Generalized MPU Implicits Using Belief Propagation

Yi-Ling Chen^{*} Shang-Hong Lai Tung-Ying Lee[†] Department of Computer Science National Tsing Hua University, Hsinchu, Taiwan

Abstract

In this paper, we present a new algorithm to reconstruct 3D surfaces from an unorganized point cloud based on generalizing the MPU implicit algorithm through introducing a powerful orientation inference scheme via Belief Propagation. Instead of using orientation information like surface normals, local data distribution analysis is performed to identify the local surface property so as to guide the selection of local fitting models. We formulate the determination of the globally consistent orientation as a graph optimization problem. Local belief networks are constructed by treating the local shape functions as their nodes. The consistency of adjacent nodes linked by an edge is checked by evaluating the functions and an energy is thus defined. By minimizing the total energy over the graph, we can obtain an optimal assignment of labels indicating the orientation of each local shape function. The local inference result is propagated over the model in a front-propagation fashion to obtain the global solution. We demonstrate the performance of the proposed algorithm by showing experimental results on some real-world 3D data sets.

1. Introduction

In the last few decades, *shape modeling* or *surface reconstruction* has been an active research topic in many fields, such as computer graphics, scientific visualization, CAD, and medical imaging. Recent trend tends to design surface reconstruction algorithms suitable for several different purposes, such as 1) reconstruction from large data sets, 2) robustness to noisy data, 3) preserving sharp features, 4) hole filling or repairing of incomplete data. Among those developed algorithms, many of them exploit domain knowledge or make specific assumption on the input data. For example, algorithms for reconstructing surfaces from a set of range data usually take advantages of the adjacency relationship between each range image [25] to merge the range



Figure 1. Reconstruction of fine-quality surfaces from unorganized point clouds by using the MPU implicits algorithm. A novel algorithm based on Belief Propagation is proposed to determine the global consistent orientation for the local implicit patches.

data. Besides, the fact that each data point is visible from the sensor simplifies the task to orientate the surface [6]. This is actually a non-trivial task when dealing with unorganized point data set. Exploiting the information accompanying the input data can help devise algorithms with superior performance in terms of, for example, time and space efficiency. However, these approaches might be difficult to adapt to another application domain or even if the input data are slightly different from their assumption.

Another aspect of designing surface reconstruction algorithm is to make as least assumption as possible, such as the work by Hoppe et al [11]. The main challenge for the problem is thus to infer the necessary information during the reconstruction process. One essential part of their algorithm is to estimate the surface normal for each data point

^{*}This work was done when the author is with Siemens Corporate Research, Princeton, USA.

[†]{yilin,lai,tylee}@cs.nthu.edu.tw

and then use them to define the signed distance function of the unknown surface. This makes the algorithm potentially vulnerable to noisy data since reliable estimation of orientation information from the data might not be possible. On the other hand, the estimated normal vectors should be consistently aligned to correctly define the overall signed distance function. This is achieved by propagating normal orientation along a minimal spanning tree (MST) constructed over the entire data set. Their method works well under the assumption that the data points are sufficiently close to each other and the directions of normal vectors vary smoothly along the surface. However, this assumption may be too rigorous because the algorithm might fail when there are abrupt changes in normal directions (e.g. around sharp features or non-uniformly sampled data) or the normal direction information is not reliable (e.g. noisy data). Thus, there is strong need for a robust orientation determination scheme which can be used with a wide variety of surface reconstruction algorithms.

In this paper, we introduce a novel approach to generalize the Multi-level Partition-of-Unity (MPU) implicit algorithm proposed by Ohtake et al [15] to deal with a broader range of input data. The MPU implicit algorithm is an adaptive octree approximation method which reconstructs implicit surfaces by smoothly blending a set of locally fitted shape functions. By using surface normals, it provides a scheme to adaptively choose local fitting models and preserve sharp features by identifying the distribution of normal vectors. In addition, the surface normals also provide an aid to orientate the local shape functions. We do not assume that the orientation information is available in the input data. Instead, we propose to determine the globally consistent orientation by propagating messages over local belief networks with the local shape functions treated as their nodes. We also demonstrate how we can benefit from the covariance analysis on point sampled surface to guide the selection of local fitting models.

We have also noticed some previous works proposed to resolve similar orientation determination problem, such as the active contour method proposed by Xie et al. [30]. It is not, however, a trivial task for their method to achieve adaptive approximation because the feature size of the model should be determined first. Our approach has no such limitations and is potentially more efficient since we propagate local orientation inference results to obtain the globally optimal solution and no global structure like the "monooriented region" used in their method needs to be determined in advance.

2. Related Works

Generally speaking, in spite of the diversity of the underlying mathematical forms, implicit surfaces typically aim to find a continuous embedding function $f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}, \mathbf{x} \in \mathbb{R}^d$, whose zero level-set describes the unknown surface. In other words, the locus of positions in \mathbb{R}^d where the function f takes the value of zero compose the surface being modeled. The implicit function f partitions the space into two parts: inside and outside of the surface, where it is positively and negatively valued, respectively. In a sense, implicit surface modeling techniques try to derive a *signed distance field* subject to the data points sampled from the unknown surface. Compared to explicit surface models, implicit surface modeling possesses the advantages of being robust to noises and non-uniform sampling density. In addition, it is easy to convert implicit models into other representations and perform CSG operations.

The pioneering works on implicit surfaces include the blobby surface [2] which used the sum of implicit primitives such as Gaussian blobs. Taubin [23] developed an approximation distance measure for implicit algebraic curve and surface fitting. Hoppe et al. [11] introduced the concept of modeling surface as a signed distance field which is defined as the distance to locally fitted tangent planes. Curless and Levoy [6] proposed an efficient algorithm to integrate a set of range scans to build the signed distance function. Alexa et al. propose to use the moving least-squares (MLS) operator to approximate point sampled surfaces [1]. Another important class of implicit modeling techniques is based on the variational method and radial basis functions (RBFs) [27, 26, 7, 8]. In a nutshell, the variational method imposes an additional smoothness constraint that minimizes the aggregate curvature of the function, thus leading to the smoothest function among all possible solution functions satisfying the interpolation conditions.

Conceptually, an implicit surface is modeled by a single implicit function. However, it is a non-trivial task to find an f that well fits to the model. For example, solving an interpolant consisting of a number of radial basis functions is highly computational expensive [27, 26, 7, 8]. Much effort has been made to seek efficient computational schemes to this interpolation problem, such as compactly supported RBFs [14, 16] and far field expansion [5]. In the case of algebraic surfaces, the result of fitting an polynomial to a complex surface is highly unpredictable. Thus it is usually easier to create complex surfaces by a set of algebraic patches. This brings another problem about how to smoothly join the patch boundaries. As a domain decomposition method, the partition-of-unity method has attracted much attention in recent research of implicit surface modeling. The concept behind it can be traced back to Shepard's blending method [20] and then modified by Franke and Nielson [10], which was originally designed to build a global function by subdividing the problem domain into a set of sub-domains where locally defined functions are solved and provides a good means of smoothly blending



Figure 2. Algorithm Flow. From left to right: The reconstruction algorithm accepts an unorganized point cloud \mathcal{P} as input. An adaptive octree subdivision is performed on \mathcal{P} and local shape functions are created for each octree cell. An orientation inference algorithm based on Belief Propagation is applied in a front-propagation fashion. The implicit surface is generated by smoothly blending the orientated implicit patches.

the local solution functions to form a good approximation for the global solution of interest. Based on this framework, many hierarchical reconstruction algorithms have been proposed. The MPU implicits method introduced by Ohtake et al. [15] used an octree-based adaptive approach for surface reconstruction from a large and accurate data set. Sharp features can be well preserved by local selection of fitting models. Xie et al. later extended the MPU implicit to handle noisy data sets [30]. They employed an active contour method and a voting process for orientation determination. Tobor et al. [24] proposed a multi-scale reconstruction algorithm by building a binary tree decomposition and performing a bottom-up thinning operation.

A thorough description of the wide variety of implicit surfaces is obviously beyond the scope of this paper. More detailed introduction to implicit surfaces is referred to the book by Bloomenthal et al. [4].

3. Generalized MPU Implicits

3.1. Review of MPU Implicits

Our approach is based on the framework of the MPU implicit algorithm [15]. In this section we briefly describe the concept of this framework. The algorithm accepts a 3D point cloud $\mathcal{P} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ sampled from an unknown surface. Different from the original algorithm, the proposed algorithm does not assume the surface normal at each point be available. Creating an MPU implicit surface starts by partitioning \mathcal{P} into a collection of overlapping subsets $\mathcal{P}^1, \mathcal{P}^2, \dots, \mathcal{P}^M$. To decompose the global domain where \mathcal{P} is occupied, the entire data set is rescaled and inserted into an unit-length bounding box. A hierarchy of octree is built by recursively subdividing the bounding box and its children into eight equal octants. Each cell has a spherical support region of radius R_i , which is centered at the cell center \mathbf{c}_i , as illustrated in Figure 3. The points of \mathcal{P} lying within the support region of an octree cell are collected as \mathcal{P}^i , and a local shape function $Q_i(\mathbf{x})$ is created through a least-squares fitting procedure. The local shape functions are then smoothly blended as a pseudo-signed distance field by a set of non-negative and compactly supported weighing functions ω_i . According to the modified Shepard's method [20, 10], the blended function f can be expressed as follows:

$$f(\mathbf{x}) = \frac{\sum_{i} \omega_i(\mathbf{x}) Q_i(\mathbf{x})}{\sum_{i} \omega_i(\mathbf{x})},$$
(1)

There is a rich choice of blending functions for $\omega_i(\mathbf{x})$ and here we also use the B-spline quadratic function proposed by Ohtake et al. [15], which has the following form:

$$\omega_i(\mathbf{x}) = \frac{3|\mathbf{x} - \mathbf{c}_i|}{2R_i} \tag{2}$$

During the construction of MPU implicits, a series of tests are performed to determine the type of local surface and an appropriate fitting model is selected. This was achieved by primarily exploiting surface normal information. After the local approximation function Q_i has been computed, an approximation error $\varepsilon = \max_{\mathbf{X} \in \mathcal{P}_i} |Q_i(\mathbf{X})| / |\nabla Q_i(\mathbf{X})|$ is estimated according to the approximation distance for algebraic surfaces developed by Taubin [22]. If the approximation error is greater than a pre-defined threshold ε_0 , the octree cell is subdivided and the same fitting procedure is performed on the subcells until the desired fitting accuracy is achieved.



Figure 3. Left: Fitting a bivariate quadratic polynomial. Right: Local fit of a general quadric.

3.2. Local Fitting Model Selection

One essential component of the MPU implicits approach is that it provides a class of local fitting models suitable for different types of local surface properties as well as a strategy to determine appropriate local fitting models. Particularly, sharp features, such as edges and corners, can be effectively detected by identifying the distribution of the normal vectors [12]. The three different types of local fitting models adapted by MPU implicits are briefly summarized below:

- general quadric for larger or separated surface patches (see Figure 3 right),
- bivariate quadratic polynomial with local coordinates aligned to averaged normal direction, which is used for local smooth patches (see Figure 3 left),
- piecewise quadric surfaces fitted to an edge or corner.

The detailed formulation of the local shape functions is referred to the original paper by Ohtake et al. [15]. To deal with more general sources of input data, we employ the notion of *surface variation* proposed by Pauly et al. [17] to guide the selection of local fitting models without utilizing surface normals. A surface variation estimate is obtained by performing principal component analysis (PCA) on local point neighborhood. Let $\bar{\mathbf{x}}^i$ be the centroid of \mathcal{P}^i within the domain of influence for an octree cell and **C** be the covariance matrix defined as

$$\mathbf{C} = \frac{1}{n} \begin{bmatrix} \mathbf{x}_{1}^{i} - \bar{\mathbf{x}}^{i} \\ \dots \\ \mathbf{x}_{n}^{i} - \bar{\mathbf{x}}^{i} \end{bmatrix}^{T} \begin{bmatrix} \mathbf{x}_{1}^{i} - \bar{\mathbf{x}}^{i} \\ \dots \\ \mathbf{x}_{n}^{i} - \bar{\mathbf{x}}^{i} \end{bmatrix}, \ \mathbf{x}_{j}^{i} \in \mathcal{P}^{i} \quad (3)$$

The surface variation σ_n can then be defined as

$$\sigma_n = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{4}$$

where *n* is size of \mathcal{P}^i and $\lambda_0, \lambda_1, \lambda_2$ are the eigenvalues of **C** in an increasing order. Note that originally σ_n is defined

by the *n*-nearest neighborhood of a specific point. The interpretation of σ_n can be seen as the extent that the local surface approaches a local tangent plane. A large σ_n indicates that the points in \mathcal{P}^i tend to be distributed isotropically. In contrast, a small σ_n means that the points in \mathcal{P}^i lie approximately on the plane spanned by the eigenvectors corresponding to λ_1 and λ_2 . The surface variation σ_n of \mathcal{P}^i can thus be used as a guidance for selecting local fitting models. We set a threshold of 0.25 on σ_n for model selection. If $\sigma_n > 0.25$, a quadric is fitted to \mathcal{P}^i ; otherwise, we choose a bivariate quadratic polynomial. If a bivariate quadratic polynomial is fitted, the local coordinate system is aligned to the eigenvector corresponding to λ_0 .

The points around a sharp feature like edges or corners are distributed on multiple surface patches and cannot be easily detected by simply examining the surface variation. It is worth noting that Pauly et al. [18] developed a method based on surface variation for detecting sharp feature in point sampled surfaces. We tentatively implement a k-means like algorithm for multiple surface fitting by using surface variation as error measurement. Although the current results are not stable, the basic idea is to iteratively group the data points into several clusters and minimize σ obtained from each cluster. The piecewise quadric fitting can then be applied to the individual clusters. We believe that the robust estimation technique used in [9] will also be helpful in solving such a clustering problem. For sharp feature preservation techniques, we also refer the readers to the papers by Xie et al. [30] and Fleishman et al. [9], which also addressed the problem of noisy input data.

4. Globally Consistent Orientation Inference

The above discussion focused on an error controlled mechanism of local shape function fitting. However, the orientation of each local shape function Q_i with respect to the unknown manifold remains ambiguous. More specifically, on which side of the local implicit patch should Q_i take positive or negative values is not properly defined. Recall that the implicit surface is modeled as a pseudo-signed distance function smoothly blended by the local shape functions Q_i (please refer to equation (1)). Unfortunately, the local shape functions resemble the unknown surface only within its region of support. In other words, the orientation of a local function is not an intrinsic property and should be defined globally in consideration with other local functions. This implies that we cannot easily obtain consistent orientations for all local functions by simply enforcing them to have the same sign at some specific locations. We thus need a more sophisticate algorithm to infer the globally consistent orientation for the local functions.

4.1. Belief Propagation Optimization

Given a set of local shape functions Q_i , we are uncertain of their orientations. The goal of globally consistent orientation inference is to determine, for each local function Q_i , a label ℓ_i so as to resolve the ambiguity of the two possible orientations for each Q_i and also to consistently align with other local functions. Typically, an implicit function f for a surface takes positive/negative values inside/outside the surface. Following this convention, we term a local function Q_i as *consistent* if $Q_i(\mathbf{x})$ has the same sign as $f(\mathbf{x})$, otherwise it is regarded as inconsistent. Note that this condition holds only within most of the support region of Q_i . The overall optimization is formulated as solving a graph labeling problem. The algorithm proceeds by iteratively constructing a local graph and applying Belief Propagation algorithm to find a label $\ell_i \in \{1, -1\}$ for each local function, which corresponds to consistent and inconsistent, respectively. Analogous to other orientation alignment schemes [11, 30], the globally consistent orientation is obtained by negating the function values of Q_i labeled as -1.

In detail, let us reconsider the problem by starting from a set of local functions Q_i that are geometrically close to each other. We construct a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with each node in \mathcal{V} corresponding to one of the local functions. For any pair of nodes in \mathcal{V} to be connected by an edge, they should correspond to octree cells that are adjacent to each other. More specifically, only when the octree cells C_i and C_j where Q_i and Q_j are defined share common corners can they compose an edge e_{ij} . We make use of the corners of a cell as auxiliary points by exploiting the observation that local shape functions which are geometrically adjacent to each other are very likely to be consistently signed in their overlapped support region, as illustrated in Figure 4. We can thus define the following energy on \mathcal{G} :

$$E(\mathcal{L}) = \sum_{i \in \mathcal{V}} E_d(\ell_i) + \sum_{(i,j) \in \mathcal{E}} E_s(\ell_i, \ \ell_j), \tag{5}$$

$$E_d(\ell_i) = -\ell_i,$$

$$E_s(\ell_i, \ \ell_j) = \sum_k -\ell_i \cdot \ell_j \cdot sign(Q_i(\mathbf{x}_k)Q_j(\mathbf{x}_k))$$

where k is the index of the common corners shared by C_i and C_j , and $sign(\cdot)$ is a binary function that returns 1 or -1when its argument is positive or negative, respectively. Note that by evaluating $sign(Q_i(\mathbf{x}_k)Q_j(\mathbf{x}_k))$, a returned value of 1 indicates that Q_i and Q_j have consistent orientation at \mathbf{x}_k , otherwise they are inconsistent. Therefore, $E_s(\ell_i, \ell_j)$ behaves like a smoothness energy which penalizes the inconsistent changes of signed distance field associated with two neighboring nodes. Denote k-connected neighbors of a cell



Figure 4. For two local shape functions Q_i and Q_j defined on adjacent octree cells. Q_i and Q_j are very likely to be consistently signed within their overlapped support region, such as the common corners illustrated in the figure.

 C_i as those adjacent cells of C_i that share k common corners with C_i . For robustness, we only include 4-connected neighbors into the graph optimization. Note that an octree cell has only six 4-connected neighbors. $E_d(\ell_i)$ is the data energy measuring how well the estimated labels fits to our prior knowledge about the model. Here we let E_d be just the sum of negated label of node i, and it acts as an additional constraint that selects one of the two optimal label assignments with more nodes labeled as 1. The optimal labels \mathcal{L} for all nodes are obtained by minimizing the total energy $E(\mathcal{L})$.

In a sense, the graph \mathcal{G} can be regarded as a joint probability of a set of binary random variables with the edges indicating the dependency between distinct nodes. The inference of *marginal* probability of a specific node means that we have to sum over all the possible states of other nodes, which can be a very laborious process. Belief Propagation (BP) is an efficient probability inference algorithm proposed by Pearl [19], which has recently been proven to be very effective in many computer vision and image processing problems [28, 21]. It comes with two variants: *sumproduct* and *max-product*. In this paper, we use the max-product algorithm to find a *maximum a posteriori* (MAP) solution formulated earlier in this section by taking negative log probabilities.

Briefly speaking, Belief Propagation works by iteratively propagating *messages* or *beliefs* along edges over a graph. Let us denote the message sent from node *i* to *j* as $m_{i\rightarrow j}$. In our formulation, message $m_{i\rightarrow j}$ is a two-dimensional vector, and can be regarded as the probability density function that node *j* takes label 1 or -1. The local message passing mechanism is for each node to receive and update message by forming product of incoming messages and local evidence as the following equation:

$$m_{i \to j}(\ell_j) = Z \max_{\ell_j} (E_d(\ell_i) \cdot E_s(\ell_i, \ell_j))$$

$$\prod_{\substack{\in Nbhd(i) \setminus j}} m_{u \to i}(\ell_i)),\tag{6}$$

where $Nbhd(i) \setminus j$ denotes the neighbors of *i* other than *j*, and *Z* is a normalization factor. Finally, the MAP solution for each node *i* is computed as

$$\ell_i^* = Z \max_{\ell_i} (E_d(\ell_i) \cdot \prod_{u \in Nbhd(i)} m_{u \to i}(\ell_i)), \quad (7)$$

For tree structured graphs, BP is guaranteed to converge to a fixed message m^* after at most T iterations depending on the longest path of the graph. For graphs with loops, Loopy BP [29] can be applied to obtain good approximate solutions.

4.2. Advancing-Front Algorithm

Intuitively, to obtain the optimal labels, a graph consisting of all the local functions should be constructed for the energy minimization procedure. However, we choose to infer the orientations locally and iteratively propagate the partially determined results over the entire model to obtain the global solution due to the following two observations:

- 1. The interaction of signed distance fields between local functions is essentially a local property. That is the reason that we construct a graph that only encodes the relationship of adjacency. In addition, local orientation inference typically gives accurate results.
- 2. Previous results provide additional information for further inference.

Therefore, we devise the following advancing-front algorithm:

Algorithm 1 OrientationPropagate($\mathcal{F} = \text{null}$)				
Select seed cell v and insert v into \mathcal{F} .				
while \mathcal{F} not empty do				
active cell $v = \mathcal{F}$.front;				
collect node set \mathcal{V} ;				
construct graph \mathcal{G} with \mathcal{V} ;				
perform BP on \mathcal{G} ;				
update labels of \mathcal{V} ;				
find and insert adjacent non-empty cells of \mathcal{G} into \mathcal{F} ;				
end while				
return				

Initially, we have a collection of MPU implicits with their orientation undetermined. Starting from a *seed* implicit patch with its orientation as the basis for alignment. The algorithm proceeds by iteratively marching an active octree cell to traverse the unchecked parts of the surface until all the implicits have been properly orientated. Our algorithm maintains a data structure of *front* \mathcal{F} , which contains

 Table 1. Performance of the proposed algorithm.

 Computational time is represented in seconds.

Model	# of points	BP	MPU
Bunny	35647	5.67	40.36
Rabbit	67038	4.03	70.35
Isis (Reduced)	34051	6.29	47.7
Igea	134345	5.53	117.27
Armadillo	172970	10.36	184.38

a queue of candidate cells to be selected for the orientation inference process in the next iteration. The basic idea of this incremental algorithm is straightforward: traversing all the non-empty leaves among the octree in the order of connected components of visited cells. The belief propagation inference is performed on a graph constructed subject to the active cell. There is, of course, not only one graph construction procedure (4th and 5th lines of Algorithm 1) suitable for the BP inference algorithm. We briefly describe our heuristic but effective graph construction procedure as follows: starting from an active cell v', we search for the 4connected neighbors of v' and link v' with them. For those non-empty neighbors, the same procedure is repeated until at least one orientated cell has been included into the graph. The advancing front \mathcal{F} will roughly trace along the locus of the underlying surface until it becomes empty (as shown in Figure 2, the third from left).

5. Results and Discussions

All of the examples presented in this paper are generated on an Intel Centrino Duo 1.66 GHz laptop equipped with 1 GB RAM. In addition, we use Bloomenthal's implicit surface polygonizer [3] to extract a triangular mesh of the reconstructed implicit surface for visualization. Table 1 briefly summarizes the computational time of our algorithm. Note that the proposed algorithm can correctly align the MPU implicits into a globally consistent signed distance field very efficiently.

Recall that we only link 4-connected neighbors while constructing a graph for the purpose of robustness. Ideally, all the auxiliary points should be consistently signed. However, there is always a small margin where two local functions take opposite signs (see Figure 4). Considering an edge linking 2-connected neighbors, one mismatch of the auxiliary points will cause this edge to be *ambiguous*, because any label assignments yield the same smoothness energy. An edge becomes *invalid* once most of the auxiliary points are mismatched, which will cause wrong label assignment. An ambiguous edge poses no harm to the al-



Figure 5. An example of reconstructing MPU implicits from non-uniformly sampled data set.

gorithm because the orientations of its endpoints can still be determined by other *valid* edges. For 1-connected neighbors, the edges can only be valid or invalid, thus they should be avoided. In our experiments, performing BP inference on graphs consisting only 4-connected edges works well for all test models.

Figure 5 and 6 demonstrate the effectiveness of the proposed algorithm on different types of data sources. Figure 7 is an illustration on controlling the level of detail by evaluating the MPU implicits at different octree levels. The MPU implicits representation is capable of reconstructing surfaces from non-uniformly sampled data. Even though noises are presented in the data, the least-squares fitting process can still produce acceptable results. Since the BP inference algorithm works by directly checking the consistency between individual distance fields, it is not affected by topology or sampling density, which might be difficult for traditional MST-based method [11]. Nevertheless, the proposed algorithm still might fail in highly disordered distance fields because the BP inference algorithm works under the assumption that neighboring local implicits compose a continuous manifold. When dealing with noisy input data, a noise reduction algorithm, such as [30], should be incorporated to alleviate the effect due to noise.

6. Conclusions

In this paper, we introduced a general surface reconstruction algorithm capable of reconstructing implicit surface from an unorganized point clouds based on the MPU



Figure 6. Left: reconstructed surface from input data with 0.2% random noises (Middle), Right: reconstructed surface from clean data.



Figure 7. Level of detail control. Left: octree level 5, Right: octree level 7

implicits [15]. The basic idea behind this work is to establish an isolated stage of orientation inference in the surface reconstruction pipeline. The proposed orientation inference algorithm resorts to the Belief Propagation technique to solve a graph optimization problem by directly examining the consistency between individual signed distance fields. As a result, it is easy to adapt to a wide variety of implicit surfaces and data sources. In the future works, we plan to keep improving our surface reconstruction algorithm to preserve sharp features from unorganized point clouds and robustness against noisy input data. It is interesting to note that the graph cuts algorithm [13] is essentially applicable to the binary optimization problem in this work. We also plan to conduct some experiments to compare the performance of these two algorithms.

Acknowledgements

The authors would like to express their sincere appreciation to Dr. Xiang Zhang for reviewing this paper. The test models are from Stanford 3D scanning repository (Bunny and Armadillo) and Cyberware (Igea, Rabbit, Isis), respectively. This work was supported by National Science Council, Taiwan, under the grant NSC- 95-2221-E-007-224.

References

- M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. *In Proceedings of IEEE Visualization*, pages 21–28, October 2001.
- [2] J. F. Blinn. A generalization of algebraic surface drawing. ACM Transactions on Graphics, 1(3):235–256, July 1982.
- [3] J. Bloomenthal. An implicit surface polygonizer. *Graphics Gems IV*, pages 324–349, 1994.
- [4] J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann, 1997.
- [5] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. *In Proceedings of ACM SIGGRAPH*, pages 67–76, August 2001.
- [6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *In Proceedings of ACM SIGGRAPH*, pages 303–312, August 1996.
- [7] H. Q. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces using anisotropic basis functions. *In International Conference on Computer Vision (ICCV)*, pages 606–613, July 2001.
- [8] H. Q. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10):1358–1371, October 2002.
- [9] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *In Proceedings of* ACM SIGGRAPH, pages 544–552, 2005.
- [10] R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *International Journal of Numerical Methods in Engineering*, 15:1691–1704, 1980.
- [11] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *In Proceedings of ACM SIGGRAPH*, pages 71–78, July 1992.
- [12] L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel. Feature sensitive surface extraction from volume data. *In Proceedings of ACM SIGGRAPH*, pages 57–66, 2001.
- [13] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.
- [14] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K. R. Subramanian. Interpolating implicit surfaces from scattered data using compactly supported radial basis functions. *In Shape Modeling International 2001*, pages 89–98, May 2001.

- [15] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *In Proceedings of ACM SIGGRAPH*, pages 463–470, July 2003.
- [16] Y. Ohtake, A. G. Belyaev, and H.-P. Seidel. A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. *In Shape Modeling International* 2003, pages 153–161, May 2003.
- [17] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. *In Proceedings of IEEE Visualization*, pages 163–170, October 2002.
- [18] M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled surfaces. *In Proceedings of EU-ROGRAPHICS*, 22(3), 2003.
- [19] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California, 1988.
- [20] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 23rd ACM national conference*, pages 517–524, 1968.
- [21] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. *In Proceedings of ACM SIG-GRAPH*, 24(3):861–868, July 2005.
- [22] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(13):1115–1138, 1991.
- [23] G. Taubin. An improved algorithm for algebraic curve and surface fitting. In International Conference on Computer Vision (ICCV), pages 658–665, May 1993.
- [24] I. Tobor, P. Reuter, and C. Schlick. Multi-scale reconstruction of implicit surfaces with attributes from large unorganized point sets. *In Shape Modeling International 2004*, pages 19–30, 2004.
- [25] G. Turk and M. Levoy. Zippered polygon meshes from range images. *In Proceedings of ACM SIGGRAPH*, pages 311– 318, July 1994.
- [26] G. Turk and J. F. O'Brien. Shape transformation using variational implicit surfaces. *In Proceedings of ACM SIG-GRAPH*, pages 335–342, August 1999.
- [27] G. Turk and J. F. O'Brien. Modelling with implicit surfaces that interpolate. ACM Transactions on Graphics, 21(4):855– 873, October 2002.
- [28] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. *In International Conference on Computer Vision (ICCV)*, pages 936–943, October 2005.
- [29] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):723–735, 2001.
- [30] H. Xie, K. T. McDonnel, and H. Qin. Surface reconstruction of noisy and defective data sets. *In Proceedings of IEEE Visualization*, pages 259–266, 2004.