International Journal of Computer Graphics
Vol. 6, No.2 (2015), pp.1-12
http://dx.doi.org/10.14257/ijcg.2015.6.2.01

A Split-Merge-Split Approach to Polygonal Approximation of Chain-Coded Curve

Bimal Kumar Ray

School of Information Technology & Engineering
VIT University, Vellore, India
bimalkumarray@vit.ac.in, raybk_2000@yahoo.com

Abstract

This paper proposes a technique for polygonal approximation of chain-coded curve after relaxing the definition of digital straight segment. The initial approximation is modified by merging quasi linear vertices followed by insertion of additional vertices. The entire process involves comparison and computation on integral domain. It uses an approximation of perpendicular distance which is shown to impose an upper bound on perpendicular distance. The experimental results show improvement upon a similar work.

Keywords*: digital straight segment, asymmetric digital straight segment, split, merge, perpendicular distance, distance to a point, integral domain*

1. Introduction

Approximation of a digital curve by a sequence of piece straight line segments has drawn considerable attention in research community and found its applications in pattern recognition, computer vision, cartography, and in other related areas. If the curve is closed then the approximation is called polygonal approximation otherwise, it is called a poly-line approximation. These algorithms can be categorized as iterative and sequential, optimal and suboptimal. A number of algorithms for polygonal approximation of digital curve had been developed. The earliest among the these algorithms was an iterative splitting technique [1] followed by an iterative split-and-merge technique [2] and then the earliest one-pass sequential technique using perpendicular distance [3]. Though iterative splitting technique depends on initial segmentation but iterative split-and-merge does not. But in either technique, the fundamental problem is initial segmentation. Ramer [1] proposed that the top left corner and bottom corner of the input curve can be taken as initial vertices of the polygon. Researchers in this area also proposed techniques such as *local vertex adjustment*, *split-and-merge post processing*, to rectify the errors in the approximation because of the ad hoc initial segmentation albeit with additional cost. The one-pass sequential technique [3] is fast, but it rounds off sharp turnings and misses corners. Wall and Danielsson [4] used area deviation per unit length as a measure of co-linearity and devised a one-pass sequential technique and succeeded to retain sharp turnings and corners using an additional test which was referred to as `peak test`.

The above mentioned techniques generate a suboptimal polygon. Dunham [5] and others (*e.g.*, [6]) used dynamic programming to find an optimal approximation of a curve by piece straight line segments.

Pikaz and Dinstein [7], Zhu and Chirlian [8], Visvalingam and Whytt [9], introduced iterative point elimination that treats polygonal approximation as problem of elimination of points from a curve that are not likely to be candidate vertices of the approximation. Genetic approach has also been used to address the problem of polygonal approximation *e.g.*, [10, 11].

ISSN: 2093-9663 IJCG

A digital curve can be defined by a sequence of vectors with its inclination a multiple of $\pi/4$ and length either unity or $\sqrt{2}$ and this sequence of vectors is called chain codes introduced by Freeman [12]. The run length of a direction is defined as the number of consecutive chains in the direction. For any two points P and Q lying on a digital curve, if every point on the curve is close to the line segment joining P and Q then the curve is said to have chord property. A digital curve is called a digital straight segment (DSS) if it satisfies chord property [13]. It is easy to prove a number of regularity properties [13] of DSS using its chord property. The regularity properties are referred to as rules that a DSS should satisfy. These rules, mentioned in [13], are reproduced here for the convenience of the reader.

Rule 1 The runs have at most two directions, differing by $\pi/4$, and for one of these directions, the run length must be unity.

Rule 2 The runs can have only two lengths which are consecutive integers.

Rule 3 One of the run lengths can occur only once at a time.

Rule 4 For the run length that occurs in runs, these runs themselves can only have two lengths which are consecutive integers and so on.

It is possible to approximate a digital curve using DSS identification algorithm. If the curve is closed then this approximation is called polygonal approximation. But using DSS for polygonal approximation of digital curves results in too many straight line segments and this is why Bhowmick and Bhattacharya [14] relaxed the definition of DSS, retaining Rule 1, modifying Rule 2 and dropping the other rules and thereby introduced the concept of what they called Approximate Digital Straight Segment (ADSS). The definition of ADSS involves determination of singular and non-singular elements, and then allowing a segment to grow beyond a DSS as long as the run length of the nonsingular elements does not exceed the smallest run length (p) by more than $floor\left(\frac{p+t}{t+1}\right)$, where t modulo 2 does not vanish. Bhowmick and Bhattacharya used $t=1$ and proposed that some other value of t *e.g.*, $t=3$, could, instead be used. They also imposed similar constraint on the value of the leftmost as well as the rightmost nonsingular run length so that an ADSS does not deviate too much from the digital curve and stated that the value of t may differ among the leftmost, the rightmost and intermediate run lengths. Though ADSS extraction results in less number of visually straight segments than DSS does, but choice of $t=1$ and $t=3$ is ad hoc in nature having no bearing on the local topology of the curve and the choice requires human intervention.

The ADSS acts as an initial segmentation of the digital curve. Usually decomposition of a digital curve by ADSS too, results in too many ADSS and this is why Bhowmick and Bhattacharya [14] merged multiple successive ADSS based on two different criterion functions namely, (i) sum of area deviation of the curve per unit length of approximating line segment (in the line of Wall and Danielsson [4]) and (ii) maximum area deviation of the curve per unit length of the approximating line segment. This results in representation of a digital curve by a polygon / poly-line approximation. Experiments with a variety of data set show that polygonal / poly-line approximation using criterion (ii) produces less accurate representation that those using criterion (i). Moreover, Wall and Danielsson [4] used a `peak test` and with it, succeeded to retain sharp turnings, but since no such test was used in Bhowmick and Bhattacharya [14] so the approximations are found to round off sharp turnings.

This paper introduces Asymmetric Digital Straight Segment (AsymDSS) that does not require ad hoc choice of parameter but it can detect visually straight segments successfully in most of the scenarios. Segmentation of a digital curve into AsymDSS is followed by merging quasi linear vertices and inserting additional vertices (splitting), if required. Perpendicular distance in integral domain is introduced and used as a metric for merging and subsequent splitting. To retain sharp turnings, distance to a point is used instead of perpendicular distance and this distance computation also is performed in integral domain. All arithmetic and comparisons involved in the entire process – decomposition of digital curve into AsymDSS, followed by merging and subsequent splitting – are in integral domain.

In the next section, AsymDSS is introduced, in Section 3, technique for merging quasi linear vertices is presented, in Section 4, procedure for insertion of additional vertices is discussed, In Section 5, a brief discussion on Figure of Merit for measuring fidelity of polygonal approximation is presented, in Section 6, time complexity of the proposed method is derived, in Section 7, experimental results are presented with a brief analysis of the same and finally in Section 8, conclusion is drawn and an enhancement is proposed.

2. Asymmetric Digital Straight Segment Detection

AsymDSS detection starts off by identifying singular and nonsingular elements and continues to merge chain codes with the segment as long as singular element occurs *singly* without imposing any constraint on the length of nonsingular run lengths. This strategy is the consequence of retaining Rule 1 of DSS and dropping the rest of the rules, mentioned in the last section. The rationale behind dropping Rule 2 through 4 is that these rules were required for maintaining symmetry in a DSS.

A number of digital curves along with the AsymDSS generated thereof are shown in Figure 1. Each of the digital curves results in a single AsymDSS because the singular element has length unity. The AsymDSS in Figure 1(a) and 1(b) are also ADSS but the curves in Figure 1(c) and

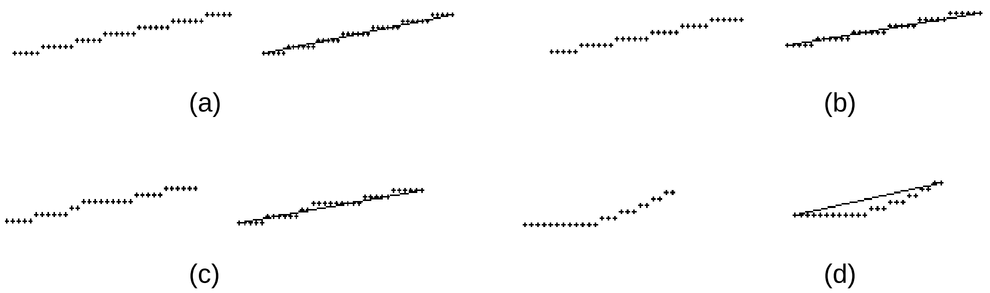

Figure 1. Approximation of Different Digital Segments (shown with dots only) by Asymmetric Digital Straight Segment (Line Segment Overlaid on Dots). The Bottom-Right Approximation has the Highest Deviation in the Digital Curve from its Approximation

1(d) are not ADSS (please see caption for detail), rather each of these curves consists of multiple ADSS, but each of these curves is detected as a single AsymDSS. It is evident from figures 1(c) and 1(d), an AsymDSS may deviate from a digital curve more than an ADSS, resulting in a coarse approximation of the curve and hence higher error in approximation, but AsymDSS does not require ad hoc parameter. As seen from Figure 1(d), the approximation error may be significantly high. So AsymDSS should be subjected to splitting at a location on the digital curve furthest from the AsymDSS.

It may be noted from the above paragraph that detection of AsymDSS needs comparison of chain-code only and no arithmetic operation is required as in the detection of ADSS. Moreover, ADSS detection requires an ad hoc choice of parameter t whereas detection of AsymDSS does not require any such parameter. Since an AsymDSS may contain multiple ADSS hence it produces a coarser approximation than that produced by ADSS detection and is a faster decomposition technique than that of ADSS. Since ADSS detection results in too many tiny segments hence it requires an immediate merging so as to merge quasi linear vertices. AsymDSS on the contrary, produces a coarser approximation and thereby it produces a visually good approximation of a digital curve (see experimental results) in most of the scenarios.

3. Merging Quasi Linear Vertices

Starting from an arbitrary vertex of polygon, generated by AsymDSS, vertices are merged one after another with an initial vertex until merging is no longer possible. The metric used for merging is the distance of a vertex to the line segment joining the initial vertex to the last vertex being considered for merging. If the maximum of the distances of the vertices from the line segment exceeds a threshold then all the vertices preceding the vertex at the maximum deviation are merged and the process is repeated using this vertex (with maximum deviation) as initial vertex and is continued beyond the starting vertex. If the starting vertex happens to be a vertex after merging then the merging terminates, otherwise, the process is continued until the index of the vertex generated last coincides with that of a vertex already generated through merging. If the digital curve is open, instead of being closed, then the merging should start at one end point and end at the other. The two end points of the curve also belong to the set of vertices of the poly-line approximation (since the curve is open hence the approximation is called poly-line approximation) of the curve. For closed digital curve, merging can be started from any of the vertices generated by AsymDSS and hence merging is independent of the starting vertex.

The metric used for testing quasi linearity of consecutive vertices for merging the same is usually maximum perpendicular distance of a vertex from the approximating line segment, computed using floating point arithmetic. But this metric may miss corners and round off sharp turnings. Dunham [5] used distance to a line segment instead of perpendicular distance. Though this metric does not round off corners and preserves sharp turnings, but it needs floating point arithmetic. This paper introduces a measure for distance of a point from a line segment so that corners are not missed, sharp turnings are preserved and computation can be performed in integral domain.

Consider three successive vertices namely, V_k, V_{k+1}, V_{k+2} and it is required to determine whether V_{k+1} is sufficiently close to the line segment V_kV_{k+2}. If V_k and V_{k+2} coincide (Figure 2(a)) then to compute the distance of V_{k+1} from the degenerate segment V_kV_{k+2}, compute the distance of V_{k+1}.

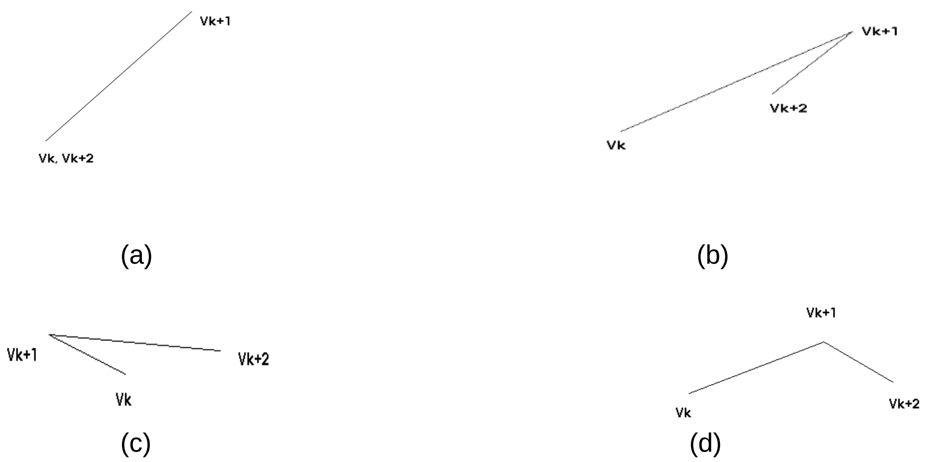

Figure 2. Measuring Distance of V_{k+1} from the Segment V_kV_{k+2} (for Detail see the Text)

from V_k (V_{k+2}) and compare it with a threshold τ . If this distance does not exceed τ then V_{k+1} is merged with the segment V_kV_{k+2}. If V_k and V_{k+2} do not coincide and the orthogonal projection of V_{k+1} is found to be farther away from V_k than V_{k+2} (Figure 2(b)) then the distance between the vertices V_{k+1} and V_{k+2} is considered as a measure of co-linearity of V_{k+1} with the segment V_kV_{k+2}. On the contrary, if the orthogonal projection of V_{k+1} on the line joining V_k and V_{k+2} is found to be farther away from V_{k+2} than V_k (Figure 2(c)) then the distance between V_k and V_{k+1} is the measure of co-linearity of V_{k+1} with respect to the segment V_kV_{k+2}. If neither of these conditions holds then the orthogonal projection of the vertex V_{k+1} on the line joining V_k and V_{k+2} falls within the line segment V_kV_{k+2} and the perpendicular distance of V_{k+1} from the line joining V_k and V_{k+2} is a measure of co-linearity of V_{k+1}.

When the orthogonal projection of V_{k+1} is father away from V_k than V_{k+2} (Figure 2(b)) then the dot product of the vectors $V_{k+2}V_{k+1}$ and V_kV_{k+2} is positive and in this case the distance between V_{k+2} and V_{k+1} is a measure of co-linearity of V_{k+1} with respect to the segment V_kV_{k+2}, on the contrary, when the orthogonal projection of V_{k+1} is farther away from V_{k+2} than V_k (Figure 2(c)) then the dot product of the vectors V_kV_{k+1} and $V_{k+2}V_k$ is positive and in this case the distance between V_k and V_{k+1} is a measure of co-linearity of V_{k+1} with respect to the segment V_kV_{k+2}. When the orthogonal projection of V_{k+1} lies between V_k and V_{k+2} (Figure 2(d)), these dot products are not positive and in this case the perpendicular distance of V_{k+1} from the line joining V_k and V_{k+2} is a measure of co-linearity. The perpendicular distance of V_{k+1} from the line joining V_k and V_{k+2} is defined by

$$d = \frac{|(y_{k+2} - y_k)(x_{k+1} - x_k) - (x_{k+2} - x_k)(y_{k+2} - y_k)|}{\sqrt{(y_{k+2} - y_k)^2 + (x_{k+2} - x_k)^2}}, \tag{1}$$

(x_k, y_k) being the coordinates of a vertex V_k.

The above computation involves floating point arithmetic including square root operation in the denominator. So it is proposed to approximate the denominator of (1) either by isothetic distance (right hand side of (2)) or by city block distance (right hand side of (3)) *i.e.*, one of the following approximations $(\approx)$ is proposed.

$$\sqrt{(y_{k+2} - y_k)^2 + (x_{k+2} - x_k)^2} \approx \max(|y_{k+2} - y_k|, |x_{k+2} - x_k|) \tag{2}$$

$$\sqrt{(y_{k+2} - y_k)^2 + (x_{k+2} - x_k)^2} \approx |y_{k+2} - y_k| + |x_{k+2} - x_k| \tag{3}$$

It may be noted that

$$\sqrt{(y_{k+2} - y_k)^2 + (x_{k+2} - x_k)^2} \geq \max(|y_{k+2} - y_k|, |x_{k+2} - x_k|), \tag{4}$$

but

$$\sqrt{(y_{k+2} - y_k)^2 + (x_{k+2} - x_k)^2} \leq | y_{k+2} - y_k | + | x_{k+2} - x_k | \tag{5}$$

So an upper bound on isothetic distance (right hand side of (4)) will impose an upper bound on Euclidean distance (left hand side of (4)). Replacing the denominator of the perpendicular distance d in (1) by isothetic distance results in a metric of the form

$$d' = \frac{| (y_{k+1} - y_k)(x_{k+2} - x_k) - (x_{k+1} - x_k)(y_{k+2} - y_k) |}{\max(| x_{k+2} - x_k |, | y_{k+2} - y_k |)} \tag{6}$$

If an upper bound (a threshold) τ is imposed on d' then this in turn, will also impose an upper bound on d and so instead of comparing d' against τ (a floating point comparison), the following comparison in the integral domain is proposed.

$$| (y_{k+1} - y_k)(x_{k+2} - x_k) - (x_{k+1} - x_k)(y_{k+2} - y_k) | \leq \max(| x_{k+2} - x_k |, | y_{k+2} - y_k |)\tau \tag{7}$$

If the orthogonal projection of V_{k+1} falls within the segment $V_k V_{k+2}$ then in order to decide whether the vertex V_{k+1} is quasi linear (and hence can be merged) with the vertices V_k and V_{k+2} the metric in (7) is used. But the co-linearity is measured by the isothetic distance between V_k and V_{k+1} when V_k and V_{k+2} coincides (Figure 2(a)), between V_{k+1} and V_{k+2} when the orthogonal projection of V_{k+1} is farther away from V_k than V_{k+2} (Figure 2(b)) and between V_k and V_{k+1} when the orthogonal projection of V_{k+1} is farther away from V_{k+2} than V_k (Figure 2(c)).

4. Vertex Insertion

Because of the presence of one or more very long non singular run lengths on AsymDSS, the curve may deviate significantly from the approximating line segment (Figure 1(d)) and so it may be necessary to split a segment into multiple ones. The splitting is performed at a point most distant from the approximating line segment. In most of the experiments carried out in this paper, this step hardly resulted in a significantly different approximation. But singular cases do exist where a large deviation of the digital curve from the approximating line segment may produce significantly high approximation error. The deviation may be a maximum at any point of the digital curve depending on the distribution of the non-singular run lengths along it. So it is necessary to make an attempt to decompose a line segment at a point on the digital curve furthest from the segment.

In order to insert vertices between a pair of consecutive vertices V_k and V_{k+1}, the deviation of all points (x_j, y_j) of the digital curve intermediate of V_k and V_{k+1} is computed and the maximum of these deviations is compared with a threshold τ. The deviation at a point (x_j, y_j) is defined by

$$\eta_j = (y_{k+1} - y_k)(x_j - x_k) - (x_{k+1} - x_k)(y_j - y_k) - \max(| y_{k+1} - y_k |, | x_{k+1} - x_k |)\tau$$

(8).

The second term on the right hand side of (8) is isothetic distance and is used here as an approximation of Euclidean distance for reasons stated in the last section. The maximum of these deviations over all points intermediate of V_k and V_{k+1} is computed and tested to find out whether it is positive and if so, the point at which maximum deviation occurs is a new vertex that should be inserted on the digital curve intermediate of V_k and V_{k+1}. New vertices are inserted repeatedly in each segment until no more insertion is feasible.

5. Time Complexity

Since a sequential scan of the chain codes is performed by AsymDSS (processing each chain code once only) hence the time complexity of the AsymDSS is linear with the data size. This complexity is the same as that of ADSS [14]. At the merging stage, only the vertices generated by AsymDSS are considered and hence the worst case complexity of this phase is of the order of input data size. At the stage of vertex insertion, it is necessary to visit points of the curve intermediate of the vertices generated by AsynDSS perhaps more than once. If m be the number of intermediate points of the two successive vertices V_k and V_{k+1} and if no vertex is required to be inserted between V_k and V_{k+1} then each of the m points is processed only once. On the contrary, if we assume that it one vertex V' (say) is required to insert and there happens to be m' points intermediate of V_k and V' then each of m points is processed once and additionally $m-1-m'$ points of the m points are processed once more. Thus the total number of points processed is $2m-1-m'$. In general, if it is necessary to insert $\upsilon(>0)$ vertices between V_k and V_{k+1} then the number of times m points are processed is $\upsilon m-\upsilon-m\sum_{j'=1}^{\upsilon}\delta_{j'}$, where $0<\delta_{j'}<1$. Since m is only a fraction of n and υ is never of the order of m hence the processing time of the curve segment from the vertex V_k till the vertex V_{k+1} linear with m. So the time complexity of the vertex insertion phase is also linear with data size n. Since the time complexity of AsymDSS and that of merging is also linear with data size hence the time complexity of the proposed method is linear with data size.

6. Fidelity of Approximation

The measures used for fidelity of polygonal approximation are usually compression ratio, maximum error, integral square error and figure of merit. If a digital curve has n points that result in an approximation consisting of m vertices then the compression ratio is defined by n/m. The approximation error e_i of a point p_i from its nearest polygonal side is its perpendicular distance from the side and maximum error is the maximum of these distances. The integral square error is the sum of squares of these errors. Since a trade-off exists between approximation error and compression ratio hence figure of merit [15] defined by the ratio of compression ratio to integral square error expressed in percentage is used to compare the polygonal approximation of a curve generated by two different techniques. In case, integral square error is found to be zero then the figure merit is undefined and the approximation is regarded as a trivial approximation.

7. Experimental Results

The technique developed here has been tested extensively on a large number of digital curves and some of these results are shown in this paper. The Figure 3 through 10 shows the polygonal approximation of different digital curves [16] overlaid on the input curve obtained using the proposed method and those obtained using the technique introduced in [14]. The vertices are marked with dots bigger than those of digital points. In each figure, the top most row show the approximations obtained by the proposed method, the second row shows the ones obtained by decomposing the curve using ADSS and then merging ADSS using maximum error and the third row show the same using cumulative error. Each row shows approximations for three different values of tolerance parameter $\tau = 1, 2, 3$ from left to right.

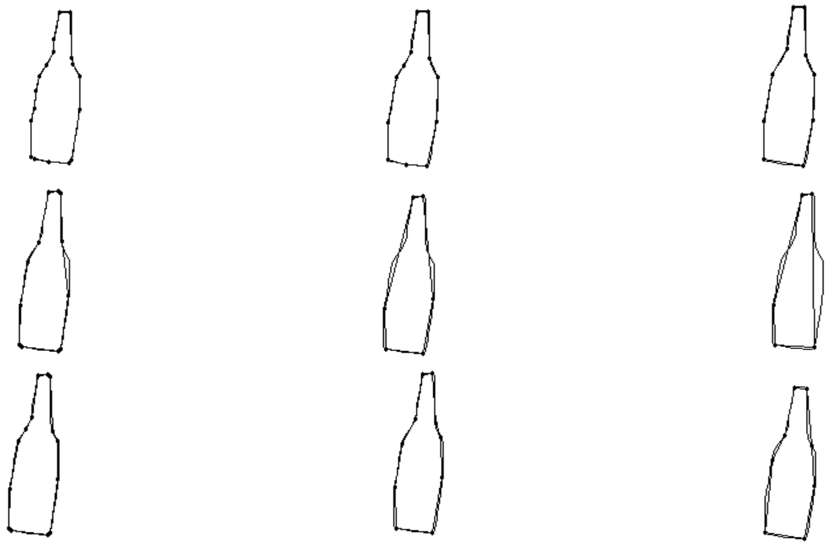

Figure 3. Digital Boundary of a Bottle and its Polygonal Approximation

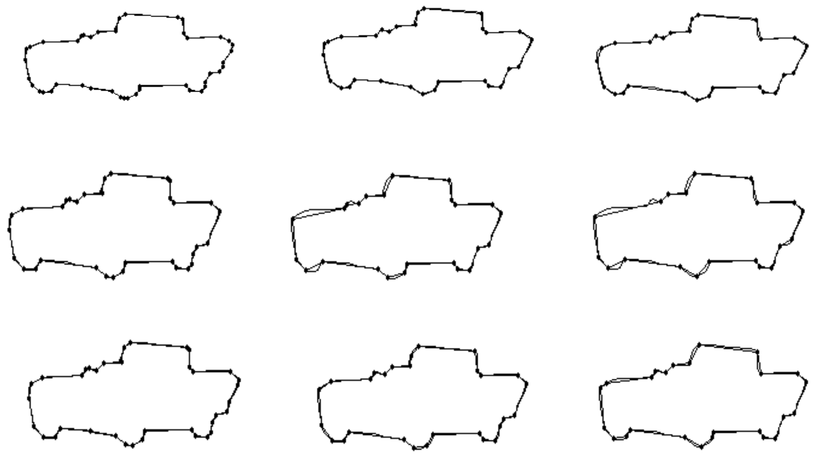

Figure 4. Digital Boundary a Car and its Polygonal Approximation

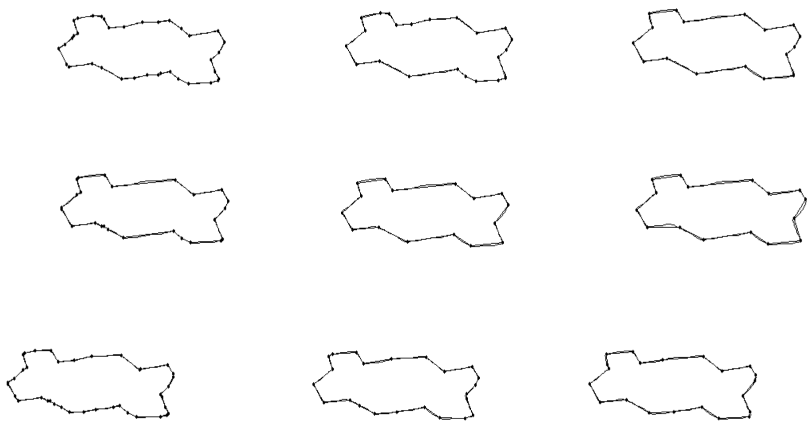

Figure 5. Digital Boundary of a Brick and its Polygonal Approximation

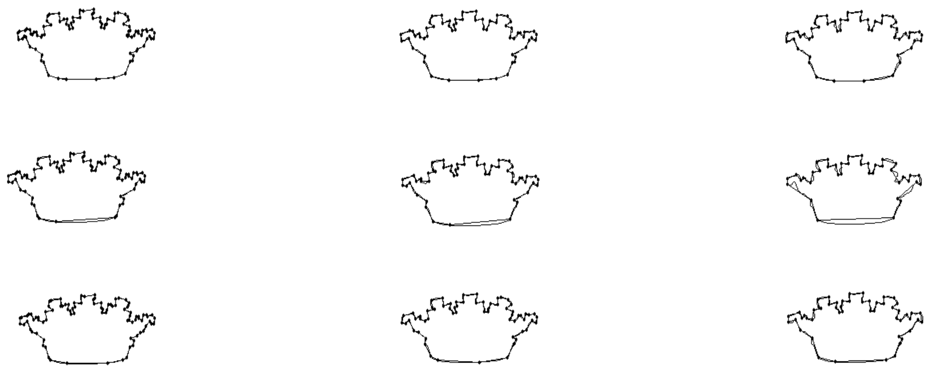

Figure 6. Digital Boundary of a Crown and its Polygonal Approximation

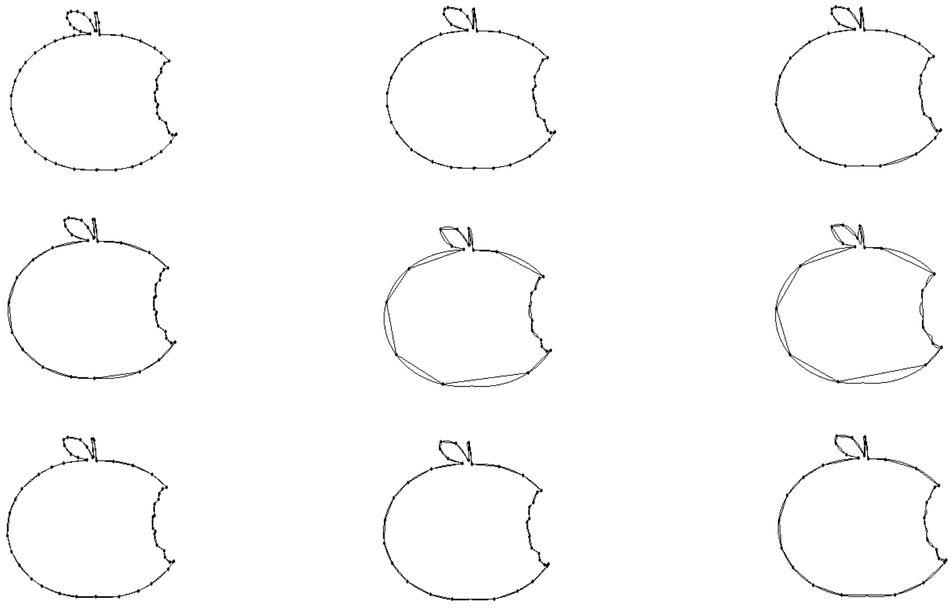

Figure 7. Digital Boundary of a Bitten Apple and its Polygonal Approximation

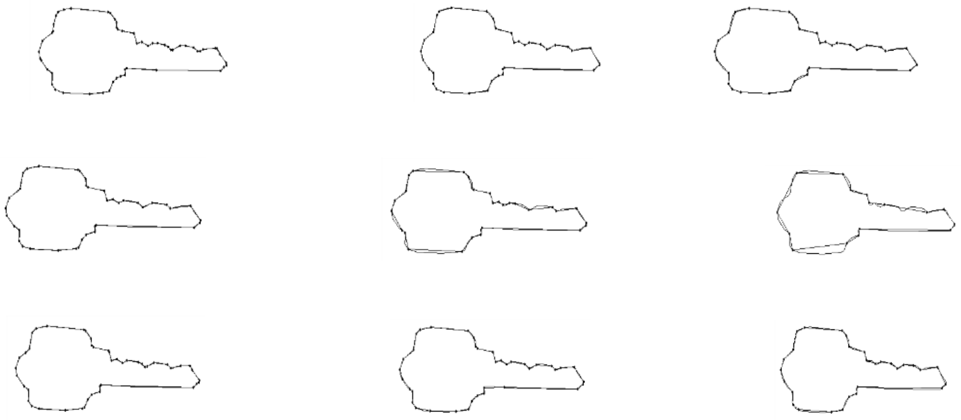

Figure 8. Digital Boundary of a Key and its Polygonal Approximation

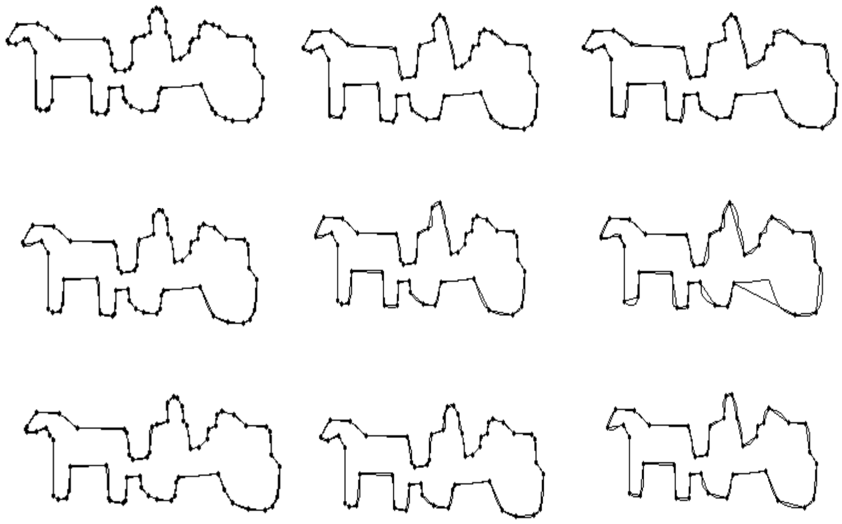

Figure 9. Digital Boundary of a Carriage and its Polygonal Approximation

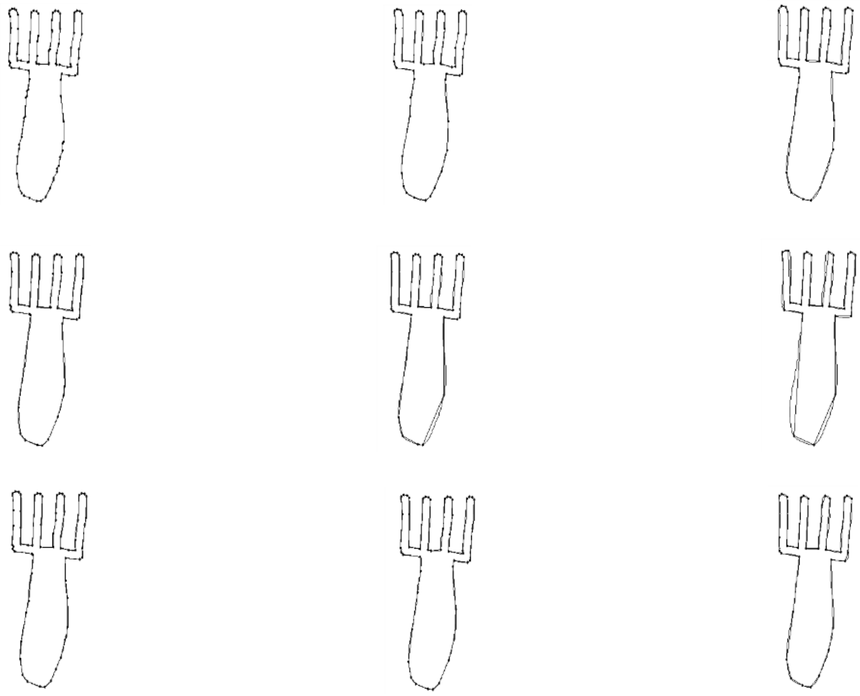

Figure 10. Digital Boundary of a Fork and its Polygonal Approximation

The fidelity of the approximations namely, number of vertices(m), maximum error(E_{∞}), integral square error(E_2), compression ratio(CR) and figure of merit (FoM, expressed in percentage) for different values of the tolerance parameter τ are shown in Table I.

It may be observed from the figures as well as from the table that the approximation error (maximum error and integral square error) produced by the proposed method is lower than that produced by applying ADSS followed by merging [14] and the proposed method detects higher number of vertices than by [14], and the figure of merit of the approximations produced by the proposed method is higher than that by [14]. This difference is found to be more significant at the higher values of the tolerance parameter τ. The ADSS followed by merging using sum of area criterion performs better than those produced using maximum error criterion and use of either criterion is outperformed by the proposed method. As evident from the figures 3 through 10, sharp turnings as well as many not so sharp turnings are rounded by [14] whereas it is not so in the proposed method. The quality of approximation based on visual perception is observed to be better than those produced by [14].

Table 1. Number of Vertices (*m*), Maximum Error (E_∞), Integral Square Error (E_2), Compression Ratio(CR) and Figure of Merit (FoM) of Polygonal Approximation of Different Digital Curve for Values of τ = 1, 2, 3

Digital curve	Number of points	T	Fidelity measures of polygonal approximation														
			Split-merge-split					ADSS & Merging									
								Maximum error criterion					Cumulative error criterion				
			m	E_∞	E_2	CR	FoM	*m*	E_∞	E_2	CR	FoM	*m*	E_∞	E_2	CR	FoM
Bottle (Figure 3)	327	1	18	1.00	50.62	18.17	35.89	12	4.28	410.55	27.25	6.64	14	1.72	168.28	23.36	13.88
		2	12	1.87	113.93	27.25	23.92	6	3.74	850.50	54.50	6.41	10	2.25	344.80	32.70	9.48
		3	10	2.15	161.43	32.70	20.26	5	8.13	4014.32	65.40	1.63	8	3.12	740.15	40.88	5.52
Car (Figure 4)	443	1	45	1.00	72.74	9.84	13.53	34	1.85	135.83	13.03	9.59	37	1.48	114.19	11.97	10.48
		2	30	1.64	161.09	14.77	9.17	24	4.96	846.15	18.46	2.18	28	2.06	240.90	15.82	6.57
		3	27	2.65	262.59	16.41	6.25	20	4.57	854.91	22.15	2.59	25	2.79	488.54	17.72	3.63
Brick (Figure 5)	500	1	37	0.99	71.32	13.51	18.95	25	2.73	407.60	20.00	4.91	39	1.27	117.38	12.82	10.92
		2	22	1.77	155.89	22.73	14.58	16	2.99	629.30	31.25	4.97	23	2.22	300.60	21.74	7.23
		3	17	2.34	336.93	29.41	8.73	16	4.00	1069.74	31.25	2.92	21	2.26	415.91	23.81	5.72
Crown (Figure 6)	570	1	94	1.00	86.90	6.06	6.98	83	3.59	569.93	6.87	1.20	93	1.68	154.75	6.13	3.96
		2	61	1.81	205.34	9.34	4.55	55	4.74	1158.65	10.36	0.89	64	2.13	333.05	8.91	2.67
		3	51	3.00	413.32	11.18	2.70	39	7.49	3883.54	14.62	0.38	52	3.16	545.86	10.96	2.01
Bitten apple (Figure 7)	697	1	61	1.00	87.96	11.43	12.99	40	3.54	874.5	17.43	1.99	50	1.70	254.54	13.94	5.48
		2	40	2.00	235.47	17.43	7.40	24	10.62	12694.57	29.04	0.23	35	2.15	586.38	19.91	3.40
		3	32	2.85	594.51	21.78	3.66	18	12.52	16463.77	38.72	0.24	29	2.68	1051.73	24.03	2.29
Key (Figure 8)	791	1	56	1.00	117.21	14.12	12.05	39	1.93	314.5	20.28	6.46	46	1.71	255.24	17.20	6.74
		2	35	1.93	356.97	22.60	6.33	29	5.41	3029.66	27.28	0.90	38	2.32	464.93	20.82	4.48
		3	31	2.53	501.44	25.52	5.09	16	14.08	15866.56	49.44	0.31	30	2.85	1513.58	26.37	1.74
Carriage (Figure 9)	729	1	78	0.98	98.53	9.35	9.49	58	2.09	218.76	12.57	5.75	64	1.77	204.29	11.39	5.58
		2	48	1.94	338.02	15.19	4.49	44	3.44	646.38	16.57	2.56	51	2.18	461.50	14.29	3.10
		3	42	2.63	513.89	17.36	3.38	34	16.24	6801.14	21.44	0.32	38	2.82	836.34	19.18	2.29
Fork (Figure 10)	1322	1	97	1.00	180.89	13.63	7.53	47	2.78	949.76	28.13	3.42	66	1.74	552.94	20.03	3.62
		2	53	1.92	508.62	24.94	4.90	33	6.05	3298.71	40.06	1.64	50	2.27	853.04	26.44	3.10
		3	34	3.00	1568.27	38.88	2.48	23	9.60	9448.94	57.00	0.60	33	3.16	1620.46	40.06	2.47

8. Conclusion

The AsymDSS has been introduced relaxing the definition of DSS followed by sequential merging and insertion of additional vertices. Merging is performed using an integral form of comparison of perpendicular distance and distance to a point so as to retain corners and sharp turnings whereas for vertex insertion only the former metric is used. It is observed that the approximation produced by the proposed method is better than those produced by [14]. It is possible to add an enhancement to this work producing symmetric approximation from symmetric digital curve if operations are performed twice traversing the curve once in clockwise direction and then in the counter clockwise direction.

References

[1] U. E. Ramer, “An iterative procedure for polygonal approximation of plane curves”, Computer Graphics and Image Processing, vol. 1, **(1972)**, pp. 244-256.
[2] T. Pavlidis and S. L. Horowitz, “Segmentation of plane curves”, IEEE Transactions on Computer, vol. 23, **(1974)**, pp. 860-870.
[3] C. Williams, “An efficient algorithm for piecewise linear approximation of planar curves”, Computer Graphics and Image Processing, vol. 8, **(1978)**, pp. 286-293.
[4] K. Wall and P.-E. Danielsson, “A fast sequential method for polygonal approximation of digitized curves”, Computer Vision, Graphics and Image Processing, vol. 28, **(1984)**, pp. 220-227.
[5] J. G. Dunham, “Optimum piecewise linear approximation of planar curves”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, **(1986)**, pp. 67-75.
[6] J. C. Perez and E. Vidal, “Optimum polygonal approximation of digitized curves”, Pattern Recognition Letters, vol. 15, **(1994)**, pp. 743-750.
[7] A. Pikaz and I. Dinstein, “An algorithm for polygonal approximation based on iterative point elimination”, Pattern Recognition Letters, vol. 16, **(1995)**, pp. 557-563.
[8] P. Zhu and P. M. Chirlian, “On critical point detection of digital shapes”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI – 17, **(1995)**, pp. 737-748.
[9] M. Visvalingam and J. D. Whyatt, “Line generalization by repeated elimination of points”, Cartographic Journal, vol. 30, **(1993)**, pp. 46-51.
[10] P. Y. Yin, “Genetic particle swarm optimization for polygonal approximation of digital curves”, Pattern Recognition and Image Analysis, vol. 16, **(2006)**, pp. 223-233.
[11] B. Wang, H. Shu and L. Luo, “A genetic algorithm with chromosome-repairing for min - # and min - ε polygonal approximation of digital curves”, Journal of Visual Communication and Image Representation, vol. 20, **(2009)**, pp. 45-56.
[12] H. Freeman, “Techniques for the digital computer analysis of chain-encoded arbitrary plane curves”, Proceedings of the National Electronics Conference, vol. 7, **(1961)**, pp. 421-432.
[13] A. Rosenfeld, “Digital straight line segments”, IEEE Transactions on Computers, vol. 23, **(1974)**, pp. 1264-1268.
[14] P. Bhowmick and B. Bhattacharya, “Fast polygonal approximation of digital durves using relaxed straightness properties”, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI, vol. 29, **(2007)**, pp. 1590-1602.
[15] D. Sarkar, “A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves”, Pattern Recognition Letters, vol. 14, **(1993)**, pp. 959-964.
[16] http://www.dabi.temple.edu/~shape/MPEG7/dataset.html.

Author

Bimal Kumar Ray received B.Sc. Honours degree in Mathematics from St. Xavier`s College, Kolkata, M. Sc. degree in Applied Mathematics from Calcutta University, Kolkata and Ph.D. degree in Computer Science from Indian Statistical Institute, Kolkata. His areas of interests are Computer Graphics, Vision and Image Processing. He has a number of published papers to his credit in peer-reviewed journals.

