Eurographics Symposium on Point-Based Graphics (2006)
M. Botsch, B. Chen (Editors)

Efficient and Prioritized Point Subsampling for CSRBF Compression

Masaki Kitago [1,2] and M. Gopi [2]

[1] Tokyo Institute of Technology, Japan
[2] University of California, Irvine, U.S.A.

Abstract

We present a novel cost function to prioritize points and subsample a point set based on the dominant geometric features and local sampling density of the model. This cost function is easy to compute and at the same time provides rich feedback in the form of redundancy and non-uniformity in the sampling. We use this cost function to simplify the given point set and thus reduce the CSRBF (Compactly Supported Radial Basis Function) coefficients of the surface fit over this point set. Further compression of CSRBF data set is effected by employing lossy encoding techniques on the geometry of the simplified model, namely the positions and normal vectors, and lossless encoding on the CSRBF coefficients. Results on the quality of subsampling and our compression algorithms are provided. The major advantages of our method include highly efficient subsampling using carefully designed, effective, and easy compute cost function, in addition to a very high PSNR (Peak Signal to Noise Ratio) of our compression technique relative to other known point set subsampling techniques.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – Curve, surface, solid, and object representations

1. Introduction

Due to the advances in 3D data acquisition techniques such as laser range scanners, point set surfaces are increasingly popular. As scanning devices produce higher detailed surfaces that contain huge number of unorganized points, surface reconstruction algorithms need large memory and running time. Clearly, the sampling by these automatic scanners are not dependent on the surface features, and hence the data sets exhibit a large proportion of redundant information. Simplification of such point sets is one of the important preprocessing techniques for making storage, transmission, computation, and display more efficient. Other than these applications, the time consuming processes, such as computation and polygonization, involved in the reconstruction of the surface from its point samples using implicit representations benefit a lot by reduced input point set.

In this paper, we present a novel point cloud simplification method that is suitable for implicit representations. The method uses the curvatures of implicit functions and the distribution of CSRBFs (Compactly Supported Radial Basis Functions) [Wen95, MYR*01] in order to prioritize the points to be removed and simplifies the model progressively.

The main contributions of our paper are as follows.

1. Our point cloud simplification technique is feature-sensitive, has guaranteed sampling density, is easy and efficient to compute, and simple to implement.
2. Specifically, our point decimation scheme does not require refitting of implicit surfaces for every iteration of point removal and re-prioritization.
3. We also propose a technique to compress the CSRBF coefficients of the implicit surface that is fit over the simplified point model. Our coder stores vertices, normals, and CSRBF coefficients without using mesh connectivity unlike other conventional techniques.
4. By virtue of our choice of CSRBF as the preferred implicit representation, as against other multi-level approaches, the compressed implicit functions can be directly used in the applications without refitting the surfaces. Further, the amount of storage required to store these functions is just proportional to the number of points in the data set.

2. Related work

There are quite a few papers on point set subsampling and resampling for rendering and representation. Some of the techniques provide guarantees on error in subsampling by implicitly or explicitly measuring the distance between the smooth or piecewise linear surface fit on the original and reduced point sets. A few other techniques use a simple distance-between-points based approach to reduce the number of points and cannot strictly guarantee any quality of simplification either in terms of surface approximation bounds or in terms of local sampling density.

One of the most commonly used surface representations is Point Set Surfaces (PSS) [ABCO*01] that reduce the redundant points by estimating a point's contribution to the Moving Least Squares (MLS) surfaces.

Pauly et al. [PGK02] introduce various surface simplification techniques to point sets. The incremental and hierarchical clustering, iterative simplification, and particle simulation algorithm are implemented to create approximations of point-based surfaces with lower sampling density. In iterative simplification, they use the influential surface simplification of the quadric error metric in [GH97]. The non-uniformity in sampling distribution is addressed by resampling using the MLS method which is based on sampling density estimation. However, this method does not provide any absolute or relative sampling density guarantee.

Moenning and Dodgson suggest a coarse-to-fine uniform or feature-sensitive point cloud simplification algorithm with user-controlled density guarantee [MD03]. In [MD04], they build upon the intrinsic point cloud simplification idea put forward in [MD03]. It supports both sub- and resampling of point sets for dealing with non-uniformity and undersampling in the input. Since both techniques are generic simplification algorithms, the simplification accuracy is not measured against any implicit or other surface representations. In other words, it has sampling density guarantee but does not guarantee any bounds on surface approximation.

Another stream of research that is relevant to our work is the explicit computation and use of curvature of the surface for simplification. In mesh simplification, the algorithm proposed by Turk [Tur92] uses repelling particles to re-sample the polygonized surfaces using curvature measurements. In [MGW05], a class of energy functions for distributing particles on implicit surfaces is used for the curvature based sampling of implicit surfaces. These techniques are fundamentally resampling techniques as against our subsampling technique. In other words, they use a (polygonal or implicit) surface representation to move the points on the surface for better positioning. They are computationally expensive and a coarser point set is not a strict subset of the finer point set.

Implicit shape representations have proven to be useful for modeling, animation, and visualization. RBFs (Radial Basis Functions) are popular for interpolating scattered data [SPOK95, TO99]. RBFs have been employed with better results than competing techniques, especially for producing high quality surfaces, repairing incomplete data and in methods requiring no topological constraints. The fundamental problem with RBF is its computation cost in terms of the need for solving large linear systems that is proportional to the size of the point data set. Hence its applicability is limited to small point sets.

For modeling the large point sets, CSRBFs that make the linear algebraic equation sparse [Wen95, MYR*01] and the fast RBF-based method using FMM (Fast Multipole Method) [CBC*01] are proposed. Moreover, multi-scale approach to interpolate point set surfaces with irregularly sampled and/or incomplete data [OBS03] and MPU (Multi-level Partition of Unity Implicits) making use of the partition of unity weights and local piecewise quadric approximation functions are suggested [OBA*03]. Since a hierarchical approach solve CSRBFs to reduce the implicit error at each level, it uses more CSRBF coefficients than single level CSRBFs. Further, since partition of unity approach needs the overlapped area to make the functions blended, we need more CSRBF coefficients too.

As for implicit representations applied for the shape compression, [CBC*01] proposes a coarse-to-fine greedy algorithm based on the fitting accuracy to reduce RBF centers to represent a surface. Since this method has to solve RBFs iteratively, it is computationally expensive.

In this paper, we propose a point cloud simplification technique for implicit representations and compress implicit functions. Since CSRBFs are able to reconstruct large point sets and CSRBF representations which are similar to RBF representations [CBC*01] are suitable for data compression, we use CSRBFs for the implicit surface reconstruction.

The rest of the paper is organized as follows. We first introduce the CSRBF implicit surfaces (Section 3), our choice of specific CSRBF surface (Section 3.1), measurement of error in fitting the CSRBF (Section 3.2), and computation of curvature in implicit surfaces (Section 3.3). Followed by this elaborate introduction on definition and computation techniques of CSRBF, we introduce our point simplification algorithm (Section 4) with special stress on the cost function. We analyze various effects and facets of this algorithm in Section 4.1. The algorithm described, reduces the number of points of the data set thus requiring less storage for the CSRBF data. We further compress this data using lossy and lossless compression techniques (Section 4.2). Finally we discuss the results of our methods with comparison to other state of the art techniques (Section 5).

3. The CSRBF Implicit Surfaces

Let us consider a set of points $\{\mathbf{p}_i\}_{i=1}^{N}$ scattered along a surface with unit normals $\{\mathbf{n}_i\}_{i=1}^{N}$. We wish to find a function f

that interpolates $\{\mathbf{p}_i\}_{i=1}^{N}$ implicitly.

$$f(\mathbf{p}_i) = 0, \quad i = 1, \cdots, N. \tag{1}$$

To make the solution of Equation 1 unique, off surface points $\{\mathbf{p}_i\}_{i=N+1}^{M}$ are appended to the input data along the unit normals $\{\mathbf{n}_i\}_{i=1}^{N}$, such that

$$f(\mathbf{p}_i) = 0, \qquad i = 1, \cdots, N, \tag{2}$$
$$f(\mathbf{p}_i) = d_i \neq 0, \qquad i = N+1, \cdots, M. \tag{3}$$

In this paper, we assume that the number of points on the surface and off the surface are the same: $M = 2N$. We create off-surface points on only the (inside or outside) of every surface point. The off-surface function values d_i are assumed to be positive outside the object and negative inside the object.

Now, the function values $f(\mathbf{x})$ at all points $\mathbf{x}$ in space can be defined using the basis functions, namely the CSRBF. Let $\mathscr{Q} = \{\mathbf{p}_i\}_{i=1}^{M}$ be input points. Then,

$$f(\mathbf{x}) = \sum_{\mathbf{p}_i \in \mathscr{Q}} \lambda_i \varphi_\sigma(||\mathbf{x} - \mathbf{p}_i||), \tag{4}$$

where $\varphi_\sigma(r) = \varphi(r/\sigma)$, $\varphi(r)$ is the Compactly Supported Radial Basis Function (CSRBF). The support size or, in other words, the range of influence of a point $\mathbf{p}_i$ is given by σ. The value of the support size σ is determined by the sampling density as recommended by [OBS03].

The CSRBF (Equation 4) implicit representation is basically given by the function centers $\{\mathbf{p}_i\}_{i=1}^{M}$ and their corresponding coefficients $\{\lambda_i\}_{i=1}^{M}$. Due to this simple representation structure, CSRBF are particularly amenable to data compression. Our scheme for CSRBF compression is described in Section 4.2.

3.1. Choice of CSRBF Basis Function

There are many different kinds of CSRBF functions that are used for different purposes and with different orders of continuity. We analyze a special kind of CSRBF called the Wendland's CSRBF [Wen95] which is generally used for surface reconstruction purposes, with C^0, C^2, C^4, and C^6 continuity as listed below, where CSRBFs are defined to be 0 if $r > 1$. We choose the one that gives us the best result.

C^0	$\varphi(r) = (1-r)_+^2$
C^2	$\varphi(r) = (1-r)_+^4(4r+1)$
C^4	$\varphi(r) = (1-r)_+^6(35r^2+18r+3)$
C^6	$\varphi(r) = (1-r)_+^8(32r^3+25r^2+8r+1)$

We reconstructed models under varying sampling density using the above four CSRBFs and measured the error of each of the reconstruction from the original point set. Figure 1 shows the reconstructed Stanford Bunny models using the four CSRBFs using the same reconstruction kernel (support size). From the graph in Figure 1, we see that for all sizes of the model, the quality of reconstruction of the surface with order C^2 CSRBF is better than the other three. Furthermore,

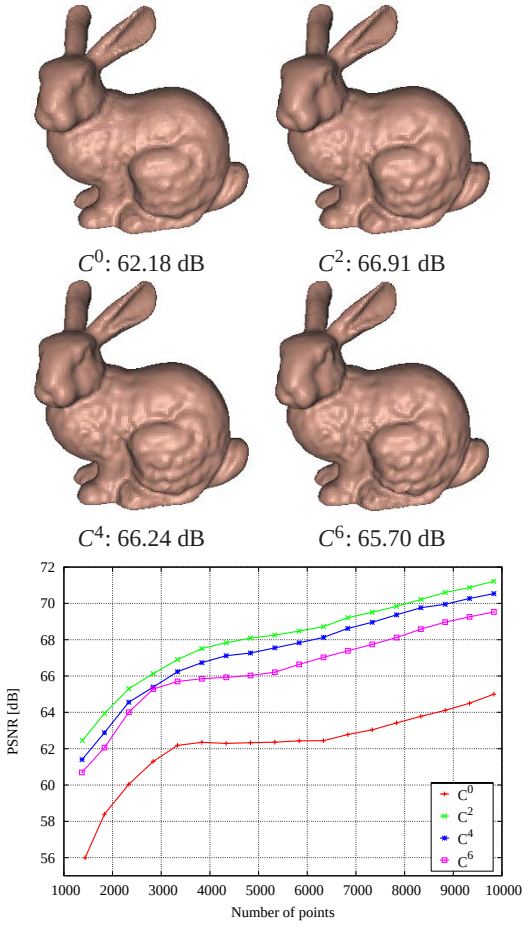


Figure 1: *Top four models: reconstruction of the simplified Stanford Bunny point set consisting of 3,334 points. Bottom graph: reconstruction accuracy of number of points and PSNR by four kinds of CSRBFs on Stanford Bunny model.*

the order C^2 CSRBF showed the best quality in other models (Armadillo and Venus) also. Hence we use Wendland's CSRBF with C^2 continuity as our representative function throughout this paper. In this paper, although we restrict our analysis to this particular CSRBF, note that our method is general enough to be applicable to any CSRBF. In the graphs shown, the quality of the reconstruction is measured using PSNR as described in Section 3.2.

3.2. Measuring Surface Fitting Error

Throughout the paper, we will be measuring the deviation of the implicit surface fit over a reduced set of points from the original model. The error in geometry is usually expressed in terms of Peak Signal to Noise Ratio (PSNR) . PSNR is defined as

$$\text{PSNR [dB]} = 20 \times \log_{10} \frac{\text{peak}}{d}, \tag{5}$$

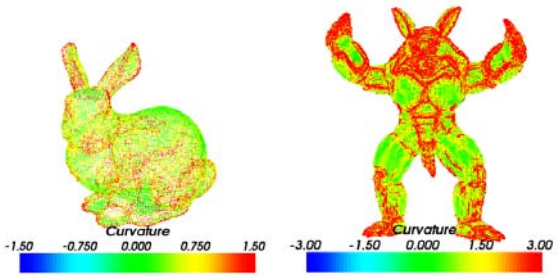


Figure 2: *Curvature measures of Stanford Bunny model and Armadillo model using MPU implicits.*

where "peak" is the model's bounding box diagonal and d is the distance of a point from the described geometry. While comparing triangular surfaces, d is measured using metrics like Haudorff metric. In case of implicit surfaces, Taubin [Tau91] justifies the use of a function specifically designed for implicit surfaces to measure d. This Taubin distance of a point $\mathbf{p}_i$ is given by $|f(\mathbf{p}_i)|/|\nabla f(\mathbf{p}_i)|$. In other words, each distance of a point from the zero set of the implicit surface is normalized using the rate of change of this function at that point. This accounts for the non-uniform scaling or warping of the coordinate system by the implicit function. For example, since an ellipsoid grows slowest along its smallest eigen vector, the 'unit-distances' along the three axes is expressed in terms of the velocity of the implicit front in the particular direction.

The above normalized distance measure is what is commonly used with implicit surfaces as done by, for example, [OBA*03] for adaptive octree subdivision of MPU implicits. We also use this distance function in the computation of PSNR as described below.

Let $\mathscr{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ be original point set. d is defined as the average of algebraic sum of Taubin distances.

$$d = \frac{\sum_{\mathbf{p}_i \in \mathscr{P}} |f(\mathbf{p}_i)|/|\nabla f(\mathbf{p}_i)|}{N}. \tag{6}$$

Note that this PSNR measurement is not specific to CSRBF, but applicable to any implicit function.

3.3. Measuring curvatures of implicit surfaces

The curvature of implicit surfaces is derived from its closed form expression, and from the first principles of differential geometry as the second order partial derivatives of the functions. In [AJ04], the computed curvature is used for polygonizing closed implicit surfaces. We make use of it for identifying the features and simplifying the point set based on these features. We briefly describe the method of computing the curvature for the sake of completion.

Let us consider the implicit function $f(\mathbf{x}) = 0$. The gradient vector $\mathbf{g}$ and the normal unit vector $\mathbf{n}$ are defined at the arbitrary point $\mathbf{x}$ using the first order partial derivative of $f(\mathbf{x})$.

$$\mathbf{g} = \nabla f, \quad \text{and} \quad \mathbf{n} = \mathbf{g}/\parallel \mathbf{g} \parallel . \tag{7}$$

The curvatures of function $f(\mathbf{x})$ are defined as matrix C by the partial derivatives of the normal $\mathbf{n}$.

$$C = \begin{bmatrix} \frac{\partial n_x}{\partial x} & \frac{\partial n_x}{\partial y} & \frac{\partial n_x}{\partial z} \\ \frac{\partial n_y}{\partial x} & \frac{\partial n_y}{\partial y} & \frac{\partial n_y}{\partial z} \\ \frac{\partial n_z}{\partial x} & \frac{\partial n_z}{\partial y} & \frac{\partial n_z}{\partial z} \end{bmatrix}. \tag{8}$$

The above matrix is the rank-deficient shape operator matrix of the surface whose first two eigen values give the principal curvatures κ_1 and κ_2, the third value being zero along the normal vector direction.

4. Our Point Cloud Simplification Algorithm

We have seen that the number of basis function centers and the coefficients for CSRBF is proportional to the number of point samples in the model. The goal of our point cloud simplification algorithm is to reduce the associated CSRBF data by reducing the number of points. The idea behind reducing the set is to assign an importance value or a cost value for every point in the data set and remove the point that is least important. Once a point is removed, its neighborhood points would gain more importance by virtue of fewer points representing that region, and hence their cost values are re-evaluated. The above process is repeated till a desired number of points have been removed.

The Cost Function: One of the main contributions of this work is the cost function to evaluate the importance of a point in the data set. Let $\mathscr{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ be input points scattered along a surface. We define the *suitability of the point* $\mathbf{p}_i$ *for removal* as

$$c_i = \frac{N_i \cdot \sum_{\mathbf{p}_j \in \mathscr{P}} \varphi_\sigma(||\mathbf{p}_j - \mathbf{p}_i||)}{K_i}, \tag{9}$$

where $\varphi_\sigma(r) = \varphi(r/\sigma)$, $\varphi(r) = (1-r)_+^4(4r+1)$ is Wendland's CSRBF [Wen95]. σ is its support size decided by the recommended support size [OBS03] scaled by a *subsampling support size factor*, α. N_i is the number of points within the radius σ of $\mathbf{p}_i$ and K_i is the curvature of the original surface at $\mathbf{p}_i$ computed as $K = \varepsilon + max(|\kappa_1|, |\kappa_2|)$, where $\varepsilon > 0$ is used to avoid division by zero in Equation 9. Since we use c as a relative measure for comparing the importance of points, the absolute value of ε is immaterial. We use $\varepsilon = 1.0$ in our experiments. More information on estimation of K is given later in this section.

The intuition behind this cost function is to have the distribution of points proportional to the local curvature. Higher curvature regions, indicating features, require higher distribution and vice-versa, and the ratio of these two quantities should be approximately the same at all points in the data set. The numerator of the cost function, $\sum_{\mathbf{p}_j \in \mathscr{P}} \varphi_\sigma(||\mathbf{p}_j - \mathbf{p}_i||)$

estimates the contribution of the point $\mathbf{p}_i$ to the CSRBF surface in $\mathbf{p}_i$'s neighborhood. The curvature at that point being a constant, if there are many points in the neighborhood, the importance of the contribution of a single point reduces and the point's importance increases if it is one of the very few points representing the neighborhood. This aspect of importance estimation is captured by the scale factor N_i. Hence, if c_i is high, the corresponding point's importance in its neighborhood is low, and hence the distribution of points in that region is unusually high with respect to the curvature. Such points are best candidates for removal.

Curvature Estimation: In order to compute the suitability of every point for removal, we need to calculate the curvature of the surface at all the points. Computation of curvature (refer to Section 3.3) requires fitting of implicit surface. Note that the goal of our algorithm is to make the CSRBF computation viable by reducing the point set, and this prohibits the use of CSRBF to the initial data set in order to compute the curvature. Since normal curvature is an inherent property of a surface and independent of its functional representation, any smooth function that arguably represents the surface of the point set correctly, can be used. We use a fast, local implicit function, the MPU implicits [OBA*03], to compute the curvature of the original model (and is computed only once). Figure 2 shows the maximum absolute curvatures of Stanford Bunny and Armadillo models. Notice the expected curvature measurements in the parts of eyes of the Bunny model and hands and legs of the Armadillo model.

4.1. Analysis of the Algorithm

Putting things together, we first compute the curvature at every point on the given data set using MPU implicit fitting. Once the curvatures are set, the suitability cost of every point for removal is calculated as given by Equation 9. The point with maximum value of c, $\mathbf{p}_{max}$ is removed. This removal would affect the cost of all points within a radius of σ from $\mathbf{p}_{max}$, hence they are recomputed. From the new set of cost values, the point with maximum value of c is again chosen, and the above process is repeated till a desired number of point reduction is achieved. Note that the curvature is computed only once based on the original model and is not recomputed for every iteration.

The distribution of the values of the cost c is an important indicator of the uniformity of distribution of points with respect to the requirement demanded by the local curvature. In effect, our simplification algorithm strives to make the cost value same for all points through a single operation, namely the point removal. Hence our algorithm is strictly a subsampling algorithm in which subsequent levels of detail of the model is a strict subset of its parent model. The histograms shown in Figure 3 illustrate the distribution of c after a sequence of point removal operations. The shrinking of the range of c indicates increasing uniformity in the values of c.

Note that over the sequence of point removal operations, the values of c_i monotonically reduces. Hence the histogram in Figure 3 will only be pushed towards $c = 0$. Ideally, there are two (probably dependent) parameters in the algorithmic design space – the value of c and the number of points in the final model. If another fundamental operation, namely, the *addition of a point* is allowed, then we can explore the entire two dimensional solution space. Specifically, given a value of c_f, we can add points in the region with lower c to increase its value, and remove points as before in the regions of higher value of c to reduce its value, in effect increasing the number of points with $c = c_f$. With this additional operation, we can perform both subsampling and resampling of the given data set.

4.2. Reconstruction and Compression of CSRBF surfaces

We fit a CSRBF surface on the subsampled point data set using the same function in Equation 4, where σ is its support size decided by the recommended support size [OBS03] scaled by a *reconstruction support size factor*, β. The value of β may be different from subsampling support size factor α whose value is not known at the reconstruction stage. We achieve a major compression in surface representation just by subsampling the point set. Further compression can be achieved by encoding the centers and the coefficients of the final CSRBF fit on the reduced point set.

Since CSRBFs solve the linear algebraic equation, the surfaces generated by the CSRBFs are very sensitive to the coefficient values. Since even small perturbations in these coefficients are undesirable, we use a lossless compression scheme like PPM to encode them. On the other hand, the CSRBF surfaces are impervious to slight perturbations of the point locations and their normal vectors. Hence we use a lossy compression for vertices and normals in our coder.

We encode the simplified point set $\mathbf{p}_i$ and its normal vectors $\mathbf{n}_i$ using lossy scalar quantization and arithmetic coding, and the CSRBF coefficients λ_i using lossless Prediction by Partial Matching (PPM) coding [CW84].

5. Results and discussion

We compare our point cloud simplification technique against known mesh simplification techniques for different models shown in Figures 4 and 5. The technique [Hop99] using quadric error metric [GH97] (Quadric Decimation), the vertex clustering technique by quadric error [Lin00] (Quadric Clustering), the decimation algorithm of triangle meshes [SZL92] (Decimation), and the progressive mesh decimation based on [SZL92, Hop96] (Progressive Decimation) are the other techniques used to reduce the number of points. We fit the CSRBF surface over these simplified points for comparison. We use Visualization Tool Kit (VTK) implementation of all these algorithms.

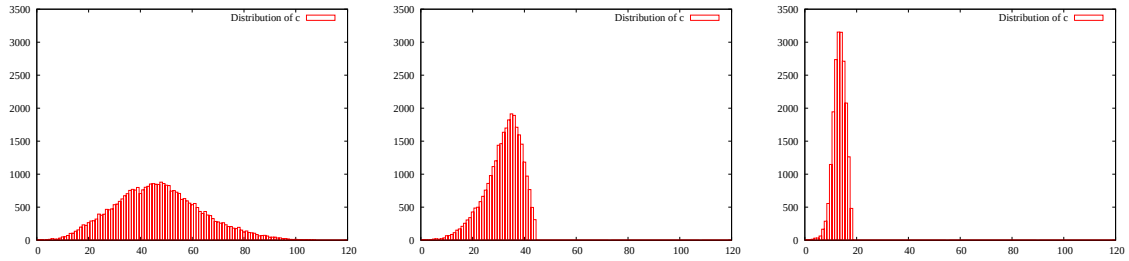


Figure 3: *Distribution of c on Stanford Bunny data set: (left) 34834 points, (middle) 29834 points, and (right) 19834 points with $\alpha = 1.0$.*

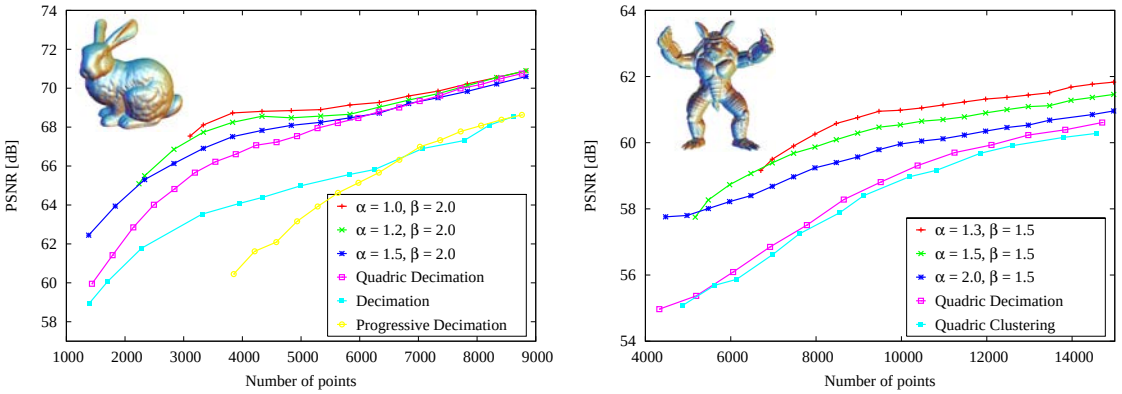


Figure 4: *Simplification accuracy of number of points and PSNR on Stanford Bunny model consisting of 35,947 points (left graph) and Armadillo model consisting of 172,974 points (right graph).*

Figure 4 shows the simplification accuracy in terms of number of points and PSNR. Our subsampling was done using three different α values and reconstruction using the same β value in all cases. The minimal number of points are different for each α because of simplifying until the number of points in the support of each CSRBF becomes one. Thus our technique can lead to minimal density with respect to the feature of the point. The PSNR is better for lower values of α since a smaller support size can capture the features of the model better than larger supports. Further, in our experiments with both the Bunny and the Armadillo models, our method is in general better than all other methods. In the Bunny model, this is especially true for simplified model with less than 5000 points, since for fewer number of points, the sampling becomes more regular in our simplification process.

Figure 5 shows our simplification results on the Venus model. The top row shows the reconstructed models with the points from our subsampling technique. The middle row shows the reconstructed models from the points reduced by quadric decimation [Hop99]. As both techniques reconstruct the models well, results of our technique seem to look smoother since at lower point count the uniformity in sampling gains more importance than the curvature. Further, this smoothening effect also distributes the error thus increasing the PSNR. These effects can be deduced from the accompanying graph.

The timings for simplifying the points depend on the model size and the support size of CSRBF. On Stanford Bunny model, points are simplified to 10% of its original size at a rate of over 4100 points per second for various values of α; over 3300 points per second to reduce the Venus model to 5% of its original size, and over 2500 points per second to reduce the Armadillo model to 5% of its original size. Time benchmarks are measured on Intel Celeron M 1.3 GHz processor and 752 MB RAM. The most often repeated operation in our algorithm is finding the point with maximum c, removing that point (and hence the c value) from the list, and recomputing the values of c in its neighborhood. We used a priority queue to do these operations.

Figure 6 shows the compression results of CSRBFs on Stanford Bunny. The top row shows the reconstructed models from the compressed data. Our coder uses PPM coding for CSRBF coefficients, and scalar quantization and arithmetic coding for vertices and normals. The middle row

shows the reconstructed models from the lossy scheme. The lossy scheme codes vertices, normals, and CSRBF coefficients by scalar quantization and arithmetic coding. On the top left model consisting of 1,381 points and the middle left model consisting of 1,834 points, our coder shows better PSNR than the lossy scheme. On the other four models, our coder has about 2.0 to 5.0 PSNR better than the lossy scheme. The graph compares rate-distortion curves with various bits of vertices (V), normals (N). The bits of CSRBF for the lossy scheme are set to 12 bits. The combinations of 12 bits of vertices and 6 bits of normals show the best PSNR. As can be seen from the lossy scheme, CSRBF coefficients are very sensitive for data loss.

6. Conclusions and future work

We have described a new point cloud simplification technique for implicit representations. By using the curvatures of implicit functions and the distribution of CSRBFs, our technique is feature-sensitive, has guaranteed sampling density, is easy and efficient to compute, and simple to implement. The quality of the simplification and reconstruction is highlighted by the PSNR graphs especially in the small number of points. We have also introduced compression framework for CSRBFs. Our coder combines the lossless scheme for CSRBF coefficients and the lossy scheme for vertices and normals. Compressed implicit functions can be directly used in all applications that use implicit functions without the need for refitting the data.

The future work includes employment of developing algorithm for the next logical fundamental operation: point addition, and finally developing algorithm that combines addition and removal of points for effective resampling of the point set. The resampling technique will be useful in data sets more irregularly sampled and/or incomplete regions. The better coder of vertices and normals should be implemented and the the way to reduce the entropy of CSRBF coefficients losslessly should be investigated.

References

[ABCO*01] ALEXA M., BEHR J., COHEN-OR D., LEVIN D., SILVA C. T.: Point set surfaces. In *Proc. of IEEE Visualization* (2001), pp. 21–28.

[AJ04] ARAÚJO B., JORGE J.: Curvature dependent polygonization of implicit surfaces. In *Proc. of SIBGRAPI/SIACG* (2004), pp. 266–273.

[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. of ACM SIGGRAPH* (2001), pp. 67–76.

[CW84] CLEARY J. G., WITTEN I. H.: Data compression using adaptive coding and partial string matching. *IEEE Trans. on Communications 32*, 4 (1984), 396–402.

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proc. of ACM SIGGRAPH* (1997), pp. 209–216.

[Hop96] HOPPE H.: Progressive meshes. In *Proc. of ACM SIGGRAPH* (1996), pp. 99–108.

[Hop99] HOPPE H.: New quadric metric for simplifying meshes with appearance attributes. *Proc. of IEEE Visualization* (1999), 59–66.

[Lin00] LINDSTROM P.: Out-of-core simplification of large polygonal models. In *Proc. of ACM SIGGRAPH* (2000), pp. 259–262.

[MD03] MOENNING C., DODGSON N. A.: A new point cloud simplification algorithm. In *3rd Int. Conf. on Visualization, Imaging and Image Processing* (2003), pp. 1027–1033.

[MD04] MOENNING C., DODGSON N. A.: Intrinsic point cloud simplification. In *Proc. of GraphiCon* (2004).

[MGW05] MEYER M. D., GEORGEL P., WHITAKER R. T.: Robust particle systems for curvature dependent sampling of implicit surfaces. In *Shape Modeling and Applications* (2005), pp. 124–133.

[MYR*01] MORSE B. S., YOO T. S., RHEINGANS P., CHEN D. T., SUBRAMANIAN K. R.: Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Proc. of Shape Modeling International* (2001), pp. 89–98.

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

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

[PGK02] PAULY M., GROSS M., KOBBELT L. P.: Efficient simplification of point-sampled surfaces. In *Proc. of IEEE Visualization* (2002), pp. 163–170.

[SPOK95] SAVCHENKO V., PASKO A., OKUNEV O., KUNII T.: Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum 14*, 4 (1995), 181–188.

[SZL92] SCHROEDER W. J., ZARGE J. A., LORENSEN W. E.: Decimation of triangle meshes. In *Proc. of ACM SIGGRAPH* (1992), pp. 65–70.

[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. on Pattern Analysis and Machine Intelligence 13*, 11 (1991), 1115–1138.

[TO99] TURK G., O'BRIEN J. F.: Shape transformation using variational implicit functions. In *Proc. of ACM SIGGRAPH* (1999), pp. 335–342.

[Tur92] TURK G.: Re-tiling polygonal surfaces. In *Proc. of ACM SIGGRAPH* (1992), pp. 55–64.

[Wen95] WENDLAND H.: Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. *Advances in Computational Mathematics 4* (1995), 389–396.



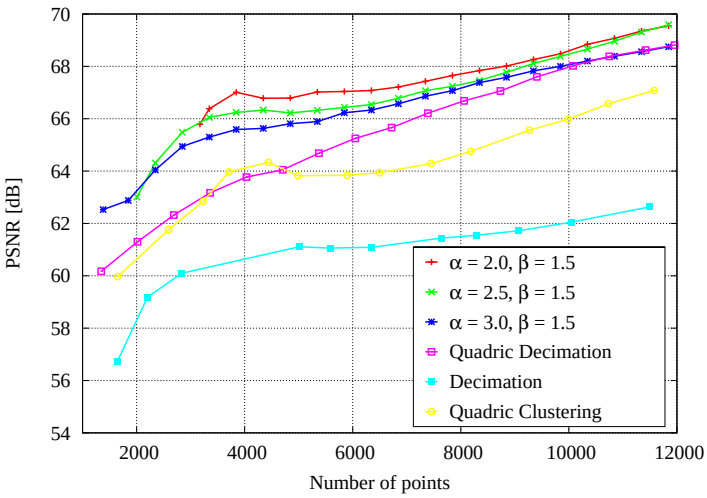
2006 points, 63.01 dB 5345 points, 66.32 dB 11345 points, 69.31 dB

Reconstructed models (with $\beta = 1.5$) from points from our subsampling technique (with $\alpha = 2.5$).



2017 points, 61.30 dB 5375 points, 64.68 dB 11421 points, 68.62 dB

Reconstruction (with $\beta = 1.5$) from the point set given by quadric decimation algorithm.



Original model: 134,345 points

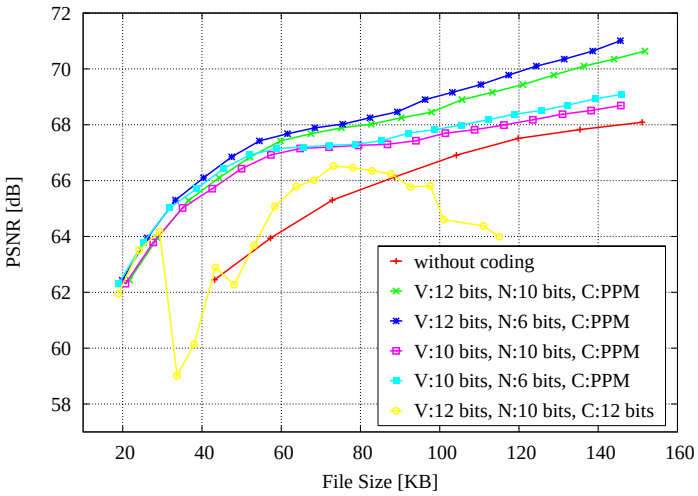
Figure 5: *Simplification results of number of points and PSNR on Venus model.*

21.7 KB, 62.44 dB | 67.5 KB, 67.68 dB | 113.3 KB, 69.16 dB

Compression using lossy and lossless schemes

18.9 KB, 61.95 dB | 68.3 KB, 66.01 dB | 111.0 KB, 64.38 dB

Compression using lossy scheme



Original model: 1,235 KB

Figure 6: *Compression results of file size and PSNR on Stanford Bunny model.*