A note on the Nagendraprasad-Wang-Gupta thinning algorithm

Rafael C. Carrasco and Mikel L. Forcada
Departamento de Tecnología Informática y Computación
Universidad de Alicante, E-03071 Alicante, Spain

Abstract

A symmetrized version of the Nagendraprasad-Wang-Gupta thinning algorithm (Digital Signal Processing 3(1993)97) is presented, which produces simpler and more elegant skeletons of handwritten characters at zero extra computational cost.

Keywords: Pattern recognition, skeleton, parallel thinning, image processing.

Pattern recognition often involves data where a great amount of redundant information hides the relevant details. For instance, handwritten characters show usually strong variations in thickness while only direction, curvature and length of the curves are significant. This superfluous information makes the recognition task much more difficult. Indeed, due to the enormous variability of handwritten symbols, time is a basic constraint when developing algorithms for its recognition. That is why a preprocess eliminating —even if partially— redundant variability could allow for the identification of characters at a reasonable velocity. To this respect, thinning algorithms may play an important role and many attempts to find suitable procedures have been made.

Recently, a very fast thinning algorithm (NWG) has been proposed by Nagendraprasad, Wang and Gupta (1993), based on a previous one by Wang and Zhang (1989). Both algorithms are equivalent, in the sense that they produce the same skeletons as output. They also preserve connectivity, a desirable property when dealing with handwritten characters. However, the last one is significantly faster and easier to program. It also allows for a simple parallel implementation, because at every iteration the value of a pixel depends only on the value of the pixel and its neighbours at the previous iteration. The algorithm uses masks in order to select pixels to be turned off. The 8 closest neighbours are numbered following a clockwise walk around the pixel p, which starts at the upper edge as shown in fig. 1.

$p(7)$	$p(0)$	$p(1)$
$p(6)$	p	$p(2)$
$p(5)$	$p(4)$	$p(3)$

Figure 1: Numbering of neighbouring pixels.

The NWG algorithm is shown in fig. 2, where $b(p)$ is the number of neighbours of p which are on (pixels with value 1), $a(p)$ is the number of off-to-on transitions when the neighbours are visited following a walk around p and the functions $c(p)$, $e(p)$ and $f(p)$ are given by:

$$c(p) = \begin{cases} 1 & \text{if } p(0)=p(1)=p(2)=p(5)=0 \text{ and } p(4)=p(6)=1 \\ 1 & \text{if } p(2)=p(3)=p(4)=p(7)=0 \text{ and } p(6)=p(0)=1 \\ 0 & \text{otherwise} \end{cases}$$

$$e(p) = (p(2)+p(4)) * p(0) * p(6)$$

$$f(p) = (p(6)+p(0)) * p(4) * p(2)$$

The performance of the NWG algorithm is shown in fig. 3. As plotted there, the skeleton obtained is connected and eliminates almost any redundant pixels which are not relevant in order to recognize the handwritten symbol. However, due to an asymmetry in the algorithm, some superfluous pixels are not removed, as seen in fig. 4. This may be corrected by simply changing the function $c(p)$ in those iterations where $g = 1$ (this means odd iterations). This means that the condition

$$a(p) = 1 \texttt{ or } \; c(p) = 1$$

```
algorithm NWG
input : Q (pixelmap)
output : Q
g = 1;h = 1; Q' = Q; (initial settings)
do (while h = 1)
   h = 0;
   Q = Q';
   g = 1 − g;
   for (every pixel p ∈ Q)
       if (1 < b(p) < 7 and (a(p) = 1 or  c(p) = 1)) then
          if(g=0 and e(p)=0)then
            erase p in Q'
          endif
          if (g=1 and f(p)=0)then
            erase p in Q'
          endif
       endif
   endfor
enddo
end NWG
```

Figure 2: Algorithm NWG for thinning

is replaced by

$$a(p) = 1 \texttt{ or } \quad (1-g) * c(p) + g * d(p) = 1$$

with

$$d(p) = \begin{cases} 1 & \text{if } p(1) = p(4) = p(5) = p(6) = 0 \text{ and } p(0) = p(2) = 1 \\ 1 & \text{if } p(0) = p(3) = p(6) = p(7) = 0 \text{ and } p(2) = p(4) = 1 \\ 0 & \text{otherwise} \end{cases}$$

which is just the mirror image — across the NE–SW diagonal — of $c(p)$ (as $e(p)$ is from $f(p)$).

The result is a more symmetric algorithm which usually produces more elegant skeletons, in the sense that some redundant or confusing pixels are removed. Figure 4 shows how our modification improves the recognition of a '0', the original algorithm produces what looks like the skeleton of an upper-case 'D'.

On the other hand, its cost is not increased, as the only difference comes from the replacement of one occurrence of the function $c(p)$ by its same-cost counterpart $d(p)$.

Summarizing, by introducing a small change in the code, which does not change the average time that the algorithm needs in order to output the result, the NWG algorithm becomes more symmetric, resulting in slightly simpler and more elegant skeletons.

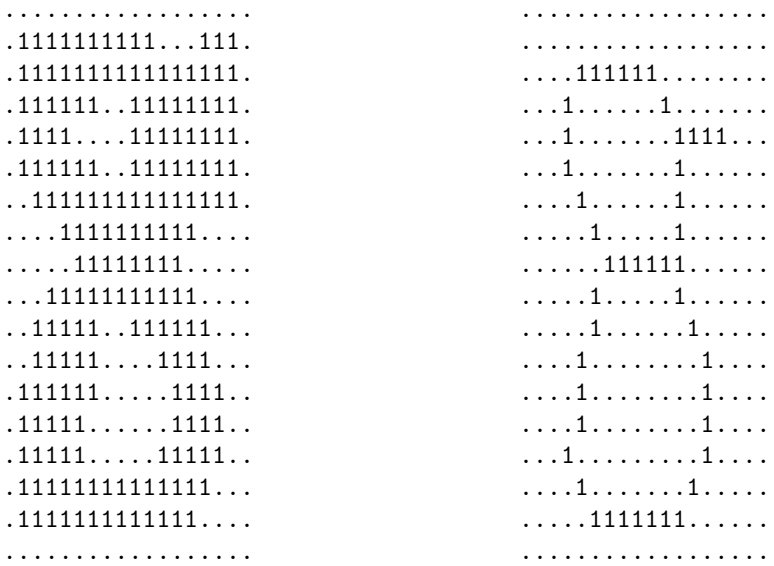

Figure 3: Example of a bitmap and its skeleton produced by the NWG thinning algorithm

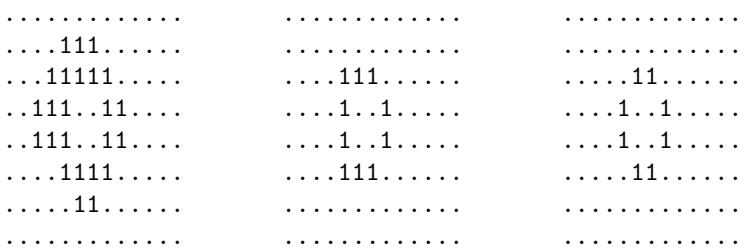

Figure 4: Handwritten symbol and its skeleton as given by the NWG algorithm and the modified version respectively

References

1. M.V. Nagendraprasad, P.S.P. Wang, and A. Gupta (1993). Algorithms for thinning and rethickening binary digital patterns. *Digital Signal Processing* 3, 97–102

2. P.S.P. Wang and Y.Y. Zhang (1989). A fast and flexible thinning algorithm. *IEEE Transactions on Computation* C-38, 741–745