Iterative Consolidation of Unorganized Point Clouds

Shengjun Liu, Kwan-Chung Chan, and Charlie C.L. Wang, *Member, IEEE*

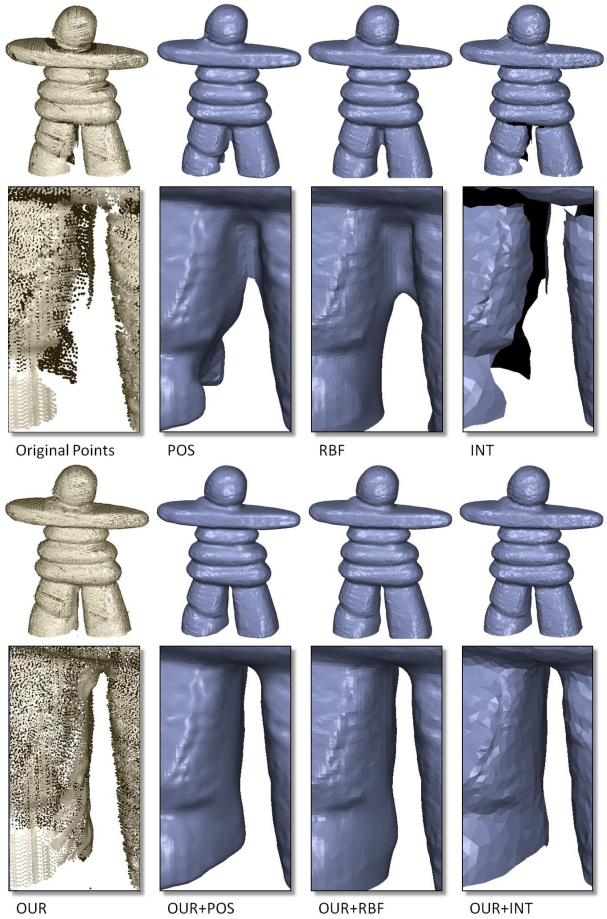
***Abstract*— Unorganized point clouds obtained from 3D shape acquisition devices usually present noises, outliers, and non-uniformities. In this article, we propose a framework to consolidate unorganized points by an iterative procedure of interlaced down-sampling and up-sampling steps. After down-sampling and up-sampling, selection operations are conducted to remove outliers while preserving geometric details. The uniformity of points is improved by moving the down-sampled particles and the following refinement of point samples, and the missed regions are filled through surface extrapolation. Moreover, an adaptive sampling strategy is employed to speed up the iterations. Experimental results demonstrate the effectiveness of the proposed point processing framework.**

***Index Terms*— Unorganized point clouds, repulsion operator, outlier removal, down-sampling, up-sampling.**

I. Introduction

Three-dimensional scanning devices are commonplace nowadays; therefore the shapes obtained by acquisition devices have become a major source for the generation of complex digital 3D models. Although optical scanners are the most economical and efficient acquisition devices to obtain the 3D digital model from a real object, they always produce incomplete and noisy point clouds due to occlusions and physical limitations of the scanners. In the regions that are invisible to the cameras (e.g., deep cavities and bifurcations), the surface of the scanned model is not covered by sample points. The under-sampled or completely missed regions on the scanned point cloud of a real-world geometry will lead to an imperfect shape on the surface reconstructed by most reconstruction algorithms (see the top row of Fig.1).

We propose a point cloud processing framework to improve the quality of point clouds and thus improve the quality of reconstructed surfaces. The input of our approach is an unorganized point cloud which may contain outliers, noises and non-uniformities. Based on the point positions alone, we focus on how to make points evenly distributed by inserting samples into sparse regions using the down-sampling, up-sampling and selection mechanism. A new point processing framework is proposed. Outliers can be removed by a mean-shift based particle section operator. As the consequence of applying a novel particle repulsion operator in the framework, the missed region on the given point set will be extrapolated by the newly inserted sample points. The resultant point cloud processed by OUR and the surfaces reconstructed by POS, RBF and INT are shown in the bottom row of Fig.1. Benefited from the new selection and repulsion operators, the quality of points is incrementally improved under this framework. Figure 2 shows the progressive results with vs. without using the repulsion operator. It is easy to find that simply applying down-sampling and up-sampling in the iterations does not complete the missed regions on the given point cloud. The improvement comes from the newly proposed repulsion operator (see Section IV-C). As will be shown in



POS	Poisson method of surface reconstruction [2]
RBF	Radial Basis Function based surface reconstruction [3]
INT	Integrating meshing method [4]
CON	Consolidation of unorganized point clouds [5]
OUR	Our iterative consolidation approach

Fig. 1. (Top Row) The point cloud of an Inukshuk model obtained from the 3D scanner with incomplete sample points. The reconstructed surfaces generated by various algorithms in literature are poor at the regions with imperfect input samples. (Bottom Row) The quality of surfaces reconstructed by various methods from the point cloud processed by our approach are all improved. The oriented normal vectors are generated by [1] for the approaches that need consistently oriented normal vectors.

Manuscript submitted in August 2010; revision prepared in December 2010.

All authors are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong. Corresponding Author: Charlie C.L. Wang (Tel: (852) 2609 8052; Fax: (852) 2603 6002; E-mail: cwang@mae.cuhk.edu.hk).

Shengjun Liu is now with the School of Mathematical Science and Computing Technology, Central South University, China. This work was completed when he was with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong.

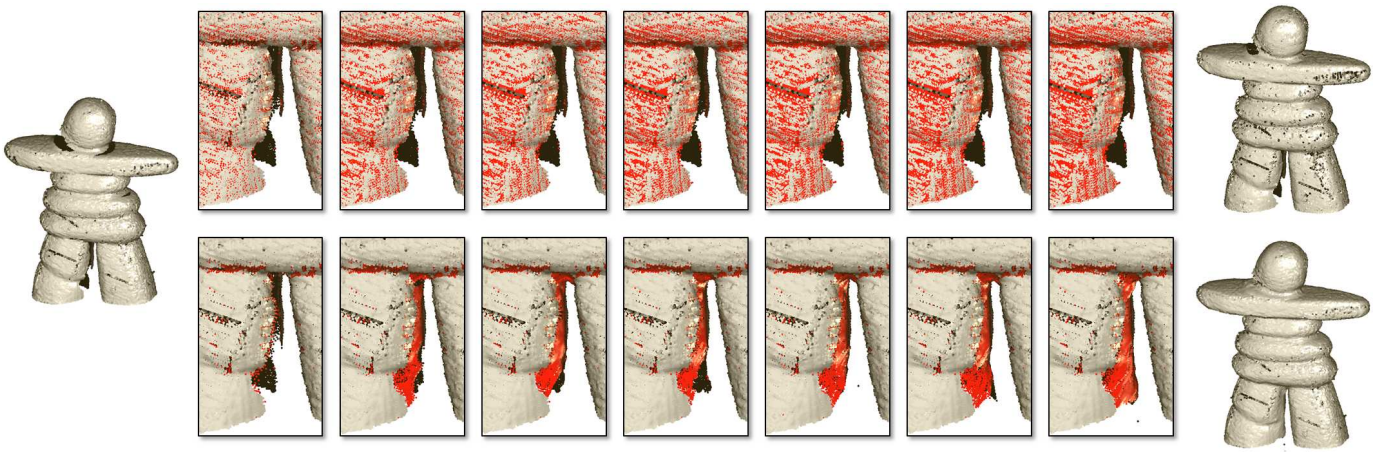


Fig. 2. The progressive results without (top row) vs. with (bottom row) using the repulsion operator. Simply down-sampling and up-sampling the points without the newly proposed repulsion operator cannot fill samples into the missed regions on given point clouds (see the top row). On the contrary, our framework can progressively improve the quality of the point cloud (see the bottom row). The up-sampled points are displayed in red.

Section IV-B, the outliers can be effectively removed with the help of a mean-shift based selection operator. Geometric details are preserved in our algorithm, which is benefitted by a selection operator for up-sampled points (see Section IV-D). As shown in Fig.3, our iterative consolidation framework generates results outperforming the non-iterative consolidation approach (CON) [5].

The main contribution of our approach is threefold:

1) an iterative framework for point cloud processing to improve the quality of point clouds while preserving the geometric details,
2) a new selection algorithm in the framework for removing outliers,
3) a novel repulsion operator for extrapolating the consolidated points therefore completing the point sets at the missed regions with large areas.

As a result, an effective pipeline for consolidating unorganized point clouds is developed.

The rests of the paper are organized as follows. After reviewing the related work in Section II, the overview of our point processing algorithm is presented in Section III. The novel operators are detailed in Section IV. The experimental results are shown and discussed in Section V. Lastly, our paper ends with the conclusion section.

II. Related Work

The problem of reconstructing a surface from points has been investigated for many years and the reconstruction methods have become a standard manner of geometry creation. A variety of techniques have been developed. However, the points acquired by scanners are typically incomplete and highly non-uniform. Some related approaches concerning the reconstruction of a surface from inhomogeneous sample density or missing data are reviewed below.

Ohtake et al. [3] and Carr et al. [6] exploited the extrapolation properties of radial basis functions to fill regions of sparse sampling. The work of Savchenko and Kojekine [7] warps a given surface model towards the missing region of the given surface using control points. This is followed by a fairing step along the boundary of the hole. This method is not automatic. It requires some manual interventions, and a prior model must be given in advance. Verdera et al. [8] also used an implicit function to represent the surface. They modelled a PDE for the smooth interpolation of a given hole based on the normal vector field around it. In [9] a surface is repaired by an optimization process. It minimizes the integral of the squared mean curvature to yield a smooth surface. Weyrich et al. [10] extended the volumetric diffusion method proposed by Davis et al. [11] to point-sampled models by replacing the distance estimation with a moving least square projection step. These methods are successful in repairing small deficiencies in the data, but have difficulties with complex holes or when large parts of the object are missing (e.g., the human model in Fig.8).

The method of Kolluri et al. [12] requires filtering of the Voronoi diagram to obtain a correct pole graph. To compute a watertight surface, they used global normalized cuts that smoothly complete large missing parts. The surface synthesis methods [13], [14] complete missing parts in the surface by integrating patches which are taken from a well annotated shape database or a given example set. If no appropriate examples exist, the result might be poor and the process might fail. The method of Hornung and Kobbelt [15] requires the definition of a watertight voxel crust in which the unknown surface is supposed to lie. To complete the crust, the authors used flood-fill and dilation operators. Sharf et al. [16] evolved an explicit mesh in a scalar field guided by the local feature size in a coarse to fine manner to avoid local minima and capture details. The method also requires a volumetric grid to evaluate the distance transformation, and the topological change has to be tracked. The computational implementation can be quite intricate (especially the topology variation on the two-manifold mesh surfaces). In [17], Sharf et al. interactively reconstructed the surface using only the positions of raw scanned data, where the user defines the general in/out orientation and assists the interpretation of data in automatically detected topologically unstable regions.

Fig. 3. The geometric details are removed by the consolidation approach [5] – CON+POS, and the reconstruction of CON+POS at the crotch (where sample points are missed) merges two legs. Neither of these occurs in OUR+POS.

Different from the mesh surface reconstruction algorithms, we focus on how to improve the quality of a given unorganized point cloud by adding sample points to change the uniformity of scattered points and removing outliers. After iteratively consolidating the given unorganized point cloud, a high-quality surface can be reconstructed from those points by various methods.

III. Overview of Algorithm

Given an unorganized set $P = \{\mathbf{p}_j\} \subset \Re^3$ presenting noises, outliers and non-uniformities, surface reconstruction from this data may cause significant misinterpretation of the data that leads to an erroneous surface. Our method presented in this paper aims to recover the structural information of P without losing the geometric details by inserting points in sparse regions to make the points evenly distributed, adding points into the missed regions with large area, and removing outliers that are far away from the up-sampled surface. Such a 'massage' procedure of point clouds is called *consolidation* [18]. Our consolidation method consists of several steps: down-sampling, outlier removal, repulsion, up-sampling and selection, which are iteratively applied to the input point set P.

Our point processing method is inspired by an image completion approach based on multi-resolution techniques [19]. Their method is based on the observation that the lower resolution representation of an image contains stronger structural information while the higher resolution representation contains more details. Therefore, the structural information are recovered at the lower resolution. The structural and non-structural information on the 3D models represented by a set of sample points is analogous. In our approach, the points in P are first down-sampled into k particles, which are redistributed on the surface defined by the samples in P. The redistribution of the particles are performed by iteratively applying the *Weighted Locally Optimal Projection* (WLOP) operator in [5]. The outliers are removed by applying a selection operator to the particles. The oriented normals of particles can be estimated on the cleaned particles by the method in [1] or [5], and a new repulsion operator based on the *Algebraic Point Set Surface* (APSS) [20] is then applied to extrapolate the surface by pushing particles into the missed regions. After that, the redistributed particles are refined into a smooth point set surface by a $\sqrt{3}$-like interpolatory refinement scheme [21] – this is an up-sampling step. The newly generated sample points are selectively merged into the given point set P, while the points in P are considered as outliers and removed if they are far away from the up-sampled points (i.e., a smooth surface interpolating the redistributed particles). The down-sampling, repulsion, up-sampling and selection steps are repeatedly applied to the point set. The iteration stops when only a few new points are inserted into the point set P. Lastly, the orientation of the points in P after the iteration can be obtained by their closest particle in X, and the consistently oriented normal vectors can be obtained by the orientation-aware *Principal Component Analysis* (PCA).

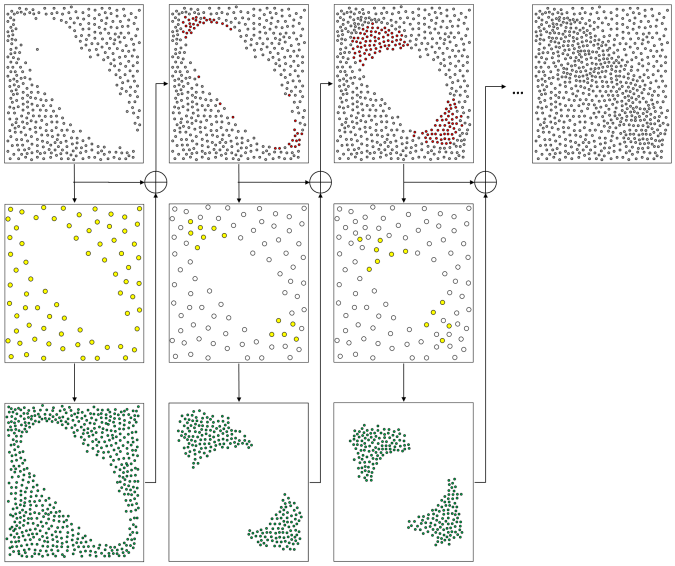


Fig. 4. Illustration of our point processing framework, where the given unorganized points (in white small dots) are first down-sampled into particles (in yellow) and redistributed, and then up-sampled into a dense point set (in green small dots). Among the points generated by the up-sampling, the ones falling in the regions that are lack of samples in the given point set are selected (shown in red small dots) and retained. In the next iteration, the points retained from up-sampling are down-sampled into new particles (yellow ones in the second and the third columns), redistributed, and up-sampled into new points (the green dots in the pictures of the last row). The iteration repeats until only a few up-sampled points are added.

Nevertheless, the repeated application of down-sampling and up-sampling to the whole set of point samples in the framework proposed above wastes a lot of time on the regions that have been processed in the previous iteration steps. To reduce the redundant computations, an adaptive framework is investigated and used here. As illustrated in Fig.4, we only down-sample the newly added points into particles while retaining the particles used in the previous iterations, and only the newly added particles are up-sampled into new points. Specifically, the pseudo-code of our point processing algorithm is listed in **Algorithm** Iteration-Consolidation. In this adaptive framework, the points/particles that are processed in the previous iterations will not be further processed so that a lot of computational redundancies are removed. The speedup compared with the primary implementation introduced in the above paragraph is about 3-5 times. We employed a hybrid terminal condition for the iteration: 1) $\frac{|P^{i-1}|-|P^i|}{|P^{i-1}|} < 20\%$ or 2) more than ten iterations have been conducted.

Algorithm 1 Iterative-Consolidation

$P^0 \Leftarrow P$ and $i \Leftarrow 0$;
Initialize a particle set X by down-sampling all points of P into m particles;
repeat
  $X \Leftarrow X \cup X^i$;
  Repeatedly move the particles in X^i by the WLOP operator[1] (section IV-A);
  if $i = 0$ **then**
    Remove the outlier particles from X^0 by a mean-shift based selection operation (section IV-B);
  end if
  Estimate the orientation of particles by [1] or [5];
  Apply the repulsion operator based on APSS to all particles in X^i (section IV-C);
  Refine the points in X^i into a set of up-sampled points Υ^i (section IV-D);
  if $i = 0$ **then**
    Remove the outliers in P according to Υ^0 (details can be found in section IV-B);
  end if
  Select the points of Υ^i into a subset P^{i+1} (section IV-D);
  $P \Leftarrow P \cup P^{i+1}$ and $i \Leftarrow i + 1$;
  Down-sample all points of P^i into X^i with m^i particles ($m^i = 2m|P^i|/|P^0|$ with $|\cdots|$ being the number of points);
until the terminal condition is reached
Estimate the consistently oriented normals on the sample points in P by [1] or [5];
return P;

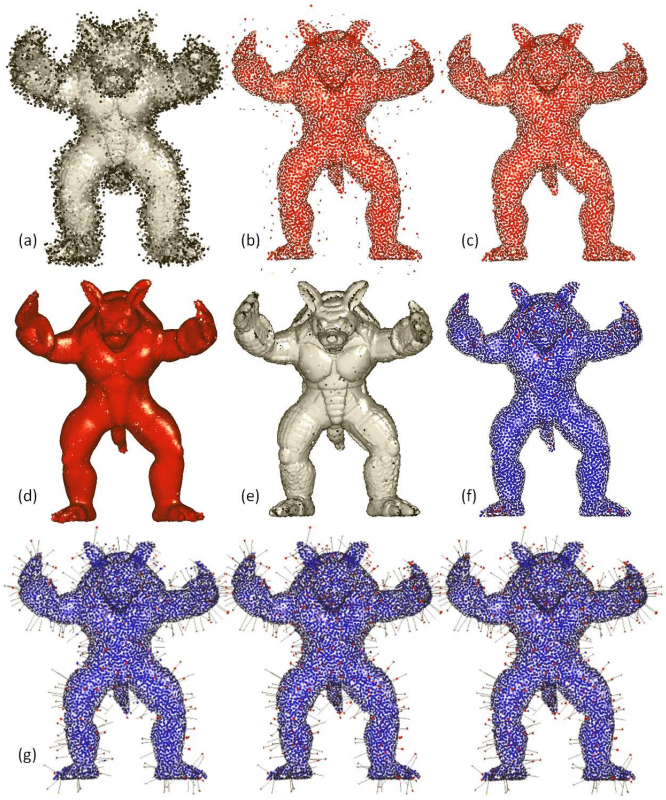


Fig. 5. An example to demonstrate the outlier removal on the Armadillo model. (a) The given point cloud has 20% noisy points randomly distributed in the range of $1/50$ of the bounding box's diagonal length. (b) The resultant particles after applying the WLOP based down-sampling and relaxation. (c) Our outlier selection operator can successfully identify the outliers and remove them. (d) The up-sampling result generated from the cleaned particles with less geometric details. (e) The resultant point set after removing outliers in sample points and merging the up-sampled points. (f) The particles to be used in the second iteration, where the blue ones are static particles and only the red one are alive particles sampled from the newly inserted points. (g) The mean-shifts on particles are illustrated by line segments – from left to right, the results after one, three and five mean-shift iterations respectively, and the identified outliers are displayed in red.

IV. Point Processing Operators

This section presents the technical details of the operators used in the iterative consolidation framework, which include a voting based particle selection operator for outlier removal, a new repulsion operator, and the selection operators for up-sampled points. To be self-contained, the down-sampling operator based on WLOP [5] and the up-sampling operator based the refinement method in [20] are also briefed.

A. Down-sampling and relaxation

For the given point set P^i, we randomly select m^i points to form a set X^i. The points in X^i are called *particles*, and m^i is selected as $m^i = \frac{2m|P^i|}{|P^0|}$ where m is a user parameter – we usually choose $m = \frac{1}{20}|P^0| \sim \frac{1}{5}|P^0|$. These m particles are then iteratively moved to a new position which is defined by two terms, where the first term attracts the particle to the given point set and the second term repulses the particles away from other particles. Details can be found in Appendix I.

A uniform support size $h = 2L_{avg}$ is adopted for the computation of particle movement with L_{avg} being the average distance between particles to their k-nearest neighboring particles, therefore the computation is adaptive to the scale of models. We choose $k = 20$ to balance the speed and the robustness. When $i \neq 0$, only the particles in X^i are moved. All other particles in X are involved in the computation but with their positions fixed. Another parameter used in this down-sampling and relaxation step is the number of particles, m. As studied in [5], using too few particles could easily damage the existing small features on the given point set, while a too large value of m will slow down the computation the relaxation as well as the following repulsion of particles. Therefore, we suggest to use $m = \frac{1}{20}|P^0|$ for processing a dense point cloud and $m = \frac{1}{10}|P^0|$ for relative sparse points.

B. Outlier removal

Although the WLOP operator can efficiently filter out the noises by the robust down-sampling and relaxation, it however cannot remove outliers that are far away from the real surface of a model (e.g., see Fig.5(b)). These unwanted particles will potentially affect the quality of the consolidation in the downstream operations and is considered to be removed. According to the nature that these outliers are usually far away from the majority of their neighbors, we propose a mean-shift based method that can detect and remove outlier particles

Algorithm 2 Outlier-Particle-Removal

for all $\mathbf{x}_i \in X$ **do**
  Initialize $\mathbf{x}'_i \Leftarrow \mathbf{x}_i$
  for $j = 1$ to b **do**
    Search in X the k-nearest neighbors of $\mathbf{x}'_i$ and let them be the subset K;
    $\mathbf{x}'_i \Leftarrow \frac{1}{|K|}\sum_{\forall \mathbf{k}_i \in K} \mathbf{k}_i$;{Mean-shift step}
  end for
end for
for all $\mathbf{x}_i \in X$ **do**
  Search in X the k-nearest neighbors of $\mathbf{x}'_i$ and let them be the subset K;
  $\bar{d}_{\mathbf{x}'_i} \Leftarrow \sum_{\forall \mathbf{k}_i \in K} \frac{\|\mathbf{k}_i - \mathbf{x}'_i\|}{|K|}$;{The average distance}
  if $\|\mathbf{x}'_i - \mathbf{x}_i\| > s\bar{d}_{\mathbf{x}'_i}$ **then**
    $\mathbf{x}_i$ is considered as an outlier particle;
  end if
end for
Remove all outlier particles from X;
return X;

during the first iteration of the consolidation.

For each particle $\mathbf{x}$ in X, we iteratively shift it to the average position of its k-nearest neighboring particles ($k = 20$). Then, the distance from the shifted mean position to the original position of $\mathbf{x}$ is compared with the average distance between its shifted neighbors. If the difference between two distances is significant, $\mathbf{x}$ is considered as an outlier particle. Pseudo-code of the mean-shift based outlier removal algorithm can be found in **Algorithm** Outlier-Particle-Removal, by which the outlier particles are removed from X. This outlier removal step is only performed in the first iteration of our point processing algorithm. Noticed that the average distance between shifted particles is computed locally so that it is adaptive to the non-uniform distribution of particles. Based on our experimental tests, choosing $b = 3$ and $s = 3$ gives a good balance between speed and quality. Figure 5 gives an example about outlier removal, where Fig.5(c) shows the result of removing outlier particles. Also, it is found from Fig.5(g) that the mean-shift iteration converges very fast. Therefore, using a larger b may not change the result of outlier identification or removal.

Simply deleting the outlier particles does not remove the outliers from the input point set P successfully since the outliers is only a small subset of the points in P. A simple selection step can be performed to remove outliers from P. After obtaining a set of cleaned particles and then up-sampling them back into points in Υ^0 as the samples of a smooth surface, the points in P which are far away from the surface represented by Υ^0 are considered as outliers. Therefore, we have:

- $\forall \mathbf{p}_i \in P$, it will be removed from P when $\|\mathbf{p}_i - \mathbf{q}_j\| > h$ $(\forall \mathbf{q}_j \in \Upsilon^0)$.

This simple selection operation can effectively remove outliers embedded in the given point cloud (see Fig.5(e) for an example).

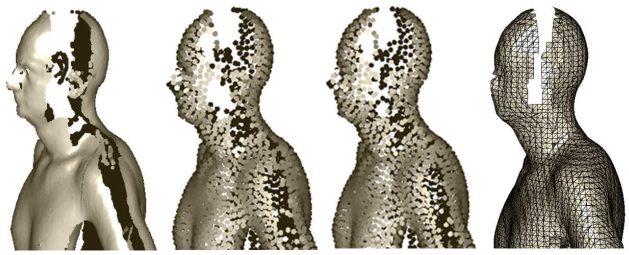


Fig. 6. The illustration of repulsion operation: (left) the given points for a human body with large regions missed, (middle-left) the redistributed particles after applying WLOP, (middle-right) the redistributed particles after applying the repulsion operator, and (right) the mesh surface tessellated from APSS where no triangle is generated in the undefined regions. Notice that one step of repulsion can fill up some missed regions (e.g., the arms); however, more iterations are needed to fill the missed regions with large area (e.g., the side of head). The final result can be found in Fig.8.

C. Repulsion of particles

The WLOP operator can evenly redistribute the particles along the surface defined by the given point set P. However, the movement of particles driven by WLOP stops at the boundary of large missed regions (e.g., the missed region on thighs in Fig.2). The same observation has also been reported in [5]. To overcome this drawback of WLOP, we introduce a repulsion operator below to move the particles into the missed regions. The observation shows that the missed regions always have their normals nearly perpendicular to the viewing direction of scanners. Smooth extension of the known regions that have been sampled could be a good heuristic to fill the missed regions. In other words, we need to extrapolate the surface defined by the existing particles along its tangential direction.

As the particles in X after the WLOP based relaxation and the mean-shift based outlier removal have become more uniform and less noisy, the consistently oriented normal vectors can be estimated on the particles by the method in [5] or [1]. Equipped with the oriented normals, an *Algebraic Point Set Surface* (APSS) [20] can be defined by the particles $\forall \mathbf{x}_i \in X$ and their normal vectors $\mathbf{n}_{\mathbf{x}_i}$. APSS is a kind of *Moving Least-square Surface* (MLS). Instead of plane fitting, APSS directly fits higher order algebraic spheres. The advantage is that APSS yields more stable results than planar MLS at the regions with high curvature (i.e., thin and sharp features). Details about the APSS fitting on $\forall \mathbf{x}_i \in X$ and $\mathbf{n}_{\mathbf{x}_i}$ can be found in Appendix II.

By the APSS defined on $\forall \mathbf{x}_i \in X$, we can then move the particles along the APSS in repulsive manner. The movement consists of two components: the tangential component and the projective component. The tangential component of a particle $\mathbf{x}_i$ is determined by rotating along an axis $\mathbf{r}_{\mathbf{x}_i}$ that passes through the center $\mathbf{c}_{\mathbf{x}_i}$ of the algebraic sphere corresponding to $\mathbf{x}_i$.

- The tangential component of repulsion is derived from the second term of WLOP as

$$\mathbf{l} = \mu \sum_{\mathbf{x}_p \in (X \setminus \{\mathbf{x}_i\})} (\mathbf{x}_i - \mathbf{x}_p) \frac{w_p \varrho}{\sum_{\mathbf{x}_p \in (X \setminus \{\mathbf{x}_i\})} w_p \varrho} - \mathbf{x}_i \quad (1)$$

with $\varrho = \theta(\|\mathbf{x}_i - \mathbf{x}_p\|)/\|\mathbf{x}_i - \mathbf{x}_p\|$. Generally speaking, $\mathbf{l}$

is not perpendicular to the normal vector $\mathbf{n}_{\mathbf{x}_i}$ at $\mathbf{x}_i$. We thus compute the corresponding orthogonal vector to $\mathbf{n}_{\mathbf{x}_i}$ by

$$\mathbf{l}' = \mathbf{l} - (\mathbf{l} \cdot \mathbf{n}_{\mathbf{x}_i})\mathbf{n}_{\mathbf{x}_i}. \quad (2)$$

- The rotation axis of the tangential component is

$$\mathbf{r}_{\mathbf{x}_i} = \frac{(\mathbf{x}_i - \mathbf{c}_{\mathbf{x}_i}) \times \mathbf{l}'}{\|(\mathbf{x}_i - \mathbf{c}_{\mathbf{x}_i}) \times \mathbf{l}'\|}, \quad (3)$$

and the rotation angle is obtained by

$$\varpi = \frac{\|\mathbf{l}'\|}{2\pi\|\mathbf{x}_i - \mathbf{c}_{\mathbf{x}_i}\|}.$$

The tangential component of the movement is then defined by

$$\mathbf{R}_{\mathbf{r}_{\mathbf{x}_i}}(\varpi)(\mathbf{x}_i - \mathbf{c}_{\mathbf{x}_i}) + \mathbf{c}_{\mathbf{x}_i} - \mathbf{x}_i, \quad (4)$$

where $\mathbf{R}_{\mathbf{r}_{\mathbf{x}_i}}(\varpi)$ is the rotation matrix [22] around the axis $\mathbf{r}_{\mathbf{x}_i}$. In order to improve the stability of the particle movement, we restrict the rotation angle by

$$\varpi = \min\{\frac{\|\mathbf{l}'\|}{2\pi\|\mathbf{x}_i - \mathbf{c}_{\mathbf{x}_i}\|}, \frac{L_{avg}}{2\pi\|\mathbf{x}_i - \mathbf{c}_{\mathbf{x}_i}\|}, \frac{\pi}{4}\}, \quad (5)$$

where L_{avg} is the average distance between particles and their k-nearest neighboring particles (with $k = 20$).

After applying the tangential component on a particle $\mathbf{x}_i$, we consecutively apply the following projective component on it for three times to retain the moved particles on the APSS defined by the particles equipped with normal vectors.

$$\mathbf{c}_{\mathbf{x}_i} + r(\mathbf{x}_i)\frac{\mathbf{x}_i - \mathbf{c}_{\mathbf{x}_i}}{\|\mathbf{x}_i - \mathbf{c}_{\mathbf{x}_i}\|} - \mathbf{x}_i \quad (6)$$

The resultant particles obtained by our repulsion operator will distribute into the missed regions. It needs to be noticed that applying this repulsion only once cannot fill up the whole missed region with large area (as shown in Fig.6). This is because that the APSS defined by the particles before repulsion is not fully defined in the missed region (which can also be proved by the tessellation result in the right of Fig.6) – in other words, the regions need to be filled iteratively.

D. Up-sampling and selection

An up-sampling step is conducted to generate more sample points on the surface that interpolates the particles in X^i as well as the normal vectors on the particles. The up-sampled point set Υ^k is expected to regularize the scattered samples and converge on a smooth surface interpolating the particles and their normals. A good candidate up-sampling scheme that satisfies these requirements is the interpolatory refinement method presented in [20], which is therefore used in our framework for generating Υ^k from X^i.

To fit in with our adaptive down-sampling/up-sampling strategy, only the 'alive' particles (i.e., particles that can be moved in the repulsion) and the points up-sampled from the 'alive' particles are used as the centers to generate refined points. The refinement at a center is prevented if it is too close to the neighbors (e.g., less than $1/3$ of the support size h of particles used in Section IV-A). The up-sampling is stopped when no refinement on any center is allowed.

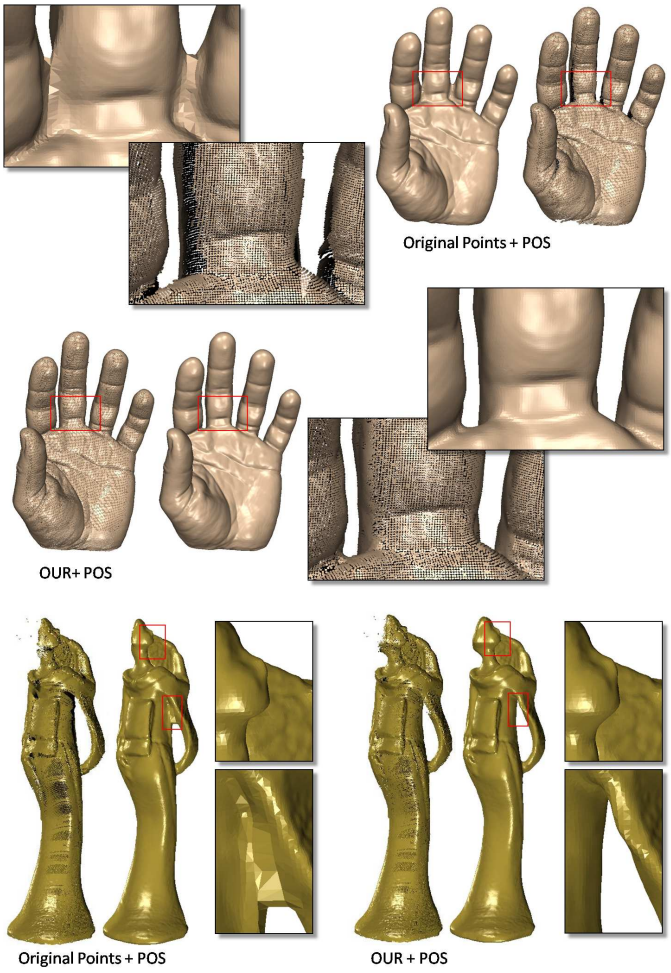


Fig. 7. Results of point processing: (Top) a hand model, where the missed regions on the input cloudy points between fingers are challenging for prior methods but can be recovered by our method, and (bottom) a Japanese lady model, where our method does not change the region that has been well presented in the input point cloud – see the top zoom-views.

In fact, the repulsion operator and the up-sampling operator work together to generate the sample points on a smooth surface extrapolating the particles generated by WLOP. Such an extrapolatory helps to improve the uniformity of sample points in the highly sparse regions. The points in Υ^k are selected to merge into the point set P^k to form a new point set P^{k+1}. In the next iteration of point processing, only the particles down-sampled from these newly inserted points are 'alive' to be moved in the WLOP based relaxation. Therefore, simply add all the points of Υ^k into P^{k+1} leads to too many particles in the later iterations, which will significantly slow down the computation. Adding too few up-sampled points make too few 'alive' particles in the next iteration, which also slow down the extrapolation generated by particle repulsion. We merge Υ^k and P^k by the criterion that:

- $\forall \mathbf{q}_j \in \Upsilon^k$, $\mathbf{q}_j$ should be excluded from P^{k+1} if $\exists \mathbf{p}_i \in P^k$ with $\|\mathbf{q}_j - \mathbf{p}_i\| < \frac{1}{2}L_{avg}^{pnt}$,

where L_{avg}^{pnt} is the average of distances between all samples in P^k to their nearest neighbors. Note that, according to our experimental tests, the quality of processed point sets by using different value for $\frac{1}{2}L_{avg}^{pnt}$ does not have significant differences.

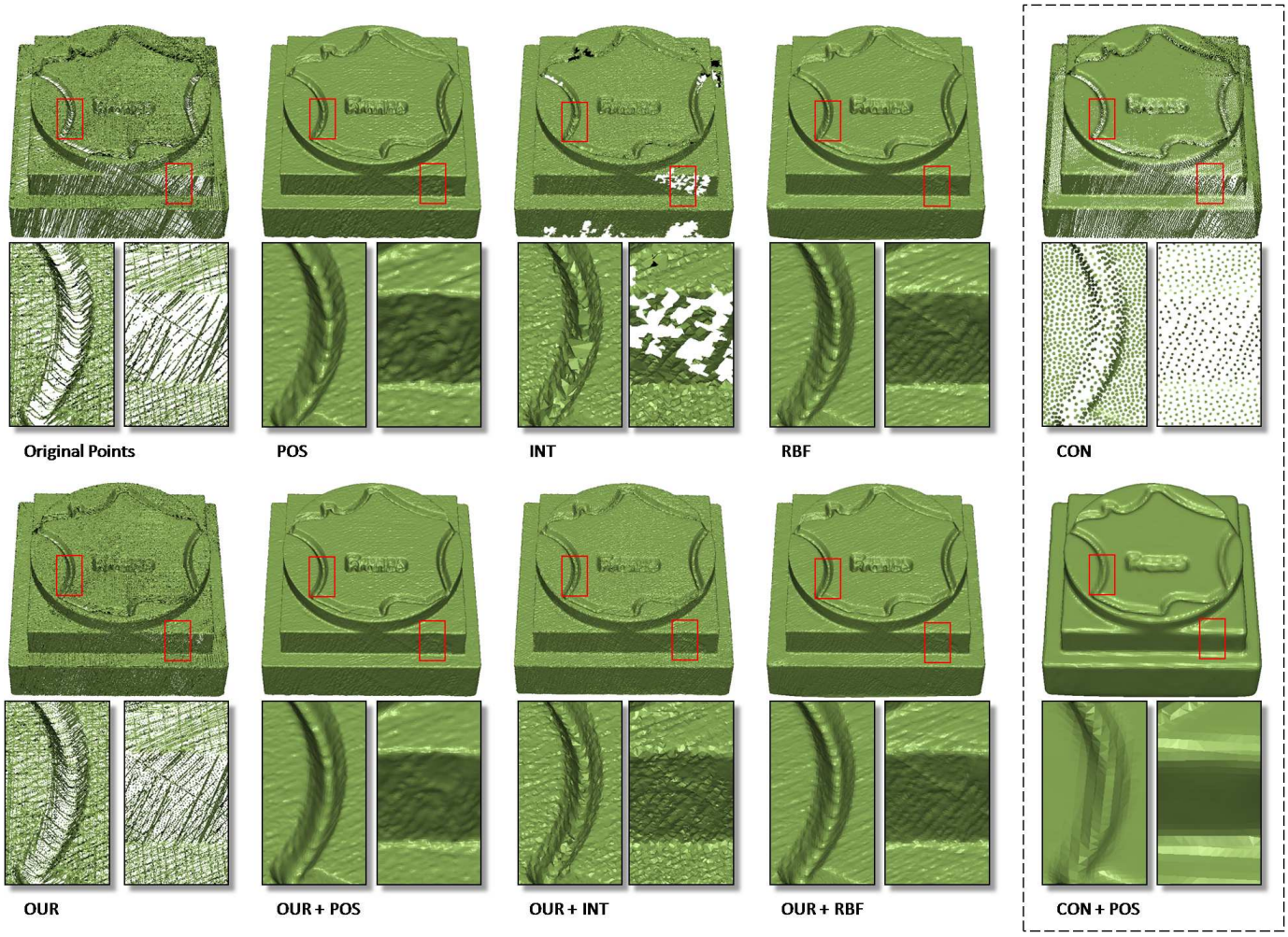


Fig. 9. An example of the seal model. The point cloud processed under our framework results in higher quality surfaces reconstructed by various methods. The geometric details are preserved by our method, while the consolidation method of [5] removes the geometric details.

The value of $\frac{1}{2}L_{avg}^{pnt}$ mainly affects the speed of computation.

V. Results and Discussion

To demonstrate the effectiveness of the proposed framework, we have tested our method on several point clouds which have problems in surface reconstruction according to the non-uniformity of sample points. This defect is usually presented in the raw data obtained from optical scanners. Our experiments exhibit difficulties in reconstructing surfaces from the raw data directly in the highly sparse areas, such as the legs of an Inukshuk model in Fig.1, the parts between two fingers on a hand in Fig.7, the armpit of a Japanese lady model in Fig.7, the miss-scanned regions between the front and the back scans of a human body in Fig.8. For the illustration purpose, we render the unoriented raw scans using the normals computed by the method in [1]. All the examples shown in this paper can be completed in less than ten iteration steps. The computational statistics are shown in Table 1.

For some point data which embeds small holes only, the reconstruction algorithms (e.g., RBF or POS) can generate a surface without being significantly affected by the holes. Different from that, our aim in this paper is to construct a high-quality point set which can recover the highly sparse regions while preserving the geometric details. Based on this expectation, high-quality surfaces are reconstructed from the point clouds processed by our iterative consolidation framework by a variety of surface reconstruction algorithms (see the examples shown in Figs.1 and 9).

TABLE I
Computational Statistics

Model	Fig.	Input Points	Initial Particle #	Iteration Steps	Total Time^{+}
Inukshuk	1	205,858	10,293	6	$3'13''$
Hand	7	194,457	38,892	8	$11'19''$
Lady	7	175,514	35,103	2	$4'26''$
Body*	8	170,346	8,518	6 + 8	$15'11''$
Seal	9	889,076	44,454	2	$6'59''$

$^{+}$ Our prototype program is implemented by Visual C++, and the statistics are resulted from tests running on a standard PC with Intel Core-i5 CPU 750 at 2.67GHz plus 3.46GB RAM.
* To obtain a good result, we conduct the pre-processing on the human body example in two phases – in the first phase, the repulsion operation is not applied and a smaller value for h is used; and in the second phase, a normal procedure is fully applied.

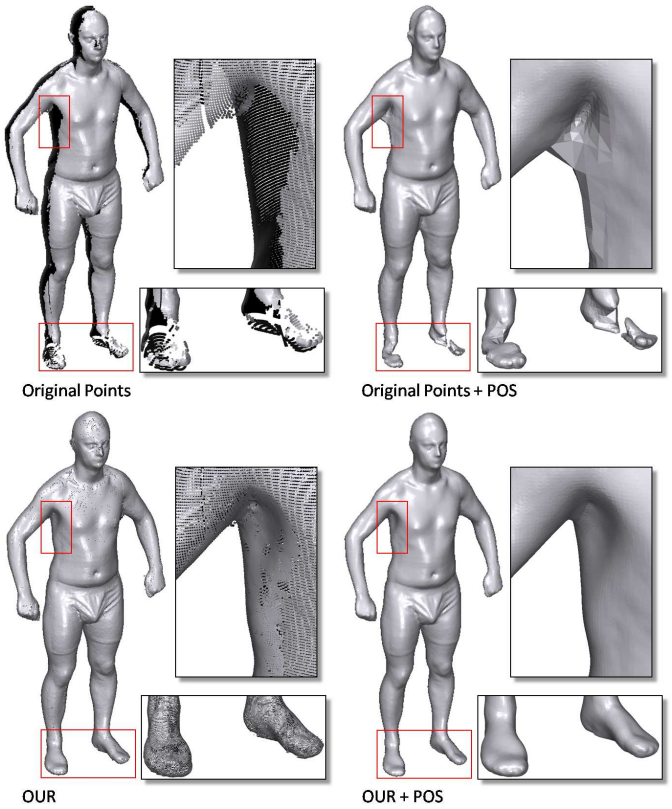


Fig. 8. The quality of the point cloud of a body scan as well as the reconstructed surface is improved by our method in the regions that are miss-scanned – see the armpit and the feet.

Another interesting study is about the robustness of our algorithm to the noises in different levels. As shown in Fig.10, we apply the point processing algorithm proposed in this paper on the Armadillo model embedding 5%, 10%, 15% until 20% noisy points. It is easy to find that the surface reconstruction method will generate mesh models highly affected by the noises without processing the given noisy points. After processing the points by our method, the same surface reconstruction method can produce very smooth model and the models still preserve geometric details on them.

In the pipeline of our point processing framework, oriented normal vectors need to be evaluated on the particles before applying the repulsion operation. The particles have been processed by the WLOP-based relaxation and the outlier removal operation. Therefore, the normal estimation on the retained particles is robust to high frequency noises (removed by the WLOP) and outliers (removed by the mean-shift based selection). Consistently oriented normal vectors can be successfully estimated by different methods (e.g., [1], [5]). Figure 11 compares the results generated by [1] and [5] on a highly noisy skull model, where no significant difference can be found.

A. Limitations

One major limitation of the proposed method is that it has problem to recover sharp/thin features, where the missed regions cannot be repaired by a simple surface extrapolation.

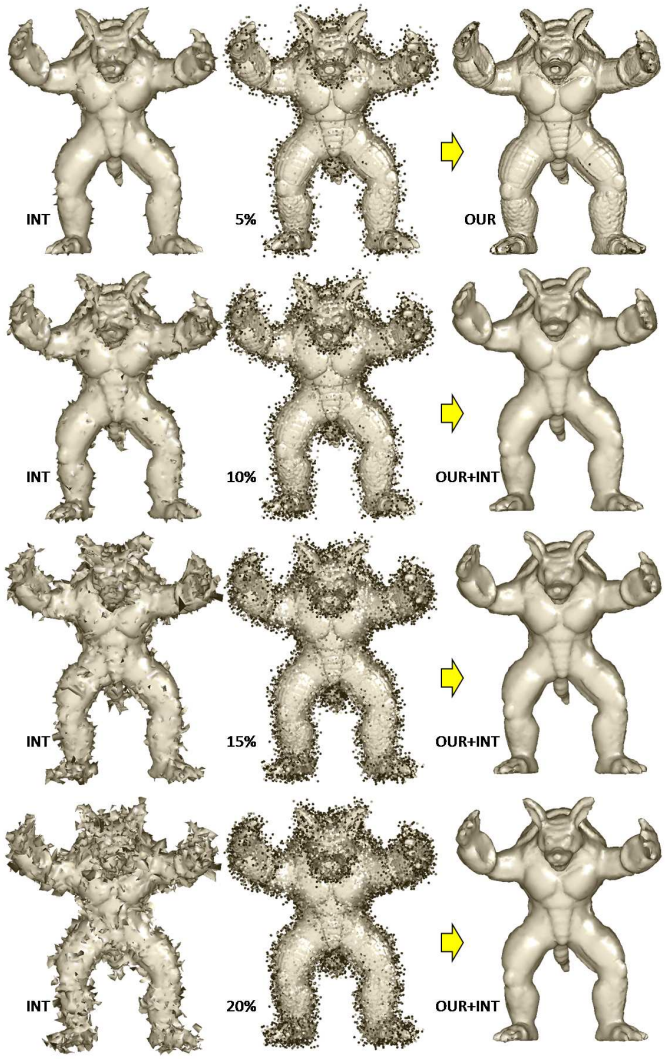


Fig. 10. Example to show the performance of our outlier removal method on the Armadillo model with different amount of noises – noisy points are randomly distributed in the range of 2% of the bounding box's diagonal length. To demonstrate how the noises affect the mesh reconstruction method, the mesh surfaces generated by the integrating meshing method (INT) [4] are also shown.

Although new particles and points are still inserted into such regions, the shape of the thin/sharp features may be destroyed (see Fig.12(b) for an example). However, this problem could be solved by adding a simple interaction. We just need to select such regions on the input point cloud to specify the place where the repulsion of particles must be restricted (see Fig.12(c) for the result of such an interaction).

Another limitation of our approach is the speed. The processing time of a point set with tens of thousand points now is around several minutes, which is far from interactive speed. To solve this problem, one of our near future work is to parallelize the computation proposed in this framework and run it on the accelerated graphics hardware.

VI. Conclusion

We have presented an iterative consolidation framework for processing unorganized point clouds. Our approach comprises

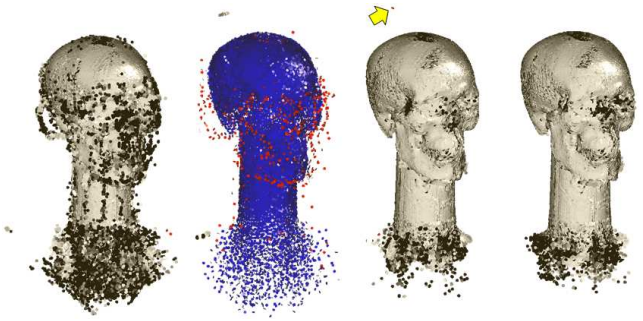


Fig. 11. The outlier removal and normal estimation on a highly noisy skull model. From left to right, the given skull model, the particles after WLOP based relaxation (outliers are specified in the red color), the point samples after one iteration in our framework with particle normals generated by [1], and the point samples after one iteration in our framework with normals estimated by [5]. It is easy to find that using different normal estimation methods does not significantly affect the processing result. Note that the red dot on top of the third skull (pointed by the yellow arrow) is an outlier particle of the second skull model.

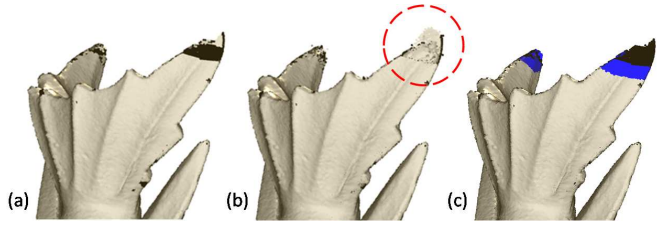


Fig. 12. The extrapolation of point sets generated by the repulsion operator may lead to mismatched surfaces: (a) the given fish model and (b) the particles move out of the model along the tangential direction driven by the repulsion operators (see the particles inside the red circle). The problem can be solved by interactively specifying the region where the particles should be static during the iteration (see the black regions in (c)). The blue ones in (c) are the points inserted by our approach after constraining the movement of particles in the black regions.

the processes of down-sampling, repulsion, up-sampling and selection. The down-sampling step generates a set of points which preserve the structural information of the original points. The up-sampling step creates the linkages between the samples and recovers the structure based on the result obtained from the down-sampling step. The iterative framework of the two steps incrementally recovers the structure of the original data. We have demonstrated the utility of our framework with several examples, and have obtained quite encouraging results.

ACKNOWLEDGMENT

The authors would like to thank the authors of [5] for providing valuable comments on the work of WLOP and sharing the executable program of their approach. This research is supported by the HKSAR RGC/GRF Grant (Ref.: CUHK/417109 and CUHK/417508).

REFERENCES

[1] S. Liu, and C.C.L. Wang, "Orienting unorganized points for surface reconstruction," *Computers & Graphics*, *Special Issue of IEEE International Conference on Shape Modeling and Applications (SMI 2010)*, vol.34, no.3, pp.209-218, June 21-23, 2010.

[2] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," *Proc. of Symposium on Geometry Processing 2006*, pp.61-70.

[3] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "3D scattered data interpolation and approximation with multilevel compactly supported RBFs," *Graphical Models*, vol.67, no.3, pp.150-165, 2005.

[4] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "An integrating approach to meshing scattered point data," *Proc. of ACM Symposium on Solid and Physical Modeling (SPM 2005)*, pp.61-69, 2005.

[5] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM Trans. Graph.*, vol.28, no.5, pp.1-7, Dec. 2009.

[6] 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," *Proc. of ACM SIGGRAPH 2001*, pp.67-76.

[7] V. Savchenko, and N. Kojekine, "An approach to blend surfaces," *Proc. of Computer Graphics International*, pp.139-150, 2002.

[8] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro, "Inpainting surface holes," *Proc. of International Conference on Image Processing*, pp.903-906, 2003.

[9] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu, "A finite element method for surface restoration with smooth boundary conditions," *Computer Aided Geometric Design*, vol.21, no.5, pp.427-455, 2004.

[10] T. Weyrich, M. Pauly, S. Heinzle, R. Keiser, S. Scandella, and M. Gross, "Post-processing of scanned 3d surface data," *Proc. of Symposium on Point-Based Graphics*, pp.85-94, 2004.

[11] J. Davis, S. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," *Proc. of First International Symposium on 3D Data Processing, Visualization, and Transmission*, Italy, June 19-21, 2002.

[12] R. Kolluri, J.R. Shewchuk, and J.F. O'brien, "Spectral surface reconstruction from noisy point clouds," *Proc. of ACM Symposium on Geometry Processing*, pp.11-21, 2004.

[13] A. Sharf, M. Alexa, and D. Cohen-Or, "Context-based surface completion," *ACM Trans. Graph.*, vol.23, no.3, pp.878-887, 2004.

[14] M. Pauly, N.J. Mitra, J. Giesen, M. Gross, and L. Guibas, "Example-based 3D scan completion," *Proc. Symposium on Geometry Processing*, pp.23-32, 2005.

[15] A. Hornung, and L. Kobbelt, "Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information," *Proc. of ACM Symposium on Geometry Processing*, pp.41-50, 2006.

[16] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, and D. Cohen-Or, "Competing fronts for coarse-to-fine surface reconstruction," *Proc. of Eurographics 2006*, pp.389-398.

[17] A. Sharf, T. Lewiner, G. Shklarski, S. Toledo, and D. Cohen-Or, "Interactive topology-aware surface reconstruction," ACM Trans. Graph., vol.26, no.3, article no.43, pp.1-9, 2007.

[18] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C.T. Silva, "Computing and Rendering Point Set Surfaces," IEEE Trans. on Vis. and Comp. Graph., vol.9, no.1, pp.3-15, January 2003.

[19] C.-W. Fang, and J.-J. Lien, "Rapid image completion system using multiresolution patch-based directional and nondirectional approaches," *IEEE Trans. Image Processing*, vol.18, no.12, pp.2769-2779, 2009.

[20] G. Guennebaud, and M. Gross, "Algebraic point set surfaces," *ACM Trans. Graph.*, vol.26, no.3, article no.23, 2007.

[21] G. Guennebaud, L. Barthe, and M. Paulin, "Interpolatory refinement for real-time processing of point-based geometry," *Computer Graphics Forum (Proceedings of Eurographics 2005)*, vol.24, no.3, pp.657-666, 2005.

[22] *Rotation matrix given an axis and an angle*, `http://en.wikipedia.org/wiki/Rotation_matrix`.

[23] M. Zwicker, M. Pauly, O. Knoll, and M. Gross, "Pointshop 3D: an interactive system for point-based surface editing," *ACM Trans. Graph.*, vol.21, no.3, pp.322-329, 2002.

APPENDIX I
POSITION UPDATE IN WEIGHTED LOCALLY OPTIMAL PROJECTION (WLOP)

In WLOP, every particle $\mathbf{x}_i \in X^k$ is moved to a new position by the formula below. The update of position consists of two terms, where the first term attracts the particle to the given point set by the weighted local density

$$v_j = 1 + \sum_{\mathbf{p}_l \in (P \setminus \{\mathbf{p}_j\})} \theta(\|\mathbf{p}_j - \mathbf{p}_l\|)$$

and the second term repulses the particles away from other particles by the density

$$w_p = 1 + \sum_{\mathbf{x}_q \in (X \setminus \{\mathbf{x}_p\})} \theta(\|\mathbf{x}_p - \mathbf{x}_q\|).$$

The updated position of $\mathbf{x}_i$ is

$$\begin{aligned}\mathbf{x}_i = &\sum_{\mathbf{p}_j \in P} \mathbf{p}_j \frac{\theta(\|\mathbf{x}_i - \mathbf{p}_j\|)/v_j \|\mathbf{x}_i - \mathbf{p}_j\|}{\sum_{\mathbf{p}_j \in P} \theta(\|\mathbf{x}_i - \mathbf{p}_j\|)/v_j \|\mathbf{x}_i - \mathbf{p}_j\|} \\ &+ \mu \sum_{\mathbf{x}_q \in (X \setminus \{\mathbf{x}_i\})} (\mathbf{x}_i - \mathbf{x}_q) \frac{w_q \theta(\|\mathbf{x}_i - \mathbf{x}_q\|)/\|\mathbf{x}_i - \mathbf{x}_q\|}{W}\end{aligned} \tag{7}$$

where $W = \sum_{\mathbf{x}_q \in (X \setminus \{\mathbf{x}_i\})} w_q \theta(\|\mathbf{x}_i - \mathbf{x}_q\|)/\|\mathbf{x}_i - \mathbf{x}_q\|$, $\|\cdots\|$ is the L^2-norm, and $\theta(r) = e^{-16r^2/h^2}$ is adopted as in [5]. $\theta(r)$ is a rapidly decreasing smooth weight function with the support radius h defining the size of the influenced neighborhood. $\mu \in [0, 0.5)$ and h are served as two parameters selected by users to tune the performance of the operator. During our tests, the default values $\mu = 0.45$ and $h = 2L_{avg}$ work well on all models with L_{avg} being the average distance between particles to their k-nearest neighboring particles – we choose $k = 20$.

APPENDIX II
ALGEBRAIC POINT SET SURFACE (APSS) FITTING

Using $\mathbf{u} = [u_0, \ldots, u_4]$ as the vector of scalar coefficients describing a general algebraic sphere in $\Re^3$, the solution of algebraic sphere fitting at a given point $\mathbf{x} \in \Re^3$ can be evaluated by

$$\mathbf{u} = \arg \min_{\mathbf{u}, \mathbf{u} \neq \mathbf{0}} \|\mathbf{W}^{\frac{1}{2}}(\mathbf{x})\mathbf{u}\|^2. \tag{8}$$

The solution of $\mathbf{u}$ can be found by solving the following linear equation system (details can be found in [20]).

$$\mathbf{W}^{\frac{1}{2}}(\mathbf{x})\mathbf{D}\mathbf{u} = \mathbf{W}^{\frac{1}{2}}(\mathbf{x})\mathbf{b} \tag{9}$$

The coefficient matrices have $4n$ rows where n is the number of particles.

$$\mathbf{W}(\mathbf{x}) = \begin{bmatrix} \ddots & & & & & & \\ & w_i(\mathbf{x}) & & & & & \\ & & \beta w_i(\mathbf{x}) & & & & \\ & & & \beta w_i(\mathbf{x}) & & & \\ & & & & \beta w_i(\mathbf{x}) & & \\ & & & & & \ddots & \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} \vdots & \vdots & \vdots \\ 1 & \mathbf{x}_i^T & \mathbf{x}_i^T \mathbf{x}_i \\ 0 & \mathbf{e}_0^T & 2\mathbf{e}_0^T \mathbf{x}_i \\ 0 & \mathbf{e}_1^T & 2\mathbf{e}_1^T \mathbf{x}_i \\ 0 & \mathbf{e}_2^T & 2\mathbf{e}_2^T \mathbf{x}_i \\ \vdots & \vdots & \vdots \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} \vdots \\ 0 \\ \mathbf{e}_0^T \mathbf{n}_{\mathbf{x}_i} \\ \mathbf{e}_1^T \mathbf{n}_{\mathbf{x}_i} \\ \mathbf{e}_2^T \mathbf{n}_{\mathbf{x}_i} \\ \vdots \end{bmatrix}$$

Here, the weight function $w_i(\mathbf{x}) = \phi(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{h})$ describes the weight of the particle $\mathbf{x}_i$ for the local evaluation of the APSS at the position $\mathbf{x}$ with

$$\phi(r) = \begin{cases} (1 - r^2)^4 & |r| < 1 \\ 0 & |r| \geq 1 \end{cases}.$$

h is the support size of the repulsion where the same value is chosen as the orientation-aware WLOP above and $\{\mathbf{e}_k\}$ represents the unit basis vector of the coordinate system. $\beta = 10^6 h^2$ is adopted as suggested in [20] to compensate the variance of scaling. Note that, in practical computations, only the particles whose distance to $\mathbf{x}$ is less than h are employed to determine the coefficients in $\mathbf{u}$ since the weight function $w_i(\mathbf{x})$ only shows non-zero values on these particles. After computing the $\mathbf{u}$ vector, the center $\mathbf{c}$ and the radius of the algebraic sphere can be calculated as

$$\mathbf{c} = -\tfrac{1}{2u_4}[u_1, u_2, u_3]^T \text{ and } r = \sqrt{\mathbf{c}^T\mathbf{c} - u_0/u_4}.$$