Skeletonization using distance transform

This exploration of skeletonization begins with H. Blums research (H. Blum, 1967)[i] in trying to find new shape descriptors. He described an algorithm (which he named medial axis function) based on the grass-fire analogy. If we imagine the object, which we are trying to find a shape descriptor for, as a grass field, and we then proceed to ignite the edges of this field. The fire will generate wave fronts and the places where these fronts meet will create a skeleton, or as H. Blum described it, the medial axis.

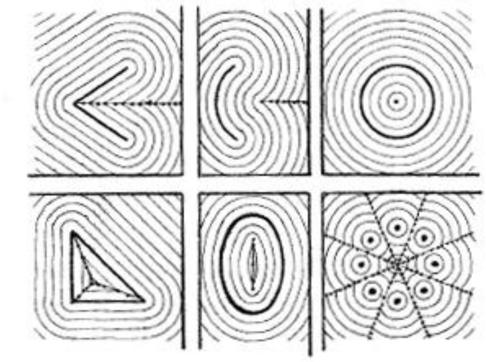


Fig. 2. Wave fronts and "corners" generated by some simple excitations (dotted lines represent locus of "corners").

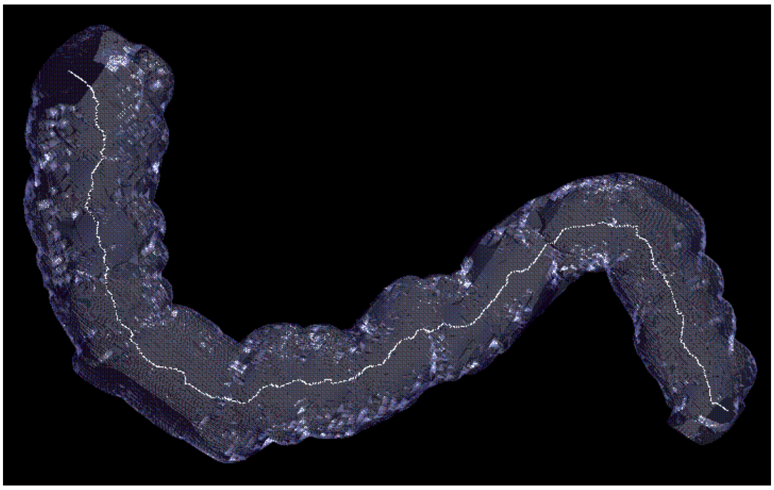
A Figure from Blum's article, depicting how the wave fronts meet, and what kind of lines that produces.

Skeletons are useful for describing properties of a shape, and in some cases it can also be used to reconstruct the original shape. Since Blum posted his article, other algorithms have been proposed with different strengths and drawbacks. Skeletonization not restricted to two dimensional images anymore, but we will not cover skeletonization in three dimensions. We will explore the algorithm which relies on distance transformations for finding the skeleton of shapes. The different algorithms does not produce the same skeletons, which shows that skeletons are not a strict defined concept. However, usually skeleton have some or all of the following properties

- Connectivity
- Topology
- Length
- Direction
- Width

Applications

Skeletons are good descriptors of shapes. Skeletons have been found used for automating shape analysis (computer vision in a factory, verifying products before shipment for example), a range of medical applications (such as assessing colons), motion analysis, and other topics.



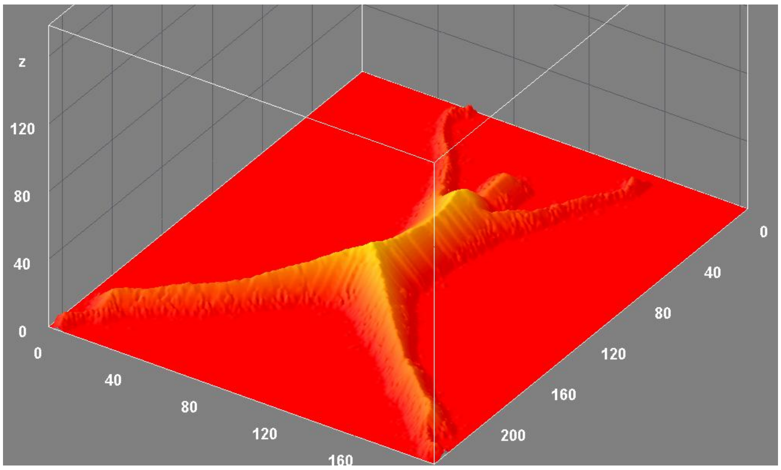
An example of skeletonization used in a medical context. A 3D image of a colon is skeletonized as a part of a more complex investigation[ii].

How to use distance transformations to find skeletons

We will now explore how distance transformations can be used to skeletonize shapes, described in binary images. Distance transformations describe the shortest distance from the edges of an object to a given point in the object. Euclidian distances can be used but any distance measure can be used, Manhattan distance for example. Throughout the rest of this report Manhattan distance will be used because of ease of implementation.

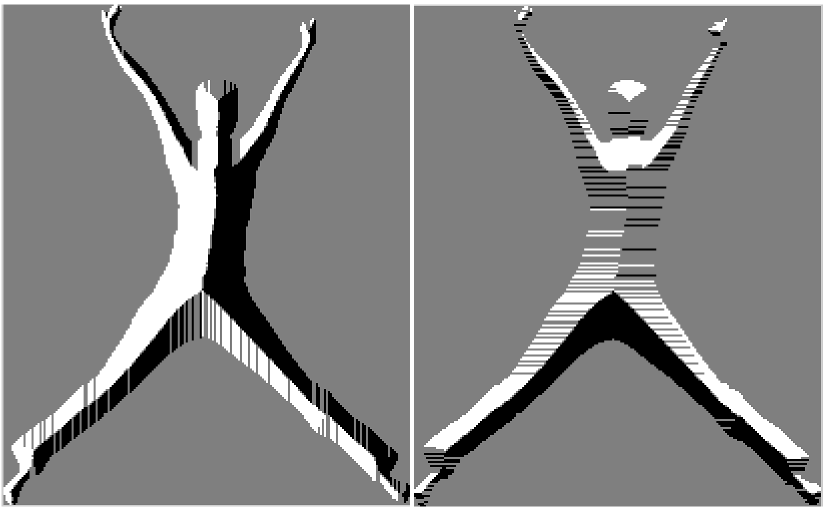
An example of a binary input image (left) and the resulting distance map using Manhattan distance (right – brightness has been increased to make map visible).

A distance map can be thought of as describing height. Following this interpretation of the distance maps, we can see that the ridges found on the hills seem to have a skeletal like structure.



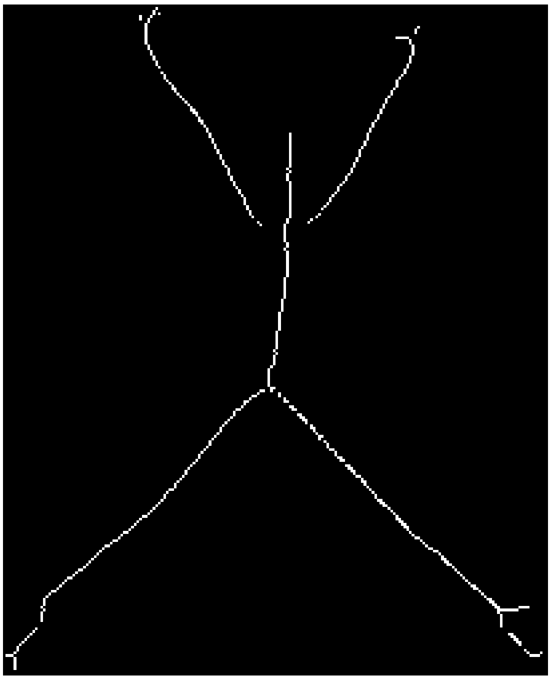
The distance map from above, augmented so the distance information is interpreted into height (colors were changed to improve readability). Notice how the ridges seem to have a skeletal appearance.

If we linger with the height interpretation we can think of the distance map describing curves, which in the local extremes describes the skeleton. There are different approaches to finding these extreme, as the most commonly known way, differentiation, is not possible, as there is no parametric description of the shape. One way of doing this is described in "Extracting Skeletons from Distance Maps" (Sukmoon, 2007)[iii] where it is suggested that the local extremes can be found by analyzing the rows and columns for different patterns. First a gradient analysis is performed. The rows in the distance map are analyzed from left to right, comparing the value of the current pixel with its neighboring pixel to the right. If the neighbor has a greater value then the analysis will return a +, if it is the same value then it will return a 0 and if the value is lesser then it will return a -. In our implementation we have chosen to visualize this by mapping + to white, 0 to grey and – to black color. The result is saved and the analysis is repeated for the distance map columns.



The image on the left is the result of the analysis of the rows, and the one on the right is the result of the analysis of the columns.

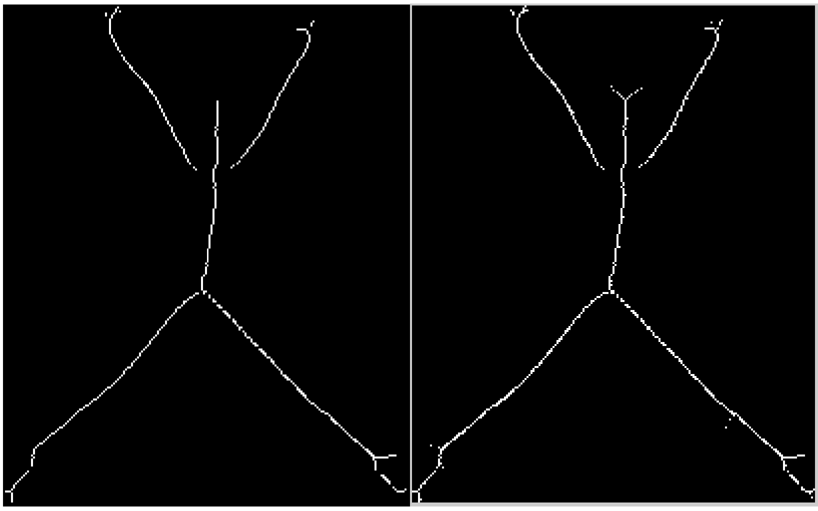
Having produced these images, we can begin to look after indicators of ridges. The most obvious indicator is a +- pattern as this only can occur on a ridge. However on a ridge, it is likely that two neighboring points have the same distance to the edges (especially if Manhattan distance is used). Therefore we will also include the +0- pattern as a strong indicator of a ridge.



Searching for +- and +0- patterns in produces the following skeleton of the original shape

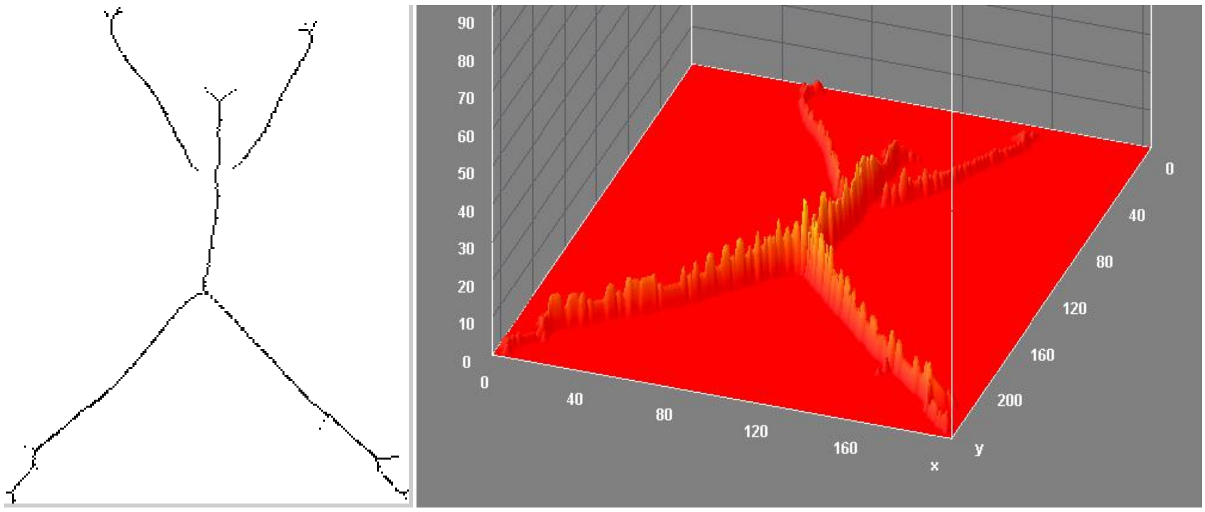
The skeleton this produces has retained a fairly good description of the shape; however we can see that there are many holes, which didn't occur in the original shape, meaning that this skeleton does not preserve information of connectivity. These holes are caused by the discrete nature of digital images. Given the holes, the topology isn't preserved either. The length is preserved, as we can tell how long the legs, upper body and arms are from the skeleton. Width is not preserved either but this topic will be discussed later.

To improve this skeleton Sukmoon[iii] suggests expanding the selection of patterns we search with +0 and 0- patterns, as these can be considered weak indicators of ridges. A +0 pattern can be the beginning of a series of 0's which may be a ridge, and a 0- may be the ending of such a series. However, these are not sure indicators of ridges; we will require there is a weak indicator in both directions before we accept is as a ridge.



Before adding the weak indicators to the pattern searching created the image on the left, while expanding the pattern searching creates the skeleton on the right.

Expanding the search did not fix the connectivity. The most notable change is added detail in the head area, as well as more noise at the end of the arms and legs. To clean up the skeleton Sukmoon[iii] suggests using graph representation. We will not explore this topic here. Instead we will examine if we can incorporate width information into the skeleton we found earlier. We propose using the skeleton as an alpha map on the distance map. This should produce an image with the points from the skeleton having the values from the distance map. This way the width of the original shape is preserved, and we will call this image the distance-skeleton.



The distance-skeleton created from the skeleton and the distance map. To convince the reader that it just isn't an inverted image of the skeleton, a 3d rendering, depicting the values on the distance-skeleton as height has been produced.

The interesting point in preserving the width is that we believe it is possible to reconstruct the original shape, once that the width information is preserved[1]. Considering we have used Manhattan distance throughout this exploration it should be possible to recreate the original shape by drawing pyramid shapes with “radius” corresponding to the value found on the points in the distance-skeleton. Let us convince ourselves through an example. We have a pyramid, in which the middle has a distance of four to the edges. If we let this distance pass through the different stages of analysis to find the distance-skeleton we end up with just one pixel with the value of four. Now that we know the width, we can draw a pyramid, with four as the “radius”.

Example:

Original image

0	0	0	1	0	0	0
0	0	1	2	1	0	0
0	1	2	3	2	1	0
1	2	3	4	3	2	1
0	1	2	3	2	1	0
0	0	1	2	1	0	0
0	0	0	1	0	0	0

Distance-skeleton

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	4	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

reconstructed image

0	0	0	1	0	0	0
0	0	1	2	1	0	0
0	1	2	3	2	1	0
1	2	3	4	3	2	1
0	1	2	3	2	1	0
0	0	1	2	1	0	0
0	0	0	1	0	0	0

In case another distance measure was used, another shape might have been better, for example circles would seem like the natural choice for Euclidian distance. The idea performs great in the example which has been used throughout this report.

The input (left) the reconstruction based on the distance-skeleton (middle) and the errors made by the reconstruction (right)

[1] We believe we are not the first ones to come up with this idea, but we haven’t seen other examples. We didn’t search for them either.

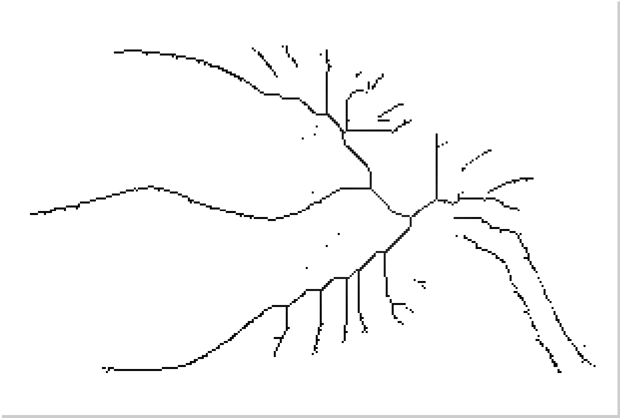
The original shape took 9994 pixels to describe, while the distance-skeleton only used 616 pixels, thereby describing the shape using only 6.2% of the pixels. The original image required a binary image, and therefore one bit per image value, while the distance-skeleton uses gray scale colors and thereby use a byte. This means that the actual compression (616*8 = 4928 pixels) is 49,3%.

Results

Here we will depict results using similar algorithm on different images.

Input

Distance-skeleton



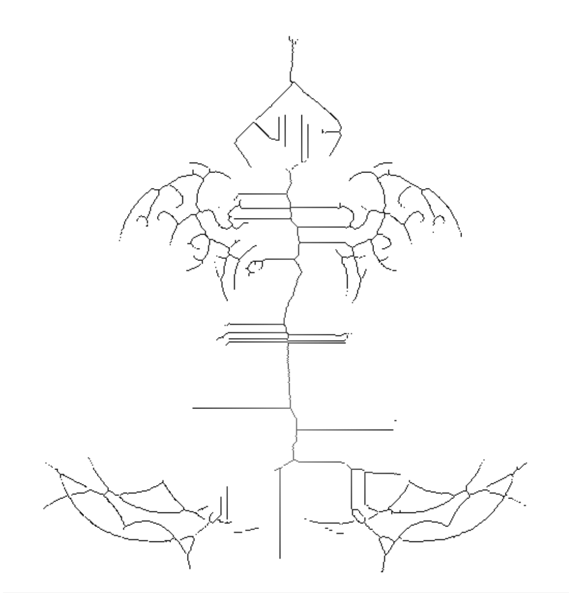
Reproduced image

Error in reproduction

13872 pixels to describe original image - 1269 pixels to describe distance skeleton (9,1%)

Input

Distance-skeleton



Reproduced image

Error in reproduction

71492 pixels to describe original image - 4622 pixels to describe distance skeleton (6,5%)

Conclusion

Our exploration of using distance transforms have revealed that they are not useful for creating skeletons which describe connectivity and topology, but they are able to preserve width information, and do this well enough to be able to reconstruct the original shape from the distance-skeleton. The entire algorithm is summarized in the figure below.

10am838

Jeppe Verium Larsen

Lars Knudsen

Rasmus Krarup Madsen

Christian Møller Takle

---

[i] http://pageperso.lif.univ-mrs.fr/~edouard.thiel/rech/1967-blum.pdf

[ii] http://www.inf.u-szeged.hu/~palagyi/skel/skel.html#Applications

[iii] Extracting Skeletons from Distance Maps, Sukmoon Chang, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.7, July 2007 - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.7424&rep=rep1&type=pdf