DECOMPOSITION OF STRUCTURING ELEMENTS FOR OPTIMAL IMPLEMENTATION OF MORPHOLOGICAL OPERATIONS

Hochong Park
Samsung Electronics Co.
8-2 Karak-Dong, Songpa-Ku
Seoul, Korea

Roland T. Chin
Computer Science Department
Hong Kong University of
Science and Technology
Clear Water Bay, Hong Kong

Abstract

The computational cost of morphological operations can be reduced by proper decomposition of structuring elements into smaller elements. In this paper, optimal decomposition of a set of commonly-used structuring elements is derived for two hardware architectures - 2-D mesh array processors and 3×3 pipeline machines, based on the decomposition algorithms reported in [1] and [2]. The resulting set of optimal decomposition provides a useful reference guide to the design and optimal implementation of morphological image filters.

1. Introduction

Morphological operations such as dilations and erosions must be implemented efficiently on existing image processing systems for practical applications. These systems, however, are designed primarily for small local operations[3], or support only local operations of fixed size due to hardware constraints[4]. These inherent system limitations require that large morphological operation be converted into a sequence of small operations, where the size of operation corresponds to the size of structuring element. A single dilation(erosion) by large structuring element can be represented equivalently by a sequence of dilations(erosions) by smaller structuring elements, provided that the original structuring element is decomposed into a set of smaller structuring elements. Hence, efficient implementation of morphological operations can be realized by the decomposition of structuring elements.

In general, the decomposition of a structuring element is not unique, which naturally leads to an optimal decomposition that minimizes the computational cost of the resulting recursive operations. In addition, the optimization criteria depend on the architecture of the system, hence different optimal decomposition is required for different system architecture.

There exist a few algorithms for the decomposition of structuring elements[5-10]. Recently, optimality issue of structuring element decomposition has been investigated by the authors[1, 2]. The following theoretical issues have been addressed: Optimization criteria for the particular architecture in terms of computational cost; convexity of the structuring element shape which imposes conditions on the decomposition; and decomposability of a given structuring element which is indicative of the existence of a solution. Algorithms for the optimal decomposition and tests for the various necessary decomposition conditions have also been proposed in these studies.

In this paper, the procedures for optimal decomposition are briefly reviewed for 2-D parallel array processors[3] and 3×3 pipeline machines[4]. Then, a set of standard, commonly-used structuring elements such as squares, rectangles, triangles, lines, hexagons, octagons, rhombuses and circles is decomposed using the given decomposition algorithms, and the resulting sets of optimal decomposition are presented. These decomposition sets provide readers with a quick reference guide for the optimal implementation of morphological filters when arbitrary large structuring elements are required for the applications on hand.

2. Optimal Decomposition of Structuring Elements

2.1. Definitions

Definition 1 : A binary image is *simply-connected* if and only if it is 8-connected and contains no holes. All images in this paper are assumed to be *simply-connected* binary images.

Definition 2 : An image A is said to be *equivalent* to an image B, denoted by $A \sim B$, if and only if A is identical to B except for a position offset, that is, A is a shifted version of B.

Definition 3 : An image A is a *factor* of an image S if and only if $S = A \oplus B$ for some image B, where $\oplus$ is a dilation operator. A factor A of S is a trivial factor if and only if A is a one-pixel image or $A \sim S$. A factor A of S is a *prime factor* if and only if every factor of A is a *trivial factor*.

Definition 4 : An image S is *decomposable* if and only if S can be represented by $S = A1 \oplus A2 \oplus ... \oplus An$ where each Ai is 3×3 or less and simply-connected.

2.2. For 2-D Mesh Array Processors

A procedure to determine optimal decomposition of structuring elements for 2-D mesh array processors is described, where the structuring elements are assumed to be *convex*. On a 2-D mesh array processor, the cost of morphological operation by a one-pixel structuring element is the number of 4-connected shifts required by the pixel in the structuring element, which is equal to the distance of pixel in city-block metric[12]. Hence, the worst-case cost of operation is the sum of distance of all the pixels in the structuring elements in city-block metric.

A dilation between two convex images can be represented by the chain code arithmetic[11]. Subsequently, it has been shown that every convex image can be decomposed into a sequence of dilations by a set of 13 prime convex factors Qi given in Figure 1. It has also been shown that in order for a given decomposition of S into Qi to be optimal, the origin of Qi must be located at a specific location as defined in Figure 1; the center of 3×3 grid indicates the origin. Note that $Q1$, $Q2$, $Q3$, and $Q4$ have more than one choice of origin. Any non-prime convex image can be further decomposed into all prime factors without increasing the cost, which implies that non-prime factors need not be considered in the determination of optimal decomposition.

The decomposition of S into prime factors can be determined recursively, that is, (i) determine a prime factor $A1$ of S, resulting in $S \sim A1 \oplus B1$; (ii) determine a prime factor $A2$ of $B1$, resulting in $S \sim A1 \oplus A2 \oplus B2$; (iii) repeat (ii) until Bi is prime and denote $An \sim Bi$, resulting in $S \sim A1 \oplus A2 \oplus ... \oplus An$; (iv) determine the origin of each Ai such that $S = A1 \oplus A2 \oplus ... \oplus An \oplus T$, where T is a one-pixel image containing a position offset. In this way, any possible decomposition into prime factors can be obtained, hence the problem of optimal decomposition is narrowed down to two sub-problems: (i) the order of prime factor selection and (ii) origin of each factor.

If the origin of Qi is assigned as in Figure 1, the cost is dependent only on T. In other words, for the decomposition of $S = A1 \oplus A2 \oplus ... \oplus An \oplus T$, where Ai is one of Qi's in Figure 1, the contribution of $A1 \oplus A2 \oplus ... \oplus An$ to the cost is always constant, and the contribution of T remains the only variable. Since T depends only on the origins of $Q1$, $Q2$, $Q3$, and $Q4$ under the constraints given in Figure 1, the origin of these four factors must be assigned such that the cost of T is minimized. A procedure to assign the origins to $Q1$, $Q2$, $Q3$, and $Q4$ is given in [1].

Finally, it has been shown in [1] that $Q1$, $Q2$, $Q3$, and $Q4$ must be selected prior to other factors in order to minimize the cost of T. In conclusion, the optimal decomposition results when the prime factors are selected recursively in the order of $Q1$, $Q2$, $Q3$, $Q4$ and the rest of factors, and the origin of each factor is assigned as in Figure 1. Such a procedure for optimal decomposition is implemented and applied to some commonly-used structuring elements, and the results are presented in Section 3.

2.3. For 3 × 3 Pipeline Machines

A procedure to determine optimal decomposition for 3×3 pipeline machines is described, where no restriction is placed on the shape of structuring elements. On a 3×3 pipeline machine, the cost of morphological operation is the number of 3×3 structuring elements required for the whole operation. It has been shown in [2] that not all structuring elements can be decomposed into 3×3 factors, hence the first step towards optimal decomposition is to check if a given S is decomposable.

The decomposability check consists of 3 tests, and is necessary only for concave images because convex images are decomposable into 3×3 factors in Figure 1. The first test checks the boundary of S. If S contains a concave boundary that can not be fitted in a 3×3 region, then S is not decomposable. The second test is to search for the existence of necessary factors of S. For each concave boundary of S, there must exist a 3×3 factor that contains the same concave boundary. The theory behind these two tests is that a new concave boundary cannot be created by dilation of 3×3 simply-connected structuring elements; the concave boundary must be copied from the structuring elements.

Finally in the third test, linear equations are solved for integer solutions, where the equations are constructed based on S and its all 3×3 prime concave factors determined above. If a solution does not exist, S is not decomposable. Otherwise, the solution defines a sequence of dilations $A1 \oplus A2 \oplus ... \oplus An \oplus B$, where Ai is one of the 3×3 prime concave factors and B is a convex factor of arbitrary size. If this sequence of dilations produces a simply-connected image, then S is decomposable.

For a given decomposable S, its optimal decomposition is sought for by further decomposing B into prime convex factors and combining all resulting small structuring elements into 3×3 images so that the total number of 3×3 images is minimized. Each solution defines one decomposition which has its own optimized version. To find the global optimal decomposition among all possible solutions, it is necessary to find all solutions of the linear equation and to optimize each. This procedure is implemented and applied to a set of commonly-used structuring elements. The results are shown in Section 3.

3. Optimal Decomposition of Commonly-Used Structuring Elements

Figures 2 – 5 show the results of optimal decomposition for symmetric polygons, rotated rectangles, triangles, and circles, respectively. The optimality of each decomposition is guaranteed from the proofs in [1] and [2]. In these Figures, an integer n in front of the factor denotes n-fold recursive dilations; when n = 0, the n-fold dilation becomes an empty set. The center of 3×3 grid denotes the origin.

In Figures 2 – 4, (a) shows the given structuring element, where □ indicates the origin, (b) shows the optimal decomposition for 2-D mesh processors, and (c) shows the optimal decomposition for 3×3 pipeline machines. Note that in Figure 2, it is assumed that not both a and b are zero; the case of $a = b = 0$ is considered in Figure 3.

Many distinctive shapes are obtained based on the values of a, b, and c in Figure 2; a horizontal or vertical line when $c = 0$ and a or b is zero; a rectangle when $c = 0$ and none of a and b is zero; a hexagon when $c \neq 0$ and a or b is zero; and an octagon when none of a, b, c is zero. In Figure 3, when a or b is zero, it is a diagonal line; and when $a = b$, it is a rhombus.

In this paper, the digital circle of radius i, Ci, is constructed by the algorithm in [13]. For example, $C4$ is shown in Figure 5(a). Some of these digital circles contain concave boundaries. Since the algorithm for optimal decomposition of concave structuring elements for 2-D mesh processors is not yet known(optimal decomposition in [1] is for convex structuring elements only), optimal decomposition of these concave circles cannot be determined for 2-D mesh processors. However, digital circles can be approximated by octagons and in such cases, optimal decomposition is available and given in Figure 2(b).

The procedure for optimal decomposition of concave circles for 3×3 pipeline machines is known[2]. All concave circles up to radius 10 pass the decomposition tests, and the optimal decomposition for 3×3 pipeline machines is shown in Figure 5(b).

4. Conclusions

In this paper, two procedures for optimal decomposition of structuring elements for 2-D mesh processors and 3×3 pipeline machines were implemented and applied to a set of commonly-used structuring elements. The algorithms are based on the theoretical studies in [1] and [2]. The decomposed results provide a quick reference guide for the design and optimal implementation of morphological image filters.

References

[1] H. Park and R. T. Chin, "Optimal decomposition of convex morphological structuring elements for 4-connected parallel array processors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, March, 1994.

[2] H. Park and R. T. Chin, "Decomposition of arbitrarily shaped morphological structuring elements," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, January 1995.

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

[4] R. M. Lougheed, D. L. McCubbrey and S. R. Sternberg, "Cytocomputers: Architectures for parallel image processing," *Proc. IEEE Workshop on Picture Data Description and Management*, 1980.

[5] J. Pecht, "Speeding-up successive Minkowski operations with bit-plane computers," *Pattern Recognition Letters*, vol.3, March 1985.

[6] R. Van den Boomgaard and R. Van Balen, "Methods for fast morphological image transforms using bitmapped binary images," *Computer Vision, Graphics and Image Processing - Graphical Models and Image Processing*, vol.54, no.3, May 1992.

[7] X. Zhuang and R. M. Haralick, "Morphological structuring elements decomposition," *Computer Vision, Graphics and Image Processing*, vol.35, Sept. 1986.

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

[9] 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.

[10] D. Li and G. X. Ritter, "Decomposition of separable and symmetric convex templates," *Image Algebra and Morphological Image Processing, Proc. SPIE*, vol. 1350, 1990.

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

[12] D. H. Ballard and C. M. Brown, *Computer Vision*, New Jersey : Prentice-Hall, 1982.

[13] J. E. Bresenham, "A linear algorithm for incremental digital display of circular arcs," *Communications of the ACM*, vol. 20, no. 2, February 1977.

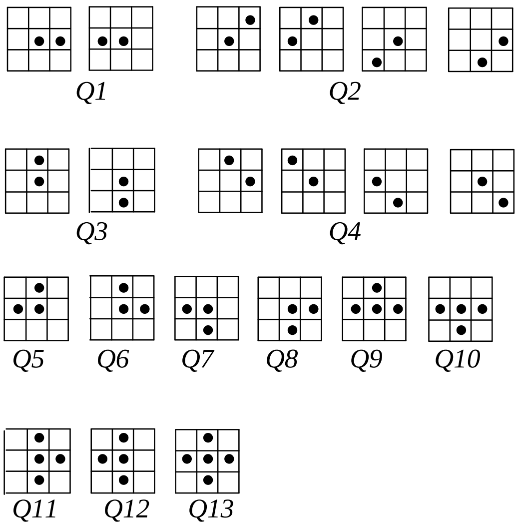


Figure 1. Thirteen prime convex factors with an origin indicated by the center of the 3×3 grid.

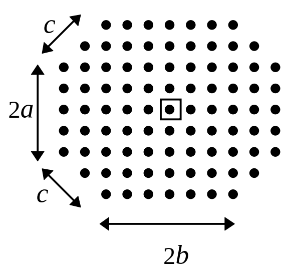


Figure 2(a)

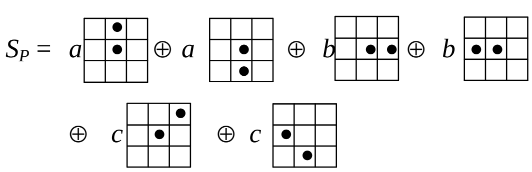


Figure 2(b)

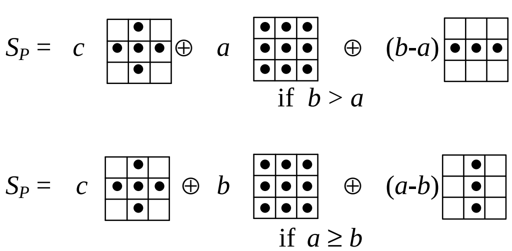


Figure 2(c)

Figure 2. Decomposition of symmetric polygons, S_P. (a) Polygon with $a = 2$, $b = 3$ and $c = 2$. It is assumed that not both a and b are zero. (b) Optimal decomposition for 2-D mesh array processors. (c) Optimal decomposition for 3×3 pipeline machines.

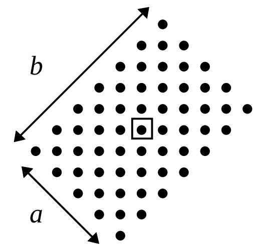


Figure 3(a)

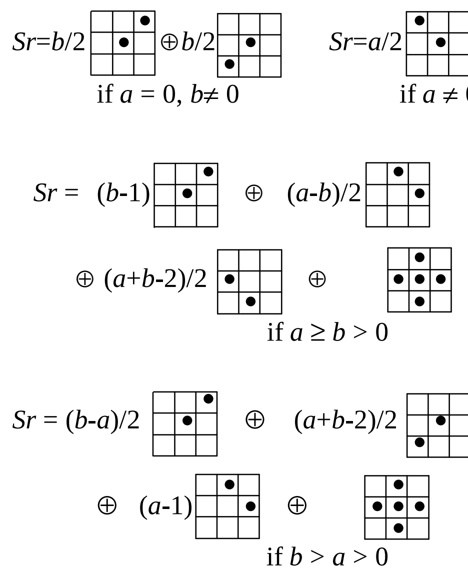


Figure 3(b)

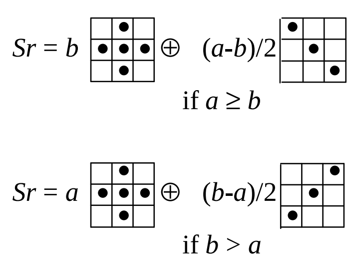


Figure 3(c)

Figure 3. Decomposition of rotated rectangles, *Sr*. (a) Rotated rectangle with $a = 4$ and $b = 6$. (b) Optimal decomposition for 2-D mesh array processors. (c) Optimal decomposition for 3×3 pipeline machines

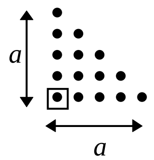


Figure 4(a.1)

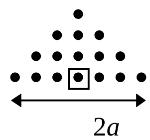


Figure 4(a.2)

Figure 4(b)

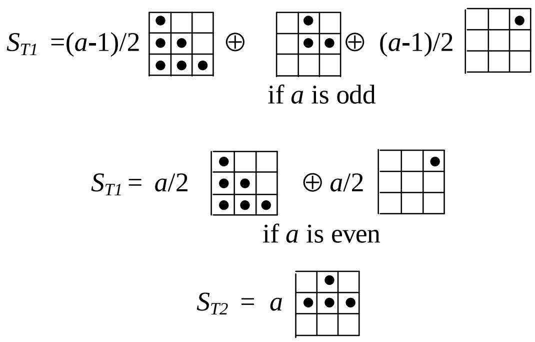


Figure 4(c)

Figure 4. Decomposition of triangles, S_{T1} and S_{T2}. (a.1) Triangle S_{T1} with a = 4. (a.2) Triangle S_{T1} with a = 3. (b) Optimal decomposition for 2-D mesh array processors. (c) Optimal decomposition for 3×3 pipeline machines.

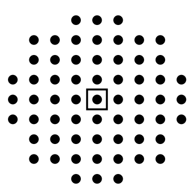
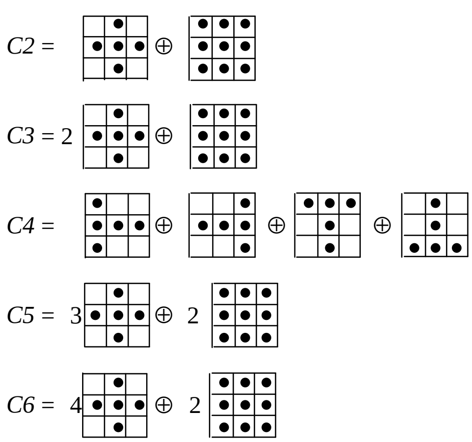


Figure 5(a)



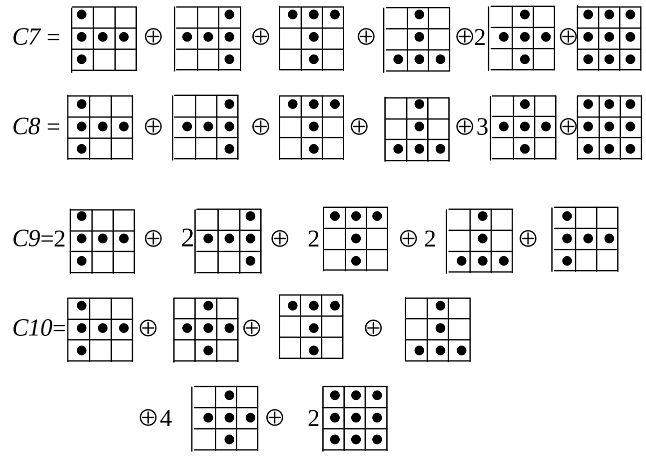


Figure 5(b)

Figure 5. Decomposition of digital circles, Ci. (a) Digital circle of radius 4, $C4$. (b) Optimal decomposition for 3×3 pipeline machines.