Proceedings of ALGORITMY 2005
pp. 280–289

SIMPLE HOLES TRIANGULATION IN SURFACE RECONSTRUCTION

MICHAL VARNUŠKA, *, JINDŘICH PARUS§ , AND IVANA KOLINGEROVÁ †

Abstract. Surface reconstruction is a common task in the modern computer graphics and computational geometry. Given a set of points P sampled from some unknown surface S, we have to create the triangle mesh interpolating or approximating the input points. Various algorithms were developed during the past years to satisfy this task, but no one is able to handle all kinds of data. We use the algorithm based on the CRUST algorithm and we present some modifications which improve the quality of the resulting triangle mesh.

Key words. Surface reconstruction, holes triangulation, mesh filtering.

AMS subject classifications. 65D18, 68U05

1. Introduction. This paper presents a simple approach for triangle mesh improvement. This mesh is created using the surface reconstruction algorithm, which produces holes in the sampled surface in the case of badly sampled data, so a holes filling step has to be done then.

This paper consists of five sections. Section 1 contains the introduction to the task and the state of the art. Section 2 describes briefly the CRUST algorithm and its problems. Next section aims to the description of the holes filing algorithm. The results are in Section 4 and Section 5 concludes the paper.

1.1. Input. One of many ways for the real objects models acquisition is sampling of the object by various scanning devices followed by the surface reconstruction algorithm. Given the input set P of the points p sampled from the surface S of the real object, we want to create a triangle mesh interpolating or approximating the surface. So, the input to our task is:

$$\forall p \in P, P \subset S : p = [x, y, z]; x, y, z \in R$$

We do not have any other additional information about the reconstructed surface, such as normal vectors in the sampled points. We do not know either, whether the surface was sampled uniformly or with regard to the surface curvature, whether it contains noise or not.

1.2. State of the art. The problem of surface reconstruction has been solved by many research groups in the whole world. Starting in the 80s, many algorithms have been developed which interpolate or approximate the input sampled points. The methods can be divided into four groups [23] (division is not strict, some methods can belong to more groups):

*§The author was supported by Ministry of Education of The Czech Republic, project FRVŠ n. 1349/2004/G1

†The author was supported by Ministry of Education of The Czech Republic, project MSM n. 235200005

Centre of Computer Graphics and Data Visualization, Department of Computer Science and Engineering, University of West Bohemia, Pilsen, Czech Republic
miva@kiv.zcu.cz, jparus@kiv.zcu.cz, kolinger@kiv.zcu.cz

- warping
- incremental surface construction
- distance function methods
- spatial subdivision

Warping works on the basic idea that we deform some starting surface to the surface that forms the object. The Müller's approach [25] uses the starting triangle mesh which deforms to the mesh which is close to the original surface. Szeliski [26] uses oriented particles, where every particle has some parameters whose values are updated during the modeling simulation.

The methods of incremental surface reconstruction start at some starting simplex (triangle, edge) and other simplices are incrementally added. Boissonat [8] presented the approach which begins on the shortest edge from all edges between points and incrementally appends the points to create triangle mesh. Mencl and Müller [22, 24] developed a similar algorithm based on creation of an extended minimum spanning tree, identification and extraction of typical features and using these properties for triangle mesh extraction.

The distance function describes the shortest distance from the point to the surface. For closed surfaces, the value of the function is negative or positive depending on whether the point is inside or outside the object. This function is computed for each point using the tangent plane. The plane can be estimated from $k-$nearest neighbours (points, the parameter k is set by the user) by the least square approximation. A typical representative of this methods class is Hoppe [19, 20]. Curless and Levoy [9, 21] gave another effective algorithm which represents the signed distance function on a voxel grid and is able to reconstruct eventual holes in a post-processing step.

The basic feature of the spatial subdivision methods is the boundary hull (convex hull, box around points, etc.) division to the independent areas forming e.g. the regular grid, octree or tetrahedra. The surface is then extracted using the relationship to the surface described by the input set (e.g. the surface triangles should be small, etc.). Delaunay triangulation [11] (tetrahedronization) and its subgraphs, such as Gabriel graph and relative neighbourhood graph, are very popular in this class of methods because the surface is contained as a subgraph in Delaunay triangulation.

The best known is the α-shape algorithm by Edelsbrunner and Mücke [16, 17]. They use the Delaunay triangulation and delete simplices whose radius is bigger than the radius α of some α-ball. Algorri and Schmitt [1] gave an effective algorithm in which the space is divided by a regular grid (voxels). In the next steps those voxels are chosen which contain points from the input set and the surface is extracted. Bernardini and Bajaj [5] developed an algorithm which gets the surface subcomplex of Delaunay tetrahedronization. This algorithm extends the idea of α-shapes and it uses the binary search for the parameter alpha to find this subcomplex. A paper by Bernardini [6] describes an algorithm to interpolate a set of points. It is not based on Delaunay sculpturing, but it extends the surface (Delaunay triangles initially) like in surface growing methods.

Amenta introduced a concept of CRUST, e.g. [3, 2]. It selects the surface triangles from triangles in Delaunay triangulation using the information from dual Voronoi diagram. Dey extended the ideas of Amenta and gave an effective COCONE algorithm. The basic idea is presented in [2]. Other papers presented by Dey introduced the way how to handle large data [13], which is the common problem of Delaunay based algorithms, and what to do with boundaries [14], undersampling and oversampling [12]. These ideas are based on the observation that the places with point density changes can be detected using shape of the Voronoi cells in these places. Both authors gave an algorithm for a watertight surface reconstruction, Nina Amenta her PowerCRUST based on medial axis transformation [4] and Tamal Dey his TightCOCONE based on tetrahedra removal [15].

The above presented methods are just the best known methods, there exist many other algorithms and it is not the goal of this paper to collect all of them. For the readers looking for other, we recommend e.g. the papers by Müller [23] or Bernardini [7].

2. CRUST algorithm. In our reconstruction program we use the CRUST algorithm developed by Nina Amenta [3, 2]. It belongs to the group of spatial division algorithms, it uses the Delaunay tetrahedronization. There are two versions of the algorithm (onepass and twopass) depending on the number of Delaunay tetrahedronizations (DT) used in the method. We use for our purpose the algorithm which uses only one DT. The reason is simple, the DT is memory-consuming and the second DT in the twopass algorithm needs three times more points than the onepass version.

2.1. Principle of the algorithm. We describe the basic principle of the algorithm very briefly, the details with strong theoretical background can be found in the related papers. The DT is the first step of the algorithm, the second step is the Voronoi diagram creation by the dualization process. The surface triangles are selected from the DT triangles using the information from the Voronoi diagram. The fundamental term used in this method is the *pole*. The positive pole p^+ for each point (see Fig. 2.1) is the farthest Voronoi vertex of the Voronoi cell related to the point while the negative pole p^- is the farthest Voronoi vertex on the "opposite side" (the dot product of the vector $\overrightarrow{p^+ - p}$ and $\overrightarrow{p^- - p}$ is approximately -1).

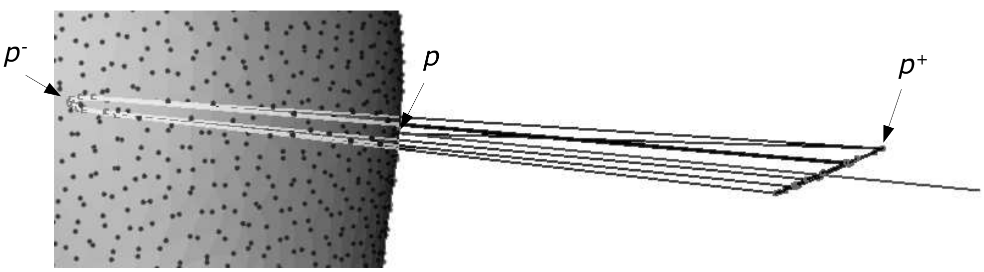

FIG. 2.1. *The part of the surface and the Voronoi cell of the point p and its poles, the positive pole p^+ and the negative pole p^-.*

As visible in the figure above, the Voronoi cell is very thin and long, so the vector from the point p to the positive pole p^+ is roughly perpendicular to the surface and can be taken as the normal vector. This observation is true for sufficiently sampled surface, the correctness of the sampling can be determined using the *LFS* criterion (local feature size, described also in [3]).

2.2. Problems in the algorithm. A big advantage of the method is its insensitivity to the changes in the sampling density, so the algorithm does not need uniformly sampled data. The problems arise when the input dataset contains big undersampled or oversampled places, boundaries, outliers, sharp edges or noise. All these properties of the algorithm come from the fact that the shapes of the Voronoi cells do not satisfy the conditions (they have to be thin and long) and the computed surface is incorrect, see Fig. 2.2. For more details please refer the Amenta's papers.

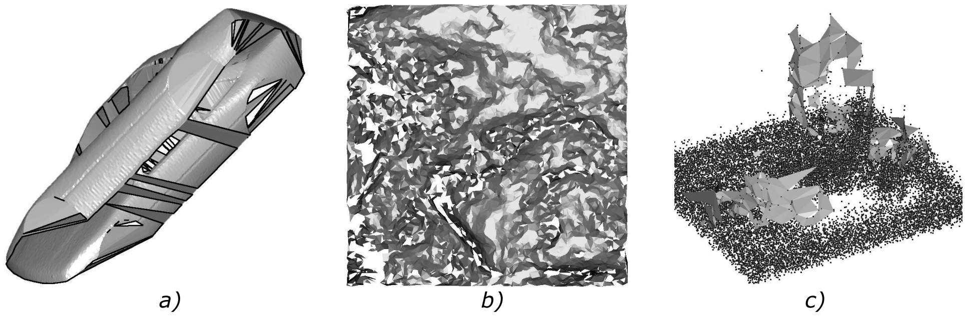

FIG. 2.2. *The reconstruction of some of problematic datasets, a) the dataset with the boundaries and highlighted incorrect parts, b) a badly sampled terrain, c) a dataset with a lot of noise.*

3. Holes filling. The presented problems cause the failure of the surface reconstruction algorithm and the triangle mesh is not correct, big unwanted triangles, a lot of overlapping triangles and holes appear there. There exist some robust algorithms able to repair the incorrectly reconstructed surfaces containing complicated holes and unconnected parts, such as David's volumetric diffusion approach [10] or Emelyanov's bridge approach [18]. The problem is paradoxically with their robustness, they are too complicated for simple holes filling. However, reconstructed surfaces in case of well sampled data usually do not contain so complicated holes, most of the holes is quite simple.

In this section we introduce a simple approach able to fill the holes in the reconstructed surface. Because the filling algorithm is not connected to the surface reconstruction algorithm, it can be used for the mesh improvement generally. Thus we assume (and our surface reconstruction algorithm produces such surfaces) that the reconstructed surface is well reconstructed with no overlapping triangles, correctly detected and reconstructed boundaries and with simple holes (two approaches how to improve the mesh containing incorrect boundary triangles and overlapping triangles can be found in [27, 28].

3.1. Holes tracing. Firstly, the holes have to be located in the triangle mesh, we can use the tracing approach (see Fig. 3.1a). The whole triangle mesh is processed and we look for the triangles without neighbours on the edges. When such a triangle is found (e.g. the triangle with the edge v_1v_2 in the figure) one vertex (v_1) is marked as the starting vertex. Then, we look for the next empty edge (empty edge is the edge associated with one triangle only) around the second vertex v_2 of the starting edge and using this approach the whole hole is found. The problem occurs in the case of point v_4 where more than two empty edges coincide.

As the traced hole should be as small as possible, we want to select v_4v_5 (not e.g. the edge v_4v_{bad}, see (Fig. 3.1b)) as the next edge. We create the plane which separates the space into two halfspaces given by the edge v_3v_4 and the normal vector n_t of the triangle t. When there are other edges lying in the same halfspace (given by this plane) as the rest of the traced hole, we take the edge with the smallest angle to the edge v_3v_4. In the other case, when no other edges are in the same halfspace, we select the edge with the biggest angle to the edge v_3v_4.

This approach of choosing next hole edge works amazingly well and we have found only few cases when it did not work correctly, especially in the noisy datasets, where the configurations of holes were awful and it was difficult to decide where to continue.

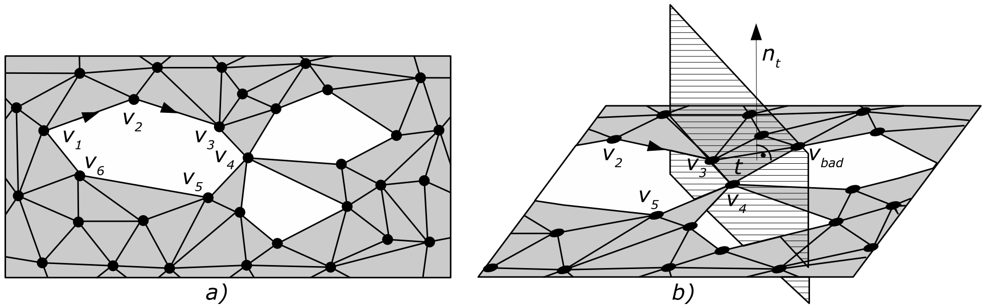

FIG. 3.1. *Holes tracing, a) the vertices* $v_1v_2v_3v_4v_5v_6$ *represent the hole, the vertex* v_4 *is problematic, b) the plane created using the edge* v_3v_4 *and a triangle* t *(with normal* n_t*) coincident to the edge, the vertex* v_{bad} *is the vertex with the smallest angle to the edge* v_3v_4*.*

According to our experiments, such problems are common in other reconstruction programs, too.

3.2. Holes filling. When the holes tracing is finished, a set of traced holes is created and for each member of this set the holes filling is done. Because we want to fill only small holes and leave the big holes, which represent the boundary, unaffected, there has to be some limit on the hole size. Unfortunately, there is not an exact way how to determine whether the hole is small or not, the user has to have the last word, but the heuristic limit of 50 edges seems to be good enough to separate small holes from big boundary holes. Thus we perform holes filling only for small holes Fig. 3.2a) and for boundary holes (Fig. 3.2b) only shape improving is performed (few triangles are added to create better boundary shape).

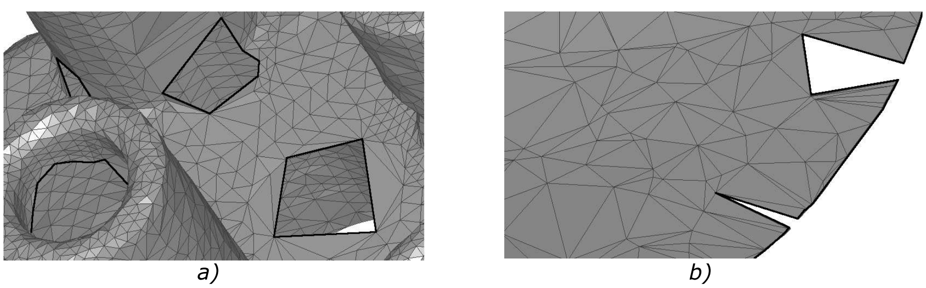

FIG. 3.2. *Typical holes, a) small holes which have to be triangulated, b) boundary holes, only a few of triangles have to be added to correct the shape.*

For the holes filling we use an approach similar to the ear cut algorithm known in polygon triangulation. The polygon, or the hole in our case, is given by the vertices $v_0v_1v_2\dots v_{n-1}$ and the ear is the triangle created by the vertices $v_{(i-1)\%n}v_iv_{(i+1)\%n}$ where "% " means modulo division. The main difference is in the fact that polygon triangulation is done in E^2 but in our case in many cases we are not able to project the holes to the plane due to complicated shapes, so we have to triangulate the hole in E^3. The algorithm is simple, see Fig. 3.2.

The first step in the holes triangulation procedure is the ear evaluation. We have tried three possible approaches how to evaluate an ear (a possible triangle $v_{i-1}v_iv_{i+1}$) based on:

```
procedure triangulate hole (hole = v_0 v_1 ... v_{n-1})
   for i = 0 to n-1
      evaluate ear v_{(i-1)%n} v_i v_{(i+1)%n}

   while the hole is not triangulated yet
   {
      v_best = -1;
      for i = 0 to n-1
         if v_i has better evaluation then v_best AND
            Δv_{(i-1)%n} v_i v_{(i+1)%n} is correct
               v_best = v_i

      if v_best=-1
         exit;

      create triangle Δv_{(best-1)%n} v_best v_{(best+1)%n}

      evaluate ear v_{(best-2)%n} v_{(best-1)%n} v_{(best+1)%n}
      evaluate ear v_{(best-1)%n} v_{(best+1)%n} v_{(best+2)%n}
      remove v_best from the hole
   }
```

FIG. 3.3. *The code for the hole triangulation (the character % means modulo division).*

- the smallest angle
- the smallest length of the edges
- the smallest neighbours angle

The smallest angle approach computes the angle between the vectors $\overrightarrow{v_{i-1} - v_i}$ and $\overrightarrow{v_{i+1} - v_i}$ using the dot product, see Fig. 3.4a. The smallest length approach computes the sum of distances between ear vertices, thus $\left|\overrightarrow{v_{i-1} - v_i}\right| + \left|\overrightarrow{v_{i+1} - v_i}\right| + \left|\overrightarrow{v_{i-1} - v_{i+1}}\right|$, see Fig. 3.4b. The last approach computes the angles α_1 and α_2 between the triangle normal vector n_1 (triangle coincident with the edge v_{i+1}, v_i), n_2 (triangle coincident with the edge $v_{i-1}v_i$) and the ear normal vector n_t, see Fig. 3.4c. Both angles are then multiplied to get the final evaluation (the second possibility is the summarization of angles, but multiplication is better because it prefers ears with both angles small).

After the evaluation procedure the ears are recursively cut depending on their evaluation in the loop. First, the ear with the best evaluation is chosen. In the case that the ear is not correct (when we insert the ear triangle to the triangle mesh, the triangles will overlap, see Fig. 3.4d) we have to choose another one. The correctness is determined using the angles between the existing triangles and the new ear triangle. When we are not able to find a correct ear, the procedure ends and the hole remains triangulated only partially. Otherwise, the ear is put to the triangle mesh, the hole is reduced by one vertex and the ears $v_{(best-2)\%n}\ v_{(best-1)\%n}\ v_{(best+1)\%n}$ and $v_{(best-1)\%n}\ v_{(best+1)\%n}\ v_{(best+2)\%n}$ are reevaluated because the hole was locally changed in this place.

The above described procedure is used for the holes triangulation of both small and boundaries holes, the difference is only in the angle limit when the correctness of the inserted ear is computed. When the boundary holes are triangulated, we use a smaller angle limit, thus the triangle normal of the newly added triangle must have a small difference from the normal vector of the coincident triangles.

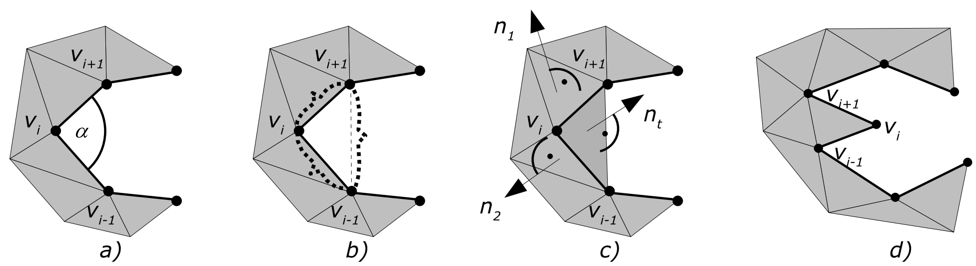

FIG. 3.4. *The ear $v_{i-1}v_iv_{i+1}$ evaluation, a) the smallest angle, α is the angle between the vector $v_{i-1} - v_i$ and the vector $v_{i+1} - v_i$, b) the smallest length of the edges, the variables d_1, d_2 and d_3 are the lengths, c) the smallest angle between neighbours, n_1 is the normal vector of the triangle with the edge v_{i+1}, v_i, n_2 is the normal vector of the triangle with the edge v_{i-1}, v_i, n_t is the normal vector of the ear, d) an invalid ear.*

4. Results. We have tried the triangle filling procedure on those datasets whose triangle meshes contain holes after the reconstruction. The holes tracing procedure worked well with the exception showed at the end of Subsection 3.1 and it traced correctly the holes.

The procedure of holes filling worked correctly. All three evaluation approaches seemed to work, but they produced different results. Fig. 4.1 and Fig. 4.2 show examples of the reconstruction followed by the holes filling. The first approach, the evaluation using the smallest angle, produces very often high number triangles coincident with one vertex (see e.g. Fig. 4.1c). The reason is following: when we cut the ear then on the place of cutting the angle becomes smaller than before and the next cutting will continue in the same place. The approach using the smallest edge length seems to be better, the triangles are not so thin as using the previous approach. But the best results, especially in the places with sharp edges, are reached using the approach with neighbours angles, the inserted triangles adapt to the local geometry, so sharp edges are preserved.

Very problematic places are the places where one part of the surface is very close to another surface and the sampling process was not completely correct. Fig. 4.3 shows an example with one of these datasets, two parts of surface are in the problematic places connected with "bridges" and the hole cannot be correctly triangulated.

Unfortunately, the approach presented in this paper was not very successful on the noisy datasets, which contain a lot of holes, because the reconstructed surface was too spare to triangulate it correctly. Although the holes were filled, the filling was not correct due to overlapped and intersected triangles. It is almost unable to reconstruct these datasets using currently existing algorithms and we suppose to concentrate on this problem more in the future.

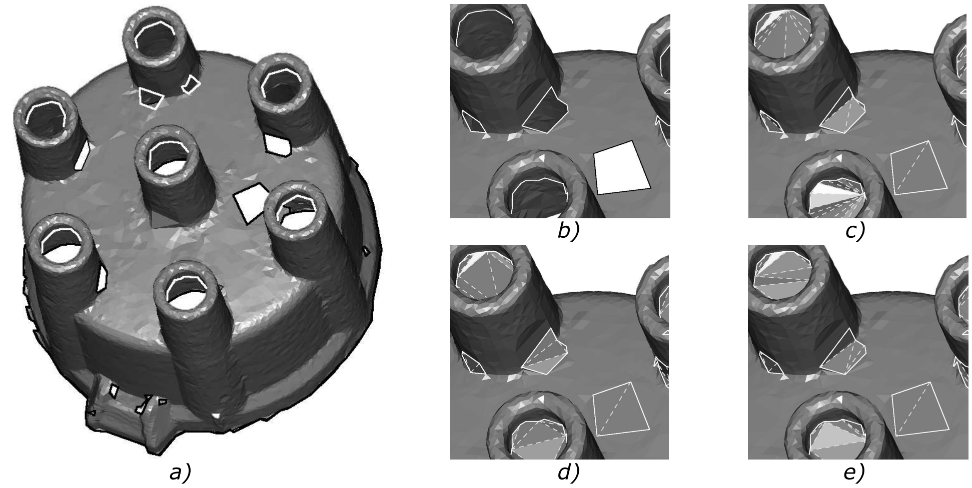

FIG. 4.1. *The "distcap" datafile, a) the whole mesh with holes highlighted, b) a zoom to one part with holes, c) the smallest angle filling, d) the smallest edge length filling, e) the neighbours dependent filling.*

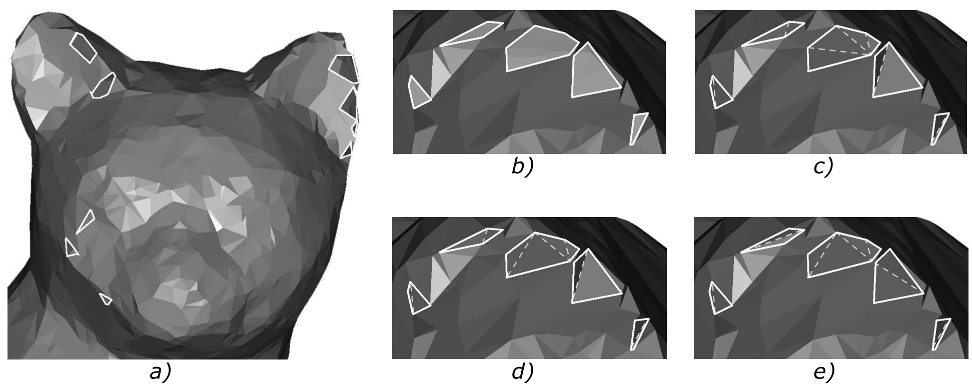

FIG. 4.2. *The "cat" datafile, a) the head of the cat with highlighted holes, b) a zoom to one ear with holes, c) the smallest angle filling, d) the smallest edge length filling, e) the neighbours dependent filling.*

5. Conclusion and acknowledgment. We have presented a simple approach to fill the holes remained after the surface reconstruction. Although it is simple to understand it and program it, the results of the algorithm are good and usable for our surface reconstruction approach. Open question and a big challenge is the problem of the noisy datasets reconstruction, our research will continue in this direction to produce higher quality results of these datasets.

6. Acknowledgement. The authors want to thank prof. Václav Skala for good work conditions and to our colleagues, especially to Petr Vaněček for technical help with the paper preparation.

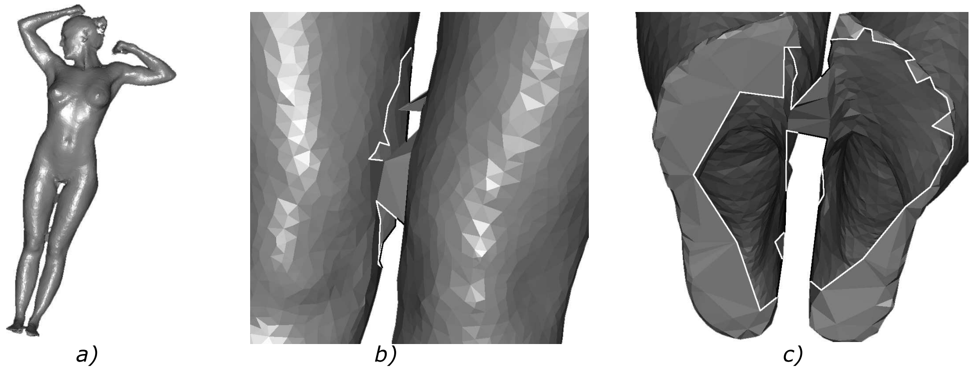

Fig. 4.3. *The "women" datafile with problematic connected places, a) the whole surface, b) the legs with big point undersampling, c) the feet with the same problem.*

REFERENCES

[1] Algorri M. E., Schmitt F. *Surface reconstruction from unstructured 3D data.* Computer Graphic Forum, 1996, pp.47-60

[2] Amenta N., Choi S., Dey T. K., Leekha N. *A simple algorithm for homeomorphic surface reconstruction.* 16th. Sympos. Comput. Geometry, 2000, pp.125-141

[3] Amenta N., Bern M., Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. SIGGRAPH, 1998, pp.415-421

[4] Amenta N., Choi S., Kolluri R. *The Power Crust.* Proc. of 6th ACM Sympos. on Solid Modeling, 2001, pp.127-153

[5] Bernardini F., Bajaj C. *A triangulation based Sampling and reconstruction manifolds using a-shapes.* 9th Canad. Conf. on Comput. Geometry, 1997, pp.193-68

[6] Bernardini F., Mittleman J., Rushmeier H., Silva S., Taubin . *The ball pivoting algorithm for surface reconstruction.* IBM Technical Report RC21463(96842).

[7] Bernardini F., Rushmeier H. *The 3d model acquisition pipeline.* State of the art report, EUROGRAPHICS 2000, pp. 41 - 62

[8] Boissonat J. D. *Geometric structures for three-dimensional shape representation.* ACM Trans.Graphics 3, 1984, pp.266-286

[9] Curless B., Levoy M. *A Volumetric Method for Building Complex Models from Range Images.* Computer Graphics 1996, pp. 303 - 312

[10] Davis J., Marschner S., Garr M., Levoy M. *Filling holes in complex surfaces using volumetric diffusion.* First Int. Symp. on 3D Data Proc., Visual. and Transmission 2002

[11] Delaunay B. *Sur la sphre vide.* Izvestia, Akademii Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk 7, 1934, pp. 793 - 800

[12] Dey T. K., Giesen J. *Detecting undersampling in surface reconstruction.* Proc. of 17th ACM Sympos. Comput. Geometry, 2001, pp.257-263

[13] Dey T. K., Giesen J., Hudson J. *Delaunay Based Shape Reconstruction from Large Data.* Proc. IEEE Sympos. in Parallel and Large Data Visualization and Graphics, 2001, pp.19-27

[14] Dey T. K., Giesen J., Leekha N., Wenger R. *Detecting boundaries for surface reconstruction using co-cones.* Intl. J. Computer Graphics and CAD/CAM, vol. 16, pp.141-159

[15] Dey T. K., Goswami S. *Tight Cocone: A water-tight surface reconstructor.* Proc. 8th ACM Sympos. Solid Modeling application (2003), pp. 127-134 [27]

[16] Edelsbrunner H. *Weighted alpha shapes.* technical report UIUCDCS-R92-1760 DCS University of Illinois at Urbana-Champaign, Urbana, Illinois, 1992

[17] Edelsbrunner H., Mücke E. P. *Three-dimensional alpha shapes.* ACM Trans. Graphics 13, 1994, pp.43-72

[18] Emelyanov A. I. *Surface reconstruction from clouds of points* PhD thesis, University of West Bohemia, Pilsen, Czech Republic 2004

[19] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. *Surface reconstruction from unorganized points.* Computer Graphics 26 (2), 1992, pp.71-78

[20] Hoppe H., DeRose T., Duchamp T., Halstead M., Jin H., McDonald J., Schweitzer J.,

Stuetzle W. *Piecewise smooth surface reconstruction.* SIGGRAPH, 1994, pp. 295 - 302

[21] Levoy M., Ginsberg J., Shade J., Fulk D., Pulli K., Curless B., Rusinkiewicz S., Koller D., Pereira L., Ginzton M., Anderson S., Davis J. *The digital Michelangelo project: 3D scanning of large statues.* Int. Conference on Comp. Graphics and Inter. Techniques 2000, pp. 131 - 144

[22] Mencl R. *A graph based approach to surface reconstruction.* Comp. Graph. forum 14(3), EUROGRAPHICS 1995, pp.445-456

[23] Mencl R., Müller H. *Interpolation and approximation of surfaces from three-dimensional scattered data points.* EUROGRAPHICS 1998, pp.223-233.

[24] Mencl R., Müller H. *Graph based surface reconstruction using structures in scattered point sets.* Proc. CGI, 1998, pp.298-312

[25] Müller J. V., Breen D. E., Lorenzem W. E., O'Bara R. M., Wozny M. J. *Geometrically deformed models: A Method for extracting closed geometric models from volume data.* Proc. SIGGRAPH, 1991, pp.217-226

[26] Szeliski R., Tonnesen D. *Surface modeling with oriented particle systems.* Comp. Graphics 26, 1992, pp.185-194

[27] Varnuška M., Kolingerová I. *Manifold extraction in surface reconstruction.* ICCS 2004, Krakow, Poland, pp. 147 - 155

[28] Varnuška M., Kolingerová I. *Boundary filtering in surface reconstruction.* ICCSA 2004, Assissi, Italy, pp. 682 - 691