Non Local Point Set Surfaces

Thierry Guillemot
Telecom ParisTech - CNRS - LTCI
Paris, France
thierry.guillemot@telecom-paristech.fr

Andres Almansa
Telecom ParisTech - CNRS - LTCI
Paris, France
andres.almansa@telecom-paristech.fr

Tamy Boubekeur
Telecom ParisTech - CNRS - LTCI
Paris, France
tamy.boubekeur@telecom-paristech.fr

Abstract—We introduce a non local point set surface model for meshless geometry processing. Compared to previous approaches, our model better preserves features by exploiting self-similarities present in natural and man-made 3D shapes. The basic idea is to decompose 3D samples into scalar displacements over a coarse smooth domain. Then, considering the displacement field stemming from the local neighboring set of a given point, we collect similar functions over the entire model and define a specific displacement value for the point by the mean of similarity-based weighted combination of them. The underlying scale-space decomposition allows for a wide range of similarity metrics, while scalar displacements simplify rotation-invariant registration of the local sample sets. Our contribution is a non local extension of all previous point set surface models, which (i) improves feature preservation by exploiting self-similarities, if present, and (ii) boils down to the underlying (local) point set surface model, when self-similarities are not strong enough. We evaluate our approach against state-of-the-art point set surface models and demonstrate its ability to better preserve details in the presence of noise and highly varying sampling rates. We apply it to several data sets, in the context of typical point-based applications.

Keywords-Point Set Surfaces; Non-Local methods; Reconstruction; Filtering;

I. INTRODUCTION

With the democratization of 3D sensors, generating high resolution surfaces from real objects has never been so easy. The typical output of the capture stage comes as an unstructured 3D point set, merged by aligning several range maps, and exhibits noise, holes, outliers and quickly varying sampling ratios. These defects influence strongly the quality of the subsequent surface mesh reconstruction, which motivates the use of intermediate mesh-less representations. Thus, the so-called *point-based* geometry processing methods have been progressively introduced in the pipeline, to improve as much as possible the data quality before taking any decision on its (e.g., meshed) structure, such as its global topology or local connectivity.

To fill the gap between unstructured raw point clouds and high quality meshes, *Point Set Surfaces* (PSS) and its numerous variants have proved to be instrumental in the definition of mesh-less surface models. Given a 3D point cloud, where each sample contains a 3D position and, potentially, a normal vector, the most popular PSS models are defined as the stationary set of $\mathbb{R}^3$ under a moving least square (MLS) scattered data approximation [23] which fits locally a simple primitive (e.g., plane, sphere) to the cloud.

Various extensions of the seminal PSS definition [5] have been proposed, to improve robustness in the presence of holes [17], better preserve shape convexity [4] or singular features such as sharp creases [14]. However, all these models share the same basic idea by defining the surface at a given location according to the local variation of the point set around this location only. Unfortunately, low sampling rates and noise stricly bound the amount of information which can be reconstructed from such a purely local point of view.

This problem bares similarities with image denoising, which may also locally lack information to reconstruct a smoother image while preserving important features convincingly. Indeed, one elegant solution to this ill-posed problem is to consider the self-similarity present in natural images with a *non-local mean* (NLM) filtering process [9]: the basic idea is to reconstruct the signal at a point using potentially distant samples having a similar local structure (i.e., neighborhood).

Not only image denoising, but also more ill-posed image resolution enhancement problems [33], [27] benefit from non-local image reconstruction models (like in inpainting [7] or random sampling [13])

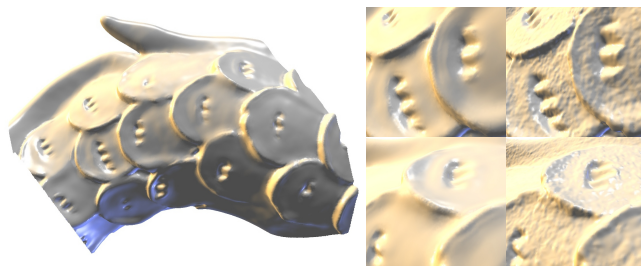This inspires our work and we propose a new PSS model

Figure 1. Left: Scanned point set Dragon (413k points) reconstructed using our NLPSS definition. Right: Two closeup views of the NLPSS reconstruction (left) and original scanned data (right).

which extends all existing ones by introducing a non-local formulation able to exploit the surface self-similarity present in natural and artificial shapes. Our Non-Local PSS (NLPSS) model is defined by means of a weighted mean of local surface patch descriptors, where weights are related to an in-

trinsic similarity measure between pairs of such descriptors. A scale-space analysis allows to go from the unstructured point cloud to a coarsely structured surface where intrinsic surface descriptors can be defined, that capture even the finest scale details present in the point cloud.

II. BACKGROUND

Point Set Surfaces.: Moving Least Squares (MLS) are a class of functional scattered data approximation methods [29] extended by Levin [22], [23] to the case of shapes approximation and used by Alexa [5] to define a mesh-less surface representation: Point Set Surfaces (PSS). Essentially, they are defined as the stationary set of an iterative projection operator. At each step, a plane is fitted locally near the evaluation point by a nonlinear minimization of the square distance to the samples. A bivariate polynomial, parameterized over the plane, is then fitted to the samples and the projection of the evaluation point onto this local surface is used to initiate the next step. This simple projection procedure can be used to remove noise, classify outliers and resample the point cloud. Its implicit form allows to extract a polygonal surface using various meshers, such as the Marching Cubes algorithm [24] or the Restricted Delaunay Triangulation [8].

Amenta et al.[6] showed that the non-linear minimization and the polynomial approximation were not mandatory to obtain a valid PSS: instead, one can simply average positions, while still providing a smoothly varying signed distance function, thus allowing for an implicit representation of the surface. Adamson et al. exploited this idea in the *Simple Point Set Surface* (SPSS) model [2], considering samples embedding normal vectors, and estimating local projection planes with simple weighted local combinations of positions and normals. As usual in point-based graphics, a local principal component analysis [18] allows to equip samples with normal estimates.

When the point cloud is not dense enough w.r.t. the operator support size, the plane fitting procedure becomes unstable. Guennebaud et al. addressed this problem by replacing the plane by an algebraic sphere [17], introducing the Algebraic Point Set Surface (APSS) model, later improved to constrain the gradient of the algebraic spheres to samples's normals [16].

The problem of sharp edge preservation was addressed by [14], introducing an implicit definition of the MLS projection operator (IMLS). While most PSS models have a tendency to shrink volumes and do not allow to control the local convexity of the resulting surface, the Hermite Point Set Surfaces [4] (HPSS) model solves this problem with an hermite combination scheme by considering the projection of the evaluation point on each local plane defined by neighboring sample's position and normal vectors. Finally, the Robust Implicit Moving Least Square (RIMLS) operator [26] uses a kernel regression method allowing to

adjust iteratively the weight of each sample. As a result, outliers have less impact and features are better preserved than with previous PSS variants.

All these local PSS models define the reconstruction scale – and therefore the smoothing effect – with the support size of the spatial kernel used to weight neighboring samples contribution to the projection primitive fitting procedure. This kernel may either be gaussian or polynomial [32].

Our approach is orthogonal to these models in the sense that it can extend any of them by bringing non-local sample information in the evaluation of a point, exploiting the object self-similarity when available while degenerating to the original operator otherwise.

Non-Local Filtering.: Feature-preserving denoising is usually addressed efficiently using the *Bilateral Filter*, originally introduced for images [30] and which uses both spatial and range proximity weights to perform a local sample combination at each pixel. The resulting anisotropic diffusion process better preserves edges than pure low-pass (e.g., Gaussian) filters and can be defined for meshes [15], [19] and point sets [20] as well. Unfortunately, using weights based on the similarity between neighborhing samples is not reliable in a number of configurations – in particular when using large support sizes – and leads to overflow phenomena above edges.

Buades et al. [9] address this problem by introducing the notion of *Non-Local Means* [9] (NLM) in the context of image filtering. The basic idea is to filter a pixel by averaging its value with the pixels having a similar neighborhood. It assumes that images are self-similar and that the noise has zero mean. This approach and its extensions offer better results than local filters, such as the bilateral one, with numerous classes of images. Extending this class of filters to 3D surfaces requires the definition of suitable local descriptors and similarity measures thereof. Adams et al. [1] propose to define the descriptors as sets of local density histograms over a noisy signal, where the latter is in turn defined as the difference between the original surface and a smoother (e.g., laplacian-filtered) version. Wang et al. [31] define a local frame using the covariance matrix. They use a combination of prefiltered points weighted by the similarity of patches defined by surface interpolations. Meanshift is used to reduce the neighborhood size of the non-local combination. Morigi et al. [25] define a non-local surface diffusion flow with the mean curvature values as patch values.

Due to the lack of any structuring information, adapting non-local means denoising to point sets is a difficult task. Schall et al. [28] address this problem by using a regular grid and apply a similar scheme as the non local image filtering.

Deschaud et al. [10] generalise to any point set by considering a local frame defined by two eigenvectors corresponding to the largest eigenvalues of the local covariance matrix. The descriptor is defined by the vector of the MLS
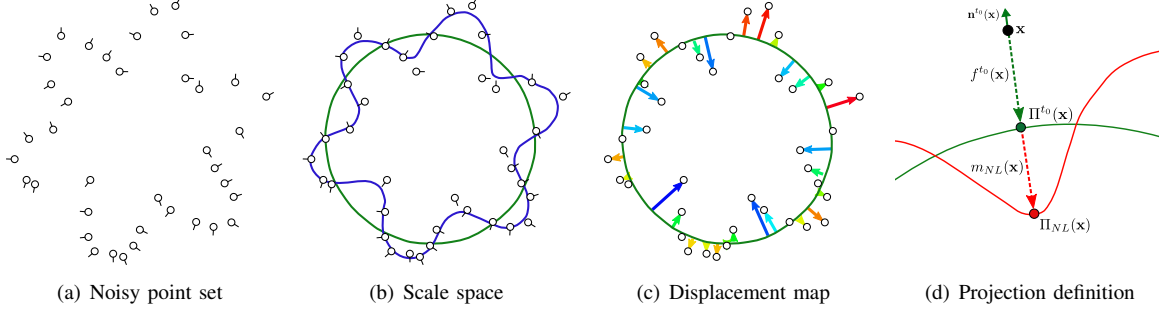
(a) Noisy point set   (b) Scale space   (c) Displacement map   (d) Projection definition

Figure 2. *Summary of our algorithm:* Starting from *(a)* a noisy point set $\mathcal{P}$ containing points $\mathbf{p}_i$ and their normals $\mathbf{n}_i$, we define *(b)* two surfaces from the scale-space of a given PSS operator: a coarse-scale surface S_{t_0} (in green) and a fine-scale surface S_{t_1} (in blue). *(c)* The coarse scale decomposes $\mathcal{P}$ into a base surface and a noisy scalar displacement $m(\mathbf{p}_i)$ beween S_{t_0} and $\mathbf{p}_i$. *(d)* The projection $\Pi_{NL}(\mathbf{x})$ of a point $\mathbf{x}$ over the non-locally reconstructed surface (in red) is obtained by first projecting $\mathbf{x}$ over S_{t_0}, and then extending this projection along the same direction, by an amount $m_{NL}(\mathbf{x})$ that is obtained as a weighted average of the scalar displacement values $m(\mathbf{p}_i)$, where larger weights are allocated to the points $\mathbf{p}_i$ whose neighborhood is similar to that of x. Comparing such neighborhoods requires the construction of patches around each point $\mathbf{p}_i \in \mathcal{P}$ and a similarity measure between them, as detailed in figure 3. Note that the scalar displacement in *(c)* is only known at irregular locations. Here the fine-scale surface S_{t_1} becomes instrumental, since it allows to resample all patches in a normalized local grid.

regression coefficients. Observing that such a parametrization is not stable enough to be used efficiently, Digne et al. [11] propose to decompose the point set by an iterative mean curvature motion filter into a smooth signal and a height vector field. Only the latter is used (after radial-basis function interpolation) when comparing two patches. Thus, the resulting similarity measure is only sensitive to local variations of the surface.

Our non local surface definition is more general than the previous approaches: we do not restrict our work to surface mesh denoising or point set denoising in the sense that we propose a new PSS operator, suitable for reconstruction, filtering, enhancement and more general processing of unorganized point sets. Consequently, point set denoising is an interesting property of our surface definition but can not be seen as the final purpose of our work.

III. NON LOCAL PROJECTION

Throughout the paper we consider as input a set of samples $\mathcal{P} = \{\mathbf{p}_i, \mathbf{n}_i\}$ with $\mathbf{p}_i \in \mathbb{R}^3$ the sample's position and $\mathbf{n}_i \in \mathbb{R}^3$ its normal vector. Let's consider $\mathbf{x} \in \mathbb{R}^3$ the evaluation point. We consider the almost orthogonal MLS projection as defined by Adamson and Alexa [3]:

$$MLS_{\mathcal{P}} : \mathbb{R}^3 \to \mathbb{R}^3, \mathbf{x} \to \Pi(\mathbf{x}) \qquad (1)$$

They are based on a projection function $\Pi(\mathbf{x})$ of $\mathbf{x}$ onto a locally weighed least squares primitive Q (e.g., plane for SPSS, algebraic sphere for APSS):

$$\Pi(\mathbf{x}) = \mathbf{x} - f(\mathbf{x})\mathbf{n}(\mathbf{x}) \qquad (2)$$

$f(\mathbf{x})$ is the implicit distance between $\mathbf{x}$ and its projection on Q and $\mathbf{n}(x)$ is the stationary MLS normal defined by Q at this location. The scale at which Q is fitted to $\mathcal{P}$ w.r.t. $\mathbf{x}$ is typically controlled by a parameter t, which relates to the support size of some underlying spatial weighting kernel.

This general definition can be instantiated for the case of SPSS for instance with:

$$\mathbf{n}(x) = \frac{\sum_{\mathbf{p}_i \in \mathcal{P}} w_t(\mathbf{x}, \mathbf{p}_i)\mathbf{n}_i}{||\sum_{\mathbf{p}_i \in \mathcal{P}} w_t(\mathbf{x}, \mathbf{p}_i)\mathbf{n}_i||}$$

$$f(x) = \langle \mathbf{x} - \mathbf{c}(x), \mathbf{n}(x) \rangle$$

$$\mathbf{c}(x) = \frac{\sum_{\mathbf{p}_i \in \mathcal{P}} w_t(\mathbf{x}, \mathbf{p}_i)\mathbf{p}_i}{\sum_{\mathbf{p}_i \in \mathcal{P}} w_t(\mathbf{x}, \mathbf{p}_i)}$$

and $w_t(\mathbf{x}, \mathbf{p}_i)$ a weighting kernel. We typically choose $w_t(\mathbf{x}, \mathbf{p}_i) = \frac{1}{Z_t(\mathbf{x})}g(\frac{1}{t}\|\mathbf{x} - \mathbf{p}_i\|)$ with g a compactly supported, piecewise polynomial function and $Z_t(\mathbf{x})$ chosen to ensure unit sum $\sum_{\mathbf{p}_i \in \mathcal{P}} w_t(\mathbf{x}, \mathbf{p}_i) = 1$. The parameter t directly controls the scale captured by each MLS projection and can somehow be seen as a filtering parameter: varying t decomposes the PSS at different scales. We call respectively $\Pi^t(\mathbf{x})$, $f^t(\mathbf{x})$ and $\mathbf{n}^t(\mathbf{x})$ the projection operator, the implicit distance and the normal defined by this PSS operator at scale t.

A. General Structure of the Algorithm

With a sufficiently large scale t, we can decompose the PSS into a coarse smooth surface S_{t_0} and a fine residual scalar displacement field $m(\mathbf{p}_i)\mathbf{n}^{t_0}(\mathbf{p}_i)$ containing features contaminated by noise.

Our key idea is to use a non local method to remove the noise of the residual scalar field, and simultaneously compute the projection (interpolation) at a fine scale for any point $\mathbf{x}$.

We can use any PSS operator to define the coarse surface S_{t_0} at the large scale t_0. Then the NLPSS operator is defined by adding to S_{t_0} a fine scalar displacement field $m_{NL}(\mathbf{x})\mathbf{n}^{t_0}(\mathbf{x})$, where $m_{NL}(\mathbf{x})$ is defined as a non local approximation of the residual distance $m(\mathbf{x})$ between the
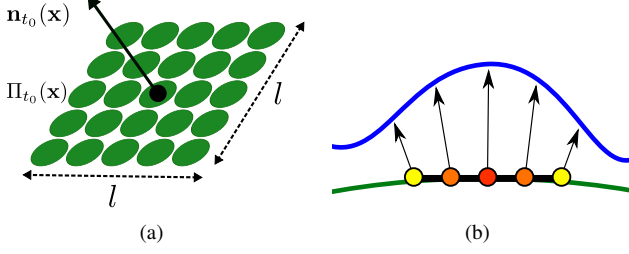
Figure 3. *Patch definition: (a)* Creation of patch of size l centered in $\Pi_{t_0}(\mathbf{x})$ sampled on a tangent plane defined by $\mathbf{n}_{t_0}(\mathbf{x})$. *(b)* The patch values are computed using S_{t_1} (in blue). Color values represent the local differences between S_{t_0} and S_{t_1}.

coarse surface and a reconstruction of $\mathcal{P}$. Consequently, we express our NLPSS projection as:

$$\Pi_{NL}(\mathbf{x}) = \Pi^{t_0}(\mathbf{x}) - m_{NL}(\mathbf{x})\mathbf{n}^{t_0}(\mathbf{x}) \qquad (3)$$

All steps of our algorithm are summarized in figure 2.

B. Displacement map definition

The scalar displacement field or residual distance between the coarse surface S_{t_0} and $\mathcal{P}$ is defined for $\mathbf{x} = \mathbf{p}_i \in \mathcal{P}$ as

$$m(\mathbf{p}_i) = \left\langle \mathbf{p}_i - \Pi^{t_0}(\mathbf{p}_i), \mathbf{n}^{t_0}(\mathbf{p}_i) \right\rangle = f^{t_0}(p_i) \qquad (4)$$

Such a distance is defined only for $\mathbf{p}_i \in \mathcal{P}$ and can be noisy.

In order to denoise these values and to extend this definition to any point $\mathbf{x}$ we consider a non local weighted average of $m(\mathbf{p}_i)$:

$$m_{NL}(\mathbf{x}) = \sum_{\mathbf{p}_i \in \mathcal{P}} w_{NL}(\mathbf{x}, \mathbf{p}_i) m(\mathbf{p}_i) \qquad (5)$$

where $w_{NL}(\mathbf{x}, \mathbf{p}_i)$ is a similarity measure between the local neighborhoods of the two points $\mathbf{x}$ and $\mathbf{p}_i$.

C. Similarity measure definition

In order to compare point cloud patches we shall resample them in a normalized fashion.

In order to construct a patch descriptor for the scalar field centered in $\mathbf{x}_{t_0} = \Pi^{t_0}(\mathbf{x})$, we define a normalized 2D coordinate system on the plane tangent to S_{t_0} at x_{t_0} with the horizontal axis pointing in a normalized orientation $\theta(x_{t_0})$ to be specified shortly. After suitable rescaling of this coordinate system we obtain a set $D = [-n, n]^2 \cap \mathbb{Z}^2$, of $(2n + 1) \times (2n + 1)$ points located on a $l \times l$ square For $(i, j) \in D$, each point $\mathbf{x}_{i,j}$ of the patch has a value $M_{\mathbf{x}_{t_0}}(i, j)$ corresponding to the displacement value $m(\mathbf{x}_{i,j})$.

As explained previously, we need to reconstruct a specific value at each $\mathbf{x}_{i,j}$. We propose to use the PSS S_{t_1} at fine scale $t_1 \ll t_0$ and define $M_{\mathbf{x}_{t_0}}$ as :

$$M_{\mathbf{x}_{t_0}}(i, j) = f^{t_1}(\mathbf{x}_{i,j}) \quad \text{for } (i, j) \in D. \qquad (6)$$

$M_{\mathbf{x}_{t_0}}$ represents the increment (with respect to the coarse surface S_{t_0}) of a fine scale surface patch of S_{t_1} which is a minimally denoised and interpolated version of $\mathcal{P}$.

We choose $l \approx \frac{t_0}{3}$ so that the distance between the points $\mathbf{x}_{i,j}$ and S_{t_0} is negligible compared to the values $M_{x_{t_0}}$. The definition is summarized in figure 3.

Finally in order to choose $\theta(x_{t_0})$ in such a way that the patch $M_{x_{t_0}}$ is rotation invariant, we define an arbitrarily oriented set $\tilde{D}$ and the corresponding patch $\tilde{M}_{x_{t_0}}$. This allows to compute the intrinsic alignment angle $\theta(x_{t_0})$ in the surface tangent plane (i.e., around the surface normal) by considering the gradient at $\mathbf{x}_{t_0}$:

$$\begin{pmatrix} dG_u(\mathbf{x}_{t_0}) \\ dG_v(\mathbf{x}_{t_0}) \end{pmatrix} = \sum_{i,j} \begin{pmatrix} -i \\ -j \end{pmatrix} \mathrm{e}^{\left(-\frac{i^2+j^2}{\sigma_l^2}\right)} \tilde{M}_{\mathbf{x}_{t_0}}(i, j)$$

$$\theta(\mathbf{x}_{t_0}) = atan2(dG_u(\mathbf{x}_{t_0}), dG_v(\mathbf{x}_{t_0})) \qquad (7)$$

where $\sigma_l \approx 3l$ is the variance of the gaussian function and $atan2$ is the arctangent function that returns an angle between $[0, 2\pi[$. Let's consider $\mathbf{y}, \mathbf{z} \in \mathbb{R}^3 \times \mathbb{R}^3$. We define the distance between two patches by:

$$d(\mathbf{y}, \mathbf{z}) = \frac{1}{2n+1} \| M_{\mathbf{y}_{t_0}} - M_{\mathbf{z}_{t_0}} \|_2 \qquad (8)$$

and the corresponding non local weight by :

$$w_{NL}(\mathbf{y}, \mathbf{z}) = \frac{1}{Z_{NL}(\mathbf{y})} \mathrm{e}^{\left(-\frac{d(\mathbf{y},\mathbf{z})^2}{h^2}\right)} \qquad (9)$$

where h is a parameter that defines how similar two patches are, and the normalization constant $Z_{NL}(\mathbf{y})$ ensures that $\sum_{\mathbf{p}_i \in \mathcal{P}} w_{NL}(\mathbf{y}, \mathbf{p}_i) = 1$. The non local weights of all point descriptors to a reference descriptor are shown in figure 4. Finally, we define our NLPSS operator as :

$$\Pi_{NL}(\mathbf{x}) = \mathbf{x} - f_{NL}(\mathbf{x})\mathbf{n}^{t_0}(\mathbf{x}) \qquad (10)$$

where $f_{NL}(\mathbf{x})$ corresponds to the implicit distance:

$$f_{NL}(\mathbf{x}) = f^{t_0}(\mathbf{x}) - \sum_{\mathbf{p}_i \in \mathcal{P}} w_{NL}(\mathbf{x}, \mathbf{p}_i) f^{t_0}(\mathbf{p}_i) \qquad (11)$$

Interestingly enough, our formulation can be easily modified to boil down to a PSS projection, in the absence of strong self-similarities in the data (which is rare). This
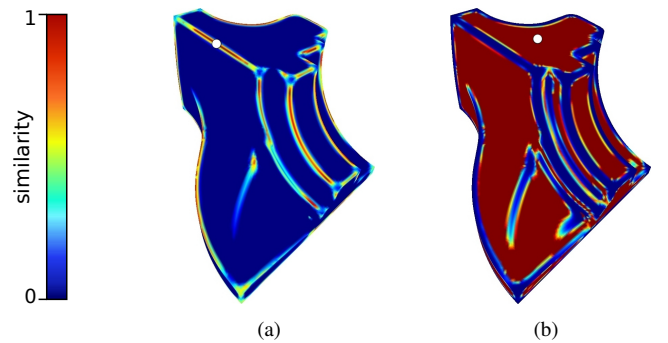

Figure 4. Similarity function to a point on an edge *(a)* and a point on a plane *(b)*, indicated in grey in each case. Color values represent how similar are all other points, from red (very similar) to blue (poorly similar).

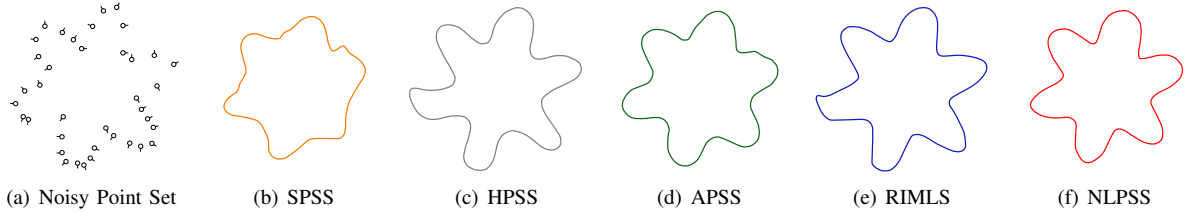| (a) Noisy Point Set | (b) SPSS | (c) HPSS | (d) APSS | (e) RIMLS | (f) NLPSS |

Figure 5.   Various reconstructions of the 2D star noisy point set.

is achieved by adding to the non-local weighting kernel w_{NL}, a spatial distance kernel w_t (e.g., Wendland's kernel) multiplied by a small constant α.

$$\tilde{w}_{NL}(\mathbf{x}, \mathbf{p}_i) = \frac{1}{Z(\mathbf{x})}(\alpha Z_t(\mathbf{x})w_t(\mathbf{x}, \mathbf{p}_i) + Z_{NL}(\mathbf{x})w_{NL}(\mathbf{x}, \mathbf{p}_i))$$

As usual the normalization constant $Z(\mathbf{x})$ is chosen to ensure unit sum. Observe that when $\mathbf{x}$ is within a highly represented structure the non-local term dominates, thus improving denoising and resolution enhancement, whereas if $\mathbf{x}$ lies on a rare patch, then the spatial term dominates, ensuring better smooth interpolation of holes.

IV. EXPERIMENTAL EVALUATION

A. Implementation and performance

We have implemented our algorithm in C++ using a kd-tree for fast neighborhoods queries in $\mathbb{R}^3$.

MLS operator choice. The above definition of our non local operator requires a PSS projection operator for local smoothing and interpolation at two scales. This provides flexibility, and the results are affected by the choice of this PSS operator. In the sequel we use the APSS [16] for all experiments of NLPSS. This choice provides a good compromise between smoothing at coarse scales and feature preservation at fine scales. More feature preserving operators, like RIMLS were discarded because of artifacts and instabilities that may affect patch comparison.

Coarse scale. The coarse scale t_0 provides a base area for our algorithm and its residual noisy scalar field. It must be selected such that changes in noise affect the scalar field only but not the base surface. The choice for t_0 depends on the amount of noise present in the point set, while ensuring a minimal size concurring to repetitive surface structures. In practice, we choose a scale between 10 and 20 times the local points spacing.

Fine scale. The fine scale t_1 provides a fine approximation of the surface to create patches. For small values, the patches will not sufficiently approximate the point set to be interesting enough. If t_1 is chosen equal to t_0 all patches will be similar. Thus, we should choose this scale sufficiently small (compared to t_0) to approximate the data of the point set. Typically, we choose t_1 between 2 to 4 times the local point spacing.

Non local weight function. Our non local weight function has three parameters: the non-local radius h, the radius l and the number n of points in each resampled patch.

The value h determines the degree of similarity of the point set. The optimal value must be chosen depending on the amount of noise and standard deviation values of scalar fields. For small values, denoising will be poor, since the scalar field will tend to take the value of the nearest neighbor in the patch distance. On the contrary for high values, denoising will be too strong, and the scalar field will take the same value everywhere, thus providing little detail enhancement with respect to the smooth surface S_{t_0} .

The radius l of the patch should be chosen (like the coarse scale t_0) depending on the size of the features present in the point cloud. We cannot choose l too large because the approximation of the coarse surface by a tangent plane would no longer be correct. If we choose l too small, the resulting surface will tend to stick to the values of the nearest points $m_{NL}(\mathbf{p}_i)$. In practice, we choose l between 0.5 and 3 times the local spacing of points.

The number n of points of the patch is used to describe more accurately the local variation of the surface. It must be chosen large enough to adequately describe the surface. The number of points is strongly linked to the speed of the algorithm. In practice, we use patch size 5×5 which is a good time/quality compromise.

Performance. As our algorithm has a quadratic complexity, the number of projections per second is lower than that of conventional PSS algorithms (table I). Several parameters influence this computation time. The projection on the coarse surface at the scale t_0 spends a lot of time because it requires finding a large number of neighbors. Much time is lost due to the creation of patches that must be projected onto the surface defined on a fine scale t_1. Nevertheless, we can accelerate these two steps by using a ball-tree for faster search of nearest neighbors as defined in [16].

The bottle-neck of our algorithm is, as for any non-local

Input point set size	25000	173000	413194
Nb of projections /s	6383	689	143

Table I
ARRAY REPRESENTING THE EVOLUTION OF THE NUMBER OF NLPSS PROJECTIONS PER SECOND RELATIVE TO THE INPUT POINT SET SIZE.
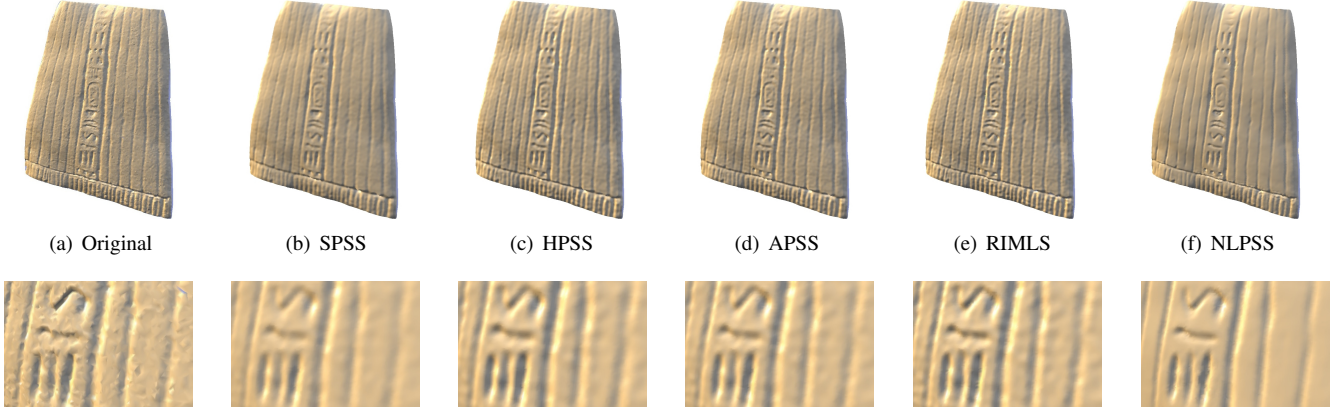
Figure 6. Various reconstructions of the 3D scanned point set Ramesses (50k points). Up: reconstructions, Down: Closeup of the reconstruction.

method, the computation of the weights for each pair of non-local patches. In general, it can be accelerated by reducing the size of the search for nearest neighbors to a smaller neighborhood, and this is a common practice in 2D imaging. Unfortunately in 3D, unlike in the 2D case, this idea causes a loss of quality of the resulting surface.

Several algorithms have proven their ability to speed up this research [1]. We plan to adapt them to our algorithm in future work.

B. Analysis of Quality

Conventional PSS operators try to extract a surface from a local neighborhood. This constraint has obliged them to make a compromise between noise removal and feature preservation.

This compromise becomes apparent in figure 6 where a scanned point set is reconstructed by various PSS operators. We can notice that SPSS tends to smooth features too much, HPSS exaggerates the noise structure of the object. The APSS must be used at small filtering scales to prevent it from completely removing features of the object. Unfortunately at such scales, it cannot completely remove the noise. To prevent excessive smoothing of curved areas while extracting the features of the object, the RIMLS used small filtering scales but this has the effect of exaggerating local noisy structures.

In comparison with these operators, our NLPSS succeeded in generating a surface that is both denoised and feature-preserving.

Our non-local surface definition is more general than the combination a non-local point filtering [10], [11] with an existing surface reconstruction model. The fundamental difference between both approches is visible in Figure 7 where the surface is reconstructed using the two different schemes: non local point denoising followed by APSS reconstruction on one side and our NLPSS surface reconstruction on the other side. Although the first scheme succeeds at moving points while exploiting self-similar information, it cannot reconstruct the surface doing so. On the contrary, our NLPSS models reconstruct the surface at **any** point exploiting self-similarity and redundant structures in the point cloud. Similarly, a third scheme reconstructing a surface first before applying a non-local mesh denoising [1], [31] does not exploit self-similarity at the fullest.

As shown figure 8, in the case of sparse and noisy clouds, existing PSS models can hardly generate a correct surface at small filtering scales. To solve this problem, conventional PSS models increase the level of filtering which has the effect of deleting all the fine-scale information present in the cloud. In contrast our NLPSS is capable of generating a more stable surface. This property is due to the use of *(i)* a coarse-scale base surface that defines the topological structure of the PSS and *(ii)* the non-local averaging of fine details that is capable of denoising them in a feature preserving way, and adding them back to the coarse-surface without introducing spurious surface structures like in other conventional PSS models. Moreover, figure 5 and 8 demonstrates that our definition is very efficient on non uniformly sampled and noisy data.

C. Discussion and Limitations

It is also important to note that the results of our operator will be better for highly self-similar point clouds. However, in the absence of self-similarity our NLPSS will have identical behavior as the underlying local PSS operator used to define it.

Moreover, our non-local definition may be corrupted by oscillations or "halos" around the edges of objects. This problem is common with non-local 2D methods. Recent solutions to this phenomenon are described in [21]. The orientation computation can be noise sensitive for gradients near zero which can be fixed using tensors. Finally, in our current formulation, as the displacement function is represented with a bivariate scalar field, fine and coarse
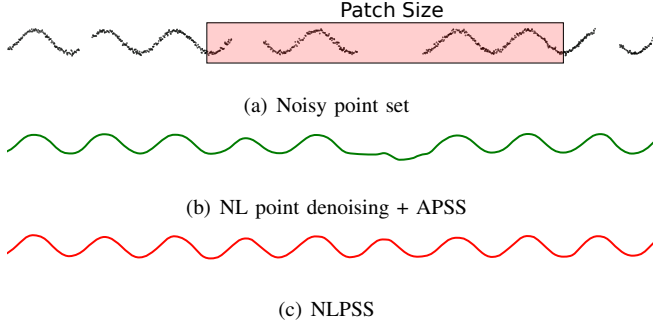
(a) Noisy point set

(b) NL point denoising + APSS

(c) NLPSS

Figure 7. *(a)* A point set corrupted by noise and holes. *(b)* The surface is defined using APSS from a non local denoising of the noisy point set. *(c)* The surface is defined using our NLPSS without pre-denoising. The patch size used by non local algorithms is represented by the red box and the number of samples per patch is 83.

surfaces are expected to be homeomorphic. This usually holds with the typical scales we use in practice.

D. Applications

We propose four different applications for our algorithm.

Filtering. As all PSS algorithms, our NLPSS can be used to define a filtered point set. Thus, for each point $\mathbf{p}_i$ from the noisy point set we associate the filtered version $\Pi_{NL}(\mathbf{p}_i)$. The set of projected points $\Pi_{NL}(p_i)$ corresponds to the filtered point set. Figure 9 shows a result of our NLPSS filtering.

Reconstruction. One of the main interests of PSS operators is that they can define a mesh from the definition of the implicit function associated with this PSS. We define our NLPSS surface as the zero set of the implicit distance f_{NL}. We can compute a mesh of the surface by using a marching cube algorihm. Figure 8(c) shows a result of our NLPSS reconstruction.

Details enhancement. In order to illustrate the potential of our NLPSS in surface editing we consider the following extension of our non-local projection:

$$f_{NL}(\mathbf{x}) = f^{t_0}(\mathbf{x}) - \beta(\mathbf{x}) \sum_{\mathbf{p}_i \in \mathcal{P}} w_{NL}(\mathbf{x}, \mathbf{p}_i) f^{t_0}(\mathbf{p}_i) \quad (12)$$

where $\beta(\mathbf{x})$ is a continuous scalar field that describes for any $\mathbf{x}$ how much of the fine-scale detail $f_{NL}(\mathbf{x})$ is added to the smooth surface S_{t_0}. The user paints over the surface slowly varying scalar values $\beta(\mathbf{x})$ for any $\mathbf{x}$ of the smooth surface.
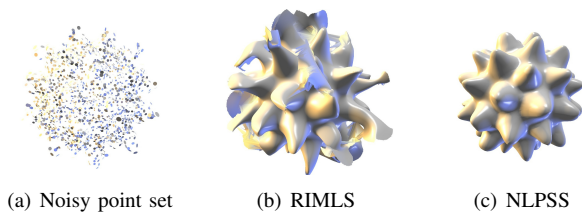

(a) Noisy point set

(b) RIMLS

(c) NLPSS

Figure 8.   Reconstruction of the 3DStar model with RIMLS and NLPSS from a non uniformly sampled and noisy point set.
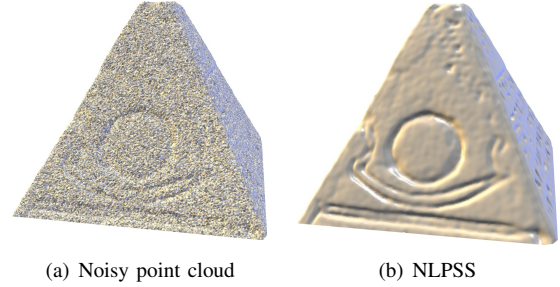

(a) Noisy point cloud

(b) NLPSS

Figure 9.   Pyramid (120k points) from [12] (noisy point cloud) denoised using our NLPSS.

$\beta > 1$ results in detail enhancement, $\beta \in (0, 1)$ results in surface smoothing between S_{t_0} and the NLPSS, whereas negative values of β produce a reversed detail enhancement. An illustration can be seen figure 10.

Outlier detection can be performed by filtering the point set with our NLPSS definition. For any $\mathbf{p}_i$ of the point set we can extract the associated scalar value $f_{NL}(\mathbf{p}_i)$. If we consider a max value f_{max}, outliers can be defined as :

$$\mathbf{p}_{out} = \{\mathbf{p}_i, f_{\mathbf{p}_i} > f_{max})\} \quad (13)$$

The value f_{max} can be set manually to correspond to the desired level of detection.

V. CONCLUSIONS

In the last few years, research in local PSS models and their associated algorithms has been intense and are now reaching a point of maturity by exploiting local structure, and prescribed feature-preservation. In order to take PSS models to the next level, we propose to introduce statistical information on the fine structures present in a particular shape and that we want to preserve.

This work is a first attempt to define a PSS model that learns and exploits non-local self-similarities of the surface. We propose to define a suitable local signal and a similarity measure adapted to unstructured point sets by structuring the point set via a scale-space representation at two scales. The coarse scale provides an intrinsic local reference system, and the fine scale is used for resampling on a regular grid. Once the similarity between local patches is established the reconstructed point set is obtained by a non-local weighted average of the residual distances between points of the point set and the coarse-scale surface.

Our experiments show that our NLPSS definition is more general than non local denoising, better preserves repeating features of the point set, and is very efficient on non uniformly sampled and noisy data.

Our future work will incorporate an explicit consideration of sharp features, acceleration using a suitable indexing structure for patch descriptors and alternative patch similarity metrics. The quality of the reconstruction, as well as the denoising ability of our NLPSS operator can also benefit
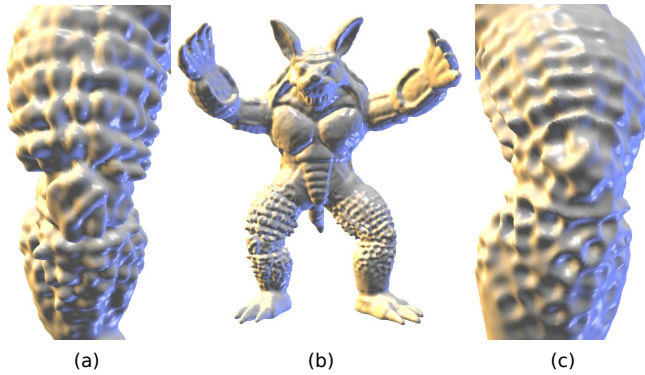
Figure 10. Details enhancement: *(a)* Armadillo (170k points) leg with a positive $\beta(\mathbf{x})$. *(b)* Armadillo models enhanced by our detail enhancement. *(c)* Armadillo leg with a negative $\beta(\mathbf{x})$.

from recent improvements in 2D non-local denoising methods [21]. Adapting such improvements (like aggregation and collaborative filtering) to the 3D case is, however, not a straightforward task.

REFERENCES

[1] A. Adams, N. Gelfand, J. Dolson, and M. Levoy. Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. Graph*, page 21, 2009.

[2] A. Adamson and M. Alexa. Approximating bounded, non-orientable surfaces from points. In *SMI*, pages 243–252, 2004.

[3] M. Alexa and A. Adamson. On normals and projection operators for surfaces defined by point sets. In *Symposium on point-based graphics*, pages 149–155. Citeseer, 2004.

[4] M. Alexa and A. Adamson. Interpolatory point set surfaces - convexity and hermite data. *ACM Transactions on Graphics (TOG)*, 28(2):20, 2009.

[5] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *IEEE Visualization 2001*, pages 21–28. IEEE Computer Society, October 2001.

[6] N. Amenta and Y. J. Kil. Defining point set surfaces. In *ACM*, volume 23 of *ACM Transactions on Graphics (TOG)*, pages 264–270, aug 2004.

[7] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A Variational Framework for Exemplar-Based Image Inpainting. *IJCV*, 93(3):319–347, Jan. 2011.

[8] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of surfaces. *Graph. Models*, 67:405–451, September 2005.

[9] A. Buades and B. Coll. A non-local algorithm for image denoising. In *In CVPR*, pages 60–65, 2005.

[10] J.-E. Deschaud and F. Goulette. Point cloud non local denoising using local surface descriptor similarity. *IAPRS*, 2010.

[11] J. Digne. Similarity based filtering of point clouds. *CVPR 2012*, 2012.

[12] Digne, J. and *et. al.* ,Farman Institute 3D Point Sets. *IPOL*, 2011.

[13] G. Facciolo, P. Arias, V. Caselles, and G. Sapiro. Exemplar-Based Interpolation of Sparsely Sampled Images. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 5681, book part (with own title) 25, pages 331–344. Springer Berlin Heidelberg, 2009.

[14] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, 2005.

[15] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003)*, 22(3):950–953, 2003.

[16] G. Guennebaud, M. Germann, and M. H. Gross. Dynamic sampling and rendering of algebraic point set surfaces. *Comput. Graph. Forum*, 27(2):653–662, 2008.

[17] G. Guennebaud and M. H. Gross. Algebraic point set surfaces. *ACM Trans. Graph.*, 26(3):23, 2007.

[18] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. *Surface reconstruction from unorganized points*. PhD thesis, University of Washington, 1994.

[19] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Trans. Graph.*, 22:943–949, July 2003.

[20] T. R. Jones, F. Durand, and M. Zwicker. Normal improvement for point rendering. *IEEE Comput. Graph. Appl.*, 24(4):53–56, July 2004.

[21] M. Lebrun, M. Colom, A. Buades, and J. Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21:475–576, 2012.

[22] D. Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67:1517–1531, 1998.

[23] D. Levin. Mesh-independent surface interpolation. *Geometric Modeling for Scientifc Visualization(2003)*, pages 1517–1523, 2003.

[24] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proc. SIGGRAPH '87*, pages 163–169, 1987.

[25] S. Morigi, M. Rucci, and F. Sgallari. Nonlocal surface fairing. *Scale Space and Variational Methods in Computer Vision*, pages 38–49, 2012.

[26] A. C. Öztireli, G. Guennebaud, and M. H. Gross. Feature preserving point set surfaces based on non-linear kernel regression. *Comput. Graph. Forum*, 28(2):493–501, 2009.

[27] G. Peyré, S. Bougleux, and L. Cohen. Non-local Regularization of Inverse Problems. In *ECCV 2008*, pages 57–68. Springer-Verlag, Marseille, France, 2008.

[28] O. Schall, A. Belyaev, and H. Seidel. Adaptive feature-preserving non-local denoising of static and time-varying range data. *Computer-Aided Design*, 40(6):701–707, 2008.

[29] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.

[30] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. ICCV '98, pages 839–, Washington, DC, USA, 1998. IEEE Computer Society.

[31] R. Wang, W. Chen, S. Zhang, Y. Zhang, and X. Ye. Similarity-based denoising of point-sampled surfaces. *Journal of Zhejiang University-Science A*, 9(6):807–815, 2008.

[32] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.*, 4(4):389–396, 1995.

[33] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *Image Processing, IEEE Transactions on*, 21(5):2481 –2499, may 2012.