Morphological contour decomposition and reconstruction by using asymmetric SEs

Minhao Lu\textsuperscript{1} and Tomoyuki Idogawa\textsuperscript{2}
\textsuperscript{1} Taiyo System Technology, Co., Ltd., Japan
\textsuperscript{2} Department of Mathematical Sciences, Shibaura Institute of Technology, Japan
E-mail: idogawa@shibaura-it.ac.jp

Abstract. We had proposed an algorithm based on mathematical morphology to decompose and reconstruct an image by using two contours \cite{4}. These contours are defined as the edge and the second edge (i.e., the edge of the image which has been removed an edge once) of the original image. Since these contours have informations of boundary and internal direction of image, the original image can be reconstructed exactly from them. However, there was a restriction in this algorithm that the structuring element (SE) used for making contours must be symmetric. In this paper, we will show an expansion of the algorithm to be able to use asymmetric SEs under a certain condition and also show some applications of this method.

1. Mathematical Morphology

Mathematical morphology is a theory for the analysis of spatial structures based on set theory, integral geometry and lattice algebra \cite{7}. In mathematical morphology, each binary image is considered as a set of points (pixels) in an affine space ($\mathbb{R}^2$ for continuous or $\mathbb{Z}^2$ for discrete image) those which construct shapes (foreground) of the image, and each image processing is done by using some set operations, e.g., union ($\cup$), intersection ($\cap$) and set difference ($\setminus$), and so on.

The most fundamental operations in mathematical morphology are the Minkowski set addition $\oplus$ and subtraction $\ominus$:

$$A \oplus B := \{a + b \mid a \in A, b \in B\} = \bigcup_{b \in B} A_b,$$

$$A \ominus B := \{x \mid \forall b \in B \ [x - b \in A]\} = \bigcap_{b \in B} A_b,$$

where $A_b := \{a + b \mid a \in A\}$ is the translate of the set $A$ by the vector $b$. Usually we consider these operations as transformations of the original image $A$ by using another (relatively small) object $B$, hence we call $B$ a structuring element (SE).

Morphological dilation $\delta_B$ and erosion $\varepsilon_B$ are also important in this theory, which are defined as:

$$\delta_B(A) := A \ominus B^s = \{z \mid B_z \cap A \neq \emptyset\}, \quad \varepsilon_B(A) := A \oplus B^s = \{z \mid B_z \subseteq A\},$$

where $B^s := \{-b \mid b \in B\}$ is the symmetric set of $B$ with respect to the origin. By definition, it is clear that they are almost same as Minkowski set addition and subtraction, respectively. The dilation $\delta_B$ expands and the erosion $\varepsilon_B$ shrinks an image, in general. Moreover, morphological closing $\phi_B$ and opening $\gamma_B$ are useful in this theory:

$$\phi_B(A) := (\varepsilon_{B^s} \circ \delta_B)(A) = (A \ominus B^s) \ominus B,$$

$$\gamma_B(A) := (\delta_{B^s} \circ \varepsilon_B)(A) = (A \oplus B^s) \ominus B.$$
These operations have some properties (see [7]).

2. Previous works

In 1986, Maragos and Schafer proposed a coding method of binary image by using morphological skeleton [5]. The skeleton of image is a kind of medial axis of shape, and there are various definitions of it. One of them is to define it as the locus of centers of the “maximal disks” inside shapes on image. Morphological skeleton was defined along this idea. When $A$ is a finite digital (discrete) image, the morphological skeleton $SK(A)$ of $A$ is defined as

$$SK(A) := \bigcup_{n=0}^{N} SK_n(A),$$

where $SK_n(A)$ describes $n$th skeleton subset of $A$:

$$SK_n(A) := \varepsilon^n_B(A) - \gamma_B(\varepsilon^n_B(A)),$$

here and hereafter, we use the notation such as $\varepsilon^n_B = \varepsilon_B \circ \varepsilon^{n-1}_B$ with $\varepsilon^0_B = id$. The number $N$ which appears in (1) means the “size” of $A$ with respect to $B$:

$$N := \max (\{ n \in \mathbb{N} \mid \varepsilon^n_B(A) \neq \emptyset \} \cup \{0\}).$$

Maragos and Shafer showed that the original image $A$ can be reconstructed from those skeleton subsets such as

$$A = \bigcup_{n=0}^{N} \delta^2_{nB}(SK_n(A)) = \bigcup_{n=0}^{N} \bigcup_{z \in SK_n(A)} (nB)_z,$$

where $nB$ denotes the Minkowski set addition of $n$ Bs ($nB = B \oplus (n-1)B$).

In 1991, Pai and Hansen proposed a new coding method which uses the skeleton and the contour (edge) of image [6]. Morphological contour $\eta(A)$ with respect to SE $B$ is defined as the difference of the eroded image from the original one:

$$\eta(A) := A \setminus \varepsilon_B(A).$$

Maragos’ method needs the information $n = n(z)$ for each $z \in SK(A)$ such that $z \in SK_n(A)$, hence it needs the total subsets $\{SK_n(A)\}_{n=0}^{N}$, essentially. While Pai’s method calculate this $n = n(z)$ by using the contour $\eta(A)$. To be concrete, it increases $n$ until $(nB)_z$ touches $\eta(A)$ two points at least and returns the maximum number $n_{\text{max}}$. Hence, Pai’s method needs only $SK(A)$ and $\eta(A)$ to reconstruct the original image (see Fig. 1).

**Figure 1.** Pai’s method (using morphological skeleton and contour).

**Figure 2.** Proposed method (using two morphological contours).
In 2009, we proposed an algorithm based on mathematical morphology to decompose and reconstruct a binary image [4]. In this algorithm, we use the two contours: the first one is the edge of the original image and the second one is the edge of the image which has been removed the first edge from the original one. Since these contours have informations of boundary and internal direction of image, the original image can be reconstructed exactly from them.

More precisely, we introduced the second and subsequent contours \( \eta_n(A) \) as

\[
\eta_n(A) := \varepsilon^n_B(A) \setminus \varepsilon^{n+1}_B(A) \quad (n = 0, 1, 2, \ldots).
\]

Then we showed the decomposability (D) and reconstructability (R) of the original image \( A \):

\[
(D) \quad A = \bigcup_{n=0}^{N} \eta_n(A), \quad \eta_n(A) \cap \eta_j(A) = \emptyset \quad (i \neq j),
\]

\[
(R) \quad \eta_n(A) = \delta_B(\eta_{n-1}(A)) \setminus \bigcup_{k=0}^{n-1} \eta_k(A) \quad (n \geq 2)
\]

when the digital binary image \( A \) is finite (\(|A| < \infty\)) and the SE \( B \) satisfies that \( \{0\} \subseteq B \) and \( B = B^* \) (i.e., \( B \) is symmetric). The property (R) shows that the contours \( \eta_n(A) \) (\( n = 2, 3, 4, \ldots \)) can be composed from \( \eta_0(A) \) and \( \eta_1(A) \), and then the original image \( A \) can be reconstructed by taking union of them (see Fig. 2).

3. Expanded algorithm

In the method stated in [4], we had to use a symmetric SE \( B \) for contour decomposition and reconstruction. But now, we revised the method in which we could use a sequence of asymmetric SEs \( \{B_n\} \) (under a certain condition) to do similar operations.

To state the revised method, we mention about expanded definition of “contours” at first. Given a sequence of SEs \( \{B_n\} \), we define \( n \)th contour \( \eta_n(A) \) with respect to the sequence of SEs \( \{B_n\} \) as follows.

**Definition 3.1** \( n \)th contour \( \eta_n(A) \) w.r.t. SEs \( \{B_n\} \); see Fig. 3 For a given sequence \( \{B_n\} \) of SEs satisfying \( 0 \in B_n \) (\( \forall n \geq 0 \)) and a binary image \( A \), let introduce \( \{A_n\} \) as

\[
A_0 := A, \quad A_{n+1} := \varepsilon_{B_n}(A_n) = \varepsilon_{\oplus_{k=0}^{n} B_k}(A) \quad (n = 0, 1, 2, \ldots),
\]

where \( \oplus_{k=0}^{n} B_k = B_0 \oplus B_1 \oplus \cdots \oplus B_n \) (note that \( \varepsilon_{B \oplus B'} = \varepsilon_B \circ \varepsilon_B' \)). Then we define

\[
\eta_n(A) := A_n \setminus A_{n+1} = A_n \setminus \varepsilon_{B_n}(A_n) \quad (n = 0, 1, 2, \ldots).
\]

If \( B_n = \{0\} \), then \( \varepsilon_{B_n} \) is the identity map and the corresponding contour \( \eta_n(A) = \emptyset \), hence such \( B_n \) is meaningless for contour decomposition (“SE” \( \{0\} \) is practically meaningless for most morphological operations).

When \( A \) is a finite image, it is easy to show that \( A \) can be decomposed to a finite number of contours \( \{\eta_n(A)\} \) with respect to \( \{B_n\} \):

**Lemma 3.2** Let \( \{B_n\} \) be a sequence of SEs satisfying \( \{0\} \subseteq B_n \) (\( \forall n \geq 0 \)). Then, for any finite image \( A \), there exists non-negative integer \( N \) (depending on \( A \)) such that \( A_N = \varepsilon_{\oplus_{k=0}^{N} B_k}(A) = \emptyset \) and hence \( \forall n > N, \eta_n(A) = \emptyset \).  

(Proof) It is clear because of the monotonicity of \( \varepsilon_{B_n} \ (\varepsilon_B(A) \subseteq A \text{ if } \{0\} \subseteq B \text{ and } 0 < |A| < \infty) \) and the finiteness of \( A \). □
\[ A_n = A_0 = A_1 = A_2 = \cdots = A_{n-2} = A_{n-1} = A_n = A_{n+1} = \cdots \]

\[ \eta_n \eta_{n-1} \eta_{n-2} \eta_1 \eta_0 \]

\[ \epsilon_{B_0} \epsilon_{B_1} \epsilon_{B_{n-2}} \epsilon_{B_{n-1}} \epsilon_{B_n} \]

Figure 3. Structure of \( \{A_n\} \) and \( \{\eta_n(A)\} \): \( A_{n+1} = \epsilon_{B_n}(A_n) \) and \( \eta_n(A) = A_n \setminus A_{n+1} \) (\( n \geq 0 \)).

**Theorem 3.3 (decomposability)** Let \( \{B_n\} \) be a sequence of SEs with \( \{0\} \subset B_n \) (\( \forall n \geq 0 \)). Then, for any finite image \( A \), it holds that

\[ A = \bigcup_{n=0}^{N} \eta_n(A), \quad \eta_i(A) \cap \eta_j(A) = \emptyset \quad (i \neq j), \]

where \( N = N(A) \) is the number mentioned in Lemma 3.2.

*(Proof)* By the definition of \( \{\eta_n(A)\} \) and inclusion relations

\[ A = A_0 \supseteq A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{n-1} \supseteq A_n \supseteq \cdots , \]

it is trivial that \( \eta_n(A) \)'s are disjoint each other. Again noticing the inclusion relation, one can easily show that

\[ \bigcup_{k=0}^{n} \eta_k(A) = A \setminus A_{n+1} = A \cap (\epsilon_{\oplus_{k=0}^{n-2}B_k}(A))^{c} \quad (n = 0, 1, 2, \ldots). \]

Hence, we get that \( \bigcup_{n=0}^{N} \eta_n(A) = A \) by using Lemma 3.2.

Moreover, if \( \{B_n\} \) satisfies a certain condition (stated below as (2); \( B_n \) may be asymmetric), then \( \eta_2(A), \eta_3(A), \ldots \) can be reconstructed from the first two contours \( \eta_0(A) \) and \( \eta_1(A) \).

**Theorem 3.4 (reconstructability)** Let \( \{B_n\} \) be a sequence of SEs with \( \{0\} \subset B_n \) (\( \forall n \geq 0 \)) and

\[ B_n \subseteq B_{n-1} \cap (\oplus_{k=0}^{n-2}B_k)^{c} \quad (\forall n \geq 2). \quad (2) \]

Then the contours \( \eta_n(A) \) (\( n = 2, 3, 4, \ldots \)) can be reconstructed inductively from the first and the second contours \( \eta_0(A) \) and \( \eta_1(A) \) by using the recursive formula:

\[ \eta_n(A) = \delta_{B_n}(\eta_{n-1}(A)) \setminus \bigcup_{k=0}^{n-1} \eta_k(A) \quad (n = 2, 3, 4, \ldots). \quad (3) \]
(Proof) By the duality between $\varepsilon_B$ and $\delta_B$ ($\varepsilon_B(A)^c = \delta_B(A^c)$) and the distributivity of $\delta_B$ ($\delta_B(A \cup A') = \delta_B(A) \cup \delta_B(A')$), we can write

$$\eta_n(A) = A_n \cap \varepsilon_{B_n}(A_n)^c = A_n \cap \delta_{B_n}(A_n^c) = A_n \cap \delta_{B_n}(\eta_{n-1}(A) \cup A_{n-1}^c)$$

$$= \left( A_n \cap \delta_{B_n}(\eta_{n-1}(A)) \right) \cup \left( A_n \cap \delta_{B_n}(A_{n-1}^c) \right).$$

By the assumption $B_n \subseteq B_{n-1}$ and the monotonicity of $\delta_B$ ($\delta_B(A) \subseteq \delta_B(A')$ if $A \subseteq A'$ and $B \subseteq B'$), we get

$$A_n \cap \delta_{B_n}(A_{n-1}^c) \subseteq A_n \cap \delta_{B_{n-1}}(A_{n-1}^c) = A_n \cap \varepsilon_{B_{n-1}}(A_{n-1}^c) = A_n \cap A_{n-1}^c = \emptyset.$$ 

On the other hand, one can easily show that $A_n = A \setminus \bigcup_{k=0}^{n-1} \eta_k(A)$ and then,

$$A_n \cap \delta_{B_n}(\eta_{n-1}(A)) = (A \cap \delta_{B_n}(\eta_{n-1}(A))) \setminus \bigcup_{k=0}^{n-1} \eta_k(A).$$

Furthermore, by the assumption $B_n \subseteq (\oplus_{k=0}^{n-2} B_k)^s$, the monotonicity of $\delta_B$ and the ordering relation of $\gamma_B$ ($\gamma_B(A) \subseteq A$), we get

$$\delta_{B_n}(\eta_{n-1}(A)) \subseteq \delta_{(\oplus_{k=0}^{n-2} B_k)^s}(A_{n-1}) = \delta_{(\oplus_{k=0}^{n-2} B_k)^s} \circ \varepsilon_{(\oplus_{k=0}^{n-2} B_k)A} = \gamma_{(\oplus_{k=0}^{n-2} B_k)A} \subseteq A$$ 

and then, $A \cap \delta_{B_n}(\eta_{n-1}(A)) = \delta_{B_n}(\eta_{n-1}(A))$. Gathering them, we get (3). \qed

Note that (2) induces that $\{B_n\}$ must satisfy

$$B_1 \supseteq B_2 \supseteq B_3 \supseteq \cdots \quad \text{and} \quad \{0\} \subseteq B_1 \cap B_0^c.$$ 

When one takes a symmetric SE $B (\supseteq \{0\})$ with respect to the origin and puts all $B_n$s to be equal to this $B$ ($B_n = B$ for $n = 0, 1, 2, \ldots$), $\{B_n\}$ satisfies (2), hence Theorem 3.3 and 3.4 include the symmetric version we proposed in [4]. For asymmetric cases, one of the simple settings to satisfy (2) is that one takes any $B_0 (\supseteq \{0\})$ and puts all successive SEs such as $B_n = B_0^n$ ($n = 1, 2, 3, \ldots$).

Algorithm 1 shown below produces the first and the second contours $\eta_0(A)$ and $\eta_1(A)$ of a given image $A$, respectively, by using given SEs $B_0$ and $B_1$. If one keeps these $\eta_0(A)$ and $\eta_1(A)$, and input them to Algorithm 2 with SEs $\{B_n\}$ satisfying the condition (2) ($B_0$ and $B_1$ must be same as ones used in Algorithm 1), then the original image $A$ is reconstructed exactly.

Note that Algorithm 2 does not use $B_0$ and $B_1$. In fact, if you just want to reconstruct the original image $A$ (does not need each $\eta_n(A)$ defined by using $\{B_n\}$), you may choose some SE $B$ satisfying $\{0\} \subseteq B \subseteq B_1 \cap B_0^c$ and use $B$ as $B_n$ ($n = 2, 3, 4, \ldots$) in Algorithm 2.

---

**Algorithm 1 Decomposition**

**Require:** $A$: a target image; $B_0$, $B_1$: structuring elements.

**Ensure:** $\eta_0$, $\eta_1$: the first and the second contours.

1: $A_1 := \varepsilon_{B_0}(A)$;
2: $A_2 := \varepsilon_{B_1}(A_1)$;
3: $\eta_0 := A \setminus A_1$;
4: $\eta_1 := A_1 \setminus A_2$;
5: **return** $\eta_0$ and $\eta_1$. 

---

5
Algorithm 2 Reconstruction

Require: $\eta_0, \eta_1$: the first and the second contours; $\{B_n\}$: a sequence of structuring elements.
Ensure: $A$: the reconstructed image.

1: $A := \eta_0 \cup \eta_1$;
2: $n := 2$;
3: $\eta_n := \delta_{B_n}(\eta_{n-1}) \setminus A$;
4: while $\eta_n \neq \emptyset$ do
5: $A := A \cup \eta_n$;
6: $n := n + 1$;
7: $\eta_n := \delta_{B_n}(\eta_{n-1}) \setminus A$;
8: end while
9: return $A$.

4. Example

Here we show an example. Usually, grayscale images in CG are based on intensity (brightness), so we treat white and black pixels as foreground and background of image, respectively. Then, the images below are reversed with respect to black and white contrary to the explanations about morphology stated in the section 1.

Fig. 4. The original image $A$: a binarized one of “Lenna” (512 × 512 pixels).

Fig. 5. The 1st contour $\eta_0(A)$ of $A$ with respect to $\{B_n\}$.

Fig. 6. The 2nd contour $\eta_1(A)$ of $A$ with respect to $\{B_n\}$.

Fig. 7. The reconstructed image $\bigcup_{n=0}^{N} \eta_n(A)$, where $N = 13$ in this case.

Fig. 8. A grayscale representation of $\{\eta_k(A)\}_{k=0}^{13}$: we set brightness of pixels on $\eta_0(A) \sim \eta_{13}(A)$ to bright $\sim$ dark gradually.

Fig. 9. An asymmetric SE $B_0$ we used (a triangle in 9 × 9); $B_n := B_0^n$ ($n = 1, 2, 3, \ldots$).

5. Application 1: an expansion to grayscale or color images

In morphological theory, each grayscale image is treated as an object in 3 dimensional space: each intensity $z = g(x, y)$ at the planar point $(x, y)$ is considered as the upper limit of the cubic object regarded as a grayscale image with respect to $z$ coordinate in the $xyz$-space.

From this point of view, the contour (edge) of the cubic object is just the graph of $g(x, y)$ and the morphological edge detection in this sense is meaningless. But when one treats each
intensity as a set of bits in the sense of binary coded number and consider a grayscale image as a set of some bitmap planes (e.g., a set of 8 bitmaps for 256-level grayscale image), our method can be easily applied. An example is shown below. The original image $A$ (Fig.10) is reconstructed from the two “contours” $\eta_0(A)$ and $\eta_1(A)$ (Fig.11 and 12, respectively), where we use the $3 \times 3$ square SE $B_0$ (Fig.13).

Furthermore, our method can be applicable to color images in the same sense stated above.
A color image is often treated as the triplet of grayscale images in which each image represents red, green and blue plane of the original one, respectively. Hence, by applying the method to each plane, the decomposition and reconstruction can be done for color images.

6. Application 2: generating new images

By using our method, any image can be decomposed to 2 contours and can be reconstructed from them exactly if we use the same SEs \( \{ B_n \} \) in each process. But if we use different SEs for reconstruction from ones for decomposition, then we may get a varied image from the original one (Fig.14).

Then, if we chose an appropriate SE, there might be possibility to generate a new (expected) image from the original one. For example, in 2000, a generating method for calligraphy characters with scratched look from normal shape of characters was proposed [2]. In this method, a morphological thinning process and some dilations by the SE which was consist of discontinuous points were used to “scratch” characters. Fig. 17 shows an example from [2].

Similar results can be obtain by slightly modifying our method. Namely, while the reconstruction of contours

\[
\eta_n(A) = \delta_{B_n}(\eta_{n-1}(A)) \setminus \bigcup_{k=0}^{n-1} \eta_k(A) \quad (n = 2, 3, 4, \ldots)
\]

were being done, we made other “contours”

\[
\tilde{\eta}_n(A) = \delta_{\tilde{B}_n}(\tilde{\eta}_{n-1}(A)) \setminus \bigcup_{k=0}^{n-1} \eta_k(A) \quad (n = 2, 3, 4, \ldots)
\]

by using randomly scratched SEs \( \{ \tilde{B}_n \} \). Note that the sets for subtraction in making \( \{ \tilde{\eta}_n(A) \} \), i.e., \( \bigcup_{k=0}^{n-1} \eta_k(A) \), are same as ones in \( \{ \eta_n(A) \} \). Some examples are shown in Fig. 18–19. The results are to be pithy, i.e., each central core (skeleton) of characters is “scratched.”

![Figure 16. Original image.](image16.png)
![Figure 17. Scratched image of Fig.16 [2].](image17.png)

![Figure 18. Scratched image to be pithy 1.](image18.png)
![Figure 19. Scratched image to be pithy 2.](image19.png)

7. Conclusion

We introduced an algorithm to decompose and reconstruct a binary image, which uses 2 contours of the original image and revised it to be applicable for asymmetric SEs. Moreover, we show some applications of the algorithm.
There are many papers (e.g., see [1]) that are extension or related to the methods of Maragoth & Shafer or Pai & Hansen, i.e., researches on reconstruction methods based on skeleton. But we could not find one similar to our method.

Our method can be thought as a kind of fill operations. The first contour $\eta_0(A)$ consists of “boundaries” (defined by SEs) of shapes on the original image $A$, while the second contour $\eta_1(A)$ has an information of “internal direction” (also defined by SEs) of each shape. The reconstruction method proposed here can be seen as a filling process of areas surrounded by the “boundaries”. The original image $A$ is reconstructed exactly since $\eta_0(A)$ and $\eta_1(A)$ have entire informations of $A$ in such a sense. So, it differs from image inpainting methods, those which guess and fill the lost information of a given image (e.g., see [3]).

At first, our motivation is to construct a new coding method of binary image using the morphological theory, especially that is to be useful for the image compression. From the viewpoint of mathematical morphology, it was successful since the number of points in the contours $\eta_0(A)$ and $\eta_1(A)$ is much less than that of the original image $A$ in general. But we doubt that not only our method but also the methods of previous works have practical benefit to the image compression. There might not so differ to these methods from the traditional ones (e.g., a simple run-length algorithm).

In other hand, our method might be a seed of some imaging technology such as a new transformation/representation of image. We showed these possibilities by some applications.

References
[1] D. Farazmanesh, A. Tavakoli, Morphological Decomposition and Compression of Binary Images via a Minimum Set Cover Algorithm, J. Math. Imag. Vision, 61 (2019), 71–83.
[2] T. Ichikawa, T. Idogawa and M. Tsutsumi, Generating Various Calligraphy Characters with Scratched Look or Blurred Look by Morphological Operator, Japan SIAM, 10 (2000), 263–272 (in Japanese).
[3] N. Jawas, N. Suciati, Image Inpainting using Erosion and Dilation Operation, Int. J. Adv. Sci. Tech., 51 (2013), 127–134.
[4] M. Lu and T. Idogawa, Contour Decomposition and Reconstruction of Image Based on Mathematical Morphology, Japan SIAM, 19 (2009), 205–234 (in Japanese).
[5] P. Maragos and R. W. Schafer, Morphological Skeleton Representation and Coding of Binary Images, IEEE Trans. ASSP, 34 (1986), 1228–1244.
[6] T. W. Pai and J. H. L. Hansen, An improved image coding algorithm using morphological operator theory, Proc. IEEE Int’l Conf. ICASSP-91, (1991), 2761–2764.
[7] Pierre Soille, Morphological Analysis, Springer-Verlag, Berlin, 1999.