Article

Orebody Modeling Method Based on the Normal Estimation of Cross-Contour Polylines

Zhaohao Wu ^{1,2}, Deyun Zhong ^{1,2,3,*}, Zhaopeng Li ^{1,2}, Liguang Wang ^{1,2,3} and Lin Bi ^{1,2,3}

¹ School of Resources and Safety Engineering, Central South University, Changsha 410083, China; wuzhaohao@csu.edu.cn (Z.W.); whpeng2020@csu.edu.cn (Z.L.); liguan_wang@csu.edu.cn (L.W.); mr.bilin@csu.edu.cn (L.B.)

² Digital Mine Research Center, Central South University, Changsha 410083, China

³ Changsha Digital Mine Co., Ltd., Changsha 410221, China

* Correspondence: deyun_zhong@csu.edu.cn

Abstract: The normal estimation of cross-contour polylines largely determines the implicit orebody modeling result. However, traditional methods cannot estimate normals effectively due to the complex topological adjacency relationship of the cross-contour polylines manually interpreted in the process of exploration and production. In this work, we present an orebody implicit modeling method based on the normal estimation of cross-contour polylines. The improved method consists of three stages: (1) estimating the normals of cross-contour polylines by using the least square plane fitting method based on principal component analysis; (2) reorienting the normal directions by using the method based on the normal propagation; (3) using an implicit function to construct an orebody model. The innovation of this method is that it can automatically estimate the normals of the cross-contour polylines and reorient normal directions without manual intervention. Experimental results show that the proposed method has the advantages of a small amount of calculation, high efficiency and strong reliability. Moreover, this normal estimation method is useful to improve the automation of implicit orebody modeling.

Keywords: implicit modeling; radial basis function; principal component analysis; normal propagation

Citation: Wu, Z.; Zhong, D.; Li, Z.; Wang, L.; Bi, L. Orebody Modeling Method Based on the Normal Estimation of Cross-Contour Polyline. *Mathematics* **2022**, *10*, 473. <https://doi.org/10.3390/math10030473>

Academic Editor: Akemi Galvez Tomida

Received: 12 December 2021

Accepted: 30 January 2022

Published: 1 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The geological interpretation of polylines produced by geological logging in the process of exploration and production is more complex than the interpretation of polylines produced by drilling in geological explorations [1]. In this case, the efficiency will be very low if the model is constructed by the contour triangulation method [2] that uses triangulation for surface reconstruction. Moreover, exploration and production are carried out during the operation of the mine, which leads to the result that the traditional modeling methods cannot keep up with the mining plan. Modeling is the basis and premise of mine digital production [3]. Efficient real-time modeling is of great significance for mine production and promoting the development of mine digital construction. The method we proposed can slightly process the geological interpretation polylines to carry out automatic implicit modeling, which greatly improves the modeling efficiency and quality.

In recent years, there has been great progress in both theory and methodology of implicit modeling. A range of implicit modeling methods based on different interpolation functions and geological rules have been proposed [4–8]. The key to implicit orebody modeling is to figure out how to interpolate the unknown 3D orebody model based on the limited geological sampling data. For the complex geological sampling data, the quality of implicit modeling results largely depends on the reliability of constructed constraints. We attempt to solve the problem of normal estimation based on cross-contour polylines (non-parallel geological interpretation polylines) to improve the automation of implicit orebody modeling.

For two kinds of common geological sampling data, drillhole data and geological interpretation polylines, the implicit modeling methods based on them require estimating their normals. The implicit equation is solved based on the estimated normals. The implicit surface reconstruction is carried out based on the solved implicit equation. The research results of normal estimation for dense point clouds are considered relatively mature [9–12]. Generally speaking, they can be divided into three categories: the method based on the local plane fitting method first proposed by Hoppe et al. [9], the method based on Delaunay/Voronoi first proposed by Amenta et al. [10], and the method based on robust statistics [11]. Compared with the point cloud, cross-contour polylines are more difficult to estimate the normals because of their more complex topological adjacency relationships [13].

However, there have been many existing studies on the 3D modeling of directed contours. Shao et al. [14] estimated the normals of the cross-section contours and the surface of the object in the design drawing, then reconstructed the surface to model the design sketch in 3D. Firstly, they proposed a method to estimate the normals of the cross-section contours and found a clear mathematical formula between the cross-section contours and the geometric relationship they convey. Secondly, they used this relationship to realize the algorithm of object surface reconstruction. The limitation of this method is that it is only suitable for constructing the model by normal estimation of the cross-section contours. Park et al. [15] estimated the normal vectors of the 3D curves by a new algorithm. In addition, they established an optimization equation that can reflect the design intention. Through the above two steps, a model with insufficient geometric information can be constructed. Kniss et al. [16] described the concept of multi-dimensional transfer functions in detail and used the functions to form a new technique for interactively constructing the surface of the object. Ijiri et al. [17] improved the traditional methods [18,19] and proposed Bilateral Hermite Radial Basis Functions for Contour-based Volume Segmentation. In addition, they also proposed a cyclic LU decomposition scheme to accelerate B-HRBF calculation, which realized real-time and intuitive volume segmentation of real object images. However, it is difficult for novices to accurately place the cross-section to specify the contour of segmentation, which will affect the effect of segmentation. Zhong et al. [20] proposed an improved method to reconstruct the orebody surface from a set of interpreted cross-sections. Moreover, they also proposed an iterative closest point correction algorithm, which can iteratively correct the distance field according to the constraint rules and the internal and external position relationship of the model. However, this method is only applicable to non-trivial and non-sparse areas of data

In the absence of prior contour polyline information, although various normal estimation methods have been proposed, it is still difficult to accurately estimate the real normals of cross-contour polylines due to the sparsity of sampling data.

In this work, we consider transforming the problem of orebody modeling based on non-parallel geological interpretation polylines in geological modeling into the problem of normal estimation of cross-contour polylines. Because geological interpretation polylines produced by geological logging in the process of exploration and production have a complex adjacency relationship, it is generally difficult to satisfy the requirements of orebody modeling using the normals obtained by the traditional normal estimation methods. On the one hand, the normal estimation of cross-contour polylines should consider more topological adjacency than that of parallel contours to obtain more accurate normal estimation results. On the other hand, the traditional method of normal estimation of contour polylines based on a single polyline is not suitable for the normal estimation of cross-contour polylines. Moreover, it is also difficult to estimate the normals of cross-contour polylines by the manual interaction method. As a result, it is necessary to provide an automatic normal estimation method of cross-contour polylines to improve the efficiency of implicit orebody modeling.

Because the adjacent segments at an intersection point have strong topological correlation characteristics and satisfy the conditions of the local fitting plane, we considered

using the least square plane fitting method based on principal component analysis (PCA) to estimate the normal of cross-contour polylines. Inspired by the point cloud normal estimation strategy, we consider the normalized line segments endpoints in the intersection point neighborhood as a point cloud and use the PCA method to estimate the normal of the intersection point. Then, according to the geometric relationship, the normals of the beginning and the end segments of the corresponding polyline are estimated. Next, the normals of middle line segments are estimated using the linear interpolation method. After that, reorient the normal direction of all cross-contour polylines through the method based on the normal propagation.

After determining the normals of the contour polylines, we can use the implicit modeling [21–24] method based on radial basis function interpolation [25–32] to carry out 3D orebody modeling. To interpolate the directed contour polylines, firstly, we obtain the on-surface interpolation constraints by sampling contour polylines. Secondly, the off-surface interpolation constraints are constructed according to the idea of signed distance field [33–37] and the way of adaptively offsetting sampling points. Finally, the radial basis function interpolation method is used to interpolate the corresponding constraints to obtain the implicit function representing the orebody model, and the implicit surface reconstruction method [38–43] is used to extract the isosurfaces.

The method we propose is based on the basic assumption that several segments connected to the intersection points of contour polylines satisfy the characteristics of local curvature. It is suitable for both the planar cross-contour polylines and non-planar cross-contour polylines. However, it requires that line segments must strictly intersect at points and can be interrupted at the points. In summary, we present an innovative method for normal estimation of cross-contour polylines and an innovative method for reorienting normal directions.

2. Method

The basic idea of normal estimation of cross-contour polylines in orebody modeling is to construct a virtual network using the orebody cross contour polylines obtained by geological interpretation, and then the normals of the cross-contour polylines can be estimated and reoriented. Figure 1 shows the overall process of the method, which mainly consists of the following three steps:

- (1) According to the orebody cross-contour polylines obtained by geological interpretation in the process of exploration and production, a virtual network is constructed to determine the topological adjacency relationship of contour polylines and the corresponding intersection points. Next, the endpoints of the normalized line segments connected to the intersection points are regarded as point clouds, and the normals at the intersection points are obtained by using the method based on local plane fitting. Then, according to the spatial geometric relationship, the estimated normals of the beginning and the end segments of the contour polylines can be obtained. At last, the normals of the middle segments of the contour polylines can be estimated by the way of linear interpolation. Based on the above steps, we can obtain the unoriented normals of all contour polylines.
- (2) The contour interpolation method requires that the normal directions of the contour polylines be consistent, so this stage is divided into two steps to realize it. Firstly, starting from one end of a contour polyline, the judgment vectors are obtained by cross multiplication of the direction vector projection and the normal vector of adjacent line segments. Then the point multiplication result of judgment vectors is used to adjust the normal directions of segments. Therefore, we can reorient the normal directions of a single contour polyline. Secondly, we can flip the normals of the polylines by comparing the normals of the adjacent line segments at each intersection points. Finally, the normal propagation method can be used to realize the normal direction reorientation of all polylines.

- (3) Calculate the implicit function representing the orebody model by using the radial basis function interpolation method and extract the isosurface by the marching cubes (MC) method. Finally, we can obtain the orebody model.

Figure 1. Overall flow chart.

2.1. Normal Estimation

According to the above general idea, to estimate the normal vectors of the cross-contour polylines, first, we should estimate the normals of the intersection points. The foundation of this step is to build a virtual network based on topological adjacency of the orebody contour polylines.

2.1.1. Construct the Network of Cross Contours

To construct a virtual network with topological adjacency relationship, we should preprocess the cross-contour polylines of the orebody (the polylines should be broken at intersection points exactly). The constructed virtual network is shown in Figure 2.

Figure 2. Construct the virtual network with topological adjacency relationship: (a) the cross-contour polylines of the orebody; (b) the constructed virtual network.

2.1.2. Normal Estimation of the Intersection Points

Based on the normal estimation strategy of point clouds, we believe that the endpoints of the beginning segments connected to the intersection point of the contour polylines can

be regarded as a point cloud. Based on the method of local plane fitting, the PCA method is used to estimate the normal of the intersection point. However, due to the different lengths of the line segments connected to the intersection point, they have different effect on the normal estimation. Normalization is carried out in this paper to ensure that the contribution of each segment for the normal estimation is same. The normalized segments can be obtained by clipping the original adjacent line segments using the unit length. And the least square plane of the intersection point can be solved by the endpoints of the normalized line segments (normalized neighborhood points). Based on the fitted plane, we can estimate the normal $\mathbf{n}_p$ of the intersection point:

$$\min_{c, \mathbf{n}_p, \|\mathbf{n}_p\|=1} \sum_{i=1}^n \left((\mathbf{X}_i - \mathbf{c})^T \mathbf{n}_p \right)^2 \quad (1)$$

Here, n represents the number of the normalized neighborhood points; $\mathbf{n}_p$ represents the estimated normal of the intersection point; $\mathbf{X}_i$ represents a normalized neighborhood point. In general, the point c can be calculated as the center of the normalized neighborhood points:

$$\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i \quad (2)$$

$$\mathbf{y}_i = \mathbf{X}_i - \mathbf{c} \quad (3)$$

After the above steps, the objective function is:

$$\begin{aligned} \min_{\|\mathbf{n}_p\|=1} \sum_{i=1}^n (\mathbf{y}_i^T \mathbf{n}_p)^2 &= \min_{\mathbf{n}_p^T \mathbf{n}_p=1} \sum_{i=1}^n \mathbf{n}_p^T \mathbf{y}_i \mathbf{y}_i^T \mathbf{n}_p = \min_{\mathbf{n}_p^T \mathbf{n}_p=1} \mathbf{n}_p^T \left(\sum_{i=1}^n \mathbf{y}_i \mathbf{y}_i^T \right) \mathbf{n}_p \\ &= \min_{\mathbf{n}_p^T \mathbf{n}_p=1} \mathbf{n}_p^T (\mathbf{Y} \mathbf{Y}^T) \mathbf{n}_p \end{aligned} \quad (4)$$

$$\mathbf{S} = (\mathbf{Y} \mathbf{Y}^T) \quad (5)$$

The final form of the objective function is:

$$f(\mathbf{n}_p) = \mathbf{n}_p^T \mathbf{S} \mathbf{n}_p \quad (6)$$

$$\min(f(\mathbf{n}_p)) \quad (7)$$

$$s.t. \mathbf{n}_p^T \mathbf{n}_p = 1 \quad (8)$$

The eigenvalue decomposition of matrix $\mathbf{S}$ is carried out, and the eigenvector corresponding to the smallest eigenvalue is the normal vector $\mathbf{n}_p$ of the intersection point to be solved, as shown in Figure 3.

Figure 3. Normal estimation of the intersection point.

When an intersection point is only connected to two polylines (that is, the segments connected with the intersection point are not collinear), there are only two normalized neighborhood points, and the least squares plane cannot be solved. In this case, the intersection point and two normalized neighborhood points may be combined to solve the least square plane, and then the normal of the intersection point can be estimated.

It is worth noting that in some special cases, there is no unique solution to the equation and it is impossible to estimate the normal of the intersection point. For example, when an intersection point is only connected with two polylines and the two segments connected with the intersection point are collinear, the least squares plane cannot be solved.

2.1.3. Normal Estimation of the Beginning and the End Segments

We use the normal n_p of the intersection point estimated in the previous steps to estimate the normals of the beginning and the end segments of a polyline, as shown in Figure 4. Set the direction vector of the segment connected to the intersection point as l_1 . If two vectors are perpendicular, their vector product is zero. Based on this fact, calculate normal vector n_f of the plane where l_1 and n_p are located:

$$n_f = n_p \times l_1 \quad (9)$$

Figure 4. Normal estimation of the beginning and the end segments.

Then, we calculate the normal vector n_1 of the plane where l_1 and n_f are located. n_1 is the estimated normal of the segment connected to the intersection point of the polyline:

$$n_1 = l_1 \times n_f \quad (10)$$

2.2. Linear Interpolation

After the above steps, we can estimate the normals of all segments connected to the intersection points. Based on the estimated normals, we can interpolate the normals of the middle segments of the polyline using the linear interpolation method, as shown in Figure 5.

Figure 5. Linear interpolation. (a,b) show the comparison before and after linear interpolation.

For most contour polylines connected with the intersection points, we can obtain the estimated normal of their beginning and the end segments. In this case, we can use the linear interpolation method based on the normals of the beginning and the end segments to interpolate the normals of the middle segments of the polyline.

The normal vectors of the middle segments are affected by both the normal vectors of the beginning and the end segments and the direction vectors of themselves. Therefore, in this paper, based on the normal vectors of the beginning and the end segments, we interpolate the tangent vectors instead of the normal vectors of the line segments. Set the direction vector and normal vector of one segment connected to the intersection point of the polyline as l_1 and n_1 . Then, we can calculate the tangent vector of the line segment:

$$t_1 = l_1 \times n_1 \quad (11)$$

Setting the direction vector and normal vector of the other segment connected to the intersection point of the polyline as l_2 and n_2 , then, we can calculate the tangent vector of line segment:

$$t_2 = l_2 \times n_2 \quad (12)$$

If:

$$t_1 \cdot t_2 < 0 \quad (13)$$

then we flip the tangent vector t_2 to ensure the reliability of linear interpolation results.

Set the tangent vector of segment i of the polyline as t_i , direction vector as l_i , normal vector as n_i . d_1 represents the online distance between t_i and t_1 . d_2 represents the online distance between t_i and t_2 :

$$t_i = t_1 \times (1 - m) + t_2 \times m \quad (14)$$

where:

$$m = \frac{d_1}{d_1 + d_2} \quad (15)$$

Then, based on the tangent vector, we can calculate the normal vector:

$$n_i = l_i \times t_i \quad (16)$$

The normal direction of the whole polyline can be interpolated one by one. It is worth noting that for the polylines near the boundary of the cross-contour of the orebody and the polylines at some special positions, which have only one segment connected to the intersection point, they have only one estimated normal. For these polylines, the normals of the middle segments should be interpolated in another way.

Firstly, based on the known normal vector of the segment connected to the intersection point, calculate the tangent vector. Set the direction vector of this segment as l_1 , normal vector as n_1 . Then, we can calculate the tangent vector t :

$$t = l_1 \times n_1 \quad (17)$$

Secondly, interpolation is carried out with 1 as the weight. It is assumed that the tangent vectors of all segments of the whole polyline are t . Set the direction vector of segment i of the polyline as l_i , normal vector as n_i :

$$n_i = l_i \times t \quad (18)$$

2.3. Reorient Normal Directions

The normal vectors of all line segments can be estimated based on the above steps, but their directions may be inconsistent. To interpolate the contour polylines using the implicit modeling method, it is necessary to reorient the normal directions. Our idea is to make the normal directions of an initial polyline consistent first. Then the normal directions of multiple polylines adjacent to the intersection are reoriented. Finally, the normal directions of all polylines can be reoriented by using the normal propagation method.

2.3.1. Reorient Normal Directions of a Single Contour Polyline

Set the segments of a polyline in the cross-contour network as $Line_i$, $i = 1, 2, \dots, n$ direction vector of them as l_i , $i = 1, 2, \dots, n$ normal vector as n_i , $i = 1, 2, \dots, n$. The tangent plane of $Line_i$ is α . The tangent plane of $Line_{i+1}$ is β . Plane γ is perpendicular to α and β , and the intersection point of $Line_i$ and $Line_{i+1}$ is on it, as shown in Figure 6.

Figure 6. Reorient normal directions of a single contour polyline.

The projection vector l'_i and l'_{i+1} are calculated by projecting the vector l_i and l_{i+1} into the plane γ . Then, we can calculate the vector t'_i and t'_{i+1} as judgment vectors:

$$t'_i = l'_i \times n_i \quad (19)$$

$$t'_{i+1} = l'_{i+1} \times n_{i+1} \quad (20)$$

If:

$$t'_i \cdot t'_{i+1} < 0 \quad (21)$$

then we flip n_{i+1} .

If:

$$t'_i \cdot t'_{i+1} \geq 0 \quad (22)$$

then we retain the original orientation.

It should be noted that when $Line_i$ and $Line_{i+1}$ are coplanar, the above method is not applicable. In this case, the normal vectors n_i and n_{i+1} of line segments are directly used as the judgment vectors. Then we can calculate the sign of point multiplication to judge whether the normal directions are consistent.

2.3.2. Reorient Normal Directions of Multiple Contour Polyline

Based on the above steps, the normal directions of single one polyline are consistent, but the normal directions may not be consistent between polylines. By calculating the sign of the normal vectors of the intersection point and the connected segments, the orientations of the normal vectors can be reoriented, as shown in Figure 7.

Set the estimated normal vector at an intersection in the cross-contour network as n_p , the estimated normal vector of the segments connected to it as n_i , $i = 1, 2, \dots, n$:

If:

$$n_p \cdot n_i < 0 \quad (23)$$

then we flip the normal vectors of the polyline where the segment i is located.

If:

$$n_p \cdot n_i \geq 0 \quad (24)$$

then we retain the original orientation.

Figure 7. Reoriented normal directions of multiple contour polylines. (a,b) show the comparison before and after reorienting normal directions of multiple contour polylines.

Based on the network of the cross-contour polylines, the normal propagation method is used to reorient normal directions of all polylines. To accelerate the speed of the reorientation process, the topological adjacency relationship between contour polylines and their intersection points in the network can be constructed in advance. Similar to the process of normal propagation of the point cloud, the breadth-first search (BFS) method is used to traverse each intersection point. According to the above correction formula, reorient the normal directions of the polylines connected to the intersection points. After determining the normals of the contour polylines, we can directly use the implicit modeling method based on RBF to reconstruct the orebody automatically.

2.4. Experiment Information

We implemented the algorithm of the proposed method using the C++ programming language and tested it on a Windows 64-bit computer equipped with a 3.70 GHz Intel (R) Core (TM) i7-8700U processor, 32 GB RAM, and an NVIDIA GeForce RTX 2080 GPU.

To make the experiments scientific and reasonable, we extract the experimental data from the real data obtained in the process of exploration and production of an actual metal mine. The experimental data is divided into four groups with different numbers of intersections and polylines. In order to verify the reliability of the proposed method, the experiment results of the first set data are compared with the results of explicit modeling using the same data.

3. Results and Discussion

We test the normal estimation method on four sets of cross-contour polylines data. These examples are taken from real geological data and obtained from geological logging during the process of exploration and production. The following experimental results clearly show the process of normal estimation, normal reorientation and implicit modeling of the original cross contour polylines step by step, as shown in Figures 8–12.

3.1. Examples

Figure 8 shows the normal estimation process and results as well as modeling results of the first set of cross-contour polylines. Figure 8a is the original cross-contour polylines of the orebody. After processing, a network Figure 8b with the topological adjacency relationship is constructed. The normals of each intersection point are estimated by using the method based on local plane fitting. As shown in Figure 8c, the green lines represent the estimated normals of intersection points. According to the spatial geometric relationship, the beginning and the end segments of the contour polylines can be estimated. As shown in Figure 8d, the red surface is the tangent plane of the line segment, the green surface and the blue surface represent the normals of the line segment. The blue surface points to the inside of the model and the green points to the opposite. As shown in Figure 8e,

the normals of the middle segments of the contour polylines can be estimated by linear interpolation, but up to now, the normal directions of the segments of a single polyline (such as polyline ④ in Figure 8e) and between polylines (such as polyline ① and ② in Figure 8e, ④ and ⑤ in Figure 8e) are inconsistent. To interpolate the contour polylines using the implicit modeling method, it is necessary to make the polylines' normal directions consistent. Therefore, firstly, we reorient normal directions of the segments of a single polyline, as shown in Figure 8f. It can be seen that the consistency is reoriented for each polyline, but the normal directions between polylines are still inconsistent. Secondly, the normal directions of multiple polylines become consistent by using the method based on the normal propagation method, as shown in Figure 8g. After the above steps, the normal directions of all polylines are consistent. Finally, implicit modeling is carried out, and the result is shown in Figure 8i.

In addition, the model obtained using the method we proposed is compared with the model using the explicit modeling method. As is shown in Figure 9. The comparison result shows that the models obtained by the two methods are similar.

Figure 8. Experimental results of the first set of cross contour polylines (a–i).

Figure 9. Comparison results of models obtained by different methods.

Figure 10. Experimental results of the second set of cross-contour polylines. (a–c) show the process of the normal estimation; (d,e) show the process of the reorientation of normal directions; (f) shows the final modeling result.

Figure 11. Experimental results of the third set of cross-contour polylines. (a–c) show the process of the normal estimation; (d,e) show the process of the reorienting normal directions; (f) shows the final modeling result.

Figure 12. Experimental results of the fourth set of cross-contour polylines. (a–c) show the process of the normal estimation; (d,e) show the process of reorienting the normal directions; (f) shows the final modeling result.

3.2. Performance

The performance of the normal estimation method of cross-contour polylines mainly depends on the number of intersection points and polylines. Table 1 shows the parameters, normal estimation time and modeling time of the cross-contour polylines. It can be found that the more intersection points and cross-contour polylines there be, the longer the normal estimation time will be.

Table 1. Experimental data statistics, including the intersection point number (N_{int}), the polyline number (N_{pl}), the normal estimation time (T_{ne}), the function solving time (T_{fs}) and the overall modeling time (T_{om}).

Examples	N_{int}/n	N_{pl}/n	T_{ne}/ms	T_{fs}/s	T_{om}/s
Figure 8	6	7	<1	0.32	1.07
Figure 10	107	179	1	1.51	5.35
Figure 11	109	172	1	0.96	1.02
Figure 12	139	225	2	1.70	2.88

3.3. Discussion

The above experimental results and data analysis show that even if there are more and more intersection points and polylines in the data, the normal estimation time is still very short and the efficiency is very high. The normal estimation method of cross-contour polylines proposed in this paper has a small computation burden and high operating efficiency. In addition, implicit modeling is carried out based on the estimated normal, and the model results are good.

The main reasons for the small amount of calculation are as follows. Firstly, the number of line segments associated with each intersection is small, so the amount of calculation of solving the equation is small in the process of using the PCA method to estimate the intersection normal. Secondly, the normals of the middle segments are estimated by linear interpolation, so the algorithm is near linear in time complexity. Thirdly, in terms of space complexity, the algorithm only needs to traverse each intersection to reorient the normals. Based on the above reasons, the algorithm has high operating efficiency.

Although an automatic normal estimation method of cross-contour polylines is discussed preliminarily, this method still has several limitations that need to be further studied. Firstly, the method requires that the local fitting surfaces calculated at the intersection points satisfy the geometry shape characteristics of the interpolated model; otherwise, it will affect the accuracy of the normal estimation of the contour polylines near the intersection points. Secondly, when the length of a single contour polyline is large or the shape changes greatly, the reliability of linear interpolation using normal of the beginning and the end segments of contour polylines may be reduced significantly. Because the result of this method relies too much on the estimated normals at the intersection points, the abnormal estimated normals will affect the normal propagation of the topological network. Finally, the method is suitable for the intersecting contour polylines, instead of the sparsely distributed contour polylines. In addition, because of the complexity of cross contour polylines, it is still necessary to manually adjust the abnormal normals of the contour polylines. Therefore, to further improve the accuracy of normal estimation for the complex contour polylines, it is necessary to develop a dynamic normal estimation method. The method allows geological engineers to adjust the normal direction of some segments dynamically according to the prior geological law and uses these correct normals as references to estimate the other normals dynamically.

4. Conclusions

In this work, we propose an implicit modeling method based on the normal estimation of cross-contour polylines. The method is suitable for orebody cross-contour polylines which are strictly intersected. For the polylines with complex topological adjacency relationship, it is difficult to estimate the corresponding normal vectors. The innovation of this paper is that the proposed method can automatically estimate the normals of the cross-contour polylines and reorient them without manual intervention. We solve the problem that it is difficult to construct interpolation constraints in orebody implicit modeling for the complex cross contour polylines. The method greatly improves the modeling efficiency for orebody modeling. Firstly, by constructing the normalized neighborhood points, the normal estimation problem at the intersection points is transformed into the normal estimation of the point cloud. Then the normal vectors of all line segments are estimated step by step according to the geometric relationship between the line segments and intersection points. Secondly, reorient the normal directions of all the contour polylines by using the normal propagation method. Experimental results show that the proposed method has the advantages of a small amount of calculation, high efficiency and strong reliability.

Author Contributions: Data curation, Z.W. and D.Z.; Formal analysis, Z.W. and D.Z.; Methodology, Z.W. and D.Z.; Project administration, D.Z.; Software, Z.W. and D.Z.; Writing—original draft, Z.W.; Writing—review & editing, Z.W., D.Z., Z.L., L.W. and L.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 52104171.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the reviewers for their comments and suggestions to improve the quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baudrillard, J.; Guillon, S.; Chassery, J.-M.; Rombaut, M.; Wang, K. Reconstruction of Piecewise-Explicit Surfaces from Three-Dimensional Polylines and Heightmap Fragments. In Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM), Funchal, Portugal, 17–19 March 2018; pp. 1–27.

2. Ekoule, A.B.; Peyrin, F.C.; Odet, C.L. A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM Trans. Graph.* **1991**, *10*, 182–199. [\[CrossRef\]](#)
3. Jin, B.-X.; Fang, Y.-M.; Song, W.-W. 3D visualization model and key techniques for digital mine. *Trans. Nonferrous Met. Soc. China* **2011**, *21*, S748–S752. [\[CrossRef\]](#)
4. Barthe, L.; Dodgson, N.A.; Sabin, M.A.; Wyvill, B.; Gaildrat, V. Two-dimensional Potential Fields for Advanced Implicit Modeling Operators. *Comput. Graph. Forum* **2010**, *22*, 23–33. [\[CrossRef\]](#)
5. Cuomo, S.; Galletti, A.; Giunta, G.; Marcellino, L. Reconstruction of implicit curves and surfaces via RBF interpolation. *Appl. Numer. Math.* **2017**, *116*, 157–171. [\[CrossRef\]](#)
6. Hillier, M.J.; Schetselaar, E.M.; Kemp, E.D.; Perron, G. Three-Dimensional Modelling of Geological Surfaces Using Generalized Interpolation with Radial Basis Functions. *Math. Geosci.* **2014**, *46*, 931–953. [\[CrossRef\]](#)
7. Turk, G.; O'Brien, J.F. Modelling with implicit surfaces that interpolate. *ACM Trans. Graph.* **2002**, *21*, 855–873. [\[CrossRef\]](#)
8. Zhong, D.-Y.; Wang, L.-G.; Bi, L.; Jia, M.-T. Implicit modeling of complex orebody with constraints of geological rules. *Trans. Nonferrous Met. Soc. China* **2019**, *29*, 2392–2399. [\[CrossRef\]](#)
9. Hoppe, H.; Deroose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Surface reconstruction from unorganized points. *ACM SIGGRAPH Comput. Graph.* **1992**, *26*, 71–78. [\[CrossRef\]](#)
10. Amenta, N.; Bern, M. Surface reconstruction by Voronoi filtering. *Discret. Comput. Geom.* **1999**, *22*, 481–504. [\[CrossRef\]](#)
11. Leroy, A.M.; Rousseeuw, P.J. *Robust Regression and Outlier Detection*; John Wiley & Sons, Inc.: River St Hoboken, NJ, USA, 1987.
12. Shi, T.; Zhong, D.; Wang, L. Geological Modeling Method Based on the Normal Dynamic Estimation of Sparse Point Clouds. *Mathematics* **2021**, *9*, 1819. [\[CrossRef\]](#)
13. Xu, C.; Dowd, P.A. Optimal construction and visualisation of geological structures. *Comput. Geosci.* **2003**, *29*, 761–773. [\[CrossRef\]](#)
14. Shao, C.; Bousseau, A.; Sheffer, A.; Singh, K. CrossShade. *ACM Trans. Graph.* **2012**, *31*, 1–11. [\[CrossRef\]](#)
15. Lee, K.; Baek, S.-Y.; Kim, T.; Park, J. An algorithm for estimating surface normal from its boundary curves. *J. Comput. Des. Eng.* **2015**, *2*, 67–72.
16. Kniss, J.; Kindlmann, G.; Hansen, C. Multidimensional transfer functions for interactive volume rendering. *IEEE Trans. Vis. Comput. Graph.* **2002**, *8*, 270–285. [\[CrossRef\]](#)
17. Ijiri, T.; Yoshizawa, S.; Sato, Y.; Ito, M.; Yokota, H. Bilateral Hermite Radial Basis Functions for Contour-based Volume Segmentation. *Comput. Graph. Forum* **2013**, *32*, 123–132. [\[CrossRef\]](#)
18. Bi, L.; Liu, X.; Chen, X.; Zhu, Z. An Automatic 3D Modeling Method Based on Orebody Contours. *Geomat. Inf. Ence Wuhan Univ.* **2016**, *41*, 1359–1365.
19. Carr, J.C.; Beatson, R.K.; Cherrie, J.B.; Mitchell, T.J.; Fright, W.R.; McCallum, B.C.; Evans, T.R. Reconstruction and representation of 3D objects with radial basis functions. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001.
20. Zhong, D.-Y.; Wang, L.-G.; Jia, M.-T.; Bi, L.; Zhang, J. Orebody Modeling from Non-Parallel Cross Sections with Geometry Constraints. *Minerals* **2019**, *9*, 229. [\[CrossRef\]](#)
21. Hua, J.; Qin, H.; Ieee, I. Dynamic implicit solids with constraints for haptic sculpting. In Proceedings of the 4th International Conference on Shape Modeling, Banff, AB, Canada, 17–22 May 2002; pp. 119–274.
22. Jin, X.; Tai, C.-L.; Zhang, H. Implicit modeling from polygon soup using convolution. *Vis. Comput.* **2009**, *25*, 279–288. [\[CrossRef\]](#)
23. Lipus, B.; Guid, N. A new implicit blending technique for volumetric modelling. *Vis. Comput.* **2005**, *21*, 83–91. [\[CrossRef\]](#)
24. Zhong, D.-Y.; Wang, L.-G.; Wang, J.-M. Combination Constraints of Multiple Fields for Implicit Modeling of Ore Bodies. *Appl. Sci.* **2021**, *11*, 1321. [\[CrossRef\]](#)
25. Buhmann, M.D. A new class of radial basis functions with compact support. *Math. Comput.* **2001**, *70*, 307–318. [\[CrossRef\]](#)
26. Iske, A. European summer school on principles of multiresolution in geometric modelling. In *Scattered Data Modelling Using Radial Basis Functions*; Munich University of Technology: Munich, Germany, 2002; pp. 205–242.
27. Larsson, E.; Fornberg, B. Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Comput. Math. Appl.* **2005**, *49*, 103–130. [\[CrossRef\]](#)
28. Macedo, I.; Gois, J.P.; Velho, L. Hermite Interpolation of Implicit Surfaces with Radial Basis Functions. In Proceedings of the 22nd Brazilian Symposium on Computer Graphics and Image Processing, Rio de Janeiro, Brazil, 11–15 October 2009; pp. 1–8.
29. Zhang, M. Radial basis function interpolation in Sobolev spaces and its applications. *J. Comput. Math.* **2007**, *25*, 201–210.
30. Zhu, S.; Wathen, A.J. Convexity and Solvability for Compactly Supported Radial Basis Functions with Different Shapes. *J. Sci. Comput.* **2015**, *63*, 862–884. [\[CrossRef\]](#)
31. Stewart, M.; de Lacey, J.; Hodkiewicz, P.F.; Lane, R. Grade Estimation from Radial Basis Functions—How Does it Compare with Conventional Geostatistical Estimation? In Proceedings of the Ninth International Mining Geology Conference, Adelaide, SA, Australia, 18–20 August 2014.
32. Zerzour, O.; Gadri, L.; Hadji, R.; Mebrouk, F.; Hamed, Y. Semi-variograms and kriging techniques in iron ore reserve categorization: Application at Jebel Wenza deposit. *Arab. J. Geosci.* **2020**, *13*, 820. [\[CrossRef\]](#)
33. Chang, B.; Cha, D.; Ihm, I. Computing local signed distance fields for large polygonal models. *Comput. Graph. Forum* **2008**, *27*, 799–806. [\[CrossRef\]](#)

34. Sagawa, R.; Ikeuchi, K. Taking consensus of signed distance field for complementing unobservable surface. In Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling (3-DIM 2003), Banff, BA, Canada, 6–10 October 2003; pp. 410–417.
35. Sanchez, M.; Fryazinov, O.; Fayolle, P.-A.; Pasko, A. Convolution Filtering of Continuous Signed Distance Fields for Polygonal Meshes. *Comput. Graph. Forum* **2015**, *34*, 277–288. [[CrossRef](#)]
36. Tang, Y.; Feng, J. Multi-scale surface reconstruction based on a curvature-adaptive signed distance field. *Comput. Graph. UK* **2018**, *70*, 28–38. [[CrossRef](#)]
37. Wu, J.H.; Kobbelt, L. Piecewise linear approximation of signed distance fields. In Proceedings of the 8th Workshop on Vision, Modeling, and Visualization (VMV 2003), Munich, Germany, 19–21 November 2003; pp. 513–520.
38. Akkouché, S.; Galin, E. Implicit surface reconstruction from contours. *Vis. Comput.* **2004**, *20*, 392–401. [[CrossRef](#)]
39. Lin, Y.; Chen, C.; Song, M.; Liu, Z. Dual-RBF based surface reconstruction. *Vis. Comput.* **2009**, *25*, 599–607. [[CrossRef](#)]
40. Liu, Y.; Song, Y.; Yang, Z.; Deng, J. Implicit surface reconstruction with total variation regularization. *Comput. Aided Geom. Design* **2017**, *52*, 135–153. [[CrossRef](#)]
41. Wang, C.-S.; Chang, T.-R.; Lin, M.-C. Reconstruction and Representation for 3D Implicit Surfaces. In Proceedings of the International Conference on Computer Science and Education (CSE 2011), Qingdao, China, 9–10 July 2011; pp. 364–372.
42. Yang, Z.W.; Wu, C.L.; Deng, J.S.; Chen, F.L. Specification of initial shapes for dynamic implicit curve/surface reconstruction. *J. Comput. Sci. Technol.* **2006**, *21*, 249–254. [[CrossRef](#)]
43. Zagorchev, L.G.; Goshtasby, A.A. A Curvature-Adaptive Implicit Surface Reconstruction for Irregularly Spaced Points. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1460–1473. [[CrossRef](#)] [[PubMed](#)]