applied sciences

MDPI

Article

Fast Implicit Surface Reconstruction for the Radial Basis Functions Interpolant

Deyun Zhong [1,2,*], Ju Zhang [1,2] and Liguan Wang [1,2]

[1] School of Resources and Safety Engineering, Central South University, Changsha 410083, China; zhangju@csu.edu.cn (J.Z.); liguan_wang@csu.edu.cn (L.W.)
[2] Research Center of Digital Mine, Central South University, Changsha 410083, China
* Correspondence: deyizhiyun@163.com

Received: 14 November 2019; Accepted: 3 December 2019; Published: 6 December 2019

Abstract: In this paper we improve an efficient implicit surface reconstruction method based on the surface following method for the radial basis functions interpolant. The method balances the reconstruction efficiency and the evaluation efficiency in the process of surface following. The growing strategy of the surface following method combines both the evaluation and reconstruction processes. Based on the analysis of the black-box fast multipole method (FMM) operations, we improve the FMM procedures for single point evaluation. The goal is to ensure that one point evaluation of the method obtains an optimum efficiency, so that it can be efficiently applied to the voxel growing method. Combined with the single point FMM, we improve the voxel growing method without manually specifying the seed points, and the leaf growing method is developed to avoid a mass of redundant computation. It ensures a smaller number of evaluation points and a higher evaluation efficiency in surface following. The numerical results of several data sets showed the reliability and performance of the efficient implicit surface reconstruction method. Compared with the existing methods, the improved method performs a better time and space efficiency.

Keywords: surface reconstruction; radial basis functions; fast multipole method; implicit modeling; marching cubes

1. Introduction

The reconstruction of implicit surfaces is commonly used in computer graphics, scientific data visualization, geometry processing, reverse engineering, etc. The reconstruction efficiency has always been a difficult problem that hinders the wide application of implicit surface (e.g., real-time shape editing). This work tries to extract the isosurface from the zero level set of an implicit function efficiently.

As an important interpolation method, the radial basis functions (RBFs) interpolant [1–3] is widely used in different applications. In numerical analysis, the RBF interpolant plays a key role in solving partial differential equations (PDEs) [4]. In nonlinear time series analysis, it is used for the approximation of the flow in the phase space [5,6]. In spatial interpolation, it is used to interpolate different geological data [7] (e.g., grade, fault and formation).

In implicit modeling, given a set of unorganized points with normals (Hermite data), the implicit surface reconstruction methods interpolate the unknown surface using an implicit function. Based on the idea of signed distance field interpolation [8], the radial basis functions interpolant can be used to implicitly represent the implicit surface satisfying the Hermite data. The unorganized points are used to construct on-surface constraints, and the offset points generated by projecting along the normals are used to construct off-surface constraints.

The implicit surface reconstruction involves two main processes, namely evaluation and reconstruction. As a classical method of surface reconstruction, the marching cubes (MC) method [9,10] extracts the isosurface by triangulating the cubes in the discretized space. The method evaluates the implicit function values of all points sampled on the regular voxel points. It takes too much time to evaluate the invalid voxels that do not intersect the isosurface. The improved surface following method [11,12] minimizes the number of evaluation points by tracking voxels intersecting the isosurface.

For the RBF interpolant with large numbers of constraints, both the evaluation and reconstruction processes should be combined to accelerate the reconstruction of the implicit surface. The fast multipole method (FMM) [13,14] is performed to efficiently calculate the sums of pairwise interactions between source points. However, different from the marching cubes method, the evaluation points are determined dynamically by tracking seed voxels for the surface following method. Therefore, on the one hand, the original FMM should be improved to efficiently evaluate the points one by one. On the other hand, to avoid the repeat computation of the coefficients of the FMM, the evaluation points in the same leaf cell of the FMM tree should be evaluated first in the reconstruction process.

According to the above analysis, the minimum evaluation points and the optimization of single point evaluation should be combined to improve the efficiency of implicit surface reconstruction. In this paper, we improve an efficient implicit surface reconstruction method for the RBF interpolant. The method balances the reconstruction efficiency and the evaluation efficiency in the process of surface following. The goal is to ensure that one point evaluation of the method obtains a similar efficiency to the original FMM, so that it can be efficiently applied to the voxel growing method. To efficiently evaluate the field points one by one in reconstruction, we improve the FMM procedures for single point evaluation. The numerical results of several data sets showed the reliability and performance of the efficient implicit surface reconstruction method.

2. Related Works

We briefly review the two bodies of work that are close to ours, namely radial basis functions and implicit surface reconstruction.

There are many implicit functions (e.g., Poisson function [15,16], the Discrete Smooth Interpolation function [17,18]) used to represent implicit surfaces. For the radial basis functions interpolant, the evaluation process is to compute the sums of the linear combinations of kernel functions. The common method for fast evaluation is the fast multipole method (FMM). The basic idea is to expand the radial basis function (RBF) kernels via low-rank approximations, and neglect the far field expansion within a given precision. At present, a number of evaluation methods have been proposed to expand different types of kernels in an analytic way, such as thin-plate splines [19], Gaussian kernel [20] and polyharmonic splines [21]. Recently, some general FMM methods are developed to evaluate kernels without the implementation of multipole expansions, including the kernel independent FMM [22] and the black-box FMM [23]. Besides the global evaluation methods, the local evaluation methods using compactly supported radial basis functions (CSRBF) are also a feasible way of acceleration. To balance the efficiency and effect, the local evaluation methods require the selection of adaptive supported radii for different interpolation centers. Based upon the comparison of a quantitative measure of approximation error, Zhang et al. [24] proposed an adaptive radial basis functions interpolation method using an error indicator.

The reconstruction process usually has nothing to do with the evaluation process. Some of the implicit surface reconstruction methods only need to know the implicit function values at any given position, such as the well-known marching cubes and marching tetrahedra algorithms. At present, many extensions of marching cubes have been proposed, such as dual marching cubes [25], adaptive marching cubes [26], extended marching cubes [27] and multiple material marching cubes [28]. To track the zero level sets of the implicit surface directly, some surface following methods are developed based on a greedy growing idea. Lee et al. [11] presented the growing-cube algorithm to track the neighboring cells with zero values based on the marching cubes algorithm.

Note that the combination of some additional conditions (e.g., the Hermite data of the implicit function) is useful to improve the mesh quality and minimize the number of evaluations.

Treece et al. [29] presented the regularized marching tetrahedra algorithm to reconstruct meshes with fewer triangles and better aspect ratios. On purely the reconstruction process, the surface following method with mesh optimization strategy is an optimum scheme for implicit surface reconstruction. In this work, we focus on improving the reconstruction efficiency, combining both the evaluation and reconstruction processes.

3. Implicit Function

In implicit modeling, the implicit surface S is defined as the zero level set $\left\{ \boldsymbol{x} \middle| f(x,y,z) = 0 \right\}$ of the implicit function $f(\boldsymbol{x})$. To reconstruct the implicit surface S, the implicit function $f(\boldsymbol{x})$ should be first obtained by interpolating the domain constraints

$$f(\boldsymbol{x}_i) = f_i, \quad i = 1, 2, \ldots, N \tag{1}$$

where f_i are function values of the geometry domain and N is the number of constraints. These interpolation constraints are usually constructed from a set of sampling points.

For the radial basis functions interpolant, the implicit function $f(\boldsymbol{x})$ has the form [3]

$$f(\boldsymbol{x}) = f_K(\boldsymbol{x}) + p(\boldsymbol{x}) = \sum_{j=1}^{N} \omega_j K\left(\boldsymbol{x}, \boldsymbol{y}_j\right) + p(\boldsymbol{x}) \tag{2}$$

where $\boldsymbol{x} = (x, y, z)$, $K\left(\boldsymbol{x}, \boldsymbol{y}_j\right)$ is a kind of globally supported radial basis functions and $p(\boldsymbol{x})$ are low-order polynomials. The unknown coefficients ω_i can be determined by solving the linear system combined by the domain constraints. The first variable $\boldsymbol{x}$ in $K(\boldsymbol{x}, \boldsymbol{y})$ is viewed as an evaluation point (or target point), and the second variable $\boldsymbol{y}$ is viewed as a source point.

Taking the polynomial part $p(\boldsymbol{x}) = c_1 + c_2 x + c_3 y + c_4 z$ as an example, the smoothest interpolant satisfies the orthogonality conditions

$$\sum_{i=1}^{N} \omega_i = \sum_{i=1}^{N} \omega_i x_i = \sum_{i=1}^{N} \omega_i y_i = \sum_{i=1}^{N} \omega_i z_i = 0 \tag{3}$$

These orthogonality conditions, along with the domain conditions, lead to a linear system. The matrix form of the linear system can be written as

$$\begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,N} & 1 & x_1 & y_1 & z_1 \\ K_{2,1} & K_{2,2} & \cdots & K_{2,N} & 1 & x_2 & y_2 & z_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{N,1} & K_{N,2} & \cdots & K_{N,N} & 1 & x_N & y_N & z_N \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ x_1 & x_2 & \cdots & x_N & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \cdots & y_N & 0 & 0 & 0 & 0 \\ z_1 & z_2 & \cdots & z_N & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $K_{i,j} = K\left(\boldsymbol{x}_i, \boldsymbol{y}_j\right)$, and the unknown coefficients ω_i and c_i can be determined by solving the equation.

There are several ways (e.g., Fast RBF method [30,31]) to efficiently solve the interpolation equation for the RBF interpolant. In this paper, we focus more on the efficient reconstruction of the implicit surface. As mentioned earlier, the evaluation process and the reconstruction process are combined to improve the reconstruction efficiency.

4. Fast Evaluation

4.1. Black-Box FMM

To efficiently evaluate the function values of the RBF implicit function $f(\boldsymbol{x})$, the fast multipole method is used to compute the sums of RBFs based on the low-rank approximations. Taking the 1-D black-box FMM [23] as an example, the background of the kernel independent FMM is reviewed briefly.

Given a set of target points $\{x_i\}$ and source points $\{y_j\}$, the kernel $K(x,y)$ is expanded in a low-rank approximation using Chebyshev polynomials

$$K(x,y) \approx \sum_{n=1}^{p}\sum_{m=1}^{p} K(\bar{x}_n, \bar{y}_m) S_p(\bar{x}_n, x) S_p(\bar{y}_m, y) \tag{4}$$

where p is the expansion order of series, $\{\bar{x}_n\}$ and $\{\bar{y}_m\}$ are Chebyshev nodes, $S_p(\bar{x}_n, x)$ is the interpolation function at node $\bar{x}_n$, and $S_p(\bar{y}_m, y)$ is the interpolation function at node $\bar{y}_m$.

To make the expansions with a rapid convergence, the evaluation points are well separated from the source points (far field). If the evaluation points and source points (near field) are not well separated, the kernels will be evaluated directly. Then the sum of kernels can be efficiently evaluated by changing the order of computation

$$\begin{aligned} f_K(x) &\approx \sum_{j=1}^{N} \omega_j \sum_{n=1}^{p}\sum_{m=1}^{p} K(\bar{x}_n, \bar{y}_m) S_p(\bar{x}_n, x) S_p(\bar{y}_m, y) \\ &\approx \underbrace{\sum_{n=1}^{p} S_p(\bar{x}_n, x)}_{L2P} \underbrace{\sum_{m=1}^{p} K(\bar{x}_n, \bar{y}_m)}_{M2L} \underbrace{\sum_{j=1}^{N^{far}} \omega_j S_p(\bar{y}_m, y_j)}_{P2M} + \underbrace{\sum_{j=1}^{N^{near}} \omega_j K(x, y_j)}_{P2P} \end{aligned}$$

where the number of source points N satisfies $N = N^{far} + N^{near}$. To divide the near field and far field adaptively, the multilevel FMM approach [32] is developed for hierarchical evaluation using the hierarchical data structure, as shown in Figure 1. The source points are divided into rectangle boxes in different levels using the Quadtree data structure. The well-known FMM operations of the near field and far field are divided into several steps, including P2M, M2M, M2L, L2L, L2P and P2P. The operations correspond to a tree structure.

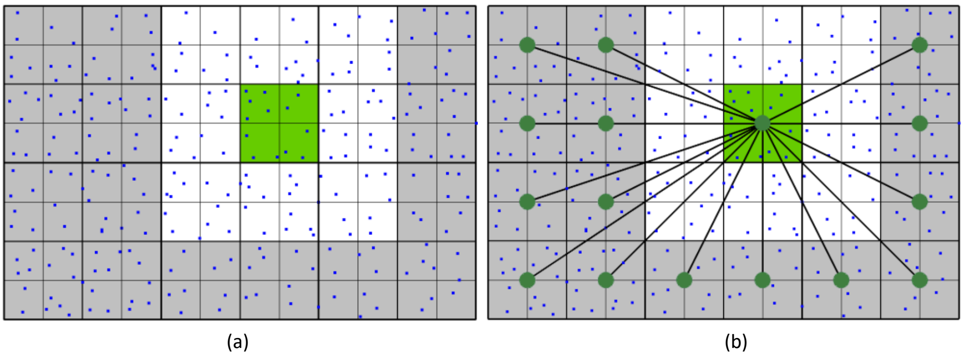

Figure 1. Quadtree decomposition of the multilevel FMM method. (**a**) In the second level of quadtree, the near field of the green box is composed by the white boxes and the green box itself (neighbor list). (**b**) The far field of the green box is composed by the gray boxes (interaction list).

4.2. Single Point FMM

The original FMM evaluates the sums of matrix-vector products. For the surface following method, the evaluation points are determined via the tracking process step by step. To efficiently evaluate the field points one by one, the tree-based procedures of the multilevel approach should be improved for single point evaluation.

For a leaf cell T in the FMM tree (e.g., quad-tree in 2D, octree in 3D), the evaluation of any target point $\boldsymbol{x}_i$ in T is decomposed into three steps based on the black-box FMM operations [23,33].

1. Set up (P2M and M2M). The P2M and M2M operations have no connection with the evaluation points and can be computed first. For any leaf cell T , the multipole moments of T are interpolated from the source points inside T at the Chebyshev nodes $\overline{y}_m^T$

$$M_m^T = \sum_{y_j \in T} \omega_j S_p\left(\overline{y}_m^T, y_j\right),\ m = 1, 2, \ldots, p \tag{5}$$

For any non-leaf cell T, the multipole moments of T are computed by gathering the child cells' multipole expansions interpolated at the Chebyshev nodes $\overline{y}_m^T$

$$M_m^T = \sum_{T'} \sum_{m'} M_{m'}^{T'} S_p(\overline{y}_m^T, \overline{y}_{m'}^{T'}),\ m = 1, 2, \ldots, p \tag{6}$$

where T' are the child cells of T and m' is associated with T'.

2. Translation (M2L and L2L). The translation of far field expansions to near field expansions depends on the evaluation points. For the cells T containing x_i in different octree levels, the local expansions of far field are computed by translating the multipole moments interpolated at the Chebyshev nodes $\overline{x}_n^T$

$$L_n^{T,J} = \sum_{J} \sum_{m} M_m^J K(\overline{x}_n^T, \overline{y}_m^J),\ n = 1, 2, \ldots, p \tag{7}$$

where J are the cells in the interaction list of T. The M2L operation of the computed cell is marked as true to avoid repeated calculation.

For the cells T containing x_i in different octree levels, the local expansions of T are computed by adding the far field interactions and translating the parent cell's local expansions interpolated at the Chebyshev nodes $\overline{x}_n^T$

$$L_n^T = L_n^{T,J} + \sum_{n'} L_{n'}^{T'} S_p\left(\overline{x}_n^T, \overline{x}_{n'}^{T'}\right),\ n = 1, 2, \ldots, p \tag{8}$$

where T' is the parent cell of T and n' is associated with T'. The L2L operation of the computed cell is marked as true to avoid repeated calculation.

3. Evaluation (L2P and P2P). Once the setup and translation steps have been completed, the evaluation for a single point can be computed as the sum of the far field and near field by

$$f_K(x_i) \approx f_K^{far}(x_i) + f_K^{near}(x_i) \tag{9}$$

For any target point x_i in leaf cell T, the interactions of far field are computed by

$$f_K^{far}(x_i) = \sum_{n} L_n^T S_p\left(\overline{x}_n^T, x_i\right) \tag{10}$$

For any target point x_i in leaf cell T, the interactions of near field are directly computed by

$$f_K^{near}(x_i) = \sum_{J'} \sum_{y_j \in J'} \omega_j K\left(x_i, y_j\right) \tag{11}$$

where J' are the cells in the neighbor list of T.

The setup step computes the multipole moments of the cells in each level via the source points. It is only executed once before the evaluation process without considering the evaluation points. The other two steps are computed based on the evaluation points. The translation step computes the conversion of multipole moments and local expansions to the target leaf cell. Therefore, it is only executed once for the evaluation points in the same leaf cell.

For the evaluation points in different leaf cells, the computed M2L and L2L operations are also reusable without repeated calculation. The evaluation step is always executed for each evaluation point.

5. Fast Reconstruction

5.1. Voxel Growing

To polygonize the surface of an implicit function, the geometry space should be discretized to construct the voxels for evaluation, as shown in Figure 2. The classical marching cubes method evaluates all the voxels to extract the zero level set (Figure 3a), while the voxel growing method only evaluates the voxels intersecting the isosurface in the geometry space (Figure 3b). However, it is time consuming to determine the first initial seed point by traversing all the cubes. Moreover, the original voxel growing method cannot ensure to recover the complete model with multiple isosurfaces for missing some possible seed points, as shown in Figure 3b.

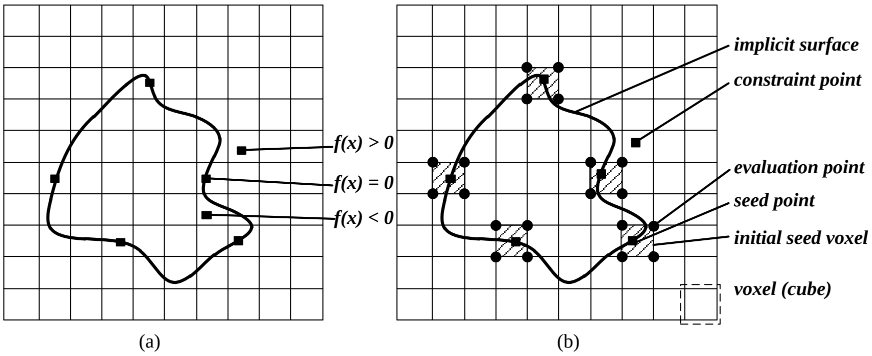

Figure 2. The voxel growing method with multiple seeds. (**a**) The domain constraints with different domain values. The zero level set represents the implicit surface. (**b**) The initial seed voxels selected from the domain constraints with zero values.

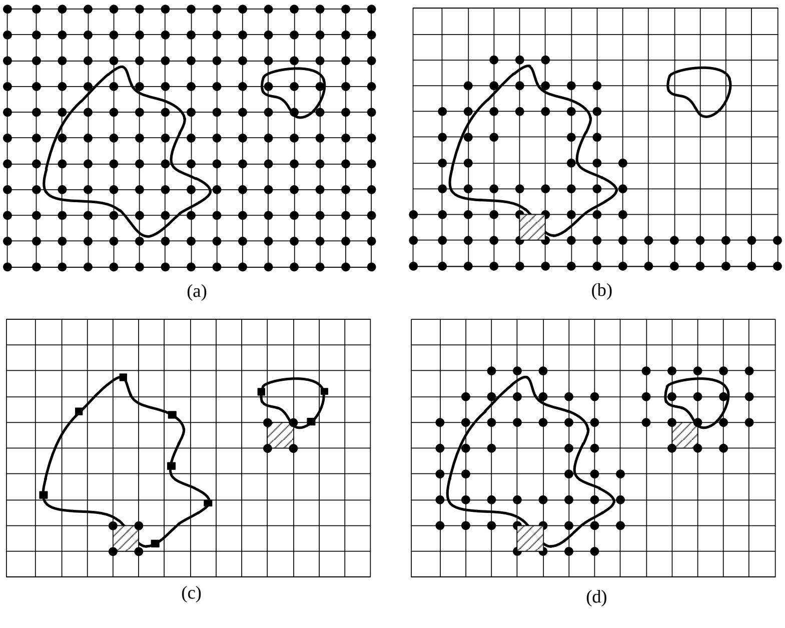

Figure 3. Comparison of evaluation points using different reconstruction methods. (**a**) The evaluation process using the marching cube method. (**b**) The evaluation process using the voxel growing method with one seed. (**c**) The initial seed voxels constructed from constraint points. (**d**) The evaluation process using the voxel growing method with multiple seeds.

In this paper, combined with the single point FMM, we improve the voxel growing method without specifying the seed points manually. For the RBF interpolant, all of the domain constraints with zero values are selected as seed points to avoid missing evaluations, as shown in Figure 2. The voxel containing at least one seed point is viewed as an initial seed voxel. Then the method tracks the voxels intersecting the isosurface using all the initial seed voxels. We construct a voxel growing queue to store the adjacent voxels intersecting the isosurface for tracking. The adjacent voxels intersected the isosurface are appended to the voxel growing queue greedily until there are no new voxels intersected the isosurface, as shown in Figure 3d. Note that the evaluated voxels should be marked to avoid recomputation. Different from the marching cubes method, the voxel growing method evaluates the voxels greedily. Though the voxel growing method reduces the number of evaluations, the evaluation points should be determined via the reconstruction process step by step, which reduces the efficiency of single point evaluation. To improve the evaluation efficiency, the single point FMM is used to evaluate the points one by one. Note that the evaluated points should be marked to avoid recomputation.

The improved voxel growing achieves the same reconstruction effect more efficiently. In the growing process, a voxel growing queue is used to track an isosurface. Reconstructing an isosurface requires only one initial seed voxel. As long as each isosurface has at least one constraint point with zero values, the original model can be completely reconstructed. Generally, the sampling points can meet the requirement in practical applications. It is worth noting that the initial seed voxels in the same isosurface are repeated and should be removed to avoid recomputation. Considering the fact that a voxel growing queue corresponds to an isosurface, an available approach is to remove the repeated initial seed voxels in the same growing queue gradually.

5.2. Leaf Growing

In the voxel growing queue, the numerical results show that assigning different priorities to the pending voxels leads to a different evaluation efficiency. It is because the order of voxel growing affects the single point evaluation efficiency. Compared with the original FMM, the third step of the single point FMM computes the L2P coefficients many times for the evaluation points in the same leaf cell. For the evaluation points in the same leaf cell, the L2P operation interpolates at the same Chebyshev nodes, and the same coefficients can be precomputed by expanding the Chebyshev polynomials.

To avoid a mass of redundant computation, we present the leaf growing method based on the improved voxel growing method. The main difference is that the growing process considers both the evaluation cell of FMM and the reconstruction cell of MC. The evaluation cell of FMM refers to leaf cell, the finest level of octree. The reconstruction cell of MC refers to the voxel or cube. An available approach is to track the implicit surface leaf by leaf instead of voxel by voxel, and then to evaluate all the voxels in the same leaf.

To reduce the number of evaluation points, the improved growing strategy is to track the implicit surface leaf by leaf in global, and voxel by voxel in local. In addition, the voxels in the growing queue are specified a priority via the position in the leaf. There are three positional relationships between the leaf and the voxel, namely interior, intersection and exterior, as shown in Figure 4. In the voxel growing process, a leaf T with the precomputed L2P coefficients is viewed as a current leaf. The precomputed L2P coefficients of T will be stored until there are no voxels inside or intersecting T in the voxel growing queue. To save the memory storage, for the voxels in the growing queue, the voxels inside the current leaf should be evaluated first. Then the voxels intersecting the current leaf will be evaluated and a new current leaf will be obtained.

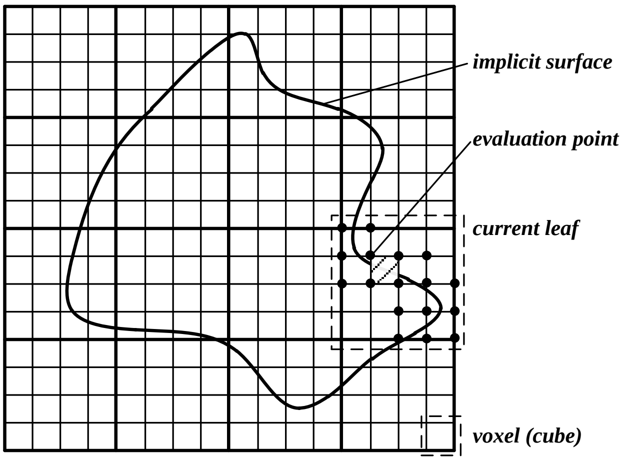

Figure 4. The evaluation process of the leaf growing method. In the voxel growing queue, the voxels in the current leaf cell are evaluated first. The evaluation process is leaf by leaf in global, and voxel by voxel in local.

Given a radial basis functions interpolant $f(x)$, and a desired reconstruction accuracy ϵ, based on the above analysis, a simplified procedure of the fast implicit surface reconstruction method for the RBF interpolant is given below.

Step 1: Space division. Compute the minimum bounding box via the constraint points and divide the space into regular voxels (cubes) via the desired reconstruction accuracy ϵ. Scale the minimum bounding box in a certain ratio.

Step 2: Construct the initial seed voxels. Select the points of the domain constraints with zero values ($f(\boldsymbol{x}_i) = 0$) as the seed points and construct an initial seed voxel set S via the positions of the seed points.

Step 3: Execute the setup step. Compute the multipole moments based on the source points $\{\boldsymbol{y}_j\}$. The source points are the interpolation centers for the RBF interpolant $f(\boldsymbol{x})$.

Step 4: Traverse the initial seed voxel set S. Construct a voxel growing queue G and add an initial seed voxel to G.

Step 5: Traversing the voxel growing queue G, take out a voxel V with the highest priority. Evaluate the points of the voxel V by executing the translation and evaluation steps. Obtain the current leaf cell of the FMM according to the L2P operation. Store the evaluated points and the evaluated voxels to avoid the repeated evaluation.

Step 6: According to the lookup table of the marching cubes method, triangulate the voxel V via the eight evaluated values. Optionally, some additional strategies can be implemented to resolve the ambiguities.

Step 7: Determine the new growing voxels for tracking. Among the six adjacent voxels of V, append the un-evaluated voxels that are intersected the isosurface to the voxel growing queue G.

Step 8: Specify a priority for the new growing voxels via the corresponding position in the current leaf cell of the FMM.

Step 9: Compare the new growing voxels with the initial seed voxel set S, and remove the duplicate initial seed voxels in S.

Step 10: Return to Step 5 until the traversing of the voxel growing queue is finished.

Step 11: Extract an isosurface based on the triangulated voxels.

Step 12: Return to Step 4 until the traversing of the initial seed voxel set is finished.

Step 13: Output the reconstructed isosurfaces.

6. Numerical Results

We implemented the improved implicit surface reconstruction method for the RBF interpolant and tested the performance on a Windows 64-bit PC with 3.00 GHz Intel(R) Core(TM) i5-7400 and 8GB RAM. The ScalFMM library was used to implement the single point FMM for fast evaluation.

6.1. Performance

To validate the performance of the single point FMM, a series of numerical experiments were conducted using the randomly generated source points and evaluation points in an ellipsoid. The same parameters were used for all the experimental results: the expansion order is $p = 10$ and the kernel is biharmonic spline. The number of evaluation points varies from 10^2 to 10^7.

Figures 5–7 demonstrate comparison results of the evaluation time using the original FMM (blue) and the single point FMM (red) with different steps. Note that the original FMM evaluates all the points as a whole in parallel. For the single point FMM, only the setup step is parallel. The comparison results show that the original FMM evaluating all the points as a whole has the best efficiency. If the points should be evaluated one by one, it is necessary to improve the FMM operations to achieve an optimum efficiency. The three steps of the single point FMM optimize the evaluation process by reducing the redundant computation as much as possible. The setup step improves the efficiency by precomputing the P2M and M2M operations. The translation step improves the efficiency by reducing the M2L and L2L operations, and the evaluation step improves the efficiency by optimizing the L2P operation.

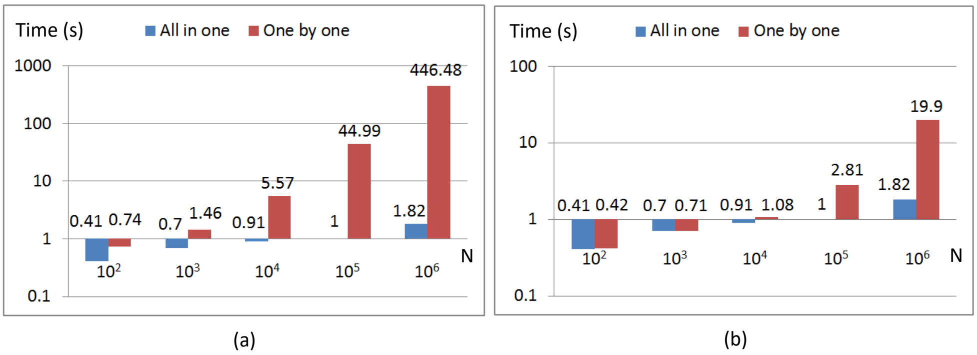

Figure 5. Comparisons of the evaluation time with 1000 source points using the original FMM (blue) and the single point FMM (red) without setup step (**a**) and with setup step (**b**).

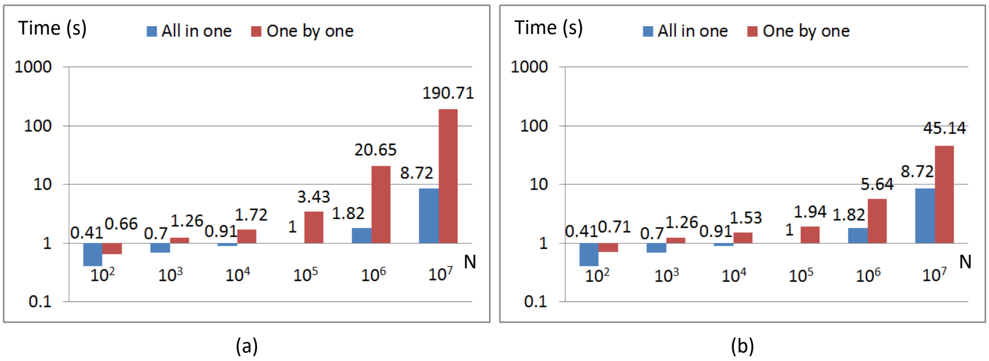

Figure 6. Comparison of the evaluation time with 1000 source points using the original FMM (blue) and the single point FMM (red) with three steps. Compared with (**a**), the single point FMM in (**b**) evaluates the points leaf by leaf.

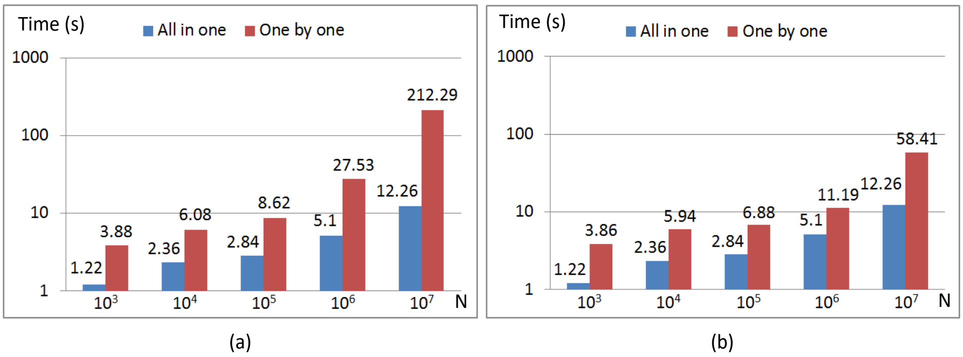

Figure 7. Comparison of the evaluation time with 10000 source points using the original FMM (blue) and the single point FMM (red) with three steps. Compared with (**a**), the single point FMM in (**b**) evaluates the points leaf by leaf.

6.2. Case Studies

We sampled sets of unorganized points with normals from a number of real objects to test the implicit surface reconstruction method, as shown in Figure 8. The unorganized points are used to construct the on-surface constraints ($f(x_i) = 0$) and the normals are used to construct off-surface constraints ($f(x_i) > 0$ or $f(x_i) < 0$) by projecting the points along the normal direction. The implicit function (the RBF interpolant) is obtained by interpolating the constraints and solving the corresponding linear system. Then the implicit surface can be reconstructed using the improved surface reconstruction method.

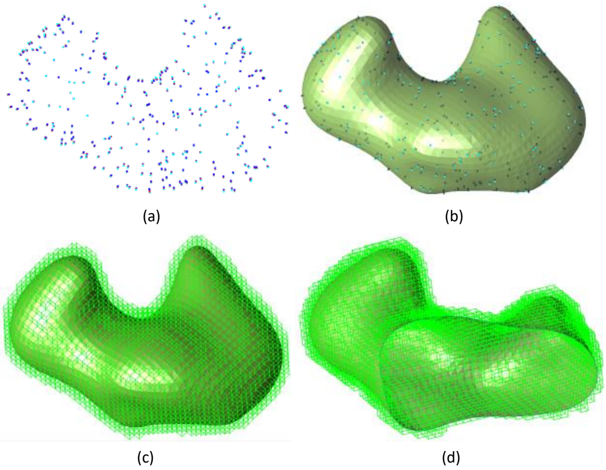

Figure 8. Implicit surface reconstruction from a set of unorganized points with normals. (**a**) The interpolation constraints, including the on-surface constraints and the off-surface constraints. (**b**) The reconstructed implicit surface. (**c**) The evaluated voxels in voxel growing. (**d**) The evaluated voxels in cutaway view.

In the process of surface reconstruction, all of the on-surface constraints are selected as the seed points. Figure 9 demonstrates comparison result of the voxel growing method with one and multiple seed points based on the on-surface constraints. The on-surface constraints are constructed by the sampling points. As mentioned, for the surface following method, the main problem is the incomplete reconstruction of sub-objects. Compared with the original voxel growing method, the improved method tends to recover the complete model as long as the sampling is sufficient.

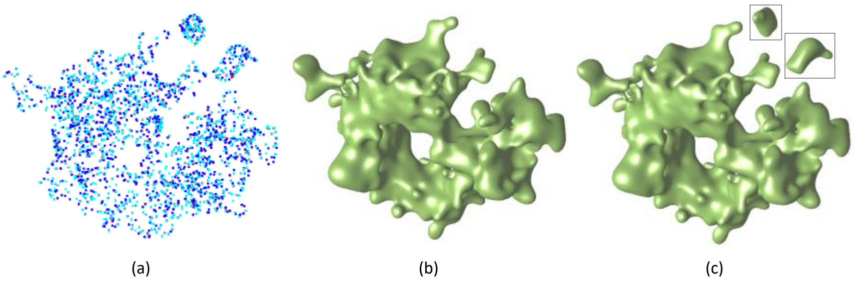

Figure 9. Comparison result of the voxel growing method with one (**b**) and multiple (**c**) seed points based on the interpolation constraints (**a**). The black boxes point to the complete reconstruction of multiple isosurfaces.

In comparison with the marching cubes method, we test the performance of the improved voxel growing method and the leaf growing method on a variety of geological data sets, as shown in Figures 10 and 11. The surface following method does not modify the triangulation of voxels. Though the evaluation process of the three methods is different, the reconstruction results are the same when using the same size of reconstruction cell.

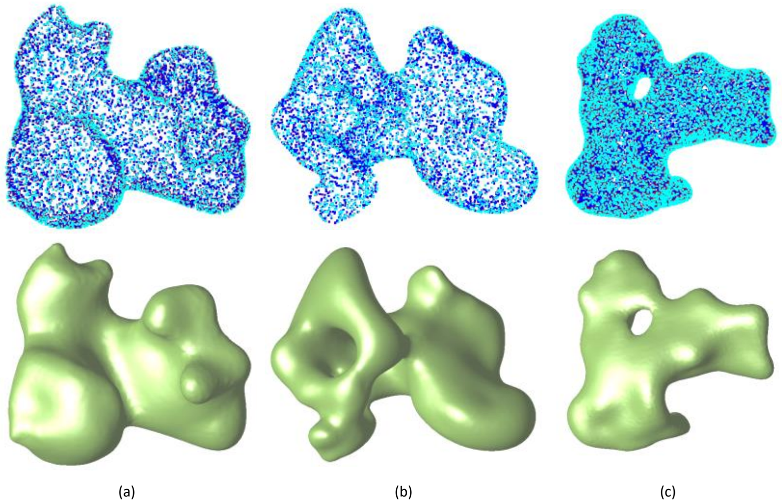

Figure 10. Reconstruction results of sets of unorganized points sampled from several real objects (**a**) orebody #1, (**b**) orebody #2 and (**c**) orebody #3.

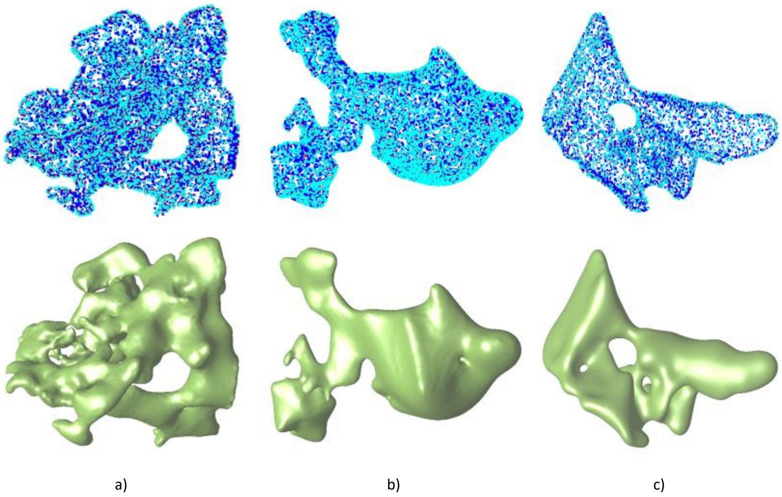

Figure 11. Reconstruction results of sets of unorganized points sampled from several real objects (**a**) orebody #1, (**b**) orebody #2 and (**c**) orebody #3.

The performance of the reconstruction method mainly depends upon the number of interpolation constraints, the expansion order and the size of the reconstruction cell. The number of interpolation constraints determines the number of source points in the FMM evaluation. The expansion order determines the precision of evaluation, and the size of the reconstruction cell determines the model precision.

Table 1 reports the timings of the evaluation of the marching cubes method, the voxel growing method and the leaf growing method on experimented examples. In the process of space division, the minimum bounding boxes of the sampling points are scaled by 0.01 times, respectively. Overall, because of the reduction of evaluation points, the voxel growing method has a better efficiency. The leaf growing method further improves the reconstruction efficiency. When the size of the reconstruction cell is large, the improvement in efficiency is not obvious. However, as the size of the reconstruction cell decreases, the performance advantage of the leaf growing method is more obvious. A series of experimental results show that the reduction of evaluation points can balance out the performance degradation for single point evaluation.

Table 1. Comparison of the evaluation time using the marching cubes method (MC), the voxel growing method (VG) and the leaf growing method (LG). Also showing the number of domain constraints N_c, the size of reconstruction cell d and the expansion order p.

Model	N_c	d	p	Evaluation Points			Time (Seconds)		
				MC	VG	LG	MC	VG	LG
Figure 10a	21,975	2.0	10	5,972,274	167,858	167,858	11.4	4.5	4.1
Figure 10b	21,792	2.0	10	16,310,430	422,615	422,615	19.1	8.2	7.5
Figure 10c	45,975	1.0	10	7,180,920	173,552	173,552	41.3	10.8	10.3
Figure 11a	29,770	1.0	10	44,498,160	759,525	759,525	61.7	14.5	12.7
Figure 11b	29,397	1.0	10	261,766,151	1,320,267	1,320,267	63.3	23.8	19.9
Figure 11c	19,065	3.0	10	56,107,920	738,284	738,284	73.2	11.8	10.6

7. Conclusions and Discussion

In this paper, we improve an efficient implicit surface reconstruction method for the radial basis functions interpolant. The growing strategy of the surface following method combines both the evaluation and reconstruction processes. It ensures a smaller number of evaluation points and higher evaluation efficiency in surface following. The single point FMM is improved to evaluate points one by one. Based on the improved voxel growing method without manually specifying the seed points, the leaf growing method is developed to avoid a mass of redundant computation. The black-box FMM is utilized to evaluate the RBF interpolant with different kernels in a wide range of applications. The numerical results showed that one point evaluation in surface following obtains an optimum efficiency to the original approach. Compared with the existing methods, the improved method performs a better time and space efficiency.

The main limitation of the method is that the evaluations of the target points are no longer mutually independent because of the surface following process. It increases the difficulty of the parallel execution of the algorithm to a certain extent. To overcome this limitation, the space should be divided into several blocks. The reconstruction process between blocks does not affect each other. If the blocks are divided according to the octree structure in FMM, the evaluation process of each block can be also performed independently. Therefore, the leaf growing method is used to reconstruct the local implicit surface in parallel for each block. Finally, the results of each block are spliced into a complete surface.

To further improve the reconstruction efficiency and mesh quality, there are still some extensions that need to be studied. For example, the cells used in voxel growing are the same size, and the operations of single point FMM can be optimized for parallel implementation. To reduce the evaluation points, the function values and the gradient values can be combined in leaf growing to reconstruct the implicit surface adaptively.

Author Contributions: D.Z. and J.Z. conceived, designed, and performed the experiments; L.W. analyzed the data and revised the methodology; D.Z. wrote the paper; all authors discussed the results and revised the paper.

Funding: This work was financially supported by the National Key R&D Program of China (2017YFC0602905) and the National Natural Science Foundation of China (41572317).

Acknowledgments: Thanks to the public datasets used in this research. We also 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. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. De Boer, A.; Van der Schoot, M.; Bijl, H. Mesh deformation based on radial basis function interpolation. *Comput. Struct.* **2007**, *85*, 784–795. [CrossRef]
2. Majdisova, Z.; Skala, V. Radial basis function approximations: Comparison and applications. *App. Math. Model.* **2017**, *51*, 728–743. [CrossRef]
3. 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; pp. 67–76.
4. Li, J.; Hon, Y. Domain decomposition for radial basis meshless methods. *Numer. Methods Partial Differ. Equ.* **2004**, *20*, 450–462. [CrossRef]
5. Krese, B.; Perc, M.; Govekar, E. The dynamics of laser droplet generation. *Chaos* **2010**, *20*, 013129. [CrossRef]
6. KRESE, B.; PERC, M.; Govekar, E. Experimental observation of a chaos-to-chaos transition in laser droplet generation. *Int. J. Bifurcat. Chaos* **2011**, *21*, 1689–1699. [CrossRef]
7. Jessell, M.; Aillères, L.; De Kemp, E.; Lindsay, M.; Wellmann, J.; Hillier, M.; Laurent, G.; Carmichael, T.; Martin, R. Next generation three-dimensional geologic modeling and inversion. *Econ. Geol.* **2014**, *18*, 261–272.
8. Calakli, F.; Taubin, G. SSD: Smooth signed distance surface reconstruction. *Comput. Graph. Forum* **2011**, *30*, 1993–2002. [CrossRef]
9. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Comput. Graph.* **1987**, *21*, 163–169. [CrossRef]
10. Newman, T.S.; Yi, H. A survey of the marching cubes algorithm. *Comput. Graph.* **2006**, *30*, 854–879. [CrossRef]
11. Lee, T.Y.; Lin, C.H. Growing-cube isosurface extraction algorithm for medical volume data. *Comput. Med. Imag. Grap.* **2001**, *25*, 405–415. [CrossRef]
12. Wang, X.; Niu, Y.; Tan, L.W.; Zhang, S.X. Improved marching cubes using novel adjacent lookup table and random sampling for medical object-specific 3D visualization. *J. Softw.* **2014**, *9*, 10. [CrossRef]
13. Darve, E. The fast multipole method: Numerical implementation. *J. Comput. Phys.* **2000**, *160*, 195–240. [CrossRef]
14. Greengard, L.; Rokhlin, V. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numer.* **1997**, *6*, 229–269. [CrossRef]
15. Hoppe, H. Poisson surface reconstruction and its applications. In Proceedings of the 2008 ACM Symposium on Solid and physical modeling, New York, NY, USA, 2–4 June 2008; p. 10.
16. Kazhdan, M.; Hoppe, H. Screened poisson surface reconstruction. *ACM Trans. Graph.* **2013**, *32*, 29. [CrossRef]
17. Frank, T.; Tertois, A.L.; Mallet, J.L. 3D-reconstruction of complex geological interfaces from irregularly distributed and noisy point data. *Comput. Geosci.* **2007**, *33*, 932–943. [CrossRef]
18. Mallet, J.L. Discrete modeling for natural objects. *Math. Geosci.* **1997**, *29*, 199–219. [CrossRef]
19. Beatson, R.K.; Ong, W.; Rychkov, I. Faster fast evaluation of thin plate splines in two dimensions. *J. Comput. Appl. Math.* **2014**, *261*, 201–212. [CrossRef]
20. Spivak, M.; Veerapaneni, S.K.; Greengard, L. The fast generalized Gauss transform. *SIAM J. Sci. Comput.* **2010**, *32*, 3092–3107. [CrossRef]
21. Beatson, R.; Powell, M.J.D.; Tan, A. Fast evaluation of polyharmonic splines in three dimensions. *IMA J. Numer. Anal.* **2006**, *27*, 427–450. [CrossRef]

22. Ying, L. A kernel independent fast multipole algorithm for radial basis functions. *J. Comput. Phys.* **2006**, *213*, 451–457. [CrossRef]
23. Fong, W.; Darve, E. The black-box fast multipole method. *J. Comput. Phys.* **2009**, *228*, 8712–8725. [CrossRef]
24. Zhang, Q.; Zhao, Y.; Levesley, J. Adaptive radial basis function interpolation using an error indicator. *Numer. Algorithms* **2017**, *76*, 441–471. [CrossRef]
25. Nielson, G.M. Dual marching cubes. In Proceedings of the IEEE Visualization 2004, Austin, TX, USA, 10–15 October 2004; pp. 489–496.
26. Shu, R.; Zhou, C.; Kankanhalli, M.S. Adaptive marching cubes. *Vis. Comput.* **1995**, *11*, 202–217. [CrossRef]
27. Congote, J.; Moreno, A.; Barandiaran, I.; Barandiaran, J.; Ruiz, O. Extending Marching Cubes with Adaptative Methods to obtain more accurate iso-surfaces. In Proceedings of the International Conference on Computer Vision, Imaging and Computer Graphics, Lisboa, Portugal, 5–8 February 2009; pp. 35–44.
28. Wu, Z.; Sullivan, J.M., Jr. Multiple material marching cubes algorithm. *Int. J. Numer. Meth. Eng.* **2003**, *58*, 189–207. [CrossRef]
29. Treece, G.M.; Prager, R.W.; Gee, A.H. Regularised marching tetrahedra: Improved iso-surface extraction. *Comput. Graph.* **1999**, *23*, 583–598. [CrossRef]
30. Beatson, R.K.; Cherrie, J.B.; Mouat, C.T. Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Adv. Comput. Math.* **1999**, *11*, 253–270. [CrossRef]
31. Beatson, R.K.; Light, W.; Billings, S. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM J. Sci. Comput.* **2001**, *22*, 1717–1740. [CrossRef]
32. Gumerov, N.A.; Duraiswami, R.; Borovikov, E.A. *Data Structures, Optimal Choice of Parameters, and Complexity Results for Generalized Multilevel Fast Multipole Methods in d Dimensions*; University of Maryland: College Park, MD, USA, 2003.
33. Takahashi, T.; Cecka, C.; Darve, E. Optimization of the parallel black-box fast multipole method on CUDA. In Proceedings of the 2012 Innovative Parallel Computing (InPar), San Jose, CA, USA, 13–14 May 2012; pp. 1–14.