A Force-Based Thinning Strategy With Sub-Pixel Precision

J. Parker, C. Jennings, D. Molaro
Department of Computer Science
University of Calgary
Calgary, Alberta, CANADA
T2N-1N4

Abstract

Thinning can be defined as the act of identifying those pixels belonging to an object that are essential for communicating the object's shape. Most vision researchers would agree that the medial axis transform often does not yield an ideal skeleton. For example, single pixel irregularities can produce gross changes in an otherwise simple skeleton. Many of the more recent thinning algorithms were designed with an eye on the clock: the speed of the algorithm is improved, while often leaving the basic principles alone. Here. a thinning strategy is proposed that is based on a definition of a 'skeletal pixel' as being one that is as far from the object outline as possible while maintaining basic connectivity properties.. The basic idea is that a skeleton is a global property of a binary object, and that the boundary should be used to locate the skeletal pixels.

1. Introduction

The generation of a digital skeleton is often one of the first processing steps taken by a computer vision system when attempting to extract features from an object in an image. A skeleton is presumed to represent the shape of the object in a relatively small number of pixels, all of which are, in some sense, *structural* and therefore necessary. In line images the skeleton conveys all of the information found in the original, wherein lies the value of the skeleton: the position, orientation and length of the line segments of the skeleton are repr sentative of those of the lines of which the image is composed. This simplifies the task of characterizing the components of the line image.

Unfortunately no generally agreed upon definition of a digital skeleton exists, as pointed out by Davies and Plummer [11], Haralick [20], and a host of others. Of the literally hundreds of papers on the subject of thinning in print, the vast majority are concerned with the implementation of a variation on an existing thinning method, where the novel aspects are related to the performace of the algorithm. The quality of the skeleton or the means by which it is found are rarely the subject of analysis. In this paper an opposite view is taken: computer resources are cheap, and will be expended without consideration if the result is a high quality skeleton. We will also take some care to define the nature of the skeleton before one is created, and will attempt to quantify the results.

2. Defining 'Skeletal'

Possibly the first definition of a skeleton is that of Blum [5] in defining the *medial axis function* (MAF). The MAF treats all boundary pixels as point sources of a wave front. Each of these pixels excites its neighbors with a delay time proportional to distance, so that they too become part of the wave front. The wave passes through each point only once, and when two waves meet they cancel each other, producing a 'corner'. The *medial axis* (MA) is the locus of the corners, and forms the skeleton (Blum says *line of symmetry*) of the object. The MAF uses both time and space information, and can be inverted to give back the original picture. It is possible to implement but is difficult, common art involving various approximations usually involving the distance function on a discrete grid. This is reasonable enough when applied to a raster image.

The survey of the literature on thinning and related issues has led to a short list of generally agreed upon properties of the digital skeleton:

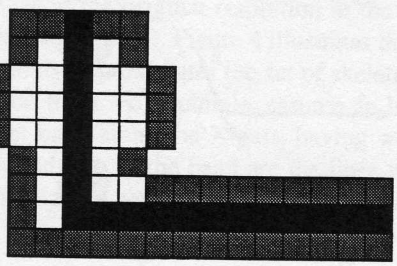
- A skeleton is a set of pixels
- Pixels in a skeleton are: 1 connected at ends, 2 connected at internal at internal points and 3+ connected at points of intersection.
- General shape of thick object must be retained by its skeleton.
- Topology must remain constant.
- The skeleton of one object should be one connected set of pixels.
- The skeleton must not contain any background pixels.

Elements implicit to definitions of skeleton include:

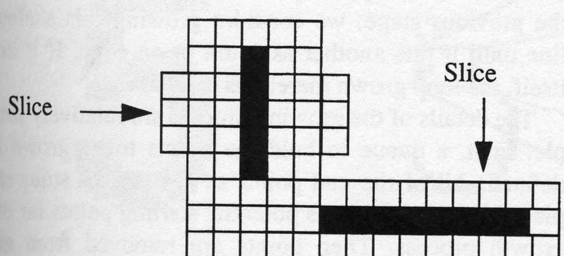
- Skeletal pixels are in some sense as far from the object boundaries as possible.

- A skeletal pixel is connected to at least one other, unless the skeleton consists of exactly one pixel.
- A line crossing the object boundary as a perpendicular will cross the skeleton exactly once before crossing another boundary, unless (a) too close to a point where lines meet, or (b) too close to the end of a line.

As an example, a simple object and its human computed skeleton is:



where grey represents a boundary pixel and a black pixel is a skeletal pixel. The skeleton above satisfies all of the discussed properties, and while a six year old human could draw it there are very few (if any) thinning algorithms that could. In most cases, humans perform thinning by computing a medial axis *in a preferred direction.* The center pixel found by slicing the object perpendicular to the stroke is chosen as skeletal wherever possible. This produces:



which is purely computational. There is also a perceptual aspect, which involves closing the gaps in the skeleton and extending the lines to the ends. This aspect can perhaps only be approximated on a computer. The direction in which to slice the object is that direction which is perpendicular to the stroke, and this may not be perpendicular to the boundary at all points. Non-local information is needed to perform this operation properly. In computer vision applications the skeleton of an object is extracted, and used to locate strokes. What is being proposed here is to reverse this process: strokes are located and used to generate the skeletons.

3. Definitions

We define a *digital band* as a set of connected pixels with the following properties:

- All pixels lie within perpendicular distance d of a discrete curve C, which does not have any loops (i.e. is simple). The minimum distance between C and any boundary pixel is d/2.
- The value of d is much smaller than the length of the curve C.
- The direction associated with each boundary pixel is approximately the same as that of the nearest point on C.

This definition would include most digital lines and curves, either thick or thin, as digital bands. A *digital band segment* is a subset of a digital band obtained by slicing the band at two places in a direction perpendicular to C at those places. This relaxes property two above so that the length of the curve C over the segment must be simply greater than 2d.

A *stub* is a digital band segment where there are constraints placed on the changes in direction undergone by C. In particular, over the segment: (1) the direction may be constant (linear stub), or (2) the direction may represent either a convex or concave curve (but not both) having an identifiable, if approximate, center and radius of curvature. Finally, the *skeleton of a stub* is the set of pixels obtained by using the center pixel of each slice across the stroke in a direction perpendicular to C. For example, in the case of a linear stroke these pixels should comprise the principal axis.

Now our approach to skeletonization can be clarified. Given a line image to be thinned we hypothesize that it can be broken down into a set of stubs that have been concatenated so that their boundaries form a continuous digital curve. These each have a clearly defined skeleton, and the first draft of the overall skeleton (the skeletal sketch) is simply the collected skeletons of all of the stubs. The skeleton may be complete at this point, although it is unlikely. The problem is that it is not possible to accurately determine the stubs comprising the object - some stubs are too short for this given that the image is discrete. It is often possible to fit hundreds of different stub combinations to a given object.

4. Use of a Force Field

The goal here is to find a method for locating skeletal pixels in a digital band that will also be useful as an approximation for objects consisting of concatenated band segments. Our idea is to have all of the background pixels which are adjacent to the boundary act as if they exerted a $1/r^2$ force on the object pixels. The skeletal pixels will lie in areas having the ridges of this force field, and these areas can be located by finding where the directions of the force vectors change significantly.

The algorithm first locates the background pixels having at least one object pixel as a neighbor and marks them. These will be assumed to exert a repulsive 'force' on all object pixels: the nearer the object pixel is to the boundary the greater is the force acting on it. This force field is mapped by subdividing the region into small squares and determining the force acting on the vertices of the squares. *The skeleton lies within those squares where the forces acting on the corners act in opposite directions.* Those squares containing skeletal areas are further subdivided, and the location of the skeletal area is recursively refined as far as necessary or possible.

The change in the direction of the force is found by computing the dot product of each pair of force vectors on corners of the square regions:

$$d_1 = \vec{f}_1 \cdot \vec{f}_2$$

$$d_2 = \vec{f}_2 \cdot \vec{f}_3$$

$$d_3 = \vec{f}_1 \cdot \vec{f}_4$$

If any one of d_1, d_2 or d_3 is negative then the region involved contains some skeletal area.

To compute the force vector at each pixel location is time consuming. For each object pixel a straight line is drawn to all marked pixels on the object outline. Lines passing through the background are discarded, as illustrated in Figure 1, , and for each of the remaining lines a vector with length $1/r^2$ and direction from the outline pixel to the object pixel is added to the force vector at that pixel.(Figure 2).

This is done for all object pixels. Then recursive sub-

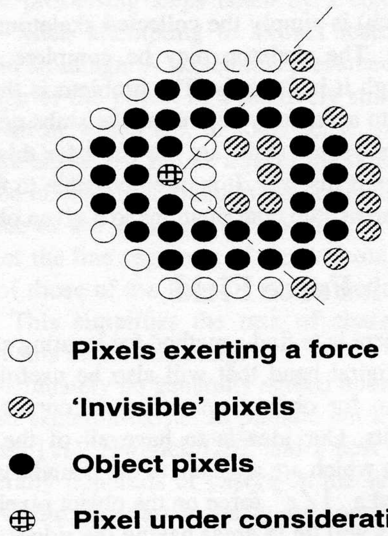


Figure 1 - Computing the force at a given pixel. Only some of the background pixels exert a force (visible ones).

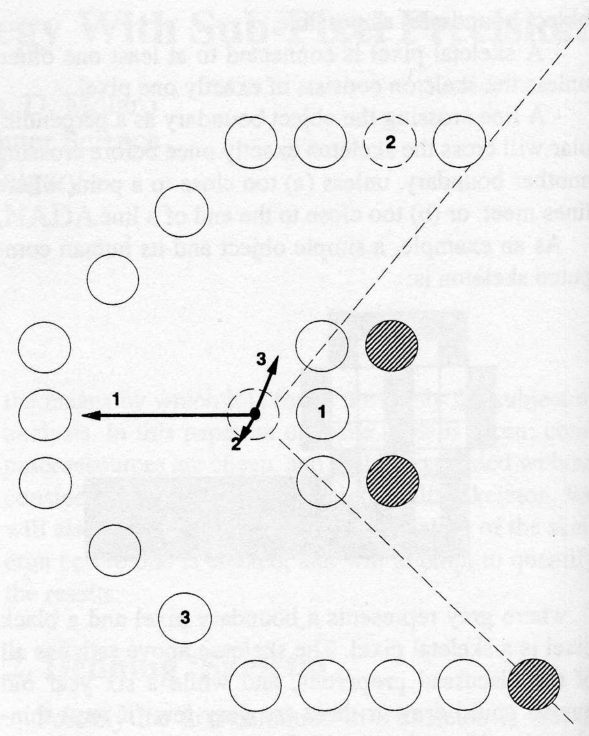


Figure 2 - A vector sum of forces from each visible pixel is accumulated at each object pixel not on the boundary.

division can be used to refine the positions of the skeletal areas. From any end points of the skeleton found in the previous stage, we consider growing this skeletal line until it hits another skeleton or an edge. If it hits itself, the loop grown thereby is deleted.

The details of the growing process are relatively simple. First, a queue to hold the points to be grown is defined. All of the end points of the current stubs are placed into the queue as potential starting points for the growth process. Then points are removed from the queue one at a time and tested to see if growth is possible; if so, it is added to the skeleton and the new skeletal point is added to the queue if it, too, is a potential starting point.

To grow from a point P, the point must satisfy two conditions. P must have exactly one or two 8-connected neighbors that are skeletal pixels, and if it has two such neighbors then these must be adjacent to each other. The preferred direction of growth is through these neighbors towards P and beyond to the next pixel. There will be three candidate pixels, and the one of these having the smallest force magnitude is 'grown into': it is added to the skeleton, and placed on the queue for further growth steps. The growing process will stop when the growth front hits an edge or other part of the skeleton.

At a sub-pixel level the growth process first attempts to find new skeletal pixels at double the previous resolution. Using the stub endpoints the regions to be refined are identified, and forces are computed for each pixel at the new resolution - resolution doubles each time. Then the dot products are computed as before, looking for zero crossings. When located, a zero crossing becomes a skeletal pixel at the current resolution and also marks all containing pixels at lower resolutions as skeletal. The refinement can be continued at higher resolutions until no change is seen; then the growth process continues at the original resolution in the original way (minimal force path). Figure 4 illustrates this.

This certainly approximates the set of skeletal pixels S for a digital band. For example, assume an infinitely long, straight band along the X axis, having width *2w*. Then the boundaries of the band are the lines *y=w* and *y=-w*. Then the force acting on the point *(x,y)* would be:

$$F(x,y) = \int_{-\infty}^{\infty} \frac{L_1}{|L_1|^3} dl_x + \int_{-\infty}^{\infty} \frac{L_2}{|L_2|^3} dl_x$$

where L_1 = (x-l, y-w), L_2 = (x-l, w+y), and l is the length along the boundary. This becomes:

$$F(x,y) = (0, \frac{4y}{(w+y)(y-w)})$$

Now, any of the dot products referred to previously can be written as:

$$d_i = (\frac{16y(y+dy)}{(w+y)(y-w)(w+y+dy)(y+dy-w)})$$

All that we need to know is when this expression is negative. Since -w+dy < y < w-dy we know that y-w and y+dy-w are negative and that w+y and w+y+dy are positive, the sign of the dot product is the sign of y(y+dy). Solving this quadratic reveals that it is negative only between 0 and -dy. Thus,

$$C(x,y,dx,dy) = \begin{cases} 1 & \text{if } -dy < y < 0 \\ 0 & \text{otherwise} \end{cases}$$

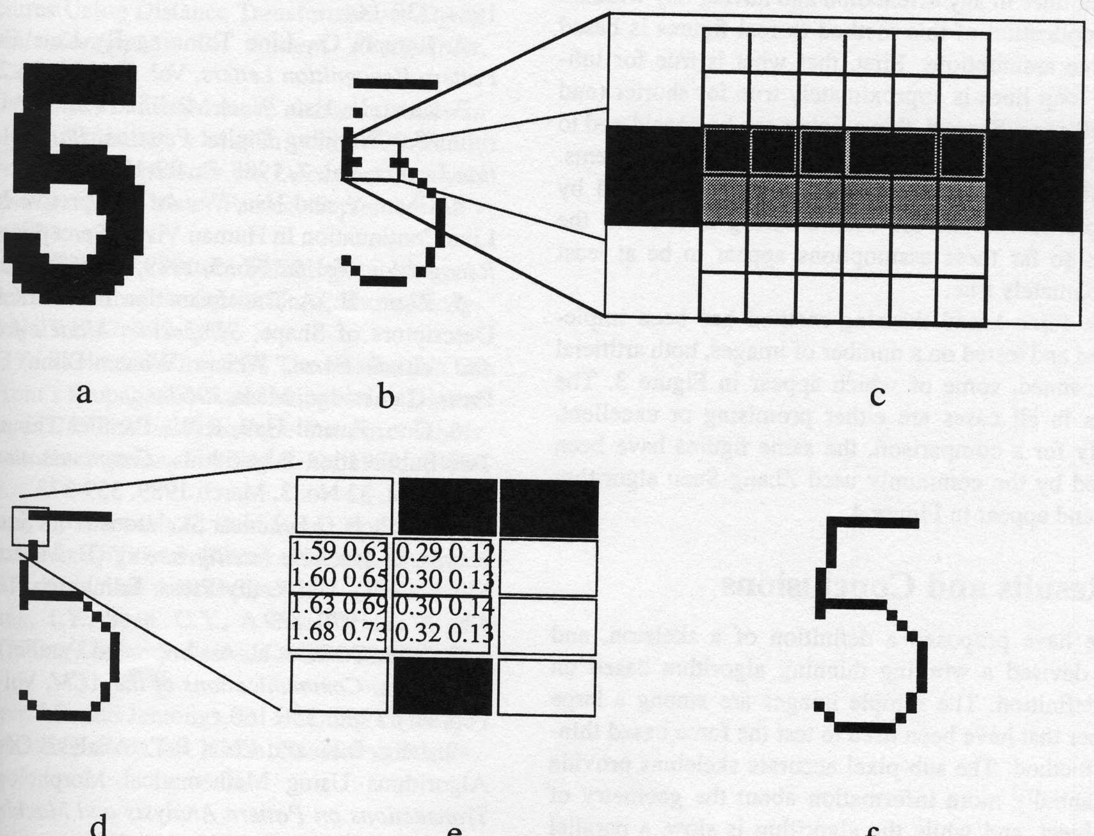


Figure 3 - (a) The original image. (b) Skeleton, level 0 (zero crossings). (c) A sub-pixel section of a gap in the level 0 skeleton. (d) The level 1 skeleton. (e) Sub-pixel force magnitudes in a gap in the level 1 skeleton. (f) The final skeleton with all gaps filled in.

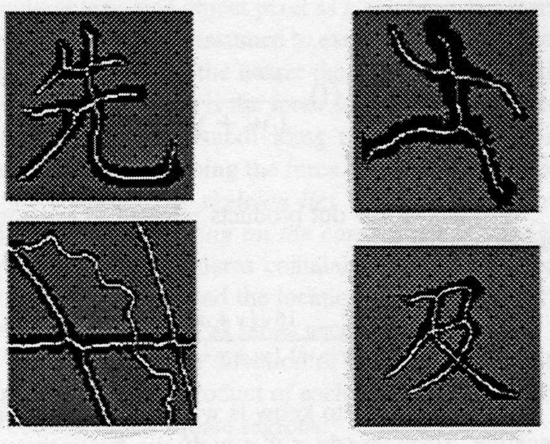


Figure 4 - Sample images thinned using the force based algorithm.

As dy approaches 0 this becomes

$$C(x, y) = \begin{cases} 1 & y = 0 \\ 0 & otherwise \end{cases}$$

which means that the X axis is the skeleton, as was suspected. This demonstration holds for infinitely long straight lines in any orientation and having any width. The application of this method to real figures is based on three assumptions. First, that what is true for infinitely long lines is approximately true for shorter (and curved) ones. Second, that a figure can be considered to be a collection of concatenated digital band segments. And finally, that intersections can be represented by multiple bands, one for each crossing line. From the results so far these assumptions appear to be at least approximately true.

The force based thinning method has been implemented and tested on a number of images, both artificial and scanned, some of which appear in Figure 3. The results in all cases are either promising or excellent. Simply for a comparison, the same figures have been thinned by the commonly used Zhang-Suen algorithm [24], and appear in Figure 4.

5. Results and Conclusions

We have proposed a definition of a skeleton, and have devised a working thinning algorithm based on that definition. The sample images are among a large number that have been used to test the force based thinning method. The sub-pixel accurate skeletons provide substantially more information about the geometry of the object, and while the algorithm is slow, a parallel version is significantly faster. We feel that the quality of the skeletons is often worth the execution time penalty.

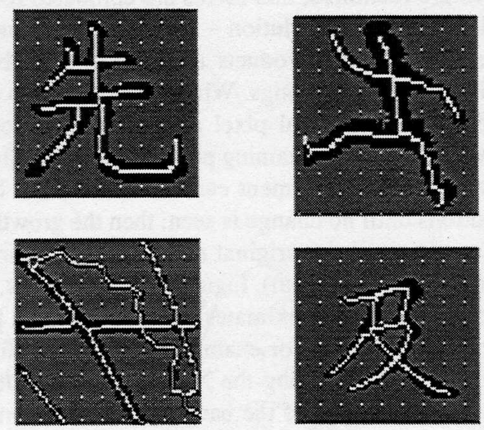


Figure 4 - Sample images thinned using the Zhang-Suen algorithm.

This research has been supported by the National Sciences and Engineering Research Council of Canada.

6. References

1. Arcelli, C., Pattern Thinning By Contour Tracing, *Computer Graphics and Image Processing*, Vol 17, 1981. 130-144.

2. Baruch, O., Line Thinning By Line Following, *Pattern Recognition Letters*, Vol. 8, 1988. Pp 271-276.

3. Chen, Y., Hsu, W., A Modified Fast Parallel Algorithm For Thinning Digital Patterns, *Pattern Recognition Letters*, Vol. 7, 1988. Pp 99-106.

4. Chen, Y. and Hsu, W., An Interpretive Model Of Line Continuation In Human Visual Perception, *Pattern Recognition*, Vol. 22 No. 5, 1989. 619-639.

5. Blum, H., A Transformation for Extracting New Descriptors of Shape, *Symposium Models for Speech and Visual Form*, Weiant Whaten-Dunn Ed., MIT Press, Cambridge, Mass, 1967.

6. Guo, Z. and Hall, R.W., Parallel Thinning With Two-Subiteration Algorithms, *Communications of the ACM*, Vol. 32 No. 3, March 1989. 359-373.

7. Hilditch, C.J., Linear Skeletons From Square Cupboards, in *Machine Intelligence IV* (B. Meltzer and D. Mitchie eds), University Press, Edinburgh, 1969. 403-420. 17, 2, 1970. 339.

8. Holt, C.M., et al, An Improved Parallel Thinning Algorithm, *Communications of the ACM*, Vol 30 No 2, February 1987. 156-160.

9. Jang, B.K. and Chin, R.T., Analysis Of Thinning Algorithms Using Mathematical Morphology, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, 1990. 541-551.

10. Kwok, P.C.K., A Thinning Algorithm By Contour Generation, *Communications of the ACM*, Vol. 31,

1988. 1314-1324.

11. Davis, E.R. and Plummer, A.P.N., Thinning Algorithms: A Critique And A New Methodology, *Pattern Recognition*, 14, 1981. 53-63.

12. Montanari, U., A Method For Obtaining Skeletons Using a Quasi-Euclidian Distance, *JACM* 15, October 1968, 600-624.

13. Montanari, U., Continuous Skeletons from Digitized Pictures, *JACM,* 16, 4, October 1969, 534-549.

14. Pal, S. K., Fuzzy Skeletonization Of An Image, *Pattern Recognition Letters*, (10) 1989. 17-23.

15. Piper, J., Efficient Implementation Of Skeletonization Using Interval Coding, *Pattern Recognition Letters*, (3) 1985. 389-397.

16. Sinha, R.M.K., A Width-Independent Algorithm For Character Skeleton Estimation, *Computer Vision, Graphics, and Image Processing*, Vol 40, 1987. 388-397.

17. Sossa, J. H., An Improved Parallel Algorithm For Thinning Digital Patterns, *Pattern Recognition Letters*,(10) 1989. 77-80.

18. Stefanelli, R., A Comment On An Investigation Into The Skeletonization Approach Of Hilditch, *Pattern Recognition*, Vol 19 no 1, 1986. 13-14.

19. Suzuki, S. and Abe, K., Sequential Thinning Of Binary Pictures Using Distance Transformation, *Eighth International Conference On Pattern Recognition*, 1986. 289-292.

20. Haralick, R.M., Performance characterization in image analysis: Thinning, a case in point, *ICDAR 91, First International Conference on Document Analysis and Recognition*, Saint-Malo, France, September 30 - October 2, 1991.

21. Suzuki, S., Binary Picture Thinning By An Iterative Parallel Two-Subcycle Operation, *Pattern Recognition* V20 No 3 1987, 297-307.

22. Xia, Y., Skeletonization VIA The Realization Of The Fire Front's Propagation And Extinction In Digital Binary Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol PAMI-11, 1989. 1076-1086.

23. Yokoi, S., Toriwaki, J., Fukumura, T., Topological Properties in Digitized Binary Pictures, *Systems Computer Controls*, Vol. 4 1973 32-39.

24. Zhang, T.Y., Suen, C.Y., A Fast Parallel Algorithm for Thining Digital Patterns, *CACM* Vol 27 no. 3, 236-239.

25. Parker, J.R. and Jennings, C., Defining The Digital Skeleton, *SPIE Conf. on Vision Geometry*, Boston, Nov. 1992.