Efficient Feature-preserving Local Projection Operator for Geometry Reconstruction

Bin Liao[a,b], Chunxia Xiao[a,*], Liqiang Jin[a], Hongbo Fu[c]

[a] *School of Computer, Wuhan University, Wuhan 430072, China*
[b] *Faculty of Mathematics and Computer Science, Hubei University, Wuhan 430062, China*
[c] *School of Creative Media, City University of Hong Kong, Hong Kong, China*

Abstract

This paper proposes an efficient and Feature-preserving Locally Optimal Projection operator (FLOP) for geometry reconstruction. Our operator is bilateral weighted, taking both spatial and geometric feature information into consideration for feature-preserving approximation. We then present an accelerated FLOP operator based on the random sampling of the Kernel Density Estimate (KDE), which produces reconstruction results close to those generated using the complete point set data, to within a given accuracy. Additionally, we extend our approach to time-varying data reconstruction, called Spatial-Temporal Locally Optimal Projection operator (STLOP), which efficiently generates temporally coherent and stable features-preserving results. The experimental results show that the proposed algorithms are efficient and robust for feature-preserving geometry reconstruction on both static models and time-varying data sets.

Keywords: Geometry reconstruction, Feature-preserving, Time-varying data, Locally optimal projection, Random sampling

1. Introduction

Reconstructing the geometry from raw scanned data has been an active research topic over the last two decades. Although various reconstruction methods have been proposed [1, 2, 3, 4, 5, 6, 7, 8], many problems still remain to be addressed due to geometry shape complexity and noise (outliers),

*Corresponding author
Email address: cxxiao@whu.edu.cn (Chunxia Xiao)

in addition, with high accuracy reconstruction requirement and new arisen applications. Surface reconstruction methods (e.g., [2, 4, 5, 9, 10, 11]) work well only for input point set data that is densely sampled and from which the orientation of the points can be accurately deduced. Point Set Surfaces (PSS) defined by local moving least squares (MLS) approximations of the point set data [10, 12] have been proven to be a powerful approach. However, due to the employment of plane fit operation, PSS is highly unstable in regions of high curvature where the sampling rate usually drops significantly.

To avoid using local surface approximation and normal estimation, Lipman et al. [7] develop a parameterization-free Locally Optimal Projection operator (LOP) for geometry reconstruction. This method is robust to noise and outliers of raw scanned data. However, the LOP method encodes only spatial relationship between input points while completely ignoring underlying surface geometry. It thus might fail to capture geometric features (Fig. 1 and Fig. 2). In addition, LOP is computationally expensive for reconstructing large point set data, while large data is commonly generated using laser and structured light scanners.

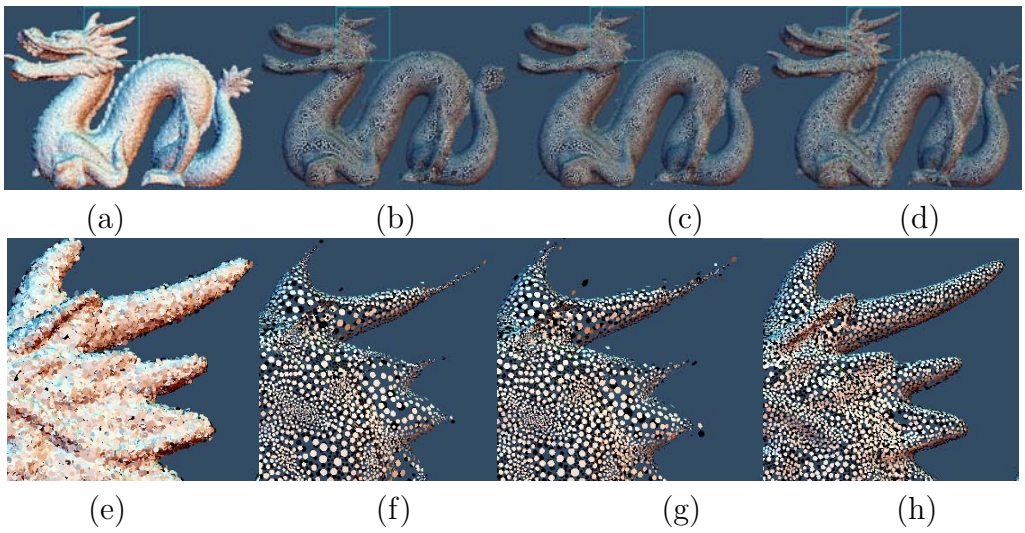


Figure 1: Our FLOP method better preserves geometric features than LOP [7] and WLOP [13]. (a) Original dragon model. (b-d) Results by LOP, WLOP and FLOP, respectively. (e-h) Respective close-up of the highlighted regions in (a-d).

In this paper, we introduce an efficient and Feature-preserving Locally Optimal Projection operator (FLOP) for geometry reconstruction. We first

develop a bilateral-weighted local optimal projection operator for preserving features, which works by taking both spatial and geometric feature information. Adaptive local-support parameter determination is exploited for robust projection. We then present an accelerated FLOP, which is based on the random sampling of the Kernel Density Estimate (KDE) [14] for the original point-set data. We show that reconstruction results by the accelerated FLOP are close to those generated using the complete point set data, to within a given accuracy, while time complexity is reduced significantly. Finally we show how FLOP can be extended for efficient and robust reconstruction of time-varying data.

In summary, this paper makes the following contributions:

- Developing a feature-preserving locally optimal projection operator (FLOP) for geometry reconstruction (Section 3.2).
- Proposing a Kernel Density Estimate (KDE) based random sampling technique to accelerate local optimal projection (Section 3.3).
- Introducing a fast spatial-temporal locally optimal projection operator (STLOP) for reconstructing time-varying data (Section 4).

Although the proposed FLOP algorithm is referred as a "reconstruction" algorithm, it can also be considered as a filtering/denoising algorithm. It is very useful for the preprocessing for the following methods, such as MLS [10], APSS [8], RBF [11], MPU [4], and Poisson surface reconstruction [5], since these methods require relatively clean input. In this sense, our FLOP is complementary with these methods.

2. Related Work

Many surface reconstruction methods have been proposed in recent years [1, 2, 3, 4, 10, 5, 7, 8, 15, 16]. Among them Point Set Surface (PSS) representation which is defined by local moving least squares (MLS) projection operator [12, 10] has been proven to be a powerful surface representation for point set data. Initial Levin's definition [12] and PSS definition [10] are relatively expensive to compute. Although significant progress [2, 17, 18] has been made to design simpler and more efficient definitions, the central limitation of the robustness of PSS is the required plane fit operation that is highly unstable in regions of high curvature where the sampling rate drops

below a threshold. Recently, Guennebaud et al. [8, 15] propose an algebraic point set surfaces (APSS) framework to locally approximate the data using algebraic spheres. Compared with MLS approximations, this strategy exhibits high tolerance with respect to low sampling densities while retaining a tight approximation of the surface.

Lipman et al. [7] develop a parameterization-free locally optimal projection operator (LOP) for geometry reconstruction, which originates from the multivariate L_1 median [19]. LOP works well on raw data without relying on any local parameterization of the points or their local orientation, and is thus robust to noise and outliers of raw scanned data. However, this method suffers from high computational cost for local optimal minimization and fails to preserve geometry features well. Recently, by incorporating adaptive density weighting into LOP, Huang et al. [13] modify the LOP operator to handle point sets with non-uniform sampling, which they call WLOP. They also present a robust normal estimation method based on priority-driven normal propagation and orientation-aware PCA, which is adopted for normal estimation in our current system.

Using range scanning techniques such as structured light [20] and space-time stereo [21], it is now possible to capture detailed 3D geometry nearly at real-time rates, which though is often corrupted with heavy noise. Several methods have been proposed to reconstruct time-varying data sets. For example, to obtain smooth and temporally coherent filtering results, Schall et al. [22] extend the non-local image denoising method [23] to 3D geometry. This method is essentially a filtering method and thus cannot work as a re-sampling tool like the LOP operator. Furthermore, such method is time-consuming since it has to compare regions of the surface. Instead, our method reconstructs clean time-varying surfaces by an efficient local optimal projection operator, which is more robust for surfaces with outliers. Furthermore, using our method, the number of the reconstructed points can be deliberately different from that of original time-varying surfaces, which is useful for producing time-varying surfaces with different resolution (sampling). Wand et al. [24] provide a system for reconstructing the topology, shape, and dense correspondences from unstructured time-varying point clouds. However, this method suffers from large computational cost, and the employed iterative assembly heuristic for inferring the discrete 4D topology does not always guarantee for finding a good solution. Mitra et al. [25] present an approach for registration of point clouds of moving and deforming objects. Without computing correspondence, this method exploits the underlying

temporal coherence in the data and directly computes object motion from the raw scanner for geometry registration.

A shorter version of this paper appeared in [26].

3. Fast Feature-preserving LOP

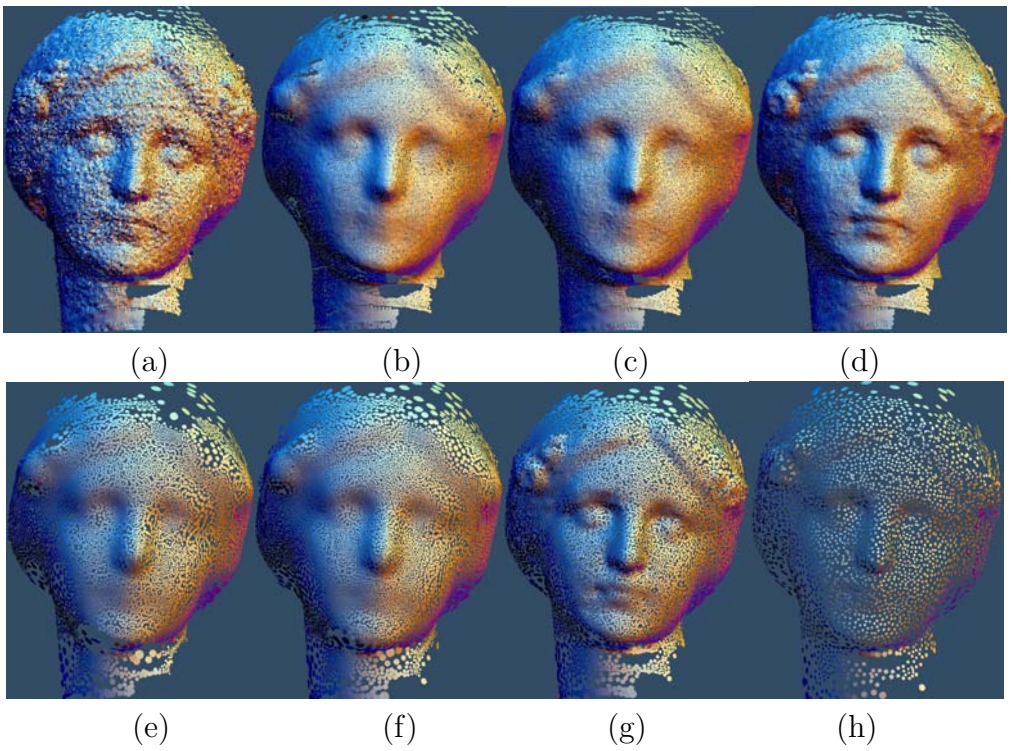


Figure 2: FLOP provides a more concise representation that captures well the input data. (a) Raw scanned data. (b-d) Reconstruction results by LOP, WLOP, and FLOP, respectively, where only 1/2 of the original point number is used. (e-g) Reconstruction results by LOP, WLOP, and FLOP, respectively, using 1/4 of the input points. (h) Reconstruction results using FLOP with 1/8 of the input points.

In this section, we first briefly review the original method of local optimal projection (LOP) [7]. A bilateral weighted LOP, which we call FLOP, is then presented for feature-preserving geometry reconstruction. Finally we introduce an acceleration technique for FLOP, which benefits from the random sampling of the Kernel Density Estimate.

3.1. Review of Local Optimal Projection

Local Optimal Projection (LOP) proposed by Lipman et al. [7] is a parameterization free algorithm for geometry reconstruction. Given the point set data $P = \{p_j\}_{j\in J} \subset R^3$, LOP projects an arbitrary point-set $X^{(0)} = \{x_i^{(0)}\}_{i\in I} \subset R^3$ onto the set P , where I, J denote the indices sets. The desired set of projected points, denoted as $Q = \{q_i\}_{i\in I}$, is defined as the fixed point solution of the equation $Q = G(Q)$. Here

$$G(C) = \arg \min_{X=\{x_i\}_{i\in I}} \{E_1(X, P, C) + E_2(X, C)\}, \tag{1}$$

with

$$E_1(X, P, C) = \sum\nolimits_{i\in I} \sum\nolimits_{j\in J} \| x_i - p_j \| \theta(\| c_i - p_j \|),$$

$$E_2(X, C) = \sum\nolimits_{i'\in I} \lambda_{i'} \sum\nolimits_{i\in I\setminus\{i'\}} \eta(\varphi_{i'i})\theta(\psi_{i'i}),$$

where $\varphi_{i'i} = \| x_{i'} - c_i \|$ and $\psi_{i'i} = \| c_{i'} - c_i \|$. The function $\theta(r)$ is a fast-decreasing smooth weight function, with compact support radius h defining the size of the influence radius (e.g., $\theta(r) = e^{-r^2/(h/4)^2}$). The term E_1 drives the projected points Q to approximate the geometry of P, which is also called multivariate L_1 median [19]. The term E_2 is a repulsion term, preventing $x_{i'}$ from getting too close to other points, where the repulsion function $\eta(r) = 1/(3r^3)$ in [7] and $\eta(r) = -r$ in [13]. $\{\lambda_i\}_{i\in I}$ are balancing terms between the two cost functions.

3.2. Feature-preserving LOP (FLOP)

Although LOP is an effective approach to reconstruct complex geometry, this method has the following drawbacks. First, when processing complex point set data with sharp features, the geometry features may not be preserved well, as illustrated in Fig. 1 and Fig. 2. Second, h is an important parameter which plays a major role in the application of LOP, but it has to be manually adjusted by trial and error to achieve satisfied results. Third, since the computational complexity of this method is superlinear in the number of input points, it is computationally expensive for large point set data. In this section, we try to address the above problems.

We observed that the multivariate L_1 median term $E_1(x)$ is defined as the sum of weighted Euclidean distances to the data points but does not consider geometry features of the underlying point-set surface. This is why LOP

cannot successfully capture geometric features, causing sharp features such as edges and corners blurred. Motivated by the geometry bilateral filtering [27, 28] and the ellipsoidal weigh function used in [29], we propose a bilateral weighted LOP operator for feature-preserving geometry approximation.

To this end we integrate a feature preservation weight θ_r into the L_1 median term E_1. Specifically the term $E_1(X, P, C)$ is redefined as follows:

$$E_1(X, P, C) = \sum\nolimits_{i \in I} \sum\nolimits_{j \in J} \parallel x_i - p_j \parallel \theta_s(\xi_{ij}) \theta_r(\zeta_{ij}),$$

where $\xi_{ij} = \|c_i - p_j\|$, $\zeta_{ij} = \langle n_i, c_i - p_j \rangle$, and n_i is the normal of point c_i, which can be estimated using the methods like [30, 31, 32]. The weight function θ_r ($\theta_r(x) = e^{-x^2/2\sigma_r^2}$ in our case) is a feature preservation weight that penalizes large variation in geometry similarity, which is defined as the height difference of point p_i over the tangent plane of the point c_i [27]. Following [7, 13], we let $\theta_s(x) = e^{-x^2/(h/4)^2}$ with the finite support radius h, whose optimal value will be automatically chosen (Section 3.2.1).

It should be pointed out, although the term $E_1(X, P, C)$ incorporates the normal information, it needs no consistent normal information, that is, it doses not require that all the normals point inside or outside. As our projected points are computed in an optimization procedure, unlike the geometry bilateral filtering [27, 28] which move the point along its normal direction, thus unoriented normals can be used in our method. This is very important since consistently orienting the normal information is a difficult problem, especially for very noisy point set [1, 13]. The state-of-the-art feature-preserving surface reconstruction methods, such as point set surface (PSS)[10], robust implicit moving least squares (RIMLS) [33], algebraic point set surfaces (APSS) [8, 15], and Poisson surface reconstruction (PSR) [5], require consistently oriented points as input, which makes them not robust for models with heavy noises. However, since our method can use unoriented normals, thus, our method is less susceptible to issues of robustness than PSS methods. We shall show later that although our formulation involves normal estimation, it performs robustly even for very noisy data.

We keep the repulsion term E_2 unaltered. Therefore, the iterative solution of the original LOP algorithm can be easily adapted here. More specifically, given the current iterate $X^{(k)} = \{x_i^{(k)}\}_{i \in I}$, the new projected point $x_{i'}^{(k+1)}$ is

computed as:

$$
\begin{aligned}
x_{i'}^{(k+1)} &= \sum_{j\in J} p_j \frac{\alpha_j^{i'}}{\sum_{j\in J}\alpha_j^{i'}} \\
&+\mu \sum_{i\in I\setminus\{i'\}} \left(x_{i'}^{(k)} - x_i^{(k)}\right) \frac{\beta_i^{i'}}{\sum_{i\in I\setminus\{i'\}}\beta_i^{i'}},
\end{aligned}
\tag{2}
$$

where

$$
\alpha_j^{i'} = \frac{\theta_s(\|x_{i'}^{(k)} - p_j\|)\theta_r(\langle n_{i'}^{(k)}, x_{i'}^{(k)} - p_j\rangle)}{\|x_{i'}^{(k)} - p_j\|},
$$

$$
\beta_i^{i'} = \frac{\theta_s(\|x_{i'}^{(k)} - x_i^{(k)}\|)}{\|x_{i'}^{(k)} - x_i^{(k)}\|} \left|\frac{\partial \eta}{\partial r}(\|x_{i'}^{(k)} - x_i^{(k)}\|)\right|.
$$

$X^{(0)}$ is a crude initial guess set. Similar to [7], the parameter $\mu > 0$ comes from the balancing parameters setting $\lambda_{i'} = \mu\frac{\sum_{j\in J}\alpha_j^{i'}}{\sum_{i\in I\setminus\{i'\}}\beta_i^{i'}}$. In our experiments, we use the repulsion parameter μ as $\mu \in [0, 1/2]$. Similar to [13], we define $\eta(r) = -r$ which produces locally regular point distribution. Local adaptive density weights [13] can also be easily incorporated into Equation 2 to make the projected points more uniformly distributed, thus avoiding excessive projected points in feature regions. Practically, the iteration procedure tends to converge in a very small number of steps, typically around 10. In Fig. 1 and Fig. 2, we compare FLOP with LOP [7] and WLOP [13]. Clearly, our proposed method preserves the features better.

Normal estimation. Although our method incorporates normal information, it does not need consistent normal information. In our experiments, we use PCA to compute the normal for each projected point and produce pleasing results. Classical PCA relies on Euclidean distances between points. In this paper, PCA relies on not only distances between points, but also bilateral weights between points. Note that the normal must be re-estimated for every iteration (Eq. 2). For models with heavy noise and/or outlier, to produce clean point set, meanwhile preserving the features, at first several iterations (one or two iterations), we set a large value as $\sigma_r \in [0.6, 0.9]$ in the feature preservation weight θ_r, which makes the FLOP work more like WLOP. It should be noted that, although large value σ_r for FLOP may smooth the features, the first one or two such FLOP iterations performed on models severely corrupted by noise and outliers will not destroy the features.

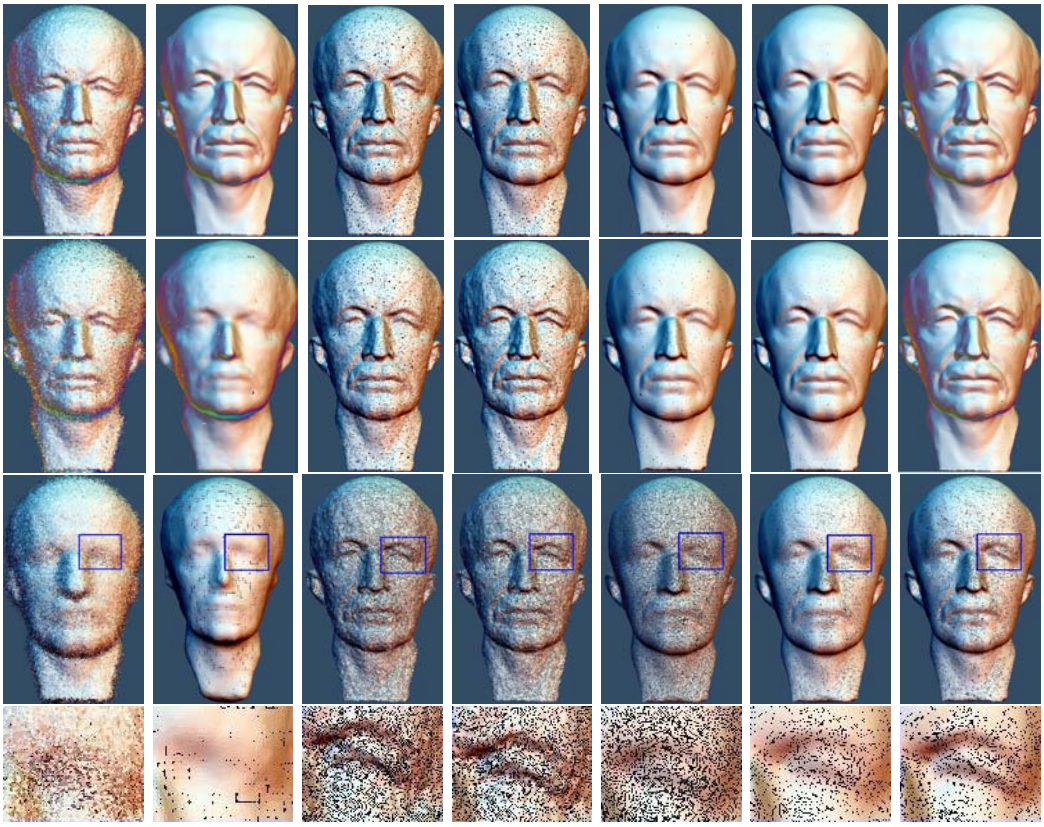


Figure 3: From top row to bottom and for left to right: input models with gradually increasing levels of noises, and reconstructed results using (b)Poisson surface reconstruction (PS)[5], (c) robust implicit moving least squares (RIMLS) [33], (d) algebraic point set surfaces (APSS) [8, 15], (e) LOP [7], (f) WLOP [13], and (g) proposed FLOP.

For the following iterations, the value for σ_r is reduced to relative smaller value as $\sigma_r \in [0.01, 0.3]$ to preserve the geometric features.

In Fig.3 we show the robustness of FLOP under different noise level, and compare with the state-of-the-art surface reconstruction methods. The Gaussian noise is added manually in this example. With the noise level gradually augmenting, our method still reconstructs the features well. This is possibly because FLOP is an optimization process and thus the inaccurate intermediate reconstruction results can be gradually rectified by our alternating iterative algorithm (i.e., alternating between normal estimation and reconstruction). Compared to LOP and WLOP, our method preserves the

features better even for models with the least noise (Fig. 3(e) and (f)). We also give the comparison results with PSR [5], RIMLS [33], APSS [8, 15], which requires consistently oriented points as input. For fairness, the same consistent normal estimation is used for these methods, the reconstructed surfaces are densely sampled and rendered using the same point-based rendering pipeline. It is easily noticeable that our method outperforms these methods for models severely corrupted by noise and generates comparable results for the case of slight noise.

3.2.1. Adaptive Local Support Radius

The local support radius h of $\theta_s(x)$ in the LOP operator has important influence on the quality of reconstruction results. Too large value of h might cause over-smoothed results (see the ears of bunny in Fig. 4(f)). In contrast, if the value of h is too small we cannot achieve desired results either (e.g., Fig. 4(d)). However, such important parameter is manually specified by trial and error in the existing literature [7, 13]. In addition, we found that having a global uniform parameter h is often insufficient for models with spatially varying features. These observations motivated us to automatically deduce optimal values of h from the data itself, which are adaptively determined with respect to local features.

The basic idea is to use small h in regions with high curvature to better capture geometric features, while use relative larger h in flat regions. We employ a bilateral weight average algorithm to decide h, which is robust for reconstructing models with heavy noises.

Bilateral weight average algorithm. We compute the geometry bilateral weights for the projection point c_i with each point in its neighborhood N_i. If bilateral weight is large, the point contributes more to the local support radius h_i. Specifically, h_i is defined as:

$$h_i = \frac{\sum_{j \in J} \theta_c(\langle n_i \, , c_i - p_j \rangle) \theta_s(\| c_i - p_j \|)}{S_i} R,$$

where $\theta_c = e^{-x^2/2\sigma_c^2}$ and $\theta_s = e^{-x^2/2\sigma_s^2}$ are the standard Gaussian filters with parameters σ_c and σ_s, respectively. σ_s is set as the radius of the neighborhood N_i, and σ_c is set as the standard deviation of the normal variation in the neighborhood N_i. Constant R defines the size of the influence neighborhood N_i for each projected point c_i and S_i is the number of points in N_i. In our experiments, the default value of R is set as $R = 6\sqrt{d_{bb}/|J|}$, where d_{bb} is

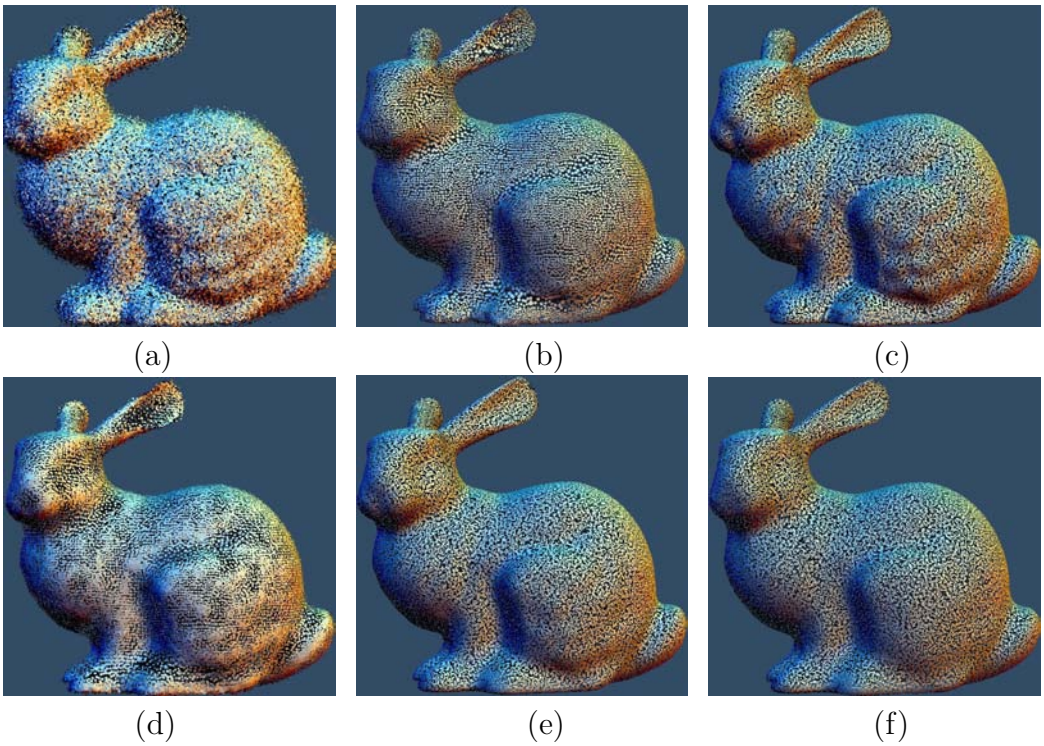
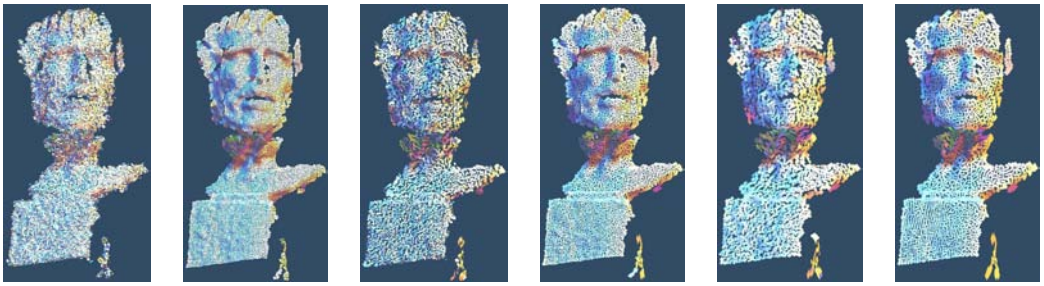


Figure 4: (a) Input point set with heavy noise. (b) Result of FLOP using uniform support radius h. (c) Result of FLOP using adaptive support radius h (bilateral weight average). (d-f) Results of LOP with increasing parameter of uniform support radius h, respectively.

the diagonal length of the bounding box of the input point set. Note that such default value R has already been used in [13] for their setting of uniform parameter h.

Fig. 4 and Fig. 5 show the effectiveness of our bilateral weighted average algorithm to decide local support radius h. In Fig. 4, for an input model with heavy noise, our method using adaptive support radius h preserves the feature better than that using uniform support radius h. We also present comparison results with LOP that uses uniform support radius h with different values. In Fig. 5, we show the robustness of bilateral weight average algorithm for the same model with different sampling densities, which are received by random sampling. Note that the adaptive support radius h works well for these model.



(a) (b) (c) (d) (e) (f)

Figure 5: Robustness of adaptive local support radius for models with different sampling densities. (a) Model with 47,256 points, (b) reconstruction of (a) using FLOP; (c) model with 10,1521 points, (d) reconstruction of (c) using FLOP; (e) model with 4,375 points, (f) reconstruction of (e) using FLOP.

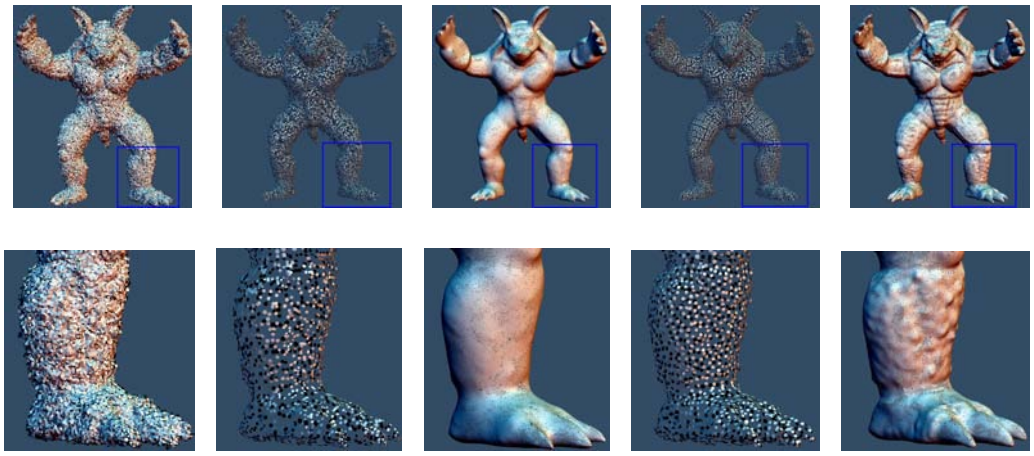
3.3. Acceleration of FLOP

The computational complexity of the multivariate L_1 median term $E_1(x)$, which is the computational bottleneck of FLOP, is superlinear in the number of input points P [19]. Therefore, it is computationally expensive to process large point set data. In this subsection, we show how FLOP can be greatly accelerated with little sacrifice of reconstruction quality.

Our goal is to find a reasonable approximation of $E_1(x)$, denoted as $\hat{E}_1(x)$, while using a much smaller point set $\hat{P} = \{\hat{p}_j\}_{j \in K} \in R^3$ to generate $\hat{E}_1(x)$, $|K| \ll |J|$. Mathematically, it can be formulated as the following minimization problem: $\min_{\{\hat{p}_k\}_{k=1}^{K}} D(E_1(x), \hat{E}_1(x))$ subject to $\hat{E}_1(X, \hat{P}, C) = \sum_{i \in I} \sum_{j \in K} \parallel x_i - \hat{p}_j \parallel \theta_s (\parallel c_i - \hat{p}_j \parallel) \theta_r(\langle n_i \, , c_i - \hat{p}_j \rangle)$, where D is a distance measure between these two terms, which can be defined using the L_p type of distance.

The subsampled point set $\hat{P}$ should be a more accurate, smooth and feature-aware approximation to the original input noisy point set P. That is, the point set $\hat{P}$ should be smooth, uniformly distributed, and moreover, the subsampled points $\hat{P}$ should be feature preserved. Then by building median term $\hat{E}_1(x)$ on $\hat{P}$, and computing the projected point set, we can reconstruct the final results in higher speed and more accurate approximation.

Motivated by [14], which presents a compact density representation for mean-shift acceleration, we use a similar sampling technique to address the above problem. The sampling procedure is based on the random sampling of the Kernel Density Estimate (KDE) $f(x) = \frac{1}{|J|} \sum_{j=1}^{j=J} \Theta_H (x - p_j)$, where



(a) (b) (c) (d) (e)

Figure 6: From left to right columns: (a) Input noisy models, (b) naive sampled point set, (c) reconstruction using naive sampled point set, (d) sampled point set using random sampling of KDE, (e) reconstruction using random sampling of KDE.

function $f(x)$ is defined on original point set data P, and Θ_H is a standard Gaussian Kernel with a symmetric positive definite $d \times d$ bandwidth matrix H. To sample points K from KDE, for each $k = 1, ..., |K|$, we choose $\hat{p}_k$ using a three-step procedure: a) Choose a random integer $r_k \in \{1, ..., J\}$; b) Choose a random sample δ_k from $\Theta_H(\bullet)$; c) Set $\hat{p}_k = p_{r_k} + h\delta_k$, where h is a bandwidth of the profile of the kernel Θ_H. Freedman et al. [14] have proved that $\hat{p}_k$ is a proper sample of $f(x)$. When $\hat{E}_1(x)$ is constructed on the samples $\hat{P}$ defined as above, the reduced multivariate L_1 median $\hat{E}_1(x)$ is close to $E_1(x)$ defined on complete data under a controlled approximation accuracy.

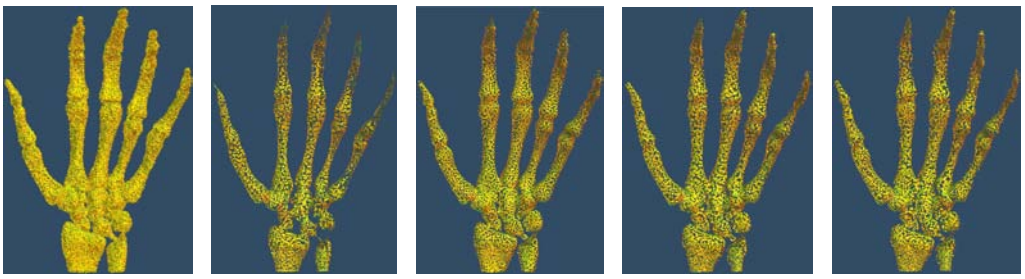
Theorem 4.1. We assume that $g(x) = \sum_{i \in J} \parallel x - p_i \parallel$ and $\hat{g}(x) = \sum_{j \in K} \parallel x - \hat{p}_j \parallel$. Let the expected squared L_2 distance between the two functions be given by

$$F = E\left[\int \left(g(x) - \hat{g}(x)\right)^2 dx\right].$$

Then,

$$F \leq |I|\left(h|K| + (|J| - |K|)D\right)^2,$$

where D is the max value of $\parallel x - p_i \parallel$ and r_j is a random integer in $\{1, ..., |J|\}$.



(a) (b) (c) (d) (e)

Figure 7: From left to right columns: Input noisy models, results by WLOP [13], results by our unaccelerated FLOP, and results by accelerated FLOP with sampling rate $|J|/|K| = 8$ and sampling rate $|J|/|K| = 16$, respectively.

Proof:

$$
\begin{aligned}
F &= E\left[\int \left(g\left(x\right) - \hat{g}\left(x\right)\right)^2 dx\right] \\
&= E\left[\int \left(\sum_{i\in|J|} \parallel x - p_i \parallel - \sum_{j\in|K|} \parallel x - \hat{p_j} \parallel\right)^2 dx\right] \\
&= E\left[\int \left(\sum_{i\in|J|} \parallel x - p_i \parallel - \sum_{j\in|K|} \parallel x - p_{r_j} - H^{1/2}\delta_j \parallel\right)^2 dx\right] \\
&= E\left[\int \left(\sum_{j\in|K|} \parallel x - p_{r_j} \parallel - \sum_{j\in|K|} \parallel x - p_{r_j} - H^{1/2}\delta_j \parallel \right.\right. \\
&\qquad \left.\left. + \sum_{i\in|J|\cap i\neq r_j} \parallel x - p_i \parallel\right)^2 dx\right] \\
&\leq E\left[\int \left(\sum_{j\in|K|} \parallel H^{1/2}\delta_j \parallel + \sum_{i\in|J|\cap i\neq r_j} \parallel x - p_i \parallel\right)^2 dx\right] \\
&= E\left[\int \left(h \sum_{j\in|K|} \parallel \delta_j \parallel + \sum_{i\in|J|\cap i\neq r_j} \parallel x - p_i \parallel\right)^2 dx\right] \\
&\leq E\left[\int \left(h|K| + \sum_{i\in|J|\cap i\neq r_j} \parallel x - p_i \parallel\right)^2 dx\right] \\
&\leq E\left[\int (h|K| + (|J| - |K|)D)^2 dx\right] \\
&= |I|(h|K| + (|J| - |K|)D)^2.
\end{aligned}
$$

According to [14], both the expected squared L_2 distance between $\sum_{i\in J}\theta_s\left(\parallel x-p_i\parallel\right)$ and $\sum_{j\in K}\theta_s\left(\parallel x-\hat{p}_j\parallel\right)$ and the expected squared L_2 distance between $\sum_{i\in J}\theta_r(\langle n_i\ ,x-p_i\rangle)$ and $\sum_{j\in K}\theta_r(\langle n_j\ ,x-\hat{p}_j\rangle)$ have been proved under a controlled approximation accuracy $4Ah+A^2h^2+\frac{B}{jh^3}+\frac{ABV}{Kh^2}$, where A, B, V are constants and do not depend on h or $|K|$. As we have proved that $g(x)$ and $\hat{g}(x)$ are close in expectation, combined with above results [14], we might reasonably expect that $E_1(X,P,C)$ and $\hat{E}_1(X,\hat{P},C)$ are close in expectation, which is empirically supported by our experiments.

Using the compact KDE based term $\hat{E}_1(X,\hat{P},C)$, the accelerated FLOP can be performed in a two-step procedure:

1. Sampling: Take $|K|$ samples of the KDE $f(x)$ to yield $\{\hat{p}_k\}_{k=1}^{K}$, and construct the new median term $\hat{E}_1(X,\hat{P},C)$,

2. Local optimal projection:
$G\left(C\right)=\arg\min_{X=\{x_i\}_{i\in K}}\{\hat{E}_1(X,\hat{P},C)+E_2(X,C)\}$.

Using the reduced set of K samples instead of the complete set of J points, $|K|\ll|J|$, the computational complexity of $E_1(x)$ is significantly reduced from $O(J)$ to $O(K)$. As the computational complexity of sampling operator is linear $O(K)$, thus, the whole optimization procedure is greatly accelerated.

To better emphasize the benefit of such our sophisticated KDE selection, Fig. 6 shows the reconstruction comparison results with a naive point set subsampling, that is, a pure stochastic selection method. Both methods subsample the original 1,354,321 points into 15,000 points. As illustrated in Fig. 6 (b) and (d), the subsampled point set using random sampling of KDE is smoother, uniformly distributed, and is more feature-preserving. Based on these two subsampling results, we apply FLOP to reconstruct the original noisy data using 160,320 points, respectively. The results show that reconstruction result generated on the KDE-based samples is smoother and preserves the features better.

The proposed acceleration method significantly reduces the computational cost while obtaining almost the same geometry reconstruction results as the FLOP method using the complete data. As illustrated in Fig. 7, even with very high sampling factor such as $|J|/|K|=16$, the reconstruction results are still visually pleasing. For the example of Stanford armadillo in Fig. 6, the original point set with 1,354,321 points is first downsampled to 84,646 points, which we use for geometry reconstruction with 32,431 points. The computational time is significantly reduced from 1901 seconds to 17 seconds.

For input model with extremely large data, and the number of projected points in also large, for example, $|J| > 10^7$ and $|I| > 10^5$, even the accelerated method will be slow. In this case, the optimization can be further accelerated based on the trivial observation: since θ_s and θ_r are fast-decreasing smooth weight functions, the points p_j far away from the projected point x_i have almost no influence on the LOP computing. Therefore, for each x_i, the computation of $\hat{E}_1(x_i)$ can be well approximated by using its nearest neighborhood N_i in $\hat{P}$, instead of all the data points in $\hat{P}$. $E_2(X, C)$ can be approximated in a similar way for acceleration. Although this naive method will reduce the accuracy of the reconstruction, it is an alterative method to process extremely large data.

4. Reconstruction of time-varying data

To reconstruct surface geometry from time-varying data, a naïve method would be to apply FLOP to individual frames. However, such independent application of FLOP to frames would easily lead to the problem of temporal incoherence, as shown in Fig. 8 and Fig. 9. It would also bring the problem of separate parameter tuning for different frames, which is time consuming. To address the problems, we introduce a fast Spatio-Temporal Locally Optimal Projection operator (STLOP) for faithful reconstruction of time-varying data.

The problem can be formally formulated as follows: given the input time-varying data $P = (P^1, P^2, ..., P^T)$, where $P^t \in R^3$ is the t-th frame surface, and the number of points in frame P^t is $|J^t|$, which possibly varies over frames, we would like to approximate P using time-varying data $Q = (Q^1, Q^2, ..., Q^T)$, where each frame Q^t may also have different number of points $|K^t|$, and $|K^t| \ll |J^t|$. The key idea to achieve an effective STLOP operator is to have the definition of the neighborhood $N(p_i)$ at point p_i involving not only the points from individual frames but also from their neighboring frames.

Simply constraining temporally adjacent points into the optimization is usually insufficient, especially for models involving large deformations or captured under low FPS, since surface patches from different frames in the neighborhood may be very different. Instead, we should respect surface features and require temporally adjacent points with similar features to be processed in a coherent manner. Thus we employ approximately invariant signature

evaluation based on surface matching to produce spatially-continuous and temporally-coherent results.

Figure 8: Top row: input noisy time-varying data. Second row: results by independent application of FLOP to individual frames. Last two rows: results by STLOP with different numbers of projected points. We recommend to see the electronic version of these images.

4.1. Approximately Invariant Signature Evaluation

In order to enable efficient matching for point set surfaces in the animated object, we endow each point with a signature that is invariant to rigid/scaling transformation [34, 35]. To compute the signature, we first build a multi-scale representation $P_j^t(x) = G_j^t(x, h_j) * P_{j-1}^t(x)$ for a input surface P^t ($P_0^t(x) = P^t$), which can be produced from the convolution of a variable-scale Gaussian $G_j^t(x, h_j) = e^{-x^2/h_j^2}$, where $*$ is the Gaussian convolution operation in x and

h_j is the parameter called scale. We apply a sequence of scale $h_j = h_0 F^j$ for some $F > 1$ to compute the jth scale representation $P_j^t(x)$ for surface P^t. We use $F = 2^{1/3}$ in our method. In our paper, we apply iterative minimization of a MLS error [34] to obtain $P_j^t(x)$, which is easy to implement and is efficient for multi-scale representation.

With the multi-scale representation $P_j^t(x)$ for the surface, given a point $p \in R^3$ with a normal n and a scale h, inspired by the SIFT method [36, 34], we can define a local signature vector $\varpi(p, n, h)$. We first define an orthogonal local frame (u, v, n) on p, then $R \times S$ points ξ_{kl} are defined, which sample a disc around point p:

$$\xi_{kl} \approx p + \frac{2lh_j}{R}(\cos(\frac{2\pi k}{S})u + \sin(\frac{2\pi k}{S})v) \tag{3}$$

where $k = 1 \ldots R$ and $l = 1 \ldots S$. Then we compute the normal v_{kl} for ξ_{kl}, which is the weighted average of the normals n_i of the points that are in the neighborhood of ξ_{kl} and come from the input data $P_j^t(x)$, $v_{kl} = \sum w_i n_i$, where the weights w_i are normalized Gaussian weights. With the normals v_{kl} of the sample points ξ_{kl}, a $R \times S$ array of values is defined by projecting v_{kl} onto the direction connecting the p and ξ_{kl} and each array element is s_{kl}. Applying the Fourier transformation to s_{kl} and following the Discrete Fourier Transform, the array of $(\tilde{s}_{kl})$ is computed, $k = 1 \ldots R$, $l = 1 \ldots S$. The upper left corner of $\tilde{s}_{kl}$ are values that are approximately invariant to the choice of u and v if S is sufficiently large.

The signature vector of point p is defined by extracting the low-frequency coefficients of $\tilde{s}$ so that:

$$\varpi(p, n, h_j) = (\tilde{s}_{kl})_{k=1\ldots R', l=1\ldots S'}. \tag{4}$$

The signature vector ϖ of point p is not invariant under shape rigid/scaling transformation. We use $R = 32$, $S = 8$ for point sampling, and set $R^{'} = 6$, $S^{'} = 4$ to define a 24-dimensional signature vector, which is found enough to capture the approximately invariant information. As signature vector ϖ is invariant to rigid/scaling transformation [34, 35], thus, it enables efficient matching for the surfaces in the animated object. To make the signature vector ϖ more robust for noisy surface, we usually compute the ϖ on the $j = 2$ or $j = 3$ scale $P_j^t(x)$ surface.

4.2. Spatio-temporal Locally Optimal Projection (STLOP)

Similar to the spatial domain, we want points from distant frames to contribute less to the new point position. Therefore we introduce the temporal

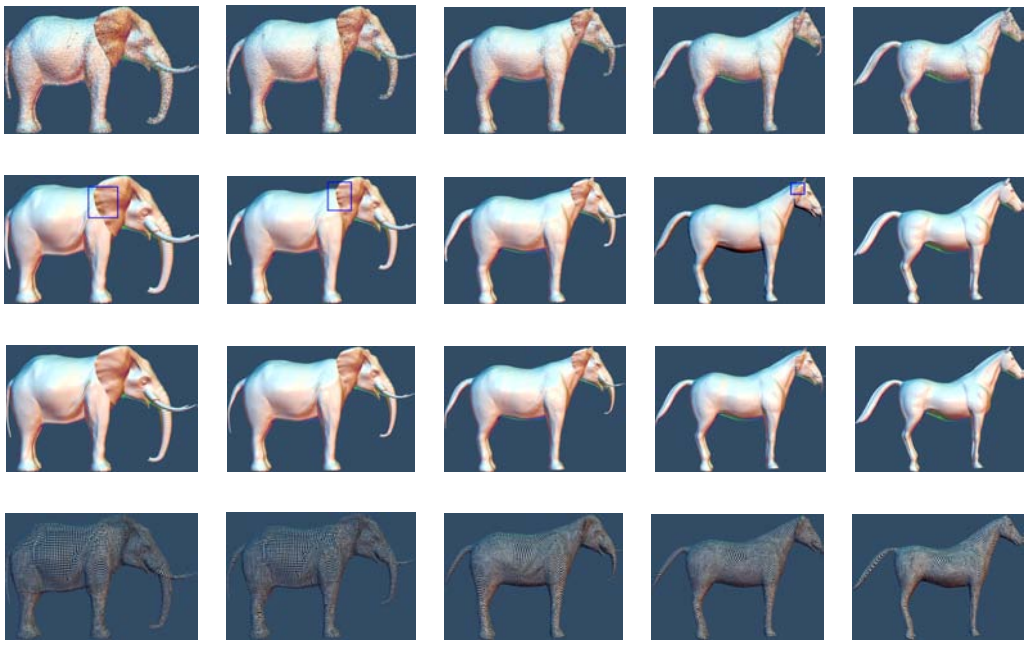


Figure 9: Top row: input noisy time-varying data. Second row: reconstructed results by applying FLOP to individual frames independently. Third and fourth rows: results by STLOP with different numbers of projected points. We recommend to see the electronic version of these images.

distance factor ψ_{dt} which weighs the contribution of the frame t. Specifically, we define $\psi_{dt} = e^{-\|t-c\|^2/\sigma^2}$, where c is the index of the current frame. In this paper, we set $\sigma = 2$. In addition, we can weigh the points in the neighboring frames based on point matching similarity. Points from frames with a higher similarity can contribute more to a smooth solution and should thus have a higher weight. We define the point similarity between the points c_i and p_j in different frames as $\phi_{dt}(\varsigma_{ij}) = e^{-\|\varpi_{c_i}-\varpi_{p_j}\|^2/2\sigma_r^2}$, which is measured based on the approximately invariant signature.

We determine the new spatial-temporal median term $E_1^{tv}(X, P, C)$ defined on the spatial-temporal neighborhood $N_t^P(p_i)$:
$E_1^{tv}(X, P, C) = \sum_{i\in Q}\sum_t \psi_{dt}\sum_{j\in N_t^P(x_i)} \parallel x_i - p_j \parallel \phi_{dt}(\varsigma_{ij})\theta_{st}(\xi_{ij})\theta_{rt}(\zeta_{ij})$. The point similarity distance ϕ_{dt}, the spatial distance θ_{st}, and the similarity distance weight θ_{rt} are defined between x_i and p_j that come from frame t. Note that when points x_i and p_j come from the same frame, it is unnecessary to compute the point similarity, i.e., setting $\phi_{dt} = 1$.

Similarly, the repulsion term $E_2^{tv}(X, C)$ also is defined on the spatio-temporal neighborhood $N_t^Q(p_i)$ for each point x_i, with $N_t^Q(p_i)$ defined in the projected time-varying point set $Q = \{q_i\}_{i \in I}$:
$E_2^{tv}(X, C) = \sum_{i' \in Q} \sum_t \psi_{dt} \sum_{j \in N_t^Q(x_i)} \eta(\| x_{i'} - c_i \|) \theta(\| c_{i'} - c_i \|)$.

Finally the optimization of the spatio-temporal locally optimal projection operator (STLOP) is defined as follows:

$$G^{tv}(C) = \arg \min_{X = \{x_i\}_{i \in Q}} \left\{ E_1^{tv}(X, P, C) + E_2^{tv}(X, C) \right\} \tag{5}$$

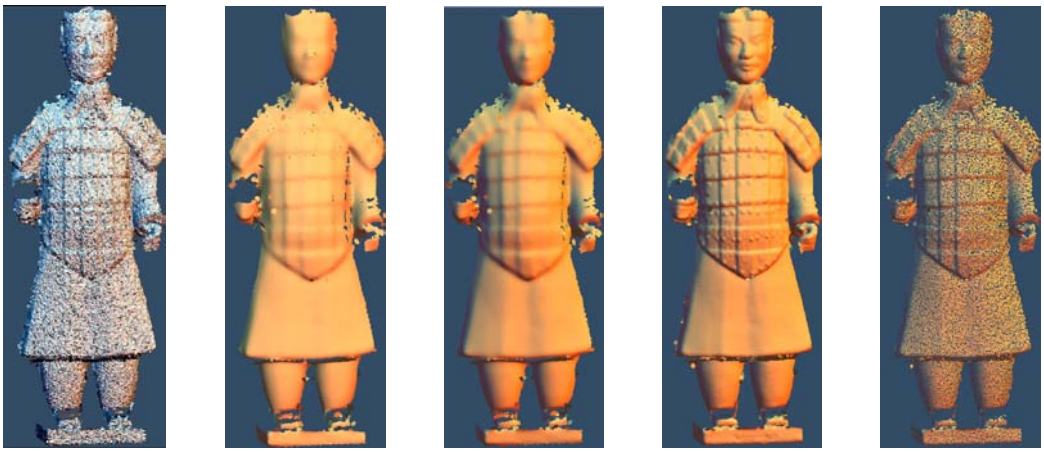
Similar to the acceleration of FLOP, STLOP can be accelerated based on the random sampling of the Kernel Density Estimate (KDE) of the time-varying data. The sampling can be either performed on each frame independently, or conducted in the spatio-temporal point set domain. For the latter case, it is usually sufficient to sample the points from the current frame and the frames right before and after the current frame. Note that to make convenient implementation, and to keep the results temporally consistent and projected points fairly distributed, we sample each frame using the same sampling factor $|J^t|/|K^t|$.

As shown in Fig. 8 and Fig. 9, the proposed STLOP operator successfully produce temporally stable and consistent results. Another example is presented in Fig.13. As our method considers neighboring point registration into the optimization, it performs well even in cases of large deformations or low FPS capturing. Besides, using the point registration based on approximately invariant signature, we do not necessarily need to compensate for motion between frames as the similarity of the temporal neighborhood is evaluated. In this way, our approach also automatically accounts for scene changes.

5. Experimental results and discussion

Our methods have been evaluated on models with various shape complexity and noise level. They have been also compared to the state-of-the-art works [7, 13] on both performance and quality.

In Fig.1, we give reconstruction comparison results on dragon model containing little noise. Our result is compared to those by the LOP method [7] and the WLOP method [13]. As shown in Fig. 1 (f) (g), the sharp features cannot be preserved well using these two methods [7, 13].



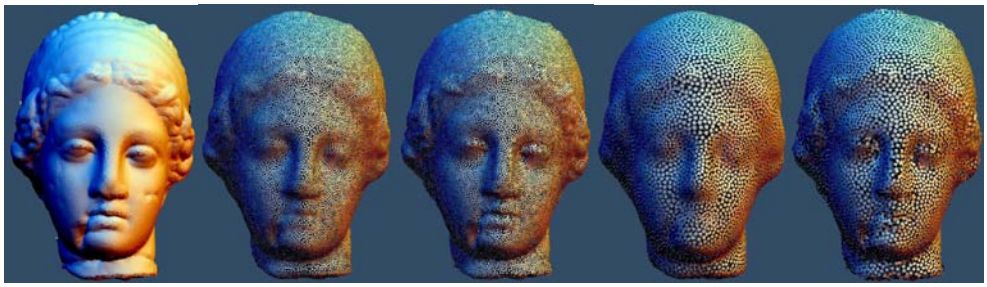
(a) (b) (c) (d) (e)

Figure 10: (a) The Terra Catta Warriors point set with moderate level of noise. (b-e) Reconstruction results by LOP [7], WLOP [13], FLOP, and FLOP with fewer projected points, respectively.

In Fig.2 and Fig. 10, we give our reconstruction results from the noisy raw scanned data, together with comparisons with [7] and [13]. Compared to LOP [7], our method preserves features better, and the projected points are distributed more fairly. Although the WLOP operator [13] achieves better point distribution than LOP, WLOP still cannot preserve the features well. We also present comparison results with [7] and [13] on fewer projected points in Fig. 2.

In Fig.4 and Fig.5 show the effectiveness of the uniformly local support radius h. In Fig.4, the bilateral weighted average algorithm is automatically selected to estimate the adaptive support radius h. As show in Fig.4, both FLOP results using uniform support radius h and using adaptive support radius h preserve features better than LOP method [7]. However, the FLOP result using adaptive support radius h is the best. Using uniformly h, when h is small, LOP method cannot filter out the noise effectively. In contrast, when h is large, the features cannot be preserved well, as illustrated in Fig.4 (d-f), where we give results of LOP with increasing uniform support radius h values, respectively. Fig. 5 illustrates the robustness of bilateral weight average algorithm for the same model with different sampling densities.

In Fig.3, we show the robustness of proposed FLOP under different noise level, and compare it with the state-of-the-art feature-preserving surface re-



(a) (b) (c) (d) (e)

Figure 11: (a) Input model. (b) and (d) are results using the LOP method with different projected points. (c) and (e) are the reconstruction results using our accelerated FLOP method with comparable number of projected points used for generated (b) and (d), respectively.

construction methods, PSR [5], RIMLS [33], and APSS [8, 15]. All these three methods require consistently oriented normal information. In this example, the noisy models are reconstructed with the same number of the points as that of the input models. For a fair comparison, we apply the MeshLab developed by The Visual Computing Group [37] to perform the implementations of [5, 33, 8, 15]. As illustrated in Fig.3, for input model with slight noise, the reconstructed results using these methods are comparable with our FLOP method. With the noise level gradually augmenting (the noise is added manually), even with extremely heavy noise, our method reconstructs the features much better. For RIMLS [33] and APSS [8, 15], it is difficult to preserve the features while filtering the heavy noise simultaneously. PSR [5] formulates the surface reconstruction as a spatial Poisson problem. This method is resilient to noise, however, it blurs the features of the the model with heavy noise. As these methods require consistently oriented normal information, however, it is well known that it is difficult to compute required normals for very noisy models, this also influence their reconstruction results. Note that we tune the parameters provided by MeshLab [37] to give the best reconstruction results for these methods.

Fig. 6, Fig. 7 and Fig. 11 show the FLOP acceleration results and compare with other methods. Fig. 6 presents the reconstruction comparison results with a naive point set subsampling method, the results conform that our KDE-based subsampling is a much better approximation method, and based on this subsampling results, we can produce much better reconstruc-

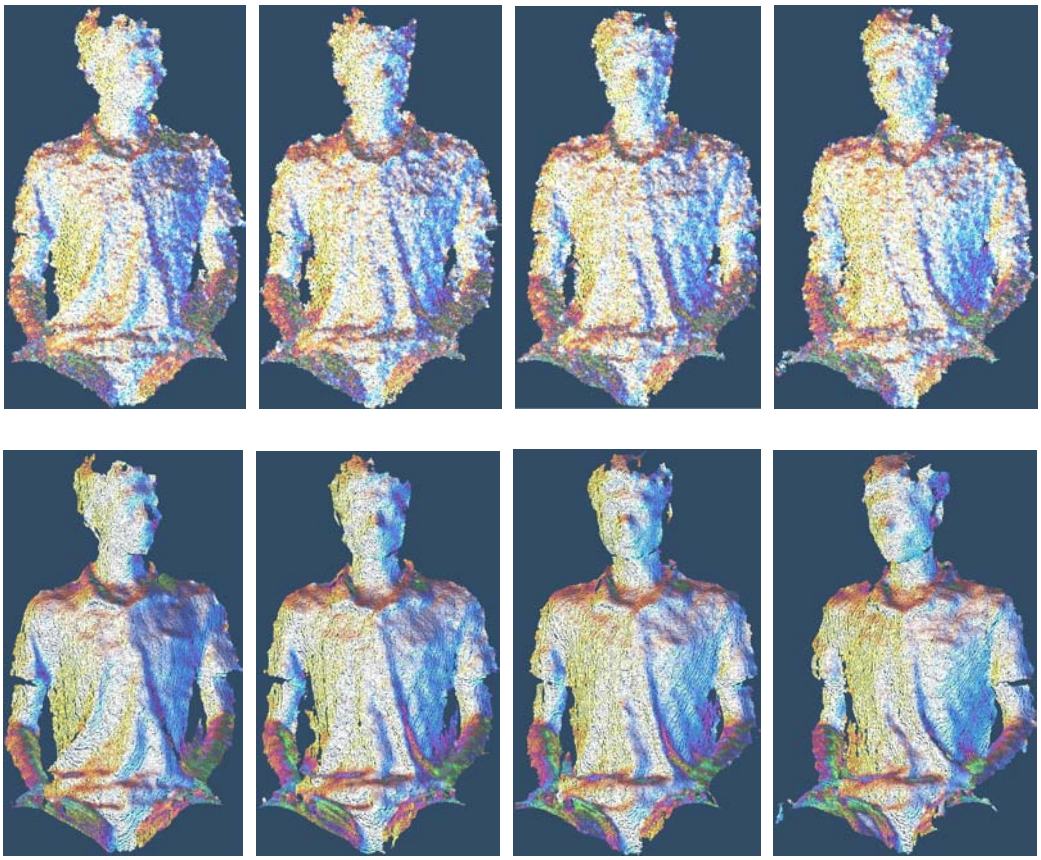


Figure 12: Top row: input noisy time-varying data. Bottom row: reconstruction results using STLOP.

tion results using FLOP. Fig. 7 gives the results by the accelerated FLOP, under different sampling factors. It is shown that even with very high sampling factor like $|J|/|K| = 16$, the reconstruction results by the accelerated FLOP are still satisfactory, with well-preserved features and fairly distributed points, while the computational cost is significantly reduced. Our acceleration technique is particularly suitable for processing massive point set data, where performing LOP on the complete set of the data is impractical on common PCs due to too high time complexity. Fig. 11 gives another example using the accelerated technique. There are 463,245 points in the original point set. In Fig. 11 (c), we first sample it into 65,345 points and reconstruct it using 25,365 points. In Fig.11 (e), we sample it in 25,313 points and reconstruct it using 8,365 points.

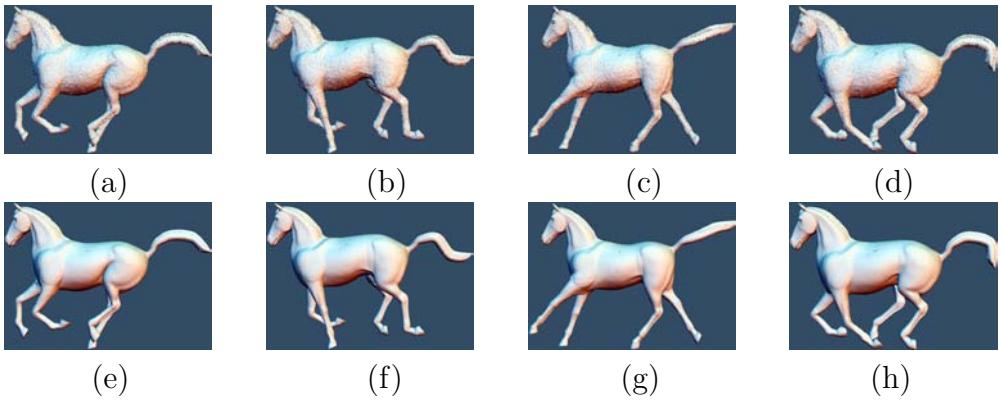


Figure 13: Geometry reconstruction from noisy time-varying data. Top row input time-varying data. Bottom row: reconstructed result.

Fig. 8 and Fig. 12 give the reconstruction results of noisy time-varying data. As indicated in Fig. 8, our STLOP method produces a faithful reconstruction. This example consists of 201 frame surfaces. The time interval among 201 frame surfaces is 42,342 seconds. The number of points in each frame is not identical, and the averaged number of points in each frame is 97,489. We also present reconstructed results using the accelerated STLOP method in the fourth row, where the average point number of each frame is 5,386. The average 13,000 samples per frame are used for reconstruction. Note that in this example, as the involved deformation is not large, we do not employ the surface registration techniques. As shown in Fig. 8, in the second row, results by independent application of FLOP to individual frames are not temporal-coherent (see the accompanying video). In addition, using the same parameters for each frame, there exit small holes in some of the reconstructed frames since no parameter is suitable for all frames. Using our STLOP method, the results are much better. Fig. 12 gives reconstruction results from the scanned noisy data. We obtain the scanned noisy data using a Kinnect. This example consists of 201 frame surfaces, and the averaged number of points in each frame is 67,359. Using the STLOP method, the results are temporal-coherent and the features are preserved well.

Fig. 9 and Fig. 13 present the reconstruction results of noisy time-varying data with large deformation. Since these examples involve low FPS capturing and/or large deformations, we have to employ the surface registration to produce spatially-continuous and temporally-coherent reconstruction results. By intergrading the surface matching results [34] into the STLOP operator,

Data set	size	Time	$\|J\|/\|K\|$	Normal estimation	KDE sampling	Signature computing	Optimization	Acceler time	Acceler ratio
Venus	65,235	410	8	7.4	0.4	0	9.3	17.1	24
Armadillo	1354,321	1901	91	11.2	0.7	0	17.8	29.7	64
Hand	221,638	509	16	10.1	0.3	0	13.3	23.7	22
Scangirl	463,245	1312	8	18.8	0.4	0	32.7	51.9	25
Warrior	323,158	863	16	9.7	0.2	0	13.6	23.5	37
Face sequences	$97,489 \times 201$	42342	18	27.3	1.9	15.4	102.1	146.7	288
Elephant to horse	$181,367 \times 201$	56732	15	41.4	2.1	32.7	133.5	209.7	271

Table 1: Performance (in seconds) with and without our accelerated FLOP and STLOP for representative data sets. The evaluation was done on a PC equipped with Pentium (R) Dual-Core CPU E5200@2.50GHz with 2GB RAM. Time column contains total timings without any acceleration, Acceler time column contains total timings using proposed acceleration method.

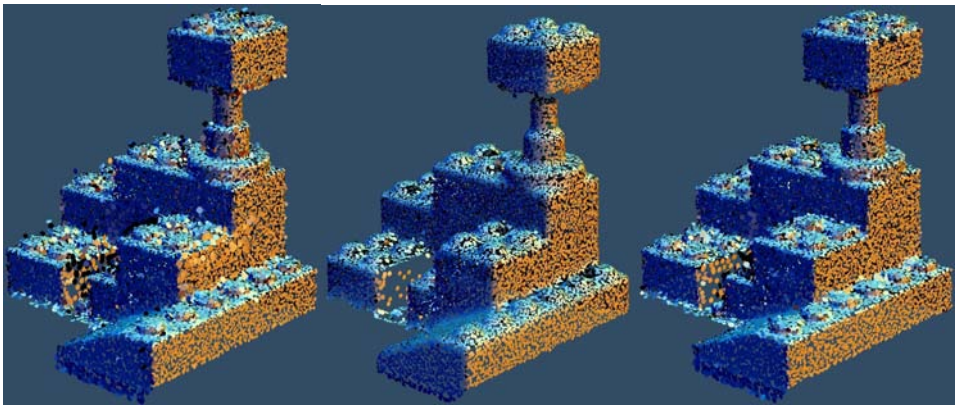
the features of the time-varying surfaces are reconstructed better than the results reconstructed by performing LOP for each frame independently. As shown in Fig. 9, the trunk is better reconstructed. The time-varying data consists of 201 frame surfaces, and the averaged number of points in each frame is 181,367.

The bilateral weight average algorithm is applied to determine h in our paper. The default value of R used to determine h is set as $R = 6\sqrt{d_{bb}/\,|J|}$. The parameter μ used in Equation 2 to compute the new projected point $x_{i'}^{(k+1)}$ is set as 0.6 in Fig.8 and Fig.9, and set as 0.45 in all other examples. For example, in Fig.8, the parameter σ in ψ_{dt} is set as 2, and σ_r in ϕ_{dt} is set as 0.18. In Fig. 9, σ in ψ_{dt} and σ_r in ϕ_{dt} are set as 2 and 0.21, respectively.

The complexity of our accelerated FLOP or STLOP techniques depends on sampling factor $|J|/|K|$. When the $|J|/|K|$ is large, our method shows much advantage. Table 1 shows the timing statistics for several models. Our accelerated method shows greater advantage when processing large data sets that cannot be efficiently processed for complete sets of data. For example, it takes less than 2.1 seconds to downsample $181,367 \times 201$ points on CPU using $|J|/|K| = 15$, and the average projected points for each frame are 12,256. However, the reconstruction results are still satisfactory even with such a high sampling factor, as shown in Fig.9.

Limitations: Compared with parameterization-free LOP [7], one limitation of our method is that, to define bilateral weighted LOP operator, we need the normal information. Even the methods [1, 13, 30] make good normal estimation, it is still difficult to compute accurate normal vectors for sparsely sampled models with sharp features and heavy outliers. Fortunately, unoriented normals can be used in our method. This is a very important and

it mitigates the limitation that needs normal information, since computing the normal direction is a quite easy task compared to consistently orienting them.



(a) (b) (c)

Figure 14: (a) The scanned point set, (b) reconstruction result using LOP [7], (c) result of our algorithm.

Another disadvantage of requiring normal estimation on projected points for feature-preserving reconstruction is the reliance for the well-distributed initial projected points. In other words, our FLOP method cannot simply project ANY set of points in space onto the input point set, while which is a notable feature of LOP and WLOP.

For sparsely sampled models with heavy outliers, it might be difficult to completely smooth out the outliers while preserving the sharp features perfectly. As illustrated in Fig.14, although the features are preserved well, the reconstructed results are not smooth enough. In this case, we have to make leverage between features-preserving results and smooth results with outliers smoothed away. Compared with our results, the results of LOP method are smoother, while the features are not preserved well.

6. Conclusion and future work

In this paper, we present an efficient and feature-preserving locally optimal projection operator (FLOP) for geometry reconstruction. It is based on a bilateral-weighted LOP, considering both spatial and geometric feature

information . We show that FLOP can be greatly accelerated by using the random sampling of the Kernel Density Estimate (KDE) and can be extended for efficient and faithful reconstruction of time-varying surfaces reconstruction.

In the future, we would like to find a novel method for more robust normal estimation to handle heavily noisy surfaces. It would be interesting to explore solutions (e.g., based on [38]) to further accelerate the bilateral weighted LOP. Another possible future work is to perform geometry completion based on the local minimization LOP, and appropriate user interaction may be incorporated to generate desirable results. Finally, for more compact representation, we are interested in producing adaptive geometry reconstruction using the LOP method, where samples are adaptively distributed with respect to features.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and insightful suggestions, and thank to Zhongyi Du for his help and discussions in making this work possible. The authors also thank Robert W. Sumner, Li Zhang and many other researchers for presenting the mesh and point cloud data. This work was partly supported by the National Basic Research Program of China (No. 2012CB725303), NSFC (No. 61070081, No.41271431), the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1208), Luojia Outstanding Young Scholar Program of Wuhan University, the Project of Science and Technology Plan for Zhejiang Province (Grant No. 2012C21004), and the Fundamental Research Funds for the Central Universities. Chunxia Xiao is the corresponding author.

References

[1] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. SIGGRAPH 1992;26.

[2] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: SIGGRAPH. 1998, p. 415–21.

[3] Levoy M, Pulli K, Curless B, Rusinkiewicz S, Koller D, Pereira L, et al. The digital michelangelo project: 3D scanning of large statues, Siggraph 2000. SIGGRAPH 2000;:131–44.

[4] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H. Multi-level partition of unity implicits. In: ACM SIGGRAPH 2003. ACM; 2003, p. 463–70.

[5] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Eurographics symposium on Geometry processing. 2006,.

[6] Xiao C, Zheng W, Miao Y, Zhao Y, Peng Q. A unified method for appearance and geometry completion of point set surfaces. The Visual Computer 2007;23(6):433–43.

[7] Lipman Y, Cohen-Or D, Levin D, Tal-Ezer H. Parameterization-free projection for geometry reconstruction. ACM Transactions on Graphics 2007;26(3):22.

[8] Guennebaud G, Gross M. Algebraic point set surfaces. ACM Transactions on Graphics (TOG) 2007;26(3):23.

[9] Hoppe H, DeRose T, Duchamp T, Halstead M, Jin H, McDonald J, et al. Piecewise smooth surface reconstruction. In: SIGGRAPH. ACM; 1994, p. 295–302.

[10] Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C. Computing and rendering point set surfaces. IEEE Transactions on Visualization and Computer Graphics 2003;:3–15.

[11] Carr J, Beatson R, Cherrie J, Mitchell T, Fright W, McCallum B, et al. Reconstruction and representation of 3D objects with radial basis functions. In: SIGGRAPH. ACM; 2001, p. 67–76.

[12] Levin D. Mesh-independent surface interpolation. Geometric modeling for scientific visualization 2003;3:37–49.

[13] Huang H, Li D, Zhang H, Ascher U, Cohen-Or D. Consolidation of unorganized point clouds for surface reconstruction. ACM Transactions on Graphics (TOG) 2009;28(5):1–7.

[14] Freedman D, Kisilev P. Fast Mean Shift by compact density representation. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2009, p. 1818–25.

[15] Guennebaud G, Germann M, Gross M. Dynamic sampling and rendering of algebraic point set surfaces. In: Computer Graphics Forum; vol. 27. Wiley Online Library; 2008, p. 653–62.

[16] Xiao C. Multi-level partition of unity algebraic point set surfaces. Journal of Computer Science and Technology 2011;26(2):229–38.

[17] Alexa M, Adamson A. On normals and projection operators for surfaces defined by point sets. In: Proceedings of Symposium on Point-Based Graphics; vol. 4. 2004, p. 149–55.

[18] Fleishman S, Cohen-Or D, Silva C. Robust moving least-squares fitting with sharp features. In: ACM SIGGRAPH 2005. ACM; 2005, p. 544–52.

[19] WEISZFELD E. Sur le point pour lequel la somme des distances de points dennes est minimum. Tohoko Math 1937;43:355–86.

[20] Fong P, Buron F. Sensing deforming and moving objects with commercial off the shelf hardware. In: CVPR Workshops. IEEE; 2005, p. 101.

[21] Weise T, Leibe B, Van Gool L. Fast 3d scanning with automatic motion compensation. In: CVPR'07. IEEE; 2007, p. 1–8.

[22] Schall O, Belyaev A, Seidel H. Adaptive feature-preserving non-local denoising of static and time-varying range data. Computer-Aided Design 2008;40(6):701–7.

[23] Buades A, Coll B, Morel J. A non-local algorithm for image denoising. In: CVPR 2005; vol. 2. IEEE; 2005, p. 60–5.

[24] Wand M, Jenke P, Huang Q, Bokeloh M, Guibas L, Schilling A. Reconstruction of deforming geometry from time-varying point clouds. In: Eurographics symposium on Geometry processing. 2007, p. 58.

[25] Mitra N, Flöry S, Ovsjanikov M, Gelfand N, Guibas L, Pottmann H. Dynamic geometry registration. In: Eurographics symposium on Geometry processing. Eurographics Association; 2007, p. 182.

[26] Liao B, Xiao C, Jing L. Efficient Feature-preserving Local Projection Operator for Geometry Reconstruction. In: EUGRAPHICS 2011, short paper. Eurographics Association; 2011, p. 13–6.

[27] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising. ACM Transactions on Graphics (TOG) 2003;22(3):950–3.

[28] Jones T, Durand F, Desbrun M. Non-iterative, feature-preserving mesh smoothing. ACM Transactions on Graphics 2003;22(3):943–9.

[29] Adamson A, Alexa M. Anisotropic point set surfaces. In: Computer Graphics Forum; vol. 25. Wiley Online Library; 2006, p. 717–24.

[30] Mitra N, Nguyen A. Estimating surface normals in noisy point cloud data. In: Proceedings of the nineteenth annual symposium on Computational geometry. ACM. ISBN 1581136633; 2003, p. 328.

[31] Xiao C, Miao Y, Liu S, Peng Q. A dynamic balanced flow for filtering point-sampled geometry. The Visual Computer 2006;22:210–9.

[32] Kalogerakis E, Nowrouzezahrai D, Simari P, Singh K. Extracting lines of curvature from noisy point clouds. Computer-Aided Design 2009;41(4):282–92.

[33] Öztireli A, Guennebaud G, Gross M. Feature preserving point set surfaces based on non-linear kernel regression. In: Computer Graphics Forum; vol. 28. Wiley Online Library; 2009, p. 493–501.

[34] Li X, Guskov I. Multi-scale features for approximate alignment of point-based surfaces. In: SGP. 2005,.

[35] Liao B, Xiao C, Liu M, Dong Z, Peng Q. Fast hierarchical animated object decomposition using approximately invariant signature. The Visual Computer 2012;28(4):387–99.

[36] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vision 2004;60:91–110.

[37] Group TVC. Meshlab@ONLINE. 2012. URL `http://meshlab.sourceforge.net/`.

[38] Adams A, Gelfand N, Dolson J, Levoy M. Gaussian kd-trees for fast high-dimensional filtering. ACM Transactions on Graphics (TOG) 2009;28(3):1–12.