ZÁPADOČESKÁ UNIVERZITA
University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitni 8
30614 Pilsen
Czech Republic

Methods of surface reconstruction from scattered data

(Appendix)

Jan Hrádek

September 2003

This work was supported by Ministry of Education of the Czech Republic – project MSM 235200005.

A. Review of the surface reconstruction methods

The properties of each class of the methods are summarized in the following text[+]. Note that only methods that don't require additional information aside from vertex coordinates are observed, ie. methods that process range data are not considered.

A.1. Sculpturing methods

All the sculpturing methods are based on the Delaunay triangulation or its dual Voronoi diagram. Some methods require MST or other structures to be created [7]. The complexity of the Delaunay diagram construction is $\boldsymbol{O}(N^2)$ [*]. Some methods require construction of Delaunay diagram few times (eg. two pass Voronoi filtering [2][3], Power Crust [4]) or to be computed from larger data set (eg. two pass Voronoi filtering needs the Delaunay diagram of *3N* points). However there are implementations where data are partitioned using octree, the algorithm is applied to each part and the refined surface parts are stitched together [17].

These methods require only the sampling stated as Local Feature Size (for sampling theorems see chapter 1.2), with some exceptions (α-shapes [20], α-solids [8]).

Because these techniques are interpolation methods, they don't handle noise or detect outliers. Therefore the data should be pre-processed to remove noise and outliers before using these methods.

There are heuristic approaches (eg. Biossonat sculpturing [14]) and approaches with strong theoretical background (eg. Two-pass Voronoi filtering [2][3], One-pass Voronoi filtering [4], Power Crust [5], Normalized mesh [6]). Some algorithms need post-processing phase where holes are filled (Two-pass and One-pass Voronoi Filtering, Normalizes meshes). Some methods need a parameter (α in α-shapes) that must be determined by user.

The main shortcomings of these methods are their sensitivity to noise and outliers (these algorithms interpolate the data points, so outliers must be removed in pre-processing), and their computational complexity (caused by the use of Delaunay triangulation).

A.2. Volumetric approaches

The volumetric methods are based on determination of distance function that is further processed. Such distance function could be seen as implicit function, where the surface has iso-value 0. Some methods use additional structures, eg. the Voronoi diagram is used in Natural neighbors approach [15].

The volumetric approaches usually need uniform sampling (Sampling path theorem) [23] with some exceptions (“medial axis with distance function“ method [13] needs just to fulfill LFS theorem).

These are approximation techniques, therefore the noise is smoothed (and the surface too) and also some methods are capable to detects outliers [30].

The volume is usually determined in regular grid, which yields a problem with the selection of the size of the grid. Large amount of triangles is produced in the case of too small grid step, the loss of detail in the other case. Also some triangles may have bad shape caused by the iso-surface extraction methods (Marching cubes algorithm). The size of the grid also influences the complexity of the algorithm.

The main shortcoming of these methods is the use of the regular grid. Using adaptive grid (such like octree) may yield better results and should be a subject of further improvement.

+ All the references in the text refers to the literature used in the paper.

* The complexity is expected to be $\boldsymbol{O}(N \log N)$ eventually $\boldsymbol{O}(N)$.

A.3. Incremental approaches

These methods grow actual mesh along its border. The local properties of the surface are used to produce the mesh and therefore some decomposition of the space must be done. Usually only the local neighbor information is necessary [11][24], but other relationship between the points (eg. MST) may be used for the reconstruction.

Because these methods work with local part of the mesh they require uniform sampling (Sampling Path theorem) or allow only small changes in sampling density. The sampling should be known a priory because these methods usually require parameter related to sampling (eg. ρ in BPA [11]).

Because these techniques are interpolation techniques they don't detect outliers and don't handle noise. Therefore the input data set has to be pre-processed to remove noise and outliers.

The main advantage of these techniques is low complexity and memory requirements and therefore the incremental approaches can easily process large data sets. However robustness issues arise when the noise makes it difficult to locally detect the correct topology of the surface.

A.4. Warping techniques

These techniques deform an initial surface along the input points. The initial triangulation should already capture the topology, because changes in topology are difficult. The warping techniques are based on different ideas (eg. spring model[1], particles [32]).

The sampling required by these methods is related to Local Feature Size theorem. These approaches are approximation techniques and therefore the noise is smoothed out, but it depends on the resolution of the initial mesh. The outliers are not detected by these methods and should be removed in a pre-processing phase.

The complexity of these methods depends on two factors: the size of the input data set and the size of the initial mesh that is deformed. Also some methods needs to be performed in a few successive steps to successfully reconstruct the surface. These methods don't need user defined parameters, rather they rely on suitable initial mesh.

The warping approaches are suitable if a rough approximation is already known. The user can control the complexity of the output mesh through the resolution of the initial mesh, which is also an advantage. But on the other hand, the initial triangular mesh is also their disadvantage, because it needs to be provided by the user or extracted from the input data set before reconstruction somehow.

A.5. Overall comparison

The overall comparison of the method classes is done in Table A.1. It can be seen from the table, that each of the classes of methods has its advantages and that each of them has some disadvantages where the classes could be improved.

The disadvantages are discussed in the following:

- *high complexity of sculpturing methods* – The complexity of the sculpturing methods depends on the complexity of the Delaunay triangulation (DT). The complexity of the DT is $\boldsymbol{O}(N^2)$, however the expected complexity is $\boldsymbol{O}(N \log N)$ eventually $\boldsymbol{O}(N)$. It was stated above that the data may be partitioned into disjunct sets, each of these sets could be processed separately and the reconstructed parts are then stitched together. The sculpturing part of the reconstruction then processes the DT in $\boldsymbol{O}(N)$.
- *noise sensitivity* – The sensitivity to noise is higher in interpolation techniques than in approximation techniques, where the noise is smoothed out in the approximation process. When there is noise present in the input data set, holes may appear as well as the reconstructed surface may be wrong. When the amount of noise is much smaller than sampling, it is possible to get some results with the incremental or sculpturing approaches. When the noise is so large that the method can not work properly, the noise must be removed. The noise can be removed before or after the reconstruction process, where both

approaches have its advantages or disadvantages. Removing the noise before the reconstruction process helps the reconstruction capture the shape of the object more easily, on the other hand, removing noise from already reconstructed mesh is easier than removing noise from the input point set, because the structure of the surface is provided. When the reconstruction algorithm is sensitive to noise, the noise must be removed before the reconstruction process.

- *uniform sampling* – The uniform sampling requires larger data set to be sampled from the object surface. Therefore more points need to be processed by the reconstruction algorithm.
- *initial mesh* – The issue of initial mesh for warping approaches was already discussed in the chapter A.4.

	Sculpturing	***Volumetric***	***Incremental***	***Warping***
Type	Interpolation	Approximation	Interpolation	Approximation
Sampling	LFS (S. Path)	S. Path (LFS)	S. Path	LFS (S. Path)
Noise	not handled	handled - smoothed	not handled	handled - smoothed
Outliers	not detected	not detected (detected)	not detected	not detected
Complexity	high [$\boldsymbol{O}(N^2)$]	variable – depends on the grid size	low [$\boldsymbol{O}(N \,.\, \log N)$]	variable – depends on the initial mesh size
Advantages	LFS sampling	Handle noise	Low complexity	LFS sampling Handle noise
Disadvantages	High complexity Sensitive to noise	Usually uniform sampling	Uniform sampling Sensitive to noise	Need initial mesh

Table A.1. Comparison of the classes of surface reconstruction methods. The text in round brackets means that such property of the method is unusual or rare in the certain class of methods.

B. The implementation

B.1. Selection of the method for further improvement

The main decision criterion of the method selection was the complexity of the algorithm. The robustness issues are also important and they are discussed below.

The incremental approach of Huang and Menq [24] was selected for further improvement. The advantage of this approach is the low computational complexity in comparison to the other classes of methods. The other incremental approaches require some expensive structure to be created or a metric parameter related to sampling density (ρ in BPA). The selected approach has only parameter k related to uniformity of the sampling and it needs just the extracted k-neighborhood for each point in data set.

The disadvantages of this approach are the sensitivity to noise and the required uniform sampling. Less uniform sampling can be compensated with higher values of K (see bellow) and the sensitivity of the noise can be passed by pre-processing the data with some filter to remove noise. The outliers should also be an issue of pre-processing.

Knowing the advantages (complexity) and disadvantages of the algorithm (uniform sampling, sensitivity noise), some initial experiments and modifications were done.

B.2. The algorithm

The original algorithm is described in chapter 2.3. The implemented algorithm is described in chapter 3.1, where some modifications were done. The first modification was that the algorithm selected the best candidate of all the best candidates for each edge in each successive step, in contrast to the original approach where the mesh grows linearly, ie. in the FIFO manner. The other modification is that in each successive step the best candidate that will form a triangle with a normal as much like as the normal of the incident triangle is selected (see Figure B.1. for details).

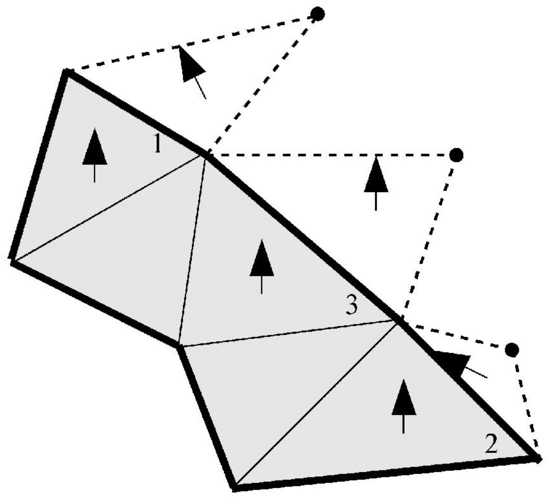


Figure B.1. The difference between the approaches. The original approach selects the candidates according to their order on the border, ie. it would select candidate belonging to the edge marked as 1 assuming the numbers represent the order on the border. The first modification approach (best candidate of all the best candidates for each edge) would select the the candidate belonging to the edge marked as 2, because the angle at the candidate is the biggest of all candidates. The second modification (the best candidate that will form a triangle with a normal as much like as the normal of the incident triangle) would select the candidate belonging to edge marked as 3, because the difference between the normals of the incident triangle is the lowest.

The algorithm that is used is in Algorithm B.1.

```
1. build_uniform_subdivision_of_the_points();
2. construct_initial_triangle();      /* see above in text */
3. while (exist_border_vertex_with_best_candidate()) {
4.    bv = find_border_vertex_with_best_candidate();
5.    cv = get_best_candidate(bv);
6.    create_new_triangle(bv,cv,bv->next);
7.    switch(which_operation_to_perform(bv,cv)) {
8.    case leaf :
9.       bcv = create_new_border_vertex(cv);
10.      nbhd = get_neighborhood(bcv,K);/* use uniform subdivision */
11.      estimate_normal(bcv, nbhd);
12.      find_best_candidate_for(bcv, nbhd);
13.      bcv->next = bv->next; bv->next = bcv;
14.      break;
15.   case fill :
16.      bcv = get_border_vertex(cv);
17.      bvn = bv->next;
18.      bv->next = cv;
19.      if(is_removable(bvn))
20.         remove(bvn);
21.      break;
22.   case bridge :
23.      bcv = get_border_vertex(cv);
24.      bcv2 = make_duplicate_border_vertex(bcv);
25.      bcv2->next = bcv->next; bcv->next = bv->next;
26.      bv->next = bcv2;
27.      break;
28.   }  /* switch */
29.   recalculate_candidates(bv,bv->prev,bv->next ... );
30.}  /* while */
```

Algorithm B.1: The recent reconstruction algorithm.

According to the Algorithm B.1. the modifications made so far are on the line 4. where different border vertex is found in each modification. The results of the modifications is discussed below.

B.3. The data structures

The following data structures were used in the implementation (the sizes include also all the needed pointers):

- Points – an array of all points. A uniform subdivision of the point space is used to speedup the neighborhood lookup problem.
  *size of the whole structure: 92 + N*24 [B], including the uniform subdivision (N – number of vertices)*
- Triangle mesh – an array of point indexes, see chapter 3.1 for details.
  *size of the whole structure: 44 + M*36 + N*13 [B] (N – number of vertices, M – number of triangles)*
- Border – the border is implemented like a linked list, where two additional structures are used to speedup the processing of border and therefore the reconstruction process. First a b-tree structure to have fast insertion/removal/search for border edge according to a decision value (the angle in the candidate in the first modification and the angle between the normals in the second modification). Second the hash table to have fast search for border edge according to the vertex (used in bridge operation).

size of the structure of border edge: B(68 + K*12) [B] (B – number of actual border edges, K – method parameter)*
*size of the b-tree node : 28+ R*32 [B] (R – rank of the b-tree)*
hash table has fixed size provided by the user.

The structures that must be stored in the memory are described in Table B.1.

Structure	***Size***
Array of points	Number of points
Uniform subdivision of point space	Related to the number of points, provided by the user
Array of triangles	Number of triangles
Triangles neighbors	Related to number of triangles
Triangle normals	Number of triangles
Border vertex (edge)	Number of edges (linked list + b-tree + hash table + array of neighbor points)

Table B.1. The data structures

Here is a little example about memory usage: Assume that the input data set has 5.10^5 points (half million) and the reconstructed mesh contains approximately 10^6 triangles (one million) and *K* was chosen as *K*=20. Assuming that the border is at most $3{,}5.10^4$ edges long, 100MB of the memory is consumed by the reconstruction process.

B.4. Parameter K

The parameter *K* is the number of neighbors (*k*-neighborhood) that are used in a few parts of the algorithm. First, the neighborhood is used for the normal estimation. Second, the selected candidate point (see the description of the algorithm in chapter 3.1) must lie in the *k*-neighborhood of both endpoints of the edge. If no point fulfills this condition, then the edge is marked as final. Notice that *K* is not metric value, but an integer number.

The size of the parameter *K* affects the normal estimation in such a way that bigger *k*-neighborhood approximates a larger area of the surface which leads to smoothing of the estimated normal. The normals define the tangent planes, where the second part of the criterion is evaluated (the selected candidate point must lie within the angle formed by neighboring edges in local tangent plane) and therefore they should capture this surface property right, see Figure B.2. Small values of *K* may lead to inaccurate estimation of the normal, mainly in the vicinity of sharp features (edges, corners etc.). Mostly, this is the case of badly sampled surfaces. These problems don't occur, when the surface is sampled so that the sampling density is changing slowly and every detail is properly captured by the samples.

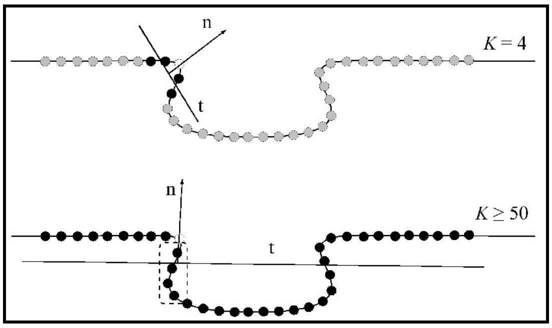


Figure B.2. The loss of detail in the case of large values of K (2D case). On the picture are the sampled points depicted as circles. The white circle represents the point where the normal has to be estimated. The black circles represents the k-neighborhood of the selected point. The estimated normal is depicted as the tangent plane (t) and normal itself (n). When the K is K = 4 the normal is estimated so that no loss of detail occur. But when the K is K ≥ 50 the points marked in the figure by dashed box are hidden.

The main purpose of the *k*-neighborhood is the evaluation of the last part of the criterion (selecting the common neighbor of both endpoints of edge and detecting boundary edges). If no suitable candidate point is found for the edge, that edge is marked as final and this edge is not processed anymore. This may lead to “undesirable holes”†, therefore the parameter *K* must be so large that such holes don't appear. If the holes are small enough (3-20 point approximately) the holes may be filled by some post-processing.

The parameter *K* should be large enough so the undesirable holes will not appear, still bear in mind that large values of *K* may lead to loss of detail (when estimating normals).

The selection of the proper *K* depends on the sampling used to capture the surface and on the interpretation of the input data set. Theoretically the minimal value of *K* is *K*=6 (mesh of equal-sided triangles). In the work of Huang-Menq the *K* is selected from the interval 10-20, where *K*=20 was used for data with changing sampling density, however noisy data were not considered. For a data with heavily changing sampling density the parameter *K* must be set even bigger (the data set `woman`, see Figure 3.3c and Table 1., was reconstructed without holes when *K* was set to *K*=40). The parameter *K* doesn't depend on the sampling density itself, but on the uniformity of the sampling density and on the interpretation of the input data set by the user (only the user is capable to distinguish between a hole in the surface and lower sampling).

The parameter *K* should be therefore selected visually by the user: inspecting the data set (the point cloud) the user chooses the size of the *k*-neighborhood needed for the proper reconstruction of the surface. After reconstruction the user considers the result (the presence of undesirable holes) and raises *K* eventually. It was experimentally proved that suitable starting values are from interval $K \in [10,20]$, with respect to sampling uniformity. Selecting *K* automatically is a problem, because it is impossible to automatically distinguish between hole in the surface and lower sampling density.

The influence of noise on the reconstruction process depends on its size. Small noise, ie. much smaller than the sampling density, affects the normal estimation and different *k*-neighborhood may be extracted for the points. The effect of such small noise presence in the data set may be compensated with higher values of *K*. Large noise, ie. comparable with the sampling density, makes it impossible to extract proper structure of the surface and therefore such data must be pre-processed to remove noise.

The parameter *K* has direct impact on the complexity of the algorithm, because larger value of *K* causes processing the criteria on larger *k*-neighborhood.

B.5. The results

The results of the reconstruction process are also discussed in chapter 3.2.

The first proposed modification of the original method of Huang-Menq tries to construct the best triangle of all possible triangles (in the sense of the criteria) in each successive step, while the triangles are constructed by the order of the edges on the border in the original work. The second modification prefer flat development of the surface and thus better handle sharp features. The comparison of all the methods is presented on Figure B.3.

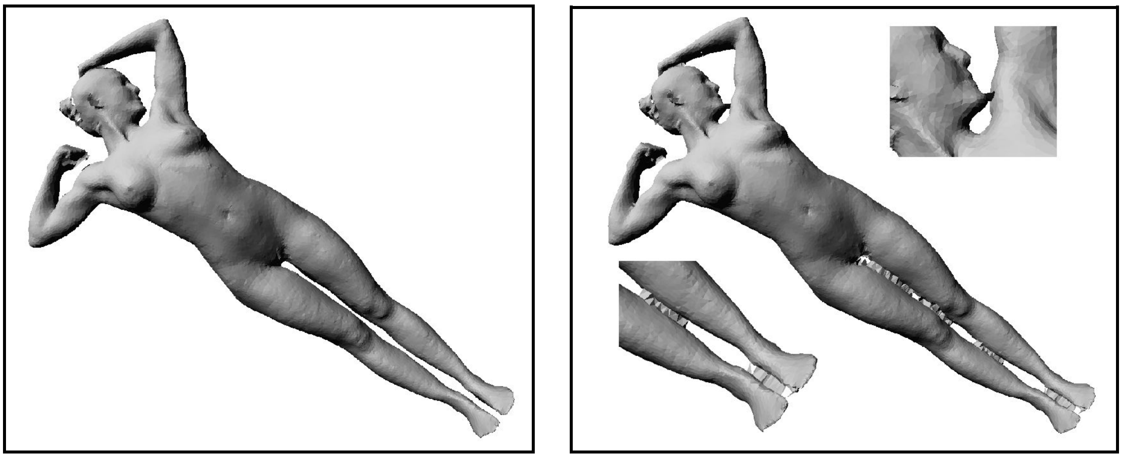
The original approach doesn't use any strategy to select the triangle to be reconstructed, it follows the order of the edges on the actual border. Such dumb selection may construct wrong triangle, that could be omitted in the following reconstruction process (see the connection of the chin and the shoulder on Figure B.3b.).

The first modification uses a strategy that select the best candidate of all the best candidates. This strategy removes the problems of the dumb selection, ie. the selection of completely wrong triangle, merely because it is in a row. However this approach doesn't solve other problems, for example under-sampling (see Figure B.3c, where the connection between the chin and the shoulder disappeared, however the wrong triangulation in the under-sampled regions between legs stayed).

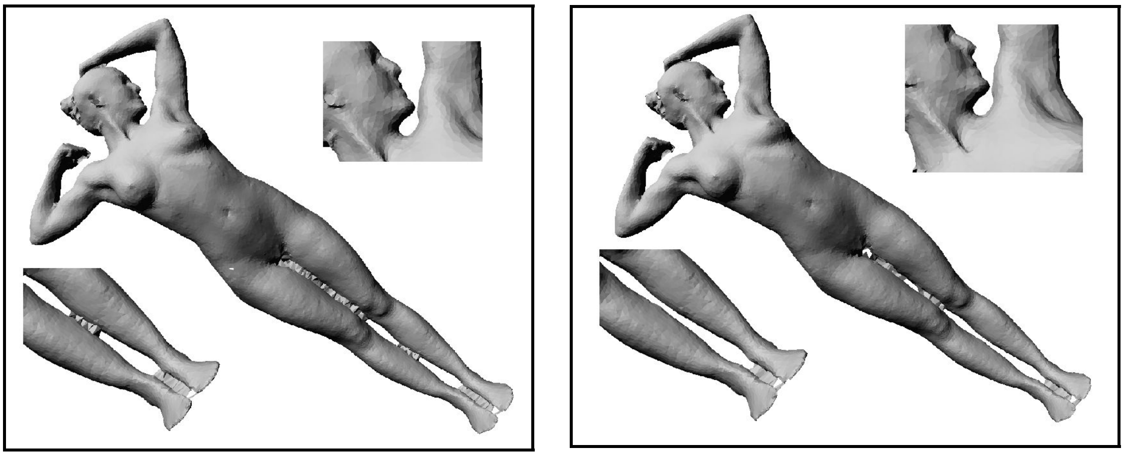
The second modification is inspired by the idea, that it is safer to make triangles in a flat area, rather than in areas of unknown surface properties. This simple strategy select the flattest triangle of all the triangles formed from the best candidate and the corresponding edge, ie. the triangle with the lowest difference

† The term “not convenient holes” depends on the interpretation of the input data set by the user.

between normals of the new and adjacent triangle is constructed. This strategy may help in under-sampled regions, where it is problematic to distinguish where the surface is actually present (see Figure B.3d, where the chin and the shoulder are still well triangulated, and some of the problems with the triangulation between legs were solved).



a) `woman2` *(22 625 points, 44185 triangles)*　　*b) original method of the reconstruction*

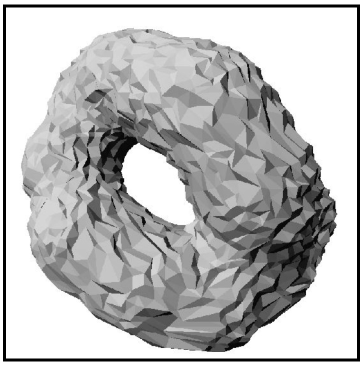


c) "best of the best" strategy　　*d) "flatest of the best" strategy*

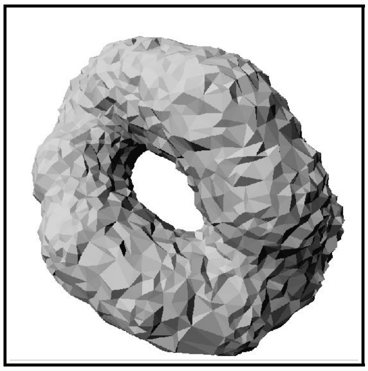
Figure B.3. The comparison of the reconstruction methods.

The choice of the parameter *K* was discussed in the previous chapter: the parameter *K* should be selected somewhere between 10 and 20 by visual observation of the data set, and when undesirable holes appear the parameter *K* should be increased, ie. the under-sampling may be compensated with higher values of K. The effect of heavily changing sampling density can be seen in Figures 3.4, 3.5 and 3.6 where *K* has been increased to successfully reconstruct the data set.

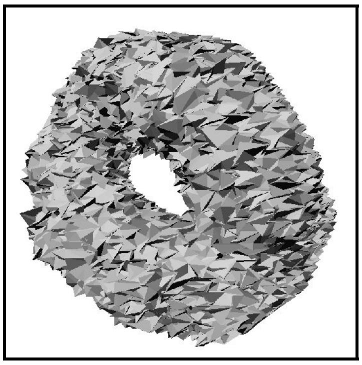
It was said that the incremental reconstruction algorithms are sensitive to noise. The algorithm was tested on artificially noised data sets. Noise of small size as well as noise of large size were considered. On Figure B.4. are the results of the tests. All the results prove the statements from the previous chapters, ie. when the noise is very small (according to the sampling density), there should not be a problem for the method to reconstruct the surface without holes, however remember that the method is interpolation technique and therefore the noise is still present in the mesh. In the case of large noise (comparable to the sampling density or even bigger) the algorithm has problem to capture the local topology and the result is wrong.



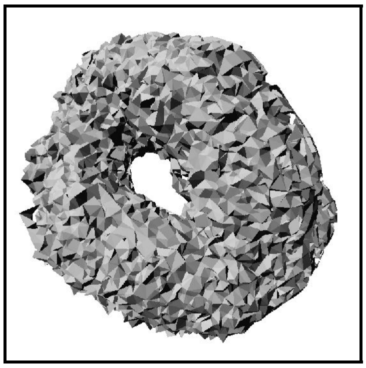
a) small noise – input



b) small noise – result



c) large noise – input



d) large noise – result

Figure B.4. The influence of the noise on the reconstruction. The noise size is 1/10 of sampling density in the test "small noise". In the test "large noise" the noise size is equal to the sampling density.

As was stated in the previous chapter the complexity of the used reconstruction algorithm is $\boldsymbol{O}(N \log N)$. This complexity is studied in the following graphs. The influence of the parameter K is shown on Figure B.5 and the influence of the number of points (and resulting triangles) is shown on Figure B.6.

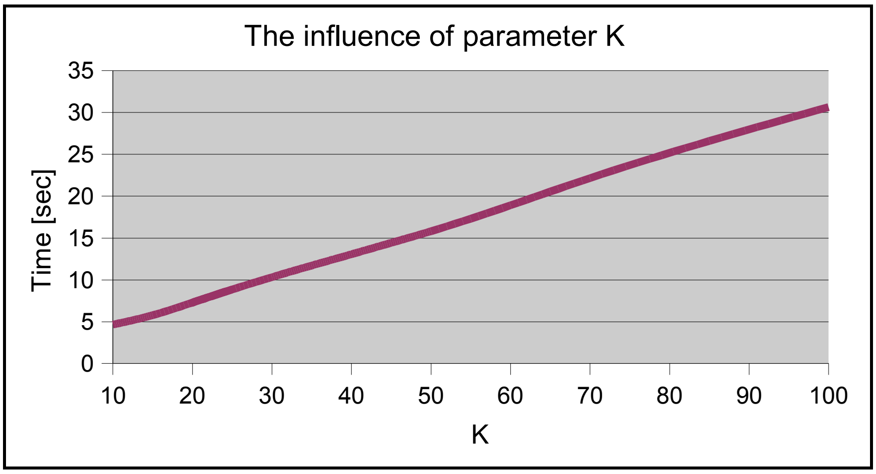
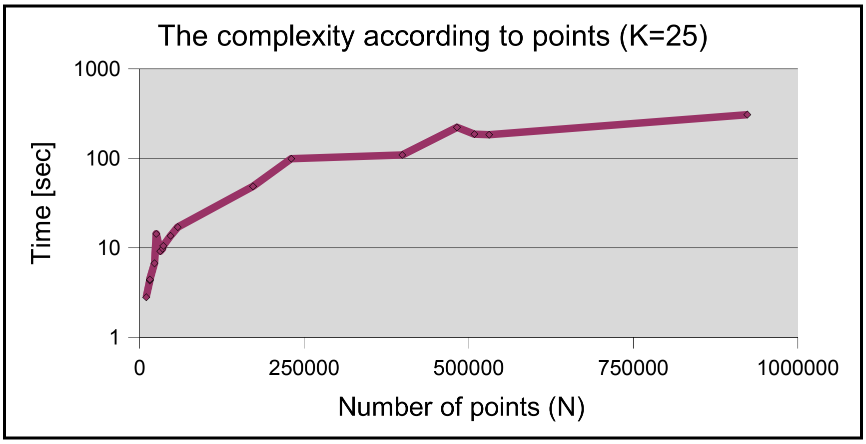
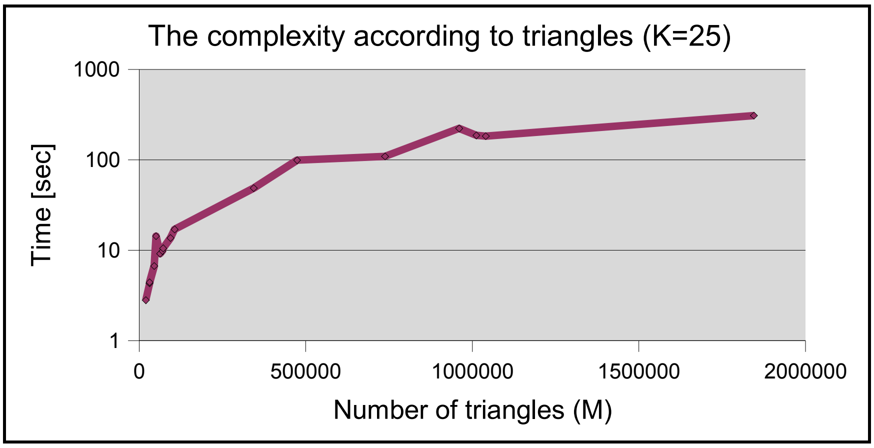


Figure B.5 The influence of parameter K. The used data set was `woman` *(22 625 points, 44185 triangles). The parameter K was taken from interval [10;100].*



a) the complexity related to the number of points



b) the complexity related to the number of triangles

Figure B.6. The complexity of the algorithm. The algorithm was tested on many real-world as well as artificial data sets. The parameter K was chosen to be 25, so the reconstruction was completed successfully on all the data sets.

C. Conclusion and future work

The presented algorithm is fast, inexpensive and has low complexity, therefore it is suitable for large data sets. The advantage of the algorithm is a limited adaptivity to non-uniform sampling caused by the use of non-metric *k*-neighborhood.

It was shown that the algorithm can handle a small scale noise, however, robustness issues arise when the noise makes it difficult to capture local topology.

The influence of the only parameter K on the reconstruction process was studied and as a result a recommended reconstruction procedure was stated.

Two modifications of the recent algorithm were proposed and their quality was visually observed. The modifications show that more properties of the local mesh should be observed during the reconstruction process, mainly in the case of sharp features and areas of under-sampling.

Therefore, in the following work we will improve the current incremental reconstruction algorithm so it will have these properties:

- more properties of the local part of the mesh will be observed to improve the quality of the mesh and to improve the robustness of the reconstruction, mainly we will focus on observing the curvature and the behavior of normals,
- we will design methods that will consider the quality of the resulting mesh according to given data set and original object,
- we would like to analyze the possibilities of adaptive parameter K, ie. try to change the values of the parameter K according to the immediate results,
- as a result we would like to get more robust algorithm for surface reconstruction yet still inexpensive and fast.