Meshless Surface Reconstruction by Kernel Clustering*

Joachim Giesen† Simon Spalinger†

Abstract

We discuss a meshless approach to the problem of reconstructing a surface in $\mathbb{R}^3$ from a finite sampling. This approach is a direct adaptation of a kernel method for clustering points in Euclidean space. The reconstructed surface is the preimage of the boundary of the smallest enclosing ball of the sample points mapped into some feature space. We have implemented this approach and report on our experimental findings that indicate that it might be suitable not only for surface reconstruction but also for feature detection and reconstructing manifolds of higher co-dimension.

Keywords. Curve and surface reconstruction, kernel methods, support vector clustering

1 Introduction

Surface reconstruction is the problem of computing a continuous model of a surface embedded in $\mathbb{R}^3$ only from a finite set of sample points. The model should share as many topological and geometric properties with the original surface as possible. The predominant approach towards surface reconstruction in computational geometry is via surface meshing from the Delaunay triangulation of the sample points [1, 2]. These meshes can be proven to share many properties with the original surface. In computer graphics recently meshless approaches to the surface reconstruction problem became popular [5]. In these approaches the sample point set is used to compute a function $f : \mathbb{R}^3 \to \mathbb{R}$ whose zero set, i.e., $f^{-1}(0)$, is used as a surface model. Sometimes the function f is chosen to be a linear combination of a set of radial basis functions [5]. This is also the approach we want to take here.

Our approach is the direct adaptation of a technique developed by Schölkopf et al. [9] for novelty detection and by Ben-Hur et al. [3] who extended this approach to cluster points in Euclidean space. The approach is to map the smallest enclosing ball problem non-linearly into some feature space. The miniball problem, i.e., the problem to compute the center and radius the smallest ball that contains all sample points, has long been studied in computational geometry [6]. It is well known that it is an instance of a convex quadratic problem (QP). In the QP formulation of the miniball problem the sample points only appear in dot products. This makes the QP formulation of the miniball problem amenable to the so called kernel trick. The kernel trick is to replace all dot products in input space by evaluations of a positive semidefinite kernel at the sample points. A positive semidefinite kernel is a positive semidefinite function $k : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$. Applying the kernel trick corresponds to implicitly mapping the sample points to some feature space. The miniball problem is then solved in the feature space. The surface model we want to study is the set of points in input space, i.e., $\mathbb{R}^3$, that are mapped to boundary sphere of the miniball in feature space. If our kernel would be just the ordinary dot product in $\mathbb{R}^3$ then the surface model would just be the surface of the miniball in $\mathbb{R}^3$. Other kernels give surfaces that adapt much better to the "geometry" of the point set, see Figure 1 for a two-dimensional example.

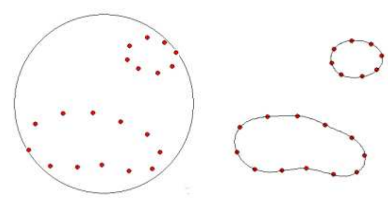

Figure 1: On the left: Miniball of a set of points in $\mathbb{R}^2$. On the right: The preimage of a miniball in feature space.

2 Miniball in Feature Space

In this section we introduce the miniball problem in feature space. Our exposition follows [3]. The feature mapping

$$\phi : \mathbb{R}^3 \to \mathcal{F}, x \mapsto \phi(x)$$

maps our input space $\mathbb{R}^3$ to some feature space $\mathcal{F}$. In general $\mathcal{F}$ has a dimension much larger than three or is even infinite dimensional. In $\mathcal{F}$ we compute the radius R_0 and the center a of the minimum enclosing ball of the mapped sample points, i.e., we solve the following optimization problem,

$$\begin{array}{ll} \min & R^2 \\ \text{s.t.} & \|\phi(x_j) - a\|^2 \leq R^2 \quad \forall \text{ sample points } x_j \end{array}$$

It turns out that the solutions of this optimization problem tend to be unstable in our application - that is in computing a surface model from the input points. The computed surfaces either consist of many components or the surface tends to be "blobby", see Figure 3 for examples. This phenomenon is quite well known in machine learning and in general referred

*Partly supported by the Swiss National Science Foundation under the project "Non-linear manifold learning".

†Institut für Theoretische Informatik, ETH Zürich, {giesen,spsimon}@inf.ethz.ch

to as overfitting. One way to deal with overfitting is to relax the constraints, i.e., we no longer demand that all sample points have to be contained in the ball but we allow some outliers. Since we do not want too many outliers we have to penalize them in the objective function. That leads to the following optimization problem,

$$\begin{aligned} \min \quad & R^2 + c\sum_j \xi_j \\ \text{s.t.} \quad & \|\phi(x_j) - a\|^2 \leq R^2 + \xi_j \\ & \xi_j \geq 0 \end{aligned}$$

With the variables $\xi_j \geq 0$ we weaken the constraints and constraint violations are penalized by the term $c\sum_j \xi_j$ in the objective function. By adjusting the value of c one can control the violation of the constraints. The Kuhn-Tucker conditions for this optimization problem guarantee the existence of $\beta_j, \mu_j \geq 0$ such that

$$\begin{aligned} (R^2 + \xi_j - \|\phi(x_j) - a\|^2)\beta_j &= 0 \\ \xi_j \mu_j &= 0 \end{aligned}$$

That is, the image of a sample point x_i lies on the boundary of the optimal ball if $\xi_i = 0$ and $0 < \beta_i < c$. Points for which $\beta_i = 0$ lie inside the sphere. For $\xi_i > 0$ the image of the point x_i lies outside the optimal ball.

It turns out for several reasons that it is much more convenient to work with the dual of this optimization problem. The dual problem is obtained from the Lagrangian function L of the primal optimization problem. The Lagrangian function is the objective function minus a linear combination of the constraints with non-negative coefficients,

$$\begin{aligned} L \;=\;& R^2 + c\sum_j \xi_j - \sum_j \beta_j (R^2 - \xi_j - \|\phi(x_j) - a\|^2) \\ & - \sum_j \xi_j \mu_j, \quad \beta_j, \mu_j \geq 0 \end{aligned}$$

The Lagrange multiplier theorem states that an optimum of the primal problem corresponds to a saddle point of the Lagrangian function. Since a saddle point is a critical point of L the gradient of L has to vanish at such a point. That is, the partial derivatives of L with respect to R, a and ξ_j have to vanish. From this we get

$$\sum_j \beta_j = 1, \quad a = \sum_j \beta_j \phi(x_j), \quad \beta_j = c - \mu_j.$$

These equations can be used to eliminate the variables R, a and ξ_j from the Lagrangian function which then only depends on the β_j,

$$L' \;=\; \sum_j \beta_j \phi(x_j)^2 - \sum_{i,j} \beta_i \beta_j \phi(x_i)^T \phi(x_j)$$

A saddle point of L corresponds to a local maximum of L'. Thus we obtain the following optimization problem,

$$\begin{aligned} \max \quad & \sum_j \beta_j \phi(x_j)^2 - \sum_{i,j} \beta_i \beta_j \phi(x_i)^T \phi(x_j) \\ \text{s.t.} \quad & 0 \leq \beta_j \leq c \end{aligned}$$

Solving this optimization problem we get the center a of the miniball as $a = \sum_j \beta_j \phi(x_j)$. For every point $x \in \mathbb{R}^3$ the squared distance $R^2(x)$ of its image in feature space from the center a of the sphere is

$$R^2(x) \;=\; \|\phi(x) - a\|^2$$

The sample points x_i whose image lie on the surface of the optimal ball are called *support vectors*. The radius R_0 of the optimal ball satisfies the following

$$R_0^2 \;=\; \{\, R^2(x_i) \mid x_i \text{ a support vector } \}.$$

Observe that in the dual optimization problem we do not need to access the mapped sample points $\phi(x_i)$ but only to the value of the dot products $\phi(x_i) \cdot \phi(x_j)$. This can be exploited by applying the so called *kernel trick* and substituting dot products in input space with kernel function evaluations in feature space, i.e.,

$$\phi(x) \cdot \phi(y) \;=\; K(x, y).$$

Making this substitution relieves us from computing dot products in $\mathcal{F}$ and makes the mapping ϕ implicit. The definition of the radius of the optimal ball as well as the distance function from the center remain the same, only the squared distance function now reads as follows,

$$\begin{aligned} R^2(x) &= \|\phi(x) - a\|^2 \qquad (1) \\ &= K(x, x) - 2\sum_j K(x, x_j) \\ &\quad + \sum_{i,j} \beta_i \beta_j K(x_i, x_j). \end{aligned}$$

For the rest of this paper we will work with the Gaussian Kernel

$$\phi(x) \cdot \phi(y) \;=\; K(x, y) \;=\; e^{-\lambda \|x - y\|^2}$$

This kernel is popular in machine learning, graphics and approximation theory. Note that the function

$$K(\cdot, x_i) : \mathbb{R}^3 \to \mathbb{R}, x \mapsto K(x, x_i) = e^{-\lambda \|x - x_i\|^2}$$

is a radial basis function. Thus $R^2(x)$ is just a linear combination of radial basis functions centered at the support vectors.

3 Reconstruction

We use the distance function to the center of the optimal ball to define a surface model. The surface S is the set of points

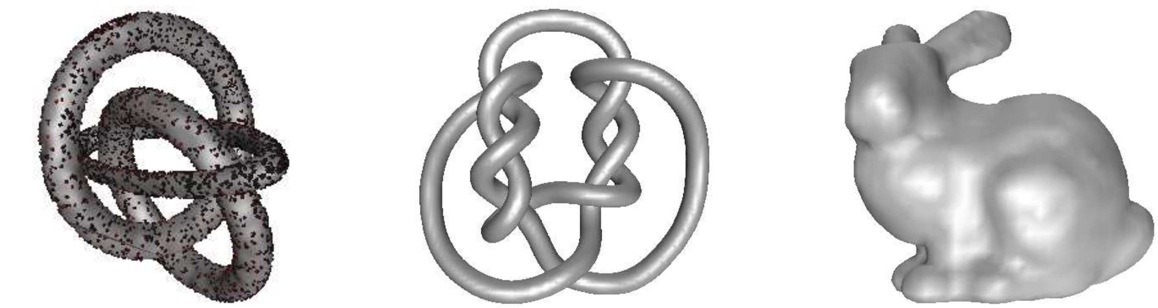

Figure 2: On the left: Three non-touching rings were reconstructed from 6000 sample points. In the middle: A knotted torus is reconstructed from ten thousand sample points. All sample points are support vectors. On the right: The Stanford bunny model is reconstructed from 33,947 sample points.

$x \in \mathbb{R}^3$ whose image $\phi(x)$ lie on the boundary of the optimal ball in feature space, namely

$$S = \{ x \in \mathbb{R}^3 \mid R(x) = R_0 \}$$

where $R(x)$ is the distance function corresponding to (1). In other words, the surface S is the preimage $R^{-1}(R_0)$ of the distance R_0 of the support vectors from the center of the optimal ball. Since R is a smooth function we get from Sards theorem [4] its critical values have Lebesgue measure zero. This implies that for all values of R_0 besides a set of measure zero $S = R^{-1}(R_0)$ is a two-dimensional manifold.

The function R depends on two parameters λ and c. If the sample points are not noisy then they should all lie on the surface S. Thus all sample points should be support vectors. This is achieved for fairly large values of λ. For small values of λ there are few supports vectors, the surface S tends to be blobby and it tends to have the topology of a sphere. On the other hand for large values of λ the surface gets disconnected - even if the original surface was connected. For very large values of λ every sample point lies on its own component of the surface S. Both phenomena (blobbiness and disconnectedness) can be controlled with the second parameter c which is called *regularization parameter* in machine learning. The effect of the regularization parameter can be seen in Figure 3. The challenge is to find values for the parameters λ and c which work well on almost all sets of sample points.

We experimented with this approach towards surface reconstruction. In Figure 2 we show some results. For rendering an implicit surface we approximated it with a mesh the we obtained using a variant of the marching cubes algorithm provided by Lewiner et al. [7]. The examples of the three rings and the knot demonstrate that this approach can handle complex topologies like several connected components or non-trivial isotopy type.

Furthermore, varying the parameters λ and c seems to reveal topological properties of the surfaces like boundaries. This is demonstrated in Figure 4 in a one-dimensional example with a zero-dimensional boundary and a two-dimensional example with a one-dimensional boundary.

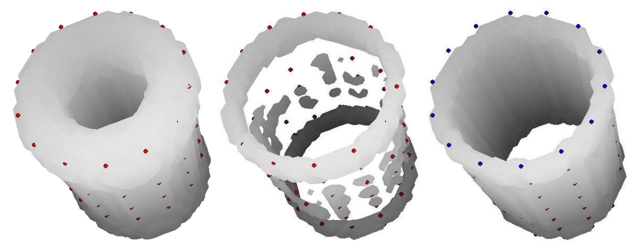

Figure 3: The reconstructed surface depends on the parameters λ and c. Left: $\lambda = 20, c = 1$ results in a blobby surface with about 75% of the sample points lying on the surface. Middle: All points lie on the disconnected surface for $\lambda = 80, c = 1$. Right: Allowing outliers $c = 0.00623$ the surface stays connected for high λ.

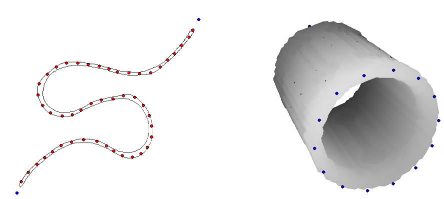

Figure 4: On the right: The two outliers are exactly the endpoints of the non-closed curve in $\mathbb{R}^2$. On the left: The outliers are exactly the points on the two one-dimensional boundaries of the tube.

We observed that sample points near sharp features of the input models tend to receive higher values β_i. This is demonstrated in Figure 5 where the sample points along the edges of the the cube get larger β_i values than the points on its facets. Thus it seems to be possible to detect not only topological but also geometric features from the samples.

Since we have only one equation this approach seems inherently restricted to allow only the reconstruction of two dimensional surfaces - the equation eliminates one degree of freedom. In our experiments we observed that when the

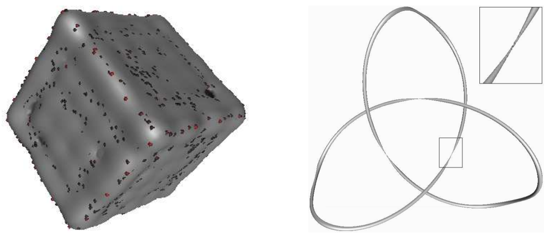

Figure 5: On the right: Randomly sampled cube. On the left: Reconstructing a curve in $\mathbb{R}^3$. Note though the reconstruction is a surface it looks one-dimensional on a large scale.

method is applied to samples from a curve then the additional dimension in the reconstruction gets furled. See Figure 5 for an example. This observation indicates that it could be possible to get also meaningful reconstructions of curves.

4 Practical Issues and Outlook

So far we could not find a shape-independent strategy to determine the parameters λ and c. Different shapes showed different sensitivity to variations of the parameters, see Figure 6.

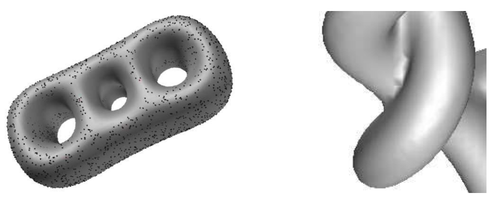

Figure 6: The Three-Holes topology is properly reconstructed with only 10% of the sample points being support vectors while in the case of the knot even 85% of the points are not enough.

Solving the dual optimization problem requires a quadratic program to be solved with the number of variables equal to the number of sample points. This makes exact solutions impractical. We computed an approximate solution by adaptating the sequential minimal optimization (SMO) method [8]. The accuracy of the solution however has a non-neglectable impact on the surface.

To render the models we computed triangular meshes with a variant of the marching cubes algorithm [7]. This method needs many evaluations of the function (1) which is costly since it is a combination of as many radial basis functions as there are support vectors. For faster meshing methods to efficiently evaluate large sums of radial basis functions as discussed in [5] should be adapted.

At the moment our approach is certainly not competitive with the state of the art in three dimensional surface reconstruction. But it seems to be a promising method to compute models of manifolds with higher co-dimension from a finite sampling. Note, that the complexity of solving the dual optimization problem depends only on the number of sample points but not on the dimension.

References

[1] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. In *Proceedings of the fourteenth annual symposium on Computational geometry*, pages 39–48. ACM Press, 1998.

[2] N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 213–222. ACM Press, 2000.

[3] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[4] T. Bröcker and K. Jänich. *Einführung in die Differentialtopologie*. Springer, 1990.

[5] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM Press, 2001.

[6] B. Gärtner and S. Schönherr. An efficient, exact, and generic quadratic programming solver for geometric optimization. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 110–118. ACM Press, 2000.

[7] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient implementation of marching cubes cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003.

[8] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines, 1998.

[9] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 582–588. MIT Press, 2000.