Elektrotehniški vestnik 73(2–3): 84–92, 2006
Electrotechnical Review, Ljubljana, Slovenija

Reconstructing 3D Curves with Euclidean Minimal Spanning Trees

Simon Kolmanič, Nikola Guid

University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova ulica 17, Maribor, Slovenia

E-mail: simon.kolmanic@uni-mb.si, guid@uni-mb.si

Abstract. In this paper, we present a new efficient algorithm for reconstruction of nonintersecting 3D curves from a sufficiently dense sample. We use the Euclidean minimal spanning trees to identify line segments reconstructing curve shapes. To deal with more than one curve in a sample and to eliminate noisy data, we introduce chains of connected line segments. With the incremental growth based on heuristics, the chains contain finally curve shapes. The method is robust and fast for both 2D and 3D curves.

Key words: point cloud, curve reconstruction, Euclidean minimal spanning trees

Rekonstrukcija prostorskih krivulj s pomočjo evklidskih minimalnih vpetih dreves

Povzetek. V članku predstavljamo nov učinkovit algoritem za rekonstrukcijo prostorskih krivulj iz dovolj gostega vzorca. S pomočjo evklidskih minimalnih vpetih dreves poiščemo tiste daljice, ki rekonstruirajo krivuljo. Za delo z več krivuljami v vzorcu in odstranitev točk šuma uporabljamo strukturo, ki jo imenujmo verige povezanih daljic. Z inkrementalno rastjo, ki temelji na hevristiki, dobimo v verigah iskano rekonstrukcijo krivulj. Predstavljena metoda je robustna in hitra tako pri rekonstrukciji ravninskih kot tudi prostorskih krivulj.

Ključne besede: oblak točk, rekonstrukcija krivulj, evklidska minimalna vpeta drevesa

1 Introduction

Suppose that a cloud of points obtained by sampling of a collection of arbitrary 3D curves is given. The number of curves is not known. A curve sample in general does not contain the same number of points and the distribution of points in a sample may vary. If the sampling is dense enough, it is an easy task for the human to find these curves and perceive their shapes. For a computer, however, the task is much harder. From the cloud of points, the computer must find those points that belong to the same curve and then to sort them in a correct order, compatible with the original trace of the curve. In this case, the curve is reconstructed.

Received 4 October 2005
Accepted 30 January 2006

There are many known methods for curve reconstruction in the plane [1, 3, 4, 8-10, 12-14, 18]. Most of these methods are based on the Delaunay triangulation, therefore, they do not work or run much slower if a 3D curve has to be reconstructed. Exceptions are methods presented in [1] and [8]. In [1], the curve reconstruction is given by a travelling salesman route while the reconstruction in [8] is based on the nearest neighbour graph [17]. Therefore, we search for a solution of reconstruction of 3D curves in the graph theory. In contrast to [1, 8], we focused on the minimal spanning trees, which were used in the early seventies for pattern recognition and cluster analysis [19, 11]. The advantage of minimal spanning trees is in their independence on the space dimension where the sample is given, and in relatively simple algorithms for their computation. In [2], methods [1, 3, 4, 8-10, 13] are classified as algorithms with guaranteed performance, i.e., algorithms that provably reconstruct curves under certain assumptions. In [13], the formal proof can be found that the Euclidean minimal spanning trees correctly reconstruct differentiable arcs from sufficiently dense samples. The correct reconstruction, also called polygonal reconstruction, $G(S, \Gamma)$ of a sample S with respect to a collection of the nonintersecting curves Γ is defined as a graph G with the vertex set S having exactly those edges that connect sample points adjacent in Γ [3, 8, 14]. In this paper, we use the definition for a dense sample given by Figueiredo and Gomes [13]:

a sample is sufficiently dense if there is a real positive number ε such that no two consecutive sample points are more than ε apart and closed discs of the radius ε, centered at the sample points, form a tubular neighbourhood of the single curve C.

It is clear that the Euclidean minimal spanning trees cannot give the correct reconstruction of a collection of nonintersecting curves without additional heuristics since the polygonal reconstructions of individual curves are connected to each other with edges, which Figueiredo terms bridges (see Fig. 1b). Only if all the bridges are eliminated, the correct reconstruction of sample S is given. To do that, we introduce a structure we term a chain of connected line segments. This is a list of line segments giving the correct reconstruction of one curve sorted in the right order. With special criteria for adding a line segment to the chain, we achieve that all the bridges form their own chains, which have to be eliminated at the end of the reconstruction process. The chains of the connected line segments are not only used to give us the correct reconstruction of single lines, but also to reduce the noise significantly. The paper is organised in seven sections. In Section 2, we introduce the Euclidean spanning trees as a possible solution of the curve reconstruction and problems connected with it. A chain of connected line segments with its incremental growth and corresponding heuristics is presented in Section 3. In Section 4, we consider the handling of closed curves and noise reduction. The procedure for speeding up the reconstruction process is described in Section 5. The experimental results are shown in Section 6 and the last section presents a conclusion.

2 Euclidean minimal spanning trees

The computation of the correct reconstruction of curves, i.e. $G(S, \Gamma)$, is possible only if a sufficiently dense sample determined by the Figueiredo-Gomes criterion is given [13]. In this case, the sample is called an ε-sample. The correct reconstruction $G(S, \Gamma)$ is not only a graph having the same topology as the curve, but its vertices are the points lying in the plane or space and the edges are line segments connecting the points into the polygonal approximation of the curve. Therefore, $G(S, \Gamma)$ is also a *geometric graph.* If we introduce line segment lengths in $G(S, \Gamma)$ as weights of its edges, the graph $G(S, \Gamma)$ becomes a weighted geometric graph. The *minimal spanning tree* (MST) for a weighted graph is a spanning tree for which the sum of edge weights is minimal. The geometric version of the minimal spanning tree is called the *Euclidean minimal spanning tree* (EMST). In the early seventies, EMST was used for solving of the particle track problem, where the shape of a single curve in a noisy sample had to be found, and for a cluster analysis [19, 11]. Finally in [13, 1], we can find the formal proof that the Euclidian minimal spanning trees can correctly reconstruct differential arcs from a sufficiently dense sample. Although Figueiredo considers only plane curves, his proof can easily be extended to 3D curves. This is not surprising since Zahn pointed out already that the Euclidean minimal spanning tree could be used in higher dimensional spaces or, in fact, in general metric spaces [19]. Besides, the minimal spanning trees are interesting because they are well understood and easily computed. In this paper, we use the simplest but by far not an optimal algorithm for computation of the minimal spanning tree - the Kruskal algorithm [7, 16]. If $G(S, \Gamma)$ is a geometric graph, the Kruskal algorithm [7] returns the edges of the Euclidean minimal spanning tree. The first step to obtain EMST for a sample S is to generate the geometric graph $G(V, E)$, where V are the points of the sample S and E are line segments connecting those points. Next, the algorithm for computing the minimal spanning tree has to be started to obtain the line segments reconstructing the curve shape. If S is a dense sample of an open single curve C, EMST returns all the edges forming correct reconstruction, but generally, we have to find their correct order. If S is a sample of several nonintersecting curves, however, the problem is much harder. With EMST, we get a set of the line segments that reconstruct the shape of curves. Then, we have to classify which line segments belong to which curve and finally the correct order of lines has to be determined. The sample of nonintersecting curves and the corresponding set of the line segments obtained by EMST can be seen in Fig. 1.

In Fig. 1b, the line segments 1-5, which are a part of EMST, are marked, but they are not a part of the original curves. Figueiredo calls those line segments bridges and they have to be removed in order to get the correct reconstruction. He proposes the same method Zahn already mentioned in [18]. Those line segments are removed according to the combination of their lengths and the degree of their edge points, where the degree of a point is given by a number of line segments meeting at that point. This works well if only the sample of one curve with noisy data is given or if the lengths of the line segments, representing the correct reconstruction, are nearly the same. In Fig. 1b, the problem of the bridge removal is much harder. The bridges 3 and 5, for example, cannot be removed according to the Zahn criterion because they are shorter than the line segments reconstructing the ellipse and their endpoints do not

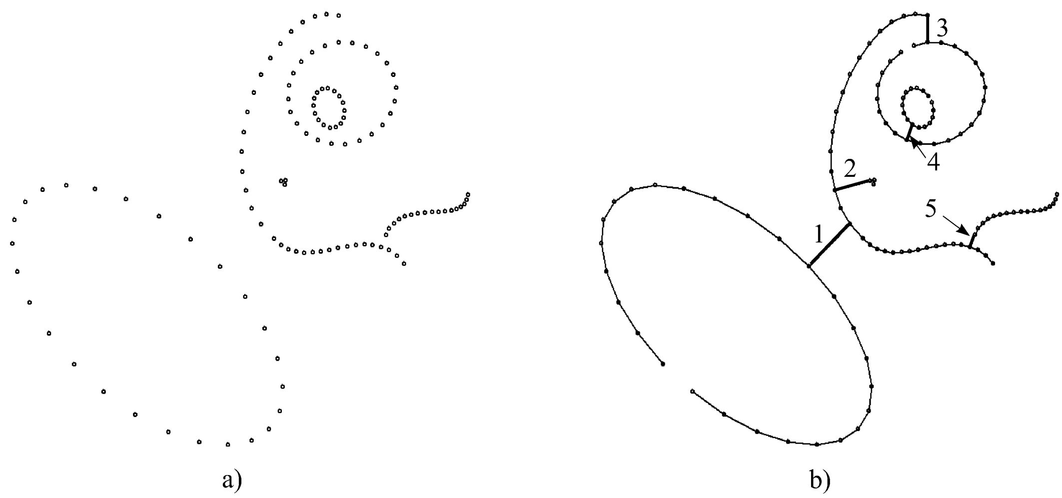

Figure 1. Implementation of EMST in the curve reconstruction: a) Sample of nonintersecting curves, b) Set of the line segments obtained by EMST with marked "bridges"

have degrees of 3 and 1. To solve similar problems and to efficiently classify the line segments to appropriate curves, we propose a structure we have named a *chain of connected line segments.*

3 The chain of connected line segments

The chain of connected line segments is a sequence of equally oriented vectors $\mathbf{v}_i$, where the end point of the vector $\mathbf{v}_i$ is at the same time the starting point of the vector $\mathbf{v}_{i+1}$ (see Fig. 2). The direction of the vector $\mathbf{v}_i$ is defined by the sequence of vertices defining the line segment i, accepted by the algorithm for computing EMST. The requirement for the same orientation in the chain assures that the points belonging to the same polygonal reconstruction are sorted in the right order. In the reconstruction of the sample of several nonintersecting curves, each chain represents one of the curves.

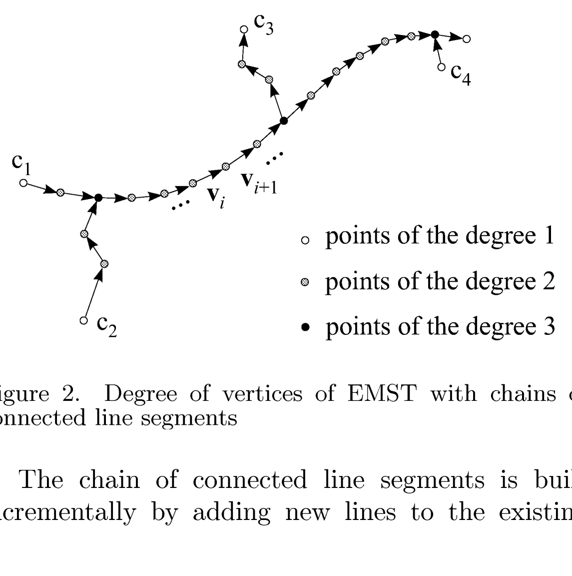

Figure 2. Degree of vertices of EMST with chains of connected line segments

The chain of connected line segments is built incrementally by adding new lines to the existing chains, since there are usually more than one chain in the reconstruction process. The line segments, accepted by the algorithm for computation of EMST, have to be sorted in the increasing order according to their lengths. This is done automatically by the Kruskal algorithm [7], but they have to be sorted first by other minimal spanning trees generation algorithms. Since their number is relatively small (n - 1, where n is the number of points in the sample) regarding to the number of edges in E, the sorting does not represent a significant time lost.

The chain orientation helps us to define a starting vertex and an ending one of the chain. Only those two vertices in the chain can have the degree 1 while all the others have the degree 2 at least. Therefore, a new line segment can be added to the chain only if it has one common vertex at the beginning or at the end of the chain and its degree is 1. In other case, the line segment generates a new chain. When the line segment is added to the chain, the degree of their common point is increased by 1. The chains, whose starting and ending vertex have the degree greater than 1, are attached to another chain and usually represent bridges or noisy data. This can be seen in Fig. 2, where four different chains of connected line segments can be seen. The degree of one from the border vertices of the chains c_2, c_3, and c_4 is larger than 1 and, therefore, the chains represent a noise.

We have already mentioned that many chains of connected line segments can be generated in the reconstruction process, since they are built incrementally by adding edges of EMST. Two of the existing chains are combined into a single one if the line segment is added whose endpoints belong to both of these chains where the orientation of the chain with shorter edges is preserved. The described procedure can be seen in Figure 3.

```
procedure AddLineSegment(l){
  i=0; // find the appropriate chain
  while(i<number of chains and not Found){
     if the line segment l can be
       added to i-th chain then{
       Found = true;
       if any of two chains can be merged then
         merge chains;
     }
     else
       i++; // find a new chain
  }
  if not Found then
    generate a new chain containing l;
}
```

Figure 3. Algorithm for adding a line segment to a current chain structure

In the algorithm in Fig. 3, the condition the line segment l can be added to i-th chain can be found. This condition guaranties that all the bridges and noisy line segments generate new chains. The simplest condition, proposed already by Zahn [19], for a line segment to generate a new chain is the degree of its endpoints. If the degree of at least one of the endpoints is greater than 1, the line segment has to be added to a new chain. Although this criterion works fine in most cases, the algorithm for calculation of EMST can accept a bridge that connects two curves/chains of the connected line segments in their edge vertices. In this case, the line segment is accepted due to this simple criterion since the degrees of endpoints of such line segment are equal to 1 (Fig. 4).

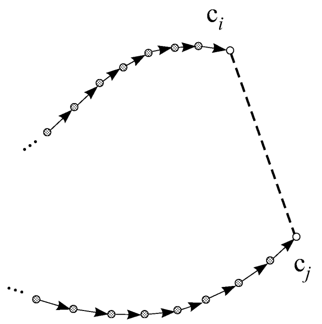

Figure 4. Bridge connecting two chains in their edge vertices

We have observed that such line segments are usually much longer than the line segments already accepted to the chain. Therefore, a logical criterion for rejection of a particular candidate besides the degree of its endpoints would be its length. If the line segment is too long according to the line segments joined into the chain, it is not accepted. But as it has turned out, it is very hard to say when the line segment is too long. If, for example, we compare the length of the candidate to an average length of line segments joined into the chain and their standard deviation, the criterion is too hard and the curve is split in two or more parts. If, on the other hand, we employ the multiple of that average as the criterion, all the bridges may not be removed if the multiplier is not chosen correctly. What makes the problem even harder is the fact that the right multiplier value depends on the number of the line segments in the chain and varies from case to case. Hence, we rather observe a change in the angle between two neighbour line segments instead. In our case, Γ is a collection of isolated points - noise and smooth curves that are pairwise disjoint, where the single curve γ given by the vector function $\mathbf{c}(u)$, $u \in [0, 1]$, is smooth if $\mathbf{c}'(u)$ is continuous and nonzero in [0, 1]. Therefore, the sudden change in the angle between two neighbour line segments cannot occur [1]. Consequently, such a change indicates the bridge that has to be removed (Fig. 4). To perceive the change in the successive angles between neighbour line segments, we use the next recurrence equation for the sample variance:

$$s_{i+1}^2 = \left(1 - \frac{1}{i}\right) s_i^2 + (i+1)(\mu_{i+1} - \mu_i)^2, i = 1, 2, ..., n-1 \tag{1}$$

where $s_1^2 = 0$, μ_i and μ_{i+1} are two successive values of the sample mean, calculated by the following recurrence equation:

$$\mu_{i+1} = \mu_i + \frac{v_{i+1} - \mu_i}{i+1}, \quad i = 1, 2, ..., n-1 \tag{2}$$

where $\mu_1 = \alpha_1$ and $v_i (v_i = \alpha_i)$ is an angle between the vectors $\mathbf{r}_i$ ($\mathbf{r}_i = \mathbf{p}_{i-1} - \mathbf{p}_i$) and $\mathbf{r}_{i+1}$ ($\mathbf{r}_{i+1} = \mathbf{p}_{i+1} - \mathbf{p}_i$), see Fig. 5.

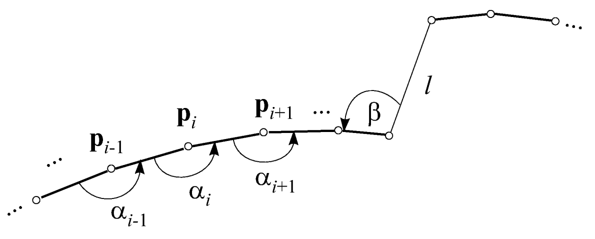

Figure 5. Angles between neighbouring line segments in a chain of connected line segments

As it can be seen in Fig. 5, the angles between neighbouring line segments denoted by α_i change slowly. Hence, the small change of the variance calculated by Eq. 1 occurs. It is not so if the line segment l were added to the chain. The angle β differs much from the other angles in the chain, which causes a large change of the variance (more than 500%), therefore, the line segment l has to be rejected to prevent that. After the line segment is accepted in an existing chain, we have to adjust the orientation of a vector defined by the order of its endpoints. The verification of suitability of line segments and orientation adjusting of the appropriate vector can be done by a single function, shown in Fig. 6.

```
bool AddLineToChain(i, l){
i:  an index of the chain to which the line
    l may be added
l:  a line segment defined by the endpoints
    (p,q)

if valid common edge between the line l and
  the start or the end of the chain i then{

  calculate the angle α between the line l
  and the border line segment;
  if α changes the dispersion of
     the chain too much then
     return false;
  if common edge is q then
     reorient the vector defined by (p,q);

  add the line l to the beginning or
  the end of the chain;
  increase the number of accepted line
  segments in the chain;
  update the length of all line segments;
  return true;
}
return false;
}
```

Figure 6. Algorithm for testing suitability of a line segment

After all edges of EMST have been examined, the set of chains of connected line segments is obtained. This set of chains represents the basis for the removal of bridges and handling of connected curves, described in the next section.

4 Reconstruction from noisy data and reconstruction of closed curves

With the introduction of the chain of connected line segments, the problem of noisy data becomes the problem of identifying noisy chains. By the identifying of noisy chains we suppose that a number of noisy points is much smaller than the number of curve sample points. Because we use the EMST, the noisy points are also connected by the line segments, which form the chains of connected line segments by themselves. Beside these chains we have to remove, there are also chains, which are formed by bridges. Both types of chains are much shorter than the chains reconstructing the shape of curves. Therefore, the criterion for the removal of noisy chains is based on their length, where the length is defined in the number of line segments forming it. As we have searched for a procedure to separate the noisy chains, from the chains that most probably reconstruct the curves shape, we observed, that in general the majority of the chains is much shorter than the few ones we are looking for. To separate them, we calculate the average length of the chains, denoted by $E(N)$, and the standard deviation of these lengths, denoted by σ_N. $E(N)$ is calculated by the equation:

$$E(N) = \frac{1}{m}\sum_{i=1}^{m} N_i. \qquad (3)$$

where m is a number of the chains of connected line segments and N_i is a number of line segments forming the chain i. σ_N is determined by:

$$\sigma_N = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(N_i - E(N))^2}. \qquad (4)$$

The chain i is removed if the following condition is true:

$$N_i < \min\{E(N) + \sigma_N,\ \ N_{max} - 2\sigma_N\}, \qquad (5)$$

where N_{max} is the maximal length of the generated chains. If the curve samples have approximately the same number of sample points, the condition $N_{max} - 2\sigma_N$ works just fine, but if this is not the case, the relatively much shorter chains are wrongfully rejected as a noise. To prevent that, we have expanded the condition with $E(N) + \sigma_N$ as seen in Eg. 5. The result of the noise reduction procedure on the point sample shown in Fig. 1a can be seen in Fig. 7.

The problem of determination whether the polynomial reconstruction has to be closed or not is present in all methods, where both open and closed curves have to be reconstructed at the same time. Reconstructing the curve shape by EMST makes this dilemma even harder since the closing edge cannot be a part of the obtained reconstruction. In literature, we can find several heuristics to determine when the curve has to be closed [13, 19]. We use the similar criterion Zahn did [19], but it is simpler and more accurate with regard to chains of connected line segments. Because of the orientation of the chain of connected line segments, we exactly define the starting and the ending vertices of the chain. Similarly as Zahn proposed [19], we connect them if the distance between them is not too long, but their degrees have to be equal to 1 at the same time. The edge vertices are connected if the following condition applies:

$$d < E(L) + \sigma_L, \quad (6)$$

where d is the distance between edge vertices, $E(L)$ is the average length of line segments in the chain calculated by Eq. 2 and σ_L is the standard deviation calculated by:

$$\sigma_L = s_i, \quad (7)$$

where s_i^2 is the current dispersion of the lengths of the line segments included in the chain calculated by Eq. 1. The result of the described procedure can be seen in Fig. 7.

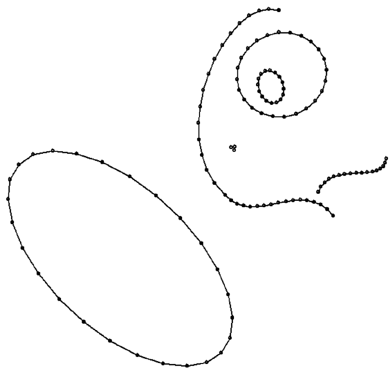

Figure 7. Elimination of noisy data and handling of closed curves

Similarly as the EMST, the above described procedures for the removing of noisy line segments and handling of the closed curves can also be used in higher dimensional spaces without any additional computational time requirements. Apart from this fact, the time requirements of the algorithm must be reduced before it is fit to reconstruct curves from large samples. This procedure is described in the next section.

5 Computing time reduction

Nevertheless, which algorithm for generating minimal spanning tree we use, the computational time needed for the MST generation depends on the number of edges in the starting graph. In our case, we start only with a set of points, therefore, the edges have to be generated. In order not to miss one, we connect all points with each other, what makes $n(n-1)/2$ edges where n is the number of points in the sample. Only the small number of those edges is needed to generate the minimum spanning tree, therefore, decreasing the number of those edges in the initial graph reduces the computational time of the reconstruction. If the long edges are eliminated, the result of reconstruction is still valid since the edges reconstructing curves are shorter than others. To estimate which edge can be eliminated, we use a recurrence formula for the mean edge length calculation (Eq. 2). To get the first approximation of the edge length average, all the edges meeting at an arbitrary point in the sample are taken. After that, only those edges are accepted as the candidates, whose length is shorter than the average length. Since the equation for the mean value calculation is recurrent, the mean reduces with each accepted edge and even more edges are eliminated. In our tests, more than 90% of initial edges in the noisy examples were eliminated. The reduction of the computation time is immense and because only long edges are eliminated, the result of the curve reconstruction remains unchanged. In the next section, we will present experimental results of reconstruction of 3D curves together with the computational time analysis.

6 Experimental results

The described algorithm for the curve reconstruction was implemented and the results are good either if a collection of 2D or 3D smooth curves has to be reconstructed. Since the most of the existing methods are based on Voronoi diagrams, for example [3, 9, 10], they are more efficient in the reconstruction of 2D curves. We compared our results with the ones of [3, 8, 9]. The reconstruction of a collection of smooth 2D curves was in our case similar to other methods mentioned above. But, if a set of both open and closed curves has to be reconstructed, results of our algorithm is more accurate. We must add here that the results of reconstruction in [9] depend on the selection of the value of parameter ρ and are almost identical to ours if ρ is selected correctly. Similarly, we get the identical results as Dey and Kumar [8] by reconstructing an implicit curve helix, but our computation of MST in 3D is simpler than the one of the nearest neighbour graph [17] and it is independent of the space dimension. Besides, the method from [8] is also less accurate if the collection of curves has to be reconstructed. In comparison to [1], based on the travelling salesmen problem (TSP), the reconstruction result of a single smooth curve was similar. In the reconstruction of a single closed curve with sharp corners, however, the method from [1] was superior although, currently, it is not capable to reconstruct the collection of curves. Additionally to [1], we also studied various TSP heuristics given in [6]. Although those heuristics give fast TSP tours that can be used to reconstruct curves, a comparison to our algorithm in their original form is not possible since they can be used only to reconstruct a single closed curve and they connect all points in the given sample, including the noise, into a single shortest tour. We also considered tests presented in [2] according to which the reconstruction results using TSP heuristics are not as good as those achieved with [1]. Our results of a single smooth curve are also equal to those of [13, 19], but the heuristics in [19] are too weak to eliminate all bridges if more than one curve has to be reconstructed, therefore, some curves can be wrongly reconstructed as a single curve. In examples with the noise, the error in reconstruction with [13, 19] is even larger. Although we add heuristics to the algorithm that originally provably reconstruct the single smooth arc in order to distinguish between particular curves in the sample and to eliminate the noise, the reconstruction results in our experiments remained still close to the results of algorithms with guaranteed performance. In the worst case, if some of our heuristics fail, a curve can be broken in two or more parts and shorter parts can be then eliminated as a noise, but the parts that remain due to [13] provably reconstruct the shape of a curve as long as it is smooth. The described procedure was tested mostly on the

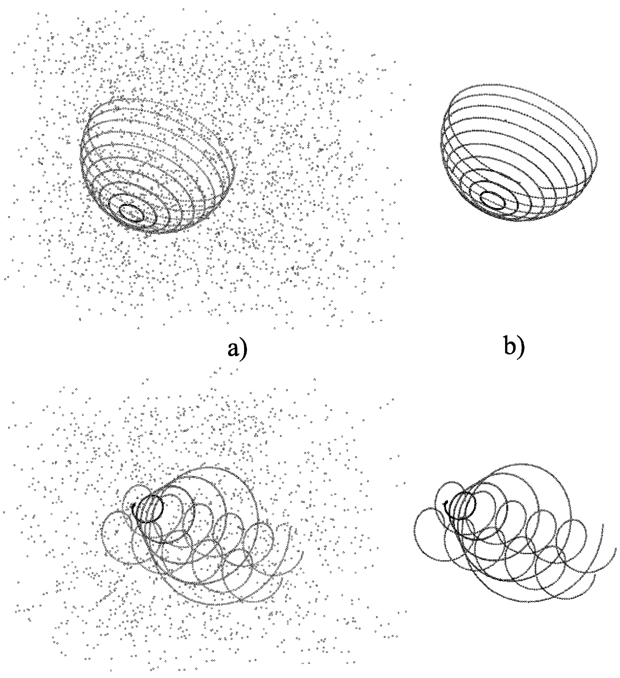

Figure 8. Reconstruction of some curves: a) Noisy sample of isoparametric curves on a hemisphere, b) Noise reduction around a hemisphere, c) Sample points of two helixes and a helispiral with noise, d) Noise reduction around the helixes and helispiral

combination of 2D and 3D B-spline curves and some geometric primitives such as a circle and an ellipse (Fig. 7). For testing reconstruction of 3D curves, we used isoparametric curves of B-spline surfaces of a degree not higher than 3. None of control points was doubled or tripled to guaranty a smooth surface. Additionally, we tested the reconstruction algorithm on some smooth frequently used solids such as cylinders, spheres, and ellipsoids (Fig. 8a). We also tried to reconstruct some implicit curves such as helixes and helispirals [15] (Fig. 8c). In all cases, we added a uniformly distributed random 3D noise in the curve neighborhood to test its effect on the reconstruction. As it can be seen, we eliminated all the noisy points in Fig. 8d, whereas we cut off the majority of them in Fig. 8c. The reconstruction result is good even if a noisy point touches a curve sample. Indeed, in this case, sample points close to the noisy point were removed as a noise, but the majority of curve sample points was correctly assigned to a curve reconstruction (Fig. 8b). Thus, we can conclude that the random noise has only a local effect on the curve reconstruction.

Additionally to artificially generated examples, we tested our algorithm on a cloud of points obtained by shoe last digitization. Similarly to the previous tests, we added a random noise (Fig. 9a). In the cloud of 3D points, we tried to remove noisy points and to identify the cross-section curves. The results were satisfiable but not as good as by other tests. Because the cloud of digitized points usually does not form an ε-sample due to the geometry of digitised objects and the used digitisation technique, the reconstruction is more sensitive to a random noise than other examples. Although again all noisy points were eliminated, some good points were lost as well (see Fig. 9b).

The algorithm for curve reconstruction was tested on the AMD Athlon 64 3000+ personal computer with 1GB of RAM. The computation time needed for the reconstruction of the examples shown in Figs. 7 and 8 can be found in Table 1.

Object	No. of sample points	CPU time [s]
Coll. of plane curves (Fig. 7)	119	≈0
Two helixes and a helispiral	3308	1.016
Hemisphere	4101	1.599
Shoe last	9151	8.120

Table 1. Computation time for the reconstruction of above presented examples

Although the presented algorithm generates larger gaps in some of the cross-section curves of the shoe last, it still can be useful to find the cross-section planes in the cloud. This information can be used to classify previously removed points of the cloud to the appropriate plane and, then, to run our algorithm for

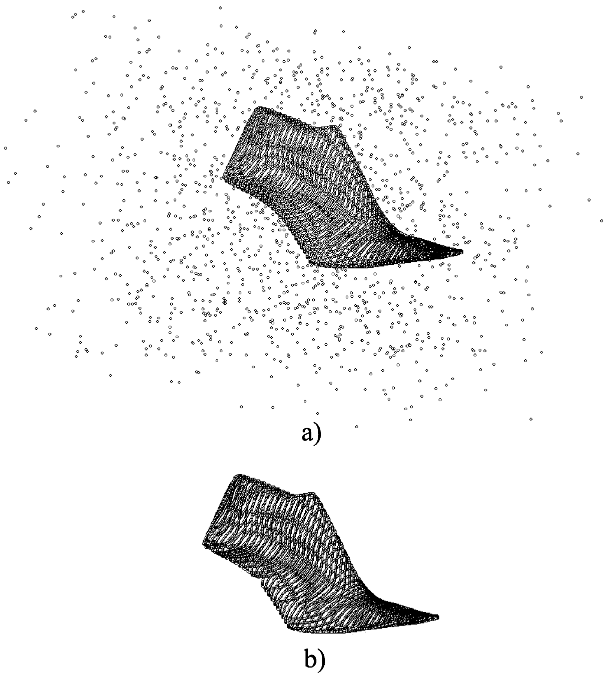

Figure 9. Reconstruction of cross-section curves: a) Cloud of 3D points obtained by shoe last digitisation, b) Cross-section curves ound in the cloud

each cross-section plane separately. In spite of the fact that the time required for such a reconstruction would be much higher than the current reconstruction time, the use of such reconstruction would still be reasonable if, in the cloud of 3D points, much noise were present and the cross-section points were non-uniformly sampled.

The time complexity of our algorithm is tightly connected to the time complexity of the algorithm for generating the minimal spanning tree. In our implementation, we used the simplest of these algorithms - Kruskal algorithm whose computational complexity is according to [7] $O(n^2 \log n)$, where n is the number of vertices. In the previous section, we already mentioned that $n(n-1)/2$ edges are generated. Since the most of the edges are eliminated due to the optimisation criterion described there, the time complexity for the computation of the minimal spanning here is the same as above but with much lower constant. As a result, we get n - 1 edges that have to be assigned to an appropriate chain of connected line segments. Thus, we have to calculate something less than n - 1 angles between each by the MST accepted line segments and the chain to which the line segment can be added. At this occasion, also the average value of angle values between two neighbour line segments in the chain and their deviation has to be calculated by fast formulas (Eq. 1 and Eq. 2). The described procedure takes $O(n)$ time. At last, the noisy chains have to be removed. For each chain of connected line segments, their average length and standard deviation have to be calculated, which again takes $O(m)$ time, where m is the number of chains, usually much smaller than the number of edges in MST. Since in the same step, we can also close the chain, this part of the algorithm does not require any additional time. Thus, the complexity of the algorithm for computing MST was not essentially changed by our supplement and the computational complexity of our algorithm remains $O(n^2 \log n)$. The complexity of the whole algorithm for curve reconstruction depends entirely on the complexity of the algorithm for computation of MST. From Table 1, it can be seen that our algorithm is already fast, but it could easily be quickened by implementing a faster algorithm for the MST calculation given in [5].

7 Conclusion

In this paper, we presented a new method for curve reconstruction based on the minimal spanning trees. The ability of minimal spanning trees to describe the shapes of smooth curves had been already observed in the early seventies. With the heuristics originated in that time, even the simple noisy points could be eliminated. From our point of view however, the ability of MST to be used in general metric spaces is their most impressive characteristic, since our goal was to find a method, which would reconstruct both 2D and 3D curves equally well. MST by themselves represents a good start for the curve reconstruction, since most of the curve shape is already captured by its edges. However, to sort those edges in a correct order, additional heuristics have to be added. Similarly, the MST cannot handle the reconstruction of the collection of nonintersecting curves or the reconstruction of closed curves.

To overcome those drawbacks, we introduced a structure called the chain of connected line segments. This is the chain of equally oriented successive connected vectors. We showed that each chain corresponds either to one of the curve in the sample or it contains the noisy data, which have to be eliminated from the reconstruction then. The noisy data are removed regarding the number of the line segments composing the chain. If a particular chain contains a too small number of line segments, it is excluded, which is an excellent tool for noise reduction. Additionally, we can easily handle the closed curves with the chain of connected line segments. If the distance between the chain endpoints is not too large, the new line segment, which connects these two points, is added and the curve is closed. With the introduction of chains of connected line segments into the reconstruction process, we did not significantly increase the computational complexity of the algorithm for calculation of MST. The computational complexity of the entire reconstruction therefore still depends on the computational complexity of the algorithm for calculation of MST. Besides, the main advantage of MST, i.e. their independence of the space dimension, remains preserved. Currently, there are still some minor problems with the reconstruction of the curves with the long straight and curved segments that can be broken in separate chains, but we think that this can be prevented by improving the heuristics for acceptance of a new line segment into the existing chain. This remains our work for the future.

8 References

[1] E. Althaus, K. Mehlhorn, "TSP-based curve reconstruction in polynomial time", *Proc. of 11th ACM-SIAM Symposium on Discrete Algorithms*, pp. 686-695, 2000.

[2] E. Althaus, K. Mehlhorn, S. Nher, S. Schirra, "Experiments on curve reconstruction", *Proc. of 2nd Workshop on Algorithm Engineering and Experiments*, pp. 103-114, 2000.

[3] N. Amenta, M. Bern, D. Eppstein, "The crust and the β-skeleton: Combitonatorial curve reconstruction", *Graph Models Image Proc*, Vol. 60, pp. 125-135, 1998.

[4] D. Attali, "r-regular shape reconstruction from unrecognized points", *Proc. of 13nd Annual ACM Symposium on Computational Geometry*, pp. 248-253, 1997.

[5] J. L. Bentley, J. H. Friedman, "Fast Algorithms for Constructing Minimal Spanning Trees in Coordinate Spaces", *IEEE Trans Comput*, Vol. 27, No. 2, pp. 97-105, 1978.

[6] J. L. Bentley, "Fast Algorithms for Geometric Travelling Salesman Problem", *ORSA J Comput*, Vol. 4, No. 4, pp. 387-411, 1992.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, , *"Introduction to Algorithms"*, The MIT Press, Cambridge MA, 1990.

[8] T. K. Dey, P. Kumar, "A simple provable algorithm for curve reconstruction", *Proc. of 10th ACM-SIAM Symposium on Discrete Algorithms*, pp. 893-894, 1999.

[9] T. K. Dey, K. Mehlhorn, E. A. Ramos, "Curve reconstruction: Connecting dots with good reason", *Proc. of Symposium on Computational Geometry*, p. 197-206, 1999.

[10] T. K. Dey , R. Wenger, "A fast reconstruction of curves with sharp corners", *Int J Comput Geom Appl*, Vol. 12, pp. 353-400, 2002.

[11] R. O. Duda, P. E. Hart, *"Pattern classification and scene analysis"*, John Wiley & Sons, New York, 1973.

[12] H. Edelsbrunner, D. G. Kirkpatrick, R. Seidel, "On the shape of a set of points in the plane", *IEEE Trans Inform Theory*,Vol. 29, pp. 551-559, 1983.

[13] L. H. de Figueiredo, J. de M. Gomes, "Computational morphology of curves", *Visual Comput*, Vol. 11, No. 2, pp. 105-112, 1995.

[14] S. Funke, E. A. Ramos, "Reconstructing a collection of curves with corners and endpoints", *Proc. of 12th ACM-SIAM Symposium on Discrete Algorithms*, pp. 344-353, 2001.

[15] G. Glaeser, H. Stachel, *"Open geometry Open GL + Advanced geometry"*, Springer-Verlag, New York, 1998.

[16] E. Horovowitz, S. Sahni, S. Rajasekaran, *"Computer algorithms"*, Computer Science Press, New York, 1998.

[17] J. W. Jaromczyk , G. T. Toussaint, "Relative Neighborhood graphs and their relatives", *Proc IEEE*, Vol. 80, No. 9, pp. 1502-1517, 1992.

[18] D. G. Kirkpatrick, J. D. Radke, "A framework for computational morphology", *In Toussaint GT, editor, Computational Geometry* , North Holland, Amsterdam: Elsevier, pp. 217-248, 1985.

[19] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters", *IEEE Trans on Comp*, Vol. 20, No. 1, pp. 68-86, 1971.

Simon Kolmanič received his B.Sc. and M.Sc. degrees in computer science from the Faculty of Electrical Engineering and Computer Sciences at the University of Maribor in 1996 and 1999, respectively. He received his Ph.D. degree in 2005 at the same faculty where he is also a teaching assistant. His current research interests include computer graphics, computer-aided geometric design and computer animation.

Nikola Guid is a Full Professor and a Head of the Laboratory of Computer Graphics and Artificial Intelligence at the Faculty of Electrical Engineering and Computer Sciences at the University of Maribor, Slovenia. His current research interests are in computer graphics, computer-aided geometric design, and modelling of natural phenomena. He received his B.Sc. degree in electronics and M.Sc. degree in computer science from the University of Ljubljana and his Ph.D. degree in technical sciences from the University of Maribor. In 1991 and 1993, each time for the period of six months, he was engaged in a research work at the Faculty of Engineering of the University of Udine, Italy. He is a member of IEEE, ACM and Eurographics.