# 볼록 구조자를 위한 최적 분리 알고리듬

論 文
48A - 9 - 16

# An Optimal Decomposition Algorithm for Convex Structuring Elements

溫 承 燁*
(Syng-Yup Ohn)

**Abstract** - In this paper, we present a new technique for the local decomposition of convex structuring elements for morphological image processing. Local decomposition of a structuring element consists of local structuring elements, in which each structuring element consists of a subset of origin pixel and its eight neighbors. Generally, local decomposition of a structuring element reduces the amount of computation required for morphological operations with the structuring element. A unique feature of our approach is the use of linear integer programming technique to determine optimal local decomposition that guarantees the minimal amount of computation. We defined a digital convex polygon, which, in turn, is defined as a convex structuring element, and formulated the necessary and sufficient conditions to decompose a digital convex polygon into a set of basis digital convex polygons. We used a set of linear equations to represent the relationships between the edges and the positions of the original convex polygon, and those of the basis convex polygons. Further, a cost function was used to represent the total processing time required for computation of dilation/erosion with the structuring elements in a decomposition. Then integer linear programming was used to seek an optimal local decomposition, that satisfies the linear equations and simultaneously minimize the cost function.

**Key Words** : Morphological image processing, Structuring element, Decomposition, Integer linear programming

## 1. Introduction

Mathematical morphology is a powerful tool for image processing and computer vision [1], [2], [3], [4], [5]. Morphological image processing is based on the following two basic operations called dilation and erosion. In the below, $A$ and $B$ are subsets of $E^N$, where $E^N$ is the $N$-dimensional Euclidean or digital space.

Dilation:

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

Erosion:

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

In the above, $A$ generally represents an image and $B$ is called a structuring element. Different image processing operations could be achieved by choosing structuring elements of appropriate sizes and shapes, and putting the dilation and erosion operations in chained sequences.

It's often inefficient to use a large structuring element when an input image has a large amount of data. Also, some parallel architectures can only compute with structuring elements that fit inside a 3×3 window centered on the origin. Therefore, it is desirable to decompose a large structuring element into a sequence of dilations of smaller structuring elements. By the chain rule for dilations [2], if a structuring element $B$ is decomposed into $B_1, B_2, \ldots, B_n$, i.e.

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

then the dilation of $A$ by $B$ can be computed by the sequence of dilations as

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

instead of a single dilation by the original structuring element. Generally, the amount of computation for the sequence of dilations as in (4) is less than that for a single dilation operation as $A \oplus B$. Similarly, the erosion of $A$ by $B$ can be computed by

$$(((A \ominus B_1) \ominus B_2) \ominus \ldots) \ominus B_n. \quad (5)$$

*正 會 員 : 弘益大 電·電·컴 工學部 傳任 講師
接受日字 : 1999年 7月 14日
最終完了 : 1999年 8月 11日

Since an erosion operation is defined as the dual operation of a dilation operation, the decomposition for an erosion operation can be similarly obtained like the decomposition for dilation operations, and we will omit the discussion on erosion operation.

Haralick and Zhuang first proposed methods for decomposition of structuring elements [6]. They developed algorithms for two-point decomposition of structuring elements for Image Flow machine. Since then, many researchers have developed algorithms and techniques for the local decomposition of convex shaped structuring elements for parallel image processing architectures. The local decomposition consists of the set of local structuring elements, which can be contained in 3×3 local window centered on the origin. Xu [7] reported an algorithm to get an optimal decomposition for Cytocomputer [8]. Park [9] reported an algorithm to find an optimal decomposition for four-connected MPP type machines. Kanungo [10] proved that there exists a linear transformation between 13 primal basis digital convex polygons and the 8 edges of an input digital convex polygon. However, general methods for finding optimal decompositions for different types of pipelined or parallel processing architectures are yet to be found. In this paper, we present such a method. The method is based on the Shephard's theorem [11], [12] for the decomposition of Euclidean convex polytopes. We derived a set of linear constraints on the length of edges of digital convex polygons or convex structuring elements in digital space. This set of linear constraints will serve as the necessary and sufficient conditions to decompose a digital convex polygon into a set of basis digital convex polygons. We define a cost function that represents the total processing time or the cost to execute morphological operations with a set of basis structuring elements. The decomposition problem is formulated as an integer programming problem where we seek to minimize the cost (or objective) function, given the set of linear constraints. Our method results in an optimal decomposition with respect to the cost function, which in turn, represents the execution time. For different computing environments and algorithms, different cost functions could be defined, depending on how morphological operations are executed on the environment.

This paper is organized as follows. In Section 2, we show how morphological operations are implemented. In Section 3, we derive the necessary and sufficient conditions to decompose a digital convex polygon into a set of basis digital convex polygons. Section 4 presents a new technique to find an optimal local decomposition of a convex structuring element. Finally, Section 5 gives our conclusion.

## 2. Implementation Algorithm of Morphological Operations

On a plain von Neumann type computer which has a CPU and a main memory, dilation of input image $A$ by structuring element $B$ can be implemented as follows. In the following, each of $A$, $B$, and $C$ is the set of the coordinates of all the foreground pixels in input image, structuring element, and output image.

Dilation Algorithm :

Let $C=\emptyset$.

For each $(a_x, a_y) \in A$

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

    put $(a_x, a_y)+(b_x, b_y)$ into set $C$.

The complexity of the above algorithm depends on the number of addition operations The number of addition required for dilation of input image $A$ by structuring element $B$ is $mn$, in which $m$ and $n$ is the number of foreground pixels in $A$ and $B$.

Erosion Algorithm :

Let $C=\emptyset$.

For each $(a_x, a_y) \in A$

  if $(a_x, a_y)+(b_x, b_y) \in A$ for every $(b_x, b_y) \in B$,

    put $(a_x, a_y)$ into set $C$.

Similarly, the number of addition operations required for erosion of input image $A$ by structuring element $B$ is $mn$. Since the input image is predetermined, the amount of computation required for executing dilation or erosion is proportional to the number of foreground pixels in structuring element $n$.

## 3. Decomposition of Digital Convex Polygons

### 3.1 Digital Convex Polygons

In our method, a digital convex polygon is represented using its boundary chain codes[17]. See Figure 1 for the different chain code directions. In the following, a digital point denotes a point in 2-dimensional digital space $Z^2$, where the $x$ and $y$ coordinates of each point in $Z^2$ have an integer value.

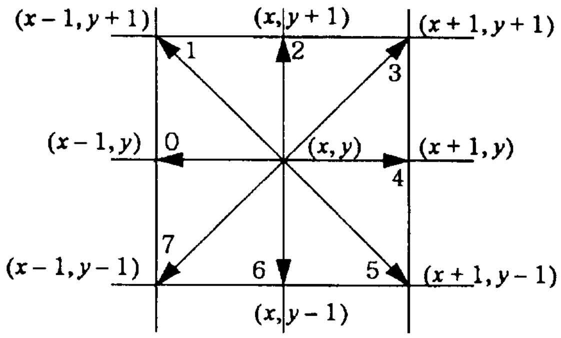


Fig. 1 Chain codes directions

*Definition 1*: The set of digital points $A$ is a digital convex polygon (DCPG) if the boundary of $A$ can be represented as a chain code in the form of $0^{c0}\ 1^{c1}\ 2^{c2}\ 3^{c3}\ 4^{c4}\ 5^{c5}\ 6^{c6}\ 7^{c7}$ and it has no hole inside.

See Figure 2 for an example DCPG $P$. The chain code representation of the boundary of $P$ is $0^7\ 1^6\ 2^6\ 3^5\ 4^9\ 5^3\ 6^{10}\ 7^4$, starting with points.

*Definition 2*: Suppose the boundary of DCPG $A$ is represented as chain code sequence $0^{c0}\ 1^{c1}\ 2^{c2}\ 3^{c3}\ 4^{c4}\ 5^{c5}\ 6^{c6}\ 7^{c7}$. The length of the $i$ th edge of DCPG $A$, denoted as $e(A, i)$, can be defined as the chain code length $c_i$ in the chain code sequence.

Note that the chain code length of an edge is different from the geometric length of the edge. the latter can be obtained by multiplying the former by $\sqrt{2}$ if the edge directions are diagonal, and remains the same if the edge directions are vertical or horizontal.

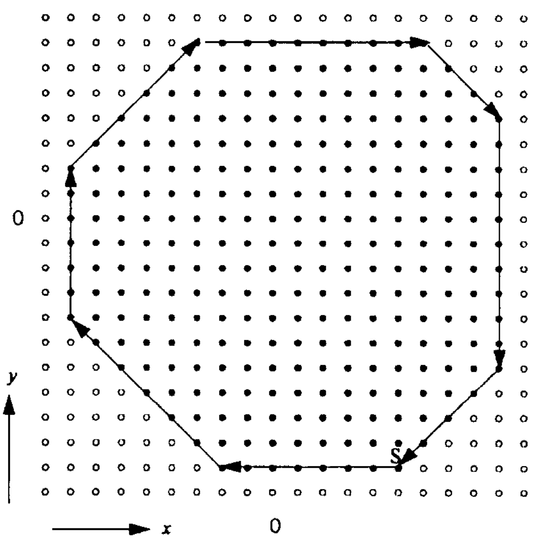


Fig. 2 An example digital convex polygon(DCPG). Arrows indicate 8 edges of the DCPG

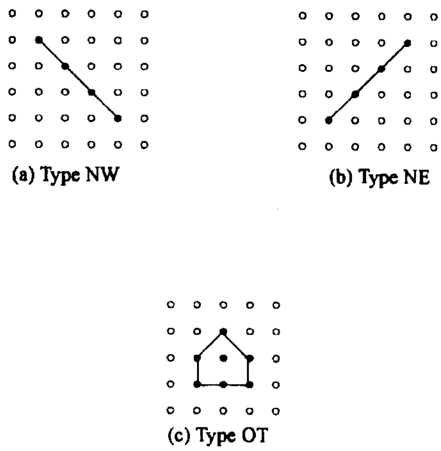


Fig. 3 Examples of 2d DCPGs and their boundaries

A non-singleton DCPG is classified according to its shape as follows.

*Definition 3*: A DCPG $P$ is of type NW if

$$e(P,1) = e(P,5)>0\ , \tag{6}$$

and, for $i$ = 0, 2, 3, 4, 6, 7,

$$e(P,i) = 0\ . \tag{7}$$

*Definition 4*: A DCPG $P$ is of type NE if

$$e(P,3) = e(P,7)>0\ , \tag{8}$$

and, for $i$ = 0, 1, 2, 4, 5, 6,

$$e(P,i) = 0\ . \tag{9}$$

*Definition 5*: A non-singleton DCPG $P$ is type OT if $P$ is neither of type NW or NE.

See Figure 3 for examples of type NW, NE, and OT DCPGs.

A type NW or type NE DCPG is an 8-connected image, and a type OT DCPG is a 4-connected image. We also define a special class of DCPGs as follows.

*Definition 6*: A non-singleton DCPG $P$ is said to be rhombus-shaped if

$$\begin{aligned} e(P,1) &= e(P,5)>0 \\ e(P,3) &= e(P,7)>0 \end{aligned}\ , \tag{10}$$

and, for $i$ =0, 2, 4, 6,

$$e(P,i) = 0\ . \tag{11}$$

See Fig. 4 for an example of rhombus-shaped DCPG.

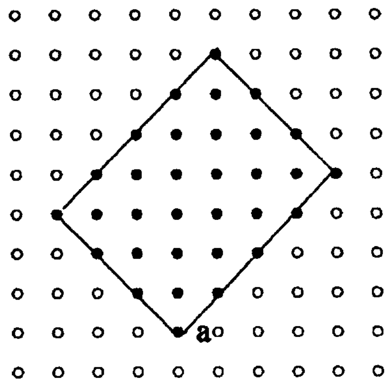


Fig. 4 An example of rhombus-shaped DCPG and its boundaries

### 3.2 Decomposition of DCPG

*Definition 7*: Two sets of points $A$ and $B$ are said to be shape-equivalent and denoted as $A \cong B$ if $A = B_t$ for a proper vector $t$, where $B_t$ represents the translation of $B$ by vector $t$.

The following proposition represents the relationships between the edges of original DCPG and decomposed DCPGs. We avoid rigorous proofs of propositions in this paper and show illustrative examples instead. Refer [20] for proofs of propositions.

*Proposition 1*: Suppose $P$, $Q$, and $R$ are DCPGs. If

$$P \cong Q + R\ , \tag{12}$$

then

$$e(P, i) = e(Q, i) + e(R, i) \tag{13}$$

for $i$ = 0, ..., 7.

Figure 5 shows an illustrative example of Proposition 1. In Figure 5, $e(Q, i) + e(R, i) = e(P, i)$ for $i$ = 0, ..., 7 and $P \cong Q + R$.

The converse of Proposition 1 is true with an exceptional case. In case that $P$ is rhombus shaped, Equation (13) can be satisfied by $Q$ and $R$ of which either one is type NE and the other is type NW. But, the dilation of a type NE and a type NW DCPG results a rhombic checker board image as in Figure 6. Thus, we exclude the exceptional case in the converse of Proposition 1 shown in Proposition 2.

*Proposition 2*: Suppose $P$ is not rhombus-shaped. If

$$e(P, i) = e(Q, i) + e(R, i) \tag{14}$$

for $i$ = 0, ..., 7, then

$$P \cong Q + R\ . \tag{15}$$

In the exceptional case that $P$ is rhombus-shaped, the converse of Proposition 2 is true when one of $Q$ and $R$ are type OT. The converse in the exceptional case is presented in Proposition 3.

*Proposition 3*: Suppose $P$ is rhombus-shaped. If

$$e(P, i) = e(Q, i) + e(R, i) \tag{16}$$

for $i$ = 0, ..., 7 and either one of $Q$ or $R$ is type OT, then

$$P \cong Q + R\ . \tag{17}$$

Proposition 1, 2, and 3 serve as the necessary and sufficient condition for a DCPG to be decomposed into two DCPGs. When $P$ is a non rhombus-shaped DCPG, the set of all the pairs of $Q$ and $R$ which satisfy the condition in Proposition 2 contains only and all the possible decomposition of DCPG $P$. Similarly, when $P$ is a rhombus-shaped DCPG, the set of all the pairs of $Q$ and $R$ which satisfy the condition in Proposition 3 contains only and all the possible decomposition of DCPG $P$.

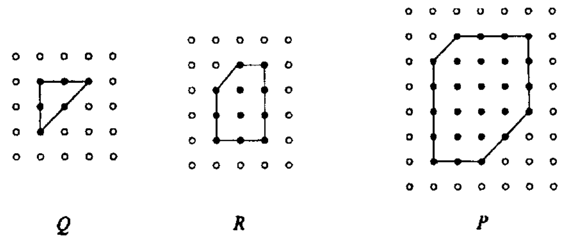


Fig. 5 Illustration of Proposition 1 and 2

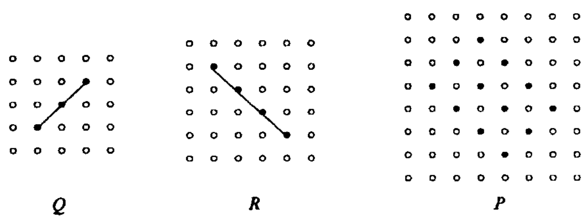


Fig. 6 Dilation of a type NE and a NW DCPGs. $Q \oplus R = P$

Dilation of any two DCPG, except the case in which one is type NW and the other is type NE, results a DCPG. Thus, from Proposition 1 and 2, the decomposition condition can be extended to a linear combination form as shown in Proposition 4.

*Proposition 4*: Suppose $P$ and $Q_k$, where $k$ = 1, ..., $n$, are DCPGs and $P$ is not rhombus-shaped, then

$$P \cong a_1 Q_1 \oplus \ldots \oplus a_n Q_n \qquad (18)$$

if and only if, for $i$ = 0, ..., 7,

$$e(P, i) = \sum_{k=1}^{n} a_k e(Q_k, i) \ . \qquad (19)$$

Similarly, from Proposition 1 and 2, we have Proposition 5.

*Proposition 5*: Suppose $P$ and $Q_k$, where $k$ = 1, ..., $n$, are DCPGs and P is rhombus-shaped.

$$P \cong a_1 Q_1 \oplus \ldots \oplus a_n Q_n \qquad (20)$$

if and only if, for $i$ = 0, ..., 7,

$$e(P, i) = \sum_{k=1}^{n} a_k e(Q_k, i) \ , \qquad (21)$$

and

$$S_{OT} > 0 \ , \qquad (22)$$

where $S_{OT} = \sum_{k=1,\ Q_k \text{ is type OT.}}^{n} a_k$.

Propositions 4 and 5 provide the necessary and sufficient condition for a DCPG $P$ to be decomposed into $a_1Q_1$s, ..., $a_nQ_n$s by considering the shape of a DCPG. Given a non rhombus-shaped DCPG $P$ and the set of basis DCPGs $\{Q_1, \ldots, Q_n\}$, if the solution $n$-tuple $(a_1, \ldots, a_n)$ satisfies (19), then composition $a_1Q_1 \oplus \ldots \oplus a_nQ_n$ is shape-equivalent to $P$. Furthermore, if any composition of $a_1Q_1 \oplus \ldots \oplus a_nQ_n$ is shape-equivalent to $P$, then the $n$-tuple $(a_1, \ldots, a_n)$ satisfies (19). Thus, the solution $n$-tuple space for (19) contains all and only $n$-tuples $(a_1, \ldots, a_n)$ such that $P$ and $a_1Q_1 \oplus \ldots \oplus a_nQ_n$ are shape-equivalent. In case of rhombus-shaped DCPG $P$, similar discussion applies for Equation (21) and (22).

### 3.3 Position of DCPG in Decomposition

So far we haven't considered the positions of the DCPGs. In this section, we discuss the relationship between the positions of the DCPGs. In the following, $\min_x(A)$ denotes the minimum $x$ coordinate of the region occupied by the set of points $A$. Similarly, $\min_y(A)$ denotes the minimum $y$ coordinate.

*Proposition 6*: Suppose $A$, $B$, and $C$ are sets of points in $Z^2$. If

$$C = A \oplus B \ , \qquad (23)$$

then

$$\begin{aligned} \min_x(C) &= \min_x(A) + \min_x(B) \\ \min_y(C) &= \min_y(A) + \min_y(B) \end{aligned} \ . \qquad (24)$$

Also, Proposition 6 can be extended to a linear combination form.

*Proposition 7*: Suppose $P$ and $Q_k$s, where $k$ = 1, ..., $n$, are sets of points in $Z^2$. If

$$P = a_1 Q_1 \oplus \ldots \oplus a_n Q_n \ , \qquad (25)$$

then

$$\begin{aligned} \min_x(P) &= \sum_{k=1}^{n} a_k \min_x(Q_k) \\ \min_y(P) &= \sum_{k=1}^{n} a_k \min_y(Q_k) \end{aligned} \ . \qquad (26)$$

From Propositions 4, 5 and 7, the necessary and sufficient conditions for a DCPG $P$ to be decomposed into $a_1Q_1, \ldots, a_nQ_n$ are as follows:

D.1) For $i$ = 0, ..., 7,

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

D.2) Only in the case that $P$ is rhombus shaped,

$$S_{OT} > 0 \ , \qquad (28)$$

where $S_{OT} = \sum_{k=1,\ Q_k \text{ is type OT.}}^{n} a_k$

D.3)

$$\min_x(P) = \sum_{k=1}^{n} a_k \min_x(Q_k) \qquad (29)$$
$$\min_y(P) = \sum_{k=1}^{n} a_k \min_y(Q_k)$$

Conditions D.1 and D.2 consider shape only. If the two conditions are satisfied, $P$ and $a_1Q_1 \oplus \ldots \oplus a_nQ_n$ are shape-equivalent. If condition D.3 is satisfied in addition to conditions D.1 and D.2, P and $a_1Q_1 \oplus \ldots \oplus a_nQ_n$ are located at the same position. Since D.1, D.2, and D.3 are the necessary and sufficient conditions, the space of solution $n$-tuples of $(a_1, \ldots, a_n)$ for conditions D.1, D.2, and D.3 contain all and only $n$-tuple $(a_1, \ldots, a_n)$ such that $a_1Q_1, \ldots, a_nQ_n$ is the decomposition of $P$.

## 4. Optimal Local Decomposition of Convex Structuring Elements

In this section, we develop a technique to find the optimal decomposition of a convex structuring element into a set of local convex structuring elements using the decomposition conditions we developed in Section 3. A structuring element is said to be convex if it forms a DCPG. If a convex structuring element can be contained in a 3×3 window centered on the origin, then it is called a local convex structuring element, or a local basis in short.

### 4.1 Local Decomposition of a Convex Structuring Element

The set of all local basis contains 117 elements, and the elements are denoted as $L_0, \ldots, L_{116}$. Figure 9 shows some example local basis. Given a convex structuring element $P$, the solution space of $n$-tuples $(a_0, \ldots, a_{116})$ satisfying decomposition conditions D.1, D.2, and D.3 is the set of all local basis such that

$$P = a_0L_0 \oplus \ldots \oplus a_{116}L_{116} . \qquad (30)$$

The solution space thus obtained represents only and all the feasible local decompositions of $P$.

### 4.2 Cost Function

Suppose $c_k$, where $k = 1, \ldots, n$, is the cost or time to compute the dilation of an image by structuring element $Q_k$. Then, the total cost or time to compute the sequence of dilation operations by $a_1Q_1, \ldots, a_nQ_n$ can be represented by the cost function

$$\sum_{k=1}^{n} a_k c_k . \qquad (31)$$

If we implement dilation operation as presented in Section 2, the amount of computation required for executing dilation operation by a structuring element is proportional to the number of foreground pixels in the structuring element. Thus, the cost function for the sequence of dilation operations by local structuring elements $a_1Q_1, \ldots, a_nQ_n$ is then

$$\sum_{k=1}^{n} a_k p_k , \qquad (32)$$

where $p_k$ is the number of foreground pixels in structuring element $Q_k$.

### 4.3 Finding the solution $n$-tuple

Given a structuring element $P$, the optimal local decomposition of $P$ determined as

$$a_0L_0 \oplus \ldots \oplus a_{116}L_{116} , \qquad (33)$$

where the $n$-tuple $(a_0, \ldots, a_{116})$ minimizes the cost function, and at the same time satisfying the decomposition conditions D.1, D.2, and D.3 in Section 3. We use linear integer programming technique [18], [19] to find the optimal solution $n$-tuple. The set of constraints used in linear integer programming is the set of linear integer equations generated from the decomposition conditions D.1, D.2, and D.3 by using the set of all local basis. The objective function to be minimized is the cost function obtained by using the set of all local basis. By giving different cost functions, optimal decompositions for different computing environments and implementation algorithms can be obtained.

### 4.4 Decomposition Examples

Table 1 shows the optimal local decompositions of example convex structuring elements in Figure 7. Figure 8 graphically depicts the local bases that appear in the local optimal decomposition in Table 1. In Table 2, we compare the cost for performing dilation by the original example convex structuring elements and the cost for performing dilation by the sequences of decomposed structuring elements shown in Table 1. The cost for performing dilation represents the number of addition operations required for executing dilation, which in turn, is the number of elements in the structuring elements involved in dilation. In case of dilation by an original structuring element, the number of addition operations is the number of the elements in the original structuring element. In case of dilation by the decomposed sequence resulted from optimal local decomposition, the number of addition operations is the number of the elements in all the bases contained in the decomposed sequence. The simulation shows more reduction in the amount of computation when using optimal local decomposition for larger structuring elements.

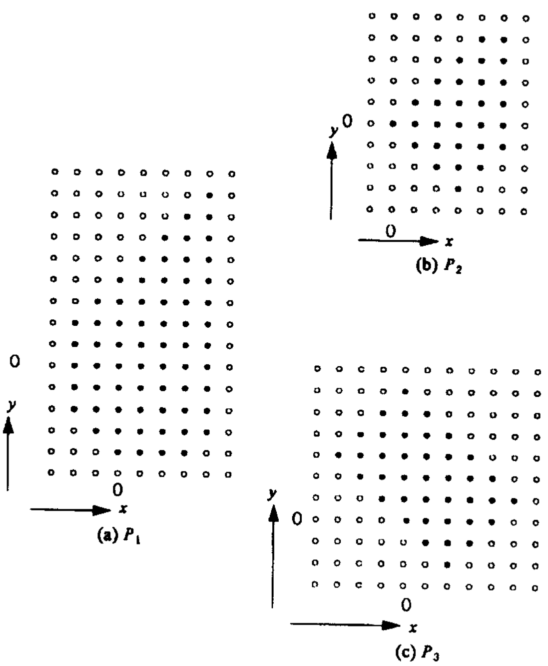


Fig. 7 Examples of convex structuring elements

## 5. Conclusion

In this paper, we formulated a general framework to find an optimal local decomposition of structuring elements. Optimal criteria for decomposition vary widely depending on how the dilation operation is implemented. We introduced the use of linear integer programming technique to solve the decomposition problem, using the length of the digital convex polygon edges and their positions as linear constraints. By choosing different cost functions as the object function for the integer programming, our method is flexible in the choice a variety of optimality criteria for different computing environments and implementation algorithms. Also, we show that the optimal local decomposition results in a great reduction in amounts of computing for dilation operations.

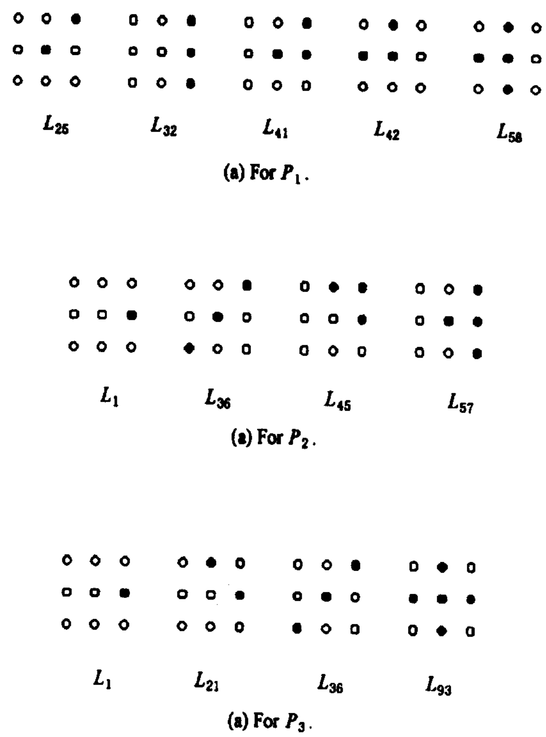


Fig. 8 Local bases that appear in the optimal local decomposition of the convex structuring elements in Figure 7.

Table 1 Decomposition results

| Example DCPG | Optimal Local Decomposition |
|---|---|
| $P_1$ | $L_{25} \oplus 2L_{32} \oplus L_{41} \oplus 2L_{42} \oplus 2L_{58}$ |
| $P_2$ | $L_1 \oplus L_{36} \oplus L_{45} \oplus 2L_{57}$ |
| $P_3$ | $2L_1 \oplus 4L_{21} \oplus L_{36} \oplus L_{93}$ |

Table 2 Comparison of the numbers of addition operations per each pixel in input image required for dilation by the original structuring element and by the decomposed sequence.

| Example DCPG | Original | Decomposed Sequence |
|---|---|---|
| $P_1$ | 66 | 25 |
| $P_2$ | 29 | 15 |
| $P_3$ | 39 | 18 |

## REFERENCES


[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] T. Kanungo and R.M. Haralick "Vector-space solution for a morphological shape-decomposition problem," *Journal of Mathematical Imaging and Vision*, vol. 2, pp. 51-82, 1992.

[11] B. Grunbaum, *Convex polytopes*, London: Interscience Publishers, 1967.

[12] G. C. Shephard, "Decomposable convex polyhedra," *Mathematika*, vol. 10, Part 2, no. 20, pp. 89-95, 1963.

[13] K. E. Batcher, "Design of a massively parallel processor," *IEEE Trans. on Computers*, vol. C-29, no. 9, pp. 836-840, 1980.

[14] D. W. Blivins, E. W. Davis, R.A. Heaton and J. H. Reif, "Blitzen: A highly integrated massively parallel machine," *Proc., The Second Symposium on the Frontiers of Massively Parallel Computation*, pp. 399-406, 1988.

[15] J.R. Nickolis, "The design of the MasPar MP-1: A cost effective massively parallel computer," *Proceedings, IEEE Compcon Spring '90*, pp. 25-28, 1990.

[16] P. E. Danielsson and S. Levialdi, "Computer architectures for pictorial information systems," *IEEE Computer*, Nov. 1981, pp. 53-65.

[17] H. Freeman, "Computer processing of line drawing images," *Computer Surveys*, vol. 6, no. 1, March 1974, pp. 57-98.

[18] T. C. Hu, *Integer Programming and Network Flows*, Reading, MA:Addison-Wesley Publishing Company, 1970.

[19] M. M. Syslo, N. Deo, and J. S. Kowalik, *Discrete Optimization Algorithms*, Englewood Cliff, NJ : Prentice-Hall, 1983.

[20] S. Y. Ohn and E. K. Wong, "Morphological decomposition of convex polytopes and its application in discrete image space," *Proceedings of ICIP-94*, vol 2, pp. 560-564, November 13-16, 1994, Austin, Texas.


## 저 자 소 개


**온 승 엽 (溫 承 燁)**

1962년 3월 20일생. 1984년 서울대학교 공대 전기공학과 졸업. 1988년 Polytechnic University 대학원 컴퓨터 및 정보과학과 졸업(석사). 1995년 동 컴퓨터 및 정보과학과 졸업(공박). 동 컴퓨터 및 정보과학과 박사후 연구과정 연구원. 1996년 한국통신 연구개발본부 선임연구원. 현재 홍익 대학교 과기대 전전컴 공학부 전임강사.

Tel : +82-415-860-2478, Fax : +82-415-865-0460

E-mail : syohn@wow.hongik.ac.kr