2D Thinning Algorithms with Revised Endpixel Preservation

Gábor Németh(✉), Péter Kardos, and Kálmán Palágyi

Institute of Informatics, University of Szeged, Szeged, Hungary
{gnemeth,pkardos,palagyi}@inf.u-szeged.hu

Abstract. Skeletons are shape descriptors that summarize the general forms of objects. Thinning is a frequently applied technique for digital binary pictures to extract skeleton-like shape features. Most of the existing thinning algorithms preserve endpixels that provide relevant geometrical information relative to the shape of the objects. The drawback of this approach is that it may produce numerous unwanted side branches. In this paper we propose a novel strategy to overcome this problem. We present a thinning strategy, where some endpixels can be deleted.

Keywords: Shape represantation · Thinning · Endpixel revision

1 Introduction

Skeletons are region-based shape descriptors that summarize the general forms of objects. Skeletonization techniques are widely applied for shape representation, geometric and topological analysis, pattern recognition, and computer vision [10,11]. Thinning as an iterative object reduction is capable of producing both 2D skeleton-like shape features (i.e., topological kernels and centerlines) in a topology preserving way [3,10,11].

In each iteration step of a thinning process, some certain boundary pixels which do not hold relevant geometric information are deleted, and it is repeated until stability is reached. Some thinning algorithms aim to preserve the shape of objects by retaining so-called endpixels that provide relevant geometric information. Unfortunately, some extremities that appear in the current object boundary are endpixels and their preservation may lead to producing of numerous unwanted side branches. As a solution Németh et al. proposed a thinning strategy with iteration-level smoothing [6,7]. Furthermore, Bertrand and Aktouf [2] also proposed some new geometric constraints called *isthmus*es that yield less unwanted side branches.

In this work we propose a novel approach for revising endpixels. Moreover, we will show that this concept can be applied in any conventional thinning algorithms as well.

A. Campilho and M. Kamel (Eds.): ICIAR 2014, Part I, LNCS 8814, pp. 65–72, 2014.
DOI: 10.1007/978-3-319-11758-4_8

2 Basic Notions and Related Results

In this section we use the fundamental concepts of digital topology as reviewed by Kong and Rosenfeld [5].

We consider (8,4) *binary pictures* [5], where black pixels form 8-connected *objects*, and 4-connectivity is used for white pixels. It is assumed that the pictures to be thinned contain finitely many black pixels, hence we can store these pictures in finite arrays.

A black pixel is called a *border pixel* if it is 4-adjacent to at least one white pixel. A black pixel is said to be an *interior pixel* if it is not a border pixel.

A reduction can delete some black pixels (i.e., changes some black pixels to white ones). Parallel reductions may delete a set of pixels at a time. A reduction in 2D is not *topology preserving* if it disconnects or completely eliminates any black component, creates or merges white components [5]. A black pixel is a *simple pixel* if its deletion is a topology preserving reduction. From various characterizations of simple pixels, we recall the following one:

Theorem 1. [5] *Black pixel p is simple in an $(8,4)$ picture, if p is a border pixel, and the black pixels that are 8-adjacent to p form exactly one 8-component.*

The simplicity of a pixel in $(8,4)$ pictures is a local property, therefore it can be decided by investigating its 3×3 neighborhood. Although deletion of an individual simple pixel is a topology preserving reduction, simultaneous deletion of a set of simple pixels may alter the topology. Various sufficient conditions have been proposed for topology preserving parallel reductions [4,6,9]. In this paper we use the following one:

Theorem 2. [4] *A parallel reduction operation is topology preserving for $(8,4)$ pictures if all of the following conditions hold:*

1. *Only simple pixels are deleted.*
2. *For any two 4-adjacent pixels p and q that are deleted, p is simple after q is deleted, or q is simple after the deletion of p.*
3. *No object contained in a 2×2 square is deleted completely.*

3 Thinning Algorithms with Revised Endpixels

Parallel thinning algorithms are composed of successive parallel reductions [3, 11]. *Endpixels* are simple pixels that hold some relevant geometrical information respect to the shape of objects [3]. The considered type of endpixels are preserved during the entire thinning process. Let us consider a phase of a thinning process (i.e., a parallel reduction) with deletion rule $\mathcal{R}$, see Alg. 1. We assume that deletion rule $\mathcal{R}$ fulfills all conditions of Theorem 2 (i.e., the reduction associated with $\mathcal{R}$ is topology preserving).

Now we introduce the *revised reduction* strategy. According to this concept the endpixel preservation is not determined at the moment of its detection.

Algorithm 1. Conventional Parallel Reduction

```
1: Input: A    // the array that stores the picture to be thinned
          R    // deletion rule
          ε    // the considered type of endpixels
2: Output: B   // the array that stores the resulted picture
3: B ← A;
4: for each pixel (x, y) do
5:   if A[x, y] is deletable by R and it is not an endpixel of type ε then
6:     B[x, y] ← WHITE;
```

Endpixel deletion and further restoration (i.e., turning back a white pixel to black one) are also allowed. The general scheme of the revised reduction is sketched in Alg. 2, where a topology preserving deletion rule $\mathcal{R}$ is applied. Note that in this case we use labeled image arrays, where cells marked as BLACK, ENDPIXEL, DELEND1, DELEND2, RESTORED, and WHITE, respectively, means object pixels, endpixels, endpixels deleted in Step 1, endpixels deleted in Step 2, restored pixels, and white pixels. Pixels marked BLACK, ENDPIXEL, and RESTORED are black pixels, while the others are white.

Here we explain the three-step process of revised reduction:

Step 1 – Endpixel deletion: Those border pixels that fulfill the endpixel characterization of type ε are labeled as ENDPIXEL. All endpixels of type ε that satisfy the deletion rule $\mathcal{R}$ are deleted simultaneously. Deleted endpixels are labeled as DELEND1, and preserved endpixels ε are labeled as ENDPIXEL.

Step 2 – Shrinking: Border pixels that fulfill the endpixel characterization of type ε are labeled as ENDPIXEL. We apply a parallel reduction with deletion rule $\mathcal{R}$ again. Those pixels that are deleted in this step and have a label of ENDPIXEL are labeled as DELEND2.

Step 3 – Restoration: Let E be the set of pixels that are black or deleted endpixels and Y be the set of black pixels in the current picture. Any pixel p deleted in Step 2 is *restorable* if it is non-simple and non-isolated in $(N_8(p) \cap (Y \cup E)) \cup \{p\}$, where $N_8(p)$ denotes the set of pixels that are 8-adjacent to p. Each restorable endpixel changes to black pixel again and marked as a restored pixel. It is composed of the following steps:

a) Each restorable endpixel deleted in Step 2 is restored. Each restorable pixel p marked as DELEND2 is labeled as RESTORED.
b) Each restorable endpixel deleted in Step 1 is restored. Each restorable pixel p marked as DELEND1 is labeled as RESTORED.

For the efficient implementation of the algorithm, we propose to store labeled pixels in a linked list rather than in a temporary array T (see Alg. 2), similarly as described in [8]. In order to avoid repeated scannings of the picture array, we can use two linked lists: first we collect the border pixels to a linked list, and this list is updated during Steps 1 and 2. The second list is a LIFO data structure which is used for the deleted endpixels. It is necessary, since in Step 3 we try to restore

Algorithm 2. Revised Parallel Reduction

```
1: Input: A    // the array that stores a labeled picture
          R    // deletion rule
          ε    // the considered type of endpixels
2: Output: B    // the array that stores the resulted picture
3: // —   Step 1: Endpixel reduction   —
4: T ← A;   // T is a temporary array
5: for each pixel p = (x, y) do
6:    if (A[x, y] = BLACK) and (p is an endpixel of type ε) then
7:       T[x, y] ← ENDPIXEL;   // p is marked as an endpixel
8:       if p is deletable by R in picture stored in A then
9:          T[x, y] ← DELEND1;   // p is marked as a deleted endpixel
10: B ← T;
11: // —   Step 2: Shrinking   —
12: for each pixel p = (x, y) do
13:    if (B[x, y] = BLACK or B[x, y] = ENDPIXEL) and (p is deletable by R) in
       picture stored in B then
14:       if p is an endpixel of type ε then
15:          T[x, y] ← DELEND2;   // p is marked as a deleted endpixel
16:       else
17:          T[x, y] ← WHITE;   // p is deleted
18: B ← T;
19: // —   Step 3(a): Restoration   —
20: for each pixel p = (x, y) do
21:    if (B[x, y] = DELEND2) and (p is a restorable pixel) then
22:       T[x, y] ← RESTORED;   // p is restored
23: // —   Step 3(b): Restoration   —
24: for each pixel p = (x, y) do
25:    if (B[x, y] = DELEND1) and (p is a restorable pixel) then
26:       T[x, y] ← RESTORED;   // p is restored
27: B ← T;
28: return B;
```

endpixels deleted in Step 2, then we continue the restoration with endpixels deleted in Step 1. The deletable pixel configurations and endpixel configurations can be stored in two precalculated look-up tables making the repeatable checking more efficient. Labels in arrays also increase the efficiency.

Let us see how the revised reduction works in an example depicted in Fig. 1. Here we present the same deletion rule and endpixel characterization with and without the revised endpixel strategy. Revised reduction works with any type of endpixels, but in this example we use the following characterization: a black pixel is an endpixel if it is 8-adjacent to at most two black pixels that are 4-adjacent to each other. Here we assume a deletion rule $\mathcal{R}$ that satisfies all conditons of Theorem 2.

Now we will show that revised reduction strategy is topology preserving.

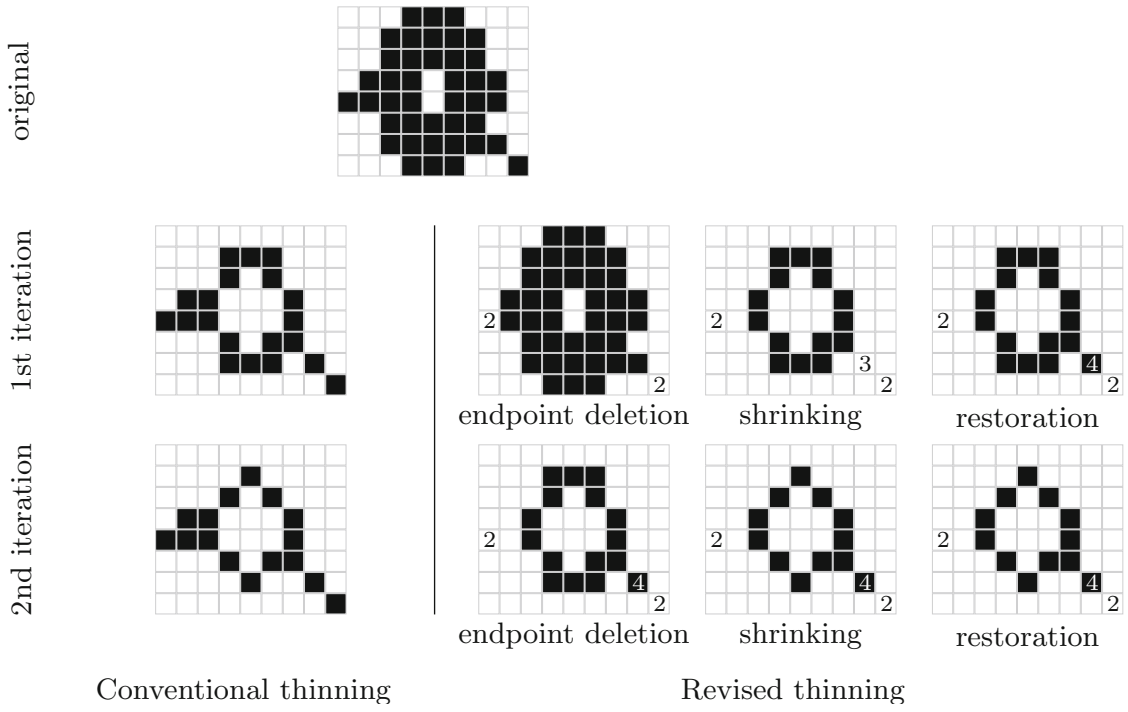

Fig. 1. Example of conventional fully parallel thinning (left) and revised fully parallel thinning (right). Labels "2","3", and "4" indicate deleted endpixels in Step 1 (i.e., DELEND1), deleted endpixels in Step 2 (i.e., DELEND2), and restored endpixels (i.e., RESTORED), respectively. Note that label "1" (i.e., ENDPIXEL) does not appear in this example, since each detected endpixel is deletable by the considered deletion rule $\mathcal{R}$ and their labels are changes to "2" in Step 1. Here an endpixel is considered as a black pixel being 8-adjacent to at most two 4-adjacent black pixels. The deletion rule $\mathcal{R}$ fulfills each condition of Therorem 2.

Theorem 3. *If the reduction with deletion rule $\mathcal{R}$ (see Alg. 1) is topology preserving, and endpixels of type ε are simple, then Alg. 2 also specifies a topology preserving reduction.*

Proof. We will show that three steps of revised reduction strategy (i.e., endpixel deletion, shrinking, and restoration) is topology preserving.

1. *Endpixel deletion:* Let us consider a set of endpixels that satisfy the conditions of deletion rule $\mathcal{R}$. Since deletion rule $\mathcal{R}$ fulfills the conditions of Theorem 2, deletion of any set of endpixels is topology preserving.
2. *Shrinking:* This phase is topology preserving, since $\mathcal{R}$ is assumed as a topology preserving reduction.
3. *Restoration:* We restore a set of deleted endpixels within two steps. The proof of topological correctness is similar in both cases. Each deleted endpixel p is deleted by a topology preserving deletion rule $\mathcal{R}$, hence any deleted pixel fulfills the conditions of Theorem 2. Let E be the set of detected endpixels and Y be the set of black pixels in the current picure. A pixel p is restored if it is non-simple and non-isolated in $(N_8(p) \cap (Y \cup E)) \cup \{p\}$ in the current

picture. According to this condition the restored pixel p does not form any new black component, since it can not be an isolated black pixel in $(N_8(p) \cap Y) \cup \{p\}$. On the other hand, each restored pixel was simple and border pixel at the moment of its deletion, hence restoration of any pixel does not merge black components, does not fill any cavity, or split any white component. Consequently, restoration of any restored pixel is a topology preserving addition.

Further advantageous property of the novel strategy is that the remaining skeletal branches are important indeed, since the environment of the restored endpixels contains some other detected endpixels (but they are already deleted and not restored). These properties are illustrated in Figs. 1-4.

Here we give a general scheme to convert any parallel thinning algorithm to its revised alternative. In some thinning algorithms, the considered type of endpixel is given explicitly [6], but usually only the deletion conditions are given and endpixel conditions stay hidden. If the cosidered type of endpixels ε is preserved by the given reduction as in Alg. 1, then Step 1 and Step 2 can be performed with no changes. However, if only the deletion condition are given, then we can express the endpixel conditions from non-deletable ones. Here we suppose that all endpixels are simple pixels. Those pixels that are simple but non-deletable by deletion rule $\mathcal{R}$ are considered as endpixels.

4 Results

Due to the lack of space here we present only three examples for applying the revised reduction strategy, but in a website[1] more results for various algorithms are presented. We implemented the often referred topology preserving parallel thinning algorithm proposed by Bernard and Manzanera (denoted by BM99) [1] and its revised alternatives (denoted by R-BM99). This algorithm does not define any endpixel characterization directly, hence the rules described at the end of the previous section are used for conversion. Results are depicted in Figs. 2-4.

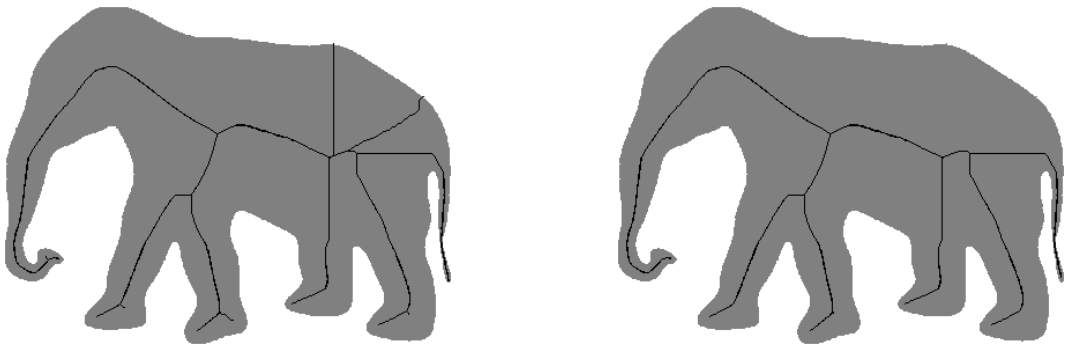

Fig. 2. Centerlines produced by BM99 (left) and R-BM99 (right) superimposed on the 400×305 picture of an elephant

[1] http://www.inf.u-szeged.hu/~gnemeth/kutatas/revisedthinning2d/

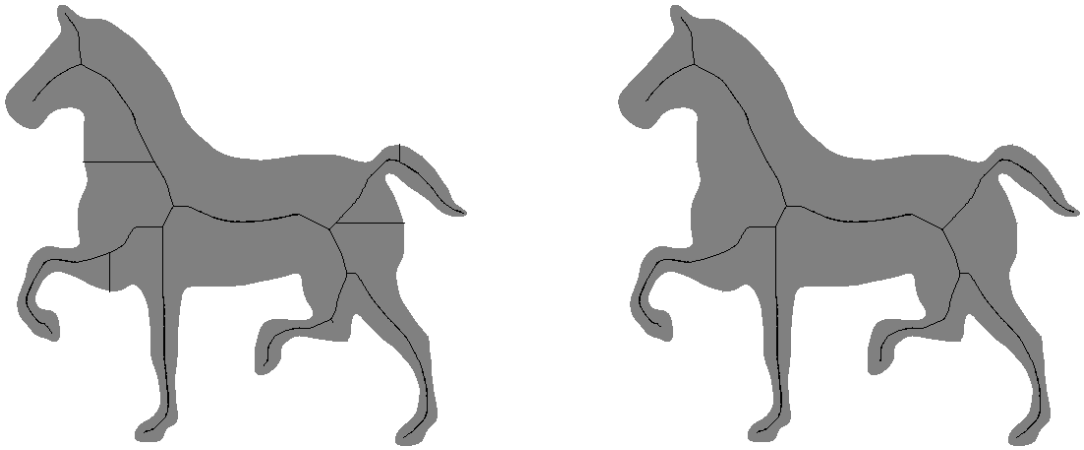

Fig. 3. Centerlines produced by BM99 (left) and R-BM99 (right) superimposed on the 470×448 picture of a horse

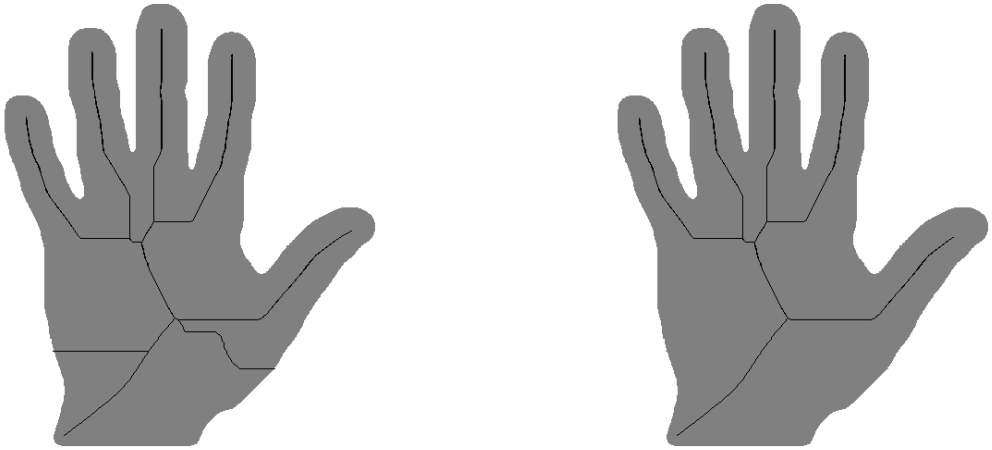

Fig. 4. Centerlines produced by BM99 (left) and R-BM99 (right) superimposed on the 500×560 picture of a hand

5 Conclusions

In this paper we propose a new strategy for thinning algorithms. Conventional thinning algorithms are composed of iterative object reductions where a set of pixels from the object boundary is deleted in each reduction phase. Some algorithms apply endpixel preservation as a geometric constraint. Since they may not delete the detected endpixels, some unwanted side branches can be produced. The new thinning strategy allows us to delete endpixels. Restored endpixels hold significant geometrical information respect to the shape, since some other endpixels are also detected in their neighborhood (but some of them are not restored). Furthermore, we gave a general scheme to apply revised reduction strategy in any conventional thinning algorithm. Thanks to the proposed strategy, thinning algorithms become less sensitive to boundary noise.

Acknowledgments. This work was supported by European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4. A/2-11-1-2012-0001 'National Excellence Program'.

References

1. Bernard, T., Manzanera, A.: Improved low complexity fully parallel thinning algorithm. In: 10th International Conference on Image Analysis and Processing (ICIP 1999), pp. 215–220 (1999)
2. Bertrand, G., Aktouf, Z.: A three-dimensional thinning algorithm using subfields. In: Vision Geometry III, vol. 2356, pp. 113–124. SPIE (1994)
3. Hall, R.: Parallel connectivity-preserving thinning algorithms. In: Kong, T.Y., Rosenfeld, A. (eds.) Topological Algorithms for Digital Image Processing, pp. 145–179. Elsevier Science B. V. (1996)
4. Kong, T.Y.: On topology preservation in 2-d and 3-d thinning. Int. Journal of Pattern Recognition and Artificial Intelligence **9**, 813–844 (1995)
5. Kong, T.Y., Rosenfeld, A.: Digital topology: Introduction and survey. Computer Vision, Graphics, and Image Processing **48**, 357–393 (1989)
6. Németh, G., Kardos, P., Palágyi, K.: 2D parallel thinning and shrinking based on sufficient conditions for topology preservation. Acta Cybernetica **20**, 125–144 (2011)
7. Németh, G., Kardos, P., Palágyi, K.: Thinning combined with iteration-by-iteration smoothing for 3D binary images. Graphical Models **73**, 335–345 (2011)
8. Palágyi, K., Németh, G., Kardos, P.: Topology preserving parallel 3D thinning algorithms. In: Brimkov, V.E., Barneva, R.P. (eds.) Digital Geometry Algorithms. Lecture Notes in Computational Vision and Biomechanics, vol. 2, ch. 6, pp. 165–188. Springer (2012)
9. Ronse, C.: Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images. Discrete Applied Mathematics **21**, 67–79 (1988)
10. Siddiqi, K., Pizer, S.M. (eds.): Medial Representations - Mathematics, Algorithms, and Applications. Series in Computational Imaging. Springer (2008)
11. Suen, C.Y., Wang, P.S.P. (eds.): Thinning Methodologies for Pattern Recognition. Series in Machine Perception and Artificial Intelligence, vol. 8. World Scientific (1994)