Geometry Refinement of 3D Surfaces Using Kriging

Brad Grinstead, Andreas Koschan, and Mongi A. Abidi
Imaging, Robotics, and Intelligent Systems Laboratory
The University of Tennessee
{bgrinste, akoschan, abidi} @utk.edu

Abstract
3D imaging is a popular method for acquiring accurate models for a variety of applications. However, the size of the geometric features that can be modeled in this manner is dependant on the scanning system’s resolution. This paper presents a method that attempts to accurately reconstruct regions whose features are at or below the system’s scanning resolution, combining automatic region selection with a form of kriging. A curvature-based segmentation is followed by an automated geometry refinement procedure in which the model of spatial correlation between the irregularly sampled 3D data is automatically determined. Geometry refinement is done by a regularized kriging approach that is designed to preserve the sharp features typical to many 3D laser range applications. This method is validated on synthetic data, showing that the accuracy of our method is higher than that of its standard competitors. Then, the performance on real data is demonstrated through several examples.

1. Introduction
3D imaging is becoming a widely used method for acquiring 3D surface data for a variety of applications. Laser scanning has been used to generate Digital Elevation Maps (DEMs) [3, 10, 13] from airborne systems, urban 3D models [4, 6, 9] from ground-based systems, 3D models of small objects and indoor mapping [7, 14, 15], and more. Often, however, the modeled objects contain regions of interest (ROIs) that have small-scale details – defined as areas whose geometric feature dimensions are much smaller than the overall size of the object – that are not digitized at a high enough resolution to be accurately represented for further analysis.

In order to enhance the details of a region in an acquired dataset, additional sampling needs to be applied to the region of interest. Re-acquiring the data at a higher resolution may not be an option for many applications, due to the time and costs involved in performing the acquisition again. Therefore software-based geometry refinement has a number of benefits, including: the ability to recover details that are near, or below, the low end of the scanning system’s resolution from an existing dataset; the ability to set up the acquisition to acquire at a lower resolution, based on the assumption that the refinement method will recover the details of interest from a lower resolution scan, yielding faster acquisition and smaller raw data files.

As an example of this type of data, Figure 1 shows a laser range dataset that was acquired at 5cm/profile resolution. This resolution is not fine enough to accurately represent the ideal (synthetic) 3D geometry seen in the bottom image of Figure 1.

Figure 1. Example data showing a real-world building façade, a laser range scan of the same, the 3D surface generated from the scanned data, and a synthetic representation of the same model.

There are a number of applications that can benefit from a method that reconstructs small-scale
features that may not have been sampled at a high enough resolution to appear clearly. In the field of robotic manipulation, some objects scanned into an a priori model that the robot needs to access may have been overlooked in the scanning process. Algorithmic geometry refinement could remove the need to reacquire that data. Geometry enhancement can also ease the task of model verification through laser scanning, crack detection from a mobile platform, or even modeling and verifying tire/soil interaction for automotive simulation and design. In fact, many 3D modeling applications that have regions of important details that are near, or below, the low end of the scanning system’s resolution and are much smaller than the overall dataset’s dimensions can benefit from this method.

In the majority of these applications, critical information is contained at the edges of the 3D structure. Preservation of these edges, or sharp features, is therefore the priority when implementing any geometry refinement method. Classical approaches for resampling range data are linear and cubic interpolation [1], and Inverse Distance Weighting (IDW). Statistical methods such as kriging have also been applied to 3D datasets [12, 18]. Despite the fact that 3D data tends to be sampled at very small separation distances, the uncertainty of resampling using these methods tends towards oversmoothing or deformation of the sharp features present in the data.

We propose a modified version of kriging that uses the spatial relationships present within the data to provide estimates of the underlying surface during the geometry refinement process, with a regularization component tailored towards preserving the edge information also present in the data. The geometry refinement process is fully automatic, from the selection of the ROIs to the final estimation of the underlying surface. We demonstrate the efficacy of this method on a synthetic dataset, as well as on data acquired from an existing mobile laser scanning system, such as the one used in [6]. The results show that our modified kriging method outperforms its common competitors in terms of accuracy and preservation of features.

The rest of the paper is presented as follows: Section 2 describes the fundamentals of geometry refinement, and the theory behind kriging in particular; Section 3 describes the modifications that we have made to automate the kriging process and tailor the output towards preserving edge information; Section 4 describes the experimental process used to validate this method, its comparison to existing interpolation methods, and its application to actual 3D laser scanning data; Section 5 concludes the paper with a summary of the work presented, a discussion of the results of our efforts, and a brief look at what the future holds.

2. Geometry refinement methods

Spatial interpolation has been defined as the procedure of estimating the value of a field variable at unsampled locations within the area covered by sample locations [17]. In the case of resampling 3D data from a laser scanning system, the process is one of geometry refinement based on the non-uniformly sampled surface points. The underlying assumption of any interpolation algorithm is that samples close to each other will likely be more similar than points that are further apart.

Linear and cubic interpolation on a non-uniformly sampled field variable is typically done by performing a Delaunay triangulation to optimally connect the sampled point into a network of triangles. Interpolation is then done by searching the network for the closest triangle and computing the linear or cubic interpolant from that triangle. This method of interpolation is typically quite fast, but is very dependant on the uniformity of the sampling of the data.

Inverse distance weighting is a weighted average interpolator, where the influence of a measured point on the new estimate decreases as the distance from to the location of the estimated point increases. The weighting power $p$ controls how the influence tapers off with respect to distance $d$

$$Z_{e} = \frac{\sum w_i Z_i}{\sum w_i},$$

where $w_i$ is the weight associated with the measured value $Z_i$, and $Z_e$ is the estimation of the new value. Frequently the power used for performing the interpolation is attached to the IDW abbreviation, thus inverse distance weighting using a power of 2 is designated as IDW2.

In its native state, IDW uses all measured data points in the weighting process, but more frequently only the points within a user-specified radius are used. One of the characteristic effects of IDW is that it tends to generate a “bulls-eye” effect around measured points [16]. Smoothing parameters are sometimes introduced to alleviate this effect.

Kriging is a spatial interpolation method that gives estimates of new surface values based on a statistical model of the spatial correlation between

$$Z_{e} = \frac{\sum w_i Z_i}{\sum w_i},$$
measured values [2]. Kriging estimates the values at specified locations, using observed data to drive the process, optimized with respect to specific error criteria. This error criteria is the squared prediction error at the unobserved locations. The measured data provide the support for the estimated values, and the quality, size, shape, and orientation of the observed values influence the ability to accurately predict the unknown sample values. Because kriging is capable of modeling the data both as a trend and as a set of residuals, it is often considered to provide the most accurate surface value predictions of all other interpolation types under the right circumstances.

The estimation of the spatial correlation of surface data in kriging theory is typically done with the use of variograms. The variogram is a function which characterizes the dependence of data points measured in a region of interest. For every sampling interval \( d \), the semivariance for distances equal to multiples of \( d \) can be computed as

\[
\gamma(d) = \frac{1}{2N(d)} \sum_{i=1}^{N(d)} (z_i - z_{i+h}),
\]

where \( \gamma(d) \) is the semivariance at lag \( d \), \( z_i \) is the measurement of a regionalized variable taken at location \( i \), \( z_{i+h} \) is another measurement taken \( h \) intervals away, \( N(d) \) is the number of separating distances: number of points - number of lag intervals.

Once the experimental variogram has been calculated, an ideal parametric model is fit to the data. The ideal parametric model is used to simplify the estimation process and to increase the robustness to measurement errors.

Kriging is the actual process of using the parametric variogram model to estimate the surface value at the specified location. The most common form of kriging used in engineering applications is ordinary kriging (OK).

In ordinary kriging, the estimate of an unknown surface value uses a weighted estimate of other nearby points:

\[
Z_e(p) = \sum w_i Z(p_i).
\]

Ideally, kriging attempts to minimize the error between the estimated point and the actual value. The variance of this error is the amount of scattering of the estimates \( Z_e \) about their true values \( Z_o \)

\[
\sigma_z^2 = \frac{\sum_{i=1}^{n} [Z_e(p_i) - Z_o(p_i)]^2}{n}.
\]

The estimation and its error are dependent on the weights chosen in (3). Optimal weights, therefore, would be those that produce the minimum estimation variance. These are found by solving a system of equations consisting of the weighted semivariances between measured points, and the estimated semivariances between the unknown point and the known values

\[
\begin{bmatrix}
\gamma(d_{i1}) & \ldots & \gamma(d_{i,p}) \\
\vdots & \ddots & \vdots \\
\gamma(d_{n1}) & \ldots & \gamma(d_{np})
\end{bmatrix} \begin{bmatrix}
w_1 \\
\vdots \\
w_p
\end{bmatrix} = \begin{bmatrix}
\gamma(d_{1p}) \\
\vdots \\
\gamma(d_{np})
\end{bmatrix},
\]

where \( \gamma(d_o) \) is the semivariance between the \( i^{th} \) and \( j^{th} \) measured values and \( \gamma(d_{ij}) \) is the estimated semivariance between the \( i^{th} \) observation and the location of the evaluation point.

3. Our improvements

There are a number of issues with standard kriging that cause it to be less than ideal for geometry refinement in the presence of small-scale details. Firstly, the global optimization of the kriging process tends towards smooth surfaces that obscure, rather than enhance, the small-scale details present in the data. Secondly, kriging is often done with a number of steps involving manual parameter selection and initialization. The size of the training and estimation regions also poses a problem. Using the entire dataset for training is not beneficial when the refinement is going to be performed only on small ROIs. Finally, the manner in which kriging itself is performed lends to a robust solution, but an oversmoothed one. The rest of this paper discusses our solutions to these issues, followed by experiments that verify our solutions.

The regions selected for enhancement in this scenario are often much smaller than the dimensions of the entire collected dataset. They also typically stand out from the rest of the data by being different – in terms of shape, curvature, spatial frequency – than the surrounding areas. These regions can typically be picked out by a human observer rather easily. In order to automate this process, we use curvature information, calculated from the acquired data, to identify areas of uncommon curvature that are then considered our ROIs. The method that we have chosen to use is similar to that shown in [11].

The ROIs are generated by randomly selecting seed triangles from the surface, performing region growing around those seeds, and assimilating neighboring patches with similar curvatures. Finally, connected regions with fewer than a pre-specified number of triangles are rejected. What remains is a set of surface patches with a relatively constant curvature. The selection is then inverted to provide those regions.
that do not have a continuous curvature, and are thus our regions of interest. Figure 2 shows a visual demonstration of the procedure. The upper left image shows the seed triangles highlighted in blue. Shown in the upper right is the segmentation result after several iterations of region growing and merging. The lower image shows the final segmentation results after culling, with the “smooth” regions highlighted in blue.

The ROIs are not only used to define the areas that need refinement, they are also used as the training data for the kriging process. By restricting the training data to lie only in the ROIs, we get an improvement in surface fit due to the removal of the influence of the data that is “not of interest”.

Typically, variograms are generated by manually selecting a model from a library of functions, and then using a least-squares method or manual parameter selection to fit the model to the estimated semivariogram. In order to automate this process, we perform a least-squares fitting of a number of ideal models, and the one with the best match to the data is used as the ideal variogram model for the kriging process. The least squares fitting is constrained to closely fit the model to those lags close to the source, since we are interested in the best local fit of the surface data. This process allows us to automatically select the appropriate model for the task, as well as providing a closer estimation of the underlying model near the estimation location.

In experimental systems, it is not possible to sample data at one location over and over without incurring some variation in the samples. This variation is due to system noise, environmental shifts, and other factors. Traditional kriging treats this micro-scale variation in a global fashion, adding a “nugget” effect to the variogram at $d=0$. However, this is inappropriate when it is known that the measurement variation in every region is likely to be different. Instead, we employ a regularization procedure that attempts to model the micro-scale variations independently. This regularized approach accounts for local measurement errors and helps to reduce the smoothing effect that the nugget parameter introduced.

Given that Equation (5) can be written as $Ax = b$, we can solve the system using Ordinary Least Squares

$$x = (A^T A)^{-1} A^T b.$$  \hspace{1cm} (6)

This is equivalent to zero-order Tikhonov regularization. Since $(X^T X)^{-1}$ is ill-conditioned, we introduce a regularization parameter $\lambda$ to increase stability of the system

$$x = (A^T A + \lambda I)^{-1} A^T b.$$  \hspace{1cm} (7)

Here $\lambda$ can be chosen to be either a scalar or a vector. It can be automatically determined through validation methods such as autocorrelation, or empirically chosen through experimentation. We suggest choosing a $\lambda$ that reflects the amount of uncertainty present in the data acquisition system. Our $\lambda$ was chosen using the L-curve analysis technique of [8].

4. Experimental results

In order to show that the augmentations made to the original kriging algorithm actually yield a better geometry refinement, a number of experiments were performed on synthetic datasets. Figure 3 shows a 2m x 2m synthetic model developed for tire/soil interaction. The model is an impression of a repeated tire tread pattern embedded into a varying terrain surface. This model contains regions of high detail, as well as smooth regions, and is useful in demonstrating the effects of a laser scanning system on such data, as well as providing a ground truth model for comparison of geometry refinement results.

A mobile laser scanning simulator was developed to synthetically recreate the process of using a laser range scanner to digitize an arbitrary surface. This simulator allows the user to specify parameters that control a virtual laser scanning system. These parameters include the scanning resolution along a laser profile, the resolution between profiles, the orientation of the “scanner” at each sampling stage, and the noise parameters of the system. The simulator then reads in a triangulated 3D model, performs the “scanning” using the specified parameters, and the
The output of this system was then used as the input data for 4 different interpolators: linear, cubic, IDW, and our modified kriging.

The input to the test algorithms was a synthetic range scan (Figure 3) with a vertical resolution of 4 mm and a horizontal resolution of 2 cm. This non-uniform sampling is consistent with many laser scanning systems, and is the cause of the majority of the undersampling problems of interest in this paper. Each of the 4 geometry refinement algorithms was then used to reconstruct the surface at a resolution of 5 mm in both the horizontal and vertical directions.

Figure 4 shows the results of the interpolation methods applied to the synthetic range scan generated from our simulator. For each of the interpolated models, we calculated the distance of each point to the reference model and color coded the surface according to its distance to the reference model. Surface points close to the ground truth are shown in blue moving through the spectrum to green and finally to red as the deviation grows large. It can be seen that our modified kriging algorithm has fewer areas of red, and more of blue and green, than the other methods, indicating that the interpolated surface follows closer to our ground truth model.

Figure 5 shows the distributions of the signed distance from the interpolated surfaces to the reference model. The tighter the distribution is centered at 0, the better it represents the underlying surface. From this graph it can be seen that our modified kriging holds the closest to the original surface, with the linear and cubic interpolation right behind. The poor results from IDW2 come from the aforementioned “bullseye” effects due to the irregularly sampled nature of our data. More global surface deviation results are given in Table 1. The fields in Table 1 list the minimum, maximum, mean, and median deviation (error) of the refined surface from the ground truth model, the variance of the errors, and the Root Mean Square error for each of the refinement methods tested. Note that our kriging algorithm outperforms the other methods across the board.

Figure 3. Geometry refinement validation using a synthetic ground truth model.

Figure 4. Results of geometry refinement methods applied to test dataset. Variations from the reference model are color coded. The lower images for each column are zoomed in photos of the upper right hand corners of the reconstructions.

Figure 5. Distribution of signed distances from the refined surfaces to the reference model.
Having shown that our modified method has the ability to outperform its competitors, we turn to the discussion of training regions and their effect on the refinement of our areas of interest. Figure 6 shows six regions selected from the ideal model to be used in this test, three each from the detailed and non-detailed regions. Semivariograms were trained on each of these regions independently, and the model was reconstructed at a resolution of 5 mm.

Figure 7 shows a graph relating the median distance to the reference model by surface patch. Each of the reconstructions is compared to the reference model as a whole, as well as to each of the individual training regions, in order to evaluate global vs. local performance. It can be seen from the graph that the reconstructions from the training regions in detailed areas have the best performance within the detailed regions, while not giving up any performance overall. This supports our claim that restricting the training set to only those regions that contain the small-scale details we are interested in enhancing is beneficial.

Figure 8 gives a visual representation of the location and magnitude of the errors in the surface reconstructed from region Detail3. The larger the dot, the larger the error in the reconstructed surface. Surface elevation contours have been overlain on the image for context. From this figure, it can be seen that the highest magnitude of errors occurs, unsurprisingly, in those areas that have the largest changes in curvature. Also, the largest errors occur in a vertical striping pattern that lies along the scan lines from the input data.

Having shown the validity of our algorithm on a synthetic dataset, the method was used to enhance data taken from real laser scanning systems. These datasets were acquired from mobile scanning platforms undergoing general motion. The sampling resolution is determined by the equipment specifications as well as the motion the platform is undergoing.

**Table 1. Quantitative comparison of interpolation results**

<table>
<thead>
<tr>
<th>Method</th>
<th>Min (mm)</th>
<th>Max (mm)</th>
<th>Mean (mm)</th>
<th>Median (mm)</th>
<th>Variance</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>9.59 e-5</td>
<td>25.95</td>
<td>2.52</td>
<td>1.52</td>
<td>1.01 e-2</td>
<td>4.04</td>
</tr>
<tr>
<td>Cubic</td>
<td>9.61 e-5</td>
<td>25.80</td>
<td>2.58</td>
<td>1.63</td>
<td>8.41 e-3</td>
<td>3.84</td>
</tr>
<tr>
<td>IDW2</td>
<td>13.42 e-3</td>
<td>44.83</td>
<td>9.84</td>
<td>8.19</td>
<td>5.76 e-2</td>
<td>12.43</td>
</tr>
<tr>
<td>Modified Kriging</td>
<td>6.19 e-4</td>
<td>25.18</td>
<td>2.50</td>
<td>1.17</td>
<td>7.59 e-3</td>
<td>3.78</td>
</tr>
</tbody>
</table>

![Figure 6. Six regions chosen to test the effect of training regions on the kriging refinement. The lower image shows the estimated variograms from each region.](image6.png)

![Figure 7. Median distance to each specified region of interest, for each of the training sets used.](image7.png)
Figure 8. Spatial location of the errors from the Detail3 interpolation, with the height contours of the data overlain for context. The larger the dot, the larger the error.

Figure 9 shows an image of a commercial building, as well as the raw data taken from a scanning platform that moved past the building in a semi-parallel fashion. Notice that the letters on the sign are undersampled. This is due to the scanning vehicle’s movement - approximately 5 cm per scanned profile. The bottom image in Figure 9 shows the refined surface output from our method. The structure on the lettering is much clearer than in the original data, showing the enhancement. Both surfaces are shown in the same 3D viewer with the same display options.

The dataset in Figure 10 is from the same mobile scanning platform. Notice in the visual image how there is a regular grid structure present on the building that was undersampled by the scanning system. The vehicle’s motion corresponded to a sampling rate of approximately 10 cm per scanned profile. However, the few samples that are present are enough to partially reconstruct the grid but that the vertical struts are incomplete and insubstantial as compared to the horizontal struts, as seen in the lower image of Figure 10. This example shows that some minimum of sampling is required to be able to recover small-scale geometry from actual scanned data.

Figure 10. Experimental results, showing a digital image of the scanned surface (top), the original data acquired from the scanning system (middle), and the refined surface (bottom) showing the partial reconstruction of the vertical grid.

5. Discussion

Work has been presented on an automatic method for identifying and refining the surface geometry of small-scale details present in 3D laser scanning data. The method was validated on synthetic datasets, showing that the quality of the refined geometry was greater than that of both the original data as well as that from other interpolation methods. The method was then applied to real-world data, showing the type of enhancements that can be performed using this algorithm.
It should be noted that kriging, in general, is a computationally intensive method. Thus, the execution times for our geometry enhancement algorithm are larger than those from the competing methods we tested against. Thus, while we showed an improvement in the approximation of the underlying surface, it came at the cost of long execution times. The decision on whether the extra computation is worthwhile for the added benefit of improved surface accuracy will have to be weighed carefully, and will be application specific.

Future efforts for this research will likely be devoted towards continued automation and improvements in the accuracy. Among other topics, investigations involving nonparametric variogram estimation such as that shown in [5] are currently underway. Another topic in need of investigation is the minimal sampling required to reconstruct details of a given size. An in-depth look at the minimal sampling requirements for feature reconstruction would be beneficial for future tasks in geometry refinement. In addition to the algorithmic investigations, data from applications such as cultural heritage preservation and tire/soil interaction are being examined for benefit from this method.

Acknowledgements

This work was supported by the DOE URPR under grant DOE-DEFG02-86NE37968 and the DOD/RDECOM/NAC/ARC Program under grant W56HZV-04-2-0001.

References