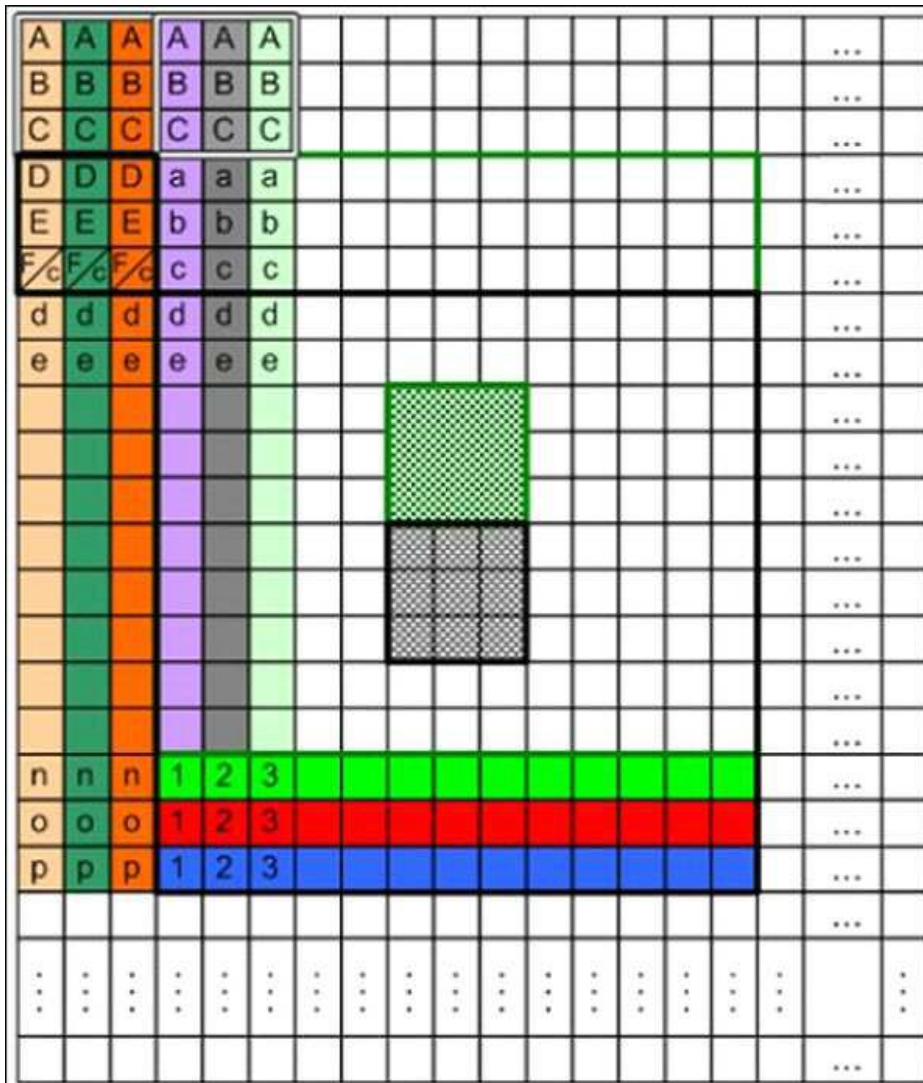
도면5c



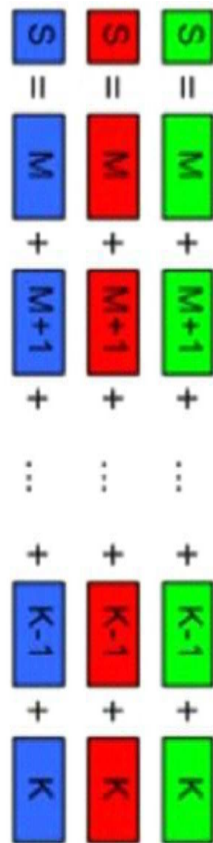
도면5d



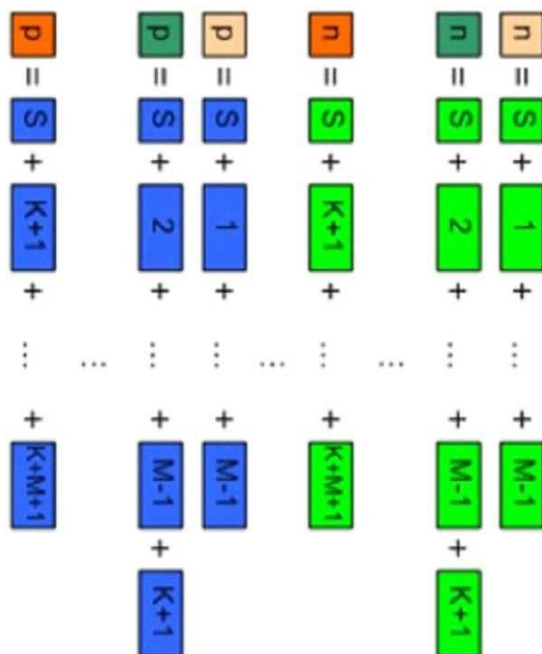
도면6a



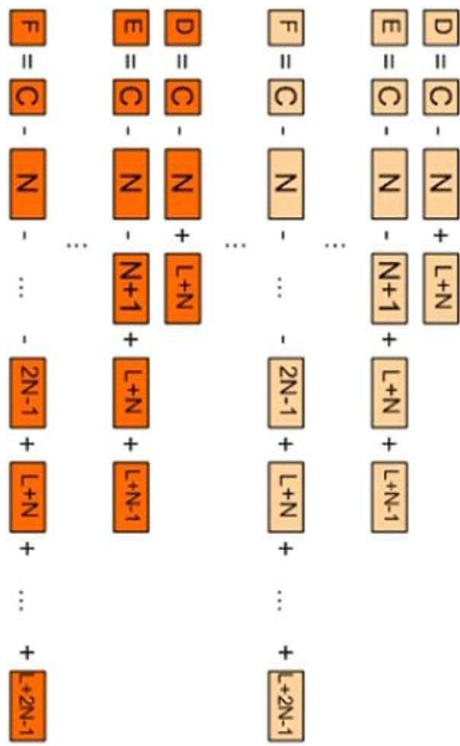
도면6b



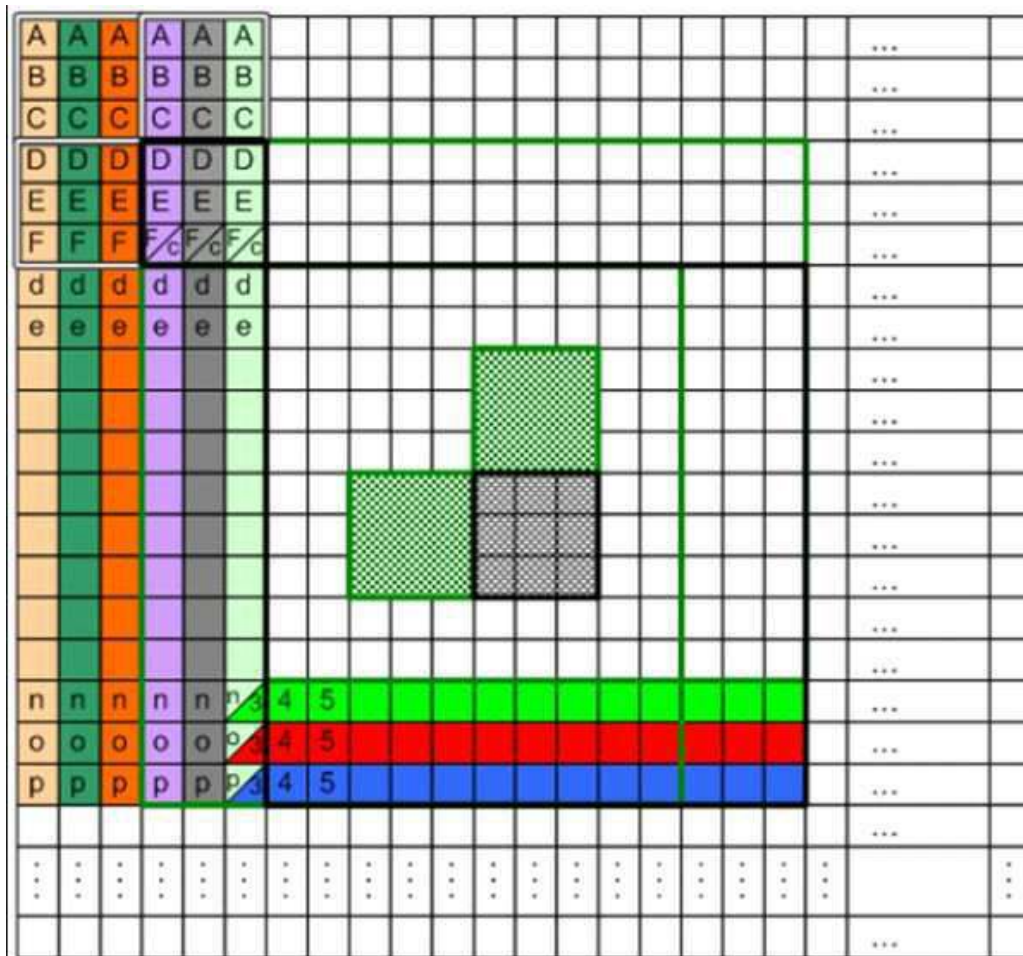
도면6c



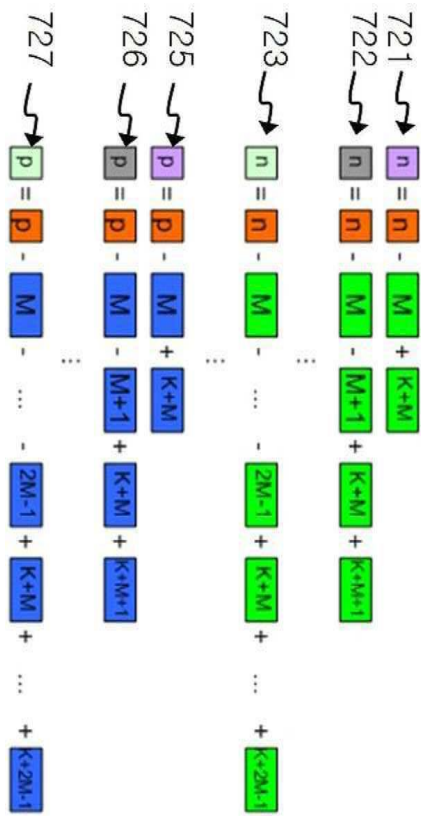
도면6d



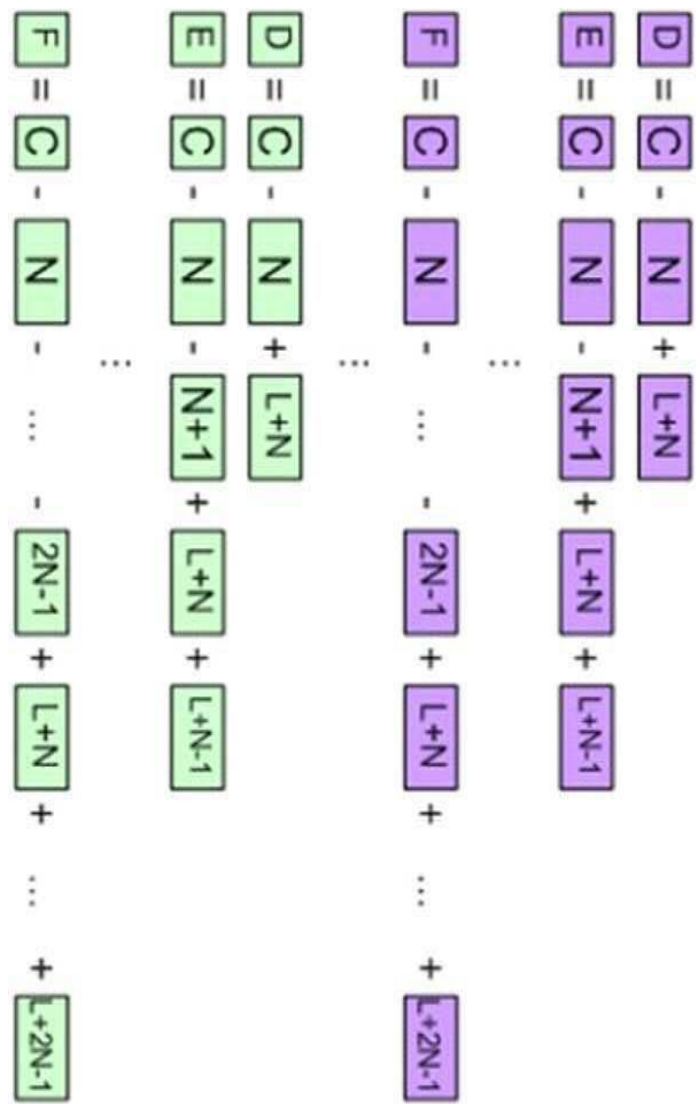
도면7a



도면7b



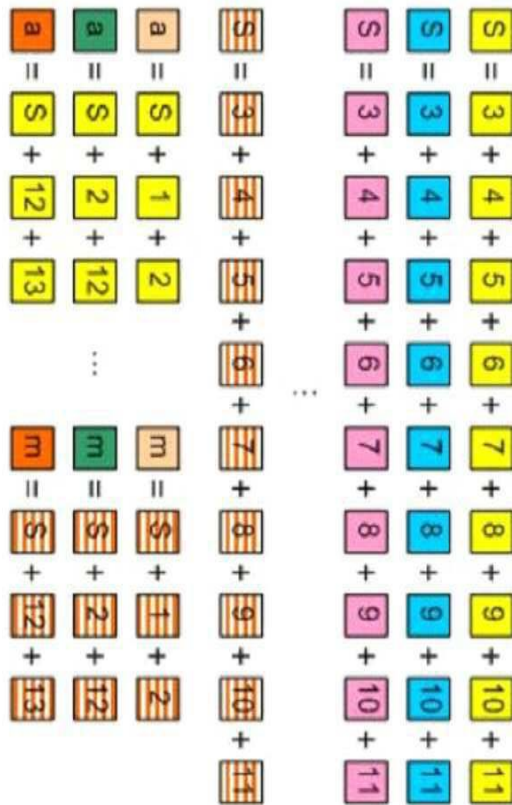
도면7c



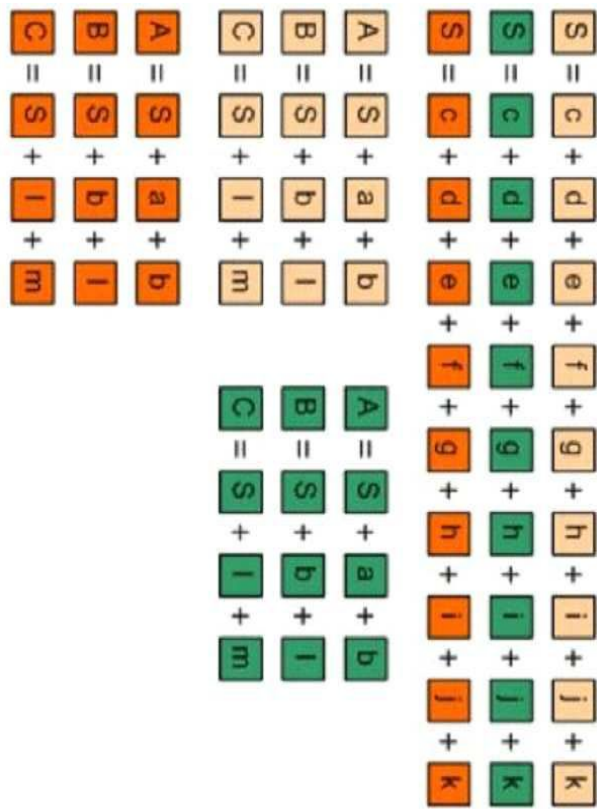
도면8a

| | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| A | A | A | | | | | | | | | | | | | | | | | ... |
| B | B | B | | | | | | | | | | | | | | | | | ... |
| C | C | C | | | | | | | | | | | | | | | | | ... |
| a | a | a | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | | | | ... |
| b | b | b | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | | | | ... |
| c | c | c | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | | | | ... |
| d | d | d | | | | | | | | | | | | | | | | | ... |
| e | e | e | | | | | | | | | | | | | | | | | ... |
| f | f | f | | | | | | | | | | | | | | | | | ... |
| g | g | g | | | | | | | | | | | | | | | | | ... |
| h | h | h | | | | | | | | | | | | | | | | | ... |
| i | i | i | | | | | | | | | | | | | | | | | ... |
| j | j | j | | | | | | | | | | | | | | | | | ... |
| k | k | k | | | | | | | | | | | | | | | | | ... |
| l | l | l | | | | | | | | | | | | | | | | | ... |
| m | m | m | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | | | | ... |
| | | | | | | | | | | | | | | | | | | | ... |
| | | | | | | | | | | | | | | | | | | | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| | | | | | | | | | | | | | | | | | | | ... |

도면8b



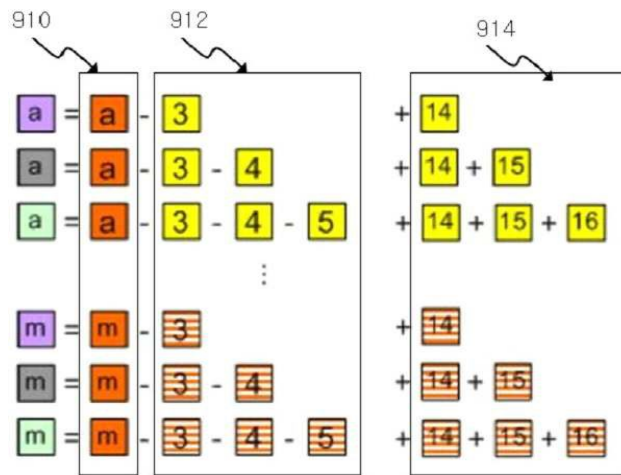
도면8c



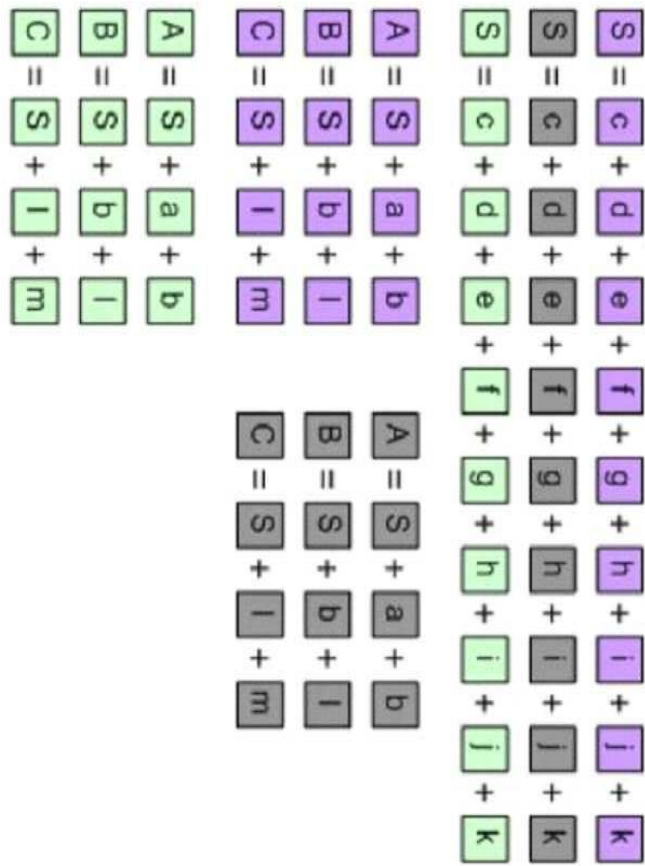
도면9a

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|----------------|---|---|---|---|---|---|----|----|----|----|----|----|-----|-----|
| A | A | A | A | A | A | | | | | | | | | | | | | ... | |
| B | B | B | B | B | B | | | | | | | | | | | | | ... | |
| C | C | C | C | C | C | | | | | | | | | | | | | ... | |
| a | a | a | a | a | a ₃ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | ... |
| b | b | b | b | b | b ₃ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | ... |
| c | c | c | c | c | c ₃ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | ... |
| d | d | d | d | d | d ₃ | | | | | | | | | | | | | | ... |
| e | e | e | e | e | e ₃ | | | | | | | | | | | | | | ... |
| f | f | f | f | f | f ₃ | | | | | | | | | | | | | | ... |
| g | g | g | g | g | g ₃ | | | | | | | | | | | | | | ... |
| h | h | h | h | h | h ₃ | | | | | | | | | | | | | | ... |
| i | i | i | i | i | i ₃ | | | | | | | | | | | | | | ... |
| j | j | j | j | j | j ₃ | | | | | | | | | | | | | | ... |
| k | k | k | k | k | k ₃ | | | | | | | | | | | | | | ... |
| l | l | l | l | l | l ₃ | | | | | | | | | | | | | | ... |
| m | m | m | m | m | m ₃ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | ... |
| | | | | | | | | | | | | | | | | | | | ... |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| | | | | | | | | | | | | | | | | | | | ... |

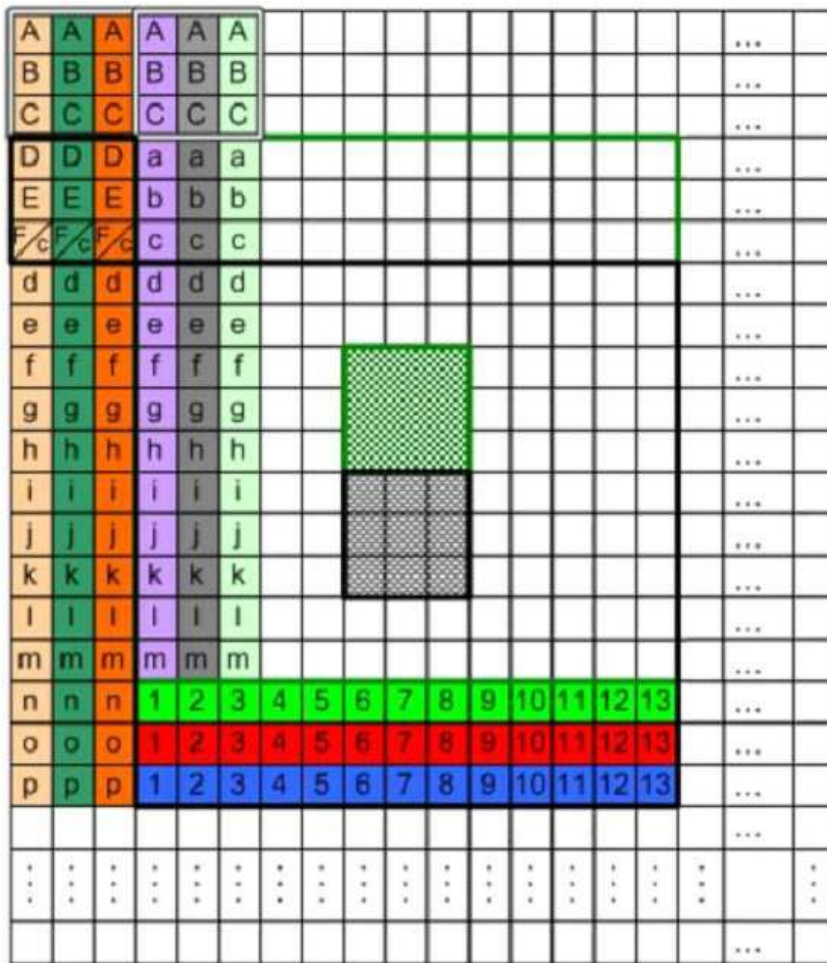
도면9b



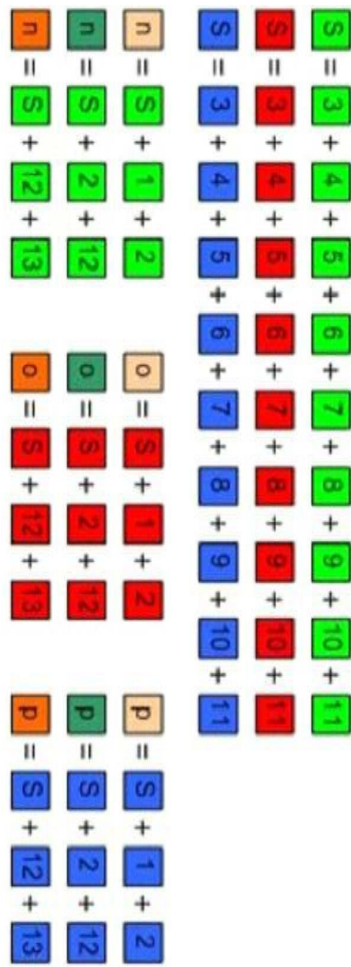
도면9c



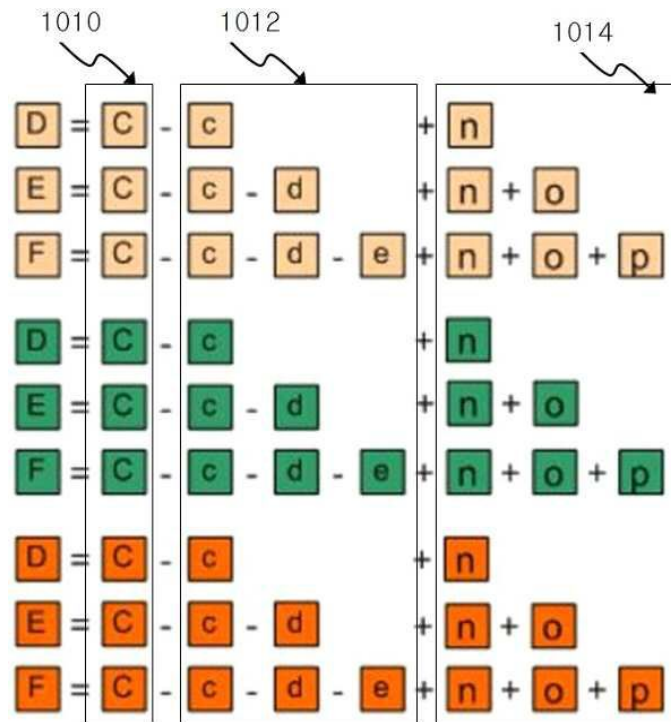
도면10a



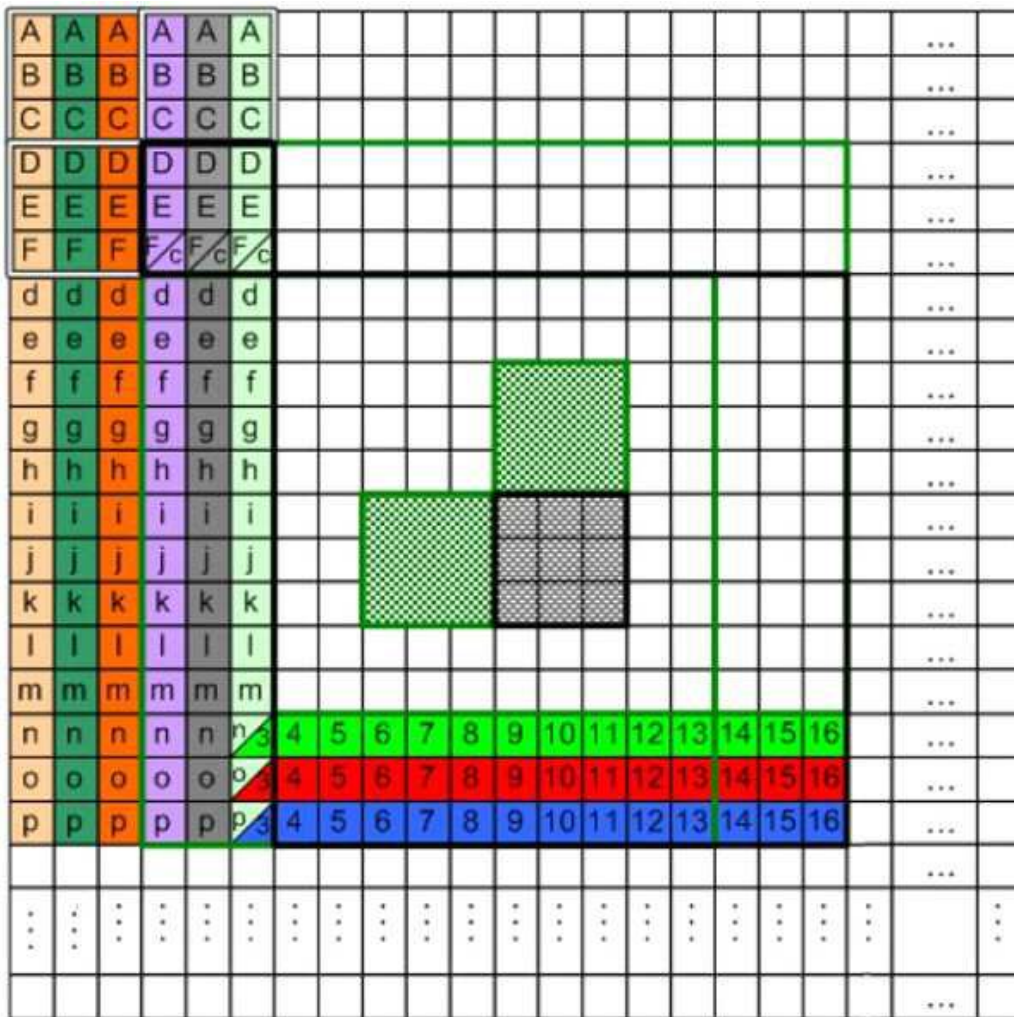
도면10b



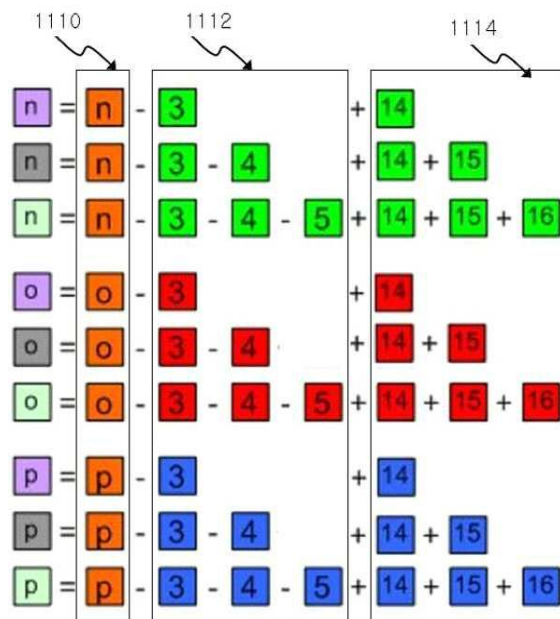
도면10c



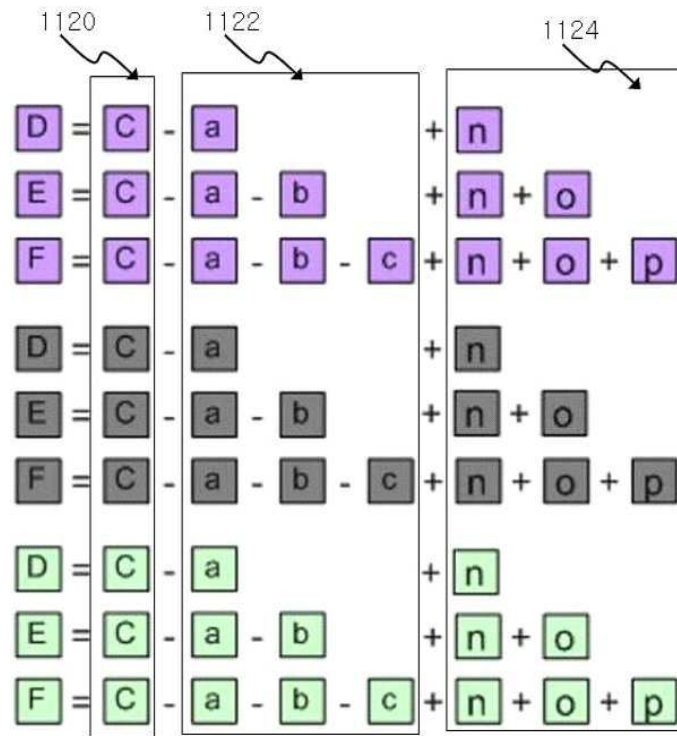
도면11a



도면11b



도면11c



도면12

