Master Thesis

High frequency feature preserving implicit surfaces from point sets using robust statistics

Author(s):
Öztireli, A. Cengiz

Publication Date:
2008

Permanent Link:
https://doi.org/10.3929/ethz-a-005677923

Rights / License:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.
High Frequency Feature Preserving Implicit Surfaces from Point Sets using Robust Statistics

A. Cengiz Öztireli

Master Thesis
September 2008

Supervisor:
Prof. Dr. Markus Gross
Abstract

We propose a new implicit Moving Least Squares (MLS) surface definition that preserves high frequency details including sharp features and reduces many artifacts present with existing MLS surface definitions. The new definition is stable under sparse sampling and sharpness of features is controllable. Although it is defined iteratively, convergence rate is high and computation is efficient. Since it is in an analytic form, derivatives such as curvatures are computed trivially and accurately. To derive this definition, we show that MLS surfaces can be interpreted as a Local Kernel Regression (LKR) on the implicit function defining the surface. This opens the way to adapt techniques from kernel regression literature to improve MLS surfaces. We use one such technique, robust kernel regression, to derive our new MLS surface definition. Many other possible methods that can be utilized to improve MLS surfaces are also pointed out.
# Contents

<table>
<thead>
<tr>
<th>List of Figures</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
</tbody>
</table>

## 1. Introduction

1.1. Contributions .......................................................... 2

## 2. Preliminaries

2.1. Differential Geometry of Implicit Surfaces .......................... 5

2.1.1. Submanifolds ......................................................... 5

2.1.2. Implicit Surface Definition .................................... 6

2.1.3. Signed Distance Fields ......................................... 6

2.1.4. Differentials ....................................................... 7

2.2. Moving Least Squares ................................................ 7

2.2.1. Moving Least Squares Surfaces ................................ 8

2.2.2. Constraints ......................................................... 9

## 3. Regression and MLS Surfaces

3.1. Kernel Regression .......................................................... 12

3.1.1. Unsupervised Kernel Regression ................................ 12

3.1.2. Supervised Kernel Regression .................................. 13

3.2. Local Kernel Regression ................................................ 13

3.2.1. Kernel Functions .................................................. 14

3.3. MLS Surfaces from LKR .................................................. 15

3.3.1. Deriving Kolluri’s [Ko05] MLS definition ...................... 16

3.3.2. Deriving Adamson et al.’s [AA03] MLS definition ............. 16
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.3. Implications</td>
<td>18</td>
</tr>
<tr>
<td>4. Robust Statistics for Local Kernel Regression</td>
<td>19</td>
</tr>
<tr>
<td>4.1. Robust Statistics</td>
<td>19</td>
</tr>
<tr>
<td>4.1.1. Breakdown Point</td>
<td>20</td>
</tr>
<tr>
<td>4.1.2. Robust Kernel Regression Techniques</td>
<td>20</td>
</tr>
<tr>
<td>4.2. Robust Local Kernel Regression</td>
<td>21</td>
</tr>
<tr>
<td>4.2.1. Iteratively Reweighted Least Squares</td>
<td>21</td>
</tr>
<tr>
<td>4.3. Applications in Computer Graphics And Vision</td>
<td>23</td>
</tr>
<tr>
<td>4.3.1. Bilateral Filtering</td>
<td>23</td>
</tr>
<tr>
<td>4.3.2. Robust MLS Surfaces</td>
<td>27</td>
</tr>
<tr>
<td>5. A New MLS Surface Definition</td>
<td>29</td>
</tr>
<tr>
<td>5.1. Foundations and Previous Work</td>
<td>29</td>
</tr>
<tr>
<td>5.1.1. Local Surface Fitting</td>
<td>29</td>
</tr>
<tr>
<td>5.1.2. Previous Work on MLS Surfaces with Robust Statistics</td>
<td>31</td>
</tr>
<tr>
<td>5.2. Our Approach</td>
<td>32</td>
</tr>
<tr>
<td>5.2.1. Our MLS Surface Definition</td>
<td>32</td>
</tr>
<tr>
<td>5.2.2. Derivatives</td>
<td>33</td>
</tr>
<tr>
<td>5.3. Properties and Results</td>
<td>35</td>
</tr>
<tr>
<td>5.3.1. Sharp Features</td>
<td>35</td>
</tr>
<tr>
<td>5.3.2. Detail Preservation</td>
<td>36</td>
</tr>
<tr>
<td>5.3.3. Sampling</td>
<td>38</td>
</tr>
<tr>
<td>5.3.4. Curvatures</td>
<td>40</td>
</tr>
<tr>
<td>5.3.5. Controllable Sharpness</td>
<td>40</td>
</tr>
<tr>
<td>5.3.6. Performance</td>
<td>41</td>
</tr>
<tr>
<td>6. Outlook</td>
<td>45</td>
</tr>
<tr>
<td>6.1. Convergence and Continuity</td>
<td>45</td>
</tr>
<tr>
<td>6.1.1. Convergence</td>
<td>45</td>
</tr>
<tr>
<td>6.1.2. Continuity</td>
<td>45</td>
</tr>
<tr>
<td>6.2. Anisotropic Setting</td>
<td>46</td>
</tr>
<tr>
<td>6.2.1. Metric Learning</td>
<td>46</td>
</tr>
<tr>
<td>6.3. Higher Order LKR</td>
<td>46</td>
</tr>
<tr>
<td>6.3.1. Robust Normal Estimation</td>
<td>46</td>
</tr>
<tr>
<td>6.4. Projection</td>
<td>47</td>
</tr>
<tr>
<td>6.5. Kernel Selection</td>
<td>47</td>
</tr>
<tr>
<td>6.6. Higher Order Derivatives</td>
<td>47</td>
</tr>
<tr>
<td>6.7. Medial Axis</td>
<td>48</td>
</tr>
<tr>
<td>7. Conclusions</td>
<td>49</td>
</tr>
<tr>
<td>A. Local Kernel Regression</td>
<td>51</td>
</tr>
<tr>
<td>A.1. Deriving Adamson et al.’s [AA03] MLS surface definition from LKR</td>
<td>51</td>
</tr>
<tr>
<td>Bibliography</td>
<td>53</td>
</tr>
</tbody>
</table>
List of Figures

1.1. Reconstructions with the Proposed Surface Definition ........................................ 2
2.1. MLS Projection Procedure ............................................................................. 8
4.1. Some Objective, Influence and Weight Functions for M-Estimation ................. 24
5.1. Normals Near A Sharp Feature ........................................................................ 32
5.2. Sharp Feature Preservation .............................................................................. 35
5.3. Sharp Feature Preservation for Difficult Cases ............................................... 36
5.4. Geometric Detail Preservation for the Ramses Model ...................................... 37
5.5. Geometric Shape Preservation for the Filigree Model ...................................... 38
5.6. Sharp Features Under Sparse Sampling ............................................................ 38
5.7. Reconstruction, Curvatures, and Convergence for the Screwdriver Model ...... 39
5.8. Sharpness Control and Convergence the Max Planck Model ............................ 40
5.9. Reconstructions of the Isidore Horse and a Greek Sculpture Model ................ 41
5.10. Convergence the Bimba Model ....................................................................... 42
5.11. Curvatures for the Ramses Model .................................................................. 43
5.12. Geometric Detail Preservation for the Bimba Model ....................................... 44
List of Tables

3.1. Some Kernel Functions ........................................ 15
Introduction

Polygonal models have been in vast use in computer graphics to process and render geometry. However, in the last decades, point based representations are gaining popularity. This is mainly due to the growing complexity of shapes, the huge amount of data obtained through several systems such as scanning, and the shrinking sizes of the polygons. The simplicity of point based representations due to the lack of topological information, and the resulting flexibility are other fundamental driving motivations behind this trend.

There are several open problems in the geometric pipeline when using point sets. The acquired point cloud data is usually noisy, contains outliers, and incomplete. Often cleaning, registering different data to a common coordinate frame and then smoothing and thinning is necessary before one arrives at the final point set surface that can be used for further processing and modeling. A fundamental problem in this pipeline is how to reconstruct the end surface mathematically or algorithmically so that one gets an efficient and effective representation.

Implicit surface reconstruction is observed to produce stable and effective surfaces. Constructing such a representation is also easier than constructing a parametric surface. If one works with parametric representations, the independent variables live in a parameter space, of which we have no samples. Estimating a parametric surface from point sets is thus a harder problem. This is expected since it corresponds to local or global parametrization, which is known to be difficult. In contrast, for the implicit representations, the domain is the spatial domain where the sampled points live, and the range is the functional values of an implicit function, whose isocontours define the surface.

Since we are dealing with points that sample an unknown function, it is natural to expect that methods from scattered data approximation/interpolation and regression literature can be borrowed and adapted to the problem of implicit surface reconstruction. The main challenge in this approach is to evaluate and select the best methods for the geometric setting at hand. Since
1. Introduction

the data is typically noisy, contains outliers, and the sampling uniformity and density may not be known, selecting best techniques becomes a difficult task and often several intuitive extensions without theoretical basis are employed. However, the success of implicit reconstruction methods suggests that this is a promising way of reconstructing surfaces.

In this thesis, we follow this approach and use techniques borrowed from regression. To get most of the theory behind, we try to cast the implicit surface reconstruction problem in a regression setting such that several methods from regression literature can be adapted and the resulting extensions have theoretical foundations. In particular, we demonstrate the links between a popular implicit representation, moving least squares (MLS) surfaces, and local kernel regression (LKR), and then use techniques developed for robust kernel regression to reach a novel surface representation. We show that the novel definition preserves high frequency features while defining a continuous surface. It also avoids most of the artifacts of the current definitions such as shrinking/expanding, and is efficient to compute. The resulting definition is iterative but in closed form, and thus can be utilized to derive accurate analytic formulas for derivatives. Illustrating this, we also derive the principle curvatures and directions of the defined surface.

Although we focus on the experiments and results of this novel surface definition, the derived links between LKR and MLS open ways to many other interesting extensions. Potentially, almost all extensions proposed in statistics that can be used with LKR can be adopted. We discuss some of these promising ideas as well.

In the following chapter, we first give some preliminary results on implicit surfaces from differential geometry and fundamental concepts about MLS surfaces. The link between local kernel regression and MLS surfaces are derived next. Then we establish theoretical facts from robust statistics and use these facts and the links derived between MLS and LKR to arrive at the novel surface definition. We finish by investigating many other possible extensions using the general framework.

1.1. Contributions

Our main contributions are the following:
1.1. Contributions

- We prove that MLS surfaces can be derived using LKR. This opens various ways to improve MLS surfaces using techniques from LKR literature.

- We show that many feature preserving mesh and point cloud smoothing techniques can be explained in terms of robust statistics.

- A novel MLS surface definition is proposed, which
  - Preserves high frequency features
  - Has controllable sharpness of features
  - Reduces artifacts
  - Produces stable surfaces under sparse sampling
  - Is computed efficiently
  - Has an analytical expression

- Accurate equations for the curvatures of the proposed surface are derived.
1. Introduction
Preliminaries

In this chapter, we state theorems and facts from differential geometry of implicit surfaces as well as fundamental concepts, ideas and definitions for MLS surfaces.

2.1. Differential Geometry of Implicit Surfaces

We briefly state the fundamental definitions, facts and theorems of the differential geometry of implicit surfaces that we use in the thesis. Note that only the parts of the field that will be useful in understanding the techniques in the following chapters will be presented. For a general and extensive discussion, we refer the reader to [Car76].

2.1.1. Submanifolds

In general, we will work on submanifolds of dimension two embedded in three dimensional space in this thesis. Since we want to construct an implicit definition for the submanifold, we now give a formal definition and properties of this map such that it defines a submanifold. For the details of the concepts used in this section, please see [Car76].

Let $W \subset \mathbb{R}^n$ be an open set and $f : W \rightarrow \mathbb{R}^k$ be a map that is a $C^\infty$ diffeomorphism and $n \geq k$. A point $x \in \mathbb{R}^n$ of this map is called a regular point if the differential $Df_x$ is surjective. A value $v \in \mathbb{R}^k$ of this map is called a regular value if all points $x \in f^{-1}\{v\}$ are regular points.

If $v \in \mathbb{R}^k$ is a regular value of the map $f$, then the set of points $f^{-1}\{v\}$ defines a submanifold of $\mathbb{R}^n$ that has dimension $n - k$. 
2. Preliminaries

2.1.2. Implicit Surface Definition

For a two dimensional surface in three dimensional space, $n = 3$ and $k = 1$. Thus the differential is simply the gradient $\nabla f$. The surjectiveness of this differential implies that $\nabla f(x) \neq 0 \ \forall x \in f^{-1}(v)$ for a regular value $v$. Intuitively, each of these regular values defines an isocontour of the function and the points that belong to these isocontours form the resulting surfaces. For the rest of the paper, we will assume that 0 is a regular value of this function and use that regular value for the surface definition. Hence, the surface $S$ is defined as

$$S = \{ x \in \mathbb{R}^3 | f(x) = 0 \}$$

with the condition that $\nabla f(x) \neq 0$ for $x \in S$.

2.1.3. Signed Distance Fields

A signed distance field is a function $f(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that at each point $x \in \mathbb{R}^3$, it gives the signed Euclidean distance to the closest point of $x$ on the surface $S$ that is defined by the set of points that belong to the isocontour of regular value 0. Since the surface divides the space into two disconnected regions, one can define two distinct regions as outside and inside and select one sign for each region. In this thesis, the sign is positive outside the surface. Formally, a signed distance field is defined as

$$f(x) = \text{sign}(x) \inf_{s \in S} ||x - s||$$

where $\text{sign}(x)$ is 1 if $x$ is outside the surface $S$, and -1 otherwise.

The gradient of this signed distance field satisfies the Eikonal equation

$$||\nabla f(x)|| = 1$$

This equation is not satisfied only at the points on the medial axis. These points do not have a unique closest point on the surface $S$ and the gradient is not defined at those points. Since the magnitude of the gradient is one everywhere except at these points, it is clear that almost every value $v \in \mathbb{R}$ of a signed distance field is a regular value and thus every isocontour is a submanifold. If we take a particular value such as 0, the signed distance field defines a unique surface.

We typically want that the implicit function also approximates a signed distance field so that we guarantee the manifoldness and the function has intuitive interpretation. Note that in general this may not be the case for an arbitrary implicit function whose zero set defines a surface. It is shown [91] that even if the implicit function does not satisfy this property, the function divided by the magnitude of the gradient will still be a first order approximation of the Euclidean distance to the surface, thus a signed distance field.
2.1.4. Differentials

First Order

Gradient and normalized gradient are the quantities used to describe the first order derivatives of the surface. The normal is defined as the normalized gradient provided that the magnitude of the gradient is not zero.

\[ n = \frac{\nabla f}{||\nabla f||}, \quad ||\nabla f|| \neq 0 \]  \hspace{1cm} (2.4)

Second Order

The Hessian matrix of the function \( f(x) \) contains second order derivatives and is defined as

\[ H = \begin{bmatrix}
\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\
\frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\
\frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2}
\end{bmatrix} \]  \hspace{1cm} (2.5)

Using this expression and the normalized gradient (normal), one can define the Weingarten Map or Shape Operator as follows [Sig06]:

\[ W = \nabla n = \frac{1}{||\nabla f||} (I - nn^T) H \]  \hspace{1cm} (2.6)

The eigenvalues and eigenvectors of this matrix give the principle curvatures and curvature directions of the surface. One of the eigenvalues of this matrix is 0 and the corresponding eigenvector is the normal. The other two eigenvalues are the minimum and maximum curvatures of the surface and the corresponding eigenvectors are the principle directions.

2.2. Moving Least Squares

Moving Least Squares (MLS) [She68] is a functional approximation method for irregular samplings. It produces local approximations of the function around the point of interest by minimizing a weighted sum of squares error metric. For the functional setting, MLS reconstruction of a function \( f \) for a sample set \( \{x_i, y_i\} \) is defined as follows:

\[ f_{MLS}(x) = \min_f \sum (f(x_i) - y_i)^2 \phi(||x - x_i||) \]  \hspace{1cm} (2.7)

The weight function \( \phi \) is a decreasing function to give samples close to the evaluation point \( x \) more weight in the minimization. This key property distinguishes MLS from ordinary least squares estimators.
2. Preliminaries

![Figure 2.1.: MLS Projection Procedure.](image)

(a) First, a plane is fit to the set of points using a weighted least squares procedure. (b) Using the fitted plane as a reference domain, a polynomial is fit by again a weighted least squares minimization. (c) The point is projected onto this polynomial.

2.2.1. Moving Least Squares Surfaces

The adoption of MLS method to 3D surface reconstruction is more recent [ABCO+01]. The surface is defined implicitly by a projection procedure. At each step, a local plane is fit to the data using a weighted least squares error minimization, and a local parametrization of the surface over this plane is computed by fitting a bivariate polynomial again in a weighted least squares sense. Since the weights in both steps depend on the projected point, the resulting projection is an MLS approximation of the actual projection on the surface. The local parametrization is directly motivated by the fact from differential geometry of surfaces that a submanifold can be locally parametrized over the tangent plane. The two steps of the algorithm are presented in Figure 2.1.

In the following works, it was observed that one can actually skip the step of polynomial fitting and define the distance to the locally fit plane as the distance to the surface [AK04]. This avoids the problem of instabilities resulting from the polynomial fit and produces more intuitive definitions.

The first simplified definition using plane fitting was introduced by [AA03]. After the link between this surface and the MLS surface definition of [ABCO+01] was established [AK04], it was extended to be used with average of normals and locations of samples [AA04]. This definition of MLS can be described mathematically as:

\[ f(x) = n(x)^T(x - c(x)) \]  

(2.8)

where \( c \) is the weighted average of the sample locations and \( n \) can be either computed using weighted averaging of normals, or fitting a local plane using covariance analysis.

With the realization of the fact that these MLS surfaces use planes as proxies, a general algebraic surface fitting framework [GG07] was introduced to improve the stability and robustness by using higher order algebraic surfaces. Guennebaud et al. use this framework [GG07] to fit a sphere instead of a plane and use the algebraic distance to this sphere as the distance to the surface. Their algebraic surface fitting framework is trivial to extend to higher order surfaces as well. However, since the fitting requires solving a generalized eigenvalue problem, a more
2.2. Moving Least Squares

An efficient procedure was formulated in the presence of normals for fitting a sphere \cite{GGG08}. MLS surface definitions using local algebraic surface fitting produce the implicit function for the surface. Since we have the analytic expression, several operations and differential properties can also be derived. If one needs the projection procedure for example, the gradient can always be used to determine the projection direction. Inside/outside tests are also implemented trivially, provided that sample normals are available. Thus in this implicit form, as opposed to the projection form, MLS surface definitions are more useful.

2.2.2. Constraints

All implicit MLS surface definitions discussed so far need constraints to avoid the trivial solution of $f(x) = 0 \ \forall x$ and to ensure a good fit. Constraining the magnitude of the gradient works well for both plane and sphere fitting. But it was also shown \cite{Kol05} \cite{GGG08} that in the presence of normals, constraining the gradient to be close to the normals at the sample points (normal constraints) produces both practically and theoretically nice surfaces. The evaluation of these surfaces are also efficient since unlike other definitions, no linear system needs to be explicitly solved. The major obvious drawback is that normals should be provided or estimated. The algorithms proposed for normal estimation typically fit a plane \cite{HDD+92} or sphere \cite{GG07} locally and then propagate the orientation of the normals to arrive at a consistent global orientation. Thus, if the normals are not known in advance (for example, from the scanning process), the efficiency of the MLS surface definitions depends on the efficiency of the off-line normal estimation step.

Among the formulations using normal constraints, we interpret and extend Kolluri’s \cite{Kol05} definition, which is based on the work by Shen et al. \cite{SOS04} and which has been proved to generate tight and manifold surfaces under uniform and sufficiently dense sampling conditions \cite{Kol05}. Unfortunately, the resulting surface is known to have artifacts such as expanding and shrinking \cite{GG07} and like other existing MLS surface definitions, high frequency parts such as sharp features and fine geometric details are not preserved. We show that our novel MLS surface solves these problems.
2. Preliminaries
Regression and MLS Surfaces

Recently, it has been observed in computer vision community that moving least squares approximation of a function is a special case of local linear regression [TFM07]. For the geometric setting in computer graphics, the link between moving least squares and regression methods has not been set up yet. In this chapter, we will establish this link, which will form a theoretical basis for our new MLS definition in the following chapters.

In dealing with regression methods for the surface reconstruction setting, one should differentiate between supervised and unsupervised function estimation. Supervised function estimation corresponds to estimating the implicit function from the samples, assuming that at some points in the space the value of the distance function is known. Whereas unsupervised regression can be utilized to fit a surface without using any explicit distance function. Since unsupervised regression for manifold learning is a fairly new area [MKMR05] [KR07], the methods are not efficient and effective yet and a direct link between MLS surfaces seem to be more difficult. Hence, we focus on supervised regression.

Among the supervised regression methods, we will concentrate on non-parametric regression methods, kernel regression in particular. The reason is that these methods directly relate to MLS, as will be established in the following sections. In fact, MLS based surface definitions can be directly derived from kernel regression techniques. We will present these derivations and possible extensions and generalizations. Thus, this link does not only provide a theoretical basis for MLS surface definitions, but also have practical implications.

If one does regression by taking values of the implicit function at the sample points as zero (since they are almost on the surface), the result is the trivial solution of zero for the implicit function. Thus one needs constraints, or in regression terms, regularization or penalty functions to avoid the trivial solution. These will be investigated and incorporated into the minimizations to get MLS definitions of surfaces.
3. Regression and MLS Surfaces

3.1. Kernel Regression

Estimating the density of a random variable using the so-called kernel functions is known as kernel density or Parzen window estimation. This setting was then extended for estimating arbitrary functions. The pioneering work by Nadaraya et al. [Nad64] accomplishes this by treating the dependent and independent variables as random variables and estimating the conditional expectation of the dependent variable. (We will derive this definition and relate it to other definitions in section 4.3.1.) Practically in the simplest form, kernel regression estimates a function sampled at some points by placing kernel functions at those points and using a weighted sum of the kernels. Formally, this means the function \( f(x) \) can be estimated by using the set of samples \( \{x_i, y_i\} \) as:

\[
f(x) = \frac{\sum y_i \phi_i(x)}{\sum \phi_i(x)} \tag{3.1}
\]

where \( \phi_i(x) = \phi(||x - x_i||) \) and \( \phi(x) \) is the kernel function (for a discussion and properties of the kernel functions, see 3.2.1).

The above formulation corresponds to a supervised setting where sample points and the values of the function at the sample points are known. In fact, one can also derive expressions for the unsupervised setting where the actual sample points are not known. To our knowledge, this setting has not been explored in computer graphics literature yet. We briefly explain the application of this method in the next section and defer the actual implementation and investigation of the method as a future research project.

3.1.1. Unsupervised Kernel Regression

In the setting of unsupervised kernel regression for surface reconstruction, the unknown variables can be assumed to live in the parameter space and the sample points we have correspond to the values of the function to be estimated. Thus this time the function \( f : \mathbb{R}^2 \to \mathbb{R}^3 \) is the parametric representation of the surface. We want to estimate this function using

\[
f(x) = \frac{\sum y_i \phi_i(x)}{\sum \phi_i(x)} \tag{3.2}
\]

where \( x_i \in \mathbb{R}^2 \) is in the parameter space and \( y_i \in \mathbb{R}^3 \) is the sample point we have for the parameter \( x_i \). Note that in this case \( x_i \)'s are unknown, which suggests a non-linear minimization to solve. Although there are some recent papers that does this non-linear minimization [MKMR05] [KR07], the robustness and stability of the approach has not been extensively studied yet. Partly for this reason and partly because it does not have direct relation to MLS surfaces, we will focus on supervised kernel regression techniques in this thesis.
3.2. Local Kernel Regression

3.1.2. Supervised Kernel Regression

In contrast to unsupervised kernel regression, supervised kernel regression can be used to estimate an implicit function of a surface. In this case, the value of the function at the sample values are assumed to be zero (since they are almost on the surface), and the implicit function is constructed using these set of \( \{x_i, 0\} \) pairs.

However, naively using the Nadaraya-Watson estimator \([\text{Nad64}]\) gives the trivial solution of zero for the function. This is also true for other local kernel regression settings, of which Nadaraya-Watson estimator \([\text{Nad64}]\) is a special case. To avoid the trivial solution, one needs constraints on the function. Several successful constraints have been proposed for MLS surface reconstruction. Nevertheless, for the regression setting it is not immediately clear how to select them.

3.2. Local Kernel Regression

Local Kernel regression (LKR) is a supervised regression method to approximate a function given its values at sampled points possibly corrupted with noise. LKR uses local fits by utilizing Taylor expansions around the point of interest and minimizing a weighted sum of squares to get the estimated function. The function to be estimated is assumed to be smooth to some degree. Since it is a nonparametric method, no assumptions are made about the distribution of the points and thus it can be used generically for many tasks. Its adoption and use in computer vision is very recent \([\text{TFM07}]\), and to our knowledge, there is no work done by explicitly using LKR in computer graphics community.

Stating formally, we want to approximate \( f(x) : \mathbb{R}^d \rightarrow \mathbb{R} \) given

\[
y_i = f(x_i) + \epsilon \quad (3.3)
\]

where \( \epsilon \) is a random variable with zero mean. If we expand \( f(x_i) \) around \( x \) we get

\[
f(x_i) = f(x) + \nabla f(x)^T(x_i - x) + \frac{1}{2}(x_i - x)^T H f(x)(x_i - x) + \ldots \quad (3.4)
\]

Here, \( H f(x) \) denotes the Hessian of \( f(x) \). The order \( o \) of the expansion is defined as the number of terms used in the Taylor expansion minus 1. Thus for \( o = 0 \), we only have \( f(x_i) = f(x) \). We can rewrite this equation as

\[
f(x_i) = s_0 + a_i^T s_1 + b_i^T s_2 + \ldots \quad (3.5)
\]

where \( a_i = (x_i - x) \) and \( b_i = \text{vech}((x_i - x)(x_i - x)^T) \). For a symmetric matrix \( A \), \( \text{vech}(A) = [A_{11} A_{12} A_{22}]^T \) if \( A \) is 2 by 2 and \( \text{vech}(A) = [A_{11} A_{12} A_{13} A_{22} A_{23} A_{33}]^T \) if \( A \) is 3 by 3.

Since we can only use finite number of terms, this expansion will be most accurate at the samples around the point \( x \). This suggests using a weighted least squares approach to find the
unknown parameters, which are the function itself ($s_0$), gradient ($s_1$), Hessian ($s_2$), etc.

$$\min_{s_k} \sum (y_i - (s_0 + a_i^T s_1 + b_i^T s_2 + \ldots))^2 \phi_i(x)$$  \hspace{1cm} (3.6)$$

Here $\phi_i(x) = \phi(||x - x_i||)$ and $\phi(x)$ is a kernel. This minimization is equivalent to minimizing the following expression

$$(y - Xs)^T \Phi (y - Xs)$$  \hspace{1cm} (3.7)$$

$$y = [y_1 \ y_2 \ \ldots \ \ y_n]^T$$

$$X = \begin{bmatrix}
1 \ (x_1 - x)^T \ \text{vech}((x_1 - x)(x_1 - x)^T)^T \ \ldots \\
1 \ (x_2 - x)^T \ \text{vech}((x_2 - x)(x_2 - x)^T)^T \ \ldots \\
\vdots \quad \vdots \quad \vdots \quad \ldots \\
1 \ (x_n - x)^T \ \text{vech}((x_n - x)(x_n - x)^T)^T \ \ldots 
\end{bmatrix}$$

$$s = [s_1 \ s_2^T \ \ldots \ \ s_{o+1}^T]^T$$

$$\Phi = \begin{bmatrix}
\phi_1(x) & 0 & 0 & \cdots \\
0 & \phi_2(x) & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots \\
0 & 0 & \cdots & \phi_n(x)
\end{bmatrix}$$

Since LKR is also a kernel based method as MLS, and since both uses local approximations to the function, it is natural to expect a direct relation between the two. As will be shown in the next section, existing MLS surface definitions can be directly and exactly derived from the LKR framework.

### 3.2.1. Kernel Functions

LKR depends on a kernel function. Selection of a kernel function for a specific problem is not a trivial task but there are several properties that kernel functions should satisfy for LKR to produce meaningful results. Denoting the kernel function as $\phi$, these can be summarized as:

- $\phi(x) = \phi(-x)$
- $\int_{-\infty}^{\infty} x \phi(x) dx = 0$
- $|x_1| < |x_2| \Rightarrow \phi(x_1) > \phi(x_2)$
3.3. MLS Surfaces from LKR

### Table 3.1.: Some Kernel Functions.

<table>
<thead>
<tr>
<th>Kernel Name</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>$\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$</td>
</tr>
<tr>
<td>Uniform</td>
<td>$\frac{1}{2} 1(</td>
</tr>
<tr>
<td>Triangle</td>
<td>$(1 -</td>
</tr>
<tr>
<td>Quartic</td>
<td>$\frac{15}{16}(1 - x^2)^2 1(</td>
</tr>
<tr>
<td>Triweight</td>
<td>$\frac{35}{32}(1 - x^2)^3 1(</td>
</tr>
<tr>
<td>Epanechnikov</td>
<td>$\frac{3}{4}(1 - x^2) 1(</td>
</tr>
<tr>
<td>Cosine</td>
<td>$\frac{\pi}{4} \cos \left( \frac{\pi}{2} x \right) 1(</td>
</tr>
</tbody>
</table>

The first property ensures that $\phi$ is symmetric around the origin, the second says that it is centered around zero, and the third implies that it is decreasing when moving away from the origin. This is crucial since otherwise the locality of the functional fitting would be lost. For density estimation and Nadaraya-Watson estimator [Nad64], the property $\int_{-\infty}^{\infty} \phi(x)dx = 1$ is also important, but for LKR this does not need to be satisfied. As examples of kernels, we tabulate some widely used ones in Table 3.1 in normalized forms. In practice, for LKR, the preceding constants can be omitted and the independent variable should be scaled to fit a certain kernel width. Although these kernels are common in kernel based density estimation and regression, the far most popular one for computer graphics and vision is the Gaussian and others that approximate it.

3.3. MLS Surfaces from LKR

In the problem of implicit surface reconstruction from point clouds, we want to approximate the signed distance function to the surface given points sampled from the surface. The points may contain noise and outliers. To approximate the signed distance function using LKR, we assume that the sampled points are close to the surface so that $f(x_i) \approx 0$. Since this will imply to take $y_i = 0$, we should impose some constraints or regularization to avoid the trivial solution of $s_k = 0 \ \forall k$.

In the context of surface fitting, several constraints have been proposed including the normal [SOS04] [Kol05] [GGG08] and the gradient [AA03] [GG07] constraints. For the normal constraints, we incorporate the normals at the sample points to constrain the gradient, hence we need to have or approximate the normals. For the gradient constraint we have $||\nabla f(x)|| = ||s_i|| = 1$.

Using these constraints, we will derive two MLS surface definitions in the next sections.
3. Regression and MLS Surfaces

3.3.1. Deriving Kolluri’s [Kol05] MLS definition

One can derive Kolluri’s [Kol05] MLS definition starting from LKR with normal constraints of [SOS04] and $o = 0$. For this case, the minimization becomes

$$\min_{s_0} \sum (y_i - s_0)^2 \phi_i(x) = \min_{s_0} \sum (n_i^T(x - x_i) - s_0)^2 \phi_i(x)$$  \hspace{1cm} (3.8)

The solution of this minimization is simply

$$f(x) = s_0 = \frac{\sum n_i^T (x - x_i) \phi_i(x)}{\sum \phi_i(x)}$$ \hspace{1cm} (3.9)

which is exactly Kolluri’s definition of the MLS surface.

Although it is quite straightforward to derive this definition using the normal constraints of [SOS04], this constraint can be interpreted as $y_i = f(x_i) = n_i^T(x - x_i)$, which is neither intuitive nor mathematically justified in the setting of LKR, since we know that the value of the implicit function at the sample points should be almost zero, not $n_i^T(x - x_i)$. Thus we derive Kolluri’s definition in another way which is geometrically sound in the framework of LKR. If we expand the gradient as:

$$\nabla f(x_i) = \nabla f(x) + H f(x_i)(x_i - x) + \cdots$$ \hspace{1cm} (3.10)

and take a zero order expansion of the gradient, then

$$\nabla f(x_i) = \nabla f(x)$$ \hspace{1cm} (3.11)

Since we assume that the function $f$ approximates a signed distance function, we can set $\nabla f(x_i) = n_i$, where $n_i$ is the normal at the sample point $x_i$. In this case, the minimization for $o = 1$ and $y_i = 0$ becomes

$$\min_{s_0, s_1} \sum (y_i - (s_0 + a_i^T s_1))^2 \phi_i(x) = \min_{s_0} \sum (s_0 + a_i^T \nabla f(x_i))^2 \phi_i(x)$$ \hspace{1cm} (3.12)

$$= \min_{s_0} \sum (s_0 + a_i^T n_i)^2 \phi_i(x)$$

It is easy to see that the solution of this minimization is again Kolluri’s definition.

3.3.2. Deriving Adamson et al.’s [AA03] MLS definition

We consider the case with the gradient constraint $||\nabla f(x)|| = ||s_1|| = 1$ and $o = 1$. In this case the minimization becomes
3.3. MLS Surfaces from LKR

\[
\min_{s_0,s_1} \sum (y_i - (s_0 + a_i^T s_1))^2 \phi_i(x) = \min_{s_0,s_1} (s_0 + a_i^T s_1)^2 \phi_i(x) \quad (3.13)
\]

given the constraint \( ||s_1|| = 1 \).

The solution of this minimization results in the following equations (see Appendix [A.1] for details)

\[
(C - cc^T)s_1 = 0, \quad ||s_1|| = 1 \quad (3.14)
\]

\[
s_0 = s_1^T(x - c) \quad (3.15)
\]

where

\[
c = \frac{\sum x_i \phi_i(x)}{\sum \phi_i(x)} \quad (3.16)
\]

and

\[
C = \frac{\sum x_i x_i^T \phi_i(x)}{\sum \phi_i(x)} \quad (3.17)
\]

Thus the solution \( s_1 \) is the eigenvector of the matrix \( (C - cc^T) \) corresponding to the smallest eigenvalue. Note that \( s_1 \) corresponds to the normal of the plane computed by weighted least squares fit to the samples as

\[
\min_{n,d} \sum (n^T x_i - d)^2 \phi_i(x) \quad (3.18)
\]

and \( s_0 \) is the MLS surface defined by Adamson et al. [AA03]. This weighted least squares plane fit is not exactly the same as the one used in [AA03], but corresponds to the plane fitting to compute the reference domain in Levin’s definition [ABCO +01] with weights depending not on the projected point but the point \( x \). This way of fitting a plane has been used for various modeling tasks too [PKKG03].

Normal Averaging

For \( s_1 \) in Adamson et al.’s definition [AA03], they also use the weighted average of the normals at the sample points, if they are available. We can also derive this form by considering zero order LKR of the gradient. First we expand the gradient as

\[
\nabla f(x_i) = \nabla f(x) + H f(x)(x_i - x) + \cdots \quad (3.19)
\]
3. Regression and MLS Surfaces

Now we can do a zero order LRK on the gradient. This corresponds to the minimization

\[ \sum \| \mathbf{n}_i - \mathbf{s}_1 \|^2 \phi_i (\mathbf{x}) \]  

(3.20)

If we first perform this minimization on the gradient, we get \( \mathbf{s}_1 \) as the weighted average of the sample normals. Then the minimization on the implicit function again gives \( s_0 = s_1^T (\mathbf{x} - \mathbf{c}) \), which is Alexa et al.’s definition with normal averaging.

3.3.3. Implications

These two cases draw a link between LKR and MLS surfaces. MLS definitions with higher order proxies may also be derived using the same framework. This link not only provides new theoretical justifications for the definitions, but also suggests to use different orders of regression and constraints to make new MLS surface definitions as well as utilizing methods and extensions developed for local kernel regression.

In the scope of this thesis, we incorporate robust kernel regression techniques into this framework to arrive at the proposed novel surface definition.
Robust Statistics for Local Kernel Regression

After establishing the link between MLS surfaces and LKR, in this chapter, we move on to well-studied techniques for kernel regression in the context of robust statistics. This is only one possible extension of MLS surfaces using LKR. We select this extension particularly because our novel MLS surface definition will be based on the theory we present here.

This chapter is mainly about the theoretical aspects and foundations related to robust kernel regression such that the results can be used in the context of MLS surfaces. On the way, several theoretical results are shown, and various techniques used in computer graphics and vision are interpreted in the same framework. Namely, we present that robust local kernel regression (RLKR) can be used to derive non-linear smoothing filters that preserve high frequency features of geometry.

We start with a discussion on robust statistics and robust kernel regression techniques and then explain how different high frequency feature preserving smoothing filters can be linked to robust statistics.

4.1. Robust Statistics

Robust statistics deals with outliers in the data. Classical statistics often assumes that the data follows a model and is outlier-free. But real world data almost always contains outliers. Even one outlier in the data can influence a classical statistical measure significantly, making it useless. Instead of depending on procedures to remove the outliers, robust statistics achieves robustness with respect to outliers by defining new statistical quantities that are not influenced by
4. Robust Statistics for Local Kernel Regression

them. In this section, we will touch upon robust statistics and explain the concepts relevant to our scope. An in depth analysis of the methods can be found in e.g. [Hub04].

To get a first intuitive feeling of robust measures, consider the well-known statistical measure, median. In contrast to mean, median is not affected even if fifty percent of the data are outliers. As an example, consider the mean of a set of numbers \(1, 2, 3\). Both the mean and median is \(2\). If we add two outliers \(10000\) and \(20000\) to this set, the new mean is \(6001.2\) and the median is \(3\). The two outliers added does not change median much while moving the mean completely from the actual data. This behavior is even more clear if one considers the case where one of the numbers is \(\infty\). In this case, this one number makes the mean \(\infty\), while almost not affecting the median.

4.1.1. Breakdown Point

To determine which measures are robust, one needs to define a criterion. Breakdown point is the mostly used one. Informally, it is defined as the fraction of outliers that can be added to the data before the statistical method starts to give arbitrarily bad results. For example, since mean can be made arbitrarily large by adding even one outlier, its breakdown point is \(0\%\), whereas the median has a breakdown point of \(50\%\) because even if half of the data are outliers, median will still stay close to the real data.

Note that \(50\%\) is the maximum breakdown point that can be achieved. If number of outliers is more than \(50\%\) of the data, it is not possible to distinguish between real data and outliers since outliers become the majority in the data. Thus the optimal breakdown point is \(50\%\) and median achieves this optimal.

4.1.2. Robust Kernel Regression Techniques

Regression tries to estimate a model using the data at hand. Specifically, we want to find a function that fits the data well while satisfying some constraints. Here we focus on non-parametric supervised regression, and in particular local kernel regression.

Local kernel regression tries to fit a function to the data by minimizing a weighted least squares sum of the deviations of the functional values at the sample points from the corresponding observed values. Consider an outlier in the data, this outlier can move the fitted function arbitrarily from the actual data points. Thus minimizing sum of squares is not robust to outliers. To overcome this limitation, several methods have been proposed including the Least Median of Squares, Least Trimmed Squares, and Repeated Median (see e.g. [Mor07]). The major problem with these approaches is that the functions to be minimized are not differentiable and thus there is no closed form iterative or non-iterative solution of the minimization as in the case of least squares. In fact, often random sampling is employed to find the minimum point. The significance of this problem for our setting is, the definitions become less efficient and cumbersome and one cannot get a final analytical expression of the function. This has significant drawbacks as will be explained in 5.1.2.
4.2. Robust Local Kernel Regression

Our goal is to use a regression method that is resistant to outliers. This method should have a formulation that allows to derive closed form iterative formulas such that we get an analytical implicit surface function in the end. To achieve this goal, we use $\psi$-type M-Estimators [Hub04]. M-Estimation is a generalization of maximum likelihood estimation. Instead of the ordinary least squares criterion, it minimizes a different objective function. First we state the idea and properties of this estimation, and then show how it can be used with LKR to get a robust kernel estimator.

4.2.1. Iteratively Reweighted Least Squares

Iteratively Reweighted Least Squares (IRLS) is a general procedure to find parameters of a model by minimizing an objective function to arrive at an M-Estimator. A familiar example of it is the well known least squares estimator. IRLS generalizes the least squares criteria to obtain the robust parameters.

Suppose we are given some data $\{x_i, y_i\}$. Then M-Estimators minimize the general function

$$\min_s \sum \rho(x_i, y_i, s)$$

(4.1)

Writing this expression more explicitly, if we denote the residual of the $i^{th}$ sample as $r_i$, M-Estimators minimizes

$$\min_s \sum \rho(r_i)$$

(4.2)

The residual is defined as the deviation of the observed dependent variable from the estimated one at the sample points. If a function $f(x)$ is to be estimated by the procedure, then the residual is $r_i = f(x_i) - y_i$. In this setting, least squares estimation is a special case with $\rho(r_i) = r_i^2$. This minimization can be solved by differentiating with respect to the parameter vector $s$. This requires the function $\rho$ to be differentiable, which is the case for $\psi$-type M-Estimators. Denoting $\Omega = \sum \rho(r_i)$

$$\frac{\partial \Omega}{\partial s} = \sum \psi(r_i) \frac{\partial r_i}{\partial s} = 0$$

(4.3)

where $\psi(x) = \frac{\partial \rho}{\partial x}$ and is called the influence function. We rewrite this equation by using the weights $w(x) = \psi(x)/x$

$$\sum w(r_i) r_i \frac{\partial r_i}{\partial s} = 0$$

(4.4)

Now consider the following minimization problem where we fix the weights $w(r_i)$
4. Robust Statistics for Local Kernel Regression

\[ \min_s \sum w(r_i) r_i^2 \]  

(4.5)

The solution of this minimization is

\[ \sum w(r_i) r_i \frac{\partial r_i}{\partial s} = 0 \]  

(4.6)

which exactly corresponds to (4.4). Thus (4.4) can be solved by iteratively minimizing the following

\[ \min_s \sum w(r_i^{k-1}) r_i^2 \]  

(4.7)

where \( r_i^{k-1} \) is the \( i \)th residual at the \( k-1 \)th iteration. This means that at each iteration the weights are fixed, and a weighted least squares problem is solved. Then new weights are computed using the new residuals and the resulting new weighted least squares system is solved. Residuals determine the weights, while weights determine the solution of the weighted least squares problem, suggesting the iterative procedure.

As an example of this procedure, suppose that we want to fit a function \( f(x) : \mathbb{R} \rightarrow \mathbb{R} \) to a data set \( \{x_i, y_i\} \) and we use a linear model for the function such that \( f(x) = ax \), where \( a \) is the parameter to be estimated. Thus in this case the residuals are \( r_i = ax_i - y_i \). Taking the derivative gives \( \frac{\partial r_i^2}{\partial a} = 2(ax_i - y_i)x_i \). If we wanted to get an ordinary least squares estimator, we would minimize \( \sum r_i^2 \) and thus had \( \sum (ax_i - y_i)x_i = 0 \), which gives \( a = \frac{\sum y_i x_i}{\sum x_i^2} \). Since we want to use an M-Estimator, we have \( a^k = \frac{\sum y_i x_i w(r_i^{k-1})}{\sum x_i w(r_i^{k-1})} \), where \( r_i^{k-1} = a^{k-1} x_i - y_i \) and \( a^{k-1} \) is the estimate of \( a \) at the previous iteration. Thus the only modification we make to the least squares minimization is the addition of the weights at each iteration.

**Initial Weights**

Let us denote the weights \( w(r_i^{k-1}) \) by \( w_i^{k-1} \). A trivial initial weight selection is to set all of them to 1 for the first iteration such that \( w_i^0 = 1 \). This leads to a standard least squares system to solve for the first iteration. It is well-known that these initial weights will not give an estimator with a high breakdown point and there are several methods to improve the estimator such as the Redescending M-Estimator [Hub04] or the more recent [HBH07]. However, in the scope of this thesis we will not use these extensions and use the initial weights \( w_i^0 = 1 \). We leave investigation of these different initial conditions as a future work.

**Weight Functions**

An obvious question about IRLS is the selection of weight functions. There are many different functions with particular properties proposed in the literature. Here, we review general properties that the functions should satisfy, and also some specific functions from the literature. We
call the function $\rho$ as the objective function, $\psi$ as the influence function and $w$ as the weight function.

Generally a good $\psi$-type M-Estimator should have the following properties:

- $\psi$ should be bounded.
- $\rho$ is convex in the parameter $s$.
- $\rho$ has a unique minimum at zero.
- $\rho$ is symmetric.
- $\rho$ is positive-definite.

Note that not all of these are satisfied for different M-Estimators. In Figure 4.1 we show some of the functions proposed in the literature. Although the choice for our setting is not clear, we chose to use Welsh’s function since it is known to eliminate large errors [Hub04], and the resulting weight function is a Gaussian, which is widely used in computer graphics and vision community for its unique spatial and frequency domain properties.

**IRLS and LKR**

IRLS, as formulated above, can be applied to all kernel regression estimators by just adding some weights that depend on the residuals. In our setting, we formulate the problem as a LKR minimization, which is weighted depending on the residuals. That is, we minimize

$$
\min_{s_m} \sum_i (y_i - (s_0 + a_i^Ts_1 + b_i^Ts_2 + ...))^2 \phi_i(x) w_i^{k-1}
$$

which is the same minimization as in (4.6) with added weights that depend on the residuals. This leads to an iterative solution, where we start with initial weights $w^0 = 1$ and update them as $w_i^{k-1} = w(y_i - (s_0^{k-1} + a_i^Ts_1^{k-1} + b_i^Ts_2^{k-1} + ...))$ at each iteration.

**4.3. Applications in Computer Graphics And Vision**

Robust statistics have been in use in computer graphics and vision, although most of the time unknowingly. Perhaps the most popular one is the Bilateral Filter [TM98], but there are also many others including MLS surfaces [FCOS05], point cloud projection operators [LCOLTE07], mesh denoising and smoothing techniques [FDCO03] [JDD03] [LW05]. In this section, we briefly review these applications to indicate the success of application of robust methods in the literature.

**4.3.1. Bilateral Filtering**

It has been observed that bilateral filtering has close links with robust statistics [Ela02]. This is also clear from the discussion in the previous section. In fact, as proposed originally in [TM98].
4. Robust Statistics for Local Kernel Regression

(a) Least Squares

(b) $L_1$ Type

(c) Cauchy

(d) Welsch

Figure 4.1.: Some objective, influence and weight functions for M-Estimation.
4.3. Applications in Computer Graphics And Vision

it corresponds to one iteration of IRLS with a zero order LKR, where the sampled points are the pixels and the functional values are the pixel colors. Not surprisingly, using ideas similar to bilateral filtering, it was then extended to be iterative [TFM07].

Here, we derive bilateral filter for the case of the functions \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) by two different approaches. The first one shows that it is a special case of the robust local kernel regression with IRLS, and the second one also gives a hint about its relation to maximum likelihood estimation, thus M-estimation.

**Derivation with LKR and IRLS**

Given the data set \( \{x_i, y_i\} \), consider the zero order LKR \((\omega = 0)\) of a function \( f \)

\[
    f(x) = \min_s \sum s - y_i)^2 \phi(||x - x_i||)
\]

and its solution

\[
    f(x) = \frac{\sum y_i \phi(||x - x_i||)}{\sum \phi(||x - x_i||)}
\]

This is a least squares metric minimization. To make it robust, one can use IRLS by adding weights that depend on the residuals and make the procedure iterative such that we get

\[
    f_k(x) = \frac{\sum y_i \phi(||x - x_i||) \alpha(||s_{k-1} - y_i||)}{\sum \phi(||x - x_i||) \alpha(||s_{k-1} - y_i||)} = \frac{\sum y_i \phi(||x - x_i||) \alpha(||f_{k-1}(x) - y_i||)}{\sum \phi(||x - x_i||) \alpha(||f_{k-1}(x) - y_i||)}
\]

Here, \( \phi \) and \( \alpha \) are kernel functions. Suppose that our data samples are regularly placed on a 2D grid (the image) and that we evaluate this expression for each point in that grid, then bilateral filtering for images corresponds to the first iteration of this iterative definition.

**Derivation with Nadaraya-Watson Estimator**

Nadaraya-Watson Estimator [Nad64], which is a special case of LKR, can be derived by the expected value of the probability \( p(y|x) \) for a data set \( \{x_i, y_i\} \).

\[
    f(x) = \mathbb{E}(y|x) = \int_{-\infty}^{\infty} y p(y|x) dy = \int_{-\infty}^{\infty} y \frac{p(y|x)}{p(x)} dy
\]

If we estimate the distributions \( p(y, x) \) and \( p(x) \) with kernel density estimation, we get
4. Robust Statistics for Local Kernel Regression

\[
\int_{-\infty}^{\infty} \frac{p(y, \mathbf{x})}{p(\mathbf{x})} \, dy = \int_{-\infty}^{\infty} \frac{p(y, \mathbf{x})}{p(\mathbf{x})} \, dy = \int_{-\infty}^{\infty} y \phi(||\mathbf{x} - \mathbf{x}_i||) \phi(y - y_i) \, dy \sum_{i} \phi(||\mathbf{x} - \mathbf{x}_i||) \\
= \sum_{i} \phi(||\mathbf{x} - \mathbf{x}_i||) \int_{-\infty}^{\infty} y \phi(y - y_i) \, dy \sum_{i} \phi(||\mathbf{x} - \mathbf{x}_i||)
\]

Making the parameter change \( z = y - y_i \)

\[
\int_{-\infty}^{\infty} y \phi(y - y_i) \, dy = \int_{-\infty}^{\infty} (z + y_i) \phi(z) \, dz = \int_{-\infty}^{\infty} z \phi(z) \, dz + y_i \int_{-\infty}^{\infty} \phi(z) \, dz
\]

Using the assumptions about the kernel function ( \( \int_{-\infty}^{\infty} z \phi(z) \, dz = 0 \) and \( \int_{-\infty}^{\infty} \phi(z) \, dz = 1 \) ), this expression becomes simply \( y_i \) and thus the expectation becomes

\[
f(\mathbf{x}) = \frac{\sum_{i} \phi(||\mathbf{x} - \mathbf{x}_i||) y_i}{\sum_{i} \phi(||\mathbf{x} - \mathbf{x}_i||)}
\]

which is the Nadaraya-Watson estimator.

Now, instead of expectation, we will use maximization of the probability \( p(y|\mathbf{x}) \) and we will assume that the kernel \( \phi \) is a Gaussian such that \( \phi(x) = e^{-x^2} \). The function is defined as

\[
f(\mathbf{x}) = \max_y p(y|\mathbf{x})
\]

If we do this maximization we get

\[
\frac{\partial p(y|\mathbf{x})}{\partial y} = \frac{\partial}{\partial y} \frac{\sum \phi(||\mathbf{x} - \mathbf{x}_i||) \phi(y - y_i)}{\sum \phi(||\mathbf{x} - \mathbf{x}_i||)}
\]

\[
\Rightarrow \frac{\sum \phi(||\mathbf{x} - \mathbf{x}_i||) \phi'(y - y_i)(y - y_i)}{\sum \phi(||\mathbf{x} - \mathbf{x}_i||)} = 0
\]

Then the final equation for the \( y \) that maximizes \( p(y|\mathbf{x}) \) becomes

\[
f(\mathbf{x}) = \frac{\sum \phi(||\mathbf{x} - \mathbf{x}_i||) \phi'(y - y_i)y_i}{\sum \phi(||\mathbf{x} - \mathbf{x}_i||) \phi'(y - y_i)}
\]

Since we assume that \( \phi' = \phi \) (\( \phi \) is Gaussian), this exactly corresponds to bilateral filtering.
4.3. Applications in Computer Graphics And Vision

4.3.2. Robust MLS Surfaces

MLS functional approximations can be derived directly from LKR [TFM07], and similarly MLS surfaces can also be derived as shown in [3]. This suggests that one can directly incorporate the robust weights into the minimization to arrive at a novel MLS surface definition. This is precisely what we do in [5]. The usefulness of robust statistics in the geometric setting of MLS surfaces has already been observed by several authors [FCOS05] [GG07] [DHOS07], but the current approaches have severe limitations and drawbacks. We leave an in-depth discussion of these methods and ours to section [5.1.2].
4. Robust Statistics for Local Kernel Regression
A New MLS Surface Definition

In this chapter, we derive a new MLS surface using our insight in the link between MLS and LKR, and robust statistics. We show that it outperforms other definitions in several aspects. In particular, the new definition can handle high frequency parts, sharp features and details naturally and elegantly, produces continuous surfaces, avoids artifacts resulting from shrinking or expanding, is very efficient, and only requires a simple modification to an already existing definition. Since it is in analytic form, one can derive derivatives such as curvature easily and thus can get exact differential quantities.

5.1. Foundations and Previous Work

5.1.1. Local Surface Fitting

We base our new MLS definition on Kolluri’s [Kol05] definition, which is directly inspired by the idea of Shen et al. [SOS04]. The main intuition of Shen et al. [SOS04] is, instead of using a plane, or higher order basis, one can just use a constant basis for the local function fitting and to avoid the trivial solution, the constraint that the implicit function is the constant zero at the samples can be replaced by a function. To clarify this idea, one can think of fitting a local surface using a weighted least squares approach as follows (please see [SOS04] for a discussion). Suppose that the function we want to fit has the form

\[ f(x) = \sum_{i=1}^{m} c_i d_i(x) \]  

(5.1)
where \(d_i(x)\)'s are basis functions. A simple example is a plane, which can be represented as
\[ f(x) = c_1 + c_2 x + c_3 y + c_4 z. \]
Local fitting of this function to a set of data \(\{x_i, y_i\}_{i=1}^n\) requires the solution of:

\[
\Phi^\frac{1}{2} D c = \Phi^\frac{1}{2} y \quad (5.2)
\]

\[
\Phi = \begin{bmatrix}
\phi(||x - x_1||) & 0 & 0 & \cdots \\
0 & \phi(||x - x_2||) & 0 & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \phi(||x - x_n||)
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
d_1(x_1) & d_2(x_1) & \cdots & d_m(x_1) \\
d_1(x_2) & d_2(x_2) & \cdots & d_m(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
d_1(x_n) & d_2(x_n) & \cdots & d_m(x_n)
\end{bmatrix}
\]

\[
c = [c_1 c_2 \cdots c_m]^T
\]

\[
y = [y_1 y_2 \cdots y_n]^T
\]

Once \(c(x)\) is found by solving this linear system, the function is simply \(f(x) = c(x)^T d(x)\), where \(d(x) = [d_1(x) d_2(x) \cdots d_m(x)]\). Depending on the type of the surface fitted, \(D\) can have different number of columns. Note that higher order surfaces will increase the number of basis functions and thus the number of columns, resulting in systems more time consuming to solve. The matrix \(\Phi\) contains the weights that depend on the evaluation point \(x\) such that the estimated function is a moving least squares fit.

For the case of surface fitting, \(y = 0\) and thus without any constraints only the trivial solution of \(f(x) = 0\) is obtained. The trivial solution is avoided using constraints as discussed in preceding sections, or creating new samples that are slightly off the surface \([OBA+03]\). It is observed that the latter approach has oscillatory behavior \([SOS04]\) and thus does not produce a stable function. Here, we focus on the method of Shen et al. \([SOS04]\) and Kolluri \([Kol05]\), which implies simply replacing \(y_i = 0\) with \(y_i = n_i^T (x - x_i)\). Although it is argued \([SOS04]\) that this method deviates from standard MLS surface fitting, we showed in [3.3.1] that it can still be interpreted in the framework of LKR.

The basis function used for the surface definition is just the constant, thus \([5.2]\) becomes

\[
[\phi(||x - x_1||)^\frac{1}{2} \phi(||x - x_2||)^\frac{1}{2} \cdots \phi(||x - x_n||)^\frac{1}{2}]^T c_1 = \Phi^\frac{1}{2} y \quad (5.3)
\]

And the solution of this system is
5.1. Foundations and Previous Work

\[ f(x) = c_1 = \sum \frac{y_i \phi(||x - x_i||)}{\sum \phi(||x - x_i||)} = \sum \frac{n_i^T(x - x_i) \phi(||x - x_i||)}{\sum \phi(||x - x_i||)} \] (5.4)

In the following sections, we base our new MLS surface on this definition. However, for the derivation we use the LKR interpretation of this surface definition and IRLS.

5.1.2. Previous Work on MLS Surfaces with Robust Statistics

It has been already observed both for point set surfaces [FCOS05] [GG07] [DHOS07], and meshes [JDD03] [CT05] [CC05] that for preserving sharp features, robust statistical techniques can be used. The basic intuition is that the points that belong to a smooth surface patch near a sharp feature can be treated as outliers for the other smooth patches around the feature. Fleishman et al. [FCOS05] proposed an approach based on this observation. Their method starts with a small subset of points in the neighborhood of the point \( x \) to be projected. This initial set is assumed to be outlier free, that is, all points in this small set come from the same smooth surface patch. An initial plane is fit to this set and this plane is used as the parameter domain for fitting a polynomial. Then a new point is added to this set iteratively. The point with the lowest residual is added at each iteration. The iteration stops when the residual of the point to be added next is bigger than a threshold. After stopping the iterations, the points added to the set so far are discarded from the set of points in the neighborhood. The same algorithm is executed on the remaining points until all smooth patches are identified. Once smooth patches are identified, the point \( x \) is projected onto each of them, and CSG rules are applied to get the final projection. Whether the neighborhood of \( x \) contains more than one smooth surface patch is determined by fitting a polynomial to the neighborhood. If the maximum residual is larger than a threshold, it is decided that the neighborhood contains a sharp feature.

This way of using robust statistics has several drawbacks. It is well observed [GG07] that polynomial fitting over a reference domain defined by the least squares plane is not stable. Also, the polynomial fitting to determine if there are multiple smooth patches in the neighborhood can make wrong classifications, and the resulting segmentation and projection create non-smooth features and jagged edges [DHOS07]. Furthermore, the CSG operations used in the projection becomes too complex once the degree of the corners (that is, the number of smooth patches intersecting at the corner) increases [FCOS05]. Since the method depends on a segmentation step, high frequency features that do not have more than one incident smooth surface patch, such as peaks, are not preserved. The sharpness is not controllable. Due to the expensive segmentation, projection is an order slower [FCOS05]. Finally, the resulting surface is defined by a projection procedure, not an implicit function, and thus is not convenient for further operators such as differentials. Although the extension proposed by Guennebaud et al. [GG07] handles peak discontinuities and the sharpness is controllable, special handling and tagging is necessary. Another extension [DHOS07] produces smooth feature lines at the cost of a more complicated procedure.

In contrast to these approaches, high frequency detail preservation is integrated into the implicit definition of our surface without any special handling. It not only reconstructs sharp features, but any details present in the geometry. In addition, it does not have many artifacts other defi-
5. A New MLS Surface Definition

![Figure 5.1.](image)

(a) Normals near a sharp feature that belong to different smooth surface patches clearly fall into distinct clusters (b) on the unit sphere. Thus normals of the first plane are outliers for the normals of the second plane and vice-versa.

Our Approach

In this section, we explain and state our new MLS surface definition and also derive formulas for gradient and curvatures.

5.2.1. Our MLS Surface Definition

Using robust statistics and treating points that belong to different smooth surface patches near a sharp feature as outliers is a promising idea. However, using complex and inefficient algorithms heuristically hinders the usefulness of the approaches so far. In contrast, we use the link derived between RLKR and MLS surfaces to arrive at a theoretically sound and effective implicit MLS surface definition.

One key observation is, the normals, not the points themselves, are the actual outliers. This has been used heuristically in many feature preserving mesh smoothing algorithms [FDCO03] [CC05] [HQZ08] [JDD03] and partition of unity implicits [OBA+03] for sharp feature preservation. We illustrate this fact in Figure 5.1. Normals belonging to different smooth surface patches are clearly far away from each other than to normals of the same patch. This is a direct result of the definition of a sharp feature, where the normals deviate from each other most.

Using the theoretical framework of LKR and IRLS and treating normals as outliers, we formulate a new minimization similar to the derivations in section 3.3. IRLS results in a weighted linear system to solve at each iteration. We use this formulation and minimize a weighted least squares sum at each iteration. Following the observation that outliers lie in the normal space, we define the residuals in terms of the deviation of the gradient from the normals.

Formally, for the IRLS setting, and with Kolluri’s definition [Kol05] (also see 5.3.1), one has to solve the minimization
5.2. Our Approach

\[
\min_{s_k} \sum (s_k + a_i^T n_i)^2 \phi_s(||x - x_i||) \phi_r(r_{i,k-1}) \quad (5.5)
\]

at each iteration. Here \(r_{i,k-1}\)'s are the residuals, \(s_k\) is the estimated function in the \(k^{th}\) iteration, and \(\phi_s\) and \(\phi_r\) are used to indicate different widths of the kernels. If we look at the equation minimized, the residuals are actually

\[
s_{k-1} = s_k - n_i^T(x - x_i).
\]

Since \(s_k\) is the fitted function, we write

\[
f_k = s_k
\]

and the solution becomes the following iteration

\[
f_k(x) = \frac{\sum n_i^T(x - x_i) \phi_s(||x - x_i||) \phi_r(f_{k-1}(x) - n_i^T(x - x_i))}{\sum \phi_s(||x - x_i||) \phi_r(f_{k-1}(x) - n_i^T(x - x_i))} \quad (5.6)
\]

Here we get that the influence of a point depends on the weights \(\phi_r(f_{k-1}(x) - n_i^T(x - x_i))\). Although there is also normal information in this weight, we choose to directly and only use the normal information in the weights, which means simply replacing this weight with \(\phi_r(||\nabla f_{k-1}(x) - n_i||)\). Thus the resulting equation for our novel surface definition becomes

\[
f_k(x) = \frac{\sum n_i^T(x - x_i) \phi_s(||x - x_i||) \phi_r(||\nabla f_{k-1}(x) - n_i||)}{\sum \phi_s(||x - x_i||) \phi_r(||\nabla f_{k-1}(x) - n_i||)} \quad (5.7)
\]

For the kernel function \(\phi_s\), we use a Gaussian of constant width such that \(\phi_s(||x - x_i||) = e^{-||x - x_i||^2/\sigma^2}\) and \(\phi_r(||\nabla f_{k-1}(x) - n_i||) = e^{-||\nabla f_{k-1}(x) - n_i||^2/\sigma^2}\). Although data-dependent or non-radially decaying kernels could be used, we selected this simple scheme to see how our definition performs without any other extensions.

5.2.2. Derivatives

One of the advantages of our new MLS surface definition is that it is in implicit form. We exploit this fact in this section to compute the gradient and Weingarten Map of the surface. Although deriving exact derivatives is possible, we choose a much simpler approach that generates approximate but accurate derivatives of the surface. Since computing the new MLS surface requires an iterative procedure, deriving exact derivatives needs considering the function at each iteration. However, if one considers the final converged implicit function \(f_k(x)\) of the surface, the gradient \(\nabla f_{k-1}\) at the previous iteration can be regarded as a constant that determines the weights.

\[
f_k(x) = \frac{\sum n_i^T(x - x_i) \phi_s(||x - x_i||) \phi_r(||n_i - \nabla f_{k-1}||)}{\sum \phi_s(||x - x_i||) \phi_r(||n_i - \nabla f_{k-1}||)}
\]

Thus, fixing the weights \(\phi_r\), one can take the derivative of \(f(x) = f_k(x)\) easily. We derive the first and second order derivatives in the next sections.

For brevity, we will use \(d_i = n_i^T(x - x_i)\), \(w_i = \phi_s(||x - x_i||)\phi_r(||n_i - \nabla f_{k-1}||)\).
5. A New MLS Surface Definition

Gradient

Starting with the equation

\[ f = \sum \frac{d_i w_i}{w_i} \]  

(5.8)

If we take the gradient with respect to \( x \) we get

\[ \nabla f = \nabla \sum \frac{d_i w_i}{w_i} \]  

(5.9)

\[ = \left( \sum \nabla d_i w_i + d_i \nabla w_i \right) \frac{1}{\sum w_i} + \frac{1}{\sum w_i} \sum d_i w_i \]

\[ = \left( \sum n_i w_i + d_i \nabla w_i \right) \frac{1}{\sum w_i} - \frac{\sum \nabla w_i}{(\sum w_i)^2} \sum d_i w_i \]

\[ = \frac{\sum n_i w_i}{\sum w_i} + \frac{\sum d_i \nabla w_i}{\sum w_i} - f \frac{\sum \nabla w_i}{\sum w_i} \]

\[ = n + \frac{\sum d_i \nabla w_i}{\sum w_i} - f \frac{\sum \nabla w_i}{\sum w_i} \]

Weingarten Map

Estimating the curvatures or fundamental forms using discretization is a well-studied problem in mesh based geometry processing (see for ex. [Rus04]). There is also some work done in surfaces defined by point sets. For example, Yang et al. [YQ07] derives the curvatures of an MLS surface defined by minimizing an energy function, using the curvature formulas derived by Goldman [Gol05]. Since they use a standard MLS surface definition, the curvatures at sharp features are not accurate [YQ07]. Also, the resulting formulas are not simple such that they can be used for further analytic analysis and derivation.

Instead of deriving formulas for the curvatures, we explicitly derive the Weingarten Map so that curvature as well as principle directions can be calculated from the eigen decomposition of this matrix. We first derive the Hessian matrix since the Weingarten Map can be calculated by using a simple formula (see 2.1.4) involving the Hessian and the gradient. To compute the Hessian, we need the second derivatives. We denote the derivatives by \( f_{jk} \) and \( w_{i,jk} \), \( 1 \geq j, k \geq 3 \) and the \( j^{th} \) component of \( n_i \) by \( n_{ij} = (n_i)_j \) (for example, \( f_{12} = \frac{\partial^2 f}{\partial x \partial y} \) and \( w_{i,12} = \frac{\partial^2 w_i}{\partial x \partial y} \)). Using the gradient computed and the same notation as in the previous section, it can be shown that the derivatives are

\[ f_{jk} = \sum \frac{w_{i,k}}{w_i} \left( -\frac{n_{ij} w_{i}}{w_i} - \frac{d_i w_{i,j}}{w_i} + f \frac{w_{i,j}}{w_i} \right) - f \frac{w_{i,jk}}{w_i} \]  

(5.10)

\[ + \frac{\sum n_{ij} w_i}{w_i} + \frac{\sum n_{ik} w_{i,j}}{w_i} + \frac{\sum d_i w_{i,jk}}{w_i} - f_k \frac{w_{i,j}}{w_i} \]
5.3. Properties and Results

Our novel surface definition preserves geometric details and sharp features, while avoiding shrinking and expanding artifacts. It is also stable under low sampling, and the number of iterations for it to converge is effectively one. Thus we get fine geometric details for a very little cost. Since the resulting surface definition has an analytic form, it allows computing derivatives. We present these properties using a variety of examples in this section. For brevity, we refer to the surface definitions by using the first letter of the surnames. We refer to our novel definition as ÖPSS (Öztireli’s Point Set Surface), Adamson et al.’s [AA03] definition as APSS, Kolluri’s [Kol05] definition as KPSS, and Guennebaud et al.’s [GG07] definition as GPSS.

5.3.1. Sharp Features

Sharp feature preservation is a very important problem of point set surfaces. As discussed in 5.1.2, the current approaches has several limitations and drawbacks. We illustrate the sharp feature preservation property of ÖPSS in Figures 5.2 and 5.3. In contrast to previous approaches,
5. A New MLS Surface Definition

Figure 5.3.: Sharp feature preservation for difficult cases. Left images show reconstruction with GPSS, and right ones are with our new definition, ÖPSS. (a) A high order corner where 4 smooth faces meet is also preserved, which is very difficult with the existing methods. (b) Peak discontinuities such as the tip of this cone is preserved without any special handling or processing.

all sharp features including peaks and high order corners are preserved with ÖPSS without any special handling or processing, and the generated feature lines are smooth. This is naturally integrated into the analytic implicit definition of the surface.

5.3.2. Detail Preservation

Some details are either smoothed out or distorted with the existing definitions. We found out that ÖPSS works very well in preserving these details. A good example of this behavior can be observed in Figure 5.4. Shape of many parts such as the hands and chest are distorted by other definitions. In contrast, ÖPSS successfully recovers the shape of these parts, while smoothing out lower scale details and noise. Sharp features are preserved as can be seen at the corners and edges of the stand, and the tip at the head. The expanding/shrinking artifact at the tip of the skirt is also prevented.

Further examples of detail preservation can be seen in Figures 5.5, 5.7, 5.9 and 5.12 Figure.
5.3. Properties and Results

Figure 5.4.: (a) The original model. (b)-(e) Reconstruction with KPSS, APSS, GPSS, and ÖPSS. (f)-(h) Details of the reconstruction with ÖPSS.

ÖPSS is a typical example of artifacts introduced by existing definitions. The circular shape of the surface part in the middle is distorted severely with GPSS while ÖPSS almost perfectly recovers it.
5. A New MLS Surface Definition

Figure 5.5.: Reconstruction with (a) GPSS and (b) ÖPSS. The distortion and expanding artifacts are reduced significantly.

Figure 5.6.: Sharp features under sparse sampling. A controllable sharpness is possible with this sparsely sampled cube.

5.3.3. Sampling

Sampling is a fundamental factor affecting the quality of the reconstructions. Thus we performed experiments under very low sampling conditions as illustrated in Figures 5.6 and 5.7. ÖPSS preserves sharp features under very sparse samplings as illustrated as an extreme case in Figure 5.6. This cube has only four points at each face and still a controllable sharpness is obtained. In this figure, $\sigma_s$ is big enough to cover all samples, hence large $\sigma_s$ values can be used to provide stability under sparse sampling.

The stability and high frequency detail preservation of ÖPSS is also illustrated in Figure 5.7 (a)-(e). The original model is subsampled to contain approximately 4% of its samples and the resulting model is reconstructed with several MLS definitions. ÖPSS preserves fine details and sharp features on the body while providing a more stable and artifact free overall shape. Note the similarity of the reconstruction from the subsampled model to the actual model.
5.3. Properties and Results

Figure 5.7.: Comparison of different definitions and our definition under very low sampling conditions. (a) Original model with 12K points. (b) Subsampled model with 0.5K points. (c) Reconstruction with APSS, (d) GPSS, and (e) ÖPSS. (f) Plot of mean curvature using ÖPSS. (g) Number of iterations for ÖPSS to converge at each point and (h) its histogram.
5. A New MLS Surface Definition

Figure 5.8.: Top row: Tuning the parameter $\sigma_r$ to obtain different sharpness for the features (from left to right: $\sigma_r = \infty$, 1, 0.5, 0.33). Bottom row: The sampling of the initial model, and the number of iterations for ÖPSS to converge for the corresponding $\sigma_r$ in the top row. Note that the case $\sigma_r = \infty$ corresponds to KPSS.

5.3.4. Curvatures

We illustrate the stability and accuracy of the computed curvatures using ÖPSS in Figures 5.7 (f) and 5.11. Since ÖPSS preserves features, and curvatures are directly computed from this surface definition, the accuracy of curvatures does not degrade around sharp features and details.

5.3.5. Controllable Sharpness

One can tune the sharpness of the features using the parameter $\sigma_r$. Figures 5.6 and 5.8 are examples of how different values of $\sigma_r$ affect the sharpness of the features. For smaller $\sigma_r$, the sharp features are more pronounced. Setting $\sigma_r$ to very low values makes the reconstruction have sharper features than the actual model.
5.3. Properties and Results

Figure 5.9.: Reconstructions of the isidore horse and a Greek sculpture model with GPSS and ÖPSS. Note that at the boundaries of the stand under the Greek sculpture model, the reconstruction for both definitions is bad. This is expected since ÖPSS is not designed to handle boundaries.

5.3.6. Performance

To evaluate performance of the definition, we investigated convergence rate of the iterative procedure. To visualize the results on the models, we plot number of iterations for ÖPSS to converge at the projected points on the surface in Figures 5.7 (g) and (h), 5.8 and 5.10. Number of iterations increases on and near the sharp features or high frequency details. This means that whenever a surface patch is smooth, ÖPSS coincides with KPSS. But when there are high frequency details, ÖPSS improves the surface at those patches. Thus extra work is done only when needed. Since most patches of the shapes are smooth, the big majority of the points need only one iteration to converge. Even on the sharp features, the number of iterations is practically at most five, showing the efficiency of the definition. Figure 5.8 also illustrates...
the effect of changing $\sigma_r$ on the number of iterations. It shows that as $\sigma_r$ is decreased, more points become near the high frequency features and thus the definition becomes less efficient. However, the number of iterations are still not more than five and points with high number of iterations are still a small minority, thus the difference in efficiency is quite small.
Figure 5.11.: Minimum, mean and maximum curvatures using ÖPSS are plotted on the model.
5. A New MLS Surface Definition

Figure 5.12.: Top row shows reconstruction with GPSS, and bottom is with ÖPSS. Note the details with ÖPSS reconstruction.
Outlook

The links we established between LKR and MLS surfaces suggest that we can use methods from kernel regression literature to achieve specific goals. In this thesis, we could only touch upon one of these directions: using robust statistics with MLS surfaces to preserve high frequency details. In this chapter, we review some promising future research directions and also possible extensions to our new MLS surface definition.

6.1. Convergence and Continuity

6.1.1. Convergence

Our novel MLS surface definition is based on an iterative minimization. Although we always observed a high convergence rate, an analytical derivation on the convergence guarantee and rate is yet to be constructed.

6.1.2. Continuity

Except for models with boundaries, the reconstructions should produce continuous surfaces. Our definition aims at reconstructing submanifolds without boundaries and thus the resulting surface should be continuous. We have not observed any discontinuities, except the ones due to numerical errors. However, a formal proof of continuity or smoothness is a future work.
6. Outlook

6.2. Anisotropic Setting

Originally MLS surface definitions were proposed to be used with uniform sampling. There has been several efforts to extend the reconstruction to the anisotropic setting either by a variable spatial kernel width [Pau03], or using non-radially decaying kernels [AA06]. In this thesis, we assumed uniform sampling and thus used radially decaying kernels with constant width. Extending the definition to handle non-uniform sampling is an important task for future.

Using the link we derived between LKR and MLS surfaces, we plan to use recently proposed techniques [WT07] of metric learning for kernel regression to make the kernel data-adaptive. So far, parameters of the kernels have been set manually. With metric learning, the selection of all can be done automatically.

6.2.1. Metric Learning

Metric Learning deals with learning a metric based on the data. In the case of kernel regression, this metric is the distance between two samples. To learn this metric, a gradient descent procedure can be applied to a distance model [WT07]. The parameters gathered can then be used to define the metric uniquely. The most obvious problem with this setting is that the parameters should be learned separately for each sample, which reduces the efficiency. But if the number of parameters to be learned is low, one can still get a practically efficient definition.

6.3. Higher Order LKR

In deriving ÖPSS, we used a zero order LKR with normal constraints. However, there are a variety of possibilities with the order and constraints that can be used for the definitions. LKR up to second order have proved its usefulness in image reconstruction [TFM07]. Thus it is natural to expect a similar behavior for the geometric setting. However, we found out that setting a constraint is not a trivial task and the stability of the surface definition highly depends on this choice. Thus we are investigating good constraints that would yield stable computations, smooth manifolds and tight surfaces.

6.3.1. Robust Normal Estimation

Using robust statistical techniques with higher order LKR can lead to robust normal estimation such that the normals are not smoothed near sharp features. Although there are heuristic feature preserving mesh smoothing methods based on normals that produce robust normals (e.g. [JDD03] [YOB02], also see section 4.3), we plan to explicitly use robust methods with a theoretical basis for the case of point sets.
6.4. Projection

Projection onto the MLS surface can be performed by the following equation

$$x_{k+1} = x_k - f(x_k)\nabla f(x_k)$$  \hspace{1cm} (6.1)

for an implicit function $f(x)$ whose zero set defines the surface. This means we seek the zeros of the function $f(x)$ by a gradient descent based method. Thus one can use many other proposed methods for root finding to obtain a better convergence rate.

Another way to look at this equation is to see it as an explicit Eulerian integration method. If one writes

$$x_{k+1} - x_k = -f(x_k)\nabla f(x_k)$$  \hspace{1cm} (6.2)

it is trivial to see that this is in fact an iteration of the solution of

$$x' = F(x)$$  \hspace{1cm} (6.3)

where the function $F$ absorbs all terms and necessary coefficients for the right hand side of (6.2). Thus other integration techniques may be utilized.

6.5. Kernel Selection

Although there are numerous analysis on the selection of kernels for specific tasks in the statistics literature, computer vision and graphics community uses a small subset of the proposed kernels. As pointed out in [TFM07], a through investigation of kernel selection have not been considered yet. It has been observed that the differentiability of the MLS surface is determined by that of the kernel, thus kernel functions of different degrees of smoothness, or a combination of them may be used to improve the surface quality.

6.6. Higher Order Derivatives

We showed that one can compute derivatives accurately with ÖPSS, and the resulting expressions are not intractably complex. Although we computed derivatives up to the second order, higher order ones can also be computed. This has numerous application such as ridge/valley detection [OBS04], feature extraction [LVJ05], or remeshing [ACSD03]. The expressions for the derivatives can also be utilized to get analytical or discrete solutions of various problems on the surfaces.
6. Outlook

6.7. Medial Axis

For a signed distance field $f(x)$, the medial axis is defined as the set of points such that $||\nabla f(x)|| = 0$. Thus, a gradient descent procedure can be used to project onto the medial axis as follows:

$$x_{k+1} = x_k - \nabla ||\nabla f(x_k)||$$  \hspace{1cm} (6.4)

One can compute

$$\frac{\partial ||\nabla f(x)||}{\partial x} = \frac{\partial}{\partial x} \sqrt{f_x^2 + f_y^2 + f_z^2}$$  \hspace{1cm} (6.5)

$$= \frac{1}{||\nabla f||} (f_x f_{xx} + f_y f_{yx} + f_z f_{zx})$$

The other two partial derivatives are computed similarly and we get the following expression:

$$\nabla ||\nabla f(x)|| = \frac{1}{||\nabla f||} H^T \nabla f$$  \hspace{1cm} (6.6)

The problem with this approach is that far from the surface (where $f(x) = 0$), the implicit function does not accurately represent the signed distance field. Thus the main challenge is designing implicit functions that accurately approximate the signed distance field near the medial axis.
Conclusions

In this thesis, we interpreted moving least squares surfaces as a special case of local kernel regression and used the derived links and robust statistics to arrive at a novel surface definition that preserves high frequency details. Extensive experiments show that the novel surface definition recovers sharp features, reduces common artifacts with the existing MLS surface definitions such as expanding and shrinking, retains fine geometric details, and produces stable surfaces even under low sampling conditions with very little extra computational cost. The definition is in implicit form and thus can be utilized to compute derivatives of the surface analytically. We also pointed out many further extensions and techniques that can be adopted from kernel regression literature to achieve more stable surfaces with desired properties.

We presented an example of the practical use of the link between MLS and LKR by using ideas from a well-known robust statistics technique called iteratively reweighted least squares and adapting it to the geometric setting. Using the observation that the actual outliers for sharp features and details are the normals and not the sample locations, we used weights depending on the gradient and sample normals for our iterative surface definition. Although it is defined iteratively, the experiments show that convergence rate is very high and at most around five iterations are performed for each evaluation. However, this does not mean that the definition works five times slower than a non-iterative one. Number of iterations increases only around the high frequency features and for smooth regions, they are almost always one. Thus no extra work is done if a region is smooth. Since high frequency regions constitute a minority in a shape, the number of iterations for the whole shape is effectively one.

One obvious drawback of the definition is that normals at the sample points need to be known or approximated in advance. Fortunately, they can be usually gathered during the acquisition process from real world, and even not so, there are several methods to approximate them. But still, a method that do not use them is obviously preferred. This is possible by incorporating the robust procedure into other definitions, which we propose as an important future work.
7. Conclusions

Deriving MLS surfaces from LKR framework has not been done before. The derivation is done for two different MLS surface definitions in this thesis, but we believe that a general framework can be deduced in the future that covers and unifies all similar surface definitions that use kernels. We also believe that this will lead to a much better understanding of which definitions are better for which conditions and thus a general surface definition selection procedure can be achieved.

The most serious problem we see with LKR based surfaces is setting the constraints. MLS surfaces can be formulated as local implicit surface fittings, thus setting constraints has also been a common problem for MLS surfaces. Since there is a bulk of work done in regression literature, we hope to find a general solution to this problem in that domain. The section about unsupervised kernel regression 3.1.1 is explaining a first attempt in this direction, where the constraints are explicitly avoided in the definitions. However, it is not clear how this may work in the geometric setting and the resulting surfaces are no longer implicit.

There are many other promising research directions using techniques from kernel regression literature. Improving surface quality through order of the expansion used in regression and kernel selection is an immediate promising direction. Kernel selection is a particularly more investigated topic in regression literature but the extension for the geometric setting is not trivial. Metric learning to learn the parameters of the kernels is another promising research direction. This can also be utilized to extend the definitions to handle anisotropically sampled point clouds. Another useful future work is computing derivatives using either kernel regression or directly from the analytical definitions. We presented an example by computing the Weingarten Map but computing further derivatives is also possible and tractable. We hope these can aid in various problems where closed form formulas for the derivatives are needed.
Local Kernel Regression

A.1. Deriving Adamson et al.’s [AA03] MLS surface definition from LKR

In this section we give the details of the derivation of Adamson et al.’s [AA03] MLS surface definition from LKR. For the discussion, results and some definitions, please see 3.3.2. We start with the minimization problem:

$$\min_{s_0, s_1} \sum (s_0 + a^T_i s_1)^2 \phi_i(x)$$  \hspace{1cm} (A.1)

with the constraint $||s_1|| = 1$. Let us define the function $m(s_0, s_1) = \sum (s_0 + a^T_i s_1)^2 \phi_i(x)$. The minimization can be solved by taking partial derivatives of $m$ with respect to $s_0$ and $s_1$ and setting the expressions to zero.

$$\frac{\partial}{\partial s_0} m(s_0, s_1) = 2 \sum (s_0 + a^T_i s_1) \phi_i(x) = 0$$

$$\Rightarrow s_0 = s_1^T \left( \sum x_i \phi_i(x) - x \sum \phi_i(x) \right) \over \sum \phi_i(x) = s_1^T (x - c) \hspace{1cm} (A.2)$$

Similarly, taking the derivative with respect to $s_1$ and setting to zero:
A. Local Kernel Regression

\[
\frac{\partial}{\partial \mathbf{s}_1} m(s_0, \mathbf{s}_1) = 2 \sum a_i(s_0 + a_i^T \mathbf{s}_1) \phi_i(\mathbf{x}) = 0
\]

\[\sum a_i \phi_i(\mathbf{x}) \mathbf{s}_1^T (\mathbf{x} - \mathbf{c}) + \sum a_i a_i^T \phi_i(\mathbf{x}) \mathbf{s}_1 = 0\]

\[
\frac{1}{\sum \phi_i(\mathbf{x})} \left( \sum a_i \phi_i(\mathbf{x}) \mathbf{s}_1^T (\mathbf{x} - \mathbf{c}) + \sum a_i a_i^T \phi_i(\mathbf{x}) \mathbf{s}_1 \right) = 0
\]

\[ (\mathbf{c} - \mathbf{x}) \mathbf{s}_1^T (\mathbf{x} - \mathbf{c}) + \frac{1}{\sum \phi_i(\mathbf{x})} \left( \sum (\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T \phi_i(\mathbf{x}) \right) \mathbf{s}_1 = 0 \]

\[-(\mathbf{x} - \mathbf{c})(\mathbf{x} - \mathbf{c})^T \mathbf{s}_1 + \left( \sum \frac{\mathbf{x}_i \phi_i(\mathbf{x})}{\sum \phi_i(\mathbf{x})} + \mathbf{x} \mathbf{x}^T - \mathbf{c} \mathbf{x}^T - \mathbf{x} \mathbf{c}^T \right) \mathbf{s}_1 = 0 \]

\[ \left( -\mathbf{x} \mathbf{x}^T - \mathbf{c} \mathbf{c}^T + \mathbf{c} \mathbf{x}^T + \mathbf{x} \mathbf{c}^T + \frac{\sum \mathbf{x}_i \phi_i(\mathbf{x})}{\sum \phi_i(\mathbf{x})} + \mathbf{x} \mathbf{x}^T - \mathbf{c} \mathbf{x}^T - \mathbf{x} \mathbf{c}^T \right) \mathbf{s}_1 = 0 \]

\[ (\mathbf{C} - \mathbf{c} \mathbf{c}^T) \mathbf{s}_1 = 0 \] (A.3)
Bibliography


[CC05] Chun-Yen Chen and Kuo-Young Cheng. A sharpness dependent filter for mesh


[KR07] Stefan Klanke and Helge Ritter. Variants of unsupervised kernel regression: Gen-


[Tau91] Gabriel Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range im-


