Available online at www.sciencedirect.com
ScienceDirect
Computer-Aided Design 39 (2007) 408–420
COMPUTER-AIDED DESIGN
www.elsevier.com/locate/cad

Surface and normal ensembles for surface reconstruction

Mincheol Yoon[a], Yunjin Lee[a], Seungyong Lee[a,*], Ioannis Ivrissimtzis[b], Hans-Peter Seidel[c]

[a] *Department of Computer Science and Engineering, POSTECH, San 31, Hyoja-dong, Pohang, 790-784, Republic of Korea*
[b] *Department of Computer Science, Durham University, South Road, Durham DH1 3LE, UK*
[c] *Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany*

Received 14 September 2006; accepted 16 February 2007

Abstract

The majority of the existing techniques for surface reconstruction and the closely related problem of normal reconstruction are deterministic. Their main advantages are the speed and, given a reasonably good initial input, the high quality of the reconstructed surfaces. Nevertheless, their deterministic nature may hinder them from effectively handling incomplete data with noise and outliers. An ensemble is a statistical technique which can improve the performance of deterministic algorithms by putting them into a statistics based probabilistic setting. In this paper, we study the suitability of ensembles in normal and surface reconstruction. We experimented with a widely used normal reconstruction technique [Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. Computer Graphics 1992;71–8] and Multi-level Partitions of Unity implicits for surface reconstruction [Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H-P. Multi-level partition of unity implicits. ACM Transactions on Graphics 2003;22(3):463–70], showing that normal and surface ensembles can successfully be combined to handle noisy point sets.

Keywords: Surface reconstruction; Normal estimation; Ensemble; Probabilistic approach

1. Introduction

Creating a 3D model of a real world object is a lengthy and complicated process. Despite recent progress, the whole procedure is still far from being optimal and may also need some manual input (e.g., see [3,4]). As a result, 3D content is still relatively scarce, slowing the spreading pace of 3D in critical applications like e-commerce.

The 3D modeling pipeline starts with the acquisition stage. We scan the physical object acquiring geometric information, usually in the form of a point cloud. The next task, which is the topic of this paper, is processing the point cloud to obtain a surface representation of the boundary of the scanned object. This problem is known in the literature as *surface reconstruction*.

Surface reconstruction is closely related to the problem of *normal reconstruction* for an unorganized point cloud (also referred to in the literature as normal estimation). The reason is that the fastest and more robust surface reconstruction algorithms require points with normals as input, instead of unorganized points. Clearly, good normal reconstructions are necessary for good surface reconstructions. In our experiments in Section 7, we found that outlier normal noise was the most likely source of problems when using the state-of-the-art surface reconstruction techniques, such as [2].

In this paper, we consider normal and surface reconstructions from unorganized 3D points. The input of the system is a point cloud without normals, typically the registered laser scans of an artistic sculpture (see Section 7). The output is an implicit surface. The quality of the implicit surface is not assessed directly, but through the triangle mesh extracted by the marching cubes algorithm [5]. This polygonization of the implicit output is customary in the research literature, especially when the main application domain is visualization, which requires triangle meshes.

The main goal of this paper is to propose an ensemble framework that improves the robustness of deterministic surface and normal reconstruction algorithms, by putting them in a probabilistic setting. The ensemble framework consists

* Corresponding author. Tel.: +82 54 279 2245.
E-mail addresses: mincheol@postech.ac.kr (M. Yoon), jin@postech.ac.kr (Y. Lee), leesy@postech.ac.kr (S. Lee), ioannis.ivrissimtzis@durham.ac.uk (I. Ivrissimtzis), hpseidel@mpi-sb.mpg.de (H.-P. Seidel).

0010-4485/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2007.02.008

of repetitive random subsampling of the input data and the averaging of the different outputs generated from these randomly sampled subsets to create a combined model. The latter is expected to have higher quality than any of the individual reconstructions that produced it.

This paper expands the idea of surface ensemble described in [6], and also applies the ensemble technique to normal estimation. In addition, we demonstrate that the combination of the normal and surface ensembles can successfully handle noisy point sets, better than the sole application of each ensemble. The main contributions can be summarized as follows;

- We describe a general ensemble framework suitable for surface reconstruction and normal estimation from a noisy point set.
- We study theoretically and experimentally the influence of different averaging rules on the performance of the ensemble.
- With an extensive validation experiment, we study how the error of the ensemble reconstruction is affected by the
  . number of ensemble members
  . sampling rate
  . amount of noise in the input data.

2. Preliminaries

2.1. Ensembles

The ensemble technique is one of the central themes of statistical learning. In a general setting, a probabilistic algorithm, running many times on the same input data, generates several different outputs which are then combined into a single model. To optimize this process, it is important to use a robust averaging formula for combining the members of the ensemble.

The tools and the methodology for the study of the averaging rules depend heavily on the categorization of the algorithm as *supervised* or *unsupervised*. In the case of a supervised algorithm, we have some knowledge of the properties of the outputs. For example, we may know the error of each output. This knowledge can be used to find a combined output which will provably converge to zero error under mild conditions [7,8].

In an unsupervised algorithm, which is the case of this paper, we do not have any knowledge of the properties of the outputs. Thus, mean averaging, or a majority vote in the discrete case, can be the only available option from a theoretical point of view. Nevertheless, we may still be able to devise a more sophisticated averaging rule that will be more robust in practice.

The main problem with mean averaging is the unduly large influence of any possible outliers in the data. The simplest alternative for solving this problem is median averaging. It is widely used in image processing in the form of median filtering [9] and it has been proved to cope very well with outliers. However, from our experiments with ensemble surface reconstructions, we found that median averaging does not fully exploit all the information about the surface which is accumulated by the multiple reconstructions. Instead, we found that the mean average of several values around the median gives better results. We discuss the ensemble averaging rules in detail in Sections 4.2 and 5.2.

In [10], the ensemble technique has been applied to the problem of surface reconstruction, showing a considerable improvement over the corresponding single reconstruction algorithm [11]. However, in [10], the ensemble technique was used over a naturally probabilistic algorithm, while here we impose a probabilistic setting on a deterministic algorithm and show that this improves its robustness.

An algorithm similar to our implementation of the ensemble technique is the probabilistic Random Sample Consensus, or RANSAC [12]. RANSAC starts with a random subset of the input data, which is used to reconstruct a model. The hypothesis is that the sample contains only inlier data and thus it produces a good model. This hypothesis is tested on the rest of the data that were not used in the reconstruction. In particular, we count how many of these other data are also classified by the model as inliers, in the sense that the error is smaller than a given tolerance. If the number of data classified as inliers is above a certain threshold, we assume that our hypothesis has been verified, in which case the final model is recreated using all the inliers rather than the initial sample only. If the hypothesis is not verified, then we start all over again with another random subset. If after a certain number of iterations we have not found a satisfactory model we either consider that the algorithm failed, or we choose the best model we have found up to that time.

Several attractive features of RANSAC, such as simplicity and robustness, make it a very popular choice for modeling data sets which contain outliers. However, we believe that is not suitable for surface reconstruction from range scans. Indeed, RANSAC is most powerful when the test models are reconstructed by very few data, so that there is a high probability they are all inliers and the hypothesis is correct. Instead, in surface reconstruction from range scans, we deal with surfaces of such complexity that we usually need several thousands of input points to create a test model of reasonable detail; thus the probability of avoiding sampling outliers in the initial seed is very small.

2.2. Normal estimation techniques

The estimation of normals for an unorganized point cloud is usually done in two steps. The first step is the estimation of a tangent plane at each point, which will give the direction of the normal at that point. The second step is to determine a consistent orientation for the tangent planes at all points.

Regarding the estimation of the tangent plane, Hoppe et al. [1] use principal component analysis on the k-neighbors of a point. As a result, the estimated tangent plane is the minimum least square fitting of the k-neighbors. Its normal is the eigenvector corresponding to the smallest eigenvalue of the covariance matrix of the k-neighbors. Pauly et al. [13] improve on this by minimizing a weighted least square distance, where the weights are given by a sigmoidal function. Gopi et al. [14] use singular value decomposition to minimize the

dot product between the estimated normal and the vectors from the point to its k-neighbors. Hu et al. [15] proposed a bilateral normal reconstruction algorithm which combines estimations obtained at different sampling rates. Mitra et al. [16] use the least square fitting of the points inside a sphere of a given radius. Dey et al. [17] compute normals pointing to the center of the largest Delaunay ball incident to that point. Techniques for normal smoothing [18] are also related to the problem of normal estimation.

The second step in normal reconstruction is the consistent orientation of the tangent planes. It has attracted relatively little research interest and the pioneering work of Hoppe et al. [1] is still widely used as the state-of-the-art. In [1], they start from an arbitrarily oriented tangent plane and propagate its orientation to its neighbors, guided by a minimum spanning tree. The edge weights of the tree are the angles between the tangent planes.

This second step looks much simpler as it only adds a binary attribute to the non-oriented tangent plane. Moreover, given a good point sample from a smooth orientable surface, one would expect very few inconsistencies. Nevertheless, by wrongly orienting an exact tangent plane, we obtain a normal vector opposite to the correct. That is, the wrong orientation of the exact tangent plane produces the highest possible error, which means that at the second step of the normal estimation, outlier normal noise might be introduced.

In this paper, we use the technique of Hoppe et al. for the experiments of normal ensemble. We demonstrate that the ensemble framework can successfully reduce the noise and outliers in the normals estimated from a noisy 3D point set using the technique.

2.3. Surface reconstruction techniques

A large category of the proposed surface reconstruction algorithms can directly process points without normals. α-shapes are used in [19], polynomial patches are fitted in an iterative region growing process in [20], while B-spline patches are fitted with detail displacement vectors in [21]. In [22–25], the surface is reconstructed from the Delaunay tetrahedrization of the point set.

Another category of algorithms requires points with normals as input. In this case, an implicit function $f : \mathbf{R}^3 \rightarrow \mathbf{R}$ is fitted to the input data and then the surface is extracted as the zero level set of f. The current state-of-the-art in this approach includes a radial basis function based technique [26] and the MPU implicits [2] which uses quadratics.

In reverse engineering, surface reconstruction techniques are tailored for inputs representing engineering objects. In [27], the faces of the mesh are obtained from the connectivity graph of a hierarchical space decomposition of the initial data. In other approaches, the output of the algorithms is a CAD object. In [28], Coons patches or tensor product surfaces are fitted to rectangular regions of the data, defined by boundary NURBS curves. In [29], after a direct segmentation of the point cloud, a framework is proposed for fitting simple analytical and swept surfaces to each segment and blending them. In [30], the framework is extended so that the reconstruction satisfies a given set of constraints between the parameters of different patches.

Surface reconstruction algorithms have been incorporated in several modeling software packages. Surveys of such commercial packages, together with reviews of the related techniques and the typical applications, can be found in [31,32].

In this paper, for surface ensemble experiments, we use the MPU implicits [2], which is suitable for reconstructing a surface from a large point set equipped with normals. The algorithm divides a bounded domain into cells, creating an octree-based partition of the space, and approximates the local shape of the points in each cell with a piecewise quadratic function. If the local approximation for a cell is not accurate enough, the cell is subdivided and the approximation process is repeated until a user-defined accuracy is achieved. Finally, all the local approximations are blended together using local weights, giving a global approximation of the whole input point set.

3. General ensemble framework

The ensemble method is a powerful statistical tool which has found a wide range of applications as it can effectively handle noise and outliers. In a general setting, a probabilistic algorithm runs several times creating different models of the same input data. The main observation is that in different reconstructions the error may be concentrated in different parts of the model. Thus, when we combine the outputs, the high error at one part of a model can be compensated by the low errors at the same part of the other models. This way, we can reduce the error by simply averaging through several outputs and, as a result, obtain a model that will be better than any of the models that were averaged to create it.

The majority of the existing algorithms for normal and surface reconstruction are deterministic and thus, the ensemble method cannot be directly applied to them. However, random subsampling can put these deterministic algorithms into a probabilistic setting and then, the ensemble technique can be used to increase the robustness of the reconstructions. In such a framework, we first create several random subsets of the initial data and process them separately with a deterministic algorithm which is treated as a black box. Then, the several outputs of the deterministic algorithm are combined with a robust averaging rule to produce the final model. Fig. 1 describes the general ensemble framework for surface and normal reconstructions.

The data of the initial input set, depicted on the left of Fig. 1, should be of a type suitable for processing from the deterministic algorithm. In the case of surface reconstruction, that means an unorganized point cloud if the algorithm does not require normal information and points with normals if it does require such information. Regarding the optimization of the ensemble, there are two main issues we have to deal with. The first is the random sampling strategy, where main concerns are the type of random sampling and the number and sizes of the generated subsets. We study this problem from both theoretical and experimental points of view in Section 6. The second issue is the averaging rule for combining the members

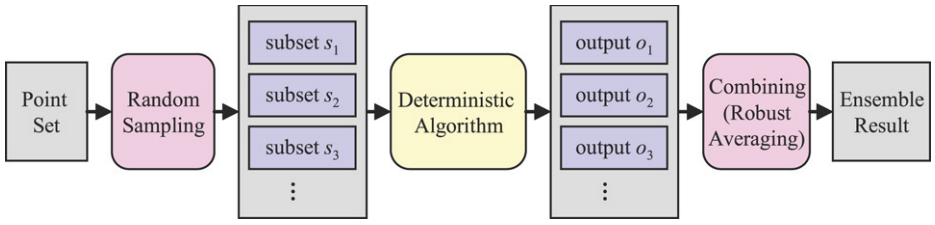


Fig. 1. General ensemble framework for a deterministic algorithm.

of the ensemble. As the relevant algorithmic details differ, these two issues will be discussed separately for normal and surface ensembles, in Sections 4 and 5, respectively.

Finally, regarding the deterministic algorithm that is a part of the framework, in this paper, we choose the method proposed by Hoppe et al. [1] for normal reconstruction and the Multi-level Partition of Unity (MPU) implicits [2] for surface reconstruction. The former is a classic algorithm which is still widely used (e.g., [33]), while the latter is one of the fastest and most up-to-date techniques available.

4. Surface ensemble

The use of ensembles for surface reconstruction was proposed in [6]. The input of the algorithm is a point cloud P with normals and the output is a surface S. The pseudocode of the algorithm is

Surface Reconstruction Ensemble
Input: Point cloud P with normals.
Output: Surface S.

1. Generate m random subsets of P. These subsets may overlap.
2. Process the m subsets separately with a deterministic surface reconstruction algorithm.
3. Use an averaging method to combine the reconstructions obtained in Step 2 into a single surface S.

4.1. Surface ensemble generation

For simplicity, the m subsets of P are generated independently. The sampling is uniformly random without repetition, meaning that a subset cannot have multiple copies of the same point and that all the points have the same probability to be in a subset. The independence of the subsets means that a point may belong to any number of subsets, from zero to m. In the case of surface reconstruction, this is not a problem because each ensemble member gives one function value to each point of the space, regardless of the distribution of the randomly sampled points. In contrast, as we will see in Section 5.1, a slightly more complicated procedure of random sampling is required for a normal ensemble, in order to guarantee an adequate number of normal estimates at every point.

After obtaining the random subsets, we use the MPU algorithm to generate an implicit surface representation for each subset. In the experiments, we used the MPU implementation available on the internet.

4.2. Surface averaging

As a result of the ensemble generation in Section 4.1, we obtain a set of m functions

$$f_j : \mathbf{R}^3 \to \mathbf{R}, \quad j = 1, \ldots, m. \tag{1}$$

The zero level set of each function f_j defines a surface S_j. However, the ensemble algorithm for implicits bypasses the explicit reconstruction of the surfaces S_j and instead computes the average f of the functions f_j. The final ensemble surface S is then obtained as the zero level set of f.

The simplest way to define f at a point $\mathbf{x}$ is to take the mean average

$$f(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^{m} f_j(\mathbf{x}). \tag{2}$$

In some cases, this simple average may be satisfactory. However, as it is generally the case with mean averaging, the influence of few outlier values may increase the error significantly. The robustness of the method can be improved by removing probable outliers from the set of functions we average. Without loss of generality, assume that at a given point $\mathbf{x}$, we have

$$f_1(\mathbf{x}) \le f_2(\mathbf{x}) \le \cdots \le f_m(\mathbf{x}). \tag{3}$$

A more robust averaging function is given by

$$f(\mathbf{x}) = \frac{1}{m - 2r} \sum_{j=r+1}^{m-r} f_j(\mathbf{x}). \tag{4}$$

In other words, we compute the mean average of the functions f_j at $\mathbf{x}$ after filtering out the $2r$ values that are furthest away from the median, that is, the r smallest and the r largest values. Experimentally, we found that a value of $r = \lfloor m/4 \rfloor$ works nicely in practice and as one would expect, the average in Eq. (4) is more efficient in handling outlier noise.

Notice that the function f in Eq. (4) is continuous. Indeed, because of the continuity of the f_j's, any possible discontinuity of f could only appear when the ordering of the f_j's given in Eq. (3) changes. However, as the f_j's are continuous themselves, their ordering can only change at points where they have the same value. The latter means that f is also continuous.

Finally, at the last step of the surface ensemble algorithm, we evaluate the function f at the nodes of a regular grid and then use the marching cubes algorithm [5] to extract a triangle mesh approximating the zero level set of f.

5. Normal ensemble

The input of the ensemble for normal reconstruction is a set P of points without normals. We first generate random subsets of P and use them to compute the members of the normal ensemble. We then combine the members into a single normal reconstruction. The process is summarized by the following pseudocode;

Normal Reconstruction Ensemble
Input: Unorganized point cloud P.
Output: Point cloud with normals P_N.

1. Create several overlapping random subsets of P.
2. Process each subset separately estimating normals for its points.
3. For each point of P, estimate a single normal by combining all the normals estimated for this point at Step 2.

5.1. Normal ensemble generation

First, we subsample P to create the subsets $P_i, i = 1, \ldots, k$. Among different possible ways to perform the subsampling, we choose the simplest solution, as long as the simplicity does not compromise the quality of the results.

In our experiments, the sets $P_i, i = 1, \ldots, k$, have the same number of points,

$$|P_1| = |P_2| = \cdots = |P_k|, \tag{5}$$

where $|P|$ denotes the number of points in P. The sampling rate $d = |P_i|/|P|$ is the density of the subsampling. Obviously, the value of d affects the quality of the normal estimation. For example, if $d = 1$, the algorithm becomes deterministic and no improvement over the single reconstruction is possible. On the other hand, a very small d may again yield bad estimates because points that are far away from each other may become closest neighbors in P_i.

The choice of k, i.e., the number of members of the ensemble, is a trade-off between speed and quality. The normal estimation algorithm will run k times on a set of size $|P_i|$ and thus, a large k will slow the process down. On the other hand, there will be about $m = k \cdot d$ different normal estimations for each point of P and the higher this number, the more accurate the estimates.

During the sampling process, it is important that each point is sampled several times, because the goal is to obtain good normal estimations for all the points of P. However, if the random sampling algorithm does not explicitly control repetitions, there is a possibility that some points of P will be sampled very few times, or never. To solve this problem, we create the subsets P_i by sampling points from P without repetition and when we exhaust all the points of P, we start the sampling without repetition all over again.

After constructing the random subsets P_i, we use the algorithm described in [1] to obtain normal estimations. In the experiments, we adopted the implementation of the algorithm available on the internet and used the default parameters in most cases.

5.2. Normal averaging

In the previous step, for each point of P, we obtained several different normal estimates. Next, we have to combine them into a single estimate with a smaller expected error. Similarly to Eq. (2), we can use the normalized mean average

$$\mathbf{n} = \sum_{i=1}^{m}(\mathbf{n}_i) \Bigg/ \left|\sum_{i=1}^{m}(\mathbf{n}_i)\right|. \tag{6}$$

To improve on the results obtained with mean averaging, we have to find a robust normal averaging analogous to Eq. (4). One possible way is to start with Eq. (6) and then find the estimates that deviate most from $\mathbf{n}$. That is, for each $\mathbf{n}_j$, we compute the angle θ_j between $\mathbf{n}$ and $\mathbf{n}_j$. Without loss of generality, assume that

$$\theta_1 \le \theta_2 \le \cdots \le \theta_m. \tag{7}$$

We can exclude the r estimates with the larger deviations from $\mathbf{n}$ and average the rest of them. That is, we finally obtain

$$\mathbf{n}_f = \mathrm{Aver}(\mathbf{n}_1, \ldots, \mathbf{n}_{m-r}), \tag{8}$$

where Aver() is the average of normals proposed in [34].

The averaging method proposed in [34] starts with Eq. (6) for the estimated average normal and iteratively improves the estimate until convergence. The iterative process involves the mapping and averaging of the normals on the tangent plane of the Gauss sphere at the current estimated average normal. The average normal obtained as the limit of this iteration has superior theoretical properties to the simple average from Eq. (6).

We experimented with $r = \lfloor m/2 \rfloor$, excluding about a half of the estimates. It is highly unlikely that outlier noise, e.g., wrongly oriented normals, covers more than a half of the samples. However, we still witnessed some problems with the quality of the normal reconstruction, which we attributed to the inaccurate ordering of the normals caused by the low accuracy of the initial mean average in Eq. (6).

To improve the accuracy of the normal estimation, we notice that Eq. (7) orders the normals according to their total variances

$$\mathrm{Var}(\mathbf{n}_i) = \frac{1}{m}\sum_{j=1}^{m}(1 - \mathbf{n}_i \cdot \mathbf{n}_j)^2. \tag{9}$$

Then, instead of directly using Eq. (9) to order the normals, we compute the sum of total variances

$$\mathrm{Var}(N) = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}(1 - \mathbf{n}_i \cdot \mathbf{n}_j)^2 \tag{10}$$

and use a constant multiple of it as the threshold for outlier detection. That is, if $\mathrm{Var}(\mathbf{n}_i)$ is larger than $c \cdot \mathrm{Var}(N)$, we consider $\mathbf{n}_i$ to be an outlier. In this approach, at different points, a different number of outliers may be removed. We consider this as an important improvement, because we do not expect the outliers to be evenly spread over P. In our experiments, we found that a value for c between 1.1 and 1.3 achieves the best

Table 1
Five point sets sampled from the tangle cube with different amounts of noise and outliers: d is the average distance from a point to its nearest neighbor in the point set without noise

Model	Noise	Outlier	Model	Noise	Outlier
Tangle A	0.5d	None	Tangle D	2.0d	None
Tangle B	1.0d	None	Tangle E	1.5d	8.0d
Tangle C	1.5d	None			

results. Notice that the value $c = 1$ means that a normal is outlier when its total variance is larger than the average of the total variances.

The same approach to outlier removal can also be applied to surface ensembles. However, in that case, we also have to consider the continuity of the final surface. Thus, the total variance approach becomes too complex on surfaces, which is why we did not use it in this paper.

6. Validation test with an implicit model

6.1. Implicit model and test point set sampling

To validate the ensemble algorithms, we tested them on point sets sampled from a smooth surface with a known analytic formula. In our experiments we used the *tangle cube*, given by the equation

$$x^4 - 5x^2 + y^4 - 5y^2 + z^4 - 5z^2 + 11.8 = 0, \tag{11}$$

which has a fairly complex shape and non-trivial topology (see Fig. 7(a)).

In the set-up of the validation experiment, we followed a strategy similar to that used in [25]. To create a test set, we first sampled a large number of points from the tangle cube, here $N \sim 250$ K points. Notice that these N points do not have any added noise and so they can be used for measuring the error. Next, we sampled N more points, to which we added noise by randomly perturbing their positions. For each of the data sets we created, the length of the displacement we added to a noisy point is a random number in the range between zero and the average distance from a point to its nearest neighbor times a constant. For our experiments, we created data sets with this constant ranging from 0.5 to 2. Finally, in one of the sets, we also added extra $0.15 \cdot N$ of outliers, which are points displaced up to 8 times the average distance between points.

Table 1 summarizes the information on the five test sets we used in our validation experiment.

6.2. Error measurement

To evaluate the normal reconstructions, we measured the error on the points without added noise using the analytical formula to compute the exact normal. We computed the root mean square error

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_j (1 - \mathbf{n}_j^e \cdot \mathbf{n}_j)^2} \tag{12}$$

where $\mathbf{n}_j^e$ is the estimated normal and $\mathbf{n}_j$ is the exact normal.

To evaluate the surface reconstructions, we measured the distances of $10 \cdot N$ points randomly sampled from the original tangle cube to the reconstructed triangle meshes. The distances were computed with the Metro tool [35], which was adjusted to measure distances from points to a mesh.

6.3. Influence of number of ensemble members and sampling rate

The first issue we used the validation test to clarify is how the error of the reconstruction is affected by the number of ensemble members, the sampling rate, and the amount of noise in the data set. We experimented with various numbers of ensemble members and sampling rates using the four models, Tangle A–D, which have an increasing amount of noise.

Fig. 2 and Fig. 3 summarize the results of this validation test for normal and surface ensembles, respectively. In the experiments, we used the total variance of Eq. (10) and robust average of Eq. (4) for the averaging methods of normal and surface ensembles, respectively.

By inspection of these figures, we notice that in all the cases, that is, regardless of the sampling rate and the noise in the data set, the error decreases as the number of the ensemble members increases. The theoretical justification of this result, which is also the central idea behind the ensemble technique, is that averaging decreases the variance component of the error. Indeed, if we consider the error of each ensemble member as a random variable, the error has a bias/variance decomposition [9],

$$\mathrm{RMS}^2 = \mathrm{bias}^2 + \mathrm{variance}. \tag{13}$$

Assuming that these random variables are independent and follow identical normal distributions, their average is a new random variable with the same bias but lower variance. As the number of ensemble members increases, the variance component in Eq. (13) tends to zero while the bias stays constant. In particular, that means that unless the bias is zero, which is highly unlikely in the kind of applications we study, the ensemble technique can only reduce but not eliminate the error.

The speed with which the variance decreases as the size of the ensemble increases depends on the averaging rule we use. From a theoretical point of view, the mean average is the optimal option in this respect, since it has the lowest variance between all the averaging rules. In particular, if the variance of the expected error from a single estimate is σ^2, then the variance of the mean average of m estimates is σ^2/m. For a comparison, for large m, the variance of the median tends to $\pi\sigma^2/2m$, meaning that using the median requires about a half more of estimates for the same reduction in the variance [36]. In terms of efficiency in reducing the variance, the averaging rules in Eqs. (4) and (8) and the one based on the total variance are between the mean and the median. However, as we mentioned above, because the mean average cannot handle outliers efficiently, it will most likely have a higher bias.

Fig. 4 shows the timings of the normal and surface ensemble algorithms for the model Tangle C. The timing data were

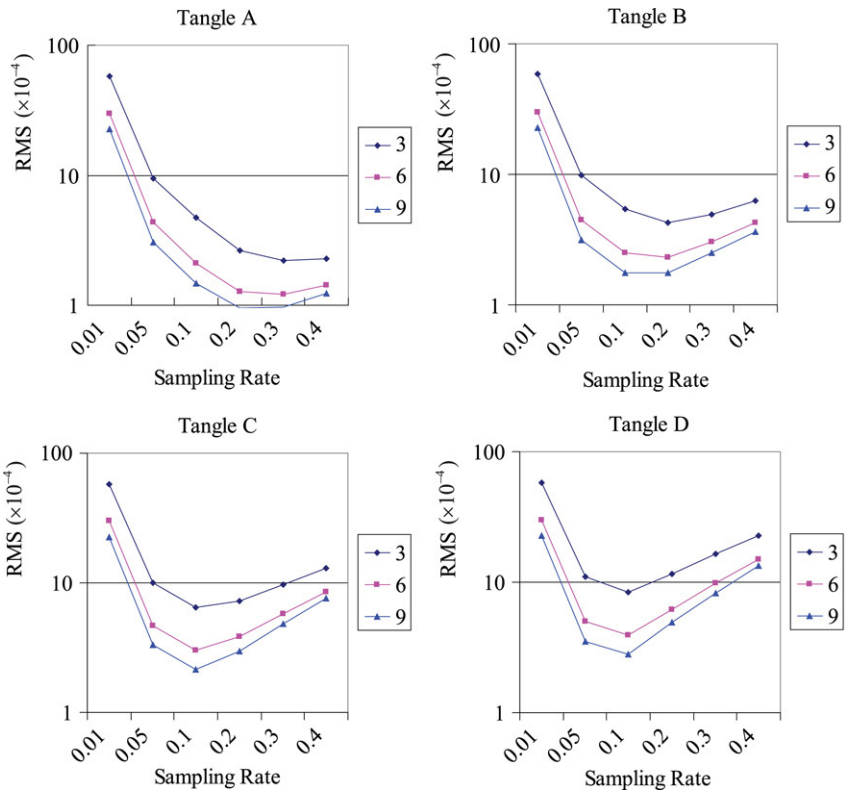


Fig. 2. RMS errors ($\times 10^{-4}$) of normal ensembles in various settings: three different curves in each chart represent results from different numbers of ensemble members.

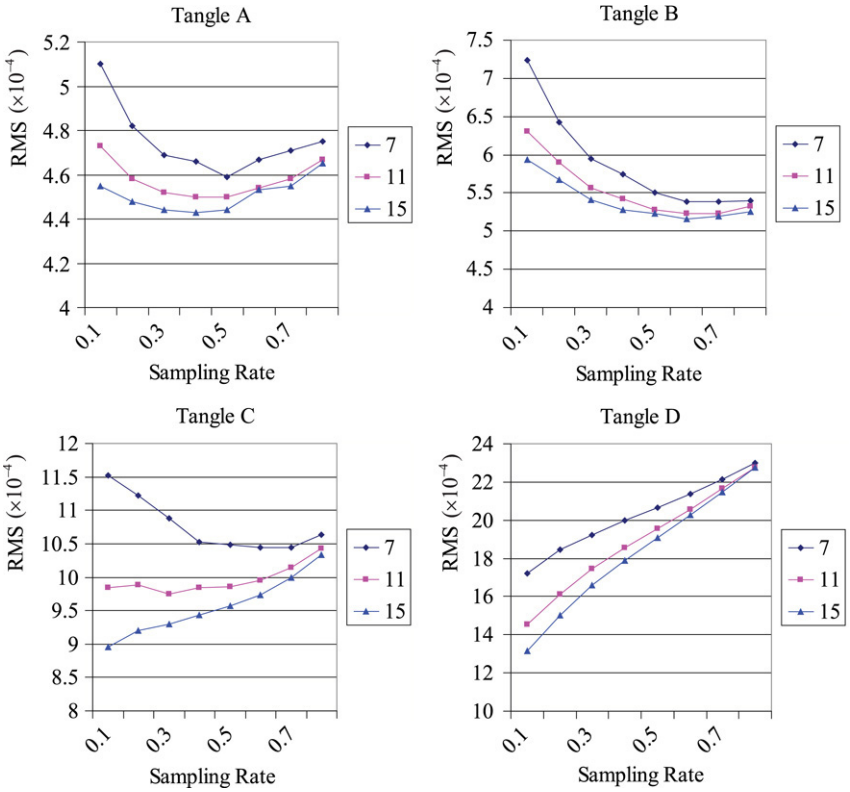


Fig. 3. RMS errors ($\times 10^{-4}$) of surface ensembles in various settings: three different curves in each chart represent results from different numbers of ensemble members.

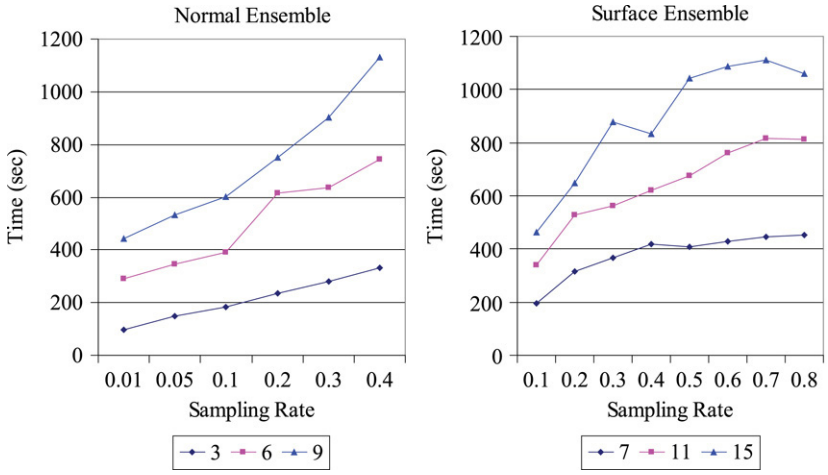


Fig. 4. Timing results for normal and surface ensembles measured in seconds: three different curves in each chart represent different numbers of ensemble members.

measured in seconds on a PC running Windows with a Pentium 4 630 processor with 2 GB memory.

Next we study the optimal sampling rate for a given data set and its relation to the amount of noise in the data.

Regarding normal ensembles, by inspecting the curves of Fig. 2, we notice that an optimal sampling rate exists. Indeed, when the sampling rate is very low, the members of the ensemble are very sparse and so, even when they are combined, still they cannot capture the detailed normal information of the underlying surface. Then, as the sampling rate increases, the error decreases until an optimal sampling rate is reached. After that point the error again increases with the sampling rate as the ensemble cannot cope with the higher normal noise of the denser sample. In the literature this phenomenon is called *overfitting* and was studied in [37] for surface reconstruction.

By comparing the optimal sampling rates of different models, we observe that as the level of noise increases, the optimal sampling rate decreases. That was expected because as the noise of the sample increases, only the coarser level information of the underlying surface is preserved and thus, only a reconstruction at a coarse level is possible. However, while a low density sampling avoids the reconstruction of normal noise and improves the performance of the normal ensemble, it also fails to preserve the features of the surface below a certain level of detail. That means that our choice of optimal sampling rate is a trade-off between removing the noise and preserving the features. In particular, if the noise has corrupted the finer features of the surface and made them unrecoverable, we have to choose a lower sampling rate, while if there is still enough information to reconstruct the features of the surface, then we can choose a higher sampling rate.

By inspecting Fig. 3, we find that similar results about the sampling rate hold for surface ensembles. However, it is also obvious that the surface ensembles are much less sensitive to changes of the sampling rate. We see this as a reflection of the fact that the normal information is much more sensitive to noise, as small perturbations in the positions of points can drastically alter the direction of their normals.

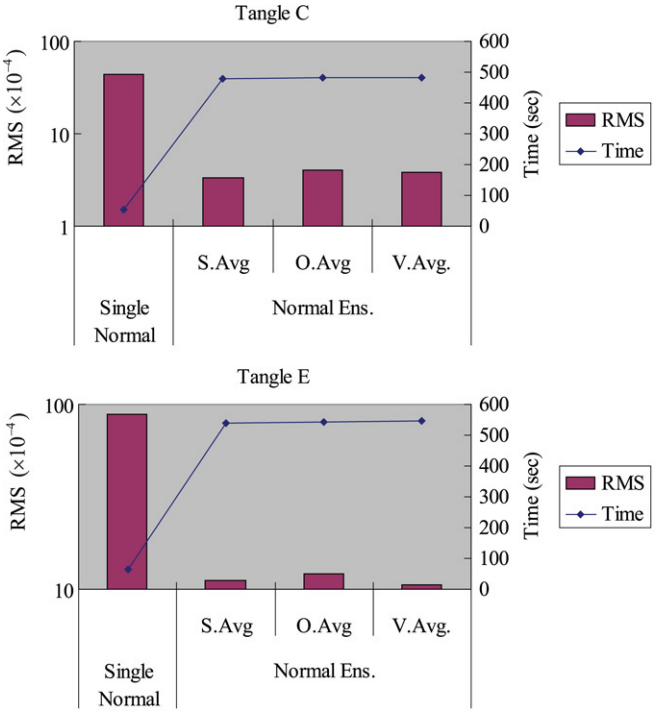


Fig. 5. Normal ensembles: "S. Avg." is the simple average of Eq. (6). "O. Avg." is the average with normal ordering of Eq. (8). "V. Avg." is the average with the total variance of Eq. (10).

6.4. Effectiveness of ensembles

In this section, we study the efficiency of the averaging rule. We also study the effectiveness of the ensemble techniques compared to the corresponding single deterministic algorithms. In the experiments, as a representative setting for ensemble reconstruction, we used the Tangle C data set with $m = 6$ estimates for the normal ensemble and 11 for surface ensemble, where the sampling rates are 0.2 and 0.3 for normal and surface ensembles, respectively. As we are particularly interested in the handling of outliers, we also used the Tangle E data set.

Fig. 5 and Fig. 6 show the results for surface and normal ensembles, respectively. As we expected, the ensemble is

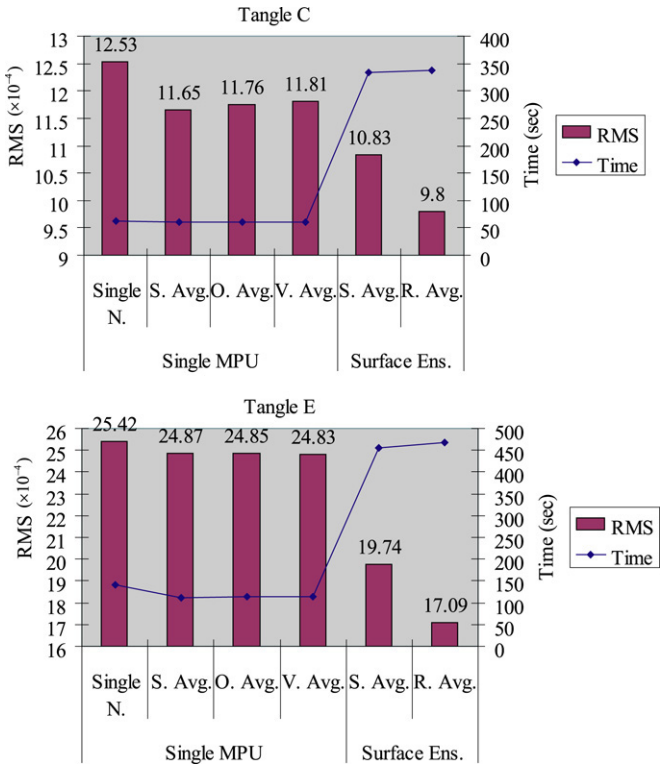


Fig. 6. Surface reconstruction: For surface ensembles, "S. Avg." is the simple average of Eq. (2). "R. Avg." is the robust average of Eq. (4). The normals obtained by the normal ensemble with "V. Avg." were used for surface ensembles.

an effective technique and lowers the error of the single reconstruction in all cases. Regarding the averaging rule, we can verify that the proposed averages in Sections 4 and 5 are more effective than mean averaging in the presence of outliers, while if there are no outliers, the mean averaging may outperform them. However, as we discussed in Section 6.3, mean averaging reduces the variance component of the error faster than the other averaging rules but usually has higher bias. Thus, on very large ensembles where the variance is practically zero and the error is equal to the bias, we expect the proposed rules to slightly outperform mean averaging.

Figs. 7 and 8 show comparisons between single and ensemble reconstructions of normals and surfaces. The figures not only verify the numerical results obtained by the validation tests, but also show that the use of the ensemble technique and the robust averaging rules can have an impact on the visual quality of the reconstructions.

7. Experimental results from real data

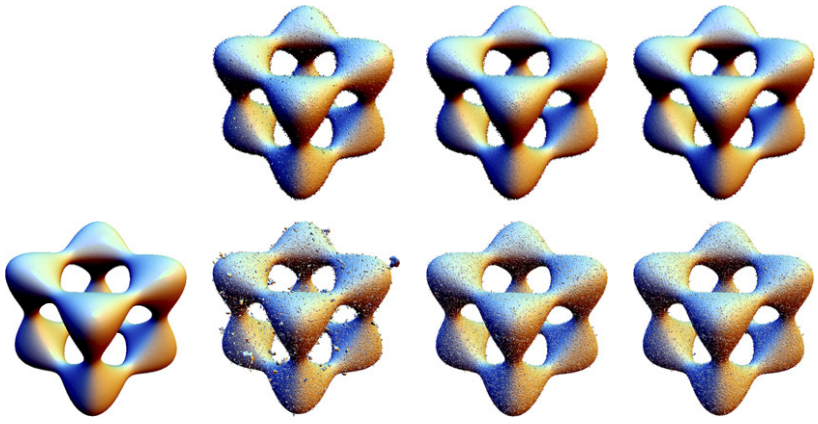
In this section, we present experimental results from four well-known point sets; the bierkrug and karaffe models from the ViHAP3D project and the Buddha and armadillo models from the Stanford 3D scanning repository. The aim is to test the behavior of the ensemble algorithms on real data with the usual amount of noise that can be found in point sets from range scans. Throughout this section, we always use Eq. (10) for normal averaging and the robust averaging with Eq. (4) for surface ensembles.

7.1. Normal ensemble

Fig. 9 shows normal estimations with and without ensemble on a point set from the bierkrug model. In this experiment the sampling rate was $d = 0.2$ and the number of ensemble members $m = 6$.

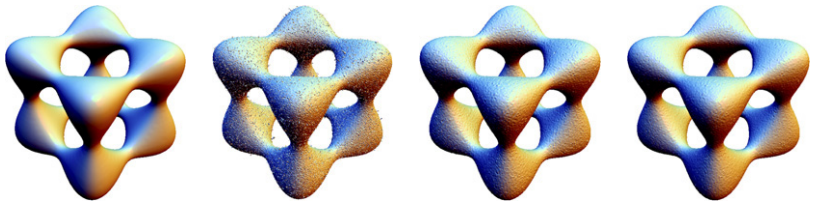
Fig. 9(a) and (b) visualize the point sets with the estimated normals and we can notice that the drawing inscribed on the bierkrug is shown in more detail on the point set with single normal estimation. However, this does not mean that the single estimated normals are more accurate. In fact, in this area of the bierkrug, we have overlapping range images and most of the single estimated normal information, even though visually pronounced, is of low quality. This is confirmed by Fig. 9(c) and (d), where in both cases the single MPU reconstructions smoothed out the inscription.

In the MPU reconstruction with single normal estimation shown in Fig. 9(c), we can see some blobby artifacts in the base



(a) Original. (b) Single normal. (c) Simple average. (d) Eq. (10) average.

Fig. 7. Tangle cube: Comparison of different normal estimates and the surfaces obtained from them. At the top of (b)–(d), each point is rendered with illumination determined by the normal vectors. At the bottom of (b)–(d), the surfaces are reconstructed using a single MPU.



(a) Original. (b) Single MPU. (c) Simple average. (d) Robust average.

Fig. 8. Tangle cube: Comparison of surfaces obtained from single MPU and surface ensembles.

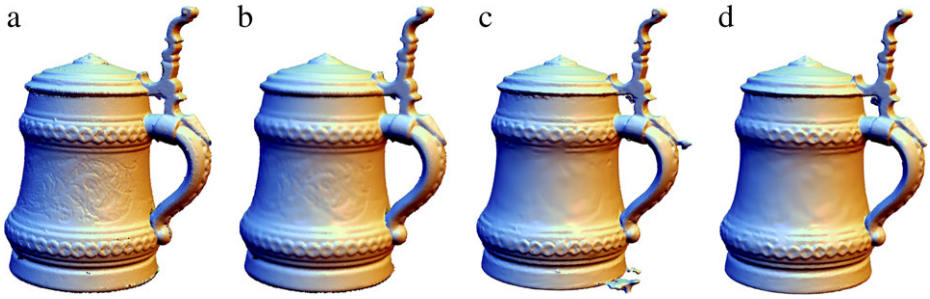


Fig. 9. Bierkrug: (a) Normals obtained by single estimation. (b) Normals obtained by ensemble. (c) and (d) Single MPU reconstructions of (a) and (b), respectively.

of the bierkrug and at the end of the handle. This is the result of outlier normals, created by inconsistencies in the orientation of the tangent planes as discussed in Section 2.2. We notice that it is very difficult to detect normal outliers by visual inspection of the point cloud, as no such problems are apparent in Fig. 9(a). That means that human intervention can also be ineffective in removing normal outliers. In contrast, the visual detection of spatial outliers is easier and human intervention is an efficient way of removing them [4].

7.2. Surface ensemble

Fig. 10 shows the input data and reconstructions of the Buddha model. The sampling rate was $d = 0.1$ and the number of ensemble members $m = 11$. The normal set was acquired by the scanner and not estimated from the unorganized point cloud. We notice that the MPU ensemble improves on the single MPU reconstruction. Note that the reconstruction of the Buddha model in the original MPU paper [2] used a different point set.

Fig. 11 shows the eleven members of the Buddha MPU ensemble. Clearly, the ensemble reconstruction in Fig. 10(c) has higher quality than every single one of its eleven members.

7.3. Combination of normal and surface ensembles

Fig. 12(a), (b), (e), and (f) show close-ups of the two single MPU reconstructions of Fig. 9. Fig. 12(c), (d), (g), and (h) show close-ups of the corresponding surface ensemble reconstructions using single estimated and ensemble normal sets. The combination of normal and surface ensembles clearly outperforms all the other methods. In particular, it is the only method that resolves the artifacts on the base of the bierkrug,

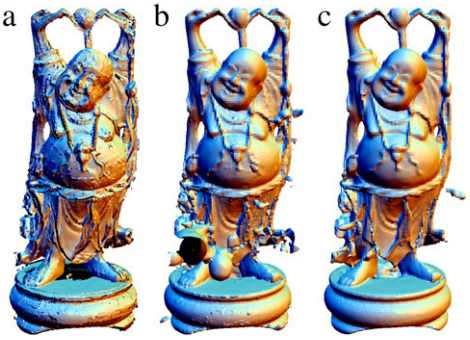


Fig. 10. Buddha: (a) Input point set. (b) Single MPU reconstruction. (c) MPU ensemble reconstruction.

as shown in Fig. 12(d). It is also the one that gives the best reconstruction of the smaller handle, as shown in Fig. 12(h).

Fig. 13 shows the point set and the single and ensemble MPU reconstructions of the armadillo model. The normals are obtained either by single or by ensemble estimation. In the normal ensembles, the sampling rate was $d = 0.1$ and the number of ensemble members $m = 6$. In the surface ensembles we used $d = 0.2$ and $m = 11$. The surface ensembles in Fig. 13(c) and (f) outperform the single reconstructions in Fig. 13(b) and (e). Again, the model in Fig. 13(f), obtained using ensembles for both normal estimation and surface reconstruction, has the highest visual quality. Indeed, the artifacts have been effectively removed, while all the important geometric detail is preserved. Note that the input of this experiment is the original scan data from the Stanford 3D scanning repository and the results should not be compared with reconstructions using processed data. In the MPU reconstructions, we did not use the confidence values provided with the point data.

Fig. 11. Buddha: The eleven members of the surface ensemble for the Buddha in Fig. 10.

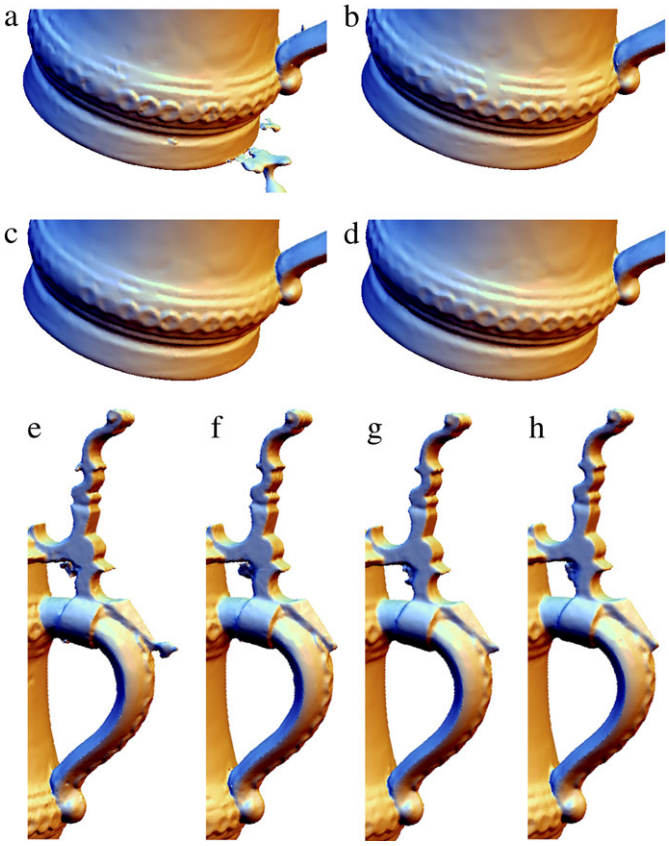


Fig. 12. Bierkrug: (a) and (e) zoom-ins of Fig. 9(c). (b) and (f) zoom-ins of Fig. 9(d). (c) and (g) show the result of the surface ensemble for Fig. 9(a). (d) and (h) show the result of the surface ensemble for Fig. 9(b).

Fig. 14 shows the single and ensemble reconstructions for the karaffe model. Here we have an exception as the surface ensemble with single normal estimation, shown in Fig. 14(c), outperforms the ensemble reconstruction with ensemble normals shown in Fig. 14(d). The reason is that the top of the karaffe was already very sparsely sampled and the subsampled members of the normal ensemble contained even less information about that part of the model. To solve this problem we have to increase the sampling density of the normal ensemble.

Table 2 gives timing statistics and information about the sizes of the models.

8. Discussion and future work

The use of ensembles to enhance the performance of a surface reconstruction algorithm was proposed in a short paper [6]. In this paper, we have extended the result by applying ensembles to normal estimation and providing analysis and validation of the proposed algorithms with extensive experimentation.

On the theoretical side, the main contributions of the paper are in Section 3, giving the description of the general ensemble framework, and in Section 6, with the analysis of the mechanism through which an ensemble reduces the error of a reconstruction.

On the experimental side, we first validated the proposed technique using artificial data and then tested it on natural scan data. In the design of the validation experiment, we first identified three parameters that are crucial for the performance of the ensemble, that is, the number of ensemble members, the sampling rate and the amount of noise in the input data. We treated these parameters as independent, each one being a separate dimension of the experiment. The results, summarized in the graphs of Section 6, verified all the expected relations between these parameters and the error of the reconstruction. In Section 7, experiments with scan data showed that the error reduction by the use of ensembles can have a serious impact on the visual quality of the reconstructed models. That confirms the relevance of the proposed method in graphics, visualization and related applications. The experiments also verified the robustness of the method, as a choice of $m \simeq 10$ and $d \simeq 0.1$ gives a reasonable quality/cost trade-off for the amount of noise expected in a range scan.

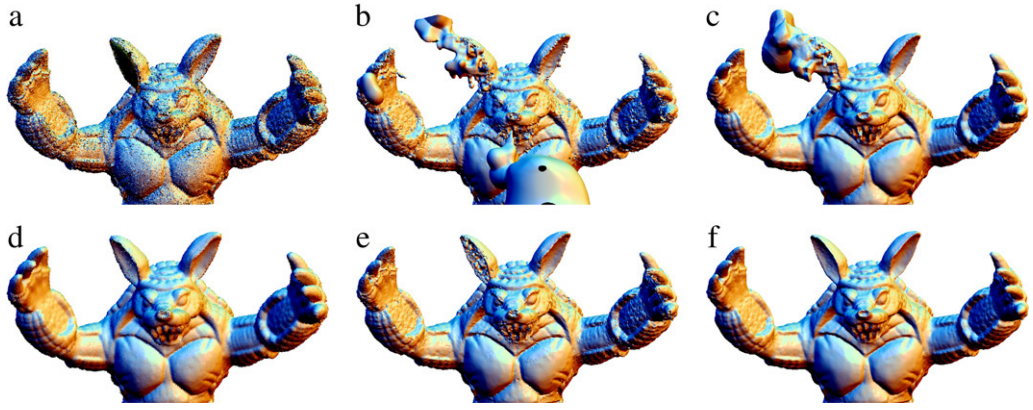


Fig. 13. Armadillo: (a)–(c) Single estimated normals. (d)–(f) Ensemble estimated normals. Left: Points with normals. Center: Single MPU. Right: MPU ensemble.

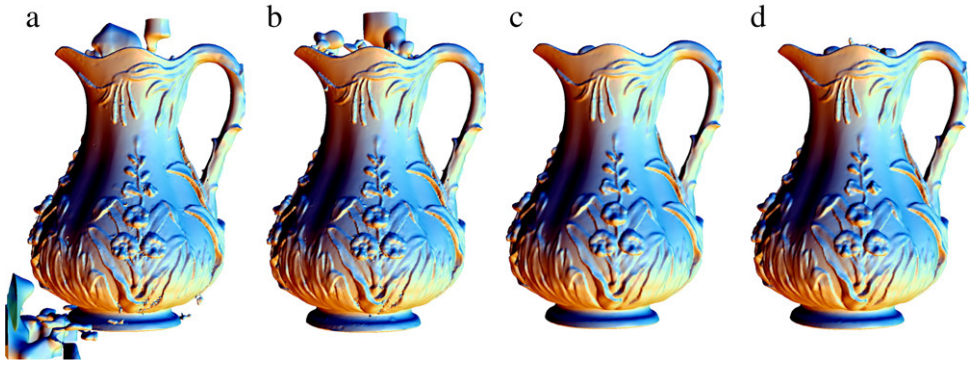


Fig. 14. Karaffe: (a) Single MPU reconstruction with single estimation of normals. (b) Single MPU reconstruction with ensemble normals. (c) and (d) Surface ensembles using the normal sets of (a) and (b), respectively.

Table 2
Timing statistics measured in seconds

Model	# Points	Normal estimation		Surface reconstruction	
		Single normal	Normal ensemble	Single MPU	Surface ensemble
Bierkrug	500,690	63.91	540.57	33.14	170.27
Karaffe	1,001,915	188.36	1502.97	50.59	98.14
Buddha	3,260,401			138.31	198.28
Armadillo	1,394,271	410.07	2102.1	107.89	304.55

As the ensemble algorithms naturally filter out the noise of the data, they have several similarities with smoothing. For example, the size of the local neighborhoods used in most smoothing algorithms is related to the density of the random subsampling of the ensembles. However, we believe that random sampling reflects the probabilistic nature of the noise better, compared to the deterministically defined k-neighbors.

Compared to the deterministic approach, one drawback of the ensembles is the higher computational cost. This can be well justified in the case of surface reconstruction, given that the process is not yet fully automated and the extra spent computational time can save human labor time. In addition, surface reconstruction is usually an one-off process, allowing more computation to obtain a better quality result.

In terms of memory cost, an ensemble technique may seem to need more memory than the corresponding single reconstruction technique because we should keep several outputs for averaging to generate the final output. However, in practice, the dominant memory consumption usually happens when we run the reconstruction technique, which may need additional data structures for processing. By applying the reconstruction technique to the subsets of input in turn and storing the results to files, we can separate the ensemble generation and averaging steps, reducing the memory cost. In the case of MPU, with a more sophisticated implementation, we can exploit the local nature of the MPU functions and perform the ensemble generation and the averaging steps locally and simultaneously. This will further reduce the memory cost, as there will be no need for storing the MPU functions.

We expect that the general framework described in Section 3 can be readily applied to improve the performance of other deterministic surface and normal reconstruction algorithms. It is our future plan to verify that claim. In another direction for future research, we also plan to improve the ensemble algorithms by using any prior knowledge about the amount of noise in the data. For example, the confidence values associated with the point data are estimates of the amount of the expected noise at that part of the scan. Thus, they can be used to fine tune the ensemble parameters.

Acknowledgements

The authors would like to thank Yutaka Ohtake for great help on the implementation of MPU ensembles. The armadillo model is courtesy of the Stanford Computer Graphics Lab. This research was supported in part by the ITRC support program, KOSEF (F01-2005-000-10377-0), and KT (5KI0671901).

References

[1] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. Computer Graphics 1992;71–8.
[2] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H-P. Multi-level partition of unity implicits. ACM Transactions on Graphics 2003;22(3):463–70.
[3] Bernardini F, Rushmeier H. The 3D model acquisition pipeline. Computer Graphics Forum 2002;21(2):149–72.
[4] Weyrich T, Pauly M, Keiser R, Heinzle S, Scandella S, Gross M. Post-processing of scanned 3D surface data. In: Proc. Eurographics symposium on point-based graphics 2004. 2004. p. 85–94.
[5] Lorensen WE, Cline HE. Marching cubes: A high resolution 3D surface construction algorithm. Computer Graphics 1987;163–9.
[6] Lee Y, Yoon M, Lee S, Ivrissimtzis I, Seidel H-P. Ensembles for surface reconstruction. In: Proc. Pacific graphics 2005. 2005. p. 115–7.
[7] Schapire RE. The strength of weak learnability. Machine Learning 1990; 5(2):197–227.
[8] Freund Y, Schapire RE. Experiments with a new boosting algorithm. In: Proc. the 13th international conference on Machine learning. 1996. p. 148–56.
[9] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: Data mining, inference, and prediction. Springer; 2001.
[10] Ivrissimtzis I, Lee Y, Lee S, Jeong W-K, Seidel H-P. Neural mesh ensembles. In: Proc. the 2nd international symposium on 3D date processing, visualization, and transmission. 2004. p. 308–15.
[11] Ivrissimtzis I, Jeong W-K, Lee S, Lee Y, Seidel H-P. Neural meshes: Surface reconstruction with a learning algorithm. Tech. rep. MPI-I-2004-4-005. Saarbrücken: Max-Planck-Institut für Informatik; 2004.
[12] Fischler MA, Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 1981;24(6):381–95.
[13] Pauly M, Keiser R, Kobbelt LP, Gross M. Shape modeling with point-sampled geometry. ACM Transactions on Graphics 2003;22(3):641–50.
[14] Gopi M, Krishnan S, Silva C. Surface reconstruction based on lower dimensional localized Delaunay triangulation. Computer Graphics Forum 2000;19(3):467–78.
[15] Hu G, Xu J, Miao L, Peng Q. Bilateral estimation of vertex normal for point-sampled models. In: Proc. computational science and its applications (ICCSA 2005). 2005. p. 758–68.
[16] Mitra NJ, Nguyen A, Guibas L. Estimating surface normals in noisy point cloud data. Special Issue of International Journal of Computational Geometry and Applications 2004;14(4–5):261–76.
[17] Dey TK, Li G, Sun J. Normal estimation for point clouds : A comparison study for a Voronoi based method. In: Proc. Eurographics symposium on point-based graphics 2005. 2005. p. 39–46.
[18] Jones TR, Durand F, Zwicker M. Normal improvement for point rendering. IEEE Computer Graphics and Applications 2004;24(4):53–6.
[19] Bajaj CL, Bernardini F, Xu G. Automatic reconstruction of surfaces and scalar fields from 3D scans. In: Proc. ACM SIGGRAPH 1995. 1995. p. 109–18.
[20] Sapidis NS, Besl PJ. Direct construction of polynomial surfaces from dense range images through region growing. ACM Transactions on Graphics 1995;14(2):171–200.
[21] Krishnamurthy V, Levoy M. Fitting smooth surfaces to dense polygon meshes. In: Proc. ACM SIGGRAPH 1996. 1996. p. 313–24.
[22] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: Proc. ACM SIGGRAPH 1998. 1998. p. 415–21.
[23] Amenta N, Choi S, Kolluri RK. The power crust, unions of balls, and the medial axis transform. Computational Geometry: Theory and Applications 2001;19(2):127–53.
[24] Dey TK, Goswami S. Tight cocone: A water-tight surface reconstructor. Journal of Computing and Information Science in Engineering 2003;3(4): 302–7.
[25] Kolluri R, Shewchuk JR, O'Brien JF. Spectral surface reconstruction from noisy point clouds. In: Proc. the 2nd Eurographics symposium on geometry processing. 2004. p. 11–21.
[26] Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, et al. Reconstruction and representation of 3D objects with radial basis functions. In: Proc. ACM SIGGRAPH 2001. 2001. p. 67–76.
[27] Azernikov S, Fischer A. Efficient surface reconstruction method for distributed CAD. Computer-Aided Design 2004;36(9):799–808.
[28] Piegl LA, Tiller W. Parametrization for surface fitting in reverse engineering. Computer-Aided Design 2001;33(8):593–603.
[29] Benkö P, Martin RR, Várady T. Algorithms for reverse engineering boundary representation models. Computer-Aided Design 2001;33(11): 839–51.
[30] Benkö P, Kós G, Várady T, Andor L, Martin RR. Constrained fitting in reverse engineering. Computer Aided Geometric Design 2002;19(3): 173–205.
[31] Remondino F. From point cloud to surface: The modeling and visualization problem. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 2003; XXXIV-5/W10:215–230.
[32] Remondino F, El-Hakim S. Image-based 3D modelling: A review. The Photogrammetric Record 2006;21(115):269–91.
[33] Sainz M, Pajarola R, Mercade A, Susin A. A simple approach for point-based object capturing and rendering. IEEE Computer Graphics and Applications 2004;24(4):24–33.
[34] Buss SR, Fillmore JP. Spherical averages and applications to spherical splines and interpolation. ACM Transactions on Graphics 2001;20(2): 95–126.
[35] Cignoni P, Rocchini C, Scopigno R. Metro: Measuring error on simplified surfaces. Computer Graphics Forum 1998;17(2):167–74.
[36] Kenney JF, Keeping ES. Mathematics of statistics. Part II. Revised ed. New York: Van Nostrand; 1951.
[37] Lee Y, Lee S, Ivrissimtzis I, Seidel H-P. Overfitting control for surface reconstruction. In: Proc. the 4th Eurographics symposium on geometry processing. 2006. p. 231–4.