Eurographics Symposium on Geometry Processing (2006)
Konrad Polthier, Alla Sheffer (Editors)

Overfitting Control for Surface Reconstruction

Yunjin Lee[1] Seungyong Lee[1] Ioannis Ivrissimtzis[2] Hans-Peter Seidel[3]

[1]POSTECH [2]Coventry University [3]MPI Informatik

Abstract

This paper proposes a general framework for overfitting control in surface reconstruction from noisy point data. The problem we deal with is how to create a model that will capture as much detail as possible and simultaneously avoid reproducing the noise of the input points. The proposed framework is based on extra-sample validation. *It is fully automatic and can work in conjunction with any surface reconstruction algorithm. We test the framework with a Radial Basis Function algorithm, Multi-level Partition of Unity implicits, and the Power Crust algorithm.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.6.5 [Simulation and modeling]: Model Development

1. Introduction

In this paper, we deal with the problem of data *overfitting* in surface reconstruction. Overfitting appears when we model a given sample so faithfully that we capture not only information about the underlying surface but also the idiosyncrasies of the sample, that is, the noise in the points. Fig. 1 shows an example of overfitting.

Instead of modifying any of the existing algorithms, we propose a general framework for handling overfitting, which can be used in conjunction with any surface reconstruction technique. Our framework is based on a simple and accurate method for error estimation, *extra-sample validation* [HTF01]. The initial data set is randomly subdivided into two distinct subsets, the *training* set and the *validation* set. Data from the training set are used for trials of surface reconstruction, while the quality of reconstruction is assessed using the validation set. To have trials of reconstruction with increasing surface complexity, we use a hierarchical partition of the training data, based on an octree. We compute a representative training sample for each octree cell and a surface is created by applying a reconstruction algorithm to the training samples from the leaf cells of the octree.

1.1. Related work

In the area of surface reconstruction, [HDD*92, TL94, BBX95, CL96] are some of the earlier algorithms that influenced the field. More recently, implicit techniques emerged

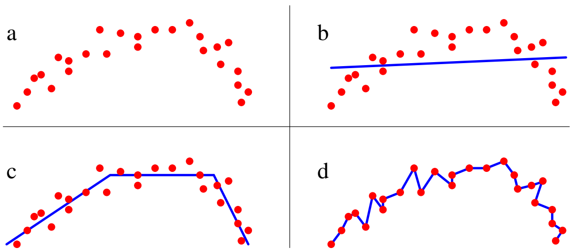

Figure 1: *Curve reconstruction: (a) sample points; (b) underfitted model; (c) correct model; (d) overfitted model.*

as the fastest and more stable techniques. The most common choices of implicits are the radial basis functions (RBFs) [CBC*01, OBS03] and quadrics [OBA*03]. Delaunay tetrahedrization has also been successfully used for surface reconstruction [ACK01, DG03]. In this paper, we experiment with the the techniques in [OBS03] and [OBA*03] and the Power Crust algorithm [ACK01].

In the literature of surface reconstruction, relatively little attention has been paid to the problem of overfitting. Ohtake *et al.* [OBS04] proposed an algorithm which penalizes overfitting by adding a regularization term to the usual distance error metric between the model and a sample. However, they did not present an automatic method to control the regularization term. Steinke *et al.* [SSB05] use Support Vector

Machines for surface reconstruction. They avoid overfitting with a regularization term which is determined with extra-sample validation. However, as in [OBS04], they treat overfitting as a global phenomenon, with one regularization term applied for the whole model. In many cases, this is not a realistic assumption.

2. Overfitting Control Framework

Following the standard terminology, the *training error* of a given model is the error measured against the training data. The *prediction error* is the expected error between the model and any sample coming from the same source as the training data. Overfitting usually arises when we try to minimize the training error instead of the prediction error. Indeed, the training error typically decreases monotonically with the model complexity, and eventually becomes zero when we interpolate the training data. In contrast, in a typical behavior shown in Fig. 2, the prediction error first decreases with the model complexity and then increases as we start overfitting.

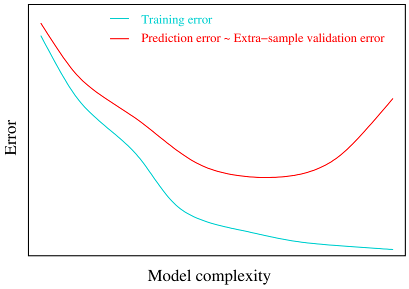

Figure 2: *Typical training and prediction error curves.*

2.1. Simple approach

In a simple approach to overfitting control, we can first split the data into the training and validation sets and then use the training data to reconstruct a series of different surfaces, indexed by one or more algorithmic parameters. We can select the surface with the least error against the validation data as the final output. However, this simple approach has certain limitations. First, the error is computed as an average over the whole surface. This error assumes a uniformly distributed noise, which is rarely the case with scan data. Second, the implementation depends on the specific parameters of a surface reconstruction algorithm and as a result, we need to devise a new strategy for each different algorithm.

2.2. Hierarchical framework

To overcome these limitations, we propose a hierarchical framework based on adaptive spatial subdivision of the input data. We first randomly select a half of the input points to be the training data. The other half of the points are the validation data. The data structure used for spatial subdivision is

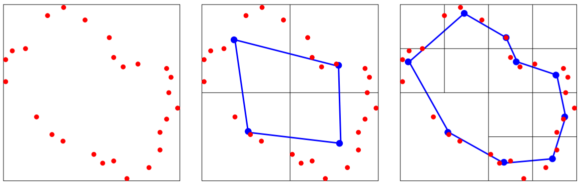

Figure 3: *The red points are training data. The blue points are cell representatives. Piecewise linear interpolating reconstructions are shown in blue.*

a progressively refined octree. At the beginning, the octree consists of a single cell corresponding to the bounding box of the input data. As an initialization step, we recursively subdivide the bounding box up to a few levels.

For a given octree O_l at level l, we split all the leaf cells that have not been marked as "completed" and create a new tree O_{l+1} at the next level $l+1$. Then, we determine a single representative point for the training points in each leaf cell c_{l+1} of O_{l+1} by weighted averaging. We feed these representative points to the surface reconstruction algorithm of our choice to produce a surface S_{l+1}. For the leaf cells of the previous and current octrees, O_l and O_{l+1}, we compute the validation errors against the surfaces S_l and S_{l+1}. A cell c_l passes the overfitting test if the validation errors are decreasing with the subdivision of c_l itself *or* the majority of the child cells c_{l+1}. In this case, we keep the children of c_l in O_{l+1} and mark them as "uncompleted". Otherwise, overfitting is detected and we mark c_l as completed and remove its children from O_{l+1}. For cells which do not have enough validation points for a reliable error estimate, we also use validation points from neighbor cells. When an octree cell contains only one training point, it is marked as completed.

We repeat this process until all leaf cells of the adaptively refined octree are marked as completed. After completing the adaptive spatial subdivision with overfitting control, we collect the representative points from the leaf cells of the final octree. The final surface is obtained by applying the surface reconstruction algorithm to these points. Fig. 3 illustrates a 2D example.

Our overfitting control framework can work with any surface reconstruction technique, which can be considered as a black box. The input of the black box is two point sets, training data and validation data, and the output consists of the reconstructed surface and the validation error. During the overfitting control process, we do not need the actual surface reconstructions but only the error measurements for the validation data. For an implicit-based technique, such as [OBA*03] and [OBS03], we can directly estimate the distance of a validation point from the surface using the Taubin distance [Tau91], as used in [OBA*03]. In the case of the Power Crust, we measure the error from a surface by the

Metro tool [CRS98]. The following pseudocode summarizes our hierarchical framework.

Overfitting control framework
Input: training and validation data, $\mathcal{P}_{tr}$ and $\mathcal{P}_{vl}$.
Output: octree partitioning of the bounding box of $\mathcal{P}_{tr} \cup \mathcal{P}_{vl}$.
$l = 0$;
O_0 = subdivision of the bounding box of $\mathcal{P}_{tr} \cup \mathcal{P}_{vl}$
    up to a few levels;
S_0 = *MakeSurface*($\mathcal{P}_{tr}, O_0$);
while (O_l has uncompleted cells) **do** {
    subdivide uncompleted cells of O_l to create O_{l+1};
    S_{l+1} = *MakeSurface*($\mathcal{P}_{tr}, O_{l+1}$);
    for (each uncompleted cell c_i of O_l) {
        if (test with $\mathcal{P}_{vl}$ detects an overfitting at level $l+1$) {
            mark c_i as completed;
            remove children of c_i from O_{l+1};
        } **else**
            keep and mark children of c_i in O_{l+1}
                as uncompleted;
    }
    l++;
}
return O_l;

MakeSurface($\mathcal{P}, O$): for each cell c of O, compute a representative for the points of $\mathcal{P}$ contained in c and use a surface reconstruction algorithm to return a surface S.

Finally, if we are not sufficiently rich in data and want the reconstruction to involve all the available data, we can perform a 2-fold cross validation [HTF01]. That is, we repeat the whole process after swapping the training and validation data, and merge the two results.

3. Experimental Results

We tested the proposed framework with the RBF interpolation [OBS03] and the Power Crust [ACK01], chosen as examples of interpolating techniques with and without normals. We also used the MPU implicits [OBA*03] as an approximating technique with the use of a small error bound.

Fig. 5 shows reconstructions by these three algorithms. We used a point set sampled from the *tangle cube*,

$$x^4 - 5x^2 + y^4 - 5y^2 + z^4 - 5z^2 + 11.8 = 0, \quad (1)$$

with added noise. Overfitting control successfully reduced the noise in the data set and the results describe the underlying shape more faithfully. Table 1 compares the reconstruction errors. In all cases, overfitting control reduces the maximum error. In the cases of RBF and MPU, overfitting control also reduces the RMS error. For Power Crust, the reconstruction with overfitting control has a smaller number of polygons and we have an increased RMS error.

		RBF	MPU	Power Crust
Max. Error	single app.	0.0208	0.0260	0.0280
	overf. control	0.0117	0.0114	0.0246
RMS Error	single app.	0.0241	0.0203	0.0351
	overf. control	0.0147	0.0148	0.0407

Table 1: *Reconstruction errors measured by the Metro tool.*

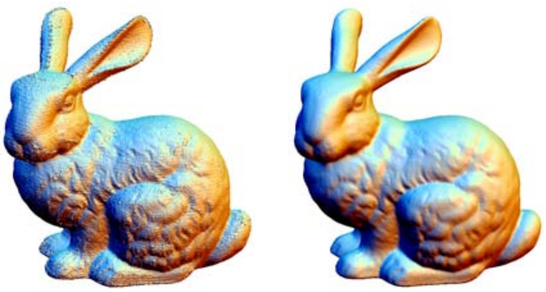

Figure 4: *Left: single RBF. Right: overfitting control.*

Fig. 4 shows the RBF reconstructions of the bunny model without and with overfitting control. The initial point set from the Stanford Digital Model repository contains the original scanning noise. We used the confidence values supplied with the points as weights in the computation of the validation errors.

4. Discussion and Future Work

We propose a framework for the systematic control of overfitting in surface reconstruction. It is fully automatic and can be used in conjunction with any surface reconstruction technique, which can be treated as a black box. The levels of detail are determined by the quality of data, which means that some parts of the reconstruction can have more details than others. On the other hand, as one would expect, there is a computational overhead compared to the corresponding single reconstruction algorithms. We think that this extra computational time is justified as it allows a more informed model selection based on the analysis of the original data.

The effect of overfitting control on the model may be considered similar to that of a postprocessing smoothing step. However, the main difference between these two techniques is that smoothing needs a user-controlled parameter, while overfitting control is based on data analysis. Consequently, in overfitting control, the amount of smoothing is *locally* and *adaptively* determined by the data. In contrast, in a smoothing technique, a user-controlled parameter is *globally* applied to control the smoothing effects because it is tedious or impossible to assign a different parameter value for each specific region.

Recently there has been considerable research on algorithms and representations of point based geometry, such as

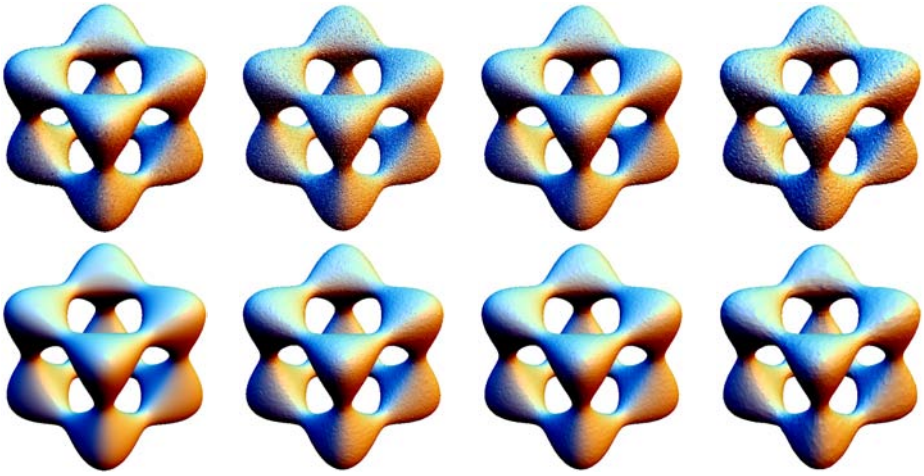

Figure 5: *Bottom left: a point set sample. Top left: the same sample with added noise. The other three figures in the top row show single reconstructions from the noisy sample, while those in the bottom row show the results of overfitting control.*

point set surfaces [ABCO*03]. These techniques are related to this paper in that they also consider processing a given point set to obtain a better one in representing the underlying surface. We hope the main idea of this paper can also be applied to those kinds of techniques, although the details would differ.

Acknowledgements

The bunny model is courtesy of the Stanford Computer Graphics Laboratory. This research was supported in part by the BK21 program, the ITRC support program, and KOSEF (F01-2005-000-10377-0).

References

[ABCO*03] ALEXA M., BEHR J., COHEN-OR D., LEVIN D., FLEISHMAN S., SILVA C. T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (2003), 3–15.

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications 19*, 2-3 (2001), 127–153.

[BBX95] BAJAJ C. L., BERNARDINI F., XU G.: Automatic reconstruction of surfaces and scalar fields from 3D scans. *In Proc. ACM SIGGRAPH 1995* (1995), 109–118.

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3D objects with radial basis functions. *In Proc. ACM SIGGRAPH 2001* (2001), 67–76.

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. *In Proc. ACM SIGGRAPH 1996* (1996), 303 – 312.

[CRS98] CIGNONI P., ROCCHINI C., SCOPIGNO R.: Metro: measuring error on simplified surfaces. *Computer Graphics Forum 17*, 2 (1998), 167–174.

[DG03] DEY T. K., GOSWAMI S.: Tight cocone: A water-tight surface reconstructor. *Journal of Computing and Information Science in Engineering 3*, 4 (2003), 302–307.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. *In Computer Graphics (Proc. ACM SIGGRAPH 1992)* (1992), 71–78.

[HTF01] HASTIE T., TIBSHIRANI R., FRIEDMAN J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2001.

[OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. *ACM Transactions on Graphics 22*, 3 (2003), 463–470.

[OBS03] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In *Proc. Shape Modeling International* (2003), pp. 153– 161.

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Proc. Shape Modeling International* (2004), pp. 31–39.

[SSB05] STEINKE F., SCHÖLKOPF B., BLANZ V.: Support vector machines for 3D shape processing. *Computer Graphics Forum (Proc. Eurographics 2005) 24*, 3 (2005), 285–294.

[Tau91] TAUBIN G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence 13*, 11 (1991), 1115–1138.

[TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. *In Proc. ACM SIGGRAPH 1994* (1994), 311–318.