A Ridge-Valley Line Extraction Method on Hole Filling of Groove Point Cloud

Jie Cheng*, Yong Yin
School of Information Engineering Wuhan University of Technology, Wuhan, 430000, China
chengjie_work@163.com

Abstract. Among several teaching methods of welding robot, the one based on 3D point cloud path extraction is a much more intuitive approach. However, due to reflectivity, corrosion and insufficient illumination, etc., the point cloud on the workpiece is squeezed into a small hole that hinders the path planning. Thus, this paper puts forward a hole segmentation and filling algorithm on the basis of ridge-valley line extraction. Firstly, the discrete curvature of the point cloud surface is estimated by Weingarten mapping properties, and the ridge and valley feature points are identified. Then, an improved connection algorithm is used to connect the ridge and valley points, and the feature lines obtained by fitting divide the holes into simple sub-holes. Finally, after the rough filling of each sub-hole with triangulation and subdivision algorithm, the positions of the newly added vertices are adjusted to the RBF (Radial Basis Function) implicit surfaces to complete the hole filling with sharp and curved features preserved. The experimental results show that this method can restore the sharp and curved features well on different groove point clouds.

1. Introduction
In traditional teaching methods of welding robot, the welding gun is guided by manual or teaching device. The accuracy is determined by visual observation. Therefore, the advantages such as high level of automation and high precision of the teaching methods based on vision system are highlighted. Among all sorts of implementations, the technique based on 3D point cloud has richer information and more accurate geometry data. For the groove workpiece, in path planning, the parameters such as the number of welding layers and passes, the position of welding wire tip and the attitude of welding gun are closely related to the geometric features of it [1]. However, due to the surface reflectance, corrosion and insufficient illumination, the groove point cloud model often results in defects such as holes. Simple holes appear on the surface with lower curvature, while complex holes appear at the intersection of different groove faces accompanied by the loss of sharp features. This will impact the acquisition of geometric information, thus making path planning deviate from its actual goal. Even in extreme cases that some grooves have small groove angle, the welding gun would collide with the groove sidewall.

In order to eliminate the influences of holes, scholars have proposed many hole filling algorithms on 3D model. Ju [2] divides space into interior and exterior volume by using an octree grid. Then repairing the surface of polygon model by contour. However, the input is changed with the loss of sharp features. Attene [3] concentrates on converting a low-quality polygon mesh to a single manifold and watertight triangular mesh without degenerating or intersecting elements. But it can't guarantee every result will eventually converge. Centin et al. [4] clean and smooth the hole by using Laplace operator, then the triangulation and subdivision algorithm are applied to achieve semi-automatic repair effect with
adjustable density. Wu et al. [5] propose two filling strategies for holes in different positions: boundary extension and boundary convergence. Both methods are able to close holes but cannot maintain sharp or even curved features well and the latter has the defects of time-consuming and uneven density. Altantsetseg et al. [6] propose a curve shortening flow method to shrink the hole boundary layer by layer. But the generated surfaces are usually difficult to integrate with the surrounding of holes. Yan et al. [7] construct an implicit surface based on RBF, and approximate the filling vertices to the surface. The reduction of the hole area can fit the original model as much as possible. Lin et al. [8] and Wang et al. [9] divide the feature hole into small non-feature holes and fill them respectively. However, the filling effect is limited by curvature.

Parts of the above approaches are done on mesh models. However, acquisition of point cloud is more direct. It avoids time-consuming triangulation and other preprocessing. In addition, although some methods can extract sharp feature lines to segment the holes, the filling effect is limited by their high curvature, which may still cause calculation errors of geometric parameters of the groove. In this paper, we propose a method of hole segmentation and filling based on the improved ridge-valley line extraction on point cloud. Firstly, Weingarten mapping is used to estimate the discrete curvatures and principal directions locally. Then, using an improved connection algorithm to establish the topological relationship between the feature points, and accurately segment the holes across sharp features into simple sub-holes by fitting and sampling the ridge and valley feature lines. Finally, triangulation and subdivision algorithm and implicit surface based on RBF are used to roughly fill and optimize the sub-holes, thus ensure the reduction of the sub-holes while retaining the sharp features.

2. Segmentation of hole

2.1. Hole detection

Since the point cloud of groove has no topological information, it is necessary to detect holes with the help of neighborhood vertices. The normal vector and tangent plane of a hole boundary vertex are estimated from its neighborhood vertices. After all the vertices in the neighborhood are projected to the tangent plane, the identification of boundary vertex can be considered on a two-dimensional plane. On this basis, Bendels et al. [10] introduce three judging criterias: the angle criterion, the halfdisc criterion and the shape criterion. The boundary extracted by the angle criterion is clearer and thinner. The halfdisc criterion performs better on small hole boundaries. And the shape criterion can effectively suppress the noise. Therefore, we combined these advantages of the three criteria to extract the hole boundary.

2.2. Extraction of ridge and valley feature points

The ridge and valley feature lines are curves conform to the concave and convex variation trend of the surface on a 3D model and can describe the shape in detail. Compared with methods of deriving curvatures from higher derivatives of locally fitted surfaces [11], in this paper, by utilizing the nature of Weingarten mapping, the principal curvatures and directions can be easily estimated by differential values.

The normal curvatures of a vertex in all directions can be given by the Weingarten mapping matrix $W$. The eigenvalues are the principal curvatures, and the corresponding eigenvectors are the principal directions. Since $W$ is a symmetric transformation, we can assume that

$$W = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \quad (1)$$

As shown in figure 1, orthogonal unit vectors $\mathbf{u}$ and $\mathbf{v}$ are perpendicular to the normal vector $\mathbf{n}_p$ of vertex $p$. 1-ring neighborhood vertices $p_i (i = 1, 2, \cdots, m)$ are projected to the tangent plane of $p$ along their normal vector $\mathbf{n}_{p_i}$ to obtain the projection points $p'_i$ and the projection vectors $\mathbf{n}'_{p_i}$. We can consider that $pp'_i$ and $\mathbf{n}'_{p_i}$ are the differential values of $pp_i$ and $\mathbf{n}_p$, respectively. Only when $m \geq 2$, we can solve equation (2) under the significance of least square.
\[
\begin{pmatrix}
a \\
b \\
c \\
d
\end{pmatrix}
\begin{pmatrix}
pp' \cdot u \\
pp' \cdot v
\end{pmatrix} =
\begin{pmatrix}
n'_{p_i} \cdot u \\
n'_{p_i} \cdot v
\end{pmatrix}
\]  \hspace{1cm} (2)

It should be noted that isolated noise vertices which are unwelcome may also have more than 2 neighborhood vertices, so we can add a distance constraint condition: For a vertex which has \( m' \) 1-ring neighborhood vertices, if the remaining \( m \) vertices do not satisfy \( m \geq 2 \) after removing its neighborhood vertices whose distance from the center are greater than \( \lambda d \) (\( d \) is the average density of the point cloud, and we let \( \lambda = 3.0 \)), this vertex is excluded from view.

---

**Figure 1.** Locally mapping.  \hspace{1cm}  **Figure 2.** Estimating the discrete curvatures.

After solving the matrix \( W \), the Gaussian curvature \( K \) and mean curvature \( H \) of the vertex \( p \) are given by equation (3), and the principal curvature \( k_{\text{max}} \) and \( k_{\text{min}} \) are obtained from equation (4) and (5). The eigenvectors of \( W \) with eigenvalues \( k_{\text{max}} \) and \( k_{\text{min}} \) are the principal directions of \( t_{\text{max}} \) and \( t_{\text{min}} \) in the \( u-v \) two-dimensional coordinate system.

\[
K = ac - b^2, \quad H = \frac{1}{2}(a + c)
\]  \hspace{1cm} (3)

\[
K = k_{\text{max}} \cdot k_{\text{min}}, \quad H = \frac{1}{2}(k_{\text{max}} + k_{\text{min}})
\]  \hspace{1cm} (4)

\[
k_{\text{max}} = H + \sqrt{H^2 - K}, k_{\text{min}} = H - \sqrt{H^2 - K}
\]  \hspace{1cm} (5)

According to the definition [12], the vertex where \( k_{\text{max}} \) or \( k_{\text{min}} \) have an extremum on the normal section through \( t_{\text{max}} \) or \( t_{\text{min}} \) is a ridge or valley point. Then there is the following discrete linear interpolation method to judge the property of a vertex \( p \): As shown in figure 2, a local triangular mesh is constructed by vertex \( p \) and its 1-ring neighborhood vertices. The normal section through \( t_{\text{max}} \) intersects with two triangular edges at point \( A \) and \( B \) respectively. After calculating the principal curvatures of vertex \( p_1, p_2, p_3 \) and \( p_4 \), we suppose the lengths of \( Ap_1, Ap_2, Bp_3 \) and \( Bp_4 \) are denoted by \( d_1, d_2, d_3 \) and \( d_4 \) respectively, then the principal curvature \( k_{\text{4, max}} \) and \( k_{\text{4, max}} \) at point \( A \) and \( B \) are simply given by equation (6). If the principal curvature \( k_{\text{max}} \) of \( p \) satisfies 1 of equation (7), then \( p \) is considered as a ridge point. The same method can be used for the judgment of a valley point, except that the normal section passes through \( t_{\text{min}} \) and the determinant conditions become II of equation (7).

\[
\begin{align*}
k_{\text{4, max}} &= k_{\text{p_1, max}} + \frac{d_1}{(d_1 + d_2)}(k_{\text{p_2, max}} - k_{\text{p_1, max}}) \\
k_{\text{4, max}} &= k_{\text{p_3, max}} + \frac{d_1}{(d_3 + d_4)}(k_{\text{p_4, max}} - k_{\text{p_3, max}})
\end{align*}
\]  \hspace{1cm} (6)
Due to the noise of the groove point cloud and the errors of curvature and normal estimation, results obtained will be more than expected. Hence the shape information of a larger area can be considered here to reduce the misjudgment. The specific optimization methods are as follows: ① We can take the neighborhood vertices of up to 2-3 rings into the overdetermined equations composed of equation (2). ② The local triangular mesh can be constructed by using 2-3 rings neighborhood vertices, and then the intersection points on two outermost triangular edges are selected as $A$ and $B$. ③ Parameters $\mu_r$ and $\mu_v$ can be set to adjust the sensitivity. That is to say, vertex $p_i$ can be determined as a ridge point if $k_{\text{max}}$ is greater than $\mu_r k_{\text{max}}$ and $\mu_r k_{\text{max}}$ or can be judged as a valley point if $k_{\text{min}}$ is less than $\mu_v k_{\text{min}}$ and $\mu_v k_{\text{min}}$. Here, we take 2-rings neighborhood vertices and set $\mu_r = \mu_v = 1.1$.

### 2.3. Reconstruction of feature lines

Up to now, the ridge point set $\mathcal{R}_r$ and valley point set $\mathcal{R}_v$ are still scattered point clouds. In order to reconstruct the feature lines, the ridge or valley feature points which belong to the same sharp edge but truncated by the hole should be connected and stored respectively.

When the normals of all vertices point to the same side of the model and the observer is also located on this side, we should say all the ridge and valley points correspond to the convex and concave features respectively. And $t_{\text{min}}$ or $t_{\text{max}}$ of ridge or valley points on the same sharp edge, are roughly tangential to the ridge or valley line. However, according to equation (2), when $u$ and $v$ at the vertex $p_i$ in figure 3 are hypothetically reversed, the solution of the equation, the eigenvalues and the eigenvectors of the matrix $W$ will not be affected. But the principal direction obtained will be reversed. So in such cases that directions of $u$ and $v$ can be arbitrarily selected, while the obtained $t_{\text{min}}$ and $t_{\text{max}}$ can still correctly represent the direction where the curvature extremum exists, the $t_{\text{min}}$ or $t_{\text{max}}$ of adjacent vertices on the same ridge or valley line may point in roughly opposite directions, leading to the change in direction or cease of the connecting process. Therefore, the existing connection algorithm [13] is improved by adding the angular constraint between principal directions and reversing the principal direction.

![Figure 3. Possible change of the principal direction.](image)

Taking connecting ridge points as an example, the improved connection algorithm is as follows:

**Step1:** For a vertex $r \in \mathcal{R}_r$, the angles between $t_{\text{min}}$ and all vectors that start with the current search point $r$ and end with the neighborhood vertices within a radius of $2d$ are calculated. The neighborhood vertex with minimum angle $\theta$ is set as $r'$, and the angle between $t'_{\text{min}}$ and $t_{\text{min}}$ at $r'$ is $\delta$. If $\theta$ is less than the threshold $\tau_\theta$, and $\delta$ meets $\frac{\pi}{2} - \left| \frac{\pi}{2} - \delta \right| < \tau_\delta$ ($\tau_\theta$ and $\tau_\delta$ are set as $\frac{\pi}{6}$), $r'$ will be the next searching point. After this, if $\delta < \tau_\delta$, the next searching direction is $t'_{\text{min}}$, and if $\delta > \pi - \tau_\delta$, $t'_{\text{min}}$ will be reversed firstly and then set as the next searching direction, as shown in...
figure 4. If there is no ridge point that meets the requirements above in this neighborhood, remove \( r' \) from \( \mathcal{R}_r \) and expand the radius to conduct the above search operation one more time. After finding the wanted ridge point, go to Step2, otherwise, go to Step3.

**Step2:** The current searching point \( r \) is removed from \( \mathcal{R}_r \) and saved in the ridge sequence \( R_n \). \( r' \) then becomes the new current searching point \( r \) and the program returns to Step1.

**Step3:** If the initial searching point is removed from \( \mathcal{R}_r \), then we put it back to \( \mathcal{R}_r \) and the searching process goes on along its direction \( -t_{\text{min}} \), otherwise it goes on along \( -t_{\text{min}} \) directly. Until no new searching points are generated, go to Step4.

**Step4:** Remove the extra vertices around \( R_n \) from \( \mathcal{R}_r \). Take another vertex in different part of \( \mathcal{R}_r \) as the initial search point of the new ridge line and create a new ridge sequence \( R_{n+1} \). Repeat Step1-3 until there are no vertices left.

A similar method is used for valley points, except that the initial searching direction becomes \( t_{\text{max}} \).

Figure 5 is a part of the experimental result of connecting ridge points, in which lines connect the ridge points, and the principal directions are shown on each ridge point where all the \( t_{\text{min}} \) have been corrected to the same direction.

The cubic b-spline curve fitting method is used to reconstruct the missing segment between two ridge or valley lines truncated by the hole. Then the reconstruction part is sampled at equal interval. Since these sampling points are now part of the final filling result, in order to ensure that the density of the hole after filling is consistent with the surroundings, samples are taken at interval of \( d \).

### 3. Hole filling

The strategy of rough filling and optimization is adopted for each sub-hole after segmentation. For scattered point cloud, most methods need to determine the directions of the new vertices by utilizing the neighborhood information of the hole, such as the boundary extension method [5]. However, there are no neighborhood vertices around the newly added ridge or valley feature points from sampling, so the correct directions of the following filling points cannot be generated.

Fortunately, in the researches of repairing triangular mesh, triangulation and subdivision algorithm has good performances in efficiency and ensuring the consistency of the density. We decide to use the algorithm of [14] for roughly filling of each sub-hole. This method does not require the existence of neighborhood vertices. Firstly, the sub-holes are triangulated according to the principle of minimum angle cutting. At this time, they are filled by triangles with large areas, and there are no newly generated vertices in them. Then, if the distances between the center and three corners of each triangle are greater than \( d \), the center point and the three corners form three new triangles, and this center point becomes a new vertex. Finally, after repeating preceding steps until no new vertices are created, the rough filling of the sub-hole is completed and only new vertices, not triangular faces, need to be preserved. It is worth mentioning that although the sampling points of each ridge or valley feature line are involved in the filling on both sides, this algorithm do not change their positions, so the sharp features are maintained.

Each sub-hole is then optimized by approximating the newly added vertices to the implicit surface based on RBF. The advantage is that the solution of the interpolation formula composed of RBF is the
solution of a linear system [15], which is very convenient for small amount of scattered data around the hole.

The implicit surface equation based on radial basis function can be expressed as equation (8).

\[ F(r) = \sum_{i=1}^{2n} \omega_i \phi(r - q_i) + P(r) \]  

\( \omega_i \) are the weight factors to be found of every point. As a linear part,

\[ P(r) = p_0 + p_1x + p_2y + p_3z \]  

\( \phi(r - q_i) \) stands for the radial basis function and usually take

\[ \phi(r - q_i) = \left( (r^x - q^x_i)^2 + (r^y - q^y_i)^2 + (r^z - q^z_i)^2 \right)^{\frac{3}{2}} \]  

in the cases of 3D scattered points interpolation. The following, \( \phi_i \), refers to \( \phi(q_i - q_i) \).

We collect the \( n \) actual boundary vertices of a sub-hole and their 1-ring neighborhood vertices as the interpolation constraint points with the constraint value of \( h_i = 0(i = 1,2,\cdots,n) \), and take the points with the distance of \( d \) along their normal directions as additional constraint points with the constraint value of \( h_i = 1(i = n + 1, n + 2, \cdots, 2n) \) [16]. Then the equation (12) can be constructed after substituting all the constraint points with the number of \( 2n \) and the orthogonality condition

\[ \sum_{i=1}^{2n} \omega_i = \sum_{i=1}^{2n} \omega_i q_i^x = \sum_{i=1}^{2n} \omega_i q_i^y = \sum_{i=1}^{2n} \omega_i q_i^z = 0 \]  

into equation (8).

\[
\begin{bmatrix}
\phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,2n} & 1 & q_1^x & q_1^y & q_1^z \\
\phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,2n} & 1 & q_2^x & q_2^y & q_2^z \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\phi_{2n,1} & \phi_{2n,2} & \cdots & \phi_{2n,2n} & 1 & q_{2n}^x & q_{2n}^y & q_{2n}^z \\
1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\
q_1^x & q_1^y & \cdots & q_{2n}^x & q_1^y & q_1^z & \cdots & q_{2n}^z \\
q_1^x & q_1^y & \cdots & q_{2n}^x & q_1^y & q_1^z & \cdots & q_{2n}^z \\
q_1^x & q_1^y & \cdots & q_{2n}^x & q_1^y & q_1^z & \cdots & q_{2n}^z \\
\end{bmatrix}
\begin{bmatrix}
\omega_1 \\
\omega_2 \\
\vdots \\
\omega_{2n} \\
p_0 \\
p_1 \\
p_2 \\
p_3 \\
\end{bmatrix}
= 
\begin{bmatrix}
h_1 \\
h_2 \\
\vdots \\
h_{2n} \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}  
\]  

(12)

Since the system is a linear equations system, it is easy to solve \( (\omega_1, \omega_2, \cdots, \omega_{2n}, p_0, p_1, p_2, p_3)^T \) and get the implicit surface equation \( F(r) \).

The newly added vertices of a sub-hole from rough filling are approximated to the surface to merge them with the surroundings. Since the sampling points of ridge or valley feature lines have characterized the sharp features, no adjustment is made to them. Gradient descent method is used in the approximation process. It is an optimal algorithm, also known as the steepest descent method, and it has the characteristics of fast speed and high precision. The specific iterative formula is equation (13).

\[ r_{k+1} = r_k - \gamma \nabla F(r_k) \]  

(13)

\( r_k \) is the point to be adjusted and \( r_{k+1} \) is the adjusted point. The learning rate \( \gamma \) equals to \( \frac{F(r_k)}{\|\nabla F(r_k)\|} \).

\[ \nabla F(r_k) = \left( \frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z} \right)_{r_k} \]  

(14)

representing the gradient of the implicit surface. During the iteration, if \( \|r_{k+1} - r_k\| > \kappa \), then \( r_{k+1} \) is replaced by \( r_k \) and the iteration continues until \( \|r_{k+1} - r_k\| \leq \kappa \). Letting \( \kappa = 10^{-4} \), will give us a good
approximation. After all the newly added vertices of all sub-holes are adjusted, the complex holes across sharp features on the groove point cloud are filled.

4. Experimental results and analysis

The method proposed in this paper is implemented with C++ and PCL1.8.0 on Visual Studio, and holes in three different types of groove point clouds are tested. Firstly, the experiment is conducted on a v-groove. As shown in figure 6(a), the hole is located at the intersection of one groove face and one upper surface. The red line in figure 6(b) connects the sampling points of ridge feature line and segment the hole into 2 sub-holes which are roughly filled by triangulation and subdivision. The light and dark blue points in figure 6(c) are the vertices adjusted by implicit surface approximation. figure 6(d) shows the filling effect after surface reconstruction. The intersection line of two planes is restored effectively. But actually, since the groove faces and the upper surfaces are flat, the planar features have already been well recovered after triangulation and subdivision. It is unnecessary to conduct the approximation process further. However, the u-groove in figure 7(a) is different that a part of its groove face is curved surface. So, the triangulation and subdivision in figure 7(b) and the approximation procedure in figure 7(c) are both needed. Figure 7(d) represents the effect of recovering the curved and planar features and the most important sharp features. The sharp feature lines of the above two kinds of groove are approximately linear. For the non-linear groove whose sharp feature lines are curves, figure 8 gives our experimental results. In figure 8(a), the hole appears at the intersection of two groove faces. The reconstructed feature line which is red in figure 8(b) is very consistent with the actual situation, and the result of filling in figure 8(d) represents a good integration with the surrounding.

5. Conclusion

The method proposed in this paper can achieve good filling effect for holes across sharp features at different positions on different type groove point clouds. By employing the improved Weingarten
mapping ridge and valley line extraction method, we can accurately identify the sharp feature lines and divide the holes into uncomplex sub-holes. Also, in order to retain sharp features, only the regions within the hole boundaries and feature lines are selected for rough filling and the approximation adjustment, so that ensure the correct geometric information of the groove point cloud as much as possible. As stated in the result analysis, the steps of constructing implicit surfaces and approximation are time-consuming and unnecessary when sub-holes are located in a plane. Therefore, in the future work, it is the main improvement direction to add the program to judge the spatial complexity of sub-holes.

References
[1] Yongce Wen, Study on Path Planning of Multi-layer and Multi-pass Robotic Welding for Thick Plate, Shandong University, 2019.
[2] Tao Ju, Robust Repair of Polygonal Models, ACM Transactions on Graphics, 2004, 23(3):888-895.
[3] Marco Attene, A lightweight approach to repairing digitized polygon meshes, Vis Comput, 2010, 26: 1393-1406.
[4] Marco Centin, Alberto Signoroni, RameshCleaner: conservative fixing of triangular meshes, Smart Tools & Apps for Graphics, 2015.
[5] Xiaojun Wu, Wenlue Chen, A Scattered Point Set Hole-filling Method Based on Boundary Extension and Convergence, World Congress on Intelligent Control and Automation, 2014.
[6] Enkhabayar Altantsetseg, Oyundolgor Khorloo, Katsutsugu Matsuyama, Kouichi Konno, Complex hole-filling algorithm for 3D models, In Proceedings of CGI ’17, June 2017.
[7] Haiping Yan, Lushen Wu, Huawei Chen, Hole repairing algorithm in scattered point cloud based on radial basis function, Computer Engineering and Design, 2014, 35(4):1253-1257.
[8] Hongbin Lin, Wei Wang, Feature Preserving Holes Filling of Scattered Point Cloud Based on Tensor Voting, IEEE International Conference on Signal and Image Processing, 2016.
[9] Yinghui Wang, Jing Tang, Yanni Zhao et al., Point Cloud Hole Filling based on Feature Lines Extraction, International Conference on Virtual Reality and Visualization (ICVRV), 2017.
[10] Gerhard H Bendels, Ruwen Schnabel, Reinhard Klein, Detecting Holes in Point Set Surfaces, Journal of WSCG, 2006, vol.14.
[11] Xufang Pang, Mingyong Pang, Chunxia Xiao, An Algorithm for Extracting and Enhancing Valley-ridge Features from Point Sets, ACTA AUTOMATICA SINICA, 2010, 36(8):1073-1083.
[12] Judd, T., Durand, F., Adelson, E., Apparent Ridges for Line Drawing, ACM Transactions on Graphics, 2007, 26(3): 19.1-19.7.
[13] Yiming Tan, Extraction of Ridge and Valley Line on Mesh Model, Zhejiang University, 2010.
[14] Ping Hu, Changsheng Wang, Baojun Li, Mingzeng Liu, Filling Holes in Triangular Meshes with Feature Preserved, Applied Mechanics and Materials, 2011, 1156(101):794-798.
[15] Vaclav Skala, RBF Interpolation with CSRBF of Large Data Sets, International Conference on Computational Science, 2017, Procedia Computer Science 108C 2433–2437.
[16] Greg Turk, James F. O’Brien, Modelling with Implicit Surfaces that Interpolate, ACM Transactions on Graphics, 2002, 21(4):855-873.