National Conference on Emerging Trends in Engineering and Science (ETES) 2014

Midcurves Generation Algorithm for Thin Polygons

Yogesh H Kulkarni*, **Anil Sahasrabudhe****, **Mukund Kale*****
*Research Student, College of Engineering Pune, India
**Director, College of Engineering Pune, India
***Siemens PLM Software, India
*kulkarniyh12.mech@coep.ac.in

Abstract

Skeletons are widely used to represent shapes. A 2D or 3D shape is reduced in dimension to a 1-dimensional "skeleton". Such skeletons are widely used in pattern recognition, matching, finite element analysis etc. Skeletons can be computed for various inputs such as images, solids, sketches etc. Although there are various methods available for computation of skeletons, their appropriateness depends on requirement posed by their applications. Some focus on exact computation, some are approximate, whereas some aim at faithful backward-reconstruction and some at proper representation for human perception etc.

This paper focuses on creation of skeletons (specifically the ones called 'Midcurves', skeletons that are at the midway of a 2D shape) for polygon. A method based on Divide-and-Conquer strategy has been proposed which works in two phases - Decomposition and Midcurves generation. Towards end of the paper some test-case results are presented along with conclusions.

Keywords: Midcurves, Polygon Decomposition, Skeleton, Medial Axis Transform, Characters

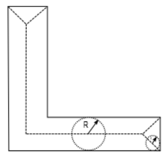
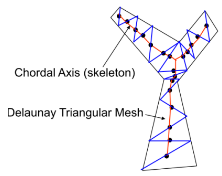
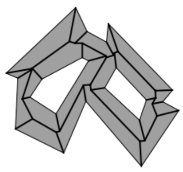
1 Introduction

A skeleton is a lower dimensional entity which represents shape of it's parent object. It being simpler than the parent object, operations like pattern recognition, approximation, similarity estimation, collision detection, animation, matching and deformation can be performed efficiently on it than on the parent object.

Skeletons can be computed via various mathematical formulations such as Medial Axis Transform (MAT), Chordal Axis Transform (CAT), Thinning etc. **Table 1** briefly summarizes these methods of Midcurves creation and their strengths-weaknesses. To overcome the weaknesses of these existing methods a novel algorithm is proposed which is based on *Divide-and-Conquer* principle.

In this paper we focus on 2D planar sketch profiles. Even in 2D profiles, shapes vary enormously. At the first level of simplification, we would deal with 2D polygons only(with an assumption that curved shapes can be converted to polygonal shape by faceting). *Divide-and-Conquer* is one of the widely used strategy for dealing with complex models. Shape decomposition partitions given shape into sub-shapes and then skeletonization can be performed on simpler sub-polygons. Later such individual midcurves can be joined to form continuous midcurves representing the original shape.

Table 1. Current Medial Computation Methods

Method	Medial	Description	Comments
MAT [1]		Locii of centers of maximal disk traversing within boundary	Computable for any shape. But has unwanted branches. Sensitive to boundary perturbations.
CAT [2]		Creates triangulation first then joins midpoints of sides	Gaps at end. Expensive triangulation.
Straight Skeleton [3]		Goes on thinning from boundary.	Bisectors mot equidistant. Have unnecessary branches.

A polygon can be decomposed into convex regions by dividing at all reflex(concave) vertices. Generally criterion for decomposition is to produce a minimum number of convex components or to minimize the total length of the boundary of these components. Within the minimum component criterion methods further classification could be based on whether or not Steiner points (brand new, non polygonal vertices) are allowed.

2 Related Work

One of the early works in the field of skeletonization was by theoretical biologist Harry Blum in1967 [4] with the medial axis of the shape. Survey papers like [5], [6] and [7] have detailed various approaches used in the skeleton creation. Decomposition of planar shapes into regular (non-intersecting) and singular (intersecting regions) and its application to skeletonization has been widely researched [8] as well.

Rocha [9, 8] used both Decomposition and Skeletonization for character recognition in images. From the results presented it appears that junctions like L, T were far from being well-connected.

Keil's algorithm [10] finds all possible ways to remove reflexity of these vertices, and then takes the one that requires fewest diagonals.

Lien et. al. [11] decomposed polygons 'approximately' based on iterative removal of most significant non-convex feature. Other methods of polygon decomposition are based on use of curvature to identify limbs-necks, using Axial Shape Graph, using events occurring during computation of Straight Skeleton, using Delaunay triangulation etc. Many of these methods need quite a bit of pre and post processing to take care of boundary noise [11]. Zou et. al.[12] partitioned a shape by

Delaunay triangulation, identified regular and singular regions and then created skeletons.

3 Proposed Method

2D polygonal profile is decomposed into sub-polygons. Sub polygons being simpler, make Midcurves creation more deterministic than getting the same from the whole profile.

3.1 Decomposition

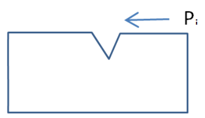
Polygons come with different types of variations. They can be simple/self-intersecting, with/without holes, concave/convex etc. This work focuses on a special type, an elongated polygon for which Midcurves can be generated e.g. alphabets made using Ribbon-like shapes. For this paper, more focus is given on profiles with constant thickness with test cases based on English alphabets.

Important points to note are [13]:

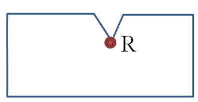
1. A polygon can be broken into convex regions by eliminating all reflex vertices.
2. A reflex vertex can only be removed if the diagonal connecting to it is within the range given by extending its neighboring edges; otherwise, its angle is only reduced.

Following steps explain the partitioning.

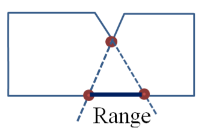
1. Go through all the vertices of the polygon one by one in counter-clockwise manner. Current vertex is called P_i



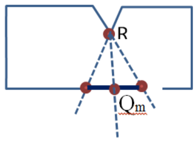
2. Check if P_i is a Reflex vertex R (Concave vertex with $InternalAngle > 180$, interior cusp of a polygon)



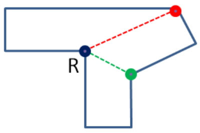
3. Extend lines incident at P_i (the line coming into P_i and going out of P_i) till they intersect remaining of the Polygon, say at Q_1 and Q_2. Contour within Q_1 and Q_2 is called $Range$



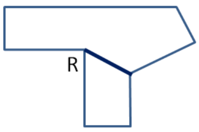
4. If there are no P_is within the $Range$, create a new one at the middle on the contour. This newly created point Q_m is called Steiner point. RQ_m is the partition-chord to divide the polygon.



5. If there are few vertices within the $Range$, choose best one based on following priorities.
   (a) Highest : Closest Reflex
   (b) Medium : Reflex
   (c) Low : Closest



6. Once vertex is chosen, say, Q_i, create partition chord RQ_i and divide the polygon.

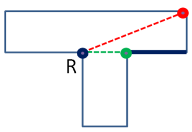


7. Send individual sub-polygons to the same process recursively till there are no reflex vertices left.

3.2 Improvements over Beyazit's algorithm [13]

Algorithm 1 improves upon the Beyazit's algorithm [13] in terms of expanding search to even include extreme vertices in the range thereby giving minimal and elongated partitions. Midcurves are typically for thin-elongated shapes. This improvement results in the sub-polygons of necessary shape characteristics.

If any of the incoming edges was hitting end points of test line or was collinear, it was getting ignored in the existing algorithm [13] and then the next closet vertex was getting chosen. Example: collinear/perpendicular lines but apart from each other. It has been avoided in the improved algorithm which can be clearly seen in the results shown in Table **Table 2**.



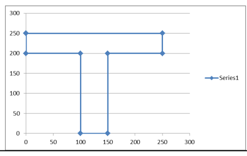
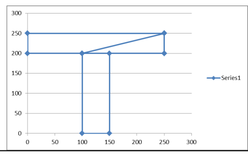
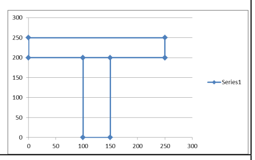
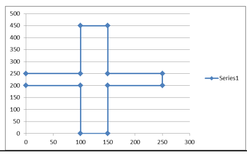
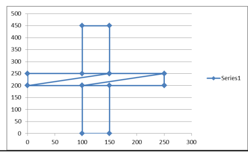
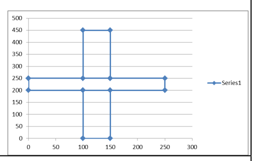
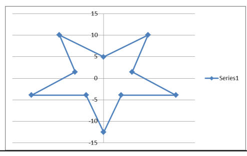
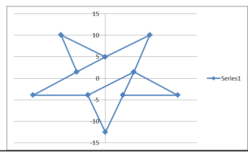
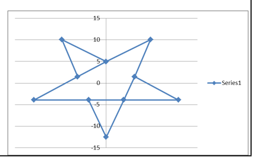
Algorithm 1 Polygon Decomposition

Require: 2D Planar polygon represented by list of vertices
Ensure: Vertices in counter-clockwise direction

```
while End of vertices list has not reached do
    Get the current vertex.
    if current vertex is a Reflex vertex R then
        Extend the edges incident at R until they hit an edge
        if Extension line and Polygon side are collinear then
            Find closest point which is not internal to the extension line
        end if
        if there are no vertices to connect to then
            choose a point in the middle
        else
            Find vertex to connect to
            Find best vertex B within the range, to form the partitioning chord
            Make sure B is visible from R
        end if
    end if
end while
Split the polygon at the cutting chord (line RB)
```

Table 2 demonstrates improvements over Beyzit's [13] algorithm with examples.

Table 2. Improvement Over current partitioning algorithm

Type	Shape	Bayazit[13]	Improvement
T: Extra-Longer split was unnecessary			
Plus : Extra-Longer splits unnecessary			
Star : Splitting was not at the closest Reflex			

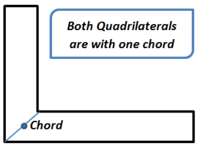
The resulting sub-polygons are sent for creating connected midcurves.

3.3 Midcurves Generation

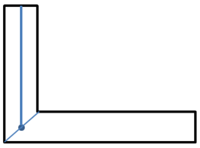
Decomposition helped getting the sub-polygons which are of primitive shapes and which are easier for Midcurves creation compared to original-whole shape.

Following steps explain the Midcurve Generation process:

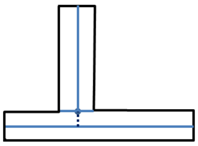
1. Partitioning: Decompose Polygons into sub-polygons of primitive shapes using **Algorithm 1**



2. Generate Midcurves for individual polygons taking chords into consideration. 'Thinness' is an important criterion in choosing midcurves for the individual shape. Midcurves are generated along longer-length and not across shorter width



3. In shapes like 'L' midcurves from both sub-polygons, across the chord, join together at a point, naturally. But in case of shapes like 'T', the horizontal midcurve does not connect with the common chord. In this case one of the midcurves needs to be extended to join the other



Each such shape can create its own Midcurves based on number of sides and also where the cutting-chords lie on this individual shape. Chord is a common interface-boundary shared between two sub-polygons. Each chord will have two sides owned by two different sub-polygons. Each sub-polygon needs to look at it's own shape, slenderness and decide own Midcurve.

After creating individual Midcurves, they all may or may not join the chords. In case it does not join to any side of the chord, some extension has to be provided from other side of that chord. Chords are processed to ignore the ones which have partial overlap with the sub-polygon sides or are collinear. This method gives cleaner (without branches) and connected Midcurves compared to previously cited methods. Pseudo code for Midcurve generation process is presented in **Algorithm 2**.

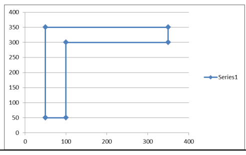
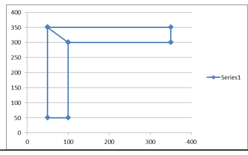
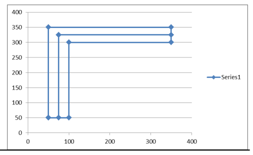
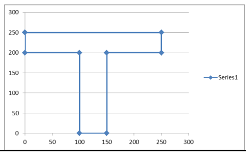
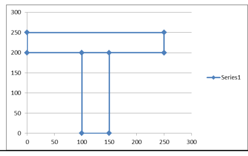
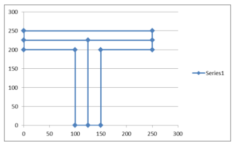
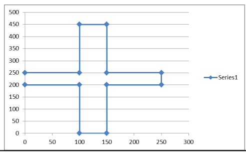
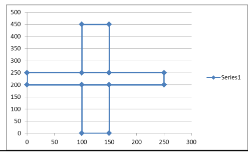
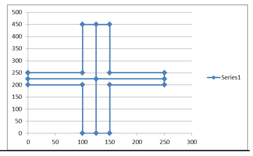
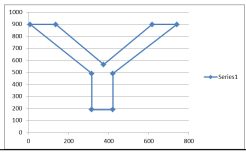
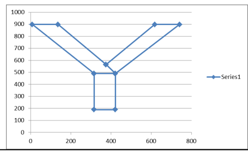
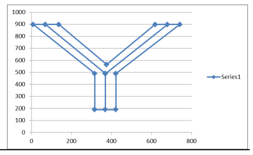
Algorithm 2 Midcurves Creation

Require: List of partitioned 2D Planar polygons represented by list of vertices in counter-clockwise direction
  Find internal-common edges called chords
  Iterate over all polygons and create chords at Full or Partial overlap
  while End of Polygons list has not reached **do**
    Get the current polygon P
    Get chords which are part of P
    Look at various combinations-configurations due to Num Sides and Num Chords
    Generate Midcurves
    Assign Midcurves on relevant side of the chord
  end while
  Extend chords which are not connected with other neighboring chords

4 Results

Table 3 briefly summarizes the results of the Decomposition Algorithm as well as that of the Midcurves Algorithms, for some sample polygons. The examples shown here are of English alphabets as they present various ways of connectivity, have curves, holes and are easier to imagine and verify.

Table 3. Partitions-Medial Computation

Type	Shape	Partitioning	Midcurves
L: Midcurves on both sides of the chord, joined			
T : Midcurve only on one side; need to extend			
Plus : Midcurve on either sides; need to extend			
Y : Collinear chord ignored; need to extend			

5 Conclusion

Skeletons generated by many widely researched methods like MAT, CAT, Straight Skeleton, do not represent the shape as per human perception, due to presence of gaps and unnecessary branches. For non-trivial shapes, instead of computing Midcurves on the whole shape, one can divide the shape in more manageable-simpler shapes for which Midcurves generation is far simpler problem.

This paper presents such *divide-and-conquer* strategy. Polygons are decomposed into primitive sub-polygons, Midcurves are created for each of them and wherever necessary they are joined by extension. The results of improved partitioning (over Bayazit's [13] approach) and then its usage in creating better-connected Midcurves are shown

References

[1] Ramanathan M and Gurumoorthy B, "Generating the mid-surface of a solid using 2d mat of its faces," *Computer Aided Design and Applications*, vol. 1, pp. 665–674, 2004.

[2] William Roshan Quadros, "An approach for extracting non-manifold mid-surfaces of thin-wall solids using chordal axis transform," *Eng. with Comput.*, vol. 24, no. 3, pp. 305–319, 2008.

[3] Haunert Jan-Henrik and Sester Monika, "Using the straight skeleton for generalisation in a multiple representation environment," *ICA Workshop on Generalisation and Multiple representation*, 2004.

[4] Blum Harry, *A Transformation for Extracting New Descriptors of Shape. In Models for the Perception of Speech and Visual Form*, MIT Press, 1967.

[5] Attali D., Boissonnat J. D., and Edelsbrunner H., "Stability and computation of the medial axis a state-of-the-art report," *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, 2004.

[6] Louisa Lam, Seong-Whan Lee, and Chiang Y. Suen, "Thinning methodologies—a comprehensive Survey," *IEEE-PAMI*, vol. 14, no. 9, pp. 869–884, Sept. 1992.

[7] Yogesh Kulkarni and Shailesh Deshpande, "Medial object extraction - a state of the art," in *Proc. of the 3rd International Conference on Advances in Mechanical Engineering*, S.V. National Institute of Technology, Surat , Gujarat, India, January 4-6, 2010.

[8] Jairo Rocha, "Polygon decomposition into singular and regular regions," Tech. Rep., University of the Balearic Islands, 1999.

[9] Jairo Rocha and Rafael Bernardino, "Singularities and regularities on line pictures via symmetrical trapezoids," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 20, 1998.

[10] Mark Keil and Jack Snoeyink, "On the time bound for convex decomposition of simple polygon," *International Journal of Computational Geometry & Applications*, 1994.

[11] Jyh-Ming Lien and Nancy M. Amato, "Approximate convex decomposition of polygons," *SCG*, 2004.

[12] Ju Jia Zou and Hong Yan, "Skeletonization of ribbon-like shapes based on regularity and singularity analyses," *IEEE Transactions on systems, man and Cybernetics*, vol. 31, pp. 401–407, 2001.

[13] Mark Bayazit, "Poly decomp http://mnbayazit.com/406/bayazit," .