

형태소 분해를 통한 형태학적 필터의 효율적 구현

정회원 박 호 중*

Efficient Implementation of Morphological Filters by Structuring Element Decomposition

Hochong Park* *Regular Member*

요 약

형태학적 필터를 영상 처리 시스템에 효율적으로 구현하기 위하여서는 영상 처리 시스템의 구조적인 제약으로 인하여 형태소의 크기가 작아야 하며, 따라서 크기가 큰 형태소를 가지는 필터의 구현을 위하여 주어진 형태소를 작은 크기의 형태소로 분해하는 과정이 필요하다. 본 논문에서는 임의의 형태학적 필터를 3×3 Pipeline Machine에 효율적으로 구현하기 위하여 모양과 크기에 아무 제한 없이 형태소를 분해하는 알고리즘을 개발한다. 먼저 주어진 형태소를 순차적 방법으로 합집합 분해하고, 각 분해된 결과를 다시 3×3 크기의 형태소로 최적 분해하여 최종 형태소 분해 결과를 얻는다. 개발된 분해 알고리즘을 형태소에 적용하여 분해 결과를 구하였으며, 최적에 매우 근접한 결과를 얻을 수 있었다.

ABSTRACT

In order to implement morphological filters on image processing systems, the size of structuring element must be small due to the architectural constraints of the systems, which requires the decomposition of structuring element into small elements for the filters with large structuring elements. In this paper, an algorithm for decomposition of structuring element with no restriction on the shape and size is developed which enables sub-optimal implementation of any morphological filter on 3×3 pipeline machine. The given structuring element is first decomposed into the union of elements using sequential search procedure, then each element is further decomposed optimally into 3×3 elements, resulting in final sub-optimal 3×3 hybrid decomposition. The proposed algorithm is applied to some structuring elements and the results close to the optimum are obtained.

1. 서론

형태학적 필터(Morphological Filters)^[1]는 영상 처리 분야에 널리 사용되고 있는 비선형 필터링 기법으로서 이들을 실제로 응용하기 위하여서는 설계된 형태학적 필터를 영상 처리 시스템에 효율적으로 구현하여야 한다. 그러나 시스템의 구조적인 제약으로 인하여 Parallel Array Processor^[2]는 Region of Support가 작은 연산만을 효율적으로 구현하고,

Pipeline Machine^[3]은 일정 크기(주로 3×3) 이하의 Region of Support를 가지는 연산만 구현할 수 있는 분해점이 있다. 형태학적 필터의 경우 사용되는 형태소의 크기가 Region of Support의 크기이므로 형태소의 크기는 구현의 효율성 또는 가능성을 결정하는 중요한 요인이 된다. 따라서 큰 형태소를 가지는 형태학적 필터를 영상 처리 시스템에 효율적으로 구현하기 위하여 주어진 형태소를 크기가 작은 형태소로 분해하여 필터를 동등하게 구현하는 방법의 개발이 필요하며, 이 과정을 "형태소 분해"

* 광운대학교 전자공학과(hcpark@daisy.kwangwoon.ac.kr)

논문번호: 99135-0413, 접수일자: 1999년 4월 13일

* 본 연구는 한국과학재단 1998 핵심 전문연구(과제번호: 981-0912-059-2)의 연구 지원에 의하여 수행되었습니다.

라 한다.

형태소 분해에 대한 많은 연구들이 진행되었으나 아직 임의 모양의 형태소를 분해하는 일반적인 알고리즘은 개발되지 못하였다^[4,8]. Zhuang과 Haralick^[4]은 임의 모양의 형태소를 2-pixel 영상으로 분해하는 방법, 즉 분해된 각 영상이 모두 2개의 점만을 가지는 영상이 되도록 분해하는 방법을 연구하였다. 그러나 이 방법은 형태소 분해를 형태학적 필터의 구현과 연결하지 않아 실제로 영상 처리 시스템에 형태학적 필터를 효율적으로 구현하기 위한 방법을 제공하지는 못한다. Xu^[5]와 Kanungo *et al*^[6]은 볼록(Convex) 형태소에 대한 분해 방법을 개발하였으나 최적의 구현을 보장하지 못한다. Park과 Chin^[7, 8]은 형태소 분해와 형태학적 필터의 구현을 연결하여 주어진 시스템에 최적의 구현을 가능하게 하는 최적 분해(Optimal Decomposition)를 구하는 방법을 처음으로 개발하였다. [7]은 Parallel Array Processor에 대한 최적 분해방법을 개발하였으나 볼록 형태소만 다루는 한계가 있고, [8]은 3×3 Pipeline Machine에 대한 최적 분해방법을 개발하였고 임의의 형태소에 대하여 적용이 가능 하지만, 3×3의 크기로 분해가 되는 않는 형태소에 대한 처리 방법을 제시하지 않는 문제점이 있다.

본 논문에서는 3×3 Pipeline Machine에서의 형태학적 필터의 구현을 위한 형태소 분해 방법을 연구하며, 특히 3×3로 평분해 되지 않는 경우의 효율적 구현 방법에 대하여 집중 연구한다. 2장에서는 형태소 분해에 대한 간단한 이론을 정리하고, 3장에서는 본 논문의 주 결과인 효율적 형태소 혼합 분해 방법을 제안하고, 4장에서는 제안된 방법을 이용한 형태소 분해의 예제를 보여주고, 5장에서 결론을 내린다.

II. 형태소 분해를 위한 이론적 배경

1. Chain Rule과 배분법칙

확대 연산과 축소 연산을 각각 \oplus 와 \ominus 로 표시하고, 만일 주어진 형태소 S 가

$$S = B_1 \oplus B_2 \oplus \dots \oplus B_N \quad (1)$$

와 같이 확대 연산으로 분해되면 형태학적 필터의 Chain Rule에 의하여

$$A \oplus S = (\dots(A \oplus B_1) \oplus B_2) \oplus \dots \oplus B_N \quad (2)$$

$$A \ominus S = (\dots(A \ominus B_1) \ominus B_2) \ominus \dots \ominus B_N \quad (3)$$

로 표시된다. 즉, S 와의 확대 연산과 축소 연산은 각 B_i 와의 순차적인 확대 연산, 축소 연산과 동일하며, 만일 각 B_i 의 크기가 3×3 이하이면, S 와의 연산을 B_i 와의 순차적인 연산을 통하여 3×3 Pipeline Machine에 구현할 수 있다.

만일 주어진 형태소 S 가

$$S = C_1 \cup C_2 \cup \dots \cup C_M \quad (4)$$

와 같이 합집합으로 분해되면 배분법칙에 의하여

$$A \oplus S = (A \oplus C_1) \cup \dots \cup (A \oplus C_M) \quad (5)$$

$$A \ominus S = (A \ominus C_1) \cap \dots \cap (A \ominus C_M) \quad (6)$$

로 표시되어 S 와의 연산은 각 C_i 와의 연산 결과들 사이의 합집합과 교집합을 구한 결과와 동일하다. 따라서 C_i 의 크기가 모두 3×3 이하이면 식(5), (6)를 이용하여 3×3 Pipeline Machine에 효율적으로 구현할 수 있다. 그러나 모든 S 에 대하여 식(4)의 방법으로 3×3 분해는 가능하지만, 식(1)과 같은 3×3 분해가 되지 않는 경우가 있다.

본 논문에서는 위의 두 가지 분해 방법을 구별하기 위하여 식(1) 방법을 평(平)분해(Plain Decomposition), 식(4) 방법을 약(弱)분해(Weak Decomposition)이라 한다. 식(4)에서 합집합 분해된 각 C_i 들을 다시 독립적으로 평분해 할 수 있는데, 이렇게 분해된 최종 결과를 혼합분해(Hybrid Decomposition)라 한다. 따라서 혼합분해가 가장 일반적인 분해 형태이고

$$S = (D_{11} \oplus D_{12} \oplus \dots \oplus D_{1k_1}) \cup \dots \cup (D_{M1} \oplus D_{M2} \oplus \dots \oplus D_{Mk_M}) \quad (7)$$

와 같이 표시된다. 여기서 $M = 1$ 이면 평분해에 해당하고, 모든 $k_i = 1$ 이면 약분해에 해당한다.

2. Pipeline Machine 구현의 Cost

주어진 형태소 S 를 분해하는 방법은 유일하지 않으며 분해 결과에 따라 구현 속도가 달라진다. S 와 의 연산을 Pipeline Machine에서 구현시키기 위한 연산량을 $Cost(S)$ 라 하면 $Cost(S)$ 는 분해결과에 의한 순차적 동작의 실행 개수에 비례한다. 따라서, 식(1)과 같이 S 를 평분해하고 B_i 의 크기가 3×3 이하라고 가정하면 $Cost(S)=N$ 이 되고, 식(4)와 같이 S 를 약분해하고 C_i 의 크기가 모두 3×3 이하이면 $Cost(S)=M$ 가 된다. 또한 식(7)과 같이 S 를 혼합분

해하고 모든 D_{ij} 의 크기가 3×3 이하이면 $Cost(S) = k_1 + k_2 + \dots + k_M$ 이다.

혼합분해에서 각 C_i 에 의한 계산량은 $Cost(C_i) = ki$ 이고, 이 값은 C_i 의 모양과 절대위치에 따라 변하며, $Cost(C_i)$ 의 최소값($Cost(C_i)_{min}$)은 $\max\{|x_{max}(C_i)|, |x_{min}(C_i)|, |y_{max}(C_i)|, |y_{min}(C_i)|\}$ 이 된다^[5]. 여기서 $x_{max}(C_i) = \max\{x | (x, y) \in C_i\}$ 로 정의되며 다른 항들도 같은 개념으로 정의된다. $Cost(C_i)_{min}$ 의 의미는 원점에 중심을 가지는 정사각형 중에서 C_i 에 외접하는 정사각형의 x 또는 y 축 절편값이다. 본 논문에서는 $R(C_i)$ 를 원점에 중심을 가지고 C_i 에 외접하는 정사각형이라 정의하고, $Size(C_i)$ 는 $R(C_i)$ 의 x 또는 y 축 절편으로 정의한다. 따라서, $Cost(C_i)_{min} = Size(C_i)$ 이고, C_i 가 $Size(C_i)$ 개의 3×3 영상으로 평분해 되면 이것이 바로 C_i 의 최적 3×3 평분해이다. 따라서 최적 혼합분해는 $Size(C_1) + Size(C_2) + \dots + Size(C_M)$ 을 최소로 하는 약분해를 구하고 각 C_i 를 최적 3×3 평분해하면 된다.

3. 3×3 평분해 검증

본 논문에서는 S 를 3×3 평분해 가능한 C_i 의 합집합으로 분해하므로 주어진 영상이 3×3 평분해 되는지를 검증하는 과정이 필요하다. 3×3 평분해 검증 방법은 [8]에서 개발되었으며 아래와 같이 정리된다.

■ 3×3 평분해 조건

- 조건 1. 오목 외곽선의 크기가 3×3 이하이다.
- 조건 2. 오목 외곽선의 모양에 따라 양 끝 점의 연결 각도가 다음을 만족한다.
 Type U : 0°
 Type J : 45° 이하
 Type L : 90° 이하
 Type V : 90° 이하
 Type r : 135° 이하
- 조건 3. 각 오목 외곽선에 대하여 그 오목 외곽선을 가지는 3×3 Factor가 하나 이상 존재한다.
- 조건 4. [8]에 정의된 선형방정식의 정수해가 존재한다.

오목 외곽선은 이웃한 두 블록 모서리점 사이의 점들로 구성된 외곽선 조각 중에서 오목 모서리점을 포함하는 외곽선 조각을 말하며, 오목 외곽선의 양 끝 점은 정의된 외곽선 조각의 양 끝 블록 모서리점을 말한다^[8]. 오목 외곽선의 Type은 [8]에 정의

되어 있다. 조건 1과 2를 검증하는 예는 그림 1에 있다. (a)는 실선으로 표시된 오목 외곽선의 크기가 3×5 이므로 조건 1을 만족 못하고, (b)는 오목 외곽선 크기 조건은 만족하지만 표시된 Type L 오목 외곽선의 끝 블록 모서리점 중의 하나의 연결 각도가 135° 가 되어 조건 2를 만족 못한다. 또한, (c)는 표시된 Type J 오목 외곽선의 한 끝 점의 연결 각도가 90° 가 되어 조건 2를 만족 못한다.

III. Pipeline Machine 구현을 위한 형태소 분해

주어진 임의의 형태소 S 를 식(4)과 같이 3×3 평분해 되는 C_i 들의 합집합으로 분해하고 각 C_i 를 최적 3×3 평분해 하여 최종적으로 S 의 효율적인 구현을 위한 혼합분해를 구한다.

모든 블록 영상은 항상 3×3 평분해 되므로 가장 간단한 합집합 분해 방법은 S 를 블록 영상들만의 합집합 분해하는 것이다^[7]. 그러나 블록 영상 이외에도 3×3 평분해 되는 영상이 많으므로 “블록” 제한을 제거하여 더 많은 유연성을 제공함으로써 더 작은 값의 $Cost(S)$ 를 가지는 분해 결과를 얻을 수 있고, 이를 통하여 보다 효율적인 구현이 가능하게 된다. 이와 같이 $Cost(S)$ 를 작게 하는 합집합 분해 방법을 아래와 같이 제안한다. 물론 이 방법이 항상 최적의 결과를 제공하여 주지는 않지만 모든 모양의 형태소에 대하여 최적에 매우 근접한 결과는 주어 만족할 성능을 얻을 수 있다.

■ 효율적인 혼합분해 방법

STEP 0. $B_1 = S, i = 1$.

STEP 1. $R(B_i)$ 외곽선 위에 있는 B_i 의 점을 많이 포함하며 $C_i \subset R(B_i)$ 이고 크기가 큰 3×3 평분해 가능한 영상 C_i 를 구한다. 이때 C_i

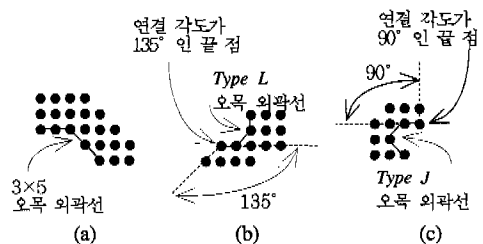


그림 1. 3×3 평분해 검증 알고리즘의 조건 1과 조건 2 적용 예.

$C_i S$ 를 만족하여야 한다.

STEP 2. $B_{i+1} = S \setminus \bigcup_{k=1}^i C_k$ 를 구한다. B_{i+1} 크기가 3×3 이하이면 STEP 3로 가고, 3×3 보다 크면 $i=i+1$ 을 하고 STEP 1로 간다.

STEP 3. $S = \bigcup_{k=1}^i C_k \cup B_{i+1}$ 이 되고 각 C_i 와 B_{i+1} 를 [8]에 의하여 최적 3×3 평분해 한다.

분해 방법은 기본적으로 순차적 C_i 검색으로 구성된다. STEP 1에서 주어진 S 에 대하여 특정 조건을 만족하는 C_1 를 선택하고, STEP 2에서 C_1 로부터 B_2 을 정의하며, 다시 STEP 1에서 B_2 로부터 조건에 맞는 C_2 을 선택하여 STEP 1-STEP 2의 과정을 되풀이한다. 따라서 각 i 번째 단계에서 어떤 C_i 를 선택하느냐에 따라 최종 분해 결과가 달라지며, 최종 결과의 $Cost(S)$ 가 최소가 되도록 하기 위하여 매 단계에서 모든 가능한 C_i 에 대하여 최종 결과를 미리 계산하여 최적의 C_i 를 선택하는 Tree 검색 방법을 사용하던지, 또는 한번의 순차적 검색 과정이 아니라 Feedback을 가지는 여러 번의 순차적 검색을 반복하여 최적의 결과를 구하여야 한다. 그러나 이 방법들은 매우 복잡한 과정을 요구하므로, 본 논문에서는 최소의 $Cost(S)$ 를 가지는 최적 분해 방법을 구하지 않고 간단한 순차적 검색 과정을 통하여 최적에 근접한 분해방법을 쉽게 구하는 방법을 연구한다. 즉, 매 i 번째 단계에서 수많은 C_i 후보 중에서 최종 $Cost(S)$ 가 작아지도록 하는 C_i 를 선택하는 기준을 구하는 것이 논문의 핵심 내용이다.

제안한 분해 방법에 대한 이론적 근거는 다음과 같다. 각 단계별로 순차적으로 하나씩의 C_i 를 선택하는 것이므로, 현재 C_i 의 선택으로 인한 다음 $i+1$ 단계에서의 영향만을 살펴보는 것이 가능하고, 가장 간단하게 i 번째 단계에서 C_i 를 선택한 후 $i+1$ 단계에서 모든 검색이 끝난다고 가정한다. 즉, i 번째 단계를 마친 후 $S = \bigcup_{k=1}^i C_k \cup B_{i+1}$ 되고 B_{i+1} 로부터 3×3 평분해 되는 C_{i+1} 을 구하여, 그 결과 $S = \bigcup_{k=1}^i C_k \cup C_{i+1}$ 로 최종 합집합 분해가 된다고 가정한다. 이 단계에서 어떤 C_i 를 선택하면 전체적인 $Cost$ 가 작아지는지를 살펴보고 이를 만족하는 C_i 를 정하는 방법을 구하면 된다.

i 번째 단계의 시작점에서 $S = \bigcup_{k=1}^i C_k \cup B_i$ 가 되고, C_i 가 선택되면 $S = \bigcup_{k=1}^i C_k \cup C_i \cup B_{i+1}$ 가 되고 $i+1$ 단계에서 검색 종료되면 최종적으로 $S = \bigcup_{k=1}^i C_k \cup C_i \cup C_{i+1}$

가 된다. 이때 $\bigcup_{k=1}^i C_k$ 는 이미 정해진 부분으로서 C_i 의 선택에 따라 바뀌지 않는 부분이므로 $Cost(C_i) + Cost(C_{i+1})$ 가 최소가 되도록 C_i 를 선택하여야 한다. 이 때 B_{i+1} 로부터 3×3 평분해 되는 C_{i+1} 을 구하여 검색이 종료되는 것으로 가정하였으므로 $Cost(C_{i+1})_{min} = Size(B_{i+1})$ 이다.

C_i 와 $R(B_i)$ 외곽선 위에 있는 B_i 점들의 관계에 따라 다음의 세 가지 경우를 살펴본다. 첫째, C_i 가 $R(B_i)$ 의 외곽선 위에 있는 모든 B_i 점을 포함하면 $Cost(C_i) \geq Size(B_i)$, $Cost(C_{i+1})_{min} < Size(B_i)$ 가 된다. 두 번째 부등식은 C_i 가 B_i 의 점 중에서 $R(B_i)$ 위에 있는 모든 점들을 포함하므로 나머지 점들로 구성되는 B_{i+1} 에 대하여 $Size(B_{i+1}) < Size(B_i)$ 을 만족하기 때문이다. 이때, $Cost(C_i) = Size(B_i)$ 가 되면 $Cost(C_i)$ 는 최소가 되므로 이를 만족하기 위하여 $C_i \subset R(B_i)$ 조건이 필요하다. 또한 $C_i \subset R(B_i)$ 조건만 만족되면 C_i 의 크기에 관계없이 $Cost(C_i) = Size(B_i)$ 로 일정하므로, C_i 의 크기를 가능한 한 크게 하면 B_{i+1} 의 크기가 상대적으로 작아져 $Cost(C_{i+1})_{min} \ll Size(B_i)$ 가 되어 $Cost(S)$ 를 매우 작게 할 수 있다. 즉, C_i 의 선택 기준은 $C_i \subset R(B_i)$ 을 만족하고 크기를 크게 하는 것이다.

둘째, C_i 가 $R(B_i)$ 의 외곽선에 있는 B_i 점의 일부 분만을 포함하면 $Cost(C_i) \geq Size(B_i)$, $Cost(C_{i+1})_{min} = Size(B_i)$ 가 된다. 두 번째 등식은 C_i 가 $R(B_i)$ 외곽선에 있는 일부 B_i 점들을 포함하지 못하므로 그 점들은 B_{i+1} 에 남게되고 $Size(B_{i+1}) = Size(B_i)$ 이기 때문이다. 따라서 $Cost$ 를 줄이기 위하여서는 $C_i \subset R(B_i)$ 조건을 주어 $Cost(C_i) = Size(B_i)$ 가 되도록 한다. 물론 이 경우 첫번째 경우에 비하여 $Cost(S)$ 는 증가한다.

셋째, C_i 가 $R(B_i)$ 외곽선의 B_i 점을 전혀 포함하지 않는 경우는 첫 번째 경우에서 선택 순서를 바꾸는 것에 불과하므로 첫 번째와 동일하게 다룬다.

첫 두 경우를 보면 $Cost(C_i) + Cost(C_{i+1})$ 을 작게 하기 위하여, $R(B_i)$ 의 외곽선 위에 있는 B_i 의 모든 점들을 포함하고 3×3 평분해 되는 C_i 중에서 $C_i \subset R(B_i)$ 를 만족하며 가능한 크기가 큰 것을 선택하면 되고, 만일 그런 C_i 가 없을 경우 $C_i \subset R(B_i)$ 를 만족하고 3×3 평분해 되는 C_i 를 찾으면 된다. 이와 같은 근거를 인하여 STEP 1 단계와 같은 방법이 구하여 졌다.

IV. 제안된 분해 방법의 적용 예

위에서 제안한 효율적인 3×3 혼합 분해 방법을

두 종류의 S 에 실제 적용하는 예를 보여주며, \square 는 원점을 나타낸다. 첫 번째 예제는 그림 2에 주어진다. (a)가 주어진 형태소 S 이고, 오목 외곽선의 크기가 3×3 보다 크므로 3×3 평분해 되지 않고, 혼합 분해를 통해서만이 3×3 Pipeline Machine에서 구현 가능하다. 먼저 $R(B_1)$ 외곽선 위에 있는 B_1 의 점을 찾으면 우측 상단의 모서리점이다. 이 점부터 시작하여 크기가 가장 큰 3×3 평분해 가능한 영상 C_1 을 구하면 (b)에서 실선으로 정의된 영상과 같다. (c)에서 \circ 은 C_1 에 의하여 제거된 부분이며, \bullet 로 표시되는 B_2 가 정의되고, $R(B_2)$ 는 점선으로 표시되어 있다. 다음, $R(B_2)$ 외곽선에 있는 B_2 의 점들을 포함하고 크기가 큰 3×3 평분해 가능한 영상을 구하면 (d)에서 실선으로 정의된 C_2 가 된다. $B_3 = \bigcap_{k=1}^2 C_k$ C_k 는 Null이 되므로 분해 과정이 끝나고, 그 결과 S 는 (e)와 같이 두 개 영상의 합집합으로 표시되며, 각 영상은 3×3 평분해 가능하므로 [8]의 방법을 이용하여 최적으로 분해하면 최종적으로 (f)와 같이 9개의 3×3 영상으로 혼합 분해된다. 특히 이 결과는 최소의 $Cost(S)$ 를 가지는 최적 분해이다. 여기서 영상 앞에 있는 숫자 n 은 각 영상이 n 번 확대 연산되는 것을 의미한다.

두 번째 예는 그림 3에 있다. (b)에서 $R(B_1)$ 위에 있는 B_1 의 점은 중앙 상단의 모서리점이고, 이로부터 시작하여 3×3 평분해 되는 C_1 을 구하면 (b)의 실선으로 정의된다. 다음 (c)에서와 같이 B_2 와 $R(B_2)$ 를 구하고 이로부터 C_2 를 구하고, (d)와 같이 C_3 를 구하면 검색이 끝난다. 그 결과 (e)와 같이 합집합 분해가 되고 각 영상을 3×3 으로 분해하면 (f)와 같이 14개의 3×3 영상으로 최종 분해된다.

그림 4의 예제는 항상 블록 C_i 로 분해하여 검색을 쉽게 하는 방법을 사용한 결과이다. 만드시 블록 C_i 를 찾아야 하므로 (b)-(e)의 C_1, C_2, C_3, C_4 가 구하여지고 (g)의 최종 분해 결과가 나온다. 이 경우 22개의 3×3 영상으로 분해되었으며 그림 3에 비하여 $Cost(S)$ 가 많이 증가한 것을 알 수 있다.

V. 결론

본 논문에서는 임의 모양과 크기의 형태소를 가지는 형태학적 필터를 3×3 Pipeline Machine에 효율적으로 구현하기 위하여 3×3 혼합 분해를 구하는 알고리즘을 개발하였다. 개발된 알고리즘은 특정 형태소에 대하여 최적의 구현을 보장하지는 않지만

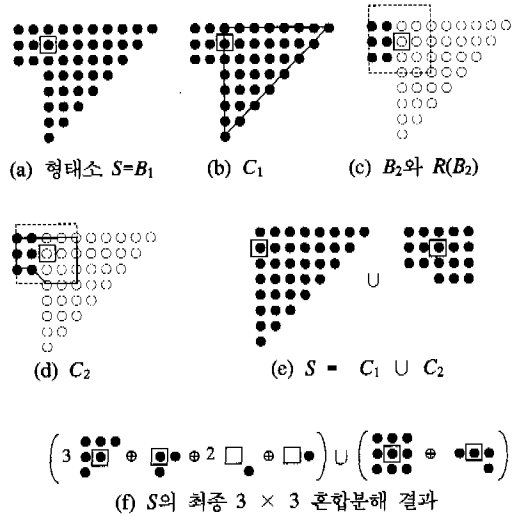


그림 2. 제안된 혼합분해 방법을 적용한 예제 1.

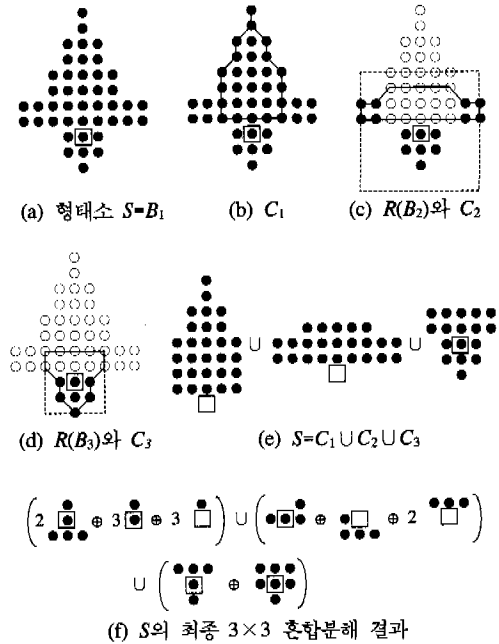


그림 3. 제안된 혼합분해 방법을 적용한 예제 2.

모양과 크기에 아무 제한이 없이 모든 형태학적 필터의 효율적 구현을 제공하는 장점을 가진다. 형태소가 주어지면 먼저 단계별로 최적화된 순차적 검색 방법으로 형태소의 약분해를 구하고, 다음 각 분해결과를 3×3 최적 평분해 하여 최종 3×3 혼합 분해를 얻는다. 개발된 분해 알고리즘을 형태소에 적용하여 매우 효율적인 혼합 분해를 얻었고, 그 결

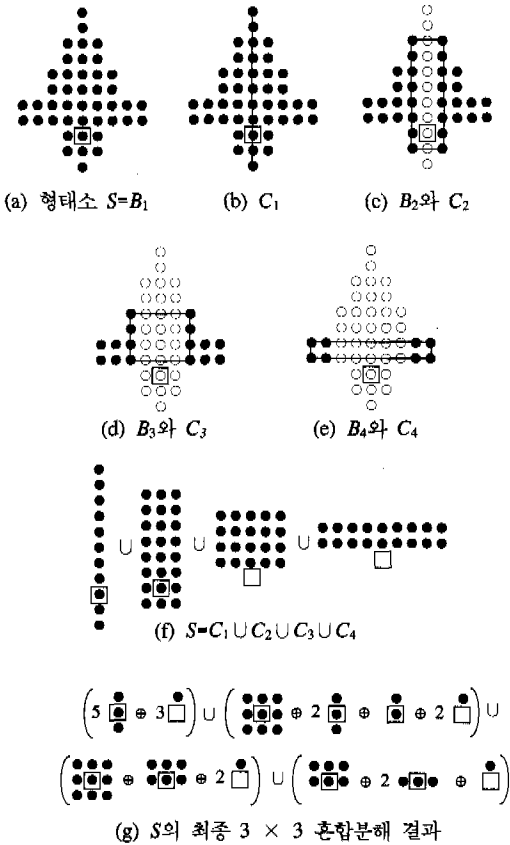


그림 4. 제안된 혼합분해 방법에 블록 조건을 추가한 예제.

과 알고리즘의 성능을 확인 할 수 있다.

참고 문헌

[1] G. Matheron, *Random Sets and Integral Geometry*, New York, N.Y.: John Wiley & Sons, 1975.

[2] K. E. Batcher, "Design of a massively parallel processors," *IEEE Trans. Computers*, vol.29, no.9, 1980.

[3] S. R. Sternberg, "Parallel architecture for image processing," *Proc. IEEE International Conference on Computer Software Applications*, Chicago, IL, 1979.

[4] X. Zhuang and R. M. Haralick, "Morphological structuring elements decomposition," *Computer Vision, Graphics and Image Processing*, vol.35,

Sept., 1986.

[5] J. Xu, "Decomposition of convex polygonal morphological structuring elements into neighborhood subsets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.13, no.2, Feb., 1991.

[6] T. Kanungo, R. M. Haralick and X. Zhuang, "B-code dilation and structuring element decomposition for restricted convex shapes," *Image Algebra and Morphological Image Processing, Proc. SPIE*, vol.1350, 1990.

[7] Hochong Park and Roland Chin, "Optimal decomposition of convex structuring elements for 4-connected parallel array processors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.16. no.3, March 1994.

[8] Hochong Park and Roland Chin, "Decomposition of arbitrarily-shaped morphological structuring elements," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.17. no.1, Jan., 1995.

박 호 중(Hochong Park) 정회원
 1986년 2월 : 서울대학교 전자공학과 졸업(공학사)
 1987년 12월 : Univ. of Wisconsin-Madison 전기공학과 졸업(M.S.)
 1993년 5월 : Univ. of Wisconsin-Madison 전기공학과 졸업(Ph.D.)
 1997년 9월~현재 : 광운대학교 전자공학과 조교수
 <주관심 분야> 음성/영상 신호처리, 비선형 신호처리