Hw3-110810006林君曆

Medial axis skeletonization

Reuse the defined static function of image converter done by HW#1.

1-1. Do the 8-distance transform

Use Two-pass algorithm, which reduces the redundant iteration.

Init the distance array by infinite, then

1. scan the image from top-left to bottom-right

   Compare the distance with (top, left) or (top, left, top-left, top-right) depends on the connectivity. Then, Assign smallest distance to the point.

2. scan the image from bottom-right to top-left

   Compare the distance with (bottom, right) or (bottom, right, bottom-left, bottom-right) depends on the connectivity. Then, Assign smallest distance to the point.

1-2. Find the medial axis

1. Remove the point that are not local maximum of distance in the 8-neighbor, but also can’t break the connectivity.

   Some condition to decide keep or remove the point and implement in will_break_connectivity function

   1. assign center point to 0, transfer distance to binary information whether the point is foreground or background
   2. can remove the point which sum(point) < 2, to remove redundant branch
   3. keep the point if two side of the grid been separated
   4. keep the point if the corner point will be separated

2. Remove the redundant points under condition below

   a. If the point’s left or right is also foreground.

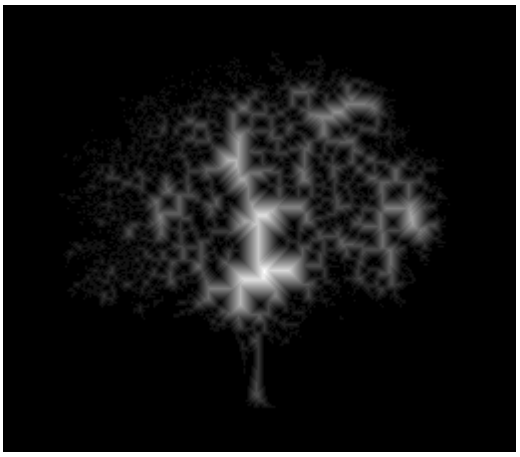
   b. Won’t break the connectivity

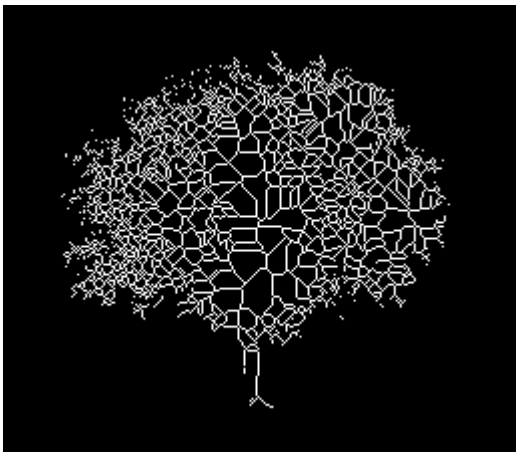
Result images

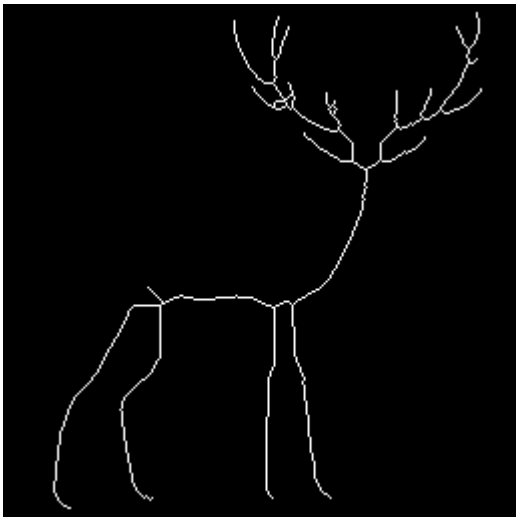
The binary image is under threshold=128

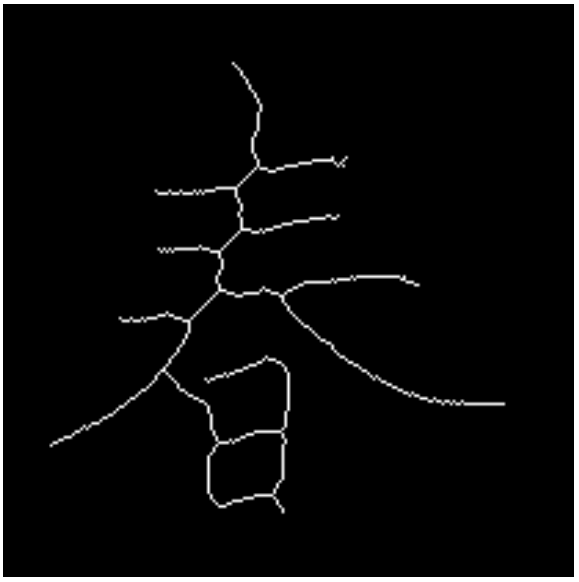
The distance transform images is 8-distance transform.

The skeleton use the 4-distance transform, but check connectivity by 8-neighbor









The things I found

1. Different binary image(with diff threshold) has different skeletonized result
2. If read a binary image from jpg binary image file, the skeletonized results go wrong.
3. cv2.imread method return in a (b, g, r) way
4. While doing the medial axis skeletonization, the input distance should be 4-connected. Otherwise, the result will be different and weird.