

Surface Reconstruction with Enriched Reproducing Kernel Particle Approximation

Patrick Reuter¹ Pierre Joyot¹ Jean Trunzler¹ Tamy Boubekeur² Christophe Schlick²

¹ LIPSI - ESTIA (Bidart, France)

² LaBRI - INRIA Futurs - CNRS - University of Bordeaux 1 (France)

Abstract

There are many techniques that reconstruct continuous 3D surfaces from scattered point data coming from laser range scanners. One of the most commonly used representations are Point Set Surfaces (PSS) defined as the set of stationary points of a Moving Least Squares (MLS) projection operator. One interesting property of the MLS projection is to automatically filter out high frequency noise, that is usually present in raw data due to scanning errors. Unfortunately, the MLS projection also smoothes out any high frequency feature, such as creases or corners, that may be present in the scanned geometry, and does not offer any possibility to distinguish between such feature and noise. The main contribution of this paper, is to present an alternative projection operator for surface reconstruction, based on the Enriched Reproducing Kernel Particle Approximation (ERKPA), which allows the reconstruction process to account for high frequency features, by letting the user explicitly tag the corresponding areas of the scanned geometry.

1. Motivation

The recent evolution in 3D acquisition devices, which has made 3D range scanners available to a larger public, has generated a shift that 3D models are more and more often acquired rather than modeled. Various techniques have been developed that directly deal with the acquired unorganized point sets, without any triangulation step, see [KB04, AGP*04] for recent surveys. These techniques are becoming more and more popular, as many topological operations, such as level-of-detail generation, are much more efficient when directly performed on point sets rather than on polygonal meshes.

One of the most popular techniques to reconstruct a continuous 3D surface from scattered point data is the *Point Set Surface* (PSS) model, initially proposed by Alexa et al. [ABCO*01, ABCO*03]. This technique involves a projection operator based on the *Moving Least Squares* (MLS) method [LS81, Lev99, Lev03] that projects any point in the neighborhood of the point set onto a continuous surface that minimizes a local least-squares error criterion (for that reason, PSS are sometimes called MLS surfaces).

PSS are then implicitly defined as the set of points that project onto themselves. This MLS projection operator generates a smooth surface that approximates the scattered data, and automatically filters out high frequency noise that may be present in raw data due to scanning errors. Due to their robustness and efficiency, PSS have been successfully used in various application fields, such as point-based modeling [ZPKG02, PKKG03], point-based animation [MKN*04], surface reconstruction [SOS04, MVdF03], and post-processing of scanned data [WPH*04].

The efficiency/accuracy of the MLS technique strongly relies on a couple of user-provided parameters that are not easy to set optimally: first the degree of the bivariate polynomials used by the MLS fitting, and second a so-called *feature size* that will be used to distinguish between noise and feature. The main drawback of MLS is a direct consequence of this feature size: sharp creases and corners are always converted into rounded shapes, with a radius at least equal to the feature size, as every spatial frequency higher than the provided feature threshold is considered as noise by the MLS fitting.

Sharp features are a fundamental issue when processing surfaces, and several authors already requisitioned their inclusion into PSS (see [ABCO*03, FACOS03, AK04b]). Several approaches have been proposed to automatically detect high frequency features in point sets [KBSS01, PKG03, GWM01]. For example, Kobbelt et al. [KBSS01] propose to perform an automatic recognition of edges and corners, by using a simple heuristic based on the clustering of normal vectors. But such approaches work well only when there is very few noise in the initial point set. As the identification of sharp features in noisy point sets is an ill-posed problem, we rather propose a user-guided approach, where the sharp features can be explicitly specified. This kind of approach is particularly interesting when working in some interactive point-based modeling environment, such as PointShop 3D [PKKG03].

Note that there exist some prior work on sharp features in point sets, but to our knowledge, all existing techniques have been specifically oriented towards detection and preservation of sharp features when applying boolean operations on two different point sets [AD03, PKKG03, WTG04, XG04, OBA*03, FSCO05], which is a very different context as the work presented here.

The main contribution of this paper is an approach that offers the possibility to represent sharp features in PSS. The key idea is to replace the initial MLS projection operator by an alternative one, based on the *Enriched Reproducing Kernel Particle Approximation* (ERKPA), developed by Joyot et al. [JTC04] as an extension of the well-known Reproducing Kernel Particle Approximation (RKPA) introduced by Liu et al. in 1995 [LUC95]. Though they have been developed in totally different scientific fields (applied mechanics for RKPA, scattered data fitting for MLS), RKPA and MLS are very closely related, as they share the same mathematical kernel, except that RKPA are based on a shifted polynomial basis that is better suited from a numerical implementation point of view [JLA01] when the shape functions are required. The enriched version of RKPA goes a step further as it allows the fitting process not only to reproduce multivariate polynomials, but also some additional functions with discontinuous derivatives, allowing to reconstruct surfaces with explicitly specified sharp features.

The remainder of this paper is organized as follows. Section 2 recalls the PSS model with a particular focus on the MLS projection operator. Section 3 deeply focuses on the Reproducing Kernel Approximation theory, which is not widely known in the field of Computer Graphics. Section 4 presents our new formulation of PSS with sharp edges and studies some of its important properties. Finally some work in progress and prospective directions are discussed in Section 5.

2. Point Set Surfaces

As said in the introduction, Point Set Surfaces involve a projection operator, based on the Moving Least Squares method, that projects any point in the neighborhood of the initial point set onto a continuous surface that minimizes a local least-squares error criterion. The PSS is then implicitly defined as the set of points that project onto themselves. For a precise definition of "neighborhood", see the definition of the domain of a PSS given by Amenta and Kil [AK04b].

Actually, for each point, interactive applications of the projection operator are applied, until the point projects onto itself. This projection operator can be divided into two steps.

1. In the first step, for a point \mathbf{p}^{near} near the surface, a local reference plane defined in Hessian normal form $H = \{\mathbf{n} \bullet \mathbf{x} - d = 0, \mathbf{x} \in \mathbb{R}^3\}$ with $\mathbf{n} \in \mathbb{R}^3$ and $\|\mathbf{n}\| = 1$ is determined by minimizing a local weighted sum of squared distances of the points \mathbf{p}_i to the plane H

$$\sum_{i=1}^N (\mathbf{n} \bullet \mathbf{p}_i - d)^2 \theta(\|\mathbf{p}_i - \mathbf{q}\|), \quad (1)$$

where \mathbf{q} is the orthogonal projection of \mathbf{p}^{near} on H , and $\theta(t)$ is a radially symmetric, positive, and monotonically decreasing weight function. Note that finding the minimum of Equation (1) is a non-linear minimization problem that has to be solved iteratively.

2. In a second step, the local reference plane for \mathbf{p}^{near} is used to compute a local bivariate degree d polynomial approximation $f \in \mathbb{R}^2 \rightarrow \mathbb{R}$ of the surface in a neighborhood of \mathbf{p}^{near} . Let \mathbf{q}_i be the projection of \mathbf{p}_i onto H , then \mathbf{p}_i has the height $h_i = \mathbf{n} \bullet (\mathbf{p}_i - \mathbf{q}_i)$ over H , and the coefficients of the polynomial approximation f are determined by minimizing the weighted least squares error

$$\sum_{i=1}^N (f(x_i, y_i) - h_i)^2 \theta(\|\mathbf{p}_i - \mathbf{q}\|), \quad (2)$$

where $[x_i, y_i]^T$ is the representation of \mathbf{q} in the local reference plane. Alexa et al. proposed to use cubic or quartic polynomials for f , since they experienced surface oscillation artifacts with higher degree polynomials. The projection $\Psi: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ of \mathbf{p}^{near} onto the surface is defined by the polynomial value in the origin $\Psi(\mathbf{p}^{near}) = \mathbf{q} + f(0, 0)\mathbf{n}$. Note that the property $\Psi(\Psi(\mathbf{p})) = \Psi(\mathbf{p})$ for points \mathbf{p} on the surface is extremely important, since this means that a point on the surface projects onto itself.

Alexa et al. used the same Gaussian function as proposed by Levin, for the weighting function $\theta(t)$ used in Equations (1) and (2): $\theta(t) = e^{-t^2/h^2}$. The parameter h , called *feature size*, influences the smoothness of the reconstructed surface, as it smoothes out any geometric feature smaller than h . So, it is usually set to the anticipated average spacing between neighboring points. Note that for any weighting function $\theta(t)$ with C^∞ continuity, Levin proved that the resulting surface also has C^∞ continuity [Lev98].

3. Reproducing Kernel Particle Approximation

3.1. Introduction

Many different techniques have been proposed in the literature to reconstruct an implicit surface from an unstructured point set. To clearly distinguish these methods from the usual mesh-based reconstruction techniques (e.g. Finite Element Methods) that work on structured locally parameterized point sets, the expression *Meshfree Methods*[†] is now ubiquitously used to qualify this research field. Mesh-free Methods were simultaneously developed in the field of applied mathematics, and in the field of physics, more precisely in mechanics, hydrodynamics and astrophysics. Surprisingly, these two streams have only merged recently, mainly thanks to the work done by Belytschko et al. in their exhaustive survey published in 1996 [BKO*96]. A more recent survey (referencing more than 400 articles) has been published in 2002 by Li and Liu [LL02].

The basic idea of meshfree methods is to take the classical *Finite Series Representation* and *Finite Integral Representation* of an N-dimensional function (that have been intensively used in mesh-based methods) and try to find out what can be done without any assumption about parameterization. It is interesting to note that the first family of mesh-free methods, called *Finite Series Meshfree Approximation*, was specifically developed in the field of applied mathematics, mainly to solve the scattered data fitting problem, whereas the second family, called *Finite Integral Meshfree Approximation*, was exclusively developed in physics (more precisely in hydrodynamics and astrophysics).

Some of these existing meshfree methods (e.g. *Partition of Unity*, *Radial Basis Functions* or *Moving Least Squares*) have recently been introduced in the field of computer graphics, mainly to provide an efficient and robust solution to the surface reconstruction from 3D range scanner measures. In this paper, we propose to introduce another technique taken from the very crowded meshfree zoo: we have chosen to use the *Enriched Reproducing Kernel Particle Approximation* (ERKPA) [JTC04] for its ability to include very general constraints on the reconstructed surface. Since the underlying theory of Reproducing Kernel Approximation is not very widely known in computer graphics, we propose to first explain the method for the 1D case, by following a step-by-step constructive approach, and then to focus on the 2D case, as required by the projection operator involved in Point Set Surfaces.

3.2. 1D Scattered Data Approximation with ERKPA

The central idea of the *Reproducing Kernel Approximation* theory, which has been borrowed from the older *Smoothed*

[†] The word *Meshless* is also quite common but, following several authors, we advocate for the word *Meshfree*, as it includes more “positive” values.

Particle Hydrodynamics method, is to define the approximated function $\bar{f}(x)$ as the convolution of the original function $f(x)$ with a kernel function $w_h(x)$:

$$\bar{f}(x) = \int_{\Omega} w_h(x-s) f(s) ds \quad (3)$$

The kernel function $w_h(x)$ is also called the weighting function or the smoothing function. Note that $w_h(x)$ depends on an additional parameter h that defines the size of the convolution kernel (it is sometimes called the *feature size*) and may vary throughout the domain. It has been shown that several properties are desirable for the kernel function $w_h(x)$:

- Positivity: $w_h(x) \geq 0, \forall x$
- Decay: $w_h(x)$ is monotonically decreasing
- Compact support: $w_h(x-s) = 0$ when $\|x-s\| > h$
- Partition of unity: $\int_{\Omega} w_h(x-s) ds = 1$
- Dirac-like shape: $w_h(s) \rightarrow \delta(s)$ when $h \rightarrow 0$

where $\delta(s)$ is the Dirac function.

A usual quality criterion of an approximation technique is its *consistency order*, that is its ability to be able to exactly reproduce a polynomial function up to a given degree n . The RKPA method has specifically been designed to automatically build approximations with n^h -order consistency. The ERKPA goes one step further by allowing the approximation to reproduce an additional function $e(x)$ called *Enrichment Function*. This enrichment function is the key issue that enables to include sharp edges in the surface, since, as we will see later, by using a Heaviside function for $e(x)$, localized discontinuity in the derivatives can be added to the approximation function.

More formally, we want $\bar{f}(x)$ to reproduce all functions of the following type:

$$\bar{f}(x) = a_0 + a_1x + \dots + a_nx^n + e(x) \quad (4)$$

The properties required for the kernel function $w_h(x-s)$ to reproduce the function given by (4) can be established step-by-step. First suppose that we want the approximation $\bar{f}(x)$ defined by Equation (3) to reproduce a constant function: $f(x) = a_0$. This means that

$$\int_{\Omega} w_h(x-s) a_0 ds = a_0, \quad (5)$$

that implies

$$\int_{\Omega} w_h(x-s) ds = 1. \quad (6)$$

So we rederive the Partition of Unity property of the kernel function. This is not surprising because PoU is precisely the condition required to be able to reproduce a constant function. Now, suppose that we want to reproduce a linear function (i.e. first-order consistency): $\bar{f}(x) = a_0 + a_1x$

$$\int_{\Omega} w_h(x-s)(a_0 + a_1s) ds = a_0 + a_1x \quad (7)$$

which can be rewritten as

$$\begin{cases} \int_{\Omega} w_h(x-s) ds = 1 \\ \int_{\Omega} w_h(x-s) s ds = x \end{cases} \quad (8)$$

By iterating this process, we obtain the reproduction of Equation (4) with the following constraints:

$$\begin{cases} \int_{\Omega} w_h(x-s) ds = 1 \\ \int_{\Omega} w_h(x-s) s ds = x \\ \vdots \\ \int_{\Omega} w_h(x-s) s^n ds = x^n \\ \int_{\Omega} w_h(x-s) e(s) ds = e(x) \end{cases} \quad (9)$$

Finding a suitable kernel function $w_h(x)$ that satisfies the reproduction conditions given by Equation (9), would be very difficult in the general case. To remove this difficulty, Liu et al. [LJL*95] proposed to shift the problem a little bit, by introducing a correction function $C(x, s)$ that acts as a scaling function. Thus, from now on, we define $\bar{f}(x)$ by

$$\bar{f}(x) = \int_{\Omega} C(x, s) w_h(x-s) f(s) ds \quad (10)$$

$$\text{where } C(x, s) = \mathbf{B}^T(x, s) \mathbf{A}(x).$$

$\mathbf{B}^T(x, s)$ represents the vector containing the functions that belong to the approximation basis, and $\mathbf{A}(x)$ is a vector containing the unknown functions that have to be determined for satisfying the reproduction conditions. Thus, Equation (9) can be rewritten as

$$\begin{cases} \int_{\Omega} \mathbf{B}^T(x, s) \mathbf{A}(x) w_h(x-s) ds = 1 \\ \int_{\Omega} \mathbf{B}^T(x, s) \mathbf{A}(x) w_h(x-s) s ds = x \\ \vdots \\ \int_{\Omega} \mathbf{B}^T(x, s) \mathbf{A}(x) w_h(x-s) s^n ds = x^n \\ \int_{\Omega} \mathbf{B}^T(x, s) \mathbf{A}(x) w_h(x-s) e(s) ds = e(x) \end{cases} \quad (11)$$

Equation (11) represents the continuous form of the ERKPA, as defined in [JTC04]. Since our goal is to reconstruct an implicit surface from a discrete point set, we have to consider the discrete form of the ERKPA that would be expressed as a finite series instead of a finite integral (with $\Delta x_i = 1$):

$$\begin{cases} \sum_{i=1}^N \mathbf{B}^T(x, x_i) \mathbf{A}(x) w_h(x-x_i) \Delta x_i = 1 \\ \sum_{i=1}^N \mathbf{B}^T(x, x_i) \mathbf{A}(x) w_h(x-x_i) x_i \Delta x_i = x \\ \vdots \\ \sum_{i=1}^N \mathbf{B}^T(x, x_i) \mathbf{A}(x) w_h(x-x_i) x_i^n \Delta x_i = x^n \\ \sum_{i=1}^N \mathbf{B}^T(x, x_i) \mathbf{A}(x) w_h(x-x_i) e(x_i) \Delta x_i = e(x) \end{cases} \quad (12)$$

resulting in the following linear system

$$\left[\sum_{i=1}^N \mathbf{R}(x_i) \mathbf{B}^T(x, x_i) w_h(x-x_i) \Delta x_i \right] \mathbf{A}(x) = \mathbf{R}(x) \quad (13)$$

where $\mathbf{R}(x)$ is the reproduction vector defined by

$$\mathbf{R}^T(x) = [1, x, \dots, x^n, e(x)] \quad (14)$$

Equation (13) allows the computation of the vector $\mathbf{A}(x)$,

$$\mathbf{A}(x) = \mathbf{M}^{-1}(x) \mathbf{R}(x) \quad (15)$$

where the matrix $\mathbf{M}(x)$ is called the *moment matrix* and is defined by

$$\mathbf{M}(x) = \sum_{i=1}^N \mathbf{R}(x_i) \mathbf{B}^T(x, x_i) w_h(x-x_i) \Delta x_i \quad (16)$$

Note that we adopt the strategy described by Joyot et al. [JTC04] (see the next section) to make the moment matrix $\mathbf{M}(x)$ invertible.

Starting from a discrete scattered data set (x_i, λ_i) , a continuous approximation function $\bar{f}(x)$ where $\bar{f}(x_i) \cong \lambda_i$ can be derived from Equations (10) and (15):

$$\bar{f}(x) = \sum_{i=1}^N \mathbf{B}^T(x, x_i) \mathbf{M}^{-1}(x) \mathbf{R}(x_i) w_h(x-x_i) \lambda_i \Delta x_i$$

which can be compactly written as

$$\bar{f}(x) = \sum_{i=1}^N \phi_i(x) \lambda_i \quad (17)$$

where ϕ_i is called the ERKPA *shape function*:

$$\phi_i(x) = \mathbf{B}^T(x, x_i) \mathbf{M}^{-1}(x) \mathbf{R}(x_i) w_h(x-x_i) \Delta x_i \quad (18)$$

3.3. 2D Sharp Features Reconstruction with ERKPA

As said above, inclusion of sharp features into PSS can be achieved by switching the projection operator from MLS to ERKPA. This section details the construction of this alternative operator. To simplify the notations, we only present the case where the reconstructed surface locally fits a bivariate quadratic polynomial with k additional enrichment functions e_1, e_2, \dots, e_k . Note that k is not the total number of sharp features of the 3D model, but only the number of features that are present in the local area processed by the projection operator. So, usual values of k are either 1 for a single crease on the surface, or 3 for a corner of a box-like shape. Similarly, as already observed by Ohtake et al. [OBA*03], a bivariate quadratic polynomial is sufficient in practice to reconstruct the smooth part of the surface. With this constraint, Equation (4) becomes:

$$\bar{f}(x, y) = a_0 + a_1x + a_2x^2 + a_3y + a_4y^2 + a_5xy + \sum_{i=1}^k a_{i+5} e_i(x, y) \quad (19)$$

In order to introduce sharp features in the reconstructed surface, the user can explicitly specify some feature curve $\Lambda_i(x, y)$, and the enrichment function $e_i(x, y)$ will be defined so that the reconstructed surface presents a tangent discontinuity along this curve. Note that the easiest way to define such feature curve in some interactive modeling environment such as PointShop 3D, would be to directly paint on the surface, and generate the implicit equation of the corresponding curve, as already used in other PointShop tools. Actually, the feature curve $\Lambda_i(x, y)$ is defined on the local reference plane computed by the first step of the projection

operator. This feature curve splits the corresponding domain $\Omega \subseteq \mathbb{R}^2$ into two subdomains Ω_0 and Ω_1 (see Figure 1):

$$\Omega_0 \cup \Omega_1 \cup \Lambda_i = \Omega \quad \text{and} \quad \Omega_0 \cap \Omega_1 = \emptyset$$

The enrichment function $e_i(x, y)$ has to satisfy the following conditions:

$$\begin{aligned} \forall (x, y) \in \Omega, e_i(x, y) \text{ is } C^0 \text{ continuous} \\ \forall (x, y) \in \Omega_0, e_i(x, y) \text{ is } C^1 \text{ continuous} \\ \forall (x, y) \in \Omega_1, e_i(x, y) \text{ is } C^1 \text{ continuous} \\ \forall (x, y) \in \Lambda_i, \nabla e_i(x, y) \cdot \mathbf{n} \text{ is discontinuous} \end{aligned} \quad (20)$$

where \mathbf{n} denotes the unit outward vector defined on the feature curve $\Lambda_i(x, y)$. One possible enrichment function $e_i(x, y)$ that satisfies the conditions (20) is the unsigned distance $\Delta_i(x, y)$ to the feature curve Λ_i . Note that it is often desirable to limit the influence of the sharp feature on the reconstructed surface, and thus we define a compactly supported enrichment function $e_i(x, y)$ with a kernel size of ε_i . This kernel size ε_i may either be computed according to the local density of sample points, or explicitly specified by the user, during the interactive reconstruction process. Note that ε_i may be computed/specified independently for each feature if needed, and also independently from the kernel size h of the ERKPA shape function (see Equation 18).

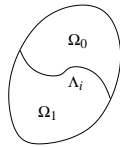


Figure 1: The feature curve Λ_i splits the local reference plane into two subdomains Ω_0 and Ω_1 .

Of course, we have to guarantee continuity between the parts of the domain with and without influence of the sharp feature. To this end, we define the enrichment functions $e_i(x, y)$ as the composition of the distance $\Delta_i : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ to the discontinuity curve, and a decay function $\theta : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, i.e. $e_i(x, y) = \theta \circ \Delta_i(x, y)$. We suggest to use the following symmetric decay function, that was defined by including some simple boundary constraints, similar to the construction of spline base functions: $\theta(0) = 1$, $\theta(\varepsilon_i) = 0$, $\theta'(0) = -1$, and $\theta'(\varepsilon_i) = 0$:

$$\theta(t) = \frac{\varepsilon_i + 2}{\varepsilon_i^3} t^3 + \frac{-2\varepsilon_i - 3}{\varepsilon_i^2} t^2 + t + 1 \quad (21)$$

So, in order to compute the reproduction function $\bar{f}(x)$ defined by Equation (17), we set:

$$\mathbf{R}^T(x, y) = [1, x, x^2, y, y^2, xy, e_1(x, y) \dots e_k(x, y)] \quad (22)$$

$$\mathbf{R}^T(x_i, y_i) = [1, x_i, x_i^2, y_i, y_i^2, x_i y_i, e_1(x_i, y_i) \dots e_k(x_i, y_i)] \quad (23)$$

$$\mathbf{B}^T(x, y, x_i, y_i) = [1, x - x_i, (x - x_i)^2, y - y_i, (y - y_i)^2, (x - x_i)(y - y_i), e_1(x_i, y_i) \dots e_k(x_i, y_i)] \quad (24)$$

Both vectors \mathbf{R} and \mathbf{B} are of dimension $(6+k)$, and hence the moment matrix \mathbf{M} of Equation (16) becomes a $(6+k) \times (6+k)$ non-symmetric matrix.

The main issue when using a compactly supported enrichment function is to ensure that the moment matrix \mathbf{M} can be inverted. This requires that for each enrichment function $e_i(x, y)$, the support of every shape function contains at least one node with $e_i(x, y) > 0$. Indeed, when the shape function only contains nodes with $e_i(x, y) = 0$, one line of the matrix \mathbf{M} will contain only null values. When this case arises, we simply remove the corresponding enrichment function, and solve the reduced linear system defined by the $(5+k)$ -dimensional vectors \mathbf{R} and \mathbf{B} , and the $(5+k) \times (5+k)$ non-symmetric moment matrix \mathbf{M} . Joyot et al. proved that this construction always ensures the correct order of consistency [JTC04].

4. Point Set Surfaces with Sharp Edges

After the construction of the ERKPA shape function $f(x, y)$, it can be integrated into the Equation (2) in the second step of the projection operator, similarly to the MLS method. We also use the same Gaussian weighting function $\theta(t) = e^{-t^2/h^2}$ as proposed by Levin [Lev99, Lev03]. The parameter h still controls the feature size and thus influences the smoothness of the reconstructed surface, but thanks to the ERKPA reconstruction, the sharp features that exist at the discontinuity curve $\Lambda_i(x, y)$ are preserved in the resulting PSS, whatever the value provided for h . Figure 2 presents the reconstruction obtained from a noisy 1D point set, by using various degrees d and different feature sizes h , both with ERKPA and with MLS. Note the accurate reconstruction of the discontinuity by the former compared to the latter.

To test our ERKPA surface reconstruction technique on 3D point sets, we have first implemented a standalone application, where the surface implicitly reconstructed by the MLS or the ERKPA operator can be tessellated using the marching cubes algorithm by Bloomenthal [Blo94]. The standard marching cube algorithm is not well adapted to represent sharp features, some improved polygonizers that are feature-sensitive (see Ohtake and Belyaev [OB02] as well as Kobbelt et al. [KBSS01]) should definitely be more adapted, but we haven't implemented them yet.

As a first simple example, consider the point set of a parallelepiped in Figure 3(a) and of a twisted parallelepiped in Figure 3(d). Using the classical RKPA implicit surface reconstruction, the edges are smoothed out as can be seen in Figure 3(b) (resp. Figure 3(e)). By specifying the surface that cuts the edges of the parallelepiped and applying the ERKPA implicit surface reconstruction techniques, the edges remain sharp as can be seen in Figure 3(c) (resp. Figure 3(f)). (Note that the artifacts along the sharp edges are only generated during the polygonization step, since we did not use a feature-sensitive polygonizer as said above).

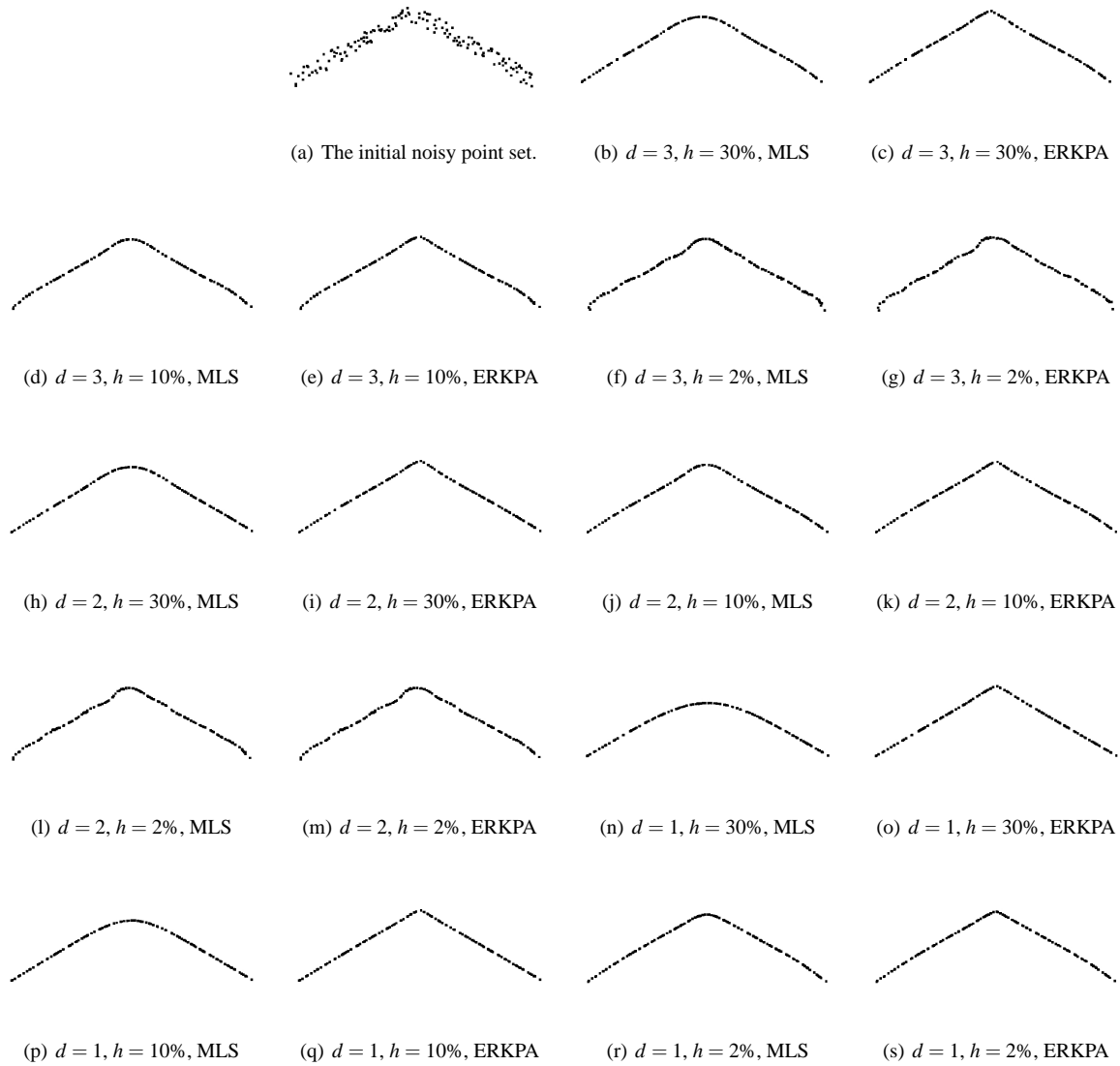


Figure 2: PSS for various degrees d and different feature sizes h in % of the bounding box, with MLS and ERKPA. The sharp feature is explicitly specified by tagging the point at $x = 0$. In all images, 3 iterations of the projection operators were applied.

As a more complex example, consider the highly non-uniformly distributed point set of the Cyberware screwdriver (Figure 4(a) with a close-up in Figure 4(d)). Again, we compare the classical MLS reconstruction (Figure 4(b) with a close-up in Figure 4(e)) with the ERKPA reconstruction that preserves sharp edges (Figure 4(c) with a close-up in Figure 4(f)). We specified two planes along the planar part of the screwdriver. Again, the polygonizing artifacts should be mentally occulted when watching the figure.

We have also implemented the ERKPA projection operator as a plugin for PointShop 3D, an interactive point-based

modeling tool. Currently, all the functionalities of PointShop are not available to the ERKPA operator, but we have already obtained some interesting results. One classical operation during an interactive session under PointShop is to generate some upsampling in the area of interest. With the ERKPA operator, this upsampling can be achieved while maintaining some sharp feature that the user wants to preserve. For example, consider the 6052 points of the Moais statue that we scanned with our Minolta Vivid 300 scanner (Figure 5(a)). We upsampled the statue to 51701 points. The resulting point set obtained with classical PSS is shown

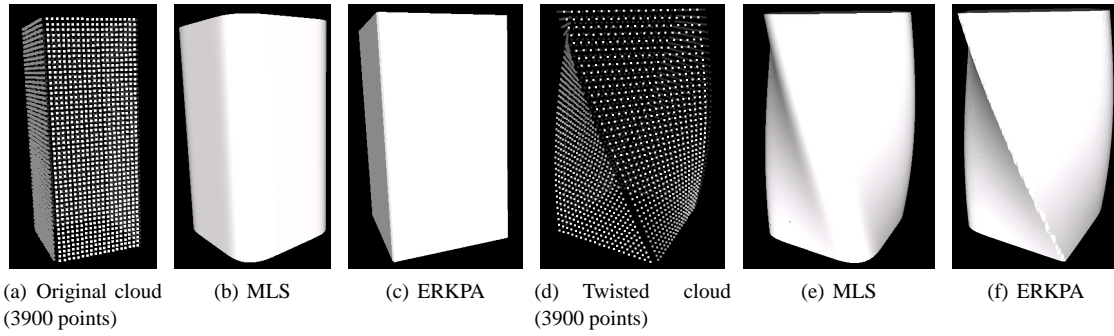


Figure 3: Reconstruction of surface with sharp edges by MLS vs. ERKPA

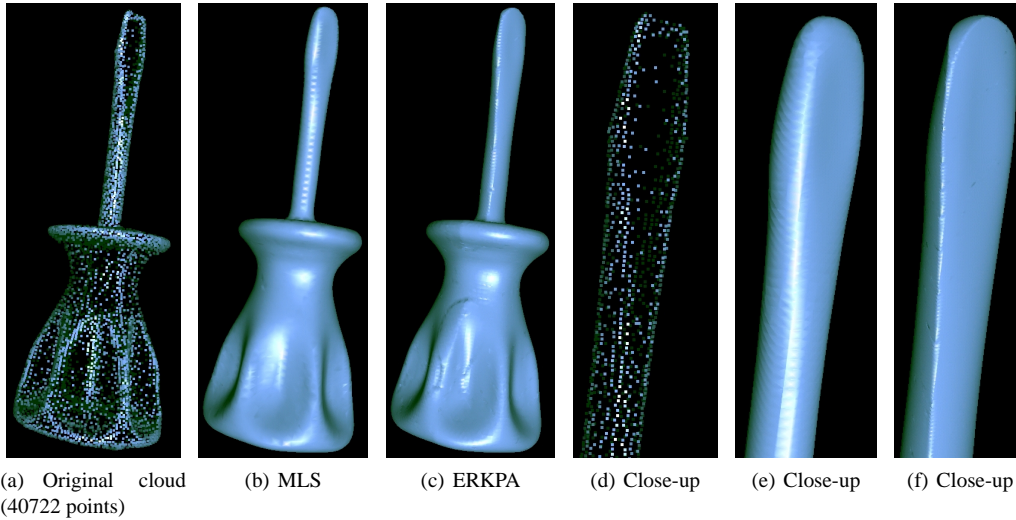


Figure 4: Reconstruction of surface with sharp edges by MLS vs. ERKPA

in Figure 5(b). Since the original model has a discontinuity at the bottom of the hat, we explicitly specified a horizontal circle as the feature curve Λ_i . The upsampling using our PSS with sharp features clearly shows the discontinuity at the bottom of the hat as shown in Figure 5(c).

Let us now study the involved complexity of PSS with sharp edges. The construction of the discrete form of the approximation function of Equation (17) requires several matrix and vector multiplications and especially the inversion of the moment matrix \mathbf{M} . The dimension of this matrix \mathbf{M} and the corresponding vectors \mathbf{R} and \mathbf{B} both depend on the polynomial degree d and the number k of simultaneous enrichment functions $e_i(x, y)$: $1 + k$ for constant reconstruction, $3 + k$ for linear reconstruction, $6 + k$ for quadratic reconstruction. Remember that usual values for k are 1 for creases, and 3 for corners. Compared to the smooth case obtained with MLS, the increase of k in the number of dimensions of \mathbf{M} , \mathbf{R} , and \mathbf{B} , the evaluation of the k enrichment functions $e_i(x, y)$, as well as the evaluating of the discontinuity curves Λ_i are

the only complexity penalties when incorporating sharp features into PSS with ERKPA.

Note that when defining PSS with sharp features, the normal of the approximating local reference plane is not the true surface normal, as shown by Adamson and Alexa [AA04]. Hence, to obtain the correct surface normal, we rather have to calculate the normal of the local bivariate function $f(x, y)$ and transform this normal into the global coordinate frame.

Let us finally give some insight to the difference of using our new PSS with sharp edges compared to the prior solutions to incorporate sharp features in point sets that apply boolean operations on two different point sets (for example [AD03, PKKG03, WTG04, XG04, OBA*03, FSCO05]).

First, applying boolean operations on two different point sets is easy to perform in 1D, but becomes much harder in higher dimensions. Second, defining sharp edges (and consequently corners) seems to be possible with boolean operations, but when more general features (defined by the in-



(a) Original cloud (6052 points) (b) Upsampling with MLS (c) Upsampling with ERKPA

Figure 5: Partial upsampling of a scanned point sets using MLS vs. ERKPA operator. The desired sharp feature has been specified by tagging the points. The feature size is locally determined by using the 50 nearest neighbors.

tersections with the arbitrary surfaces Λ_i) should be incorporated, this is not trivial with boolean operations, especially in 3D. Finally, consider again Figure 2 that demonstrates that the incorporation of the polynomial parts of the approximation functions of the other side of the discontinuity curve Λ_i influences the resulting surfaces.

5. Conclusion

In this paper, we extended Point Set Surfaces, introduced by Alexa et al. [ABCO*01, ABCO*03], to include a simple definition of sharp features in the reconstructed surfaces. This is done by replacing the MLS projection operator by an ERKPA projection operator, that allows not only to reproduce polynomials, but also some enrichment functions with discontinuous derivatives. A user can explicitly specify the locations of the discontinuous derivatives that generate the sharp features in the resulting PSS. Thanks to a compactly supported enrichment function with a user-specifiable support size, the influence domain of the sharp features can be perfectly controlled.

In addition to the existing applications of PSS (e.g. point-based modeling and point-based animation), we believe that our new projection operator may also find other application fields, such as selected noise reduction of scanned surface data or the reconstruction of surfaces from noisy data while keeping sharp features. The MLS projection operator has also been successfully used to reconstruct semi-regular polygonal meshes [MVdF03]. Recently, Shen et al. [SOS04] used the MLS method to reconstruct a trivariate defining function of implicit surfaces from polygon soups. We believe that the ERKPA method can replace the MLS method in both techniques in order to account for sharp features.

Note that RKPA and MLS are very closely related, as they share the same mathematical kernel, except that RKPA are based on a shifted polynomial basis that is better suited from

a numerical implementation point of view [JLA01] when the shape functions are required. Furthermore, RKPA allows to reconstruct interpolating rather than approximating surfaces without requiring the use of a singular kernel function w_h by using a so-called *Collocation Preprocessing* [Alu00]. This is subject of current ongoing work.

We are also investigating various other directions for future work. First, we are trying to accelerate the evaluation of the local bivariate function $f(x, y)$, in particular by shifting the involved matrix inversion of the ERKPA projection operator into a preprocess. We think that this is possible, at least for the parts of the domain that are completely out of the compact support of the enrichment functions $e_i(x, y)$. Second, we are currently examining the best way to automatically detect features like edges and corners in the point sets in order to let the user decide between an automatic and a semi-automatic process to define sharp features. We believe that we can use a forward search method from robust statistics to detect the sharp features [FSCO05] and couple it with our method. Finally, we want to study the domain of the PSS with sharp features, and more precisely how it can be explicitly defined in terms of the critical points of an energy function on lines determined by a vector field, following the work of Amenta and Kil [AK04b, AK04a].

References

- [AA04] ALEXA M., ADAMSON A.: On normals and projection operators for surfaces defined by point sets. *Symposium on Point-based Graphics* (2004).
- [ABCO*01] ALEXA M., BEHR J., COHEN-OR D., LEVIN D., FLEISHMAN S., SILVA C. T.: Point set surfaces. In *Proceedings of IEEE Visualization* (2001).
- [ABCO*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics* 9, 1 (2003).
- [AD03] ADAMS B., DUTRE P.: Interactive boolean operations on surfel-bounded solids. In *Proceedings of ACM SIGGRAPH* (2003).
- [AGP*04] ALEXA M., GROSS M., PAULY M., PFISTER H., STAMMINGER M., ZWICKER M.: Point-based computer graphics. *ACM SIGGRAPH Course Notes* (2004).
- [AK04a] AMENTA N., KIL Y. J.: Defining point-set surfaces. In *Proceedings of ACM SIGGRAPH* (2004).
- [AK04b] AMENTA N., KIL Y. J.: The domain of a point set surfaces. *Symposium on Point-based Graphics* (2004).
- [Alu00] ALURU N.: A point collocation method based on reproducing kernel approximations. *International Journal for Numerical Methods in Engineering* 47 (2000).
- [BKO*96] BELYTSCHKO T., KRONGAUZ Y., ORGAN D., FLEMING M., KRYSL P.: Meshless methods: An

- overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 139 (1996).
- [Blo94] BLOOMENTAL J.: An implicit surface polygonizer. *Graphics Gems IV* (1994).
- [FACOS03] FLEISHMAN S., ALEXA M., COHEN-OR D., SILVA C. T.: Progressive point set surfaces. *ACM Transactions on Computer Graphics* 22, 4 (2003).
- [FSCO05] FLEISHMAN S., SILVA C., COHEN-OR D.: Robust moving least-squares fitting with sharp features. In *Proceedings of ACM SIGGRAPH* (2005). To appear.
- [GWM01] GUMHOLD S., WANG X., MACLEOD R.: Feature extraction from point clouds. In *Proceedings of the 10th International Meshing Roundtable* (2001).
- [JLA01] JIN X., LI G., ALURU N.: On the equivalence between least-squares and kernel approximations in meshless methods. *CMES: Computer Modeling in Engineering and Sciences* 2, 4 (2001).
- [JTC04] JOYOT P., TRUNZLER J., CHINESTA F.: Enriched reproducing kernel approximation: Reproducing functions with discontinuous derivatives. In *Meshfree Methods for Partial Differential Equations II*, vol. 43. Springer, 2004, pp. 93–108.
- [KB04] KOBBELT L., BOTSCH M.: A survey of point-based techniques in computer graphics. *Computers & Graphics* 28, 6 (2004).
- [KBSS01] KOBBELT L. P., BOTSCH M., SCHWANECKE U., SEIDEL H.-P.: Feature-sensitive surface extraction from volume data. In *Proceedings of ACM SIGGRAPH* (2001).
- [Lev98] LEVIN D.: The approximation power of moving least-squares. *Mathematics of Computation* 67, 224 (1998).
- [Lev99] LEVIN D.: Mesh-independent surface interpolation. In *4th International Conference on Curves and Surfaces* (1999).
- [Lev03] LEVIN D.: Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*. Springer, 2003.
- [LJL*95] LIU W., JUN S., LI S., ADEE J., BELYTSCHKO T.: Reproducing kernel particle methods for structural dynamics. *Int. J. Numer. Meths. Eng.* 38 (1995), 1655–1679.
- [LL02] LI S., LIU W.-K.: Meshfree particle methods and their applications. *Applied Mechanics Review* 54 (2002).
- [LS81] LANCASTER P., SALKAUSKAS K.: Surfaces generated by moving least squares methods. *Mathematics of Computation* 37, 155 (1981).
- [LUC95] LIU W. K., URAS R. A., CHEN Y.: Enrichment of the finite element method with reproducing kernel particle method. In *Current Topics in Computational Mechanics* (1995). A journal version appeared in the *Journal of Applied Mechanics ASME* (1997).
- [MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Symposium on Computer Animation* (2004).
- [MVdF03] MEDEROS B., VELHO L., DE FIGUEIREDO L. H.: Moving least squares multiresolution surface approximation. In *Proceedings of SIBGRAPI* (2003).
- [OB02] OHTAKE Y., BELYAEV A. G.: Dual/primal mesh optimization for polygonized implicit surfaces. *Journal of Computing and Information Science in Engineering* 2, 4 (2002).
- [OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. *ACM Transactions on Graphics (TOG)* 22, 3 (2003), 463–470.
- [PKG03] PAULY M., KEISER R., GROSS M.: Multi-scale feature extraction on point-sampled models. In *Proceedings of Eurographics* (2003).
- [PKKG03] PAULY M., KEISER R., KOBBELT L., GROSS M.: Shape modeling with point-sampled geometry. In *Proceedings of ACM SIGGRAPH* (2003).
- [SOS04] SHEN C., O'BRIEN J. F., SHEWCHUK J. R.: Interpolating and approximating implicit surfaces from polygon soup. In *Proceedings of ACM SIGGRAPH* (2004).
- [WPH*04] WEYRICH T., PAULY M., HEINZLE S., KEISER R., SCANDELLA S., GROSS M.: Post-processing of scanned 3D surface data. In *Symposium on Point-Based Graphics* (2004).
- [WTG04] WICKE M., TESCHNER M., GROSS M.: CSG tree rendering of point-sampled objects. In *Proceedings of Pacific Graphics* (2004).
- [XG04] XIAOHU GUO JING HUA H. Q.: Touch-based haptics for interactive editing on point set surfaces. *IEEE Computer Graphics and Applications* 24, 6 (2004).
- [ZPKG02] ZWICKER M., PAULY M., KNOLL O., GROSS M.: Pointshop 3D: An interactive system for point-based surface editing. In *Proceedings of ACM SIGGRAPH* (2002).