Parameterization of unorganized cylindrical point clouds for least squares B-spline surface fitting

Seonghyeon Moon
School of Mechanical Engineering
Gwangju Institute of Science and Technology
61005, Gwangju, South Korea
moonsh@gist.ac.kr

Jin-Eon Park
School of Mechanical Engineering
Gwangju Institute of Science and Technology
61005, Gwangju, South Korea
sigma@gist.ac.kr

Kwanghee Ko
School of Mechanical Engineering
Gwangju Institute of Science and Technology
61005, Gwangju, South Korea
khko@gist.ac.kr

ABSTRACT
In this study, a method for parameterizing unorganized cylindrical point clouds is suggested. The proposed method creates an initial base surface onto which points are projected to estimate parameter values for each point. To produce an initial base surface, we suggest the concept of a virtual turntable and apply multilevel B-splines. Grid points are projected to the point cloud to increase the accuracy of the initial base surface. During the projection process, a modified weight factor is introduced. Lastly, global B-spline surface interpolation is executed to generate an initial base surface, which undergoes a process of refinement and relaxation. Parameter values are then obtained by projecting the point cloud onto the surface orthogonally. Experiments show that the proposed method successfully estimates parameters and solves the problems of self-loop and crossover. Furthermore, the results of experiment show that the proposed weight factor is more effective than the existing weight factor for point directed projection.

Keywords
Parameterization, Base surface, Multilevel B-splines, B-spline surfaces

1 INTRODUCTION
Recently, three-dimensional (3D) scanners have been used to obtain 3D unorganized points called a point cloud. However, such points cannot be utilized in many applications directly because the point cloud possesses only depth or position information. Thus, much research has focused on means by which to obtain geometric information from the point data for use in applications. To obtain geometric properties from point data, a surface representing the point data can be constructed. Various properties can be then obtained from the surface directly. One means of obtaining such a surface is to fit data points using a parametric surface such as a B-spline, where parametrization of data points is required for fitting. A parametric surface can then be obtained from the parameters and the data points in the least squares sense. In particular, surface fitting is a crucial part [VMC97] in reverse engineering, and parametrization is essential in this process. However, a-point-based parameterization method for a cylindrical shape has not been proposed and no robust method has been suggested for parameterization.

Many authors have proposed methods for surface reconstruction. However, the main purpose of our method is improving the parameterization of point clouds for B-spline surface fitting using a surface reconstruction method as an intermediate step. Therefore, a complete review of the papers on surface reconstruction was not included in this paper [BTS17] [GGG08] [KH13].

For parameterization of unorganized points, Ma and Kruth [MK95] proposed a base-surface. Boundary or section curves are used to create a base surface. This surface roughly reflects the shape of the points. Then, parameterization values are estimated by projecting the points onto the surface orthogonally. However, this approach is hard to apply to irregularly scattered data and more complexly shaped point clouds. Azariadis [Aza04] introduced the notion of dynamic base surface (DBS) in which the concept of points projection is applied to improve an initial base surface iteratively by projecting the surface grids onto an input point cloud.
The error function and weight factor are suggested for calculating a point direct projection (DP). DBS achieves parameterization of more complex point clouds by overcoming the limitations of the base surface method. However, DBS encounters problems of self-loops and cross-overs during surface generation and four boundary curves information should be provided to produce an initial base surface. Azariadis subsequently introduced a subdivision technique for a point cloud in order to improve the accuracy of the DBS [AS07]. The author also suggested a new weight factor which not only considers the distance between the projection points and point cloud but also the distance of the axis direction from the point cloud [AS05]. However, this method still assumes that four boundary curves are given and does not consider the problems of self-loops and crossovers. Following the DBS method, many authors have suggested improvements to projection methods. Liu et al. [LPY06] proposed a method to determine unknown normal vectors. This approach enables projection without information about the project direction. Du and Liu [DL08] suggested projecting the points onto noisy point clouds. Zhang and Ge [ZG10] introduced a method to determine a virtual base surface close to the point cloud without requiring boundary curve information used by some of the previous approaches. Furthermore, this approach drastically minimizes the problems of self-loops and crossovers and suggests the modified weight factor for projecting points to the point cloud more effectively for the squared distance error computation.

The main contributions of this study are twofold as follows. First, we introduce the concept of a virtual turntable and propose a method of creating an accurate initial base surface by applying the multilevel B-spline method [LWS97] using the hidden point removal technique [KTB07]. Second, we suggest a modified weight factor for point projection to cylindrical point clouds that is more efficient than existing weight factors.

2 BACKGROUND THEORY

In this section, multilevel B-spline approximation [LWS97], points direct projection [Az04] and dynamic base surface [Az04] are explained briefly.

2.1 Multilevel B-spline approximation

B-spline approximation

Let $\Omega = \{ (x, y) \mid 0 \leq x \leq m, 0 \leq y \leq n \}$ be a rectangular domain in the $xy$-plane and there is a lattice $\Phi$ which has $(m+3) \times (n+3)$ control points as in Fig. 1. $\Phi_{ij}$ is the value of the $ij$-th control point on $\Phi$, located at $(i, j)$ for $i = -1, 0, 1, \ldots, m+1$ and $j = -1, 0, \ldots, n+1$. Then, using these control points the following equation is derived that defines a B-spline surface near the position $(i, j)$.

$$F(x, y) = \sum_{k=-1}^{3} \sum_{l=-1}^{3} B_k(s)B_l(t)\Phi_{i+k,j+l},$$

(1)

where $i = |x| - 1$, $j = |y| - 1$, $s = x - |x|$ and $t = y - |y|$. $B_k$ and $B_l$ are the uniform cubic B-spline basis functions. Consider there is a point $p(x_c, y_c, z_c)$ and $1 \leq x_c, y_c < 2$. Then, Eq. (1) is given as

$$F(x, y) = z_c = \sum_{k=-1}^{3} \sum_{l=-1}^{3} B_k(s)B_l(t)\Phi_{il}.$$

(2)

$\Phi_{il}$ are control points to be determined. Using the least-squared sense, they are computed by

$$\Phi_{il} = \frac{B_k(s)B_l(t)z_c}{\sum_{a=0}^{3} \sum_{b=0}^{3} (B_a(s)B_b(t))^2}.$$

(3)

Suppose there is a point cloud. Each point has its own $4 \times 4$ control points by computing Eq. (3). Some of the points close to each other may have overlapped control points. Therefore, to deal with this situation the following equation for $\phi_i$ is suggested:

$$\phi_i = \frac{B_k(s)B_l(t)z_c}{\sum_{a=0}^{3} \sum_{b=0}^{3} (B_a(s)B_b(t))^2}.$$

(4)
and \( \phi_{ij} \) is chosen to minimize the error
\[
e(\phi_{ij}) = \sum_{k=0}^{2} \sum_{t=0}^{3} (B_k(s)B_t(t)\phi_{ij} - B_k(s)B_t(t)\phi_j)^2.
\]

Then, differentiating \( e(\phi_{ij}) \) with respect to \( \phi_{ij} \) gives
\[
\phi_{ij} = \frac{\sum_{k=0}^{3} \sum_{t=0}^{3} (B_k(s)B_t(t))^2 \phi_k}{\sum_{k=0}^{3} \sum_{t=0}^{3} (B_k(s)B_t(t))^2}.
\]

When there are shared control points \( \phi_{ij} \), Eq. (5) provides the least squared solution for minimizing a local approximation error. If there is no overlapped control points, a zero value is allocated to \( \phi_{ij} \). Alternatively, the following equation also suggested [Hje01], which is less sensitive to outliers.
\[
\phi_{ij} = \frac{\sum_{k=0}^{3} \sum_{t=0}^{3} (B_k(s)B_t(t)) \phi_k}{\sum_{k=0}^{3} \sum_{t=0}^{3} (B_k(s)B_t(t))^2}
\]

### 2.2 Point directed projection (DP)

Assume that there is a point cloud consisting of \( n \) points: \( p_d = (x_d, y_d, z_d) \), \( d = 0, 1, \cdots n-1 \) with \( a_d \).

Let us assume that there is a point \( p = (x, y, z) \) to be projected onto the point cloud. Then, the sum of the weighted squared distances can be computed:
\[
E(p) = \sum_{d=0}^{n-1} a_d ||p - p_d||^2
\]
\[
= \sum_{d=0}^{n-1} a_d [(x-x_d)^2 + (y-y_d)^2 + (z-z_d)^2].
\]

Eq. (7) can be computed quickly using a five dimensional vector \( c = (c_0, c_1, c_2, c_3, c_4) \) by [EM99],
\[
c_0 = \sum_{d=0}^{n-1} a_d,
\ c_1 = \sum_{d=0}^{n-1} a_d x_d,
\ c_2 = \sum_{d=0}^{n-1} a_d y_d,
\ c_3 = \sum_{d=0}^{n-1} a_d z_d,
\ c_4 = \sum_{d=0}^{n-1} a_d (x_d^2 + y_d^2 + z_d^2),
\]
\[
E(p) = c_0 (x^2 + y^2 + z^2) - 2(c_1 x + c_2 y + c_3 z) + c_4.
\]

Using Eq. (10), Eq. (9) can be written as
\[
E(p^*) = c_0 ((x^*)^2 + (y^*)^2 + (z^*)^2)
\ - 2(c_1 x^* + c_2 y^* + c_3 z^*) + c_4.
\]

#### 3 PROPOSED METHOD

### 3.1 Overall process of the proposed method

The main idea of the proposed method is to produce an initial base surface close to a point cloud. Multilevel B-splines are used for this purpose. However, the multilevel B-splines method can only be applied to 2.5D data. Therefore, a virtual turntable is introduced and the hidden point removal technique [KTB07] is employed.
to transform the point cloud to 2.5D data. The transformed data are used as input for the multilevel B-spline method. Surfaces are generated and grid points on the surfaces are computed. These points in the same layer are used for B-spline curve interpolation and projected to the point cloud. This process is repeated until the user-defined terminate condition is satisfied. An initial base surface is then generated by interpolating the grid points and it undergoes a process of refining and relaxation.

The simple human head point cloud (Fig. 3) is used to describe the proposed method. Fig. 2 shows the outline of the proposed algorithm and the following subsections explain each step in detail.

3.2 Virtual turntable and hidden points removal

Motivated by the real scanning process, we introduce a virtual turntable concept for the next step of generating surfaces using multilevel B-spline [LWS97]. However, the multilevel B-spline method is not applicable to data points sampled from a nonplanar surface. Therefore, we convert the point cloud to 2.5D data by combining the virtual turntable with the hidden point removal method [KTB07].

![Figure 2: The flowchart of the proposed method](image)

Figure 2: The flowchart of the proposed method

Assume that there is a point cloud \( p_i = (p_{ix}, p_{iy}, p_{iz}) \), \( (i = 1, \ldots, n) \). The point cloud is projected to the \( \text{xy} \)-plane and the two points the distance of which is the largest are selected for making a bounding sphere by using an efficient bounding sphere [Rit90]. Let \( c_0 \) be the middle of these two points. Then we have

\[
P_i' = M \begin{pmatrix} p_{ix} - c_{0x} \\ p_{iy} - c_{0y} \\ p_{iz} - c_{0z} \end{pmatrix}.
\]

Using this \( c_0 \), the point cloud’s center can be moved to the \( y \)-axis by computing Eq. (17) and the point cloud can be rotated maintaining its center position by the \( y \)-axis rotation matrix \( M \). The result is named as \( P_i' \).

The view position is set to be oriented in the \( z \)-axis direction from \(+z\) to \(-z\) for applying the hidden point removal method.

\[
P_i'' = M^{-1} \begin{pmatrix} p_{ix}' \\ p_{iy}' \\ p_{iz}' \end{pmatrix} + \begin{pmatrix} c_{0x} \\ c_{0y} \\ c_{0z} \end{pmatrix}.
\]

The point cloud \( P_i'' \) can be moved to the original position \( P_i'^{'} \) by computing Eq. (18). By rotating 90 degrees, four different results are acquired and moved to the original position after the hidden point removal process (Fig. 4).

3.3 Generate surfaces and compute grid points

After the point cloud is rotated and the hidden points are removed, the multilevel B-spline method is applied using the \( xy \)-plane control lattice. The generated surfaces are moved to the origin location (Fig. 5). Then, the equal number of parameters \( u \) and \( v \) are used to produce the surface grid points. Using the \( u \) direction grid points positioned at the same height (\( y \) coordinate), the closest
pairs between the two surfaces are treated as intersection pairs. By analysis of these pairs, the grid points irrelevant to the point cloud are discarded. Note that the surface should produce the same number of grid points for finding the intersection pairs and the $u$ direction is parallel to the $xz$-plane. During the implementation of the multilevel B-spline method, the control lattice resolution affects the accuracy of the generated surfaces.

3.4 B-spline curve interpolation

The remaining grid points are used for global B-spline curve interpolation [PT97] along the row direction ($u$ direction) layer by layer (Fig. 6). The column grid points close to the point cloud are selected as the starting and ending interpolation points. Interpolations are implemented through the remaining points after outliers are removed. Uniform sampling is then implemented based on the numbers of points remaining in each layer. During the interpolation process, parameterization is conducted by using the chord length method and knot vectors are computed by using the averaging method [MK95]. All grid points are then interpolated layer by layer. The results of all curves are then identified as the initial curves. Note that each layer has a different number of sample points as a result of using uniform sampling. These sample points are employed following the step of point projection.

3.5 Project points to the point cloud

The grid points acquired by uniform sampling are employed to project to the point cloud. Projection directions are attained by the derivative of B-spline curves. Directed projection (DP) approach is applied for the projection process. In the experiment, the projection using the weight factors Eq. (15) or Eq. (16) does not produce different results. Therefore, the following modified weight factor is used.

$$a_u = \frac{1}{\|p - p_u\|^4 \| (p_u - p) \times n \|^2}, \quad a_u = [0, \infty] \quad (19)$$

After projection to the point cloud, resampling and projection are repeated until the user defined threshold is satisfied to increase the accuracy of each curve.

3.6 Initial base surface

After projection is terminated, the same number of samples is implemented by each curve for the global B-spline surface interpolation [PT97]. This is because each curve has a different number of points. If the number of samples is inadequate, the accuracy of the surface is degraded. Therefore, the sampling number is set to the number of points in the curve that has the highest number of samples. During surface interpolation, parameterization is performed through uniform parameterization. In addition, non-uniform knot vectors are selected. An initial base surface is then generated by interpolating points from the final curves.

3.7 Refining and relaxation

After the initial B-spline surface is generated, refining and relaxation are conducted. In the former, the grid is projected over the point cloud along the surface normal direction. The latter is the method used in the DBS approach. This grid is then used to create a B-spline surface again. Finally, we obtain a B-spline surface that approximates the point cloud, and the parameters are obtained by projecting the point cloud onto the surface orthogonally.
4 RESULTS AND DISCUSSION

The simple human head point cloud, the Lisa skull head point cloud, and the Stanford bunny point cloud are all employed to demonstrate the proposed method. Surfaces are generated to approximate each point cloud and parameterization is achieved by projecting the point cloud onto the surface orthogonally. Parameterized points are then exploited for the least squares B-spline surface fitting in order to show that the parameterization has been successfully accomplished. Figs. 7a-9a show the point cloud used to test the proposed method of parameterization and the point cloud is approximated by the final surface (Figs. 7b-9b). After parameterization using the final surface, least squares B-spline surface fitting is implemented as shown in Figs. 7c-9c.

In our experiment, final surfaces accurately approximated point clouds and parameterization was implemented successfully as shown in Figs. 7-9. Compared to the DBS method [Aza04], the proposed method is much simpler and minimizes user intervention for problems of self-loops and crossovers, as initial base surfaces are generated close to the point cloud without boundary curve information. To avoid these problems, each curve and the projection directions are restricted to a plane. This approach drastically minimizes problems.

In addition, the modified weight factor can project a point to point clouds in few iterations (Figs. 10-11). Detailed results are listed in Tables 1-2. In the experiment, the modified weight factor in Eq. (19) was more efficient than those of Eq. (15) or Eq. (16) in terms of RMSE accuracy and speed. Eq. (15) requires more iterations to project a point onto a point cloud because the value of $t$ in Eq. (10) is small when the points that are far from the projection direction have strong weights. Eq. (16) overcomes this problem by introducing the axis distance. However, it is inefficient in the surrounding point clouds. Points near the axis of direction that is behind projection direction also get strong weights. By Comparison, the modified weight factor using Eq. (19) is efficient for the surrounding point clouds, because it not only considers the projection direction but also gives stronger weights to near points.

Note that the values of root mean square error (RMSE) are calculated using the minimum distances between the point cloud and projection points.

5 CONCLUSION

This study presented a method for parameterization of unorganized cylindrical point clouds. A virtual turntable concept was introduced for applying the multilevel B-splines method to a point cloud. The multilevel B-splines method can generate a surface close to a point cloud in less time. This more accurate surface reduces the problems of self-loops and crossovers for the projection process. Furthermore, four boundary curves are not required to generate an initial base surface. During projection process, a modified weight factor can efficiently project points onto a point cloud. In our study, selected point clouds were tested to demonstrate the proposed method. Our experiments confirmed that parameterization was successfully accomplished, as shown in Figs. 7-9.

However, many possibilities remain for improving the proposed method. Using the T-spline would generate a surface that is efficient and accurate without having to resample for the global B-spline surface interpolation. Decreasing or increasing rotation angles of a virtual turntable based on the complexity of point clouds interpolation. Decreasing or increasing rotation angles of a virtual turntable based on the complexity of point clouds would make the proposed method more robust. These are recommended suggestions for future work.
Figure 7: Result of the proposed method and application in least squares B-spline surface fitting. (a) The simple human head point cloud (N=64009, Bounding box(158x246x190)). (b) The final surface (51x101 control points, RMSE=0.221). (c) Result of the B-spline surface fitting (55x55 control points).

Figure 8: Result of the proposed method and application in least squares B-spline surface fitting. (a) The Lisa skull head point cloud (N=28384, Bounding box(22x182x234)). (b) The final surface (37x120 control points, RMSE=0.144). (c) Result of the B-spline surface fitting (39x39 control points).

Figure 9: Result of the proposed method and application in least squares B-spline surface fitting. (a) Part of the Stanford bunny point cloud (N=16234, Bounding box(270x117x138)). (b) The final surface (24x110 control points, RMSE=0.184). (c) Result of the B-spline surface fitting (30x30 control points).
Figure 10: Results of a single iteration projection to the simple human head point cloud based on the weight factor. Black points are sampled from a B-spline curve. Red points are results of projection. (a) DP method using Eq. (15) (b) DP method using Eq. (16) (c) DP method using the proposed Eq. (19).

Table 1: Comparison of RMSE errors between the proposed weight factor Eq. (19) and DP method using Eq. (15) during projecting points to the simple human head point cloud.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Proposed weight factor (RMSE)</th>
<th>Azariadis’s 2004 factor (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.289</td>
<td>0.289</td>
</tr>
<tr>
<td>1</td>
<td>0.240</td>
<td>0.252</td>
</tr>
<tr>
<td>2</td>
<td>0.236</td>
<td>0.252</td>
</tr>
<tr>
<td>3</td>
<td>0.236</td>
<td>0.248</td>
</tr>
<tr>
<td>4</td>
<td>0.236</td>
<td>0.247</td>
</tr>
<tr>
<td>5</td>
<td>0.236</td>
<td>0.245</td>
</tr>
</tbody>
</table>

Figure 11: Results of a single iteration projection to part of the Stanford bunny point cloud based on the weight factor. Black points are sampled from a B-spline curve. Red points are results of projection. (a) DP method using Eq. (15) (b) DP method using Eq. (16) (c) DP method using the proposed Eq. (19).

Table 2: Comparison of RMSE errors between the proposed weight factor Eq. (19) and DP method using Eq. (15) during projecting points to part of the Stanford bunny point cloud.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Proposed weight factor (RMSE)</th>
<th>Azariadis’s 2004 factor (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.305</td>
<td>0.305</td>
</tr>
<tr>
<td>1</td>
<td>0.168</td>
<td>0.181</td>
</tr>
<tr>
<td>2</td>
<td>0.149</td>
<td>0.151</td>
</tr>
<tr>
<td>3</td>
<td>0.134</td>
<td>0.148</td>
</tr>
<tr>
<td>4</td>
<td>0.126</td>
<td>0.141</td>
</tr>
</tbody>
</table>
6 ACKNOWLEDGMENTS

This work was supported by the GIST Research Institute (GRI) in 2017.

7 REFERENCES


