Compact implicit surface reconstruction via low-rank tensor approximation☆

Maodong Pan, Weihua Tong, Falai Chen *

School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, 230026, PR China

ARTICLE INFO

Keywords:
Compact representation
Implicit surface
Surface reconstruction
Low-rank approximation
Tensor

ABSTRACT

Implicit representations have gained an increasing popularity in geometric modeling and computer graphics due to their ability to represent shapes with complicated geometry and topology. However, the storage requirement, e.g. memory or disk usage, for implicit representations of complex models is relatively large. In this paper, we propose a compact representation for multilevel rational algebraic spline (MRAS) surfaces using low-rank tensor approximation technique, and exploit its applications in surface reconstruction. Given a set of 3D points equipped with oriented normals, we first fit them with an algebraic spline surface defined on a box that bounds the point cloud. We split the bounding box into eight sub-cells if the fitting error is greater than a given threshold. Then for each sub-cell over which the fitting error is greater than the threshold, an offset function represented by an algebraic spline function of low rank is computed by locally solving a convex optimization problem. An algorithm is presented to solve the optimization problem based on the alternating direction method of multipliers (ADMM) and the CANDECOMP/PARAFAC (CP) decomposition of tensors. The procedure is recursively performed until a certain accuracy is achieved. To ensure the global continuity of the MRAS surface, quadratic B-spline weight functions are used to blend the offset functions. Numerous experiments show that our approach can greatly reduce the storage of the reconstructed implicit surface while preserve the fitting accuracy compared with the state-of-the-art methods. Furthermore, our method has good adaptability and is able to produce reconstruction results with high quality.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Over the past two decades there has been an immense amount of efforts dedicated to obtain digital representations of objects in the real world. Techniques for digitizing objects include optical laser-based range scanners, structured light scanners, LiDAR scanners, multi-view stereo and so on. A recent trend has seen the massive proliferation of point clouds from commodity real-time scanners such as the Microsoft Kinect. As a result of the ability to acquire point cloud data, the need for the development of surface reconstruction techniques continues to increase. Moreover, since acquisition methods tend to produce point clouds containing a variety of properties and imperfections, e.g., the presence of geometric features or missing data, noise or outliers, etc., surface

reconstruction is a challenging task. A good survey can be found in [1].

Existing surface reconstruction methods can be broadly categorized as polygonal mesh approaches and implicit surface approaches [2]. The methods in former class typically generate a triangular mesh by interpolating a subset of the input points as vertices. Although these approaches produce mesh representation directly, they are difficult to handle non-uniform, incomplete or noisy data. Please refer to [3,4] for details.

Implicit surface approaches usually create an implicit function or an indicator function for the underlying surface the point cloud is sampled from, and perform iso-surfacing to extract a triangular mesh for rendering. Implicit representations greatly facilitate the classification problem of whether a given point is on, inside or outside a surface. They are able to represent shapes with complicated topology and geometry, even with dynamic topology [5,6]. Implicit representations are more suitable for reconstructing surfaces from datasets that are noisy, incomplete or non-uniformly distributed. As a result, they are widely used in surface reconstruction. Many representations of implicit surfaces have been proposed, e.g. the Blobby model, the signed distance

fields, the radial basis functions (RBFs), the moving least squares (MLS) surfaces, the algebraic spline (AS) surfaces, the multilevel partition of unity (MPU) and so on. However, no matter which implicit representation is used, the storage requirement for state-of-the-art reconstruction methods is usually large for complex models [7]. To our knowledge, little work has been done on compact representations in implicit surface reconstruction.

In this paper, we propose an algorithm for creating compact representations for Multilevel Rational Algebraic Spline (MRAS) surfaces using low-rank tensor approximation technique, and exploit its applications in surface reconstruction. MRAS surfaces have an analytic spline representation which is advantageous over mesh representation in problems such as point classification, and in subsequent computations such as function/derivative evaluation, boolean operations, shape blending, etc. An MRAS surface can be conveniently stored by its control coefficients. However, for complex models the storage requirement for the control coefficients can be very large, ranging from several hundred megabytes to tens of hundred megabytes [7]. Large storage requirement is a big burden for memory costs in the computation with the implicit surface and in the transition of the implicit surface on the internet. Thus how to reduce the number of control coefficients in MRAS representation is an important research problem. Our approach is outlined as follows.

Given a point cloud with oriented normals, our aim is to build a MRAS surface that approximates the scanned surface as much as possible while uses less storage space. We start with fitting an algebraic spline surface defined on a box that bounds the given point set. We split the box into eight sub-cells if the fitting error is greater than a given threshold. For each cell over which the fitting error is larger than the given threshold, a local offset function represented by an algebraic spline function is computed to reduce the fitting error. In order to obtain a compact representation, a low-rank regularization term is introduced in the fitting model, and the optimization problem is solved locally based on the alternating direction method of multipliers together with the CP decomposition of tensors. The above procedure terminates when the fitting error over each cell is less than a user-specified threshold or the maximum subdivision level reaches.

The remainder of this paper is organized as follows. Section 2 reviews some related work. Section 3 presents some preliminary knowledge about algebraic spline surfaces and tensor decompositions. In Section 4, we introduce a new implicit representation, i.e., the MRAS surface. Section 5 describes our adaptive surface reconstruction algorithm in detail. To achieve compact representations, a convex optimization model equipped with a low-rank regularization term and a corresponding algorithm based on the alternating direction method of multipliers and the CP decomposition are proposed. Section 6 demonstrates some experimental results and performance of our algorithm. Comparisons on the storage requirement with the state-of-the-art methods are also included. Finally, we conclude the paper with proposals for future work.

2. Related work

Since surface reconstruction has been well studied in the past several decades, there is a large body of related work. For the sake of clarity, we shall focus on the approaches that are most related to ours.

2.1. Surface reconstruction

Most implicit surface reconstruction methods are based on Blinn's idea of blending local implicit primitives [8], called *blobs*. Fitting scattered data with algebraic surfaces was discussed by Pratt [9]. Muraki [10] combined the above two ideas and proposed the Blobby model for fitting an implicit surface to a given point set.

Hoppe et al. [11] proposed a reconstruction algorithm based on the signed distance function, which is locally defined. Curless and Levoy [12] used the volumetric representation consisting of a cumulative weighted signed distance function. Kazhdan et al. [13] presented the Poisson surface reconstruction by approximating the indicator function of the underlying surface. To avoid over-smoothing of the data, they further introduced positional constraints into the optimization, resulting in a screened Poisson problem [2]. The work of Manson et al. [14] solved the Poisson equation using a wavelet basis, which provided a localized, multi-resolution representation of functions.

Using projection moving least squares (PMLS) to reconstruct a C^∞ surface from the point cloud (i.e., point-set surface) was originally proposed by Levin [15], and was used by Alexa et al. [16] in point-based graphics. An explicit definition of point-set surface as the set of local minima of an energy function was given by Amenta and Kil [17]. An implicit MLS (IMLS) approach proposed by Shen et al. [18] based on the classical MLS method, has been used to build interpolating or approximating implicit surfaces from polygonal data. Fleishman et al. [19] devised the robust MLS fitting technique, which could deal with noise and sharp features simultaneously. To separate mixed scanning points received from a thin-wall object, Feng et al. [20] proposed a model called moving multiple curves/surfaces approximation that is an second-order extension of PMLS.

Although the signed or unsigned distance function and the MLS surface have some desirable properties, the absence of analytical expression limits their usage in many applications. Radial basis functions (RBFs) interpolant is a linear combination of radially symmetric functions with distinct centers. The early surface reconstruction algorithms based on RBF interpolants are usually credited to Carr et al. [21], and Turk and O'Brien [22]. To reconstruct surfaces from large datasets, Morse et al. [23] proposed to use the compactly supported RBFs. A thorough treatment of the RBFs is given by Buhmann [24].

Another family of implicit surface reconstruction algorithms is the partition of unity. Ohtake et al. [25] used the multilevel partition of unity (MPU) together with three types of local approximation quadratic functions to reconstruct implicit surfaces from very large sets of points, including surfaces with sharp features. Furthermore, Ohtake et al. [26] combined RBFs and the partition of unity as a mean for large, non-uniform datasets. Piecewise algebraic surface patches defined within a tetrahedral lattice of control points were firstly introduced by Sederberg [27]. Its successor A-patch proposed by Bajaj et al. [28] was used to reconstruct a C^1 continuous surface and a scalar field defined over it. To have some desirable properties from both global and local representations, Jüttler and Felis [29] applied algebraic spline surfaces into fitting scattered data. By extending the geometric distance minimization and using the trust region technique, Yang et al. [30] proposed to use active implicit B-spline curves for fitting unorganized point clouds. Rouhani and Sappa [31] presented an extension of the 3L algorithm to the implicit tensor-product B-spline solution space. Recently, they [32] further developed a reconstruction algorithm using the partition of unity technique. The applications of algebraic spline surfaces in approximate implicitization can be found in [33]. Polynomial splines over hierarchical T-meshes (PHT-splines) [34] are a useful generalization of B-splines over T-meshes. Wang et al. [7] proposed an adaptive surface reconstruction algorithm based on implicit PHT-splines, which can produce high quality reconstruction surfaces very efficiently at the cost of large storage requirement.

2.2. Low-rank tensor approximation

Low-rank approximation of matrices is helpful for dimension reduction, classification, denoising, and has been successfully applied in many fields like signal processing, computer vision, pattern recognition, computer graphics, etc. A thorough survey on this topic is out of our scope, and we refer the reader to [35,36] and references therein. Tensors, as a generation of matrices in higher dimensions have important applications in science and engineering, e.g., psychometrics, psychometrics and data mining [37]. The details of low-rank tensor approximation and applications can be found in [38]. Recently, the low-rank tensor optimization is also applied in finding the upright orientation of 3D shapes in [39]. Mantzaflaris et al. [40] applied low-rank matrix approximation for accelerating the assembly process of stiffness matrices in isogeometric analysis. For other applications of low-rank tensors in geometric modeling and processing, please refer to [41] and references therein. However, to the best of our knowledge, no previous attempts have been made at compact representations for implicit surfaces and surface reconstruction using low-rank tensor approximation technique.

3. Preliminaries

In this section, we give some preliminary knowledge about algebraic spline surfaces followed by the basic concepts on tensor and tensor decomposition.

3.1. Algebraic spline surfaces

Let $f(x, y, z)$ be a trivariate tensor-product spline function of tri-degree (d_1, d_2, d_3) defined over a cubic domain Ω:

$$f(x, y, z) = \sum_{r=1}^{m} \sum_{s=1}^{n} \sum_{t=1}^{l} c_{rst} N_r(x) N_s(y) N_t(z) \tag{1}$$

where $\{c_{rst}\}_{m \times n \times l}$ are the control coefficients, $\{N_r(x)\}_{r=1}^{m}$, $\{N_s(y)\}_{s=1}^{n}$, $\{N_t(z)\}_{t=1}^{l}$ are B-spline basis functions w.r.t knot sequences $\{\xi_r\}_{r=1}^{m+d_1+1}$, $\{\eta_s\}_{s=1}^{n+d_2+1}$, $\{\zeta_t\}_{t=1}^{l+d_3+1}$ respectively. The zero level set of the function f in Ω is defined by

$$V(f) = \{(x, y, z) \in \Omega \mid f(x, y, z) = 0\} \tag{2}$$

and it is called an *algebraic spline surface* [29].

3.2. Tensor and CP decomposition

A *tensor* is a multidimensional array. More formally, an nth-order or n-way tensor is an element of the tensor product of n vector spaces, each of which has its own coordinate system [37]. A first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors. An nth-order tensor is denoted by boldface Euler script letters, e.g. $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_n}$. It is sometimes convenient to unfold a tensor into a matrix. The unfold operation along the kth order on a tensor $\mathcal{X}$ is defined as $unfold_k(\mathcal{X}) := X_{(k)} \in \mathbb{R}^{I_k \times (I_1 \cdots I_{k-1} I_{k+1} \cdots I_n)}$.

An nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_n}$ is *rank one* if it can be written as the outer product of n vectors, i.e.,

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(n)},$$

where $\circ$ denotes the outer product, and $\mathbf{a}^{(i)} \in \mathbb{R}^{I_i}$.

(CP decomposition) Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_n}$ be an nth-order tensor, the *CP decomposition* factorizes $\mathcal{X}$ into a sum of component rank-one tensors as follows:

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(n)}, \tag{3}$$

where R is a positive integer and $\lambda_r > 0$, $\mathbf{a}_r^{(i)} \in \mathbb{R}^{I_i}$ for $r = 1, \ldots, R$. It is often useful to assume that $\mathbf{a}_r^{(i)}$ are normalized to length one with the weights absorbed into λ_r.

The *rank* of a tensor $\mathcal{X}$, denoted as rank($\mathcal{X}$), is the smallest number of components in an exact CP decomposition, where "exact" means that (3) is an equality. However, finding the above "exact" decomposition with the minimum R is NP-hard [37]. In practice, the rank of $\mathcal{X}$ is usually determined numerically by solving the following rank-R CP model:

$$\arg\min_{\hat{\mathcal{X}}} \quad \|\mathcal{X} - \hat{\mathcal{X}}\|$$
$$\text{s.t.} \qquad \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \mathbf{a}_r^{(n)}, \tag{4}$$

with different R until one is satisfied. In this paper, the initial value of R is automatically provided by our method, and we adjust R adaptively according to the fitting error. Once R is given, there are many algorithms to compute the CP decomposition. State-of-the-art methods include the alternating least squares (ALS) method, which is proposed in the original papers by Carroll and Change [42]. For recent developments of the ALS method, e.g. acceleration, convergence analysis, efficiency and robustness etc., please refer to the survey papers [38,37].

The rank of a tensor is a nonconvex function, and solving a rank-constrained problem is often NP-hard. Recently several works [43,44] use the trace norm of a tensor to approximately calculate the rank, which leads to a convex optimization problem. The trace norm of a tensor is defined as follows:

$$\|\mathcal{X}\|_* := \sum_{i=1}^{n} \gamma_i \|X_{(i)}\|_*, \tag{5}$$

where $\{\gamma_i\}$ are constants satisfying $\gamma_i \geq 0$ and $\sum_{i=1}^{n} \gamma_i = 1$, and $\|X_{(i)}\|_*$ denotes the trace norm of the matrix $X_{(i)}$, i.e.,

$$\|X_{(i)}\|_* = \sum_j \sigma_j(X_{(i)})$$

where $\sigma_j(X_{(i)})$ is the jth largest singular value of $X_{(i)}$.

4. Multilevel rational algebraic spline surfaces

Before going into the details of our reconstruction algorithm, we introduce a new implicit representation, i.e., Multilevel Rational Algebraic Spline (MRAS) surfaces. Such representation offers several advantages, such as compact and analytic expression, efficient evaluation and sufficient flexibility for subsequent processing.

The key to constructing a multilevel or multiresolution model is choosing a proper sequence of nested spaces

$$V_1 \subset V_2 \subset V_3 \subset \cdots \subset V_K,$$

where $\{V_k\}$ are spanned by a set of basis functions. These basis functions are also called the scaling functions, and in this paper we choose tensor-product B-spline basis functions defined in (1) as the scaling functions.

In order to build a multiresolution function, the domain Ω is subdivided in an adaptive manner (see 5.1). Let Ω_j^k and M_k denote the jth cell and the number of cells at level k respectively, we have $\Omega = \cup_{j=1}^{M_k} \Omega_j^k$. For each cell Ω_j^k, we define a tensor-product B-spline function

$$\phi_j^k(\mathbf{p}) = \sum_{r=1}^{m_k} \sum_{s=1}^{n_k} \sum_{t=1}^{l_k} c_{rst}^k N_r^k(x) N_s^k(y) N_t^k(z), \tag{6}$$

where $\mathbf{p} = (x, y, z) \in \Omega_j^k$ and $\{N_r^k(x)\}_{r=1}^{m_k}$, $\{N_s^k(y)\}_{s=1}^{n_k}$, $\{N_t^k(z)\}_{t=1}^{l_k}$ are B-spline basis functions whose knot sequences compose a
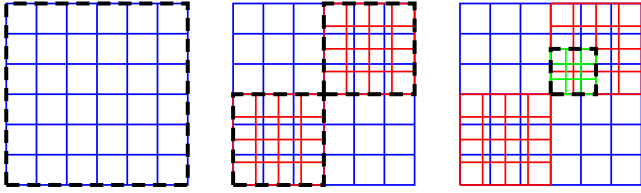
Fig. 1. An illustration of the adaptive subdivision of domain: from left to right are level 0, level 1 and level 2 meshes, and the blue, red and green grid denote the tensor-product mesh of level 0, 1, 2 respectively, the black dashed lines highlight the adaptive subdivision of the domain.

tensor-product mesh $\mathscr{T}_j^k$. Fig. 1 illustrates how our adaptive subdivision scheme works in 2D.

We construct the sequence of nested spaces in the following way:

$$V_k = \text{span}\{\phi_j^l, j = 1, 2, \ldots, M_l, l = 1, 2, \ldots, k\}.$$

A multilevel model is constructed by finding a sequence of functions $\{f^k\}_{k=1}^K$ from $\{V_k\}_{k=1}^K$ respectively. Similar to the work of Forsey and Bartels [45], we represent $\{f^k\}_{k=1}^K$ as follows:

$$\begin{cases} f^k(\mathbf{p})\mid_{\Omega_j^k} := f_j^k(\mathbf{p}), & j = 1, 2, \ldots, M_k, \\ f_j^k(\mathbf{p}) = f^{k-1}(\mathbf{p}) + \phi_j^k(\mathbf{p}), & \mathbf{p} \in \Omega_j^k, \end{cases} \tag{7}$$

where $\phi_j^k(\mathbf{p})$ is called the *local offset function* and is used to represent the fine details of the model.

In our method, the local offset functions $\{\phi_j^k(\mathbf{p})\}$ are computed locally and independently. Thus, in order to obtain a global smooth function, we adopt the technique of the partition of unity [25]. The main idea is to blend the local functions together using smooth and local weight functions, which sum up to one everywhere on the domain. The resulting global function inherits many properties from the local functions, such as the maximum error and convergence order.

For each cell Ω_j^k, we introduce the weight function by a quadratic B-spline basis function

$$w_j^k(\mathbf{p}) = b\left(\frac{3|\mathbf{p} - \mathbf{c}_j|}{2\alpha R_j}\right), \tag{8}$$

where $\mathbf{c}_j$ and R_j are the center and radius of Ω_j^k respectively, b is a quadratic B-spline function and α is constant. Typically, we set $\alpha = 0.75$. Then, the partition of unity functions $\{\varphi_j^k\}$ can be generated by

$$\varphi_j^k(\mathbf{p}) = \frac{w_j^k(\mathbf{p})}{\sum_{i=1}^{M_k} w_i^k(\mathbf{p})}.$$

We finally construct the global function as follows:

$$f^k(\mathbf{p}) := \sum_{j=1}^{M_k} \varphi_j^k(\mathbf{p}) f_j^k(\mathbf{p}). \tag{9}$$

Consequently, a sequence of implicit surfaces defined by $\{V(f^k)\}_{k=1}^K$ are obtained:

$$V(f^k) = \{(x, y, z) \in \Omega \mid f^k(x, y, z) = 0\}$$

and they are called the MRAS surfaces.

5. Adaptive surface reconstruction in compact representation

Now we turn to our adaptive surface reconstruction method. Given an unorganized collection of points $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$

sampled from a surface S in $\mathbb{R}^3$, and associated with oriented normals $\mathcal{N} = \{\mathbf{n}_1, \ldots, \mathbf{n}_N\}$, our aim is to construct a sequence of MRAS functions $\{f^k\}_{k=1}^K$ whose zero level sets provide a coarse-to-fine approximation to the scanned surface $\mathcal{S}$. At the same time, the storage requirement for $\{f^k\}_{k=1}^K$ should be as small as possible.

5.1. Outline

Our algorithm for creating $\{f^k\}_{k=1}^K$ is driven by adaptive octree-based subdivision of the domain Ω, which typically is the bounding box of point set $\mathcal{P}$. We start with $f^0 = 0$, $M_1 = 1$, $\Omega_1^1 = \Omega$, and repeat the following two steps:

1. Solve a convex optimization problem (15) locally and do the CP decomposition (4) to obtain the local offset function $\phi_j^k(\mathbf{p})$ defined in each Ω_j^k, then generate the global function f^k using (7) and (9) at level k.
2. Check the fitting error for every cell Ω_j^k, and split it into eight sub-cells if the fitting error is greater than a given threshold; Let $k \leftarrow k + 1$.

Over each cell, the fitting error is defined according to the Sampson distance of the point set to a surface [46]:

$$\varepsilon = \max_{\mathbf{p}_i \in \Omega_j^k} \frac{|f^k(\mathbf{p}_i)|}{|\nabla f^k(\mathbf{p}_i)|}. \tag{10}$$

If the number of sample points in Ω_j^k is greater than $N_{\min}$ (default value $N_{\min} = 16$ in our implementation) and the fitting error is greater than a user-specified ε_0, then we split Ω_j^k into eight sub-cells. The above procedure stops when no cell needs to be subdivided or the maximum level of subdivision reaches.

Fig. 2 shows an example of our adaptive surface reconstruction in multilevel representation.

5.2. Fitting model

A key ingredient in our approach is to compute the local offset function of low rank ϕ_j^k defined in Ω_j^k. Before proceeding further, let us explain the reason for using the low-rank representation. Let $\mathcal{C}_k$ be a third-order tensor associated with the coefficients $\{c_{rst}^k\}$ as defined in (6), and denote $\mathbf{N}(x) = (N_1(x), N_2(x), \ldots, N_{m_k}(x))^T$. $\mathbf{N}(\mathbf{y})$ and $\mathbf{N}(\mathbf{z})$ are defined similarly. If the rank of $\mathcal{C}_k$ is R_k and perform the CP decomposition of $\mathcal{C}_k$ numerically as

$$\mathcal{C}_k \overset{(4)}{\approx} \sum_{l=1}^{R_k} \lambda_l^k \mathbf{a}_l^{(k)} \circ \mathbf{b}_l^{(k)} \circ \mathbf{c}_l^{(k)} := \hat{\mathcal{C}}_k, \tag{11}$$

then ϕ_j^k can be approximately represented in a separated form:

$$\phi_j^k \approx \sum_{l=1}^{R_k} \lambda_l^k g_l^{(1)}(x) g_l^{(2)}(y) g_l^{(3)}(z) := \hat{\phi}_j^k, \tag{12}$$

where $g_l^{(1)}(x) = \mathbf{N}(\mathbf{x})^T \mathbf{a}_l^{(k)}$, $g_l^{(2)}(y) = \mathbf{N}(\mathbf{y})^T \mathbf{b}_l^{(k)}$ and $g_l^{(3)}(z) = \mathbf{N}(\mathbf{z})^T \mathbf{c}_l^{(k)}$. Now, we only need to store about $(m_k + n_k + l_k) R_k$ entries for recording ϕ_j^k instead of $m_k n_k l_k$ entries. The key observation is that if we want to have a compact representation of ϕ_j^k, the rank of tensor $\mathcal{C}_k$ should be as small as possible. In such a case, $\hat{\phi}_j^k$ is called a *low-rank approximation of function* ϕ_j^k and R^k is called the rank of $\hat{\phi}_j^k$.

Based on the above analysis, our MRAS surface fitting model takes the following form:

$$\begin{aligned} &\arg\min_{\mathcal{C}_k} \quad \text{rank}(\mathcal{C}_k) \\ &\text{subject to} \quad \mathcal{E}^{\frac{1}{2}}(\phi_j^k) \leq \delta, \end{aligned} \tag{13}$$
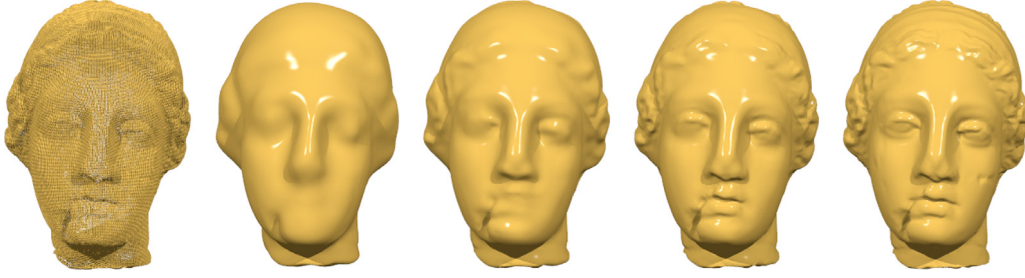
Fig. 2. Adaptive reconstruction of the Igea model, from left to right are the input point set, and the intermediate results at level 1 to level 4.

where $\mathcal{E}(\phi_j^k)$ is an energy functional which measures how well the MRAS surface approximates the given point cloud:

$$\mathcal{E}(\phi_j^k) = \sum_{u=1}^{N_j^k} [f_j^k(\mathbf{p}_u)]^2 + \omega_1 \sum_{u=1}^{N_j^k} \|\nabla \phi_j^k(\mathbf{p}_u) - \mathbf{n}_u\|^2$$
$$+ \omega_2 \iiint_{\Omega_j^k} \|\nabla^2 \phi_j^k\|_F^2 \, dxdydz. \tag{14}$$

Here N_j^k is the number of sample points contained in Ω_j^k, ω_1 and ω_2 are non-negative weights, and δ is the error tolerance.

In the energy functional, the first term describes the sum of squared algebraic distances, the second term reflects the normal difference that also helps to avoid the trivial solution of (13), and the third term is a smoothing term or regularization term that tries to pull the resulting surface to a simple shape [29].

By choosing an appropriate multipliers, we convert the constrained optimization problem (13) into an unconstrained optimization problem as follows:

$$\arg \min_{\mathcal{C}_k} \mathcal{E}(\phi_j^k) + \lambda \text{rank}(\mathcal{C}_k)$$

where λ is the multiplier and depends on the parameter δ. Since the function $\text{rank}(\mathcal{C}_k)$ is nonconvex, we further replace it with the trace norm $\|\cdot\|_*$ [44] and thus we obtain the following optimization problem:

$$\arg \min_{\mathcal{C}_k} \mathcal{E}(\phi_j^k) + \lambda \sum_{i=1}^{3} \gamma_i \|C_{(i)}^k\|_*, \tag{15}$$

which is a convex optimization problem and the problem is locally defined.

5.3. Numerical algorithm

Next, we present some details of our numerical algorithm for solving (15) based on the alternating direction method of multipliers (ADMM) [47] and the CP decomposition of tensors. The ADMM can be viewed as an attempt to blend the benefits of dual decomposition and augmented Lagrangian methods, and is used to solve constrained optimization problems with separable objective functions.

Variable splitting Since the objective function in (15) is the sum of two functions and one of which is dependent on the other, using variable splitting technique leads to the following constrained optimization problem:

$$\min_{\mathbf{c}} \quad \mathcal{E}(\phi_j^k) + \lambda \sum_{i=1}^{3} \gamma_i \|Z_i\|_*$$
$$\text{s.t. } P_i \mathbf{c} = \mathbf{z}_i \quad (i = 1, 2, 3) \tag{16}$$

where $\mathbf{c}$ is the vectorization of $\mathcal{C}_k$ as before, P_i is the matrix representation of mode-i unfolding (note that P_i is a permutation matrix, and satisfies $P_i^T P_i = I$), Z_i is an auxiliary matrix of the same

size as the mode-i unfolding of $\mathcal{C}_k$, and $\mathbf{z}_i$ is the vectorization of Z_i. For the sake of simplicity, we omit the superscript k for some variables in (16).

Augmented Lagrangian One typical way for solving (16) is to use an augmented Lagrangian scheme. In our problem, the augmented Lagrangian function is defined as

$$\mathcal{L}_\eta(\mathbf{c}, \{Z_i\}_{i=1}^3, \{\boldsymbol{\alpha}_i\}_{i=1}^3) = \mathcal{E}(\phi_j^k) + \lambda \sum_{i=1}^{3} \gamma_i \|Z_i\|_*$$
$$+ \sum_{i=1}^{3} \boldsymbol{\alpha}_i^T (P_i \mathbf{c} - \mathbf{z}_i) + \sum_{i=1}^{3} \frac{\eta}{2} \|P_i \mathbf{c} - \mathbf{z}_i\|_2^2, \tag{17}$$

where $\boldsymbol{\alpha}_i$ is a vector of Lagrangian multiplier corresponding to the constraint $P_i \mathbf{c} = \mathbf{z}_i$, and $\eta > 0$ is the penalty parameter.

ADMM algorithm Now the ADMM algorithm can be outlined as follows:

Algorithm 1 The ADMM algorithm

Input:
  λ, ω_1, ω_2, $\{\gamma_i\}_{i=1}^3$, $\eta > 0$, and
  initial values for $\{Z_i^0\}_{i=1}^3$, $\{\boldsymbol{\alpha}_i^0\}_{i=1}^3$
Output:
  an optimal $\mathbf{c}^*$
1: $t \leftarrow 0$
2: **repeat**
3: $\mathbf{c}^{t+1} = \arg \min_{\mathbf{c}} \mathcal{L}_\eta(\mathbf{c}, \{Z_i^t\}_{i=1}^3, \{\boldsymbol{\alpha}_i^t\}_{i=1}^3)$
4: $Z_i^{t+1} = \arg \min_{Z_i} \mathcal{L}_\eta(\mathbf{c}^{t+1}, \{Z_i\}_{i=1}^3, \{\boldsymbol{\alpha}_i^t\}_{i=1}^3)$
5: $\boldsymbol{\alpha}_i^{t+1} = \boldsymbol{\alpha}_i^t + \eta(P_i \mathbf{c}^{t+1} - \mathbf{z}_i^{t+1})$
6: $t \leftarrow t + 1$
7: **until** stopping criterion is satisfied.

---

c-*subproblem* The subproblem for **c** is

$$\arg \min_{\mathbf{c}} \mathcal{L}_\eta(\mathbf{c}, \{Z_i^t\}_{i=1}^3, \{\boldsymbol{\alpha}_i^t\}_{i=1}^3) = \mathcal{E}(\phi_j^k)$$
$$+ \sum_{i=1}^{3} (\boldsymbol{\alpha}_i^t)^T (P_i \mathbf{c} - \mathbf{z}_i^t) + \sum_{i=1}^{3} \frac{\eta}{2} \|P_i \mathbf{c} - \mathbf{z}_i^t\|_2^2. \tag{18}$$

This is a quadratic optimization problem and the solution can be obtained by solving a sparse and symmetric linear system of equations. The preconditioned conjugate gradient method with incomplete Cholesky factorization is applied in our algorithm.

Z-subproblem The subproblem for $\{Z_i\}_{i=1}^3$ is

$$\arg \min_{Z_i} \mathcal{L}_\eta(\mathbf{c}^{t+1}, \{Z_i\}_{i=1}^3, \{\boldsymbol{\alpha}_i^t\}_{i=1}^3) = \lambda \sum_{i=1}^{3} \gamma_i \|Z_i\|_*$$
$$+ \sum_{i=1}^{3} (\boldsymbol{\alpha}_i^t)^T (P_i \mathbf{c}^{t+1} - \mathbf{z}_i) + \sum_{i=1}^{3} \frac{\eta}{2} \|P_i \mathbf{c}^{t+1} - \mathbf{z}_i\|_2^2, \tag{19}$$

which has the following closed form solution

$$Z_i^{t+1} = \text{prox}_{\lambda\gamma_i/\eta}^{\text{tr}}(P_i\mathbf{c}^{t+1} + \boldsymbol{\alpha}_i^t/\eta), \quad i = 1, 2, 3. \tag{20}$$

Note that the argument $P_i\mathbf{c}^{t+1} + \boldsymbol{\alpha}_i^t/\eta$ must be converted into a matrix of the same size as Z_i^t. Here the proximal operator $\text{prox}_{\lambda\gamma_i/\eta}^{\text{tr}}$ can be considered as a shrinkage operation on the singular values and is defined as follows:

$$\text{prox}_{\lambda\gamma_i/\eta}^{\text{tr}}(Y) = U\max(S - \lambda\gamma_i I/\eta, 0)V^T, \tag{21}$$

where $Y = USV^T$ is the singular value decomposition (SVD) of Y, and the max operation is taken element-wise. See [48] for the derivation.

In our implementation, the stopping criterion is that the value of **c** has almost no change or the maximum number of iterations reaches.

5.4. Compact representation

To obtain a compact representation of ϕ_j^k, we need to do the CP decomposition of $\mathcal{C}_k$ and store the resulting coefficients λ_l^k and rank-one tensors $\mathbf{a}_l^{(k)} \circ \mathbf{b}_l^{(k)} \circ \mathbf{c}_l^{(k)}$. We complete this task by the fast ALS algorithm [49], which is contained in the TENSORBOX package [50]. To start, we need to provide the number of rank-one components R_k. Since the trace norm $\| \cdot \|_*$ can give a good approximation of $\text{rank}(\mathcal{C}_k)$, we choose

$$R_k^0 = \left\lceil \sum_{i=1}^{3} \gamma_i \|C_{(i)}^k\|_* \right\rceil$$

as an initial guess. Then we gradually increase R_k based on the fitting error until it is less than a user-specified tolerance ε_{cp}^k. Algorithm 2 gives a detailed description of this strategy.

Algorithm 2 Adaptive CP decomposition

Input:
    $\mathcal{C}_k, \varepsilon_{cp}^k$, initial guess R_k^0
Output:
    $R_k, \{\lambda_l^k\}_{l=1}^{R_k}, \{\mathbf{a}_l^{(k)}\}_{l=1}^{R_k}, \{\mathbf{b}_l^{(k)}\}_{l=1}^{R_k}, \{\mathbf{c}_l^{(k)}\}_{l=1}^{R_k}$
  1: $R_k \leftarrow R_k^0$
  2: Obtain $\hat{\mathcal{C}}_k$ by the CP decomposition (11) with R_k components
  3: **while** $\|\mathcal{C}_k - \hat{\mathcal{C}}_k\| > \varepsilon_{cp}^k$ **do**
  4:     $R_k \leftarrow R_k + \left\lceil 3\log\frac{\|\mathcal{C}_k - \hat{\mathcal{C}}_k\|}{\varepsilon_{cp}^k} \right\rceil$
  5:     Obtain $\hat{\mathcal{C}}_k$ by the CP decomposition (11) with R_k components
  6: **end while**

---

6. Implementation and results

In this section, we evaluate the performance of our compact surface reconstruction method based on the MRAS representation. Some details of our implementation, comparisons with other methods and discussions are provided.

6.1. Implementation

The algorithms are implemented in C++ and MATLAB, and we use the Bloomenthal's polygonizer [51] for polygonization (i.e., isosurface extraction). All the experiments were performed on a PC with a quad-core Intel i5 @3.1 GHz processor and 8GB of RAM.

In our method, there are several parameters. Most of them can be set as default values or chosen by programs automatically, e.g., the convex combination coefficients γ_i in (15) are set to $\frac{1}{3}$ as suggested in [43], the penalty parameter η in (17) is typically

| (a) Input. | (b) $\lambda = 0.0005$. | (c) $\lambda = 0.004$. |

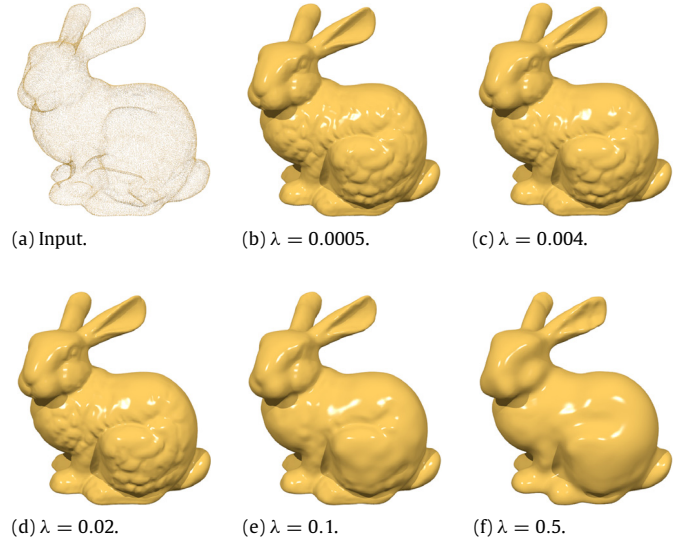| (d) $\lambda = 0.02$. | (e) $\lambda = 0.1$. | (f) $\lambda = 0.5$. |

Fig. 3. Reconstruction of the Bunny model using various parameters λ while leaving other parameters fixed.

set to 1, and the tolerance in Algorithm 2 is set as $\varepsilon_{cp}^k = 10^{-k-1}$. In our experiments, the knot sequences of B-spline basis functions in (6) are chosen uniformly in domain Ω_j^k for the sake of simplicity, although nonuniform knot sequences may achieve better results. The degrees of freedom in (6) are determined by the numbers of B-spline basis functions, i.e., m_k, n_k, l_k, which lead to a tradeoff between efficiency and accuracy. Since the fitting error is decreasing dramatically as the level k increases, we can choose $m_k = m_0 - \lfloor \beta k \rfloor$, $\beta \in [1, 3]$, n_k and l_k can be chosen in a similar way. The values of m_0, n_0 and l_0 are usually specified by users. Unless specified, we use quadratic B-spline basis functions, i.e., $d_1 = d_2 = d_3 = 2$, in our examples, which is proved to work well.

There are three weights, i.e., w_1, w_2 and λ in our local fitting model (15). We typically set $w_1 \in$ [5E-5, 1E-3]. The weight λ can be used to balance the reconstruction quality and the storage requirement for the reconstructed implicit function. Clearly, smaller λ leads to reconstruction results of higher quality while larger λ can reduce the storage requirement greatly. We observe that $\lambda \in$ [0.01, 0.05] provides a good compromise between them. For an illustration, see Figs. 3 and 4.

6.2. Results

Low-rank approximation of algebraic surfaces To verify the validity of our method, we sample some points from several algebraic surfaces, and then test our method on the resulting point sets. Due to the simplicity and the symmetry of algebraic surfaces, only a single level representation is used here. As demonstrated in Fig. 5, our method can produce very compact representations of the ground-truth surfaces with the specified accuracy, i.e., low-rank representations.

Storage. To demonstrate the compactness of our low-rank MRAS representation, we provide a comparison on storage requirement with several state-of-the-art surface reconstruction methods: Algebraic Spline surfaces (AS) [29], Multi-level Partition of Unity (MPU) [25], implicit PHT-splines (IPHT) [7], and Screened Poisson (SP) [2]. We measure the storage space of the implicit representations by the number of coefficients/parameters in the representations. For example, our method needs to record the information of all the local offset functions. Each local offset function ϕ_j^k has $(m_k + n_k + l_k)R_k$ coefficients to store. Thus the total storage requirement by our method is the sum of the
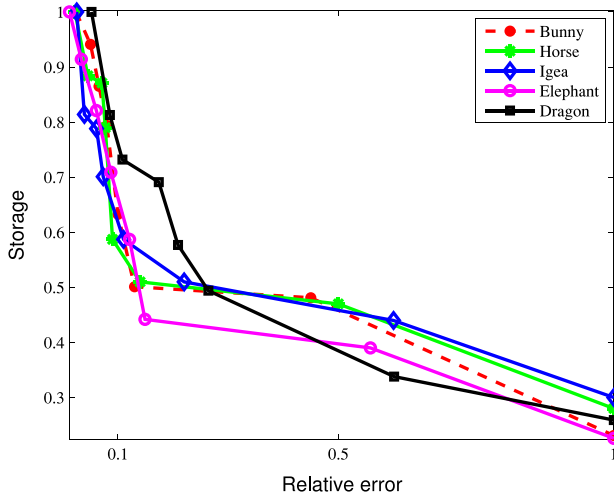
Fig. 4. Plot of the storage requirement for the reconstructed implicit function as a function of the relative fitting error. Here, the relative fitting error is measured by the maximum distance between the given point set and the reconstructed surface, which is then divided by the length of the main diagonal of the bounding box. And to provide better illustration, the storage requirement and the relative fitting error are normalized by their maximum values respectively.
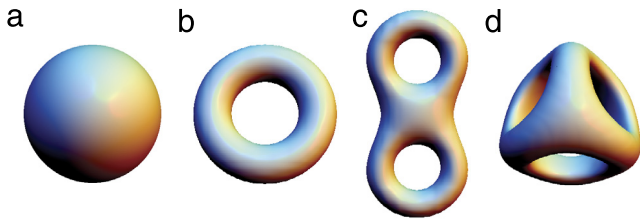

Fig. 5. Low-rank approximation of some algebraic surfaces with (a) sphere, $\varepsilon =$ 9.68E-4, $R_k = 4$; (b) torus, $\varepsilon = 8.32$E-4, $R_k = 10$; (c) double torus, $\varepsilon =$ 2.88E-3, $R_k = 18$; (d) 3-fold torus, $\varepsilon = 4.68$E-4, $R_k = 12$.

number of coefficients in all the offset functions, plus the number of parameters in all the quadratic blending functions. For MPU method, the storage requirement is the sum of the number of coefficients in all the three kinds of local shape functions over each cell, i.e., the general 3D quadric, the bivariate quadratic polynomial in local coordinates and the piecewise quadric surface. To store IPHT splines, we need to store the Bézier ordinates defined over each cell in the hierarchical T-mesh. The SP method needs to record the information in each node, which includes the width, the center and the corresponding weight coefficient.

For the sake of fairness, we carefully adjust the parameters for each approach to produce reconstruction results with roughly the same accuracy. Complete storage statistics are provided in Table 1. Clearly our method significantly outperforms other state-of-the-art surface reconstruction methods in terms of the storage requirement, thanks to the multilevel representation of MRAS and the low-rank tensor approximation technique. Some of our reconstruction results are shown in Fig. 6.

Fine details When the given point set contains fine-scale scans, the reconstruction approaches should be fully self-adaptive and have the capability to recover fine details of the scanned surface. The construction of our MRAS representation is a dynamic process and is particularly suitable for producing multilevel description of the underlying shape in an adaptive fashion, e.g., see Fig. 2. As shown in Fig. 7, the outputs of ours are comparable in quality to two state-of-the-art methods: MPU and SP, although our method uses less storage space.

Raw scan data and non-uniform sampling data We have tested our method on the raw scan data, e.g., the Stanford bunny raw
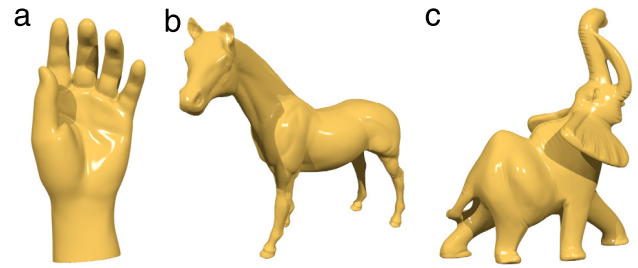

Fig. 6. Reconstructions of (a) the Hand model with 36,619 points, (b) the Horse model with 48,485 points, and (c) the Elephant model with 1,512,290 points.

dataset of 362,272 points assembled from ten range scans as shown in Fig. 8(a). Here, the oriented normals are estimated using the connectivity information within each individual range scan. And as demonstrated in Fig. 8(b), our method is also robust for the highly non-uniformly sampled input data. In this example, the sharp drop of the sampling density does not cause obvious visual artifact in the output of ours, even though the right part of the original Igea model was resampled by removing 90% points.

Noisy data To evaluate the robustness to noise and outliers, we have also applied our method to some point clouds from multiple virtual scans. Fig. 9 shows the reconstruction results with different levels of noise and outliers applied to different regions of the input point cloud. Our method outperforms MPU and SP in terms of the resilience to noise and outliers, because the low-rank regularization term in our fitting model is able to suppress noise and outliers efficiently. Fig. 10 gives another example, in which the histograms of the fitting errors are shown. Again, our method achieves the best result among three methods.

Incomplete data As we might expect, the presented method can reconstruct surfaces from the incomplete data well owing to the multilevel representation using overlays and the low-rank inducing regularization technique. Fig. 11 shows an example.

Running time Table 2 summarizes the running time performance of our method on a variety of models. When compared to some state-of-the-art methods, our current implementation with minimally optimized code takes more computational time. The reasons are as follows. Currently, our main algorithms (ADMM algorithm and ALS algorithm) are implemented by MATLAB which can be speed up by using C/C++ and MKL library. Secondly, accelerating the convergence of the ADMM algorithm and the fast ALS algorithm is also possible. Thirdly, our method is very suitable for parallelization, due to the independence of the MRAS representations in all the cells of each level. Hence, we believe that the running time of our method can be reduced dramatically by considering the above factors. On the other hand, we should emphasize that our principal goal is to find a compact representation of MRAS surfaces, and it is worthwhile to sacrifice some speed.

7. Conclusions and future work

In this paper, we have developed an adaptive surface reconstruction method based on a new implicit representation—Multilevel Rational Algebraic Splines. To generate a compact representation in order to reduce the storage requirement, we propose a local fitting model by introducing a low-rank regularization term. We then convert this model into a convex optimization problem, which can be solved by the ADMM algorithm efficiently. We obtain the compact representation of the MRAS surface using the low-rank tensor approximation technique based on CP decomposition. A number of experimental results have shown that our approach not only produces very compact representations, but also

Table 1
Comparison of the storage requirement for our method with AS, MPU, IPHT, and SP under roughly the same relative error. The storage in each row is normalized by ours to provide easy comparisons.

| Model	AS		MPU		IPHT		SP		Ours	
	Relative	Storage error	Relative	Storage error	Relative	Storage error	Relative	Storage error	Relative	Storage error
Igea	3.42e−3	4.94	2.94e−3	5.33	3.41e−3	169.54	2.96e−3	11.62	2.97e−3	1
Bunny	1.38e−3	5.18	1.72e−3	6.22	1.36e−3	127.90	1.36e−3	35.02	1.42e−3	1
Sphere	1.11e−3	12.45	9.13e−4	32.97	1.05e−3	2293	1.1e−3	36.12	9.68e−4	1
Torus	7.85e−4	3.76	9.23e−4	31.58	8.60e−4	620.18	7.96e−4	33.69	8.32e−4	1
Double torus	3.47e−3	4.22	3.15e−3	15.72	3.62e−3	219.11	3.26e−3	52.63	2.88e−3	1
3-fold torus	5.99e−4	17.26	5.07e−4	266.13	6.51e−4	1412	6.31e−4	498.23	4.68e−4	1
Hand	2.7e−3	4.23	2.66e−3	4.54	2.36e−3	91.71	2.57e−3	10.51	2.45e−3	1
Horse	3.54e−3	4.29	4.01e−3	4.48	3.96e−3	82.46	3.56e−3	13.43	3.41e−3	1
Elephant	2.22e−3	5.21	2.02e−3	4.94	2.05e−3	29.75	1.88e−3	8.28	1.92e−3	1
Dragon	6.1e−3	5.70	6.35e−3	8.32	5.65e−3	23.98	5.19e−3	14.26	5.71e−3	1
Buddha	6.21e−3	4.81	6.02e−3	4.47	6.15e−3	26.08	6.3e−3	9.22	6.18e−3	1

(a) MPU.   (b) SP.   (c) Ours.
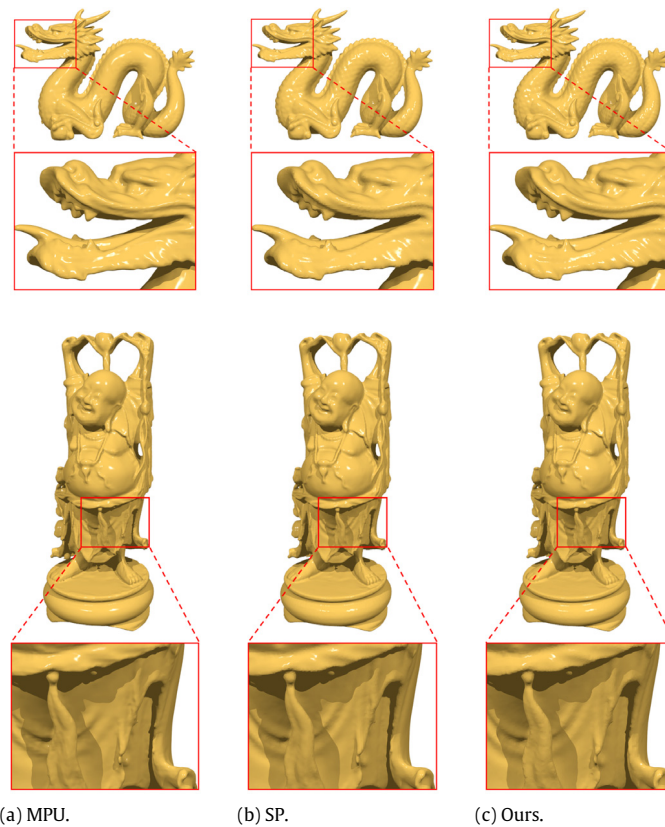
Fig. 7. Reconstructions of the Dragon model and the Happy Buddha model using our method and two state-of-the-art methods: MPU and SP.
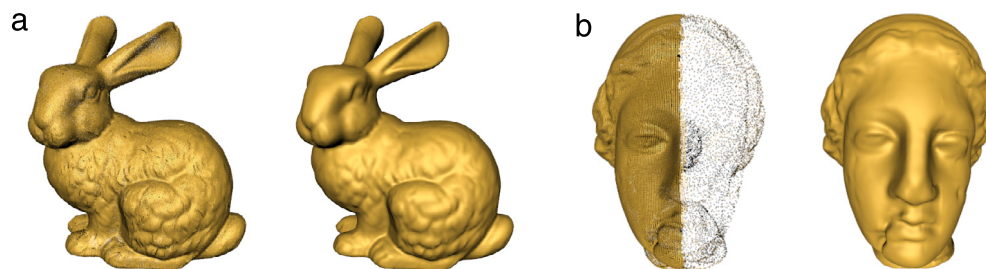

Fig. 8. Reconstructions of (a) the Stanford bunny raw dataset, and (b) the Igea model with non-uniform sampling density.

achieves comparable results with some state-of-the-art surface reconstruction methods. The capability to handle the non-uniform sampling data, the noisy data and the incomplete data has also been evidenced by the numerical examples.

Regarding future work, our method should greatly benefit from the parallelization of our approach on GPU for acceleration, since the calculations of the MRAS representation in all the cells of each level are independent and can be carried out simultaneously. Another interesting direction for acceleration is to develop specialized fast algorithms for solving our local fitting model and performing the CP decomposition. We believe that our method can be useful in many applications, such as mesh
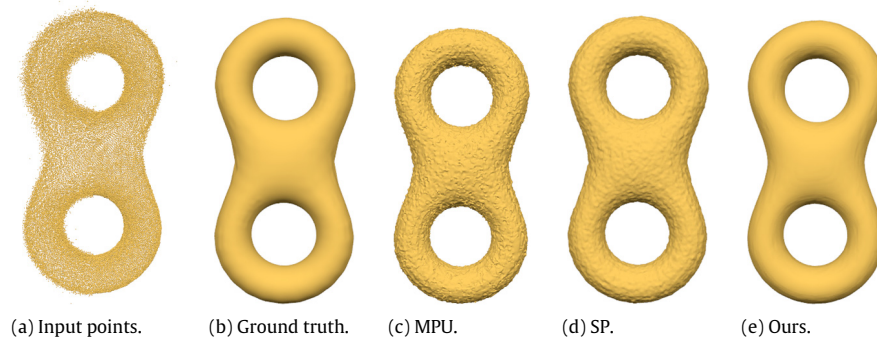
(a) Input points.　　(b) Ground truth.　　(c) MPU.　　(d) SP.　　(e) Ours.

Fig. 9. Reconstructions from the input point cloud of the double torus model corrupted by four different levels of noise ($\sigma = 0.002, 0.006, 0.01, 0.018$) and outliers at bottom-right, top-right, bottom-left and top-left regions, respectively.
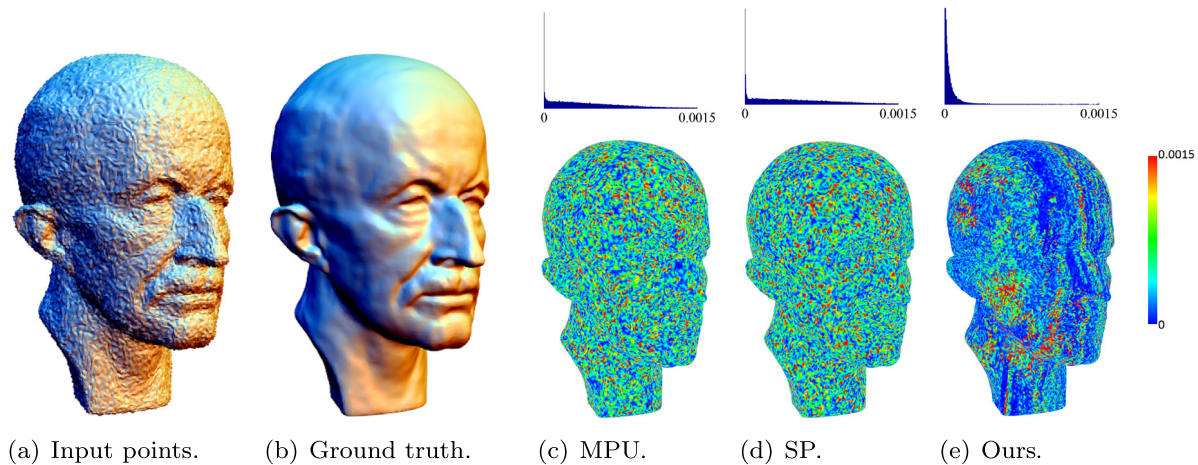

(a) Input points.　　(b) Ground truth.　　(c) MPU.　　(d) SP.　　(e) Ours.

Fig. 10. Reconstruction errors on the noisy scan of the Max Planck model. The reconstruction errors are measured by the distance between the reconstructed mesh and the ground truth and visualized by the color-coding, where the errors are normalized by the diagonal of the bounding box of the ground truth model. The histograms of the errors are also displayed on the top row, which show that our method results in an error distribution closest to the zero.
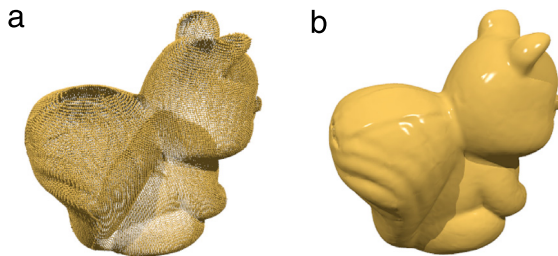

Fig. 11. Reconstruction of the incomplete data: (a) the Squirrel model in the presence of holes on the back and the head. (b) the reconstructed surface with holes filled.

Table 2
Runtime performance of our method on different datasets. All timings are measured in seconds.

Model	#Points	Octree depth	Relative error	Time
Igea	134,345	4	2.97e−3	505.35
Bunny	72,027	4	1.42e−3	248.21
Torus	33,693	1	8.32e−4	5.46
Hand	36,619	4	2.45e−3	194.19
Horse	48,485	4	3.41e−3	242.63
Elephant	1,512,290	5	1.92e−3	1383.64
Dragon	435,545	5	5.71e−3	1257.78
Buddha	537,142	5	6.18e−3	1368.45

compression, level of details and progressive compression and transmission, where the ability to produce compact and multilevel representation is crucial. Applications of tensors in other geometric modeling problems are also worthy of further study.

Acknowledgments

References

[1] Berger M, Tagliasacchi A, Seversky LM, Alliez P, Levine JA, Sharf A, et al. State of the art in surface reconstruction from point clouds. In: Lefebvre S, Spagnuolo M, editors. Eurographics 2014 — state of the art reports. The Eurographics Association; 2014. p. 1–25.

[2] Kazhdan M, Hoppe H. Screened Poisson surface reconstruction. ACM Trans. Graph. 2013;32(3):29.

[3] Cazals F, Giesen J. Delaunay triangulation based surface reconstruction: Ideas and algorithms. In: Effective computational geometry for curves and surfaces. Springer-Verlag; 2006. p. 231–73.

[4] Dey TK. Curve and surface reconstruction: algorithms with mathematical analysis. Cambridge monographs on applied and computational mathematics, Cambridge University Press; 2006.

[5] Bajaj C. Introduction to implicit surfaces. Morgan Kaufmann; 1997.

[6] Gomes A, Voiculescu I, Jorge J, Wyvill B, Galbraith C. Implicit curves and surfaces: mathematics, data structures and algorithms. 1st ed. Springer-Verlag; 2009.

[7] Wang J, Yang Z, Jin L, Deng J, Chen F. Parallel and adaptive surface reconstruction based on implicit pht-splines. Comput Aided Geom Design 2011;28(8):463–74.

[8] Blinn JF. A generalization of algebraic surface drawing. ACM Trans. Graph. 1982;1(3):235–56.

[9] Pratt V. Direct least-squares fitting of algebraic surfaces. ACM SIGGRAPH Comput Graph 1987;21(4):145–52.

[10] Muraki S. Volumetric shape description of range data using blobby model. In: Proceedings of the 18th annual conference on computer graphics and interactive techniques. ACM; 1991. p. 227–35.

[11] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on computer graphics and interactive techniques. ACM; 1992. p. 71–8.

[12] Curless B, Levoy M. A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques. ACM; 1996. p. 303–12.

[13] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Proceedings of the fourth eurographics symposium on geometry processing. Eurographics Association; 2006. p. 61–70.

[14] Manson J, Petrova G, Schaefer S. Streaming surface reconstruction using wavelets. In: Proceedings of the symposium on geometry processing. Eurographics Association; 2008. p. 1411–20.

[15] Levin D. Mesh-independent surface interpolation. In: Brunnett G, Hamann B, Müller H, Linsen L, editors. Geometric modeling for scientific visualization. Springer-Verlag; 2003. p. 37–49.

[16] Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT. Point set surfaces. In: Proceedings of the conference on visualization '01. VIS '01, IEEE Computer Society; 2001. p. 21–8.

[17] Amenta N, Kil YJ. Defining point-set surfaces. In: ACM SIGGRAPH 2004 papers. ACM; 2004. p. 264–70.

[18] Shen C, O'Brien JF, Shewchuk JR. Interpolating and approximating implicit surfaces from polygon soup. In: ACM SIGGRAPH 2004 papers. ACM; 2004. p. 896–904.

[19] Fleishman S, Cohen-Or D, Silva CT. Robust moving least-squares fitting with sharp features. In: ACM SIGGRAPH 2005 papers. ACM; 2005. p. 544–52.

[20] Feng W, Yang Z, Deng J. Moving multiple curves/surfaces approximation of mixed point clouds. Commun Math Stat 2014;2(1):107–24.

[21] Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, et al. Reconstruction and representation of 3d objects with radial basis functions. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM; 2001. p. 67–76.

[22] Turk G, O'brien JF. Modelling with implicit surfaces that interpolate. ACM Trans. Graph. 2002;21(4):855–73.

[23] Morse BS, Yoo TS, Rheingans P, Chen DT, Subramanian KR. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In: ACM SIGGRAPH 2005 courses. ACM; 2005.

[24] Buhmann MD. Radial basis functions. Cambridge University Press; 2003.

[25] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H-P. Multi-level partition of unity implicits. In: ACM SIGGRAPH 2003 papers. ACM; 2003. p. 463–70.

[26] Ohtake Y, Belyaev A, Seidel H-P. Multi-scale and adaptive cs-rbfs for shape reconstruction from clouds of points. In: Dodgson NA, Floater MS, Sabin MA, editors. Advances in multiresolution for geometric modelling. Springer-Verlag; 2005. p. 143–54.

[27] Sederberg TW. Piecewise algebraic surface patches. Comput Aided Geom Design 1985;2(1–3):53–9.

[28] Bajaj CL, Bernardini F, Xu G. Automatic reconstruction of surfaces and scalar fields from 3d scans. In: Proceedings of the 22nd annual conference on computer graphics and interactive techniques. SIGGRAPH '95, ACM; 1995. p. 109–18.

[29] Jüttler B, Felis A. Least-squares fitting of algebraic spline surfaces. Adv. Comput. Math. 2002;17(1–2):135–52.

[30] Yang Z, Deng J, Chen F. Fitting unorganized point clouds with active implicit *b*-spline curves. Vis Comput 2005;21(8–10):831–9.

[31] Rouhani M, Sappa AD. Implicit *b*-spline fitting using the 3l algorithm. In: 18th IEEE international conference on image processing, ICIP. IEEE; 2011. p. 893–6.

[32] Rouhani M, Sappa AD, Boyer E. Implicit b-spline surface reconstruction. IEEE Trans Image Process 2015;24(1):22–32.

[33] Dokken T, Jüttler B. Computational methods for algebraic spline surfaces. Springer-Verlag; 2005.

[34] Deng J, Chen F, Li X, Hu C, Tong W, Yang Z, et al. Polynomial splines over hierarchical *t*-meshes. Graph Models 2008;70(4):76–86.

[35] Markovsky I. Low rank approximation. 1st ed. Springer-Verlag; 2012.

[36] Ma Y, Wright J, Yang AY. Sparse representation and low-rank representation in computer vision. In: ECCV 2012 short course, 2012.

[37] Kolda TG, Bader BW. Tensor decompositions and applications. SIAM Rev 2009; 51(3):455–500.

[38] Grasedyck L, Kressner D, Tobler C. A literature survey of low-rank tensor approximation techniques, 2013. arXiv preprint arXiv:1302.7121.

[39] Wang W, Liu X, Liu L. Upright orientation of 3d shapes via tensor rank minimization. J. Mech. Sci. Technol. 2014;28(7):2469–77.

[40] Mantzaflaris A, Jüttler B, Khoromskij BN, Langer U. Matrix generation in isogeometric analysis by low rank tensor approximation. In: Boissonnat J-D, Cohen A, Gibaru O, Gout C, Lyche T, Mazure M-L, Schumaker LL, editors. Curves and surfaces: 8th international conference. Springer; 2015. p. 321–40.

[41] Xu L, Wang R, Zhang J, Yang Z, Deng J, Chen F, et al. Survey on sparsity in geometric modeling and processing. Graphical Models 2015;82:160–80.

[42] Carroll JD, Chang J-J. Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart–Young decomposition. Psychometrika 1970;35(3):283–319.

[43] Liu J, Musialski P, Wonka P, Ye J. Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. 2013;35(1): 208–20.

[44] Tomioka R, Hayashi K, Kashima H. Estimation of low-rank tensors via convex optimization, 2010. arXiv preprint arXiv:1010.0789.

[45] Forsey DR, Bartels RH. Hierarchical *b*-spline refinement. In: Proceedings of the 15th annual conference on computer graphics and interactive techniques. New York, NY, USA: ACM; 1988. p. 205–12.

[46] Taubin G. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Trans Pattern Anal Mach Intell 1991;13(11):1115–38.

[47] Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 2011;3(1):1–122.

[48] Cai J-F, Candès EJ, Shen Z. A singular value thresholding algorithm for matrix completion. SIAM J Optim 2010;20(4):1956–82.

[49] Phan A-H, Tichavský P, Cichocki A. Fast alternating ls algorithms for high order candecomp/parafac tensor factorizations. IEEE Trans Signal Process 2013; 61(19):4834–46.

[50] Phan A-H, Tichavský P, Cichocki A. Tensorbox: a matlab package for tensor decomposition, 2015. Available online at http://www.bsp.brain.riken.jp/~phan/tensorbox.php.

[51] Bloomenthal J. Olv. 8. Graphics gems IV 4, 324, 1994.