Geometrical Concepts and Graph Theory For Linear Curve Approximation

[1]K.Venkatasubramanian, [2]Dr. S.K.Srivatsa, [3]Dr. C.Parthasarathy
[1,2,3]Dept. of CSE, SCSVMV University, Kancheepuram, India

Abstract
In this paper, a brand new methodology for curve approximation is bestowed. The tactic is appropriate for each self-intersected and non self-intersected curves, it combines elements from graph theory and from parabolic geometry and it is absolutely machine-controlled. Additional specifically, graph theory tools square measure utilized in order: (1) to get rid of the small print that square measure irrelevant to the general form of the curve below study and (2) to decompose the curve into non self-intersecting smaller curves. Then, each such smaller curve is processed via geometrical tools so as to approximate it with efficiency with linear segments. Experimental results show that the planned technique compares well with several alternative ways of constant purpose.

Keywords
Geometry, Segment, Intersecting, Curve

I. Introduction
Digital planate curves square measure utilized in many fields of special effects, separate pure mathematics and digital image analysis. Several results are created concerning their geometric behavior since [1]. A special topic is digital curve compression.

Besides straightforward techniques like chain secret writing, a usual approach is to partition the curve into line segments [2] for compression. These ways usually target straightforward curves with no self-intersections, and assume the preliminary data on the sequent order of the curve points. The progressive approach

JBEAM [3] considers an alphabet of transportation system segments (called beamless) to compose the curve. This methodology divides the binary image containing the curve victimization quad tree decomposition until having one linear curve section in each quad tree cell that may be substituted by a beamless. The advantage of this approach is that any curves are often handled by sufficiently fine quad tree decomposition. However, a downside is that the obligation of moldering afterwards, once a cell contains such segments that already may be coded severally. In this paper, we have a tendency to propose a graph theoretical approach to trace curves having impulsive topology to obtain higher compression performance, once rending the curve into line segments. Because of the tracing step, the planned methodology has higher compression performance than JBEAM [3]. The most improvement lies within the undeniable fact that we have a tendency to perform a whole tracing of the curve rather than moldering its storing canvas recursively, whereas solely line segments stay in the quad tree cells. The structure of this paper is as follows. In section II we have a tendency to recall the graph theoretical background that is a basis for our approach in tracing curves. We have a tendency to additionally justify however the acceptable graph representation of the digital curve is obtained.

II. Tracing Curves Using Graph Theory
In this section we have a tendency to recall some notions and results of graph and curve theory that we have a tendency to apply to trace a curve and conjointly some techniques that were thought-about to get the corresponding graph representation of the curve.

A .Graph theoretical background
A graph G is defined as a pair (V,E), where V is a set of vertices, and $E \subseteq V \times V = \{\{u,v\}/u,v \in V\}$ is a set of edges between the vertices. As we use graph representations of curves, we focus on undirected graphs, so $\forall$ u,v $\in$ V: {u,v} = {v,u} holds. To cover a wide class of curves, we allow loops (edges of type {u, u}) and multiple edges (more edges between two vertices). The degree of a vertex is the number of edges containing the vertex. A path is a list of vertices {u1, u2, . . . , un} having edges between any two consecutive vertices: {u1, u2}, {u2, u3}, . . . , {un−1, un}, with u1 = un in the case of a route (closed path). G is connected, if any two of its vertices have a path connecting them. A path through G which includes every edge exactly once is called an Euler path (or an Euler route if the start and end vertices coincide) [4-5]. Note that any Euler route is also an Euler path. G is an Euler graph, if it contains an Euler path through all of its edges. An Euler decomposition of G has the form Gi such that all the Gi's are disjoint Euler graphs (in the sense that they cannot contain the same edge). We recall some well-known facts on Euler graphs and their decomposition

1. Every Euler graph is connected.
2. A connected graph contains an Euler route if all of its vertices have even degree. The route can start from any vertex.
3. A connected graph contains an Euler path if f at most two of its vertices have odd degree. If there are two vertices with odd degree, the path starts from either of them and ends in the other.
4. Every connected graph has an Euler decomposition into disjoint Euler graphs.

B. Assignment of a Graph to a Digital Curve
The definition of simple curves in the Euclidean space was given by P. Urysohn in 1923 and K. Menger in 1932 independently (see [8] for a review). The curves were classified based on the branching indeces of the curve points, where a branching index of a curve point is equal to the number of curve segments meeting at the given point. The adequate mathematical formulation for the Euclidean space can be found in [8-9]. For the discrete domain Z2, this definition can also be adapted using the well-known 8-neighboring relation. We find the tip points of the sides as regular points being 8-neighbors to junctions (if each of their 8-neighbors square measure branch points, the sting is degenerated having length 1). Then, the sting finish points square measure organized into pairs (edges) supported the condition that Associate in nursing 8-connected path are often found between them whose components square measure regular points. Figure 1a depicts the results of locating finish points and junctions (shown framed, in light-weight gray), while 1b take a better search for the choice of edge finish points (dark gray), and for the sides outlined by them. These figures additionally indicate the branching indices of the curve points.

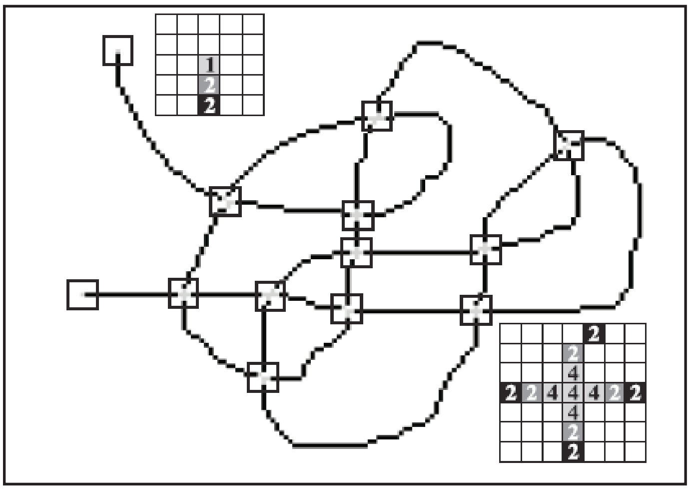

Fig. 1(a):

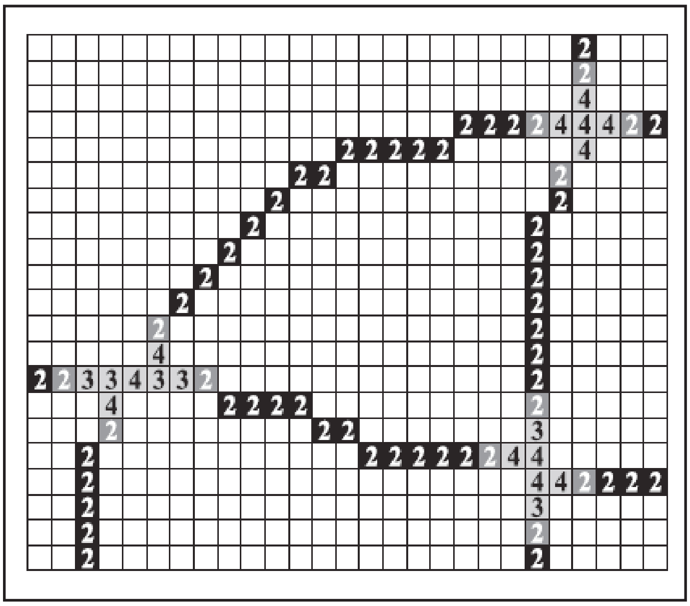

Fig. 1(b):

Fig. 1: Locating vertices and extracting the edges for the abstract curve graph GC = (VC,EC). (a) Input test planar curve C with its end points and junctions to compose VC are framed. (b) Extracting edges for EC via locating edge end points (dark grey) and connecting them with 8-paths

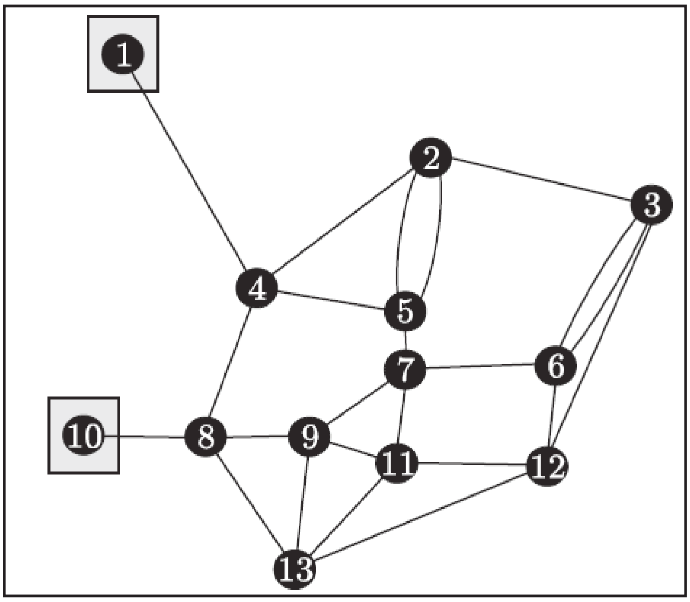

Fig. 2: The Simplified Abstract Curve Graph of the Curve Shown in fig. 1(a) With Vertices of Odd Degree Framed

III. Optimized Tracing For Compression

The first step to trace AN Leonhard Euler curve is to find a beginning vertex in step with statement. We tend to check the vertices and choose one having odd degree. If all the degrees square measure even, we are able to select a discretionary vertex to start out from. Then we tend to take a footing from the beginning vertex to initialize the tracer. As an example, within the graph shown in Figure two 2 vertices (1, and 10) have odd degrees. Thus, the Leonhard Euler path ought to begin from vertex one to end at vertex ten, or vice versa. As additional Leonhard Euler methods might exist, we've to choose that edge to require next, once reaching junctions. As our intention is to substitute the curve give line segments for curve compression, the natural call is to travel on straight ahead at junctions.

IV. Curve Compression by an Alphabet of Line Segments

We can make a choice from an enormous range of techniques to partition a curve into digital straight line segments [2]. These techniques are often classified as offline (the curve is examined globally to find associate degree best partitioning), or on-line (the curve is rotten into line segments throughout its traversal).

Thorough our projected approach is appropriate for each tasks, we tend to discuss a web coding risk here. To partition the curve into digital straight segments we tend to use a linear online methodology conferred in [14]. To obtain a secret writing theme from the line decomposition, we tend to replace all the created linear curve segments by components of associate degree alphabet of line segments. We tend to produce a finite alphabet whose letters are digital line segments of all doable orientations having length at the most T pixels.

As a plain consequence, we've to prevent process the curve once the top section length T is reached and that we need to explore for succeeding section, albeit the coded one would continue straight. Moreover, to stay the cardinality of little, we tend to contemplate distinctive straight line segments to attach 2 points. For this purpose, we tend to contemplate the Bresenham drawing algorithm [15] to form the letters of. Note that this fashion we tend to permit some info loss, since the Bresenham segments could slightly disagree from those extracted throughout the net curve segmentation method.

On the opposite hand, these variations are very minor perceptually, since digital straightness is our essential demand.

VI. Comparative Analysis

A. Comparing with JBEAM

To test and compare the compression efficiency of our method, we fixed the following setup. We used T = 32 as a threshold for the maximum line segment length for all the test curves, and considered the default loss JBEAM parameterizations [3]. Our experimental results are shown in Table 1.

Table 1: Comparative Quantitative Results Against JBEAM

Test curve	# of pixels	JBEAM (# of bits)	Proposed method (CT)	
			# of bits	# of segments
General	2127	1586	744	62
Lines	2745	1398	468	39
Spring	4113	2308	1224	102
Script	2511	1419	828	69
Non-Euler	1242	834	$2 \times 240 = 480$	$2 \times 20 = 40$

We can conclude that the proposed method has a 50% improvement on average in compression against JBEAM. Figure 7 depicts the coding results for our sample curves. We marked the end points of the line segments found by our coding method. For the sake of completeness, we mention that the coordinates of the starting point of every Euler path should be stored, as well. However, we ignored this issue in our calculations, since it produces only insignificant increase in the number of bits of the compressed curve.

VII. Compressing EDGES Separately

The process presented in Section II-B is suitable to extract the edges of an abstract curve graph, where the edges are actually curve segments. We might as well execute our compression approach at edge level without the intention to perform any tracing of the curve. Though this simple approach avoids the over segmentation of the quad tree cells considered by JBEAM [3], it has several drawbacks. In this case, curve segments (edges) are compressed separately, so we have to store the coordinates of the start pixel of each curve segment for appropriate geometric positioning. Moreover, since no junction point information is stored, we have to connect the consecutive curve segments during the decoding process (e.g. using the Bresenham algorithm [15]), which may lead to curve distortion at junctions.

VIII. Conclusion

In this paper, we have a tendency to propose curve compression techniques supported graph theory and therefore the use of a line phase alphabet. we have a tendency to use the abstract curve graph illustration of the curve to be compressed by locating junctions and considering curve elements as graph edges. The abstract curve graph is rotten into Leonhard Euler subgraphs, which may be copied on an individual basis. to exchange the curve elements with linear segments, we have a tendency to fix associate alphabet of delimited line segments. We have a tendency to additionally discuss some variants of the planned approach. Experimental results area unit conferred as compared with state-of-the-art curve compression approaches, further we have a tendency to should mention that JBEAM [3] additionally realizes a progressive approach in curve compression. That is, once solely a region of bits has been received, the digital curve is recovered about. Our methodology doesn't have a right away support for progressive encryption, however, some kind of changeability might be reached by re-ordering the transmitted Bresenham segments. For example, to own an impact regarding the form of the curve, we will send the segments having indices equidistantly sampled on the graph route. Another approach could relate to the case, when junctions area unit a lot of vital. Then we will transmit those segments initial that contain or touch branch points. All told these cases, to boot information area unit required to be transmitted, since the segments are not any longer subsequent ones.

References

[1] A. Rosenfeld,"Arcs and Curves in Digital Pictures", Journal of ACM, Vol. 20, No. 1, pp. 81–87, 1973.
[2] R. Klette, A. Rosenfeld,"Digital straightness - a review", Disc. Appl. Math., Vol. 139, No. 1-3, pp. 197–230, 2004.
[3] X. Huo, J. Chen,"JBEAM: multiscale curve coding via beamlets", IEEE Transactions on Image Processing, Vol. 14, No. 11, pp. 1665–1677, 2005.
[4] N.L. Biggs, E.K. Lloyd, R.J. Wilson, Graph Theory", Calendon Press, Oxford, 1998.
[5] L. Euler,"Solutio problematis ad geometrian situs pertinentis", Commentarii academiae scientarum Petropolitanae, Vol. 8, pp. 128–140, 1736.
[6] H. Fleischner,"Eulerian Graphs and Related Topics", Part 1. Vol. 1", Vol. 45 of Annals of Discrete Mathematics, North-Holland Publishing Co., Amsterdam, 1990.
[7] H. Fleischner,"Eulerian Graphs and Related Topics. Part 1. Vol. 2", Vol. 50 of Annals of Discrete Mathematics, North-Holland Publishing Co., Amsterdam, 1991.
[8] R. Klette, A. Rosenfeld,"Digital Geometry - Geometric Methods for Picture Analysis", Morgan Kaufmann, San Francisco, 2004.
[9] G. Klette,"Branch Voxels and Junctions in 3D skeletons", in Proc. Int. Conf. 10th IWCIA 2006, Berlin, Germany, Lecture Notes in Computer Science, vol. 4040, Springer, Berlin, pp. 34–44, 2006.
[10] T. Pavlidis,"Algorithms for Graphics and Image Processing", Computer Science Press, Rockville, MD, 1982.
[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms", Second Edition. MIT Press and McGraw-Hill, 2001.
[12] Fleury,"Deux problemes de geometrie de situation", Journal de mathematiques elementaires, pp. 257–261, 1883.
[13] S. Skiena,"Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica", Reading, MA: Addison-Wesley, 1990.
[14] I. Debled-Rennesson, J. Reveill´es,"A linear algorithm for segmentation of digital curves", Int. Journ. Patt. Recogn. and Artif. Intell., Vol. 9, pp. 635–662, 1995.
[15] J.E. Bresenham,"Algorithm for computer control of a digital plotter", IBM Systems Journal, pp. 25–30, 1965.

K.Venkatasubramanian has been working as Assistant Professor in the Department of Information Technology at Ramachandra College of Engineering, Eluru. He has completed M.Tech.(IT) at Sathyabama University, Chennai. Presently, he is doing Ph.D in Information Technology at SCSVMV University, Kancheepuram. His research interest is in the area of Graph Theory Algorithms.

Dr. S. K. Srivatsa was born at Bangalore on 21st July 1945. He received his Bachelor of Electronics & Communication Engineering Degree (Honors) from Javadpur University (Securing First Rank & Two Medals). Master Degree in Electronics & Communication Engineering (with Distinction) and PhD from Indian Institute of Science, Bangalore. In July2005, he retired as Professor of Electronics Engineering from Anna University. He has taught 22 different course at U.G level and 40 different courses at P.G, IEEE. He has functioned as Member of the Board of studies in reputed Educational Institutions. His name is included in Computer Society of India database of Resource Professionals. He has received about a dozen awards for his Excellency. He has produced more than 45 PhDs. He is the author of well over 400 research publications.

 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Currently he is working as Senior Prof. Department of CSE, Pratyusha Institute of Technology & Management (PITAM), Chennai 602 025.

Dr.C.Parthasarathy has been working as Assistant Professor in the Department of Information Technology in SCSVMV University, Enathur, Kancheepuram. Since 2006. He has completed M.C.A at Madras University, and M.Tech at Sathyabama University and M.Phil in Computer Science at Annamalai University. Since, January 1st 2001 C.Parthasarathy has been a Lecturer in various colleges. He had completed Ph.D in network security at S.C.S.V.M.V University, Kancheepuram. He has attended international and National seminars, Conferences, Workshops and also presented more number of papers. Presently, he is working as HoD of Information Technology at SCSVMV University, Kancheepuram.