This is an author produced version of An improved rotation-invariant thinning algorithm.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/785/

Article:

http://dx.doi.org/10.1109/TPAMI.2005.191

promoting access to
White Rose research papers

eprints@whiterose.ac.uk
http://eprints.whiterose.ac.uk/
An Improved Rotation-Invariant Thinning Algorithm

Peter I. Rockett

Abstract—Ahmed and Ward [2] have recently presented an elegant, rule-based rotation-invariant thinning algorithm to produce a single-pixel wide skeleton from a binary image. We show examples where this algorithm fails on two-pixel wide lines and propose a modified method which corrects this shortcoming based on graph connectivity.

Index Terms—Thinning, skeletonization, graph theory.

1 INTRODUCTION

The thinning (or skeletonization) of segmented binary images is a much-used and well-studied topic in image processing and related fields. In a frequently cited review published in 1995, Lam and Suen [1] reviewed around 100 thinning algorithms and significant numbers of new algorithms have been proposed in the intervening years. Of recent note is the rotation-invariant thinning algorithm of Ahmed and Ward [2] which used the particularly elegant device of deriving a set of predicates over the 8-neighbors of a given pixel to determine if that pixel is on the boundary of a shape and can thus be deleted. As part of our present work, we have used the Ahmed-Ward (A-W) thinning algorithm to extract the center lines of arteries segmented from X-ray angiograms as a prelude to further processing. These vascular trees are significantly more complex than any of the characters examined by Ahmed and Ward and, in the course of our work, we have observed a number of cases where the A-W algorithm fails to produce a center line of single pixel width.

In Section 2, we describe a modified method which corrects this shortcoming based on graph connectivity. In Section 3, we present some results. Finally, we offer some remarks and conclusions in Section 4.

2 DESCRIPTION OF THE MODIFIED ALGORITHM

Our algorithm comprises two stages: First, we apply the 20 rules of Ahmed and Ward over the 8-neighbors of each pixel, in turn, to determine if that pixel is on the boundary of the shape to be thinned and hence can be deleted. Like the A-W algorithm, this first stage is applied iteratively where the pixels are marked for deletion if they are adjudged to be on the shape’s boundary and then all marked pixels are deleted at the end of an iterative pass. Again, like the A-W algorithm, we skip pixels which are found to be at the extrema of diagonal lines. Most importantly, any pixels which are found to be part of two-pixel wide vertical or horizontal lines, that is, which fit any of the rules: [0 0 0 1 0], [0 1 0 1 0], [0 0 1 0 0], or [0 1 0 0 1] are skipped—two-pixel wide lines are processed in the second stage. Similar to the A-W algorithm, the first stage progressively removes pixels from the boundary of a shape until the shape has been eroded to a skeleton of mostly single pixel width.

The second processing stage takes the provisional skeleton from the first processing stage—where we know all the pixels, by definition, boundary pixels—and examines every pixel which makes up part of a two-pixel wide line. If deletion of such a pixel does not disrupt the connectivity of the skeleton, we remove it immediately from the skeleton in a single pass. Thus, at the end of this single pass, the skeleton comprises only single-pixel wide segments. If any two pixel wide blocks remain, these cannot be deleted without producing a disconnected skeleton.

In order to efficiently determine if deletion of a pixel in a 2-pixel wide line will disrupt connectivity, we build an undirected graph of the local pixel connectivity over its eight neighbors. The process is illustrated by the example in Fig. 3 and Table 2, where the pixel labeling convention we have used is shown in Fig. 2.

Fig. 3a shows an initial pixel configuration in the provisional skeleton produced by the first thinning stage. Note that, for this example, we assume that the central two pixels form part of a two-pixel wide vertical line as implied by the zero pixel shown dotted to the left. Fig. 3b shows the graph representation of the connectivity in the 3 × 3 region for the nonzero pixels of Fig. 3a where arcs represent physical adjacency of the (nonzero) pixels. For example, x0 can only connect to x2, x3, x7, and x8, assuming that the pixels at both ends of an arc are “1.” In other words, an arc is only allowable if it is possible to pass from one pixel to the other without passing through a third pixel. A full list of allowable arcs is shown in Table 1.

From Fig. 3b, it is readily apparent that the central pixel in Fig. 3a (vertex x0) can be safely deleted without breaking the connectivity of the skeleton since x0 can be removed from Fig. 3b leaving a subgraph where every vertex has at least one arc connected to it. In order to implement this notion, we construct an adjacency matrix for the connectivity graph and examine the scenario where the central pixel is deleted. In fact, it is sufficient (and faster) to construct only the
adjacency matrix for the subgraph resulting from excluding the central pixel. The adjacency matrix for the corresponding subgraph of the graph in Fig. 3b is shown in Table 2. From this table, it is clear that in order for the deletion of the central pixel not to create a disconnected skeleton, every row (or column) of the adjacency matrix must contain at least one nonzero entry. The speed of searching the adjacency matrix can be improved by an “early jump-out” approach: When searching a row, as soon as the first nonzero entry is encountered, we can move on to searching the next row since the presence of a single “1” is enough to guarantee connectivity (for that row). Similarly, as soon as we find the first row that contains only zero entries, we can terminate the search since a single empty row tells us we cannot delete the central pixel under consideration.

By way of counterexample, Fig. 4 shows a pixel configuration in the first-stage skeleton where the central pixel cannot be removed as evidenced by the fact that vertex $x_2$ will become disconnected if vertex $x_0$ is deleted. Deducing this conclusion from the corresponding adjacency matrix is trivial.

In practice, constructing and searching the adjacency matrix is fast and efficient. Since we treat a $3 \times 3$ image patch and the central pixel and at least one other pixel are set (in order to constitute a two pixel wide line), the adjacency matrix has to consider the connectivity of only the remaining seven pixels in the $3 \times 3$ patch. Since, at this stage of the algorithm, we are dealing with exterior pixels, strictly less than seven pixels can ever be set. As a consequence, the adjacency matrix is strictly smaller than $7 \times 7$ and, typically, much smaller than even this. (In fact, we show below that the algorithm described here can be faster than the A-W algorithm.)

### 3 Results

Fig. 5 shows the thinning results for the portion of X-ray angiogram image shown in Fig. 1 with the algorithm presented here and for which the A-W algorithm fails. Note that the algorithm described here does indeed produce a single-pixel wide skeleton although the overall skeletonization is (unsurprisingly) slightly different.

Figs. 6 and 7 contain two more examples of pixel configurations taken from X-ray angiograms for which the A-W algorithm fails to produce a single-pixel wide skeleton. Both Figs. 6a and 7a show the results of the A-W algorithms and in Figs. 6b and 7b, the results obtained here. Again, the skeleton obtained from the new algorithm is of the desired single-pixel width.

In addition to the examples taken from the complex vascular trees obtained from X-ray angiograms, Figs. 8 and 9 compare the A-W and present algorithms for the task of skeletonizing two
Chinese characters taken from Lin and Chen [3] and which have also been used by Ahmed and Ward [2]. Although the both algorithms yield acceptable, single-pixel wide skeletons for these characters, the skeletons obtained are slightly different in nature. Since there is no established method for objectively comparing thinning algorithms, we confine ourselves to subjective observations based on the examples of the angiogram images and the Chinese characters in Figs. 8 and 9.

First, whereas the A-W algorithm often tends to skeletonize diagonal lines with a "staircase" structure comprising two horizontal pixels followed by two vertical pixels, the modified algorithm presented here tends to produce diagonal runs of pixels connected NW-to-SE (or NE-to-SW). Thus, the new algorithm arguably achieves a greater degree of thinning in that the resulting skeletons are more generally made-up of single-pixels rather than "staircases" of two-pixel long "risers" and "treads."

Second, the A-W method appears to be more aggressive in eroding the ends of lines than the new algorithm; this is particularly evident from the Chinese character results. Whether this is an advantage or not probably depends on the end-application for the skeleton. Certainly, for our work on X-ray angiograms, the present thinning algorithm is preferable since we are interested, among other things, in identifying the end points of terminal capillaries in arterial networks.

The numbers of pixels comprising the final skeletons of the Chinese characters for each algorithm are shown in Table 3. There appears to be no great difference in the overall numbers although the new algorithm tends to use fewer pixels in the interior of skeleton segments and rather more at the ends of strokes.

We have also examined the operation of our modified algorithm on the sequence of rotated symbols used by Ahmed and Ward [2, Fig. 3b]. Since the basis of our modified algorithm is the A-W rule set, the modified algorithm produce results which differ only in its treatment of diagonal lines. These differences can be conveniently summarized by the results of thinning the triangular shape (from the third column in A-W’s Fig. 3b) and are shown in Fig. 10.

The original A-W algorithm thins diagonal segments down to the "staircase" structures described above whereas our algorithm makes greater use of diagonal connectivities to produce a smoother skeleton. The difference between Figs. 10a and 10b are shown in white. This difference illustrates that the modified algorithm tends to use fewer pixels in generating a skeleton from a diagonal segment. Similarly, the two pixels which are present in Fig. 10b, but absent from Fig. 10a are arrowed; the fact that these two pixels are both next to pixels used by the A-W algorithm indicates that the modified algorithm is simply making an alternative choice of skeleton pixel in these cases.

As to the relative execution times, we have compared the average run times over 10 executions for a 928 × 342 X-ray angiogram image and the new algorithm runs ~ 6 percent faster (234mS versus 219mS) since there are fewer rules to be evaluated in the iterative phase of the
new algorithm. By contrast, when comparing the execution times of both algorithms on Ahmed and Ward’s Fig. 3, the algorithm presented here ran ~18 percent slower than the A-W algorithm due to the large number of diagonal segments present in this image and, hence, the extensive application of the graph-based thinning stage. For yet other images, the execution times of the two algorithms were indistinguishable. In general, therefore, it appears that the comparative execution times are similar but detailed differences depend on the particular image under consideration. Both algorithms required the same number of iterations of applying the thinning rule set.

4 DISCUSSIONS AND CONCLUSIONS

The principal contribution of this work is to remedy a deficiency in the thinning algorithm of Ahmed and Ward [2] by modifying the way in which lines of two-pixels width are handled. We have shown that a few pathological configurations exist for which the A-W algorithm produces results which are qualitatively different from the A-W algorithm although the claim of rotation invariance needs to be treated somewhat carefully. For a two-pixel wide line, the skeleton is considered as running between the two pixels and which of the two is deleted to yield a single-pixel wide center line is completely arbitrary. Ahmed and Ward chose the bottom-most pixel in a horizontal 2-pixel line and the right-most in a vertical 2-pixel line. Here, we tend to delete the top-most and left-most pixels although this choice is implementation-dependent and determined by the scan order (from top-left) in the second stage of the algorithm; this could trivially be reversed to follow the same choice as Ahmed and Ward by scanning from the bottom-right. Nonetheless, since the choice of which pixel to delete from a 2-pixel wide line is arbitrary and implementation-dependent, no thinning algorithm can be truly rotation invariant. To select a consistent pixel to delete, independent of rotation would require recognition of the shape’s pose and, therefore, recognition of the shape. But as one of the main uses of skeletonization is recognition, using knowledge of the shape’s pose to guide the thinning process is, in most cases, a paradox.

ACKNOWLEDGMENTS

This work was supported by the UK Joint BBSRC/EPSRC/MRC Discipline Hopping Programme under Grant No. GR0300594. The author is also grateful to Maher Ahmed and Rabab Ward for kindly providing copies of the images used in their paper for comparison purposes.

REFERENCES