

# 모폴로지 연산에 사용되는 볼록 구조요소의 분해를 위한 알고리즘\*

온승엽\*\*

## A Decomposition Algorithm for Convex Structuring Elements in Morphological Operation

Syng Yup Ohn

### Abstract

The decomposition of a structuring element for a morphological operation reduces the amount of the computation required for executing the operation. In this paper, we present a new technique for the decomposition of convex structuring elements for morphological operations. We formulated the linear constraints for the decomposition of a convex polygon in discrete space, then the constraints are applied to the decomposition of a convex structuring element. Also, a cost function is introduced to represent the optimal criteria for decomposition. We use linear integer programming technique to find the combination of basis structuring elements which minimizes the amount of the computation required for executing the morphological operation. Formulating different cost functions for different implementation methods and computer architectures, we can determine the optimal decompositions which guarantee the minimal amounts of computation on different computing environment.

**Key Words:** morphological operation, structuring element, decomposition, optimization

\* 이 논문은 과학기술부 한국과학재단 지정 경기도 지역 협력연구센터(RRC)인 한국 항공대학교 인터넷 정보검색 연구센터의 지원에 의한 것임.

\*\* 한국 항공대학교 전자 정보통신 컴퓨터 공학부

## 1. 서론

수학적 형상학(Mathematical Morphology)은 영상처리와 컴퓨터 비전에서 영상의 향상 및 분석에 널리 이용된다[1], [2], [3]. 수학적 형상학의 기본연산은 팽창연산(dilation operation)과 침식연산(erosion operation)이며 다음과 같이 정의된다. 다음의 정의에서  $A$ 와  $B$ 는 2 차원 불연속공간(2D discrete space),  $Z^2$ , 의 점의 좌표를 원소로 하는 집합이다[4].

팽창연산 :

$$A \oplus B = \{a + b \mid a \in A, b \in B\} \quad (1)$$

침식연산 :

$$A \ominus B = \{c \mid c + b \in A \text{ for every } b \in B\} \quad (2)$$

위의 정의를 영상처리의 관점에서 보면  $A$ 는 입력영상,  $B$ 는 템플레이트 역할을 하는 구조요소(structuring element)를 나타낸다. 수학적 형상학을 이용하여 영상처리를 할 때에는 처리의 목적에 따라 팽창연산과 침식연산에 사용되는 구조요소의 형상과 크기를 적절하게 선택하고, 기본연산과 집합 연산을 적절한 순서에 따라 실행함으로써 여러 가지 영상처리 결과를 얻을 수 있다.

팽창 또는, 침식연산을 수행할 때 필요한 계산량은 일반적으로 선형 이상의 급수로 증가하게 된다. 큰 구조요소와의 연산의 경우 필요한 계산량이 대폭적으로 커진다. 이 경우 원래의 큰 구조요소를 여러 개의 작은 구조요소의 조합으로 분해하고 구조요소 조합에 포함된 구조요소에 의하여 차례로 연산을 수행하면 원래의 구조요소에 대한 연산의 결과와 같은 결과를 얻으면서 연산의 수행에 필요한 총 계산량을 줄일 수 있다. 즉, 팽창연산의 연결법칙에 의하여 구조요소  $B$ 가 다음과 같이  $B_1, B_2, \dots, B_n$ 으로 분해된다면

$$B = B_1 \oplus B_2 \oplus \dots \oplus B_n \quad (3)$$

영상  $A$ 에 대한 구조요소  $B$ 의 팽창연산은 아래와 같이 각각  $B_1, B_2, \dots, B_n$ 에 대하여 차례

대로 팽창연산을 실행함으로써 구할 수 있다. [2], [5]

$$(((A \oplus B_1) \oplus B_2) \oplus \dots) \oplus B_n \quad (4)$$

일반적으로, (4)에서의 일련의 팽창연산들에 필요한 계산량의 총합은 원래 구조요소에 의한 한 번의 팽창연산,  $A \oplus B$ 에 필요한 계산량보다 작게 된다. 침식연산의 경우에는 다음과 같이 실행될 수 있다.

$$(((A \ominus B_1) \ominus B_2) \ominus \dots) \ominus B_n \quad (5)$$

침식연산의 경우에도 팽창연산과 마찬가지로 구조요소의 분해에 의하여 연산의 실행에 필요한 계산량을 줄일 수 있다. 침식연산에 대한 구조요소의 분해에 대하여서는 팽창연산의 경우와 유사한 논의가 가능하므로 이 논문에서는 중복을 피하기 위하여 팽창연산만을 기준으로 하여 논의하기로 한다.

블록한 형상의 구조요소는 수학적으로 우수한 특성을 가지고 있으므로 수학적 형상학에서 자주 쓰이며, [7] 이러한 종류의 구조요소의 분해에 대하여 많은 연구가 행하여 졌다 [6]. 최초의 블록 구조요소의 분해 알고리즘은 Xu에 의하여 발표되었다[7]. 이 알고리즘은 블록 구조요소를 3 x 3의 지역 윈도우(local window)영역에 포함될 수 있는 기본구조요소의 조합으로 분해하는 방법이며, 블록 구조요소에 이 알고리즘을 적용할 경우 블록 구조요소로부터 Cytocomputer[8]와 같은 파이프라인 구조(pipeline architecture)에 최적화된 기본구조요소의 분해 집합을 구할 수 있다. Park은 4점 연결(4-connected)된 MPP형식의 컴퓨터구조에 최적한 분해를 구하는 알고리즘을 제시하였다[9].

영상처리에서의 형상 분해는 다양한 분야에 응용된다. 2차원 2진 영상에서의 형상 표현(shape representation)[10], [11], 텍스트 분류(text classification)[12], [13], 영상 영역 분리(region segmentation)[14] 등에 응용된다.

이 논문에서는 블록 구조요소를 주어진 기본구조요소(basis structuring element)의 조합으로 최적 분해하는 알고리즘을 제시하고자

한다. 최적분해조합이란 팽창연산 수행시 최소의 계산량을 필요로 하는 기본구조요소들의 조합이다. 팽창연산은 다양한 컴퓨터구조에서 서로 다른 방법으로 구현이 되므로 최적분해조합도 연산이 실행되는 컴퓨터구조와 구현방법에 따라 달라지게 된다. 기존의 분해 알고리즘들이 각각 특정한 한 종류의 컴퓨터구조에 최적한 분해조합을 구하는 데 비하여 이 논문에서 제시되는 방법은 다양한 컴퓨터구조나 형상 연산의 구현방식에 대하여 최적한 분해조합을 용이하게 구할 수 있는 일반적인 방법이다. 이 논문에서 블록 구조요소는 8개의 변을 가지는 디지털 블록다각형으로 정의되었다. 그리고 원래의 구조요소가 기본구조요소의 조합으로 분해하기 위한 필요충분조건으로서 구조요소들의 변의 길이에 대한 선형제한조건(linear constraint)이 유도된다. 분해조합에 포함된 기본구조요소들에 의하여 일련의 팽창연산을 수행할 때 필요한 계산량을 나타내는 비용함수를 선형방정식의 형태로 정의하였다. 따라서 구조요소의 최적분해조합을 구하는 문제는 분해에 대한 필요충분조건을 만족하며 동시에 비용함수를 최소화하는 기본구조요소 조합을 구하는 선형계획(linear programming)의 문제로 변환될 수 있다. 비용함수는 팽창연산이 수행되는 컴퓨터구조와 연산의 구현방식에 따라 달라지며 분해에 대한 최적화 기준을 나타낸다. 비용함수는 각 기본구조요소에 의한 팽창연산에 소요되는 계산량으로부터 용이하게 구하여 질 수 있다. 비용함수와 기본구조요소 집합을 적절히 선택함으로써 다양한 계산환경 및 구현방법에 대하여 최적한 분해를 결정해낼 수 있다.

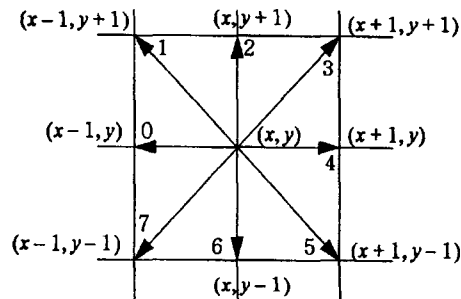
본 논문의 구성은 다음과 같다. 2장에서 블록다각형을 정의하고 블록다각형의 분해조건을 유도하였고 3장에서는 분해조건을 블록 구조요소의 최적 지역 분해에 적용하였다. 4장에서는 블록 구조요소 분해의 예를 보여주고 5장에서는 결론을 맺는다.

## 2. 디지털 블록다각형의 분해

### 2.1 디지털 블록다각형

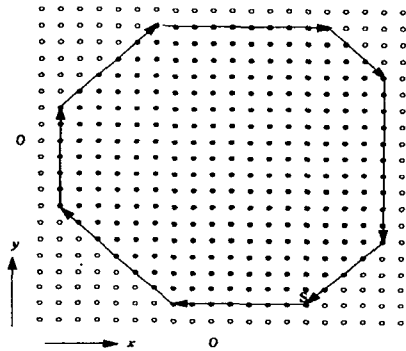
2차원 불연속공간(2D discrete space),  $Z^2$ 에서는 각 점의 좌표에서  $x$ 와  $y$  성분이 정수로 나타내어진다. 통상적으로 2차원 이진영상은 직사각형모양의 2차원 불연속공간 영역 내부의 각 점에 위치한 화소(pixel)에 화소값을 지정하여 나타낸다. 이진영상의 경우에 화소 값은 0 또는, 1의 두 정수 값으로 나타내어지며 해당 화소가 각각 배경(background), 또는 물체foreground)의 영역에 해당되는 부분에 위치해 있음을 나타낸다. 수학적 형상학에서는 이진 영상을 1인 값을 가지는 화소들의 집합으로 간주한다.

디지털 블록다각형은 2차원 불연속공간,  $Z^2$ 의 유한 부분집합으로서 이 집합이 나타내는 이진영상의 물체 영역의 경계가 0<sup>a1</sup> a2 a3 a4 a5 a6 a7 a와 같은 형식의 경계체인 코드(boundary chain code)[15]로 나타내어질 수 있고 영역 내부에 빈 점(hole)이 없는 집합으로 정의된다. 이 논문에서의 블록 구조요소는 디지털 블록다각형으로 정의된다. <그림 1>은 체인 코드의 방향을 나타내며, <그림 2>는 경계가 점 S로부터 시작하여 체인 코드 0<sup>1</sup> 1<sup>2</sup> 6<sup>3</sup> 5<sup>4</sup> 9<sup>5</sup> 3<sup>6</sup> 10<sup>7</sup> 4로 나타내어지는 디지털 블록다각형의 예를 보여준다. 이후의 설명에서 블록다각형은 디지털 블록다각형을 의미한다.



<그림 1> 경계 체인코드 방향

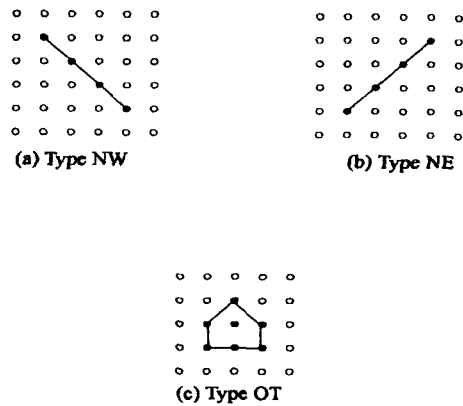
블록다각형은 8개의 변을 가지며 각 변들은 하나의 체인 코드 방향과 대응된다. 블록다각형  $A$ 의  $i$ 번째 변의 길이는  $e(A, i)$ 로 표기하며  $A$ 의 경계 체인 코드에서  $i$ 번째 체인 코드의 반복 회수로 정의된다. 즉, 블록다각형  $A$ 의 경계 체인 코드가  $0\ a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a$ 이면  $i=0, \dots, 7$ 에 대하여  $e(A, i) = C_i$  이다. <그림 2>의 블록다각형을  $B$ 라고 하면, 0, 3, 6 번째 변의 길이는 각각  $e(B, 0) = C_0 = 7$ ,  $e(B, 3) = C_3 = 5$ ,  $e(B, 6) = C_6 = 10$  이다.



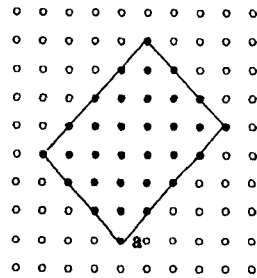
<그림 2> 디지털 블록다각형의 예

비단일점 블록다각형이란 2개 이상의 화소의 집합으로 이루어진 블록다각형을 말한다. 비단일점 블록다각형들을 다음과 같이 분류한다. 블록다각형  $P$ 의 변들이  $e(P, 1) = e(P, 5) > 0$  이고  $i = 0, 2, 3, 4, 6, 7$ 에 대하여  $e(P, i) = 0$ 인 조건을 만족시키면  $P$ 는 NW(north-west)형이다. 블록다각형  $P$ 의 변들이  $e(P, 3) = e(P, 7) > 0$ 이고  $i = 0, 1, 2, 4, 5, 6$ 에 대하여  $e(P, i) = 0$ 인 조건을 만족시키면  $P$ 는 NE(north-east)형이다. 그리고 NW, 또는 NE형이 아닌 모든 비단일점 블록다각형들은 OT(other type)형으로 분류된다. <그림 3>은 NW, NE, OT 형의 블록다각형의 예를 보여준다. 또한, 대각 방향의 변의 길이가 0보다 크고, 비대각 방향의 변의 길이가 0

인 비단일점 블록다각형은 마름모꼴 블록다각형으로 정의된다. 즉, 비단일점 블록다각형  $P$ 의 변들 간에  $e(P, 1) = e(P, 5) > 0$ ,  $e(P, 3) = e(P, 7) > 0$  이고,  $i = 0, 2, 4, 6$ 에 대하여  $e(P, i) = 0$ 인 관계가 만족되면  $P$ 는 마름모형 블록다각형이다. <그림 4>는 마름모꼴 블록다각형의 한 예를 보여준다.



<그림 3> 디지털 블록다각형의 분류



<그림 4> 마름모형 블록다각형

### 2.2 디지털 블록다각형의 형상 분해

이 절에서는 블록다각형들의 위치를 무시하고 형상만을 고려하는 형상 분해에 대하여 논의한다. 만일, 두 화소 집합  $A, B$ 가 벡터  $t$ 에 대하여  $A = B_t$  이면  $A$ 와  $B$ 는 형상동가라고 정의되고  $A \cong B$ 로 표기한다. 여기에서

$B_i$ 는  $B$ 의 모든 원소 점이 각각 벡터  $t$ 로 이동 변환된 점들을 원소로 가지는 집합이다. 두 집합이 형상등가 일 경우 두 집합이 형성하는 영상의 모양은 서로 닮은꼴이다.

정리 1. 집합  $P, Q, R$ 이 디지털 블록다각형 일 때, 만일

$$P \cong Q \oplus R \quad (6)$$

이면,  $i = 0, \dots, 7$ 에 대하여

$$d(P, i) = d(Q, i) + d(R, i) \quad (7)$$

이 성립한다.

$P$ 와  $Q \oplus R$ 이 형상등가일 때 즉,  $P$ 가  $Q$ 와  $R$ 로 분해가 가능할 때  $P$ 의  $i$ 번째 변의 길이는 이 변과 대응되는  $Q$ 와  $R$ 의 변의 길이의 합과 같다.

정리 1의 역은  $P$ 가 마름모형 블록다각형인 경우를 제외하고 항상 성립한다.  $P$ 가 마름모형 블록다각형인 경우,  $Q$ 와  $R$  중 하나가 NE형이고 다른 하나가 NW형 일 때 (7) 식이 만족될 수 있다. 그러나 NE형과 NW형의 블록다각형을 팽창연산 시켰을 때는 그림 4와 같이 블록다각형이 아닌 장기관 무늬를 갖는 마름모 형태의 영상을 얻게 된다. 따라서 정리 1의 역은  $P$ 가 마름모형이 아닌 경우와 마름모형인 경우로 나누어 각각 정리 2와 정리 3으로 명제화할 수 있다.

정리 2 :  $P$ 가 마름모형이 아닌 블록다각형일 때, 만일,  $i = 0, \dots, 7$ 에 대하여,

$$d(P, i) = d(Q, i) + d(R, i) \quad (8)$$

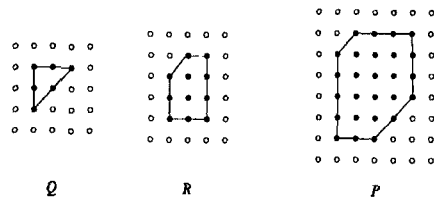
이면, 다음이 성립한다.

$$P \cong Q + R \quad (9)$$

정리 1, 2는 하나의 블록다각형이 두 블록다각형으로 분해되기 위한 필요충분조건을 나타낸다. 마름모형이 아닌 블록다각형  $P$ 가 주어졌을 때,  $P$ 가 임의의 블록다각형  $Q$ 와  $R$ 로 분해될 필요충분조건은  $P$ 의  $i$ 번째 변의 길이가  $Q$ 와  $R$ 의  $i$ 번째 변의 길이의 합과 같아야 한다는 것이다. <그림 5>는 정리 1과 2의 예를 보여준다. 이 그림에서  $P \cong Q + R$  이고 각 다각형의 변의 길이에 대하여  $d(P, 0) =$

$$\begin{aligned} d(Q, 0) + d(R, 0) = 2 \quad d(P, 1) = d(Q, 1) + d(R, 1) = 0 \\ d(P, 2) = d(Q, 2) + d(R, 2) = 4 \quad \dots \quad d(P, 7) = d(Q, 7) + d(R, 7) = 2 \end{aligned}$$

등으로  $i = 0, \dots, 7$ 에 대하여  $d(P, i) = d(Q, i) + d(R, i)$ 을 만족한다.



<그림 5> 정리 1과 2의 예

블록다각형  $P$ 가 마름모형인 경우, 분해의 필요충분조건에는 정리 3에서와 같이 블록다각형  $Q$ 와  $R$  중 하나 이상이 OT형 블록다각형이어야 한다는 조건이 추가된다.

정리 3 :  $P$ 가 마름모형 블록다각형일 때, 만일,  $i = 0, \dots, 7$ 에 대하여,

$$d(P, i) = d(Q, i) + d(R, i) \quad (10)$$

이고  $Q$ 와  $R$  중 하나 이상이 OT형 블록다각형이면, 다음이 성립한다.

$$P \cong Q + R \quad (11)$$

두 블록다각형을 팽창연산하면 두 블록다각형 중 한 쪽이 NW형이고 다른 한 쪽이 NE형인 경우를 제외하고는 팽창연산의 결과로서 블록다각형을 얻게 된다. 그러므로 정리 1, 2의 분해조건은 정리 4에서와 같이 선형조합(linear combination)의 형태로 확장될 수 있다. 이하의 설명에서 점집합  $A$ 의  $n$ -겹 팽창  $nA$ 는  $n$ 개의  $A$ 를  $n-1$  번의 팽창연산으로 결합한 것이다.

정리 4.  $P$ 와  $Q_1, Q_2, \dots, Q_n$ 가 블록다각형이고  $P$ 는 마름모형이 아닐 때

$$P \cong a_1 Q_1 \oplus \dots \oplus a_n Q_n \quad (12)$$

가 만족될 필요충분조건은  $i = 0, \dots, 7$ 에 대하여

$$d(P, i) = \sum_{k=1}^n a_k d(Q_k, i) \quad (13)$$

이다.

정리 4는 마름모형이 아닌 블록다각형  $P$ 가  $a_1$ 개의  $Q_1$ ,  $a_2$ 개의  $Q_2$ , ...,  $a_n$ 개의  $Q_n$ 의 분해 조합으로 형상분해되기 위한 필요충분조건을 나타낸다. 즉, 순서쌍  $(a_1, \dots, a_n)$ 이 식 (13)을 만족시키면  $a_1Q_1 \oplus \dots \oplus a_nQ_n$ 은  $P$ 와 형상등가이다. 역으로,  $a_1Q_1 \oplus \dots \oplus a_nQ_n$ 가  $P$ 와 형상등가이면 순서쌍  $(a_1, \dots, a_n)$ 은 식 (13)을 만족시킨다. 따라서 (13)을 만족시키는 순서쌍의 집합은  $P$ 의 기본블록다각형 집합  $\{Q_1, \dots, Q_n\}$  으로의 모든 가능한 분해조합의 집합과 1 대 1 대응관계에 있다.  $P$ 의 기본블록다각형 집합으로의 분해조합은 식 (13)에 의하여 결정되므로,  $P$ 의 형상 분해조합을 구하기 위해서는 식 (13)을 만족시키는 순서쌍  $(a_1, \dots, a_n)$ 을 구하면 된다.

마름모형 블록다각형의 분해에 대한 필요충분조건은 정리 1과 3으로부터 정리 5와 같이 유도해 낼 수 있다.

정리 5.  $P$ 와  $Q_1, Q_2, \dots, Q_n$ 이 블록다각형이고  $P$ 는 마름모형일 때

$$P \cong a_1Q_1 \oplus \dots \oplus a_nQ_n \quad (14)$$

가 만족될 필요충분조건은  $i = 0, \dots, 7$ 에 대하여

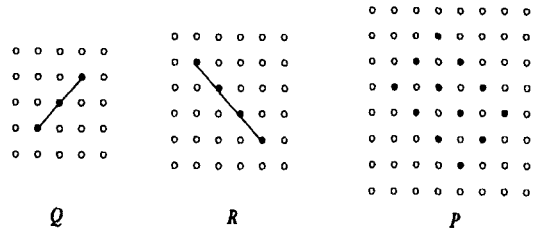
$$\alpha(P, i) = \sum_{k=1}^n a_k \alpha(Q_k, i) \quad (15)$$

$$S_{OT} = \sum_{k=1}^n a_k > 0 \quad (16)$$

$Q_k$  is type OT.

이다.

마름모형 블록다각형의 분해에 대한 필요충분조건에는 식 (16)에 의하여 분해조합 안에 OT 형의 블록다각형이 한 개 이상 포함되어야한다는 조건이 추가된다.



<그림 6> NE 형과 NW 형 블록다각형의 팽창연산의 결과

### 2.3 블록다각형의 위치 조건

지금까지의 블록다각형의 분해에 대한 논의에서 블록다각형의 위치는 고려하지 않았다. 이 절에서는 블록다각형의 위치에 대하여 논의한다. 아래의 설명에서  $\min_x(A)$ 는 점집합  $A$ 의 원소가 가지는 최소의  $x$  좌표를 의미한다. 마찬가지로  $\min_y(A)$ 는 최소의  $y$  좌표를 의미한다.

정리 6 :  $A, B, C$ 가  $Z^2$ 의 유한 부분집합일 때, 만일

$$C = A \oplus B \quad (17)$$

이면

$$\begin{aligned} \min_x(C) &= \min_x(A) + \min_x(B) \\ \min_y(C) &= \min_y(A) + \min_y(B) \end{aligned} \quad (18)$$

이다.

정리 6은 정리 7과 같이 선형조합 형태로 확장될 수 있다.

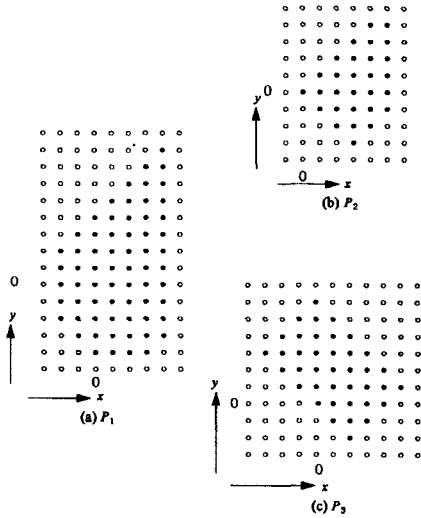
정리 7:  $P$ 와  $Q_1, Q_2, \dots, Q_n$ 이  $Z^2$ 의 유한 부분집합일 때, 만일

$$P = a_1Q_1 \oplus \dots \oplus a_nQ_n \quad (19)$$

이면

$$\begin{aligned} \min_x(P) &= \sum_{k=1}^n a_k \min_x(Q_k) \\ \min_y(P) &= \sum_{k=1}^n a_k \min_y(Q_k) \end{aligned} \quad (20)$$

이다.



<그림 7> 블록 구조요소의 예

2.4 블록다각형의 분해조건

정리 4와 7로부터 마름모형이 아닌 블록다각형  $P$ 가  $a_1$ 개의  $Q_1$ ,  $a_2$ 개의  $Q_2$ , ...,  $a_n$ 개의  $Q_n$ 을 포함하는 분해조합으로 분해되기 위한 필요충분조건은 다음과 같다.

형상 조건 :

$i = 0, \dots, 7$ 에 대하여

$$e(P, i) = \sum_{k=1}^n a_k e(Q_k, i) \tag{27}$$

위치 조건 :

$$\min_x(P) = \sum_{k=1}^n a_k \min_x(Q_k)$$

$$\min_y(P) = \sum_{k=1}^n a_k \min_y(Q_k) \tag{28}$$

마름모형 블록다각형인 경우에는 정리 5에 의하여 다음의 마름모 조건이 추가되어야한다.

마름모 조건 :

$$S_{OT} = \sum_{k=1}^n a_k > 0$$

$Q_k$  is type OT. (29)

마름모형이 아닌 블록다각형  $P$ 의 기본블록다

각형 집합  $\{Q_1, \dots, Q_n\}$ 으로의 분해는 형상조건과 위치조건에 의하여 구하여질 수 있다.  $P$ 와  $\{Q_1, \dots, Q_n\}$ 에 대하여 형상조건과 위치조건을 만족하는 순서쌍 해  $(a_1, \dots, a_n)$ 는  $a_1$ 개의  $Q_1$ ,  $a_2$ 개의  $Q_2$ , ...  $a_n$ 개의  $Q_n$ 으로 구성된 분해조합을 결정한다. 필요충분 조건을 만족하는 순서쌍 해  $(a_1, \dots, a_n)$ 의 집합은 분해조합들의 집합과 1 : 1 대응이므로 두 조건을 만족하는 모든 순서쌍 해  $(a_1, \dots, a_n)$ 의 집합은  $P$ 의 기본블록다각형 집합  $(a_1, \dots, a_n)$ 으로 구성된 모든 분해조합을 나타낸다. 마름모형인 블록다각형의 분해의 경우에는 순서쌍 해가 형상 조건과 분해조건 외에 마름모 조건을 만족시켜야 한다.

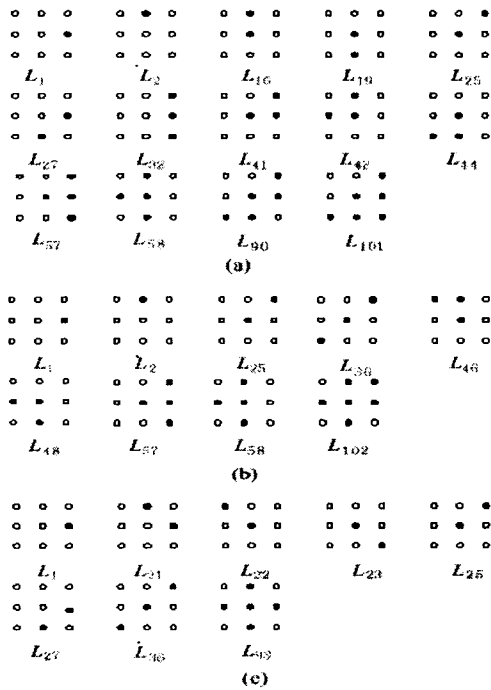
3. 최적 지역 분해

블록 구조요소의 최적분해조합은 실행할 때 최소의 계산량을 필요로 하는 분해조합이다. 최적분해조합은 팽창연산이 실행되는 계산 환경과 구현방법에 따라 달라진다. 이 절에서는 블록다각형 분해의 필요충분조건을 이용하여 블록 구조요소의 최적 지역 분해를 구하는 방법에 대하여 논의한다. 지역분해란 구조요소를 지역 블록 구조요소의 조합으로 분해하는 것을 뜻한다. 지역 블록 구조요소는 원점에 위치한  $3 \times 3$  크기의 지역 윈도우에 포함될 수 있는 블록 구조요소이다. 지역 블록 구조요소의 예는 <그림 8>에 나타나있다.

3.1 블록 구조요소의 지역 분해

지역 블록 구조요소는 모두 117개이고 각각  $L_0, L_1, \dots, L_{116}$ 으로 표시한다. 지역 블록 구조요소의 예가 <그림 8>에 나타나 있다. 블록 구조요소  $P$ 와 지역 블록 구조요소 집합  $\{L_0, \dots, L_{116}\}$ 에 대하여 형상조건과 위치조건을 만족하는 순서쌍  $(a_0, a_1, \dots, a_{116})$ 은 블록 구조요소  $P$ 의 지역 블록 구조요소 집합으로의 분해조합을 결정한다. 순서쌍  $(a_0, a_1, \dots, a_{116})$

에 의하여 결정되는 분해조합은  $a_0$  개의  $L_0$ ,  $a_1$  개의  $L_1, \dots, a_{116}$  개의  $L_{116}$ 으로 구성된다. 형상조건과 위치조건은 분해를 위한 필요충분조건이므로 두 조건을 만족하는 모든 순서쌍해의 집합과  $P$ 의 가능한 모든 지역 분해조합의 집합은 1 : 1 대응관계에 있다.



<그림 8> 그림 7의 블록 구조요소의 최적분해조합에 나타나는 지역 블록 구조요소

3.2 비용함수

입력의 입력 영상을 구조요소  $Q_k$ 에 의하여 팽창연산을 수행하기 위하여 소요되는 비용 또는 계산량을  $c_k$ 라고 할 때  $a_1$ 개의  $Q_1, a_2$ 개의  $Q_2, \dots, a_n$ 개의  $Q_n$ 의 구조요소 조합으로 일련의 팽창연산을 수행하는 동안 소요되는 총 비용 또는 총 계산량은 비용함수

$$\sum_{k=1}^n a_k c_k \tag{30}$$

로 나타낼 수 있다. 비용함수는 분해조합에 포

함된 구조요소들로 일련의 팽창연산을 수행할 때 필요한 총 계산량을 나타낼 수 있다. 팽창연산이 실행되는 계산환경 및 구현방법에 대하여 각각 다른 비용함수를 용이하게 유도할 수 있으므로 비용함수는 팽창연산의 여러 가지 구현방식에 대한 분해 조합의 최적화 기준을 효과적으로 나타낸다. 다음에서는 여러 가지 환경에서의 비용함수를 유도한다.

폰노이만 형식의 구조를 갖는 컴퓨터에서는 팽창연산이 아래와 같이 구현 될 수 있다 [2]. 아래에서  $A, B, C$ 는 각각 입력영상, 구조요소, 출력영상을 나타낸다.

알고리즘 1.

Let  $C = \emptyset$ .

For each  $(b_x, b_y) \in B$

    Translate  $A$  by vector  $(b_x, b_y)$  and let it  $A'$ .

    Let  $C = C \cup A'$ .

위의 알고리즘에서 영상끼리의 합집합 연산은 영상내의 이진 화소의 OR 연산으로 구현될 수 있다. 위의 알고리즘을 수행할 때 소요되는 계산량은 구조요소  $B$ 에 포함된 화소의 개수와 비례한다. 따라서  $a_1$ 개의  $Q_1, a_2$ 개의  $Q_2, \dots, a_n$ 개의  $Q_n$ 의 구조요소로 일련의 팽창연산을 수행할 때의 비용함수는

$$\sum_{k=1}^n a_k d_k \tag{31}$$

로 나타낼 수 있다. 여기서  $d_k$ 는 구조요소  $Q_k$ 에 포함된 점의 개수이다.

MPP 형식과 같은 병렬배열구조를 가진 컴퓨터[16]에서는 입력영상의 각 화소를 한 개의 처리기(processor)에 할당하고 영상전체를 병렬이동(parallel shift)한 후 OR 연산을 수행함으로써 팽창연산을 구현할 수 있다. 이 경우 팽창연산의 계산량은 입력영상의 병렬이동 횟수에 비례하므로  $a_1$ 개의  $Q_1, a_2$ 개의  $Q_2, \dots, a_n$ 개의  $Q_n$ 의 구조요소 조합으로 일련의 팽창연산을 수행할 때의 비용함수는



$$\sum_{k=1}^n a_k s_k \quad (31)$$

로 나타낼 수 있다. 식 (31)에서  $s_k$ 는 구조요소  $Q_k$ 에 의한 팽창연산을 수행하기 위하여 필요한 영상의 병렬이동의 회수이다. 4점 연결(4-connected)된 MPP 타입의 병렬 배열 구조를 가진 컴퓨터에서는 각 처리기가 동서남북에 위치한 이웃 처리기와 직접 연결되어 있으므로 각 화소를 동서남북 방향의 이웃 화소 위치로 이동시키기 위해서는 한 번의 병렬이동연산이 필요하고, 대각 방향의 이웃 화소 위치로 이동시키기 위해서는 두 번의 병렬이동연산이 필요하다. 따라서 지역 구조요소  $Q_k$ 에 의하여 팽창연산을 실행할 때 필요한 병렬이동연산의 회수  $s_k$ 는

$$s_k = p_s + 2p_w \quad (32)$$

로 나타낼 수 있다. 여기에서  $p_s$ 와  $p_w$ 는 각각  $Q_k$ 에 포함된 동서남북 방향의 화소의 개수와 대각방향의 화소의 개수이다. 8점 연결(8-connected)된 병렬 배열 구조에서는 각 화소를 8개의 이웃 화소 위치로 이동시키기 위해서 한 번의 병렬이동 연산이 필요하다. 지역 구조요소  $Q_k$ 에 의하여 팽창연산을 실행할 때 필요한 병렬이동연산의 회수,  $s_k$ 는  $Q_k$ 가 가진 화소 중 원점에 위치한 화소를 제외한 나머지 화소의 개수와 같다.

Cytocomputer와 같은 파이프라인형의 컴퓨터구조에서는 파이프라인의 각 단계는 지역 구조요소에 의한 팽창연산을 한 번에 수행할 수 있다. 일련의 지역 구조요소에 의하여 팽창연산을 수행할 때, 출력지연시간은 이 팽창연산들을 구현하기 위하여 필요한 파이프라인 단계 수, 또는 지역 구조요소의 개수와 비례한다. 출력 지연 시간을 최소화하기 위한 최적 분해의 기준은 분해조합이 포함하는 지역 구조요소의 개수이다.  $a_1$ 개의  $Q_1$ ,  $a_2$ 개의  $Q_2$ , ...,  $a_n$ 개의  $Q_n$ 의 구조요소 조합으로 일련의 팽창연산을 수행할 때 소요되는 파이프라인 단계 수 또는 분해조합이 포함하는 구조요소

의 수는 다음과 같은 비용함수로 나타낼 수 있다.

$$\sum_{k=1}^n a_k \quad (33)$$

위에서와 같이 팽창연산이 실행되는 다양한 계산환경 및 구현방법에 대하여 각 구조요소에 의한 팽창연산에 소요되는 계산량을 구할 수 있고 다른 비용함수를 유도할 수 있다.

### 3.3 최적 지역 분해조합

블록 구조요소의 최적 지역 분해조합은 팽창연산을 수행 할 때 최소의 계산량을 필요로 하는 지역 분해조합이다. 블록 구조요소  $P$ 와 지역 블록 구조요소 집합  $\{L_0, L_1, \dots, L_{116}\}$ 에 대하여 분해의 필요충분조건을 만족하며 동시에 비용함수를 최소화하는 순서쌍  $(a_0, a_1, \dots, a_{116})$ 은 최적 지역 분해조합을 결정한다. 따라서 최적 지역 분해조합을 찾는 문제는 형상조건과 분해조건을 만족하며 비용함수를 최소화시키는 순서쌍을 구하는 문제로 변환될 수 있다. 이러한 순서쌍 해는 선형 정수 계획법(linear integer programming)으로 구할 수 있다[17]. 이 때 선형 정수 계획법의 제한 조건은 형상 조건과 위치 조건의 선형방정식이며 목적함수는 비용함수를 최소화하는 것이다.

## 4. 분해 예제

<그림 7>에 나타난 예제 블록 구조요소에 대하여 최적 지역 분해 조합을 구하고 원래의 구조요소와 직접 팽창 연산할 경우와 분해조합과 팽창연산 할 경우의 계산량을 비교하였다. <그림 7>에 나타난 구조요소들은 이진영상에서 분리된 영역(segmented region)의 특징 추출(feature extraction) 또는, 대조(matching)를 위하여 사용될 수 있는 구조요소들이다. <그림 8>은 <그림 7>의 예제 구조요소의 분해 조합에 나타나는 지역 블록 구조

요소들이다. <표 1>에 나타난 분해조합은 알고리즘 1의 팽창연산의 구현방법에 대하여 최적화된 분해조합이다. 분해의 최적화 기준은 분해조합 내의 지역 블록 구조요소들에 포함된 화소의 개수의 총합이다. <표 2>는 원래의 블록 구조요소에 의하여 팽창연산을 수행할 경우와 <표 1>에 나타난 분해조합으로 일련의 팽창연산을 수행할 경우의 계산량을 비교한 것이다. 표2에서 계산량은 각 입력영상 화소 당 필요한 참조 연산의 횟수이다. <표 3>과 <표 5>는 각각 4점 및 8점 연결된 병렬배열구조 컴퓨터에 대한 최적분해조합이며 원래의 구조요소를 적용한 경우와 최적분해조합을 적용한 경우의 계산량이 각각 <표 4>와 <표 6>에 비교되어 있다. <표 4>와 <표 6>에서의 계산량은 팽창연산을 수행하는 동안 필요한 병렬이동 연산의 총 회수이다. <표 7>은 Cytocomputer를 위한 최적분해조합을 나타낸다. <표 8>은 최적분해조합을 적용하여 파이프라인을 구성 할 때 필요한 파이프라인의 단계 수를 나타낸다. 팽창연산이 실행되는 각 구현 방식에 대하여, <그림 8>에서 나타나는 블록 구조요소에 의하여 직접 팽창연산을 수행하는 경우에 비하여 구조요소 들의 최적 지역 분해조합에 의하여 팽창연산을 수행하는 경우가 계산 속도가 대폭 향상됨을 알 수 있다. 선형 정수 계획법에 의하여 블록 구조 요소의 분해조합을 나타내는 해를 구하는 시간은 영상에 팽창 연산을 수행하는 시간에 비하여 무시할 수 있을 정도로 작다. 개인용 컴퓨터에서 예제에 나오는 분해 요소를 구하는 데 소요되는 계산시간은 0.1 초 미만이다. 또한, 컴퓨터 비전 시스템에서는 같은 구조요소를 반복하여 사용하는 경우가 많이 있고, 이러한 경우에는 분해조합을 구하는 데 걸리는 시간은 상대적으로 작아지게 된다.

<표 1> 폰노이만 구조의 컴퓨터를 위한 최적 분해조합

구조요소	최적분해조합
$P_1$	$L_{25} \oplus 2L_{32} \oplus L_{41} \oplus 2L_{42} \oplus 2L_{58}$
$P_2$	$L_1 \oplus L_{36} \oplus L_{46} \oplus 2L_{57}$
$P_3$	$2L_1 \oplus 4L_{21} \oplus L_{36} \oplus L_{93}$

<표 2> 폰 노이만 컴퓨터구조에서 원래 구조요소와 최적 지역 분해조합의 계산량 비교

구조요소	원래 구조요소	최적분해조합
$P_1$	66	25
$P_2$	29	15
$P_3$	39	18

<표 3> 4점 연결된 병렬 배열 처리를 위한 최적분해조합

구조요소	최적분해조합
$P_1$	$3L_1 \oplus 3L_{16} \oplus L_{19} \oplus 2L_{27} \oplus 3L_{42} \oplus 2L_{58}$
$P_2$	$3L_1 \oplus 2L_{25} \oplus L_{48} \oplus 2L_{58}$
$P_3$	$L_1 \oplus 4L_{21} \oplus L_{36} \oplus L_{27} \oplus L_{93}$

<표 4> 4 점 연결된 병렬 배열 처리에서 원래 구조요소와 최적 지역 분해조합의 계산량 비교

구조요소	원래 구조요소	최적분해조합
$P_1$	298	21
$P_2$	120	15
$P_3$	155	17

<표 5> 8점 연결된 병렬 배열 처리기를 위한 최적분해조합

구조요소	최적분해조합
$P_1$	$4L_{16} \oplus 3L_{25} \oplus L_{19} \oplus L_{27} \oplus 3L_{42} \oplus 2L_{58}$
$P_2$	$L_1 \oplus 2L_2 \oplus 2L_{25} \oplus L_{46} \oplus 2L_{58}$
$P_3$	$4L_1 \oplus 3L_{22} \oplus L_{23} \oplus 2L_{25} \oplus L_{33}$

<표 6> 8점 연결된 병렬 배열 처리기에서 원래 구조요소와 최적 지역 분해조합의 계산량 비교

구조요소	원래 구조요소	최적분해조합
$P_1$	194	17
$P_2$	92	13
$P_3$	114	14

<표 7> Cytocomputer를 위한 최적분해조합

구조요소	최적분해조합
$P_1$	$2L_2 \oplus 2L_{32} \oplus 2L_{58} \oplus L_{90} \oplus 3L_{101}$
$P_2$	$L_1 \oplus L_{25} \oplus L_{57} \oplus L_{102}$
$P_3$	$2L_1 \oplus 3L_{21} \oplus L_{25} \oplus 2L_{33}$

<표 8> Cytocomputer에서 최적 지역 분해 조합에 의하여 팽창을 수행하기 위하여 필요한 파이프라인 단계 수

구조요소	결과 출력의 지연시간
$P_1$	17
$P_2$	13
$P_3$	14

### 5. 결론

본 논문에서는 디지털 블록다각형의 분해에 대한 필요충분조건을 유도하였고 이 조건을 블록 구조요소의 지역 분해에 적용하였다. 구조요소의 분해에 대한 최적화 기준은 형상 연산을 실행하는 컴퓨터의 구조와 구현방법에 따라 달라진다. 이 논문에서 소개된 비용함수는 분해에 대한 최적화 기준을 효과적으로 나타내며 비용함수를 적절하게 선택함으로써 각 경우에 대하여 최적한 지역 분해조합을 구할 수 있다. 또한, 분해 예제를 통하여 블록 구조요소의 최적 지역 분해가 형상연산에 필요한 계산량을 크게 감소시킬 수 있음을 보여 주었다. 임의의 형상을 갖는 구조요소의 경우 블록 구조 요소와 같이 분해 조건 등을 수학적으로 나타내는 것이 불가능하다. 따라서 본 논문에서 소개된 방법은 블록 구조 요소의 조건에 적합한 구조요소에만 적용이 가능하다. 형상연산의 계산량, 또는 계산시간의 단축은 형상연산을 응용하는 군사용, 의학용 영상시스템과 같은 실시간 처리시스템의 수행 속도를 크게 향상시킬 수 있다.

### 참고문헌

- [1] G. Matheron, Random Sets and Integral Geometry, New York: Wiley, 1975.
- [2] R. M. Harlick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," IEEE Trans. on PAMI, vol. PAMI-9, no.4, pp. 532-550, 1987
- [3] J. Serra "Introduction to mathematical morphology," Computer Vision, Graphics and Image Processing, vol. 35, pp. 285-305, 1986.
- [4] J. Serra, Image Analysis and Mathematical Morphology, London : Academic Press, 1982.

- [5] P. Maragos and R. W. Schafer, "Morphological systems for multidimensional signal processing," Proceedings of the IEEE, vol. 78, no. 4, pp. 690-719, Apr. 1990.
- [6] X. Zhuang and R. M. Haralick, "Morphological structuring element decomposition," CVGIP 35, pp 370-382.
- [7] J. Xu, "Decomposition of convex polygonal morphological structuring elements into neighborhood subsets," IEEE Trans. on PAMI, vol. PAMI-13, no. 2, pp. 153-162, 1994.
- [8] S. R. Sternberg, "Biomedical image processing," Comput. Jan. 1983, pp. 22-34.
- [9] H. Park and R. T. Chin "Optimal decomposition of convex morphological structuring elements for 4-connected parallel array processors," IEEE Trans. on PAMI, vol. PAMI-16, no. 3, pp. 304-313, 1994.
- [10] Jianning Xu, "Morphological decomposition of 2-D binary shapes into convex polygons: a heuristic algorithm", IEEE Transactions on Image Processing , Volume: 10, Issue: 1 , Jan. 2001 pp. 61 - 71
- [11] Jianning Xu; "Morphological decomposition of 2-D binary shapes into conditionally maximal convex polygons", Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference , Volume: 2 , 13-16 Nov. 1994 pp. 96 - 100
- [12] Lam, W -K.; Li, C.-K, "Rotated texture classification by improved iterative morphological decomposition", Vision, Image and Signal Processing, IEE Proceedings- , Volume: 144 , Issue: 3 , June 1997, pp. 171 - 179
- [13] Lam, W.K.; Li, C.K, "Scale invariant texture classification by iterative morphological decomposition", Electronics Letters , Volume: 32 , Issue: 6 , 14 March 1996, pp. 534 - 535
- [14] Haris, K.; Efstratiadis, S.; Maglaveras, N. "Hierarchical image segmentation based on contour dynamics Image Processing", 2001. Proceedings. 2001 International Conference on Image Processing, Volume: 1 , 7-10 Oct. 2001, pp. 54 - 57.
- [15] H. Freeman, "Computer processing of line drawing images," Computer Surveys, vol. 6, no. 1, March 1974, pp. 57-98.
- [16] K. E. Batcher, "Design of a massively parallel processor," IEEE Trans. on Computers, vol. C-29, no. 9, pp. 836-840, 1980.
- [17] M. M. Syslo, N. Deo, and J. S. Kowalik, Discrete Optimization Algorithms, Englewood Cliff, NJ : Prentice-Hall, 1983.

주 작 성 자 : 은 승 엽

논문 투고일 : 2003. 07. 12

논문 심사일 : 2003. 08. 08(1차), 2003. 09. 18(2차)

심사판정일 : 2003. 09. 30

● 저자소개 ●

은승엽



1984 서울대학교 공과대학 전기공학과 학사

1988 Polytechnic University 대학원 컴퓨터 및 정보과학과 석사

1995 Polytechnic University 대학원 컴퓨터 및 정보과학과 석사

1996~1997 한국통신 연구개발본부 선임연구원

1997~2000 홍익대학교 과학기술대 전자전기 컴퓨터정보통신 공학부 조교수

2000~현재 한국 항공대학교 전자 정보통신 컴퓨터 공학부 조교수

관심분야: 컴퓨터 비전, 패턴 인식, 영상 처리