# Fast Algorithms for Binary Dilation and Erosion Using Run-Length Encoding

Wook-Joong Kim, Seong-Dae Kim, and Kyuheon Kim

*ABSTRACT—Fast binary dilation and erosion algorithms using run-length encoding (RLE) are proposed. RLE is an alternative way of representing a binary image using a run, which is a sequence of '1' pixels. First, we derive the run-based representation of dilation and erosion and then present the full steps of the proposed algorithms in detail.*

*Keywords—Binary image processing, morphological operator, run-length encoding.*

## I. Introduction

Morphological operators are some of the most fundamental and popular nonlinear tools for various binary image processing. Morphological operators are constructed by combining two basic operators: dilation and erosion. Assuming that $Z$ is a region in binary image $I$, the dilation and erosion of $Z$ by structuring element $S$ are defined as

$$\text{Dilation:}\quad Z \oplus S = \left\{ p \middle| \hat{S}_p \cap Z \neq 0 \right\} \tag{1}$$

$$\text{Erosion:}\quad Z \Theta S = \left\{ p \middle| S_p \subseteq Z \right\}, \tag{2}$$

where $\hat{S}$ is the reflection about the origin and $S_p$ is the translation by position $p$. All other morphological operators are defined by concatenating the two operators. For example, the opening operator is defined by dilation after erosion, and the closing operator is defined by erosion after dilation.

In this paper, fast binary dilation and erosion algorithms using run-length encoding (RLE) [1] are proposed. A run is a sequence of '1' (or '0') pixels horizontally (or vertically), and RLE is a representation of a binary image using a combination of runs. Handling binary images in runs, rather than in pixels, can bring a considerable reduction in computation because the number of runs is usually much smaller than the number of pixels. Previously, several ideas for run-based morphological operations were presented in [2]-[4]. Piper and Tang [2] introduced the concept of using RLE for binary morphological operations, and Cardoner and Thomas [3], [4] applied the concept for skeletoning and line drawing. However, a detailed theoretical background and an effective algorithm with full numerical analysis have yet to be presented.

In section II, we provide a mathematical investigation on dilation and erosion in terms of runs, and in section III, fast algorithms for dilation and erosion are proposed based on the investigation. The proposed algorithms have inputs and outputs as run-length encoded forms. In section IV, simulation results are presented, and we conclude in section V.

## II. Dilation and Erosion Using RLE

For the sake of simplicity, a run is considered as a sequence of '1' pixels in the horizontal direction throughout this paper. Assume that binary image $I$ has region $Z$ of '1' pixels, and the RLE representation of $Z$ is defined as $Z = \bigcup_{n=1}^{N} R_n$, where $N$ is the number of runs, and $R_n = \langle xL_n, xR_n, y_n \rangle$ is the $n$-th run of $Z$. The $xL_n$, $xR_n$, and $y_n$ of $R_n$ respectively indicate the $x$-coordinate of the left-most (starting) pixel, the $x$-coordinate of the right-most (ending) pixel, and the $y$-coordinate of $R_n$.

Regarding RLE, we define a '*compact representation of RLE*' when concatenated '1' pixels are defined as a single run, not divided into several runs with or without overlapping.

Manuscript received Feb. 16, 2005; revised Aug. 26, 2005.

Wook-Joong Kim (phone: +82 42 860 6123, email: wjk@etri.re.kr) and Kyuheon Kim (email: kkim@etri.re.kr) are with Digital Broadcasting Research Division, ETRI, Daejeon, Korea.

Seong-Dae Kim (email: sdkim@sdvision.kaist.ac.kr) is with the Department of Electrical Engineering, KAIST, Daejeon, Korea.

Unless a compact run-length encoding is made, unnecessary computation is required due to redundancy among the runs.

### 1. Dilation Using RLE

Here, a run-based expression of dilation is presented. Let $Z = \bigcup_{n=1}^{N} R_n^Z$ and $S = \bigcup_{m=1}^{M} R_m^S$ be the compact RLE representation of $Z$ and $S$, respectively. The intersection of $\hat{S}_p$ and $Z$ is expressed as

$$\begin{aligned}\hat{S}_p \cap Z &= \hat{S}_p \cap \left[\bigcup_{n=1}^{N} R_n^Z\right] = \bigcup_{n=1}^{N}\left[\hat{S}_p \cap R_n^Z\right] \\ &= \bigcup_{n=1}^{N}\left[\bigcup_{m=1}^{M}\left(\hat{R}_m^S\right)_p \cap R_n^Z\right] = \bigcup_{n=1}^{N}\bigcup_{m=1}^{M}\left[\left(\hat{R}_m^S\right)_p \cap R_n^Z\right].\end{aligned} \tag{3}$$

Applying (3) into the dilation definition (1), we have

$$\begin{aligned} Z \oplus S &= \left\{p \middle| \hat{S}_p \cap Z \neq 0\right\} = \left\{p \middle| \bigcup_{n=1}^{N}\bigcup_{m=1}^{M}\left[\left(\hat{R}_m^S\right)_p \cap R_n^Z\right] \neq 0\right\} \\ &= \bigcup_{n=1}^{N}\bigcup_{m=1}^{M}\left\{p \middle| \left[\left(\hat{R}_m^S\right)_p \cap R_n^Z\right] \neq 0\right\} \\ &= \bigcup_{n=1}^{N}\bigcup_{m=1}^{M}\left(R_n^Z \oplus R_m^S\right).\end{aligned} \tag{4}$$

Therefore, the dilation of $Z$ by $S$ is the same as the union of the dilation of each run of $Z$ by each run of $S$.

Regarding dilation $R_n^Z \oplus R_m^S$, it is the same as finding all possible positions where $R_n^Z$ and $(\hat{R}_m^S)_p$ overlap. Two runs can have an overlapping relation when they are located at the same $y$ position and share more than one pixel. For a given $R_m^S = \left\langle xL_m^S, xR_m^S : y_m^S\right\rangle$ and $R_n^Z = \left\langle xL_n^Z, xR_n^Z : y_n^Z\right\rangle$, the translated reflection of $R_m^S$ can be represented as $(\hat{R}_m^S)_p = \left\langle -xR_m^S + x_d, -xL_m^S + x_d : -y_m^S + y_d\right\rangle$, where $p = (x_d, y_d)$. Hence, the two runs $R_n^Z$ and $(\hat{R}_m^S)_p$ overlap when

$$\begin{matrix} -y_m^S + y_d = y_n^Z \\ xL_n^Z \le -xL_m^S + x_d \le xR_n^Z \end{matrix} \quad \text{or} \quad \begin{matrix} -y_m^S + y_d = y_n^Z \\ xL_n^Z \le -xR_m^S + x_d \le xR_n^Z. \end{matrix} \tag{5}$$

Modifying (5), we obtain

$$\begin{aligned} y_d &= y_n^Z + y_m^S, \\ xL_n^Z + xL_m^S &\le x_d \le xR_n^Z + xR_m^S. \end{aligned}$$

Therefore, the dilation of $Z$ by $S$ is expressed as

$$Z \oplus S = \bigcup_{n=1}^{N}\bigcup_{m=1}^{M}\left\langle xL_n^Z + xL_m^S, xR_n^Z + xR_m^S : y_n^Z + y_m^S\right\rangle. \tag{6}$$

### 2. Erosion Using RLE

Now, a run-based representation of erosion is presented. From the definition of erosion (2), we have

$$\begin{aligned} Z \Theta S &= \left\{p \middle| S_p \subseteq Z\right\} = \left\{p \middle| \bigcup_{m=1}^{M}\left(R_m^S\right)_p \subseteq Z\right\} \\ &= \left\{p \middle| \left[\left(R_1^S\right)_p \cup \cdots \cup \left(R_M^S\right)_p\right] \subseteq Z\right\} \\ &= \left\{p \middle| \left(R_1^S\right)_p \subseteq Z\right\} \cap \cdots \cap \left\{p \middle| \left(R_M^S\right)_p \subseteq Z\right\} \\ &= \bigcap_{m=1}^{M}\left[Z \Theta R_m^S\right]. \end{aligned} \tag{7}$$

Meanwhile, the erosion of $Z$ by a single run $R$ is defined as

$$\begin{aligned} Z \Theta R &= \left\{p \middle| R_p \subseteq Z\right\} = \left\{p \middle| R_p \subseteq \left(\bigcup_{n=1}^{N} R_n^Z\right)\right\} \\ &= \bigcup_{n=1}^{N}\left\{p \middle| R_p \subseteq R_n^Z\right\} = \bigcup_{n=1}^{N}\left[R_n^Z \Theta R\right]. \end{aligned} \tag{8}$$

Combining (7) and (8), erosion of $Z$ by $S$ is expressed as

$$Z \Theta S = \bigcap_{m=1}^{M}\left[\bigcup_{n=1}^{N}\left(R_n^Z \Theta R_m^S\right)\right]. \tag{9}$$

Next, let's consider the erosion of single runs: $R_n^Z \Theta R_m^S$. From (2), the erosion operation is identifying all possible positions $p$ where $\left(R_m^S\right)_p$ is included in $R_n^Z$. Run $R_1$ includes $R_2$ in the following situations: (i) when the length of $R_1$ is equal to or longer than $R_2$; (ii) when they are at the same $y$ position; and (iii) when the left- and right-most pixels of $R_2$ are within $R_1$. Hence, $R_n^Z$ includes $\left(R_n^S\right)_p$ when the following are true.

$$\begin{aligned} &\text{From (i)} \Rightarrow && \left(xR_m^S - xL_m^S\right) \le \left(xR_n^Z - xL_n^Z\right), \\ &\text{From (ii)} \Rightarrow && y_m^S + y_d = y_n^Z, \\ &\text{From (iii)} \Rightarrow && xL_m^S + x_d \ge xL_n^Z, \\ & && xR_m^S + x_d \le xR_n^Z. \end{aligned} \tag{10}$$

Therefore, modifying (10) and applying it into (9), we finally obtain

$$Z \Theta S = \bigcap_{m=1}^{M}\left[\bigcup_{n=1}^{N}\left\{\begin{array}{ll}\left\langle xL_n^Z - xL_m^S, xR_n^Z - xR_m^S : y_n^Z - y_m^S\right\rangle, & \\ \quad \text{if } \left(xR_m^S - xL_m^S\right) \le \left(xR_n^Z - xL_n^Z\right) & \\ \phi, \quad \text{otherwise} & \end{array}\right\}\right]. \tag{11}$$

## III. Proposed Algorithms

In this section, the proposed dilation and erosion algorithms are presented. Before presenting the algorithms, we define *primary runs* as the runs generated from (6) or (11). The outcomes from (6) and (11) are the results for dilation or erosion. However, it is not guaranteed that primary runs are compactly represented. To have further operation with the primary runs, an additional process for removing redundancy, that is, a compacting process, among primary runs is required to eliminate unnecessary computations.

The proposed dilation algorithm is as follows:

Input: compactly run-length encoded $Z$ and $S$;
Output: compact representation of $Z \oplus S$;
Procedure:
Step 1. Using runs of $Z$ and $S$, find $N \times M$ primary runs by (6).
Step 2. Sort the primary runs by the value of a $y$-coordinate, and identify the runs that have the same $y$-coordinate.
Step 3. If more than two runs exist at a certain $y$-coordinate, go to step 4. Otherwise, terminate the process.
Step 4. Repeat the following two operations until no overlapping or inclusion of primary runs exists:
① If any two runs overlap as similarly to (5), combine them into one.
② If any a primary run is included in another primary run as similarly to (10), delete the included run.

The flow of the dilation algorithm is straightforward: Obtain primary runs using (6) and then eliminate redundancy among them.

Next, the proposed erosion algorithm is presented. The erosion algorithm needs more consideration than the dilation algorithm due to the finding of intersections of primary runs.

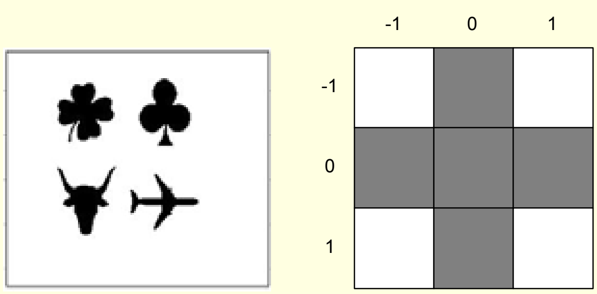
Input: compactly run-length encoded $Z$ and $S$;
Output: compact representation of $Z \Theta S$;
Procedure:
Step 1. Using runs of $Z$ and $S$, find a primary run using (11). For each primary run, store an index that indicates the $m$ of $S = \bigcup_{m=1}^{M} R_m^S$. The index is additional information that helps to identify which run of $S$ is used for obtaining a primary run.
Step 2. Sort the primary runs by the values of a $y$-coordinate, and identify the runs that have the same $y$-coordinate.
Step 3. Eliminate all primary runs at a certain $y$-coordinate, if:
① The number of primary runs at a certain $y$-coordinate is less than $M$, or
② The number is more than or equal to $M$, but a certain index value $m$ among $[1,\ldots, M]$ is missing.
Step 4. Find the common regions of all the runs that survived from step 3.

In the case of erosion, primary runs are obtained in step 1, and the intersection of primary runs is calculated in step 4 after eliminating unnecessary primary runs in step 3.

Let's consider the computational complexity of the proposed algorithms. Assuming $z$ and $s$ are the number of pixels in $Z$ and $S$, respectively, the computational complexity of pixel-unit dilation and erosion are proportional to $O(z \times s)$ [1]. Compared to this, the proposed dilation and erosion algorithms require $O(N \times M + sorting(N \times M))$ operations. Generally, the computational complexity for fast sorting of $N$ numbers is proportional to $O(sorting(N)) = N\log(N)$. Hence, the computational complexity of the proposed algorithms can be modified as $O((N \times M)\cdot(1+\log(N \times M)))$. Because $z \times s$ is always bigger than $N \times M$ without losing generality, we can conclude that $O(z \times s) > O((N \times M)\cdot(1+\log(N \times M)))$ is always satisfied.

## IV. Simulations

In this section, we show how much computational reduction can be obtained from the proposed algorithms. To show their performance, we compared the execution times between the proposed algorithms and their pixel-unit processing under the same machine. This comparison may not be able to show quantitative evidence, but at least, it can provide the strengths of the proposed algorithms from a qualitative perspective. A pixel-unit processing is implemented as two dimensional filtering, that is, a structuring element is regarded as a filter window and is scanned throughout all possible pixels. Figure 1 shows the test image and structuring element used for our simulations. The image is composed of 380 runs and its size is $256 \times 256$ pixels.




(a) 'Symbol' image (b) Structuring element

Fig. 1. Test image and structuring element.

Figure 2 shows a comparison of computation times for dilation, erosion, opening, and closing. The vertical axis represents the elapsed time after 1000 repetitions, and the ratio of the execution time is shown at the bottom. Pixel-unit erosion takes less time than pixel-unit dilation because scanning is restricted to the input regions.

The results show that the proposed algorithms require only less than 9% of computation time for pixel-unit processing.

1) $O(\bullet) =$ Order of $(\bullet)$

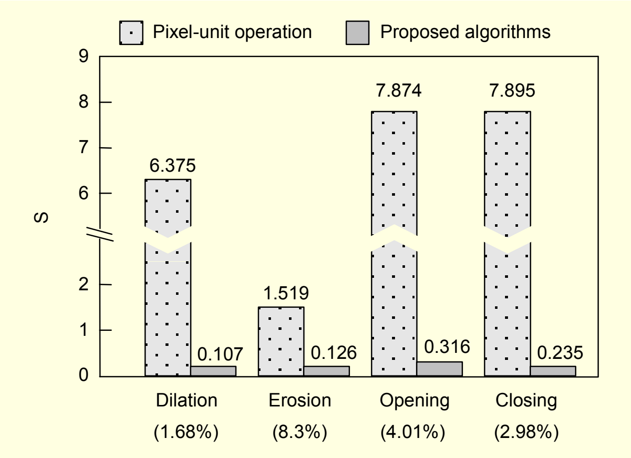



Fig. 2. The computation times for 1000 repetitions of each operation using a PC with a 1.59 GHz CPU and 1 Gbyte RAM.

## V. Conclusions

In this paper, fast RLE-based dilation and erosion algorithms are presented. The proposed algorithms are developed based on the run-based expressions of dilation and erosion as described in section II. In addition to the fast computing performance of the proposed algorithms, we believe that the run-based investigation on morphological operators can also be used in various image processing applications [5] and as instructive information for the further development of relevant technologies.

## References

[1] S. Kim, J. Lee, and J. Kim, "A New Chain-Coding Algorithm for Binary Images Using Run-Length Codes," *Computer Vision, Graphics, and Image Proc.*, vol. 41, no. 1, Jan. 1988, pp. 114 - 128.

[2] L. Piper and J.Y. Tang, "Erosion and Dilation of Binary Images by Arbitrary Structuring Elements Using Interval Coding," *Pattern Recognition Letter*, vol. 9, no. 3, Apr. 1989, pp. 201-209.

[3] R. Cardoner and F. Thomas, "Residuals + Directional Gaps = Skeletons," *Pattern Recognition Letters*, vol. 18, no. 4, May 1997, pp. 343-353.

[4] R. Cardoner and F. Thomas, "Efficient Morphological Set Transformations on Line Drawings," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 11, no. 6, Sep. 1997, pp. 947-960.

[5] Jong-Hun Lee, Min-Ho Park, and Yong-Il Kim, "An Application of Canonical Correlation Analysis Technique to Land Cover Classification of LANDSAT Images," *ETRI J.*, vol.21, no.4, Dec. 1999, pp.41-51.