Morphological design of Discrete-Time Cellular Neural Networks

Brugge, Mark Harm ter

Chapter 6 Optimal and Unique Decomposition of Convex Structuring Elements

The previous chapter describes a decomposition method for convex binary structuring elements as part of the decomposition method that decomposes arbitrary binary structuring elements. The number of templates in the decomposition depends on the order in which the reduction rules are applied to the initial prime decomposition. A good order exists but is hard to define in general because of the large amount of reduction rules that apply to the 13 prime factors. In this chapter we present an approach that transforms the convex structuring element into an orientation that allows for an initial decomposition in terms of only 7 prime factors. The set of reduction rules that can be applied to this decomposition reduces so drastically that it is possible to compactly enumerate these rules together with the order in which they should be applied, giving a simple optimal decomposition algorithm.

This chapter is based on [11].

6.1 Introduction

In Section 5.5 we showed how an arbitrary convex SE can be decomposed into a sequence of dilations of convex 3 × 3 SEs. As a first step the SE is divided into convex sub−sets. This division is certainly not unique and influences the number of 3 × 3 SEs in the final solution. Up until now, no method exists for making the optimal decomposition. After this, an algorithm is applied to all convex subsets. In a nutshell, this algorithm consists of the following steps. First, the chain code of the SE to be decomposed is determined. Then Appendix A.3 is used to transform this chain code into a shape decomposition. To compensate for the difference in location between the shape decomposition and the SE to be decomposed shifting SEs are added. Finally the reduction rules in Appendix A.4 are applied as often as possible to reduce the number of SEs and to obtain the final solution.

The method proposed in Section 5.5 is rather straightforward but does not guarantee an optimal solution. An optimal solution is found only when the appropriate set of reduction rules is applied in the appropriate order. This appropriate set and order, however, is hard to define in a concise manner since the number of different SEs and therefore the number of applicable rules rises steadily during the reduction process. In [107] this problem is tackled by merging similar reduction rules into generic ones and defining an order on them. Rather than being defined in terms of exact images, these generic reduction rules are defined in terms of bounding boxes. An example of such a rule and two of its instances is given in Figure 6.1.

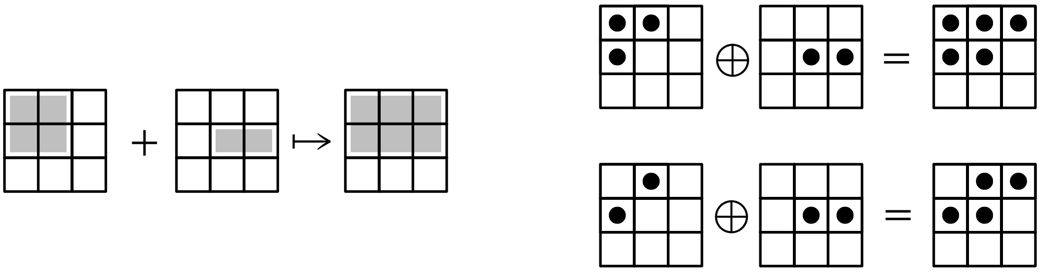


Figure 6.1: A generic reduction rule and two of its instances. The gray boxes in the generic rule define the bounding boxes of the SEs in the specific rules.

To further reduce the number of possibilities, the SE to be decomposed is first rotated into an orientation for which only right and down shifting SEs (Q_{32} and Q_{128}) are required to shift the shape decomposition to its correct position. After adding the shifting SEs, the generic reduction rules are applied in the specified order to obtain an optimal decomposition. Finally each of the elements in the decomposition is rotated with the inverse of the previously mentioned angle in order to transform the decomposition of the rotated version of the SE into a decomposition of the original SE.

Though the algorithm proposed in [107] is cleverly found, it is full of morphological operations and difficult to understand, especially for researchers in the CNN community who generally are not Morphological experts. For this reason we will present an alternative decomposition algorithm, which on the one hand is very straightforward to use since all morphological operations are eliminated and on the other still has the advantage of being optimal.

6.2 Transformation of the Original SE

Similar to [107], we first transform the SE to be decomposed, say A, to reduce the number of cases that has to be distinguished and therefore the complexity of the algorithm. In [107] the SE was rotated by a multiple of 90 degrees to make sure that only right and down shifting SEs are required to shift the shape decomposition to its correct position. Here we use rotation and reflection to make sure that chain code of the transformed image $B = T_i(A)$ satisfies:

$$b_0 \leq b_4 \wedge b_2 \leq b_6 \wedge b_3 \geq b_7 \qquad (6\text{–}1)$$

First, counterclockwise rotation of 90 degrees and reflection in the x–axis are defined by, respectively:

$$R(x,y) = (y, -x), \qquad M_x(x,y) = (x, -y) \qquad (6\text{–}2)$$

It is easy to see that 8 different transformations can be generated by combining zero or more of these transformations. These transformations can be defined by:

$$T_i(x,y) = \begin{cases} R^i(x,y) & ;\ 0 \le i \le 3 \\ R^{i-4}(M_x(x,y)) & ;\ 4 \le i \le 7 \end{cases} \tag{6–3}$$

Note that allowing also reflection in the y–axis does not generate additional transformations. In Figure 6.2 all transformations are visualized. Note that the chain code of every transformed image $B = T_i(A)$ is a permutation of the chain code of A.

Now we are going to show that we can always find a transformation $T_i, i = 0..7$, such that Equation (6–1) is satisfied. Since A is a convex polygon, the length of the sides of A and therefore the elements of the chain code of A satisfies (see [87],[107]):

$$\begin{aligned} a_1 + a_2 + a_3 &= a_5 + a_6 + a_7 \\ a_0 + a_1 + a_7 &= a_3 + a_4 + a_5 \end{aligned} \tag{6–4}$$

These two equations can be combined to obtain the following two equations:

$$\begin{aligned} a_0 + 2a_1 + a_2 &= a_4 + 2a_5 + a_6 \\ a_0 + a_6 + 2a_7 &= a_2 + 2a_3 + a_4 \end{aligned} \tag{6–5}$$

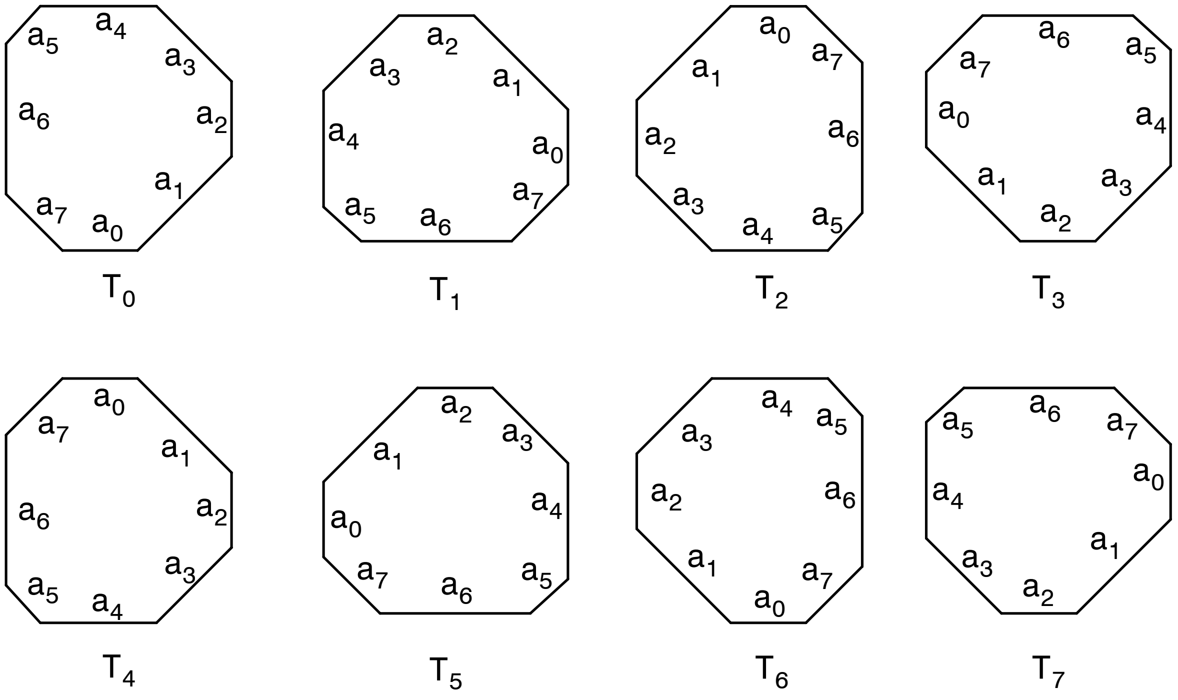


Figure 6.2: All possible permutations obtained by rotation of multiples of 90 degrees and reflection in the x–axis.

Using Formula (a–1) and the fact that $a_i \ge 0$, we can now derive the following relations between the elements of the chain code of A:

$$\begin{aligned} a_0 \le a_4 \wedge a_2 \le a_6 &\Rightarrow a_1 \ge a_5 \\ a_0 \ge a_4 \wedge a_2 \ge a_6 &\Rightarrow a_1 \le a_5 \\ a_0 \le a_4 \wedge a_2 \ge a_6 &\Rightarrow a_3 \le a_7 \\ a_0 \ge a_4 \wedge a_2 \le a_6 &\Rightarrow a_3 \ge a_7 \end{aligned} \tag{6–6}$$

This implies that certain combinations of length relations between the sides of A are invalid. In fact, Formula (6–6) reduces the number of combinations of length relations between opposite sides from 16 to the 8 valid combinations listed in Table 6.1. For each valid combination, this table also gives the transformation T_i, for which the chain code of $B = T_i(A)$ satisfies Equation (6–1). For example, if the chain code of A satisfies:

$$a_0 \geq a_4 \wedge a_2 \leq a_6 \wedge a_1 \leq a_5 \wedge a_3 \geq a_7 \tag{6–7}$$

the appropriate transformation is T_3, since for this transformation we have:

$$\begin{array}{llll} b_0 = a_{2,} & b_1 = a_{3,} & b_2 = a_{4,} & b_3 = a_{5,} \\ b_4 = a_{6,} & b_5 = a_{7,} & b_6 = a_{0,} & b_7 = a_{1,} \end{array}$$

Substituting this into Equation (6–7) gives:

$$b_6 \geq b_2 \wedge b_0 \leq b_4 \wedge b_7 \leq b_3 \wedge b_1 \geq b_5 \tag{6–8}$$

which implies Equation (6–1).

$a_0 \quad a_4$	$a_2 \quad a_6$	$a_1 \quad a_5$	$a_3 \quad a_7$	Transformation
≤	≤	≥	≤	T_7
≤	≤	≥	≥	T_0
≤	≥	≤	≤	T_6
≤	≥	≥	≤	T_1
≥	≤	≤	≥	T_3
≥	≤	≥	≥	T_4
≥	≥	≤	≤	T_2
≥	≥	≤	≥	T_5

Table 6.1: This table contains all possible combinations of length relations between opposite sides. For each combination the transformation that transforms the shape into the desired orientation is listed in the last column.

After finding the appropriate transformation using Table 6.1, the transformed image $B = T_i(A)$ is found using Table 6.2. For each transformation T_i this table gives the shape (chain code) and location of B. To see that the listed locations are correct one can use Formula (6–3).

T_i	Chain Code of B	$X_{max}(B)$	$Y_{max}(B)$
T_0	$< a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7 >$	$X_{max}(A)$	$Y_{max}(A)$
T_1	$< a_6, a_7, a_0, a_1, a_2, a_3, a_4, a_5 >$	$Y_{max}(A)$	$-X_{min}(A)$
T_2	$< a_4, a_5, a_6, a_7, a_0, a_1, a_2, a_3 >$	$-X_{min}(A)$	$-Y_{min}(A)$
T_3	$< a_2, a_3, a_4, a_5, a_6, a_7, a_0, a_1 >$	$-Y_{min}(A)$	$X_{max}(A)$
T_4	$< a_4, a_3, a_2, a_1, a_0, a_7, a_6, a_5 >$	$X_{max}(A)$	$-Y_{min}(A)$
T_5	$< a_6, a_5, a_4, a_3, a_2, a_1, a_0, a_7 >$	$-Y_{min}(A)$	$-X_{min}(A)$
T_6	$< a_0, a_7, a_6, a_5, a_4, a_3, a_2, a_1 >$	$-X_{min}(A)$	$Y_{max}(A)$
T_7	$< a_2, a_1, a_0, a_7, a_6, a_5, a_4, a_3 >$	$Y_{max}(A)$	$X_{max}(A)$

Table 6.2: Recipe for computing the shape and location of the transformed image $B = T_i(A)$.

6.3 Shape Decomposition

After applying the appropriate transformation T_i to the original image A giving image $B = T_i(A)$, we make a decomposition of the shape of B. This means that we have to find a decomposition for some image $B' \sim B$, where (as discussed in Section 5.5) $B' \sim B$ denotes that images B' and B are identical except for translation. To this purpose we distinguish two types of images: the rhombus images and the non−rhombus images. A *rhombus image* is defined as any image B whose chain code satisfies:

$$(0, b_1, 0, b_3, 0, b_1, 0, b_3); \qquad b_1, b_3 \neq 0 \tag{6–9}$$

Logically, a non−rhombus image is any image that is not a rhombus image. We first assume that B is a non−rhombus image. For such an image, a shape decomposition is given by:

$$B' = b_5Q_{10} \oplus (b_4 - b_0)Q_{11} \oplus b_7Q_{17} \oplus b_2Q_{18} \oplus b_0Q_{24} \oplus (b_3 - b_7)Q_{89} \tag{6–10}$$

To see that this is true, we use the theorem given in [87], which gives a very interesting relation between image dilation and vector addition of chain codes.

Theorem a.1 *Let images $S = (s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7)$, $A = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$, and $B = (b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7)$. If $s_i = a_i + b_i$ for every $i = 0..7$ then $S \sim A \oplus B$, except when $S = (0, s_1, 0, s_3, 0, s_1, 0, s_3)$, for nonzero s_1, s_3 and the chain code of A and B are $(0, s_1, 0, 0, 0, s_1, 0, 0)$ and $(0, 0, 0, s_3, 0, 0, 0, s_3)$, or vise versa.*

Retrieving the chain code of each Q_i in Equation (6−10) from Table a.2, multiplying these chains codes with the appropriate factors, and adding them gives:

$$\begin{array}{rcccccccccccc}
b_5Q_{10} & = & (& 0, & b_5, & 0, & 0, & 0, & b_5, & 0, & 0 &) & \\
(b_4-b_0)Q_{11} & = & (& 0, & b_4-b_0, & 0, & 0, & b_4-b_0, & 0, & b_4-b_0, & 0 &) & \\
b_7Q_{17} & = & (& 0, & 0, & 0, & b_7, & 0, & 0, & 0, & b_7 &) & \\
b_2Q_{18} & = & (& 0, & 0, & b_2, & 0, & 0, & 0, & b_2, & 0 &) & \\
b_0Q_{24} & = & (& b_0, & 0, & 0, & 0, & b_0, & 0, & 0, & 0 &) & \\
(b_3-b_7)Q_{89} & = & (& 0, & b_3-b_7, & 0, & b_3-b_7, & 0, & 0, & 2b_3-2b_7, & 0 &) & \\ \hline
 & & (& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & b_7 &) & +
\end{array} \qquad (6\text{–}11)$$

where the relations:

$$b_1 = b_5 + b_4 - b_0 + b_3 - b_7\,, \qquad b_6 = b_4 - b_0 + b_2 + 2b_3 - 2b_7$$

follow directly from the length relations that hold on the sides of a convex polygon like we already saw in Formula (6–4) and Formula (6–5). Repeated application of Theorem a.1 shows that $B' \sim B$. To avoid the exception in Theorem a.1, the addition needs to be done in the correct order. The only two additions that may possibly not be performed are the addition of the chain codes Q_{10} and Q_{17}. Therefore, first add the chain code of all Q_i's except for Q_{10} and Q_{17} giving the chain code $(b_0, b_1 - b_5, b_2, b_3 - b_7, b_4, 0, b_6, 0)$. Since the last element is zero, we may now add b_5Q_{10} giving $(b_0, b_1, b_2, b_3 - b_7, b_4, b_5, b_6, 0)$. Finally we add b_7Q_{17}. For $b_7 = 0$, there is nothing to add and the proof is completed. For $b_7 > 0$, the pre–condition of Theorem a.1 is only violated when:

$$b_1 = b_5 \neq 0 \;\wedge\; b_3 = b_7 \;\wedge\; b_0 = 0 \;\wedge\; b_2 = 0 \;\wedge\; b_4 = 0 \;\wedge\; b_6 = 0$$

This, however, cannot be the case since B is not a rhombus (see Equation (6–9)), which completes the proof.

Notice that after applying the appropriate transformation, the shape of every SE except for the rhombus SEs (Equation (6–9)) can be decomposed using only 6 different 3 × 3 SEs, called prime factors. This is of interest since all literature on the decomposition of convex SEs mention a primitive set of 13 elements (see also Figure a.3). However, even more interesting is that _ contrary to what was assumed originally _ the decomposition is unique. To see that this is true, first notice that the number of Q_{24}'s in the decomposition must be b_0 according to Formula (6–11), since this is the only prime factor that can contribute to the first element of the chain code of B. For the same reason, the number of SEs Q_{18}, Q_{10}, and Q_{17} must be b_2, b_5, and b_7 respectively. Since the number of Q_{17}'s is uniquely determined by b_7 and since Q_{11} is the only other basic prime factor that influences b_3, the number of Q_{11}'s is also uniquely determined. After this the only left–over choice is Q_{89}, which completes the proof.

It remains to give the decomposition for rhombus shapes. These shapes cannot be decomposed into one or more of the 6 prime factors described above. For these shapes we need an additional prime factor Q_{186}. The decomposition is given by:

$$B' = (b_5 - 1)Q_{10} \oplus (b_7 - 1)Q_{17} \oplus Q_{186} \qquad (6\text{–}12)$$

Again Theorem a.1 can be used to show the correctness. Notice that this decomposition requires only 3 different Q_i's. However, the decomposition into these Q_i's is not unique. Alternative decompositions are obtained by replacing one or more of the (Q_{10}, Q_{17}) pairs in Formula (6–12) by Q_{186}. Joining the prime factors used in the rhombus and non–rhombus decomposition gives the set of prime factors shown in Figure 6.3.

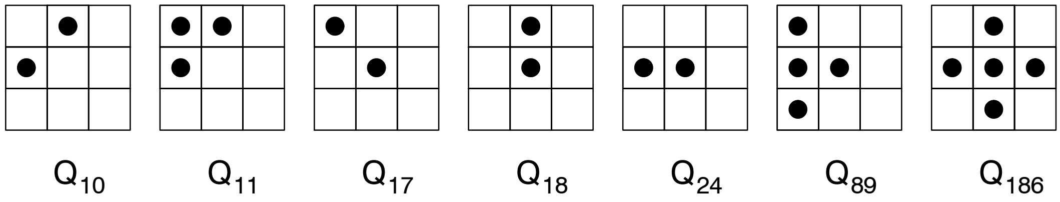


Figure 6.3: Prime factors of convex images.

Notice that transforming the SE as we did, before analyzing and decomposing it has great advantages. First, the number of different cases that has to be distinguished is much smaller and the decomposition for these cases is much simpler. We only had to distinguish between rhombus and non–rhombus shapes. In [107] four cases had to be distinguished (the cases listed in Appendix A.3). The decomposition for each of these cases contains eight different SEs and all multiplication factors are min/max expressions of two or more chain code elements. In our approach, the multiplication factors are simply chain code elements or a subtraction of two chain code elements. In [87] the decompositions are less complicated than in [107] but in this article, 9 cases (1 rhombus and 8 non–rhombus shapes) had to be distinguished and the multiplication factors still contain the conditional min and max operators, which are onerous in calculations. Finally, in both [87] and [107] the shape decomposition is not unique and uses 13 different primitives. In our approach we use only 7 primitives. Furthermore, non–rhombus shapes can be uniquely decomposed into six primitives and for rhombus shapes the decomposition is not unique, but the possibilities are easily enumerated. This gives a more clear understanding of convex images and the way in which they can be decomposed.

6.4 Addition of Shifters to the Shape Decomposition

In order to compensate for the difference in location between image B and its shape decomposition B', we have to add zero or more shifting SEs ($Q_2, Q_8, Q_{32}, Q_{128}$). The required amount of shift in horizontal and vertical direction is defined by:

$$\Delta_X = X_{max}(B) - X_{max}(B'), \quad \Delta_Y = Y_{max}(B) - Y_{max}(B') \tag{6–13}$$

where the values $X_{max}(B')$ and $Y_{max}(B')$ can be evaluated using Table a.2 (that lists the X_{max} and Y_{max} value for each convex SE) and the following properties that hold for arbitrary nonempty images A and B (Proposition 3.6 in [87]):

$$\begin{aligned} X_{max}(A \oplus B) &= X_{max}(A) + X_{max}(B) \\ Y_{max}(A \oplus B) &= Y_{max}(A) + Y_{max}(B) \end{aligned} \tag{6–14}$$

If B is not a rhombus, the values $X_{max}(B')$ and $Y_{max}(B')$ are derived from Formula (6–10):

$$X_{max}(B') = 0, \quad Y_{max}(B') = b_3 - b_7 \tag{6–15}$$

For rhombus images, the location of the shape decomposition B' (see Formula (6–12)) is defined by:

$$X_{max}(B') = 1, \quad Y_{max}(B') = 1 \tag{6–16}$$

Formula (6–14) and the values X_{max} and Y_{max} of the shifting SEs (see Table a.2) imply that the number of shifting SEs that need to be added to the shape decomposition is defined by:

$$\begin{aligned}
\Delta_X \geq 0 \;\wedge\; \Delta_Y \geq 0 \;&\Rightarrow\; B = B' \oplus \Delta_X Q_{32} \oplus \Delta_Y Q_{128} \\
\Delta_X \geq 0 \;\wedge\; \Delta_Y < 0 \;&\Rightarrow\; B = B' \oplus \Delta_X Q_{32} \oplus (-\Delta_Y) Q_2 \\
\Delta_X < 0 \;\wedge\; \Delta_Y \geq 0 \;&\Rightarrow\; B = B' \oplus (-\Delta_X) Q_8 \oplus \Delta_Y Q_{128} \\
\Delta_X < 0 \;\wedge\; \Delta_Y < 0 \;&\Rightarrow\; B = B' \oplus (-\Delta_X) Q_8 \oplus (-\Delta_Y) Q_2
\end{aligned} \tag{6–17}$$

which gives the so called initial decomposition. Notice that shifting SEs only shift images but do not change the shape. This follows directly from Theorem a.1 and the fact that all elements in the chain code of shifters are zero.

6.5 Reduction

After having added the shifting SEs to the image B' we have a decomposition for image B. Generally this decomposition is not optimal and the number of elements should be reduced. This is done by repeatedly replacing two 3×3 SEs in the decomposition by a single 3×3 SE until no reductions are possible anymore. To this purpose one can use Table a.3, which lists all possible reduction rules for convex 3×3 SEs. In Appendix A.4 we already saw that the SEs and in particular the number of SEs in a final reduction, depends on the sequence of reduction rules that is applied. This implies that certain reduction traces lead to optimal solutions while others don't. According to [107] a strategy that leads to an optimal reduction of our initial decomposition (as derived in the previous two sections) is given by the generic reduction scheme shown in Figure 6.4.

The rules in Figure 6.4 are slightly different from the rules given in [107]. For some rules we have given rotational variants since some of the SEs in our initial decomposition are rotational variants of the SEs in initial decompositions of [107]. Furthermore, a rotational variant of Type 1 has been omitted since the limited number of SEs in our initial decomposition makes this rule superfluously. Finally, the rules of Type 8, 9, and 10 are extensions of the rules as they were originally presented. This extension was necessary since we use four different shifters in the initial decomposition (see Formula (6–17)), whereas only two different shifters are used in [107]. Since we have a small number of SEs in the initial decomposition, we can easily enumerate all possible reduction rules, by substituting the SEs in the initial decomposition and (for the following steps) taking into account, the SEs that are produced by these specific rules. For the four cases distinguished in Formula (6–17), this leads to the specific set op reduction rules given in Table 6.3.

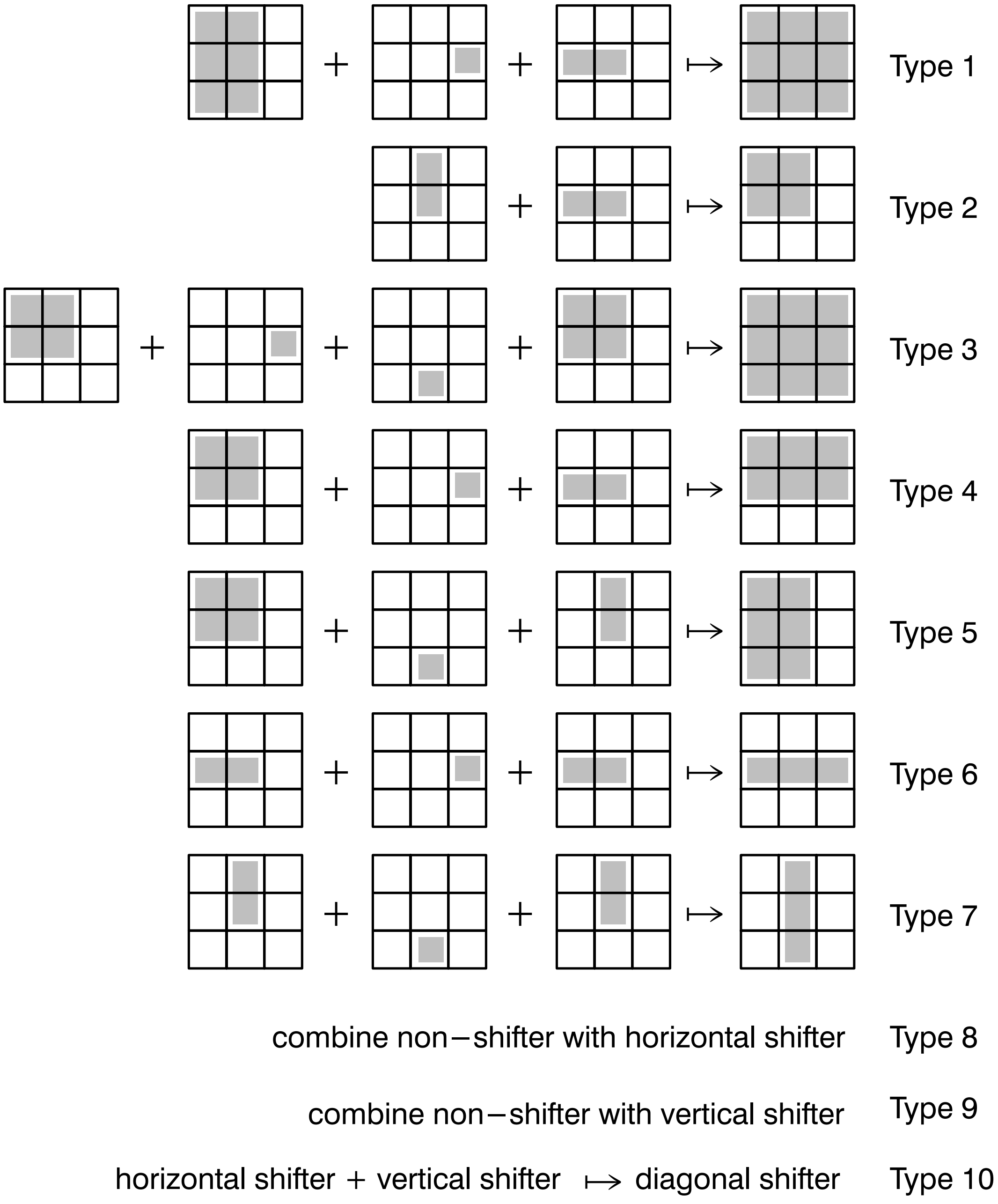

Type 1
Type 2
Type 3
Type 4
Type 5
Type 6
Type 7
combine non−shifter with horizontal shifter
Type 8
combine non−shifter with vertical shifter
Type 9
horizontal shifter + vertical shifter ↦ diagonal shifter
Type 10

Figure 6.4: Generic reduction rules for optimal reduction.

$\varDelta_X \geq 0 \wedge \varDelta_Y \geq 0$

Reduction Rule	Type
89 + 32 + 24 ↦ 251	1
18 + 24 ↦ 27	2
10 + 32 + 128 + 10 ↦ 84	3
10 + 32 + 128 + 11 ↦ 94	3
10 + 32 + 128 + 17 ↦ 186	3
10 + 32 + 128 + 27 ↦ 254	3
11 + 32 + 128 + 11 ↦ 95	3
11 + 32 + 128 + 17 ↦ 187	3
11 + 32 + 128 + 27 ↦ 255	3
17 + 32 + 128 + 17 ↦ 273	3
17 + 32 + 128 + 27 ↦ 443	3
27 + 32 + 128 + 27 ↦ 511	3
10 + 32 + 24 ↦ 30	4
11 + 32 + 24 ↦ 31	4
17 + 32 + 24 ↦ 51	4
27 + 32 + 24 ↦ 63	4
10 + 128 + 18 ↦ 90	5
11 + 128 + 18 ↦ 91	5
17 + 128 + 18 ↦ 153	5
27 + 128 + 18 ↦ 219	5
24 + 32 + 24 ↦ 56	6
18 + 128 + 18 ↦ 146	7
10 + 32 ↦ 20	8
11 + 32 ↦ 22	8
17 + 32 ↦ 34	8
18 + 32 ↦ 36	8
24 + 32 ↦ 48	8
27 + 32 ↦ 54	8
89 + 32 ↦ 178	8
90 + 32 ↦ 180	8
91 + 32 ↦ 182	8
146 + 32 ↦ 292	8
153 + 32 ↦ 306	8
219 + 32 ↦ 438	8
10 + 128 ↦ 80	9
11 + 128 ↦ 88	9
17 + 128 ↦ 136	9
18 + 128 ↦ 144	9
20 + 128 ↦ 160	9
22 + 128 ↦ 176	9
24 + 128 ↦ 192	9
27 + 128 ↦ 216	9
30 + 128 ↦ 240	9
31 + 128 ↦ 248	9
34 + 128 ↦ 272	9
36 + 128 ↦ 288	9
48 + 128 ↦ 384	9
51 + 128 ↦ 408	9
54 + 128 ↦ 432	9
56 + 128 ↦ 448	9
63 + 128 ↦ 504	9
32 + 128 ↦ 256	10

$\varDelta_X < \wedge \varDelta_Y < 0$

Reduction Rule	Type
18 + 24 ↦ 27	2
18 + 8 ↦ 9	8
24 + 2 ↦ 3	9
2 + 8 ↦ 1	10

$\varDelta_X \geq 0 \wedge \varDelta_Y < 0$

Reduction Rule	Type
89 + 32 + 24 ↦ 251	1
18 + 24 ↦ 27	2
10 + 32 + 24 ↦ 30	4
11 + 32 + 24 ↦ 31	4
17 + 32 + 24 ↦ 51	4
27 + 32 + 24 ↦ 63	4
24 + 32 + 24 ↦ 56	6
10 + 32 ↦ 20	8
11 + 32 ↦ 22	8
17 + 32 ↦ 34	8
18 + 32 ↦ 36	8
24 + 32 ↦ 48	8
27 + 32 ↦ 54	8
89 + 32 ↦ 178	8
24 + 2 ↦ 3	9
48 + 2 ↦ 6	9
56 + 2 ↦ 7	9
2 + 32 ↦ 4	10

$\varDelta_X < 0 \wedge \varDelta_Y \geq 0$

Reduction Rule	Type
18 + 24 ↦ 27	2
10 + 128 + 18 ↦ 90	5
11 + 128 + 18 ↦ 91	5
17 + 128 + 18 ↦ 153	5
27 + 128 + 18 ↦ 219	5
18 + 128 + 18 ↦ 146	7
18 + 8 ↦ 9	8
146 + 8 ↦ 73	8
9 + 128 ↦ 72	9
10 + 128 ↦ 80	9
11 + 128 ↦ 88	9
17 + 128 ↦ 136	9
18 + 128 ↦ 144	9
24 + 128 ↦ 192	9
27 + 128 ↦ 216	9
8 + 128 ↦ 64	10

Table 6.3: Optimal reduction scheme.

As an example, the generic rule Type 3 only applies when $\Delta_X \geq 0 \wedge \Delta_Y \geq 0$ since this is the only case, in which the initial decomposition may contain both Q_{32} and Q_{128} (see Formula (6−17)). The only SEs in the initial decomposition that have a 2×2 bounding box are Q_{10}, Q_{11}, and Q_{17}. Furthermore the generic rule Type 2 may have produced one or more elements Q_{27}. Therefore the instances op Type 3 are obtained by making every possible combination of two elements of the set $\{Q_{10}, Q_{11}, Q_{17}, Q_{27}\}$ with Q_{32} and with Q_{128} and determining the result using Table a.3. For example, the first instance of Type 3 is explained by the fact that $Q_{10} \oplus Q_{32} = Q_{20}$, $Q_{20} \oplus Q_{128} = Q_{160}$, and $Q_{160} \oplus Q_{10} = Q_{84}$. Note that the third instance of Type 3 is a tricky reduction rule. Like before, we do have $Q_{10} \oplus Q_{32} \oplus Q_{128} = Q_{160}$, but $Q_{160} \oplus Q_{17} \neq Q_{186}$ due to the exception in Theorem a.1 (see Table a.2 for chain codes). However, to see that in our case the rule is a valid reduction rule, observe that the shape decomposition as defined in Formula (6−10) and (6−12) never has the following form:

$$B' = uQ_{10} \oplus vQ_{17}, \quad u, v > 0 \tag{6–18}$$

To see that this is true, observe that the shape decomposition of a rhombus always contains one element Q_{186} and therefore does not satisfy Formula (6−18). For non−rhombus images, Formula (6−10) and (6−18) imply:

$$b_5 = u \ \wedge \ b_4 = b_0 = 0 \ \wedge \ b_3 = b_7 = v \ \wedge \ b_2 = 0$$

Combining this with the side length relation of convex images (see Formula (6−4)) we must have that the chain code of $B$is $(0, u, 0, v, 0, u, 0, v)$, which contradicts with Formula (6−9). Therefore, if the decomposition of a non−rhombus shape consists of $uQ_{10} \oplus vQ_{17}$ for $u, v > 0$, then the decomposition also contains at least one element of the set $\{Q_{11}, Q_{18}, Q_{24}, Q_{89}\}$. This context, and the context Q_{186} for rhombus shapes imply that we may always apply the reduction rule $Q_{10} + Q_{32} + Q_{128} + Q_{17} \mapsto Q_{186}$. As an example we will show this for the context Q_{11}. Using Theorem a.1, Formula (6−14), and the chain codes and location information given in Table a.2 we have $Q_{10} \oplus Q_{32} \oplus Q_{128} \oplus Q_{17} \oplus Q_{11} = Q_{186} \oplus Q_{11}$, since we have:

$$\begin{array}{rcll} Q_{10} & : & (0,1,0,0,0,1,0,0) & 0\ 0 \\ Q_{32} & : & (0,0,0,0,0,0,0,0) & 1\ 0 \\ Q_{128} & : & (0,0,0,0,0,0,0,0) & 0\ 1 \\ Q_{17} & : & (0,0,0,1,0,0,0,1) & 0\ 0 \\ Q_{11} & : & (0,1,0,0,1,0,1,0) & 0\ 0 \\ \hline & & (0,2,0,1,1,1,1,1) & 1\ 1 \end{array} + \qquad \begin{array}{rcll} Q_{186} & : & (0,1,0,1,0,1,0,1) & 1\ 1 \\ Q_{11} & : & (0,1,0,0,1,0,1,0) & 0\ 0 \\ \hline & & (0,2,0,1,1,1,1,1) & 1\ 1 \end{array} +$$

Like we did for Formula (6−11), Theorem a.1 needs to be applied in the correct order to avoid the the exception.

6.6 Transforming the Decomposition

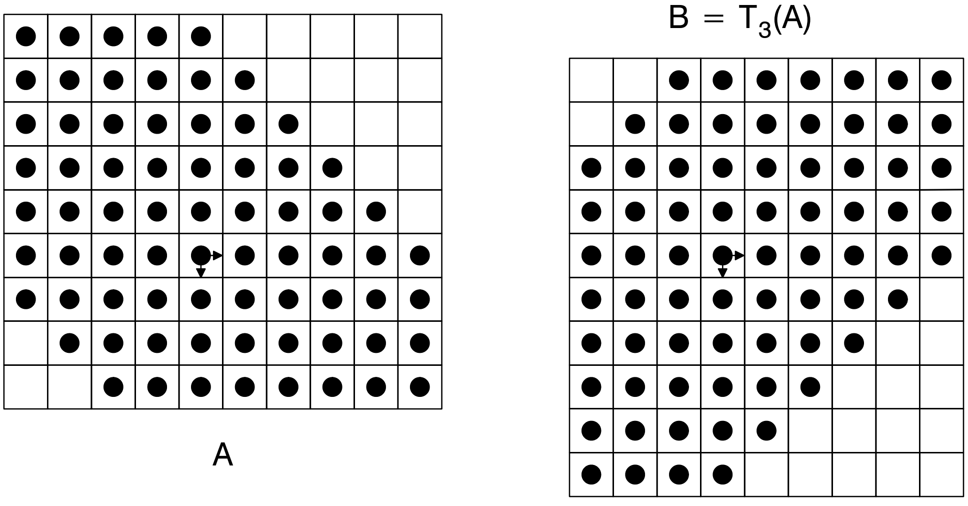
The last step of the algorithm is deriving the decomposition for A by transforming each element $Q_{j_k}, k \leq 1 \leq n$ in the decomposition of $B = T_i(A)$. Since rotation and reflection distribute over dilation, the decomposition of A is given by:

$$A = T_i^{-1}(B) = T_i^{-1}(Q_{j_1} \oplus \ldots \oplus Q_{j_n}) = T_i^{-1}(Q_{j_1}) \oplus \ldots \oplus T_i^{-1}(Q_{j_n})$$

Formula (6−2) and (6−3) can be used to find the inverse of each transformation $T_i, 0 \le i \le 7$. The results are listed in Table 6.4. Due to the naming convention we have adopted for 3×3 SEs (see Appendix A.1), the indices of the transformed 3×3 SEs are easily found by permuting the bits in the binary representation (leftmost bit is least significant bit) of indices j_k using the permutation vectors listed in Table 6.4. For example, $T_1^{-1}(Q_{11})$ is found by permuting the bit sequence 110100000 according to the permutation vector $[6, 3, 0, 7, 4, 1, 8, 5, 2]$, giving 011001000 which is the binary representation of number 38. Therefore $T_1^{-1}(Q_{11}) = Q_{38}$. The result can easily be verified using Table a.1 and the fact that T_1^{-1} is a clockwise rotation of 90 degrees.

T_i	T_i^{-1}	Bit Permutation
T_0	T_0	$[0, 1, 2, 3, 4, 5, 6, 7, 8]$
T_1	T_3	$[6, 3, 0, 7, 4, 1, 8, 5, 2]$
T_2	T_2	$[8, 7, 6, 5, 4, 3, 2, 1, 0]$
T_3	T_1	$[2, 5, 8, 1, 4, 7, 0, 3, 6]$
T_4	T_4	$[6, 7, 8, 3, 4, 5, 0, 1, 2]$
T_5	T_5	$[8, 5, 2, 7, 4, 1, 6, 3, 0]$
T_6	T_6	$[2, 1, 0, 5, 4, 3, 8, 7, 6]$
T_7	T_7	$[0, 3, 6, 1, 4, 7, 2, 5, 8]$

Table 6.4: This table shows how the elements Q_{k_j} in the decomposition of B are transformed into elements $T_i^{-1}(Q_{k_j})$ to obtain the decomposition of A. The first column contains the transformation that was used to obtain image $B = T_i(A)$. The second column contains the inverse of this transformation. The last column contains the bit permutation that should be applied to the index k_j such that $T_i^{-1}(Q_j) = Q_{perm(k_j)}$.



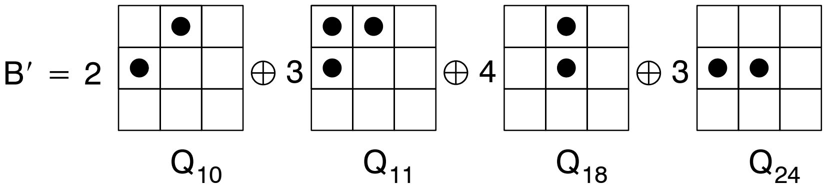
(a) (b)

Figure 6.5: An example SE A and it's transformation $B = T_3(A)$.

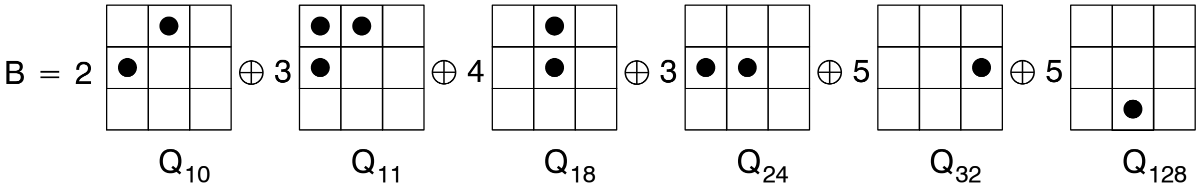
6.7 An Example

As an example we will derive an optimal decomposition for the image A shown in Figure 6.5.a First, note that the chain code of A is $(7,0,3,5,4,0,6,2)$. This chain code is used to find the appropriate transformation $T_i, 0 \leq i \leq 7$ so that he chain code of image $B = T_i(A)$ satisfies Equation (6–1). According to Table 6.1 we can choose $i = 3$ or $i = 4$, since we have $a_0 > a_4$, $a_2 < a_6$, $a_1 = a_5$, and $a_3 > a_7$. We choose $i = 3$ and determine the image $B = T_3(A)$, as shown in Figure 6.5.b.

From Table 6.2 we can derive that the chain code of B is given by $(3,5,4,0,6,2,7,0)$. Furthermore, this table implies $X_{max}(B) = -Y_{min}(A) = 5$ and $Y_{max}(B) = X_{max}(A) = 5$. Notice that this shape and location information is confirmed by Figure 6.5.b. Since B is not a rhombus, the shape decomposition for B is found by substituting the chain code of B into Equation (6–10), giving the decomposition for image B':



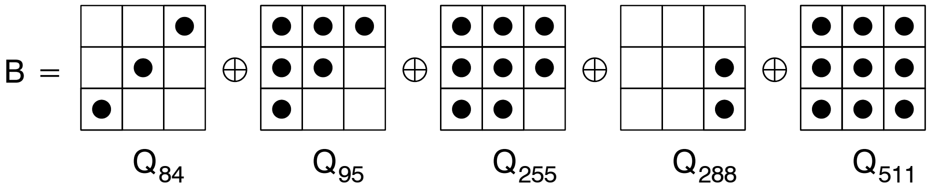
Now Formula (6–13) is used to determine the difference in location between image B' and B. Since $X_{max}(B) = 5$ and $Y_{max}(B) = 5$ and since Formula (6–15) implies $X_{max}(B') = 0$ and $Y_{max}(B') = b_3 - b_7 = 0$, we have $\Delta_X = 5$ and $\Delta_Y = 5$. Substituting these values into Formula (6–17) gives the initial decomposition of image B:



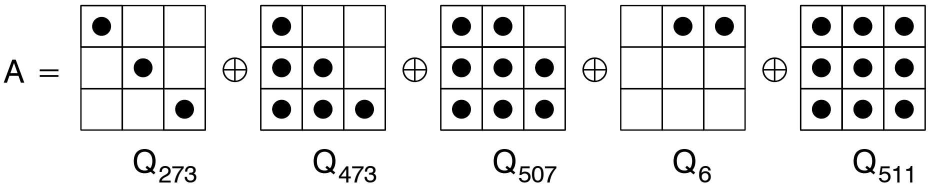
To reduce this expression and to find an optimal decomposition of B we use Table 6.3. According to this table the following reductions should be applied:

$$\begin{array}{rcl}
 & Q_{18} + Q_{24} \mapsto Q_{27} & (3\,\times) \\
Q_{10} + Q_{32} + Q_{128} + Q_{10} & \mapsto Q_{84} & \\
Q_{11} + Q_{32} + Q_{128} + Q_{11} & \mapsto Q_{95} & \\
Q_{11} + Q_{32} + Q_{128} + Q_{27} & \mapsto Q_{255} & \\
Q_{27} + Q_{32} + Q_{128} + Q_{27} & \mapsto Q_{511} & \\
 & Q_{18} + Q_{32} \mapsto Q_{36} & \\
 & Q_{36} + Q_{128} \mapsto Q_{288} &
\end{array}$$

This results in the following optimal decomposition of B:



Finally, since $B = T_3(A)$, the optimal decomposition of $A$is found by transforming each $Q_{j_k}, k \leq 1 \leq 5$ using transformation $T_3^{-1} = T_1$. According to Table 6.4 the indices of the transformed 3 × 3 SEs are found by permuting the bits in the binary representation of the indices j_k using the permutation vector $[6, 3, 0, 7, 4, 1, 8, 5, 2]$. This results in the following optimal decomposition of A:



6.8 Conclusions

In this chapter we introduced an optimal decomposition method for binary convex structuring elements. Optimal decomposition for this class of structuring elements is not new [107]. The difference with the algorithm proposed in [107] is that the algorithm presented in this chapter is much simpler. The existing approach starts with a rotation to reduce the number of cases that have to be distinguished in the following steps. Then a decomposition of the structuring element into 6 of the 13 prime factors is made. Which 6 of the 13 prime factors are in the decomposition depends on the relations that hold for the length of the sides of the structuring element. The initial decomposition is then reduced using a complex 9–step reduction scheme that (despite of the initial rotation) still has to distinguish many cases to obtain the optimal solution. One of the reasons for this complexity is that the algorithm has to deal with a rapidly growing number of different structuring elements that arise from applying the reduction rules to the initial 13 different structuring elements and to the results of these rules.

The approach presented in this chapter also starts with a transformation of the convex structuring element to reduce the number of cases. Instead of just considering the 4 rotational variants, we also use reflection giving 8 possible transformations. The appropriate transformation is found by a simple table look–up. As a second step, the initial decomposition is obtained by substituting the chain code elements into a generic decomposition formula. The decomposition of a rhombus image contains at most 3 different prime factors. A non–rhombus image is decomposed into at most 6 different prime factors. Contrary to the reduction algorithm in [107] that has to deal with 13 initial different structuring elements, the reduction algorithm presented here only has to deal with 7 different prime factors (decomposition for rhombus and non–rhombus have 2 prime factors in common). This reduces the number of possibilities so drastically that the reduction rules and the order in which they should be applied could be made explicit in a compact table. After applying this reduction scheme, the elements in the decomposition have to be transformed to compensate for the initial transformation. Again, the appropriate transformation is found by a simple table look–up.

Besides presenting a very elegant algorithm for decomposing a binary convex structuring element into a minimum number of 3 × 3 elements, we have proved a remarkable property. Up until now it was believed that convex structuring elements had many possible decompositions into 13 prime factors. We have shown that non–rhombus convex structuring elements, when transformed into the appropriate orientation, are decomposable into only 6 structuring elements and that this decomposition is even unique. This is a great contribution to the analysis and understanding of convex images.