A Genetic Approach to Detect Boundaries for Meshless Parameterization and Surface Reconstruction

Jan Maes *Adhemar Bultheel*

Report TW 384, January 2004



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

A Genetic Approach to Detect Boundaries for Meshless Parameterization and Surface Reconstruction

Jan Maes *Adhemar Bultheel*

Report TW 384, January 2004

Department of Computer Science, K.U.Leuven

Abstract

In this paper we present a boundary detection algorithm for unorganized point sets and apply it for surface reconstruction using the meshless parameterization method from *M. S. Floater and M. Reimers, Meshless parameterization and surface reconstruction, Comp. Aided Geom. Design, 18:77-92, 2001*. The algorithm consists of two steps. Under mild assumptions, we can prove that the first step of our algorithm detects all boundary vertices of a sample. Hence, the first step of the algorithm is rather conservative in detecting boundary vertices and might declare some interior vertices as boundary vertices. In the second step we use a genetic algorithm to distinguish the 'real' boundary vertices from the 'false' boundary vertices. The effectiveness of the algorithm is shown with a number of experimental results.

Keywords : boundary, parameterization, triangulation, surface reconstruction, genetic algorithm
AMS(MOS) Classification : 68U05, 68U07, 92D99

A Genetic Approach to Detect Boundaries for Meshless Parameterization and Surface Reconstruction

Jan Maes Adhemar Bultheel

January 2004

Abstract

In this paper we present a boundary detection algorithm for unorganized point sets and apply it for surface reconstruction using the meshless parameterization method from *M. S. Floater and M. Reimers, Meshless parameterization and surface reconstruction, Comp. Aided Geom. Design, 18:77-92, 2001.* The algorithm consists of two steps. Under mild assumptions, we can prove that the first step of our algorithm detects all boundary vertices of a sample. Hence, the first step of the algorithm is rather conservative in detecting boundary vertices and might declare some interior vertices as boundary vertices. In the second step we use a genetic algorithm to distinguish the 'real' boundary vertices from the 'false' boundary vertices. The effectiveness of the algorithm is shown with a number of experimental results.

Keywords: boundary, parameterization, triangulation, surface reconstruction, genetic algorithm

AMS(MOS) classification: 68U05, 68U07, 92D99

1 Introduction

We are interested in the following problem: compute a piecewise linear approximation to a smooth surface S from a point set P sampled from S. Many applications in CAD, computer graphics and scientific computations involve such reconstruction problems. Floater and Reimers [5] have presented a simple method for triangulating unorganized points which are assumed to be sampled from a single surface patch. The basic idea is to map the points into some convex parameter domain in the plane. This is called *meshless parameterization* since the mapping is independent of any given topological structure. Then, by triangulating the parameter points, one obtains immediately a corresponding triangulation of the original data set.

Meshless parameterization generalizes the convex combination method of [4], and assumes that one can identify a sequence of points to serve as the boundary of the triangulation. Floater and Reimers used a straightforward method in [5] to detect the boundary. A point of the data set is identified as a boundary point if it belongs to the boundary of a local triangulation. Otherwise it is regarded as an interior point. This method might work well on many point sets, but one can imagine that it fails when dealing with undersampled regions in a sample.

Dey *et al.* [2, 3] presented a boundary detection algorithm that could be justified theoretically when imposing severe restrictions upon the samples. Their method makes use of the Voronoi diagram of the input samples. Because computing the Voronoi diagram is time consuming it is not very well suited to be used with the meshless parameterization method, but their method is easily invoked in the CO–CONE algorithm by Amenta *et al.* [1].

We present a boundary detection algorithm that can easily be used for meshless parameterization. Our algorithm for generating a boundary consists of two steps. For sufficiently dense samples the first step of the algorithm identifies all boundary vertices correctly. It might also declare some of the interior vertices boundary. Moreover the algorithm assigns a probability to each detected boundary vertex. The second step of the algorithm distinguishes the 'real' boundary vertices from the 'false' boundary vertices by finding the best combination of high probability boundary points.

A genetic algorithm is used to accomplish this task. This combination of boundary points will then serve as the boundary of the triangulation. The robustness of our algorithm is tested by a number of experiments. All of them give very nice results.

2 The Basic Detection Method (step 1)

2.1 Definitions of Boundaries

Consider a compact smooth surface patch S which is sampled by a point set P. By a surface patch we mean a surface with disk–topology. Interior points of S can be distinguished from boundary points by the following definitions. An *interior point* of S has a neighbourhood homeomorphic to the open disc

$$\mathbb{D}^2 = \{x \in \mathbb{R}^2 : |x| < 1\}.$$

A *boundary point*, on the other hand, has a neighbourhood homeomorphic to the halfdisc $\mathbb{D}^2 \cap \mathbb{H}^2_+$ which is open in the halfspace

$$\mathbb{H}^2_+ = \{(x, y) \in \mathbb{R}^2 : x \geq 0\}.$$

These definitions are valid for the continuous case, i.e. the surface S itself. Now, suppose we are given a sample P of S. It is possible that all points in P are interior points of S, but the characteristics of a boundary should be reflected in the sample points. We need a definition for interior and boundary points of a finite sample P from a surface S that captures the intuitive difference between interior and boundary points. First we will give a definition for the neighbourhood of a sample point. In this definition we still assume that the surface S is known, because this will define the geodesics.

Definition 2.1 *The k–neighbourhood N_p of a sample point p from a sample P of S is defined as the interior of the smallest compact area of S that contains the geodesic curves on S that join the k geodesic nearest sample points of p to each other, union the point p.*

We can use this definition of neighbourhood to define interior sample points and boundary sample points analogous to the definition in the continuous case.

Definition 2.2 *A sample point p from a sample P of S is called an interior sample point if its k–neighbourhood is homeomorphic to the open disc $\mathbb{D}^2$. Sample points that are not interior are called boundary sample points.*

Remark that some sample points might be declared as boundary sample point, though they are in fact interior points of the surface S. In Section 2.3 we shall modify our definition of interior sample point such that only the sample P is used, since in practice the surface S is unknown. In Section 2.4 we prove that the boundary sample points of Definition 2.2 are a subset of the detected boundary samples.

2.2 Local Surface Properties

Since Definition 2.2 depends on local neighbourhoods it would be interesting to estimate the surface S locally. In this section we describe how local surface properties can be estimated from a sample P. It has been demonstrated in earlier work (e.g. [6]) that eigenanalysis of the covariance matrix of a discrete local neighbourhood $N_p \cap P$ is very valuable. The 3×3 covariance matrix C for a sample point p is given by

$$C = \begin{bmatrix} p_1 - \bar{p} \\ \dots \\ p_k - \bar{p} \end{bmatrix}^T \cdot \begin{bmatrix} p_1 - \bar{p} \\ \dots \\ p_k - \bar{p} \end{bmatrix}, \qquad p_i \in N_p \cap P \setminus \{p\},$$

where $\bar{p}$ is the centroid of the neighbours p_i of p. Since C is symmetric and positive semi–definite, all eigenvalues λ_l, $l \in \{1, 2, 3\}$, of C are real–valued and the corresponding eigenvectors v_l form

an orthonormal frame. They correspond to the principal components of the point set defined by $N_p \cap P \setminus \{p\}$. The eigenvalues λ_l measure the variation of the p_i along the direction of the corresponding eigenvectors. Hence the vectors $\lambda_l v_l$ define an ellipsoid that captures the variability of the points p_i. This ellipsoid is called the *covariance ellipsoid*.

Assuming $\lambda_1 \leq \lambda_2 \leq \lambda_3$, it follows that the plane

$$T : (x - \bar{p}) \cdot v_1 = 0$$

through $\bar{p}$ minimizes the sum of squared distances to the neighbours of p. In other words, v_2 and v_3 span the least squares plane fitted through the neighbours of p. This least squares plane will be used as an approximation for the tangent plane of S at p.

In the remainder of the paper we make four assumptions about the samples. The first assumption guarantees that the projection onto the plane T is well defined.

Assumption 2.3 *For each sample point $p \in P$ we assume that the projection of the continuous local k–neighbourhood N_p onto the least squares plane T of the point set $N_p \cap P \setminus \{p\}$ is 1–1.*

To accent the 2–dimensionality of S we do not want that the second principal direction v_2 is negligible against the main principal direction v_3. The second assumption deals with this problem.

Assumption 2.4 *Define $\lambda_1 \leq \lambda_2 \leq \lambda_3$ as the eigenvalues of the covariance matrix C of a sample point $p \in P$. Since S is a smooth 2–dimensional surface, we assume that P is sampled densely enough such that $\frac{\lambda_2}{\lambda_1+\lambda_2+\lambda_3} \geq \delta$ for each sample point p, with $\frac{1}{2} > \delta > 0$ a constant. In other words, the second principal direction v_2 cannot be neglected.*

We make a third assumption concerning the geodesic distance versus the Euclidean distance. The geodesic distance has the advantage of being sensitive to global features of the surface S like two layers of S coming very close to each other. However, since S is unknown we must use the Euclidean distance.

Assumption 2.5 *We assume that P is sufficiently densely sampled such that the k geodesic nearest sample points to p are equal to the k Euclidean nearest sample points to p for each sample point $p \in P$.*

The last assumption concerns the error of the approximation of the tangent plane by the least squares plane T.

Assumption 2.6 *Define ϕ_p as the angle between the tangent plane of the surface S at p and its least squares approximation T, with p a sample point in P. Define ϕ as the maximum of all angles ϕ_p for all sample points $p \in P$. Then we assume that $\phi \leq \gamma$ with γ a constant and $0 \leq \gamma < \frac{\pi}{2}$.*

Note that Assumptions 2.4 and 2.6 are actually more like a definition for the constants δ and γ than a restriction for the sample P.

2.3 Chi Square Goodness–of–Fit Test

To distinguish boundary vertices from interior vertices we cannot use definition 2.2 directly because the surface S is unknown. Therefore we will give a new definition that can be used in practice. The idea is that we construct a homeomorphism h that maps the k–neighbourhood N_p onto a flat disk–like surface. This flat disk–like surface will then be examined and, according to the results of a goodness–of–fit test, we conclude that p is either a boundary point or an interior point.

Suppose p is an arbitrary sample point of sample P. If P is sampled densely we may assume that the k Euclidean nearest sample points p_i to p are contained in N_p (Assumption 2.5). This discrete set of points will be our new definition of neighbourhood. Define T as the least squares plane fitted through these points, and define π as the orthogonal projection onto T. Because of Assumption 2.3 we know that π, restricted to N_p, is a homeomorphism, and π maps N_p onto a flat surface.

The variability of N_p is estimated by the covariance ellipsoid of the neighbouring sample points p_i of p. Since a disk has equal variability in all directions we can use the covariance ellipsoid to construct another homeomorphism σ that rescales the discrete neighbourhood of p to a point set with equal variability in both the main principal directions. Hence, the homeomorphism $h = \sigma \circ \pi$ maps N_p onto a flat disk–like surface, see Figure 1.

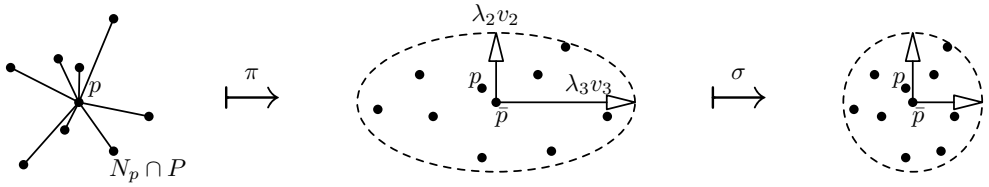


Figure 1: Discrete neighbourhood $N_p \cap P$ and homeomorphism $h = \sigma \circ \pi$

For each of the k nearest samples p_i to p we compute its polar coordinates (r_i, θ_i) with respect to origin p. Assuming that p is an interior point, we expect that the angles θ_i lie rather uniformly spread in the interval $[0, 2\pi)$. This can be tested with a *Pearson χ^2 goodness–of–fit test.*

We partition the interval $[0, 2\pi)$ in r disjunct intervals $S_1, S_2, \ldots, S_r$ of equal length $\frac{2\pi}{r}$. Assuming that the angles θ_i are uniformly distributed over $[0, 2\pi)$ we expect k/r angles θ_i in each interval S_j. The *Pearson–statistic* is computed as

$$\chi_k^2 = \sum_{j=1}^{r} \frac{(n_j - k/r)^2}{k/r}, \tag{1}$$

with n_j the number of angles θ_i lying in S_j. The Theorem of Pearson says that the distribution function $F_k(x)$ of the Pearson–statistic χ_k^2 tends to a χ^2–distribution with $r-1$ degrees of freedom as $k \to \infty$. This theorem can be applied in practice if k is big enough, i.e. k/r has to be greater than or equal to 5.

Before performing the Pearson χ^2 goodness–of–fit test we choose a significance level α. If the probability $Pr(\chi_{r-1}^2 \geq \chi_k^2)$ is smaller than or equal to α we conclude that the angles θ_i are not nicely spread over the interval $[0, 2\pi)$ and the sample point p is defined as a boundary vertex. Otherwise, we conclude that p is an interior vertex.

2.4 Justification

In this section we give evidence of our classification method. We will make use of the implicit function theorem, which we state here without proof.

Theorem 2.7 *Given a surface S, implicitly defined by $f(x, y, z) = 0$, and a point $p \in S$ such that $f_x(p) \neq 0$ and $f_y(p) \neq 0$, then there exists a small neighbourhood W_p of $p \in S$ such that S can be considered locally as a height function $z = h(x, y)$. The map π that projects a point $(x, y, z) \in W_p$ onto the tangent plane of S at p is a 1–1 map with its image being an open set $V_p \in \mathbb{R}^2$. Moreover the map π is a diffeomorphism.*

We impose no conditions on the samples, except that they are sampled densely enough such that Assumptions 2.3, 2.4, 2.5 and 2.6 are valid. Assumptions 2.3 and 2.6 are always possible because of Theorem 2.7, and Assumptions 2.4 and 2.5 are always possible since S is a 2–dimensional surface patch. Under these weak conditions we can prove that all boundary vertices (Definition 2.2) are detected.

Let us assume that our surface S has a C^1 continuous boundary ∂S. Choose an arbitrary boundary point b of S. Since S is a smooth surface, we can define the normal of the surface S at b as the limit of the normals of the interior points p_i of S for which $\|p_i - b\| \to 0$. This limit exists because the surface S is smooth. The tangent plane of S at b is defined as the plane orthogonal to the normal of S at b. Project an arbitrary local neighbourhood of b onto the tangent plane of

S at b. We assume that this local neighbourhood of b is small enough such that the projection is 1–1, which is possible because of Theorem 2.7. Construct two half lines in the tangent plane at b that start in b and point outward. Let θ be the angle between the two half lines. If θ is small enough there exists a rotation of the two half lines in the tangent plane at b such that no point that is contained in the local neighbourhood of b, is projected into the region in between the two half lines (see Figure 2). Define R as the maximum angle θ with this property over all boundary points b of the surface S. Then R is a constant proper to the surface S, and $R > 0$ because ∂S is C^1 continuous.

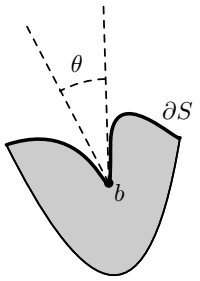


Figure 2: The shaded area is a local neighbourhood of boundary point b, projected onto the tangent plane of the surface S at b. The fat line represents the boundary points of S that are projected onto the tangent plane. The two dotted lines define an area in the tangent plane that contains no projected points.

Let $q \in P$ be a boundary sample point according to Definition 2.2, let q_i be the neighbouring sample points of q and map all these points onto a plane with the homeomorphism h from Figure 1. Recall formula (1) with variables k, r, n_i and intervals S_i. Suppose that r satisfies $r > \frac{2\pi}{\delta \cdot \cos\gamma \cdot (R/2)}$, with δ as defined in Assumption 2.4 and γ as in Assumption 2.6. Then there exists an interval S_j with n_j equal to 0. The Pearson χ^2 test statistic is equal to

$$\chi^2_k = \frac{k}{r} + \sum_{i=1, i\neq j}^{r} \frac{(n_i - k/r)^2}{k/r},$$

hence χ^2_k is of the form $a \cdot k + o(k)$, with $a \neq 0$ a constant. The probability $Pr(\chi^2_{r-1} \geq \chi^2_k)$ will be smaller than the significance level α if k is big enough and the boundary vertex q is detected. We can summarize this in the following theorem.

Theorem 2.8 *Let S be a smooth surface and suppose that S has a C^1 continuous boundary. Let P be a sample of S such that Assumptions 2.3, 2.4, 2.5 and 2.6 are valid. Suppose that r of (1) satisfies $r > \frac{4\pi}{\delta \cdot \cos\gamma \cdot R}$, with δ as in Assumption 2.4, γ as in Assumption 2.6 and R as defined above. Then all boundary vertices of P are detected if k is big enough.*

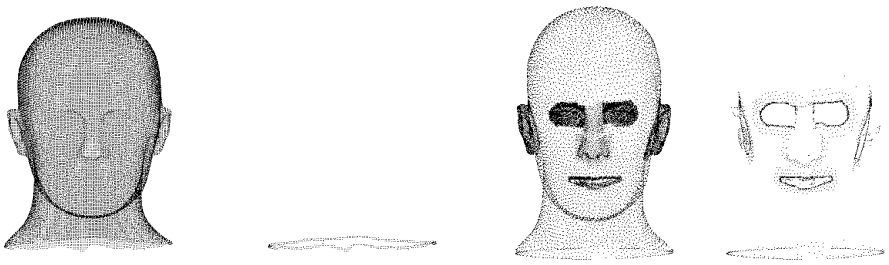
Theorem 2.8 is of theoretical interest only and can't be applied to most of the practical samples, but it gives some justification to the algorithm. The denser P is sampled, the more points can be chosen to define a neighbourhood, the more boundary points are detected. We assume that S has a C^1 continuous boundary ∂S, but, since in practice we are given a sample P and we don't know anything about the real surface S, the assumption that ∂S is C^1 is justified because it is impossible to reflect discontinuity of derivatives in a sampling.

We also give some intuitive explanation that interior vertices generally have a lower Pearson χ^2 test statistic than boundary vertices. In most cases our homeomorphism h will map an interior vertex relatively close to the centroid of its discrete neighbourhood while a boundary point will be mapped relatively far from the centroid. Moreover our homeomorphism is constructed in such a way that the variability of the neighbouring samples is equal in the v_2 and v_3 direction. These arguments suggest that, in the case of an interior vertex, the number of neighbouring sample points

in each interval S_i should be approximately equal, especially when the sample points are uniformly spread over the surface S. Hence the Pearson χ^2 test statistic score of an interior vertex is generally lower than the Pearson χ^2 test statistic score of a boundary vertex.

We conclude that our algorithm is conservative in detecting boundary points. As a consequence it will be easier to construct closed paths of boundary vertices (see Section 3.2). Interior vertices might be detected as boundary vertex. The most common cause herefore is undersampling.

2.5 Some Experiments



(a) Uniform sample

(b) Undersampled regions

Figure 3: Two different mannequin data sets

We have tested the algorithm on two different mannequin data sets (see Figure 3) . The samples of the first data set are approximately uniformly distributed and we can see that the algorithm works in this case. The second data set is sampled more densely around the eyes, ears and mouth than the other parts of the face. In this case the algorithm detects the real boundary but also the borders between the densely sampled regions and the undersampled regions.

To cope with this problem we need to add some extras to the algorithm. A great advantage of our algorithm is that every sample point has a score. Points with a high score are more likely a boundary point than an interior point. We will use this information to construct a genetic algorithm that extracts the only real boundary.

3 Extracting the Real Boundary (step 2)

3.1 Genetic Algorithms

Genetic algorithms [7] are a family of computational models that encode a potential solution to a specific problem on a simple chromosome–like data structure and apply recombination operators to these structures so as to preserve important information. An implementation of a genetic algorithm begins with a population of chromosomes. These structures are evaluated and one allocates reproduction probabilities in such a way that those chromosomes which represent a better solution to the target problem are given more chances to reproduce than those chromosomes which are poorer solutions. The basic algorithm for a Simple Genetic Algorithm can be described as follows:

1. Generate an *initial population.* Each member of this population will be a binary string of length L which corresponds to the problem encoding. Each string is referred to as a *chromosome.*

2. Evaluate each chromosome in the population according to the *evaluation function.*

3. Apply selection to the current population to create an *intermediate population*. The probability that strings in the current population are duplicated and placed in the intermediate population is proportional to their evaluation.

4. Apply recombination and mutation to the intermediate population to create the *next population*. Crossover is applied to randomly paired strings. After recombination, we can mutate each bit in the population with some low probability.

5. Determine whether the stopping criteria are satisfied. If satisfied, stop the iteration. Otherwise go to step 2.

3.2 The Chromosomes

We need some initial population whose members will be referred to as chromosomes. Our chromosomes will be binary strings of length L with L the number of sample points. Each bit in such a string is assigned 1 if the corresponding sample point belongs to the boundary, otherwise the bit is assigned 0. The construction of the chromosomes is based upon the property that the real boundary is a closed path of boundary vertices. Of course sample points that are detected by the basic algorithm as interior points will always have bit 0, whilst sample points that are detected as boundary point might have bit 1.

For each detected boundary point b we apply an algorithm that searches for a closed path of boundary points starting from boundary point b. If a closed path is found, this path will become a chromosome of the initial population. If the initial population contains two identical chromosomes then we keep only one. The construction of a closed path of boundary vertices is described in Algorithm 3.1. It is executed by `FIND_BOUNDARY_PATH(b, b)`, with b a member of the set of boundary points $B \subset P$. The algorithm returns *true* if a closed path is found, otherwise it returns *false*. Initially the boolean variables *inBoundaryPath* and *passed* are set to *false*, the boolean variable *isBoundary* is set by the basic algorithm.

Algorithm 3.1 (FIND_BOUNDARY_PATH($p \in B \subset P$, $b \in B \subset P$))

```
if (b.inBoundaryPath = false)
   countNumberOfBoundaryPointsInPath := 0
compute the discrete k–neighbourhood N_p of p
if (countNumberOfBoundaryPointsInPath ≥ threshold–value and b ∈ N_p)
   return true
else
   p.passed := true
   p.inBoundaryPath := true
   increase countNumberOfBoundaryPointsInPath
   for each neighbour q ∈ N_p do
      if (q.isBoundary and not q.passed)
         if (FIND_BOUNDARY_PATH(q, b))
            return true
p.inBoundaryPath := false
decrease countNumberOfBoundaryPointsInPath
return false
```

The algorithm makes use of a threshold–value in line 4. A good choice for this threshold is k+1, or a greater value. One can see that smaller values for this threshold will always result in trivial boundary paths.

3.3 The Evaluation Function

As explained before, the basic algorithm is able to give each detected boundary point an individual score. In our implementation we have used an approximation of the following formula:

$$score(b) = \frac{1}{Pr(\chi^2_{r-1} \geq \chi^2_k(b))}. \tag{2}$$

The more we are convinced that b is a boundary point, the higher its score will be.

The evaluation function for a chromosome C_i will be based on the score of the boundary points that it contains. In our implementation the evaluation of a chromosome C_i is given by

$$e(C_i) = \frac{\sum_{b \in C_i} score(b)}{(\text{number of boundary vertices in } C_i)^{9/10}}. \tag{3}$$

Since C_i is defined as a binary string of length L we say that $b \in C_i$ if and only if the bit of C_i that corresponds to sample point b is equal to 1. To avoid giving too high scores to very long boundary paths, we divide the sum of the scores by some function dependent on the number of boundary points. On the other hand we don't want that very small trivial boundary paths get a high score. Therefore we divide the sum of the scores by the number of boundary points to the power 9/10. We have determined the value 9/10 experimentally, but it might be interesting for future research to look for a more theoretically founded evaluation function.

3.4 Extraction of the Boundary with a Genetic Algorithm

In the previous sections we defined our chromosomes and the evaluation function. We are ready to perform the five steps of a Simple Genetic Algorithm as described in Section 3.1. The algorithm stops after K iteration steps.

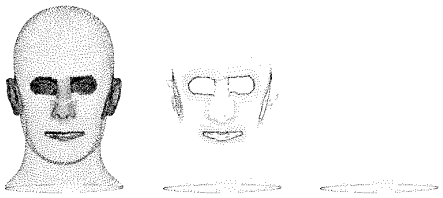
Algorithm 3.2 (APPLY_GENETIC_ALGORITHM)

for all boundary points b **do**
  if (**FIND_BOUNDARY_PATH**(b, b))
    if *path* **not** in *chromosome stack*
      push *path* in *chromosome stack*
for each chromosome C in *chromosome stack* **do**
  evaluate C with evaluation function
set the 100 best chromosomes as the *initial population*
for i **from** 1 **to** K **do**
  create an *intermediate population*
  apply cross–over to the *intermediate population*
  set resulting population as *next population*
  evaluate each chromosome C in the *next population*
return the best chromosome C

4 Experimental Results

We have applied the basic detection method and the genetic algorithm to some example data sets. All the examples are quite severe tests as there is considerable distortion when mapping neighbourhoods into a plane. For each example the basic detection method was applied with k equal to 40 and r equal to 8. Then the condition $k/r \geq 5$ is satisfied. The significance level α was equal to 10%. For the genetic algorithm we chose $k = 40$ (in Algorithm 3.1) for all data sets, except for the Teapot data set we chose $k = 10$. The number of iterations K was equal to 100.

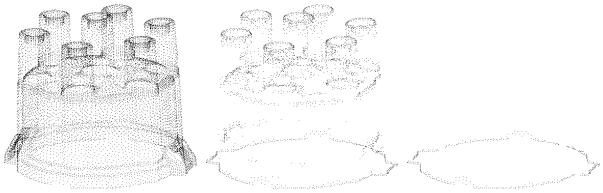
The results of our numerical experiment are depicted in Figure 4. The middle picture of a data set shows the result of the basic detection method and the right picture shows the resulting chromosome of the genetic algorithm. To test robustness we have included some examples with holes in it: for instance the *Distcap* data set, the *Nascar* data set or the *Teapot* data set.



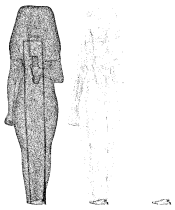
(a) Mannequin data set



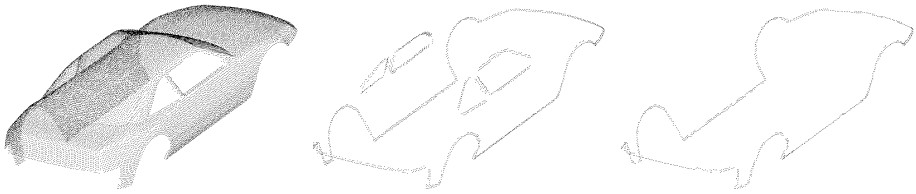
(b) Spock data set



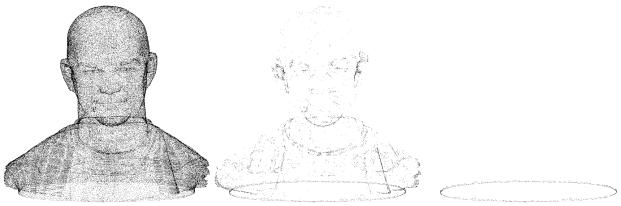
(c) Distcap data set



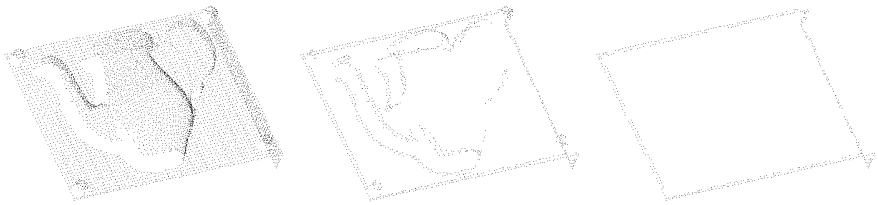
(d) Isis data set



(e) Nascar data set



(f) Male data set



(g) Teapot data set

Figure 4: Results of the boundary detection algorithm

5 Meshless Parameterization

Our goal is to reconstruct a smooth compact single patch surface S from a finite sample $P \subset S$. Floater and Reimers [5] have developed a method for parameterizing and triangulating single patch unorganized point sets P. The points are mapped into a planar parameter domain by solving a sparse linear system. Then a standard triangulation method is applied to the parameter points, and this way a triangulation of the original data set is obtained.

Their method has two steps. In the first step the boundary points $b \in B \subset P$ are mapped into the boundary of some convex polygon D in the plane. Thus the corresponding parameter points lie around ∂D in some predefined order, say anticlockwise. In the second step, some neighbourhood is calculated for each interior point $p \in P/B$. Then a linear system is solved that demands that the parameter point of each interior sample point p is some convex combination of its neighbours. Thus the parameter point corresponding to p will be contained in the convex hull of the parameter points corresponding to the neighbours of p. We can write this down as

$$u_i = \sum_{j \in N_i} \lambda_{ij} u_j, \qquad i = 1, \ldots, n \tag{4}$$

with $u_1, \ldots, u_n$ the parameter points corresponding to the interior points, and $u_{n+1}, \ldots, u_N$ the parameter points corresponding to the boundary points.

We have implemented this parameterization method as an application of our boundary detection algorithm. We chose D to be a disk because of simplicity, and for the λ_{ij}'s we chose the *reciprocal distance weights*

$$\lambda_{ij} = \frac{1}{\|p_j - p_i\|} \Bigg/ \sum_{k \in N_i} \frac{1}{\|p_k - p_i\|} \; .$$

Our triangulation method is the *Delaunay triangulation*. The results can be seen in Figure 5.

6 Conclusion and Future Work

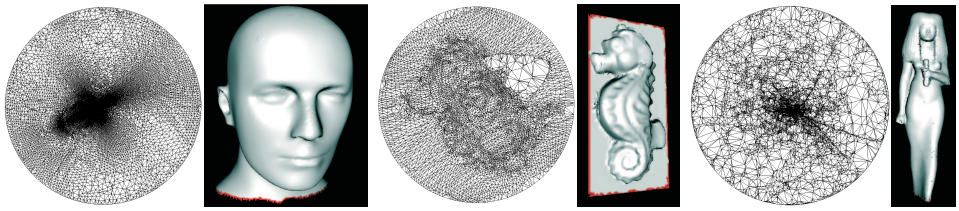
In this paper we present an algorithm to detect the boundary of a surface with disk–topology. This information can be used to reconstruct the surface. As exhibited by our empirical results, our boundary detection algorithm correctly identifies the vertices that are lying on the boundary. We provide a partial justification of the basic algorithm, though we believe that more can be stated theoretically by imposing slightly severe restrictions on the samples.

A side effect of the basic algorithm is that it identifies the regions of undersampling. This aspect of the algorithm might be useful to identify sharp features such as sharp edges and corners from the samples. Also we can use this side effect to fill up holes: for instance the windows in the *Nascar* data set are detected by the basic algorithm and this information can be used to fill up these windows in order to get a surface with disk–topology (see Figure 4).

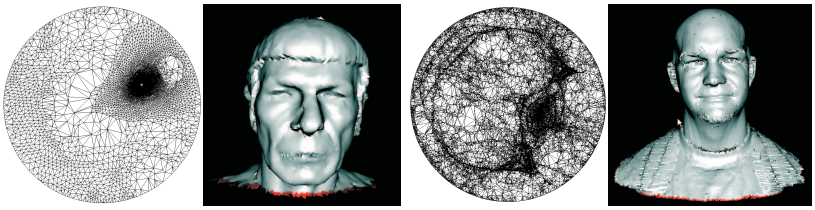
The genetic approach comes in handy to distinguish the more important boundary points from the less important. However our evaluation function is experimental and it could be interesting to investigate this part of the algorithm more profoundly.

Acknowledgements

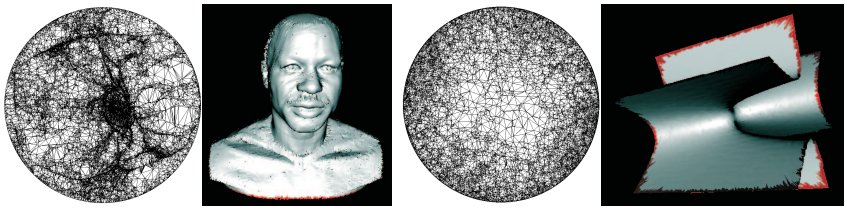
This work is partially supported by the Flemish Fund for Scientific Research (FWO Vlaanderen) project MISS (G.0211.02), and by the Belgian Program on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with the authors. We wish to thank Patrick Van gucht for suggesting the use of genetic algorithms. We also wish to thank Cyberware, Metris and Hugues Hoppe for the data sets.



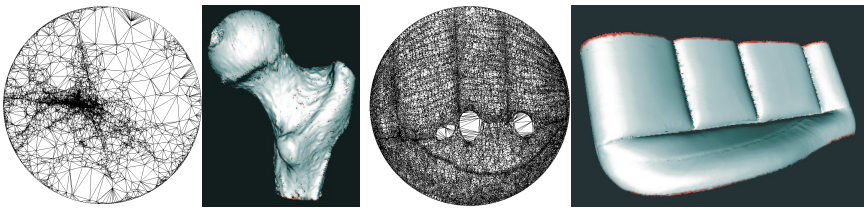
(a) Mannequin data set (b) Seahorse data set (c) Isis data set



(d) Spock data set (e) Male data set



(f) Male 2 data set (g) Monkey data set



(h) Balljoint data set (i) Sofa data set

Figure 5: Results of the meshless parameterization algorithm

References

[1] N. Amenta, S. Choi, T. K. Dey and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *International Journal of Computational Geometry and Applications*, 12(1-2):125-141, 2002.

[2] T. K. Dey and J. Giesen. Detecting undersampling in surface reconstruction. In *Proc. 17th ACM Sympos. Comput. Geom.*, pages 257-263, 2001.

[3] T. K. Dey, J. Giesen, N. Leekha and R. Wenger. Detecting boundaries for surface reconstruction using co–cones. *Intl. J. Computer Graphics & CAD/CAM*, 16:141-159, 2001.

[4] M. S. Floater. Parameterization and smooth approximation of surface triangulations. *Comp. Aided Geom. Design*, 14:231-250, 1997.

[5] M. S. Floater and M. Reimers. Meshless parameterization and surface reconstruction. *Comp. Aided Geom. Design*, 18:77-92, 2001.

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stueltzle. Surface reconstruction from unorganized points. *SIGGRAPH 92*, 1992.

[7] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65-85, 1994.