Published in final edited form as:
Image Vis Comput. 2015 January 1; 33: 1–14. doi:10.1016/j.imavis.2014.10.008.

Multiview stereo and silhouette fusion via minimizing generalized reprojection error☆

Zhaoxin Li[a], **Kuanquan Wang**[a,*], **Wenyan Jia**[c], **Hsin-Chen Chen**[b,c], **Wangmeng Zuo**[a], **Deyu Meng**[d], and **Mingui Sun**[b,c]

[a] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

[b] Department of Electrical & Computer Engineering, University of Pittsburgh, Pittsburgh, PA, USA

[c] Department of Neurosurgery, University of Pittsburgh, Pittsburgh, PA, USA

[d] Institute for Information and System Sciences, Xi'an Jiaotong University, Xi'an, China

Abstract

Accurate reconstruction of 3D geometrical shape from a set of calibrated 2D multiview images is an active yet challenging task in computer vision. The existing multiview stereo methods usually perform poorly in recovering deeply concave and thinly protruding structures, and suffer from several common problems like slow convergence, sensitivity to initial conditions, and high memory requirements. To address these issues, we propose a two-phase optimization method for generalized reprojection error minimization (TwGREM), where a generalized framework of reprojection error is proposed to integrate stereo and silhouette cues into a unified energy function. For the minimization of the function, we first introduce a convex relaxation on 3D volumetric grids which can be efficiently solved using variable splitting and Chambolle projection. Then, the resulting surface is parameterized as a triangle mesh and refined using surface evolution to obtain a high-quality 3D reconstruction. Our comparative experiments with several state-of-the-art methods show that the performance of TwGREM based 3D reconstruction is among the highest with respect to accuracy and efficiency, especially for data with smooth texture and sparsely sampled viewpoints.

Keywords

Multiview stereo; 3D reconstruction; Silhouette fusion; Convex relaxation; Reprojection error

1. Introduction

Accurate reconstruction of a 3D geometrical shape from a sequence of calibrated 2D images has many real world applications, such as augmented and mixed reality [1,45], urban reconstruction [2,46], object detection [3,47] and object recognition [4,48]. 3D

☆This paper has been recommended for acceptance by Philippos Mordohai.

* Corresponding author at: School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China. Tel.: +86 451 8641 2871. wangkq@hit.edu.cn (K. Wang)..

reconstruction based on multiple images is an active field of research in computer vision [5,6,42] and has attracted considerable recent interest [40,41,43,44], due to its capability to produce high-quality reconstruction for indoor and outdoor scenes and drastically decreased cost in image acquisition by digital cameras and cellphones.

There are several image cues that can be utilized for multi-image based 3D reconstruction [7], such as texture, defocus, shading, stereo, and silhouette. Among them, stereo cues are the most common ones. The reconstruction based on stereo cues is also known as shape from stereo and multiview stereo. Based on the Lambertian reflectance model of illuminated surface [8], multiview stereo methods can infer a 3D shape by finding local correspondences among the input 2D images. However, due to noise, illumination variation, inaccurate camera calibration and/or lack of texture on the object, correspondences are often misidentified, resulting in degraded 3D reconstruction accuracy and sometimes unpleasant visual effect. Although the introduction of neighbor [9–11] or non-local information [12] may alleviate this problem, these remedies are inadequate in finding reliable correspondences for objects which do not contain sufficient textures. To solve this problem, energy minimization techniques based on weighted area function [13, 14] have been proposed by adding surface smoothness as a regularization term. These techniques lead to compact and smooth surfaces and thus reduce outliers. However, regularization techniques lead to over-smoothed surfaces and thus generate artifacts in certain cases, such as objects with thin or protruding structures.

Another commonly used cue for 3D reconstruction is silhouette. The methods that use silhouette cues to recover 3D surfaces are called *shape from silhouette*. These methods [15–17] aim at reconstructing a visual hull by using multiple silhouette images. The visual hull is the intersection of the visual cones associated with all image silhouettes. For any 3D point inside the visual hull, its projected point in each 2D image is expected to be enclosed by the silhouette. Since the projections of thin or protruding structures are generally located on boundaries of silhouettes, such structures can be recovered in the boundaries of the visual hull. The quality of the visual hull is directly affected by the segmentation quality of the silhouettes. With recent advances in image segmentation methods, visual hull reconstruction has become increasingly accurate and robust. However, there are still problems using this approach, e.g., concavities in a shape cannot be reconstructed from the visual hull at all.

To recover deep concave, thin and protruding structures, we plan to take advantages of both stereo and silhouette-based techniques by integrating these cues in the process of reconstruction. The accuracy of 3D reconstruction not only depends on how the objective function is designed, but also depends on how the surface is represented. Volumetric based methods are capable of freely changing topology of surface and efficient for optimization, but the drawback is their high cost in memory for large volume data. The mesh based methods can represent high-resolution surface without large memory requirements, however, most of them are limited in handling topology changes and sensitive to initial condition. It is thus important to integrate the advantages of volumetric and mesh based methods. In this paper, we present an effective and accurate multiview reconstruction method based on a two-phase optimization for generalized reprojection error minimization (TwGREM). A generalized reprojection error is formulated to integrate stereo and silhouette

consistencies into an energy function, which allows the recovery of concavities while preventing over-smoothing of protruding structures. By utilizing the advantages of the implicit and explicit representations of surfaces, a two-phase optimization method is used to minimize this function. In the first phase, with the implicit representation of surface on 3D volumetric grids and the convex relaxation technology, the method is robust against initial conditions and can be efficiently solved using variable splitting and Chambolle projection with freely changing surface topology. In the second phase, the result is further parameterized and refined on a triangle mesh to produce the high-quality reconstruction output.

The rest of the paper is organized as follows: the related work is reviewed in Section 2. The proposed TwGREM method is presented in Section 3, which includes main concepts of the method, formulation of an energy function, and energy optimization. Experimental results are analyzed in Section 4 before the paper is concluded in Section 5.

2. Related work

In some existing studies, the visual hull was used as an initial surface for optimization, and the constant balloon term was added to the energy function so as to penalize the over-smoothing of the surface [18,19]. However, the preference of large shape volumes of this method makes it difficult to reconstruct deep concavities. This problem can be addressed by replacing the constant balloon term with depth maps [20]. The stereo based method is used to compute depth maps on highly textured regions while the silhouette based method is used to compute depth maps on textureless and occluded regions. Then, these depth maps are merged into an energy function on 3D volumetric grids to optimize the final surface. The reconstruction process of this method is thus split into two sub-problems: computing depth maps and optimizing the energy function on the volumetric grids. However, noise and outliers in depth maps could be propagated to the final result. Liu et al. [39] proposed a depth map estimation and fusion method which uses both silhouette and epipolar geometry to constrain the search for admissible solutions. In particular, in their depth map estimation process, multiple starting points are selected for high-quality multi-scale variational depth estimation. However, the final models need to be generated via meshing point clouds using surface reconstruction algorithm. This may lead to an over-smoothing effect in some thinly protruding structures. Sinha et al. [21] proposed to use a graph cut framework to integrate both silhouette and stereo cues into in a single optimization formulation. Due to the reconstruction being computed on the volumetric grids, memory requirements increase rapidly as the grid resolution increases. The high memory requirements limit the utility of this method. Alternatively, one can dynamically integrate silhouette-aligning forces in each optimization step of stereo reconstruction. Kolev et al. [22] used a regional term on volumetric grids to enforce the silhouette constraint, which is similar to the Chan-Vese model in 2D/3D image segmentation [23,24]. This regional term can be updated during energy optimization, so the occluded regions can be determined based on currently estimated surface. Hernandez et al. [25] first used silhouette-aligning forces on the triangle mesh based deformable surface to provide a robust way to recover protruding structures with lower memory requirement. Since triangle mesh based surface representation may be trapped in local minima, the method needs a good initial condition which may not be

attainable. Cremers and Kolev et al. [26,27] reformulated a weighted area function model with silhouette constraints as a constrained convex optimization problem, which solves a convex function in the admissible convex domain. In order to implement the silhouette-aligning constraint, this method needs to compute the first and last voxels for ray-volume intersection along a visual ray in the preprocessing step and store them in lists. As a result, the size of such data structure and memory requirements grows significantly as the increase of the image resolution. By taking advantage of visibility variation in the derivatives of the reprojection error [28,29], stereo and silhouette consistencies can be naturally integrated. However, since these methods need to consider the variation of surface visibility, the models' complexity makes the optimization procedure susceptible to the local minimum problem.

3. Methods

Inferring a 3D shape can be considered as an inverse problem of imaging. The aim of this problem is to reconstruct an object surface S from multiple 2D images observed in different views. In order to assess the quality of reconstruction, the observed images are reprojected via *S* to generate predicted images. The reprojection error between the observed and predicted images thus provides an effective measure of reconstruction quality and naturally formulates the reconstruction as a minimization problem. It has been shown [29] that this formulation with additional prior information of the surface, e.g., local smoothness, corresponds to the following Bayesian formulation:

$$P(S|D) = \frac{P(D|S)\,P(S)}{P(D)} \quad (1)$$

where *S* is the estimated surface and *D* is the set of observed images. *P*(D) only affects *P*(*S*|*D*) up to a scale, and Eq. (1) can thus be reformulated as: $\log(P(S|D)) \propto \log(P(D|S)) + \log(P(S))$. The log-likelihood term $\log(P(D|S))$ measures the reprojection error, while log-prior term $\log(P(S))$ is the regularization term reflecting the smoothness of the surface.

3.1. Concepts of the TwGREM

In the two-phase optimization method for generalized reprojection error minimization (TwGREM), we maximize the consistency between the original and predicted images by utilizing both stereo and silhouette information. Based on this concept, we redefine the reprojection error as a combination of two error components: a *stereo reprojection error* component and a *silhouette reprojection error* component.

As shown in Fig. 1(a), the *stereo reprojection error* is the same as the traditional reprojection error. The predicted image of an observed image is estimated by first projecting one of neighboring images to the reconstructed surface and then projecting back to the image space of this observed image. We refer to this set of projection operations as *stereo reprojection*, and this predicted image as *stereo predicted image*. The *stereo reprojection error* measures the inconsistency between the *stereo predicted* and the observed images.

Additionally, a set of predicted images are generated by projecting each image to the 3D scene and then projecting back to the same image space, as shown in Fig. 1(b). We refer to this set of projection operations as *silhouette reprojection*, and this predicted image as *silhouette predicted image*. The comparison between the *silhouette predicted image* and the silhouette image of the observed image define a different form of the reprojection error. The depth information cannot be estimated from this reprojection error, since no variation in viewpoints is presented. However, it can measure the consistency of reconstructed surface and silhouette images, indicating whether the reconstructed surface is over-smoothed, as shown in Fig. 2. So, we define this reprojection error as *silhouette reprojection error*.

Introducing the *silhouette reprojection error* generalizes the traditional definition of reprojection error which is only defined on stereo image pairs. The *generalized reprojection error* is thus stereo reprojection error + silhouette reprojection error. Ideally, the minimum of generalized reprojection error can be reached if the estimated surface is the best-consistent with all stereo image pairs and silhouette images. The minimization of the generalized reprojection error can be cast as an energy minimization problem. First, an energy function is constructed based on the generalized reprojection error. Second, this energy function is minimized using a two-phase optimization procedure. Overall, the proposed TwGREM method is illustrated in Fig. 3.

3.2. Construction of the energy function

In this section, we define the stereo reprojection and silhouette reprojection errors and integrate them over all observed images to form a complete energy function.

3.2.1. Stereo reprojection error—The *stereo reprojection error* is defined between image stereo pairs. Let the 3D scene include both the target object and background. The background is assumed to be located at infinity, and ideally its radiance is totally black. Let the surfaces $\bar{S} \subset \mathbb{R}^3$ and $S \subset \mathbb{R}^3$ be the ground truth and the reconstructed surface, respectively. Let $I_i{:}\Omega_i \subset \mathbb{R}^2 \rightarrow \mathbb{R}^m$ be the observed image captured by camera *i*, where *m* = 1 for grayscale images, and *m* = 3 for color images. Let $\pi_i{:}\mathbb{R}^3 \rightarrow \Omega_i$ be a projection from 3D points **x** to 2D pixel **p**. Let $\pi_{i,S}^{-1}{:}\Omega_i \rightarrow \mathbb{R}^3$ be an inverse projection. Let $S_i{=}\pi_{i,S}^{-1}(\Omega_i)$ be the visible part of *S* with respect to image I_i. As illustrated in Fig. 4(a), for a pair of neighboring images I_i and I_j, the visible surfaces are S_i and S_j, respectively, and their shared visible part is $S_{ij}{=}S_i \cap S_j$. For each 3D point **x** on *S*, we define a visible image subset V_x in which visual rays connecting camera centers and **x** are not occluded by other parts of *S* (Fig. 4(b)).

Assuming I_i is a reference image and I_j is a neighboring image, a predicted image $\hat{I}_{i,j,S}$ of I_i is generated by first projecting I_j to *S*, and then reprojecting onto I_i. Obviously, the valid definition domain of predicted image is $\pi_i(S_{ij}) \subset \mathbb{R}^2$, which are the image projections of shared visible surface S_{ij}.

Let $\delta(\mathbf{p})$ be the Kronecker delta that returns 1 for pixel **p** in the region $\pi_i(S_{ij})$ and 0 otherwise. Let $\rho{:}\Omega_i \rightarrow \mathbb{R}$ measures color consistency of reference image I_i and a predicted

image $\hat{I}_{i,j,S}$ with respect to a square window centered at pixel **p**. The energy function that measures *stereo reprojection error* can thus be written as:

$$E_{ij}^{stereo}(S) = \int_{\Omega_i} \delta(\mathbf{p})\, \rho\left(I_i(\mathbf{p}), \hat{I}_{i,j,S}(\mathbf{p})\right) d\mathbf{p} \quad (2)$$

where d **p** is the area measure in the image plane, $E_{ij}^{stereo}(S)$ measures the consistency of reconstructed 3D surface with respect to the observed image in the image pairs. When S^{m+1} is closer to the ground truth surface than S^m, we have the inequality $E_{ij}^{stereo}\left(S^{m+1}\right) < E_{ij}^{stereo}(S^m)$.

3.2.2. Silhouette reprojection error—The silhouette image $H_i : \Omega_i \to \{0, 1\}$ is a binary image whose value is assigned to 1 inside and on the silhouette and 0 otherwise. The predicted image $\hat{I}_i : \Omega_i \subset \mathbb{R}^2$ is obtained using *silhouette reprojection* as follows: visual rays emitted from camera center *i* is back-projected to the scene using π_i^{-1}. These visual rays may intersect either *S* or the background at infinity. If the ray intersects *S*, *S* will be colored by using image I_i. If the ray intersects the background, radiance of background is black by definition. Then, the projection from the scene to image I_i generates the predicted image $\hat{I}_i$.

$$\hat{I}_i(\mathbf{p}) = \begin{cases} I_i & \pi_i^{-1}(\mathbf{p}) \in S \\ 0 & else \end{cases} \quad (3)$$

The silhouette $\hat{H}_i$, a binary image, is defined as:

$$\hat{H}_i(\mathbf{p}) = \begin{cases} 1 & \pi_i^{-1}(\mathbf{p}) \in S \\ 0 & else \end{cases} . \quad (4)$$

Let $\tau : \Omega_i \to \{0, 1\}$ be a binary function measuring the difference between the silhouette image H_i and predicted silhouette image $\hat{H}_i$, 1 for $H_i(\mathbf{p}) \neq \hat{H}_i(\mathbf{p})$, and 0 for $H_i(\mathbf{p}) = \hat{H}_i(\mathbf{p})$. The new energy term that measures *silhouette reprojection error* for image I_i is thus expressed as:

$$E_i^{silhouette}(S) = \int_{\Omega_i} \tau\left(H_i(\mathbf{p}), \hat{H}_i(\mathbf{p})\right) d\mathbf{p} \quad (5)$$

Minimizing Eq. (5) forces the surface towards silhouette consistency since it has lower energy when the surface approximates the visual hull. One may notice that $\hat{H}_i$ can also be obtained by directly projecting all surface points to the image instead of *silhouette reprojection*. However, such direct projection scheme would be inefficient, since it needs the surface to be uniformly discretized and each image projection of surface points has to be considered. Fortunately, our *silhouette reprojection* can be very efficiently implemented by rendering the surface using graphics processing unit (GPU), as shown in Section 3.3.

3.2.3. Objective energy function—By integrating the energy terms for *stereo reprojection error* and the *silhouette reprojection error* on all observed images, the complete objective energy function becomes:

$$E(S)=\sum_i\left(\sum_{j,j\neq i}\int_{\Omega_i}\delta(\mathbf{p})\,\rho\left(I_i(\mathbf{p}),\hat{I}_{i,j,S}(\mathbf{p})\right)d\mathbf{p}+\lambda\int_{\Omega_i}\tau\left(H_i(\mathbf{p}),\hat{H}_i(\mathbf{p})\right)d\mathbf{p}\right) \quad (6)$$

where λ is a tradeoff parameter to adjust the weight of the *silhouette reprojection error*. To simplify Eq. (6), we define Ψ_1 as all the image domains for *stereo reprojection error*, Ψ_2 as all the image domains for the *silhouette reprojection error*, $\tilde{\rho}(\mathbf{p})=\delta(\mathbf{p})\,\rho\left(I_i(\mathbf{p}),\hat{I}_{i,j,S}(\mathbf{p})\right)$, and $\tilde{\tau}(\mathbf{p})=\tau\left(H_i(\mathbf{p}),\hat{H}_i(\mathbf{p})\right)$. To ensure smoothness of the surface, a regularization term is added as an integral of surface area unit $d\sigma$. This regularization term prefers a smooth and compact surface, and is effective in improving the robustness against noises and outliers. The proposed energy function is formulated as:

$$E(S)=\int_{\Psi_1}\tilde{\rho}(\mathbf{p})\,d\mathbf{p}+\lambda\int_{\Psi_2}\tilde{\tau}(\mathbf{p})\,d\mathbf{p}+\kappa\int_S d\sigma \quad (7)$$

where κ is a regularization parameter.

Intuitively, the optimal surface has the maximum of stereo and silhouette consistencies and corresponds to the minimum of energy defined in Eq. (7). However, the first two energy terms in Eq. (7) are integrals over the image domain and the third energy term is an integral over the surface. So its optimization is difficult and the process is easily trapped in a local minimum. It is thus desirable to define an energy criterion in the 3D domain, where extensive study has been conducted to solve the optimization problem.

3.3. Energy minimization

We proposed a two-phase coarse-to-fine method to optimize the energy function in Eq. (7). First, we represent the surface using a characteristic function and reformulate (Eq. (7)) into a new function defined in the 3D space domain. The new function has a form of $TV_g + L_1$ norm [24]. TV_g is a weighted total variation, where weights represent stereo consistency. The total variation (TV) is well known for its edge-preserving smoothness effect in image restoration. TV_g has the smoothness effect like TV regularization and its weights are set based on whether surface edges should be smoothed or preserved. Ideally, 3D points on the true 3D surface have minimal reprojection errors, and thus TV_g prefers to preserve these points because that does not increase the energy of TV_g. In summary, TV_g can preserve the points on the true 3D surface while suppressing noises. The second term is a constraint for silhouette consistency, where L_1 is well known for its robust performance against outliers, and thus is adopted to robustly measure the silhouette reprojection error. Attributed to the recent development of the convex relaxation technique, $TV_g + L_1$ can be solved using variable splitting and Chambolle projection. To further improve the quality of the result, we refine the optimized surface based on triangle mesh to obtain a high-quality estimate. We assume that all the silhouette images are available via image segmentation algorithm [33], and thus the visual hull can be used as the initial surface for this two phase optimization.

3.3.1. First phase optimization: optimization on volumetric grids

3.3.1.1. TVg + L_1 norm representation: For a $N_x \times N_y \times N_z$ bounding box B, $\varphi:\mathbb{R}^3 \rightarrow \{0,1\}$ be a characteristic function with value, 1 for the 3D points inside/on the visual hull and 0 otherwise. Let $u:\mathbb{R}^3 \rightarrow \{0,1\}$ be another characteristic function, which equals 1 for the 3D point inside/on the surface and 0 otherwise.

For a 3D point **x**, its *stereo reprojection error* is the summation of *stereo reprojection errors* over all the image projections in visible image subset:

$$g(\mathbf{x}) = \sum_i \sum_{j,j\neq i} \tilde{\rho}(\pi_i(\mathbf{x})) = \sum_i \sum_{j,j\neq i} \delta(\pi_i(\mathbf{x}))\, \rho\left(I_i(\pi_i(\mathbf{x})), \hat{I}_{i,j,S}(\pi_i(\mathbf{x}))\right). \quad (8)$$

Similarly, the *silhouette reprojection error* of **x** is:

$$f(\mathbf{x}) = \sum_i \tilde{\tau}(\pi_i(\mathbf{x})) = \sum_i \tau\left(H_i(\pi_i(\mathbf{x})), \hat{H}_i(\pi_i(\mathbf{x}))\right). \quad (9)$$

Based on Eqs. (8) and (9), and the fact that $d\mathbf{p} = -\frac{\mathbf{x}\cdot\mathbf{n}}{(\mathbf{x}_z)^3} d\mathbf{x}$ [28], Eq. (7), which is defined in image domain, can be rewritten as an integral over the volume:

$$E(u) = \int_{\mathbb{R}^3} -\frac{\mathbf{x}\cdot\mathbf{n}}{(\mathbf{x}_z)^3} g(\mathbf{x}) |\nabla u| d\mathbf{x} + \lambda \int_{\mathbb{R}^3} -\frac{\mathbf{x}\cdot\mathbf{n}}{(\mathbf{x}_z)^3} f(\mathbf{x}) \|u(\mathbf{x}) - \varphi(\mathbf{x})\|_1 d\mathbf{x} \quad (10)$$

where $|\nabla_u| = \sqrt{(\mathbf{D}\mathbf{x}_x)^2 + (\mathbf{D}\mathbf{x}_y)^2 + (\mathbf{D}\mathbf{x}_z)^2}$, and $\mathbf{x}_x$, $\mathbf{x}_y$, $\mathbf{x}_z$ are three components of 3D point **x** in the X, Y and Z axes, **D** is the gradient operator, **n** is the unit normal of **x**, and $\|\cdot\|_1$ is L_1 norm. Let $\tilde{g}(\mathbf{x},\mathbf{n}) = -\frac{\mathbf{x}\cdot\mathbf{n}}{(\mathbf{x}_z)^3} g(\mathbf{x})$, and $\tilde{f}(\mathbf{x},\mathbf{n}) = -\frac{\mathbf{x}\cdot\mathbf{n}}{(\mathbf{x}_z)^3} f(\mathbf{x})$. The minimization of $E(u)$ can be represented as:

$$u_b = \min_{u\in\{0,1\}} \left(\int_{\mathbb{R}^3} \tilde{g}(\mathbf{x},\mathbf{n}) |\nabla u| d\mathbf{x}_{TV_g} + \lambda \int_{\mathbb{R}^3} \tilde{f}(\mathbf{x},\mathbf{n}) \|u(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{x})\|_1 d\mathbf{x}_{L_1} \right) \quad (11)$$

Eq. (11) has a form of $TV_g + L_1$. Since the TV model inherently enforces the regularization, an explicit smoothness term is unnecessary.

3.3.1.2. Minimization of TVg + L_1: The energy criterion $E(u)$ is defined in a non-convex set function because the characteristic function u is defined in a non-convex set, i.e., binary set. The optimization for non-convex function is easily trapped in a local minimum. The principle of convex relaxation is to relax u from a non-convex set {0, 1} to a continuous interval [0, 1] which is a convex set, and thus the corresponding function $\tilde{E}(u)$ becomes a convex function:

$$u_c = \min_{u\in[0,1]} \left(\int_{\mathbb{R}^3} \tilde{g}(\mathbf{x},\mathbf{n}) |\nabla u| d\mathbf{x} + \lambda \int_{\mathbb{R}^3} \tilde{f}(\mathbf{x},\mathbf{n}) \|u(\mathbf{x}) - \varphi(\mathbf{x})\|_1 d\mathbf{x} \right). \quad (12)$$

Because of the non-smoothness property of $|\nabla u|$, direct optimization of $\tilde{E}(u)$ is not trivial. According to the Chambolle projection method [24, 32], $\tilde{E}(u)$ can be reformulated with an auxiliary variable v by using variable splitting approach as:

$$\tilde{E}(u,v) = \int_{\mathbb{R}^3} \tilde{g}(\mathbf{x},\mathbf{n}) |\nabla u| d\mathbf{x} + \frac{1}{2\theta} \int_{\mathbb{R}^3} (u+v-\varphi)^2 d\mathbf{x} + \lambda \int_{\mathbb{R}^3} \tilde{f}(\mathbf{x},\mathbf{n}) \|v\|_1 d\mathbf{x} \quad (13)$$

where the parameter $\theta(\theta > 0)$ is set to be a small value, and so that $1/2\theta$ is sufficiently large in order to constrain ϕ close to $u + v$. Since $\tilde{E}(u,v)$ is convex, its minimizer can be computed by minimizing $\tilde{E}(u,v)$ with respect to u and v separately. The process is iteratively performed until convergence. Thus, the following minimization procedure is utilized:

1. Searching for optimal u when v is fixed by:

$$\min_u \left\{ \int_{\mathbb{R}^3} \tilde{g}(\mathbf{x},\mathbf{n}) |\nabla u| d\mathbf{x} + \frac{1}{2\theta} \int_{\mathbb{R}^3} (u+v-\varphi)^2 d\mathbf{x} \right\} \quad (14)$$

2. Searching for optimal v when u is fixed by:

$$\min_v \left\{ \frac{1}{2\theta} \int_{\mathbb{R}^3} (u+v-\varphi)^2 d\mathbf{x} + \lambda \int_{\mathbb{R}^3} \tilde{f}(\mathbf{x},\mathbf{n}) \|v\|_1 d\mathbf{x} \right\}. \quad (15)$$

Eq. (14) can be optimized using Chambolle projection with convergence rate $O(1/n^2)$, and Eq. (15) can be solved point-wise for v. After convex optimization on relaxed convex set, the output u_c is projected to binary set $\{0, 1\}$, which can be implemented by using a constant threshold $\mu \in (0, 1)$:

$$u_b(\mathbf{x}) = \begin{cases} 1 & u_c(\mathbf{x}) > \mu \\ 0 & u_c(\mathbf{x}) \leq \mu \end{cases} \quad (16)$$

where u_b approximates a minimizer of non-convex energy function in Eq. (11) and labels each $\mathbf{x} \in B$, 1 for a point inside surface, 0 for a point outside surface. The calculations of $\tilde{g}$ and $\tilde{f}$ depends on the consistency measure of observed and predicted images via reconstructed surface S, respectively, so they should be updated after generating the newly reconstructed surface. After $\tilde{g}$ and $\tilde{f}$ are updated, the above optimization is again performed to update the reconstructed surface. Overall, we alternatively update $\tilde{g}$, $\tilde{f}$ and S until the optimization converges.

In order to generate predicted images, the *stereo reprojection* and *silhouette reprojection* are used as shown in Fig. 1. To implement these projection operations, a triangle mesh is extracted from the zero level set of u_b to represent the reconstructed surface. In our study, we use an efficient GPU-based marching cubes algorithm [34] to extract the triangle mesh. Then, these projection operations are performed efficiently by rendering the triangle mesh using projective texture mapping on the GPU [35]. However, for $\mathbf{x} \notin S$, we cannot use this method to generate the *stereo predicted image*. Instead, a small planar patch P is generated

which intersects at **x**. The normal and visible image subset of patch *P* is set the same to its nearest point on *S*. Then, for any image pairs in visible image subset, we set one as the reference image. The *stereo predicted image* for a small patch in the reference image is generated by projecting another image to 3D patch *P* and then to the reference image. In the implementation, we update the g and f in each 500 and 10 iterations, respectively.

The computational procedure of first phase optimization of the TwGREM is provided as follows:

3.3.2. Second phase optimization: optimization on triangle mesh—For the first phase optimization, in order to generate a high-resolution and high-quality model, it is necessary to use high-resolution volumetric grids, which lead to a large memory requirements. In contrast, the triangle mesh based explicit representation can provide a high-resolution estimate with fine details and low memory requirements, since only the parameterized surface is stored in the memory.

Let *S* be parameterized to a triangle mesh *M*, with a set of indexes $\mathscr{V}=\{v_1, v_2, \ldots v_n\}$ representing *n* vertices, and a set of triangular faces $\mathscr{F}=\{f_1, f_2, \ldots f_n\}$ connecting them, $f_i \in \mathscr{V} \times \mathscr{V} \times \mathscr{V}$. The geometric embedding of a triangle mesh into $\mathbb{R}^3$ is specified by associating a 3D position $\mathbf{x}_{v_i}$ to each vertex $v_i \in \mathscr{V}$. The evolution equation for surface refinement on the triangle mesh is given as[1]:

$$\frac{\partial M}{\partial t}(\mathbf{x}_{v_i}) = -\kappa \boldsymbol{\omega}_{v_i} + \left(\mu_{v_i}^{min} + \lambda \beta_{v_i}\right) \mathbf{n}_{v_i} \quad (17)$$

with

$$\mu_{v_i}^{min} = arg\min_{\mu} \left(\sum_{v_j \in \mathscr{N}(v_i)} W_{v_j} \left| \boldsymbol{\xi}_{v_j} - \mu \right| \right) \quad (18)$$

$$W_{v_j} = exp\left(\frac{-\|\mathbf{x}_{v_i} - \mathbf{x}_{v_j}\|_2}{\gamma} \right) \quad (19)$$

where $v_j \in \mathscr{N}(v_i)$ is a vertex in the neighborhood of vertex v_i on the triangle mesh *M*, $\mathbf{n}_{v_i}$ is the surface normal of vertex v_i, ξ_{v_i} is the derivative of the *stereo reprojection error* for vertex v_i, $\mu_{v_i}^{min}$ minimizes the weighted sum of the differences over ξ_{v_i} within the neighborhood of vertex v_i, W_{v_j} is the weight for each vertex $v_j \in \mathscr{N}(v_i)$, γ is the parameter to adjust the influence of weight, $\boldsymbol{\omega}_{v_i}$ is a discrete Laplace-Beltrami operator for surface regularization, and β_{v_i} is the term considering the *silhouette reprojection error*. Eq. (18) can be easily and exactly solved by weighted median filtering [36]. In order to generate predicted images for the calculation of stereo and silhouette reprojection error, the triangle mesh *M* is first rendered by the *stereo reprojection* operation and then by the *silhouette*

[1]We have put the details of deriving evolution equation in the Appendix A of the paper.

reprojection operation using projective texture mapping on GPU. Because we do not consider the points outside the surface, the patch-based projection operation for *stereo predicted image* (as illustrated in first phase optimization) do not need to be performed. Thus, the total process for generating predicted images is very efficient. The derivative of *stereo reprojection error* ξ_{v_i} is calculated according to [30,31].

The evolution of the triangle mesh is conducted based on Eq. (17) by using gradient decent with a small time step *t*. The advantages of Eq. (17) are: 1) the integration of a silhouette term that penalizes the vertex which violates the constraint of the silhouette consistency; 2) the enforcement of a smoother surface by considering the entire vertexes in the neighborhood of vertex v_i, 3) the robustness of L_1 norm to noise and outliers. In practice, the neighborhood $\mathcal{N}(v_i)$ can be set as a first or second order ring of the triangular faces around the vertex in the triangle mesh.

The computational procedure of the second phase optimization of the TwGREM is listed below:

3.3.3. Complexity analysis—In the first phase optimization, the convex relaxation of $TV_g + L_1$ energy function is adopted. With abuse of the notation, denote the size of the volume, i.e. the number of the volumetric grids, by $N = N_x \times N_y \times N_z$, the number of input images by M. Due to capability to handle the illumination discrepancy problem, the Normalized Cross Correlation (NCC) was used to measure the *stereo reprojection error*. Suppose that the computation of NCC is proportional to the window size w, and each voxel is visible on V image pairs in average. As illustrated in Algorithm-1, there are at most three steps per iteration:

1. **Minimization of** Eq. (11): This step involves three operations, i.e., Chambolle projection, soft-thresholding, and projection to binary set. The complexity of each operation are O(N), and thus the total complexity of this step is O(N).
2. **Marching cubes**: In the worst-case, the complexity of marching cubes algorithm is $O(N)$ for iso-surface extraction. Note that we adopt a GPU-based parallel implementation [34] in our experiments, which makes the running time of this step is negligible while compared with the other steps.
3. **Generation of predicted images**: The main complexity of this step is from the computation of *silhouette reprojection error* and *stereo reprojection error*, which are $O(NM)$ and $O(NVMw^2)$, respectively. It should be noted that, in our implementation, *stereo reprojection error* is updated after each $C_s = 500$ iterations, and *silhouette reprojection error* is updated after each $C_h = 10$ iterations.

Denote the number of iterations of Algorithm-1 by n_1. Taking these three steps into account, the overall complexity of Algorithm 1 is $O(n_1(N + NM/C_h + NVMw^2/C_s))$.

In the second phase optimization, a triangle mesh based surface evolution algorithm is adopted to refine the reconstructed model. Denote the size of triangle mesh by T, i.e. the number of vertices, and the resolution of images by R. Suppose that each vertex on the

triangle mesh is visible on U image pairs in average. As illustrated in Algorithm-2, there are two steps per iteration:

1. **Evolve the triangle mesh with a small time step:** This step involves two operations, i.e., deforming each vertex of triangle mesh to a new position according to a speed v and a small time step t, and smoothing the triangle mesh using discrete Laplace-Beltrami operator. The complexity of each operation are $O(T)$, and thus the total complexity of this step is $O(T)$.
2. **Generation of predicted images**: The main complexity of this step is from the computation of *silhouette reprojection error* and *stereo reprojection error*, which are $O(TM)$ and $O(TUMR)$, respectively. In particular, the calculation of NCC for the *stereo reprojection error* can be independent to window size w by using image integral technology.

Denote the number of iterations of Algorithm-2 by n_2. Taking these two steps into account, the overall complexity of Algorithm-2 is $O(n_2(TM + TUMR))$.

For the image data used in the experiments, n_1 is typically set to 2000 for first phase optimization and n_2 is typically set to 200 for the second phase optimization.

4. Experimental results

To evaluate the performance of the TwGREM method, we implemented our algorithm using C++ with OpenGL, OpenCV and CGAL libraries and tested this method using the Middlebury multiview stereo benchmark [5] and several other public datasets.

For each test data, Table 1 lists the number of images, image resolution and running times of algorithm. The window size for calculating NCC was set to 5 × 5 pixels. The visibility of 3D points at the surface was determined using the Z-buffer of OpenGL. The λ is used to balance the weight of stereo and silhouette reprojection. In all experiments in the paper, we have fixed the λ to 0.4 in the first phase optimization and fixed it to $0.2 \times l_e$ for the second phase optimization, where l_e is the average edge length of triangle mesh. The κ is used to control the weight of smoothness term. Because the TV norm has implicit smoothness effect, the explicit smoothness term is not necessary for the first phase optimization. And in the second phase optimization, the value of κ is set to 0.2 for all the datasets in the experiments. For the first phase optimization, the resolution of volumetric grids was first set to 64 × 64 × 64 and then increased to 128 × 128 × 128. The output of optimization on the low resolution setting was used as initial points for the optimization on the high resolution setting. The second phase optimization was performed on the resulting surface from the first phase. The running times of TwGREM for these datasets on a laptop computer with Intel Core i7 2.40 GHz CPU are between 40 and 150 min depending on the size of datasets and the image resolution.

4.1. Middlebury datasets

The Middlebury benchmark consists of two objects: a dinosaur and a temple, as shown in Fig. 5. The dinosaur dataset is characterized by smooth and low texture surfaces while the temple dataset composes of high texture surfaces. According to the number of images

sampled around the object, the standard benchmark datasets can be divided into sparse ring, ring, and full datasets. We tested our algorithm on the *dino sparse ring*, the *dino ring*, the *temple sparse ring*, and the *temple ring* datasets, respectively. The ground truth data of the two objects were obtained using a high-accuracy laser scanner.

The intermediate and final results for *dino sparse ring* are illustrated in Fig. 6. The visual hull was set as the initial point (Fig. 6(a)). The resulting surface from the first phase optimization with resolution 64 × 64 × 64 is shown in Fig. 6(b). It was a coarse approximation of the object's actual surface. Further optimization on the 128 × 128 × 128 volumetric grids generated a more accurate result as shown in Fig. 6(c). The resulting surface was further refined by the second phase optimization on triangle mesh as shown in Fig. 6(d). The same intermediate and final results for the *temple sparse ring* dataset are illustrated in Fig. 7.

For *dino ring and temple ring* datasets, the reconstruction procedure is similar to that for *dino sparse ring and temple sparse ring*. Due to limited space of the paper, only final results for these two datasets are shown in Fig. 8.

The proposed method was also quantitatively evaluated based on two performance indicators: accuracy and completeness. These two indicators are used to describe how close the reconstructed surface *S* is to the ground truth of surface, and how much of the ground truth is modeled by *S*, respectively. The accuracy threshold was set to 1.25 mm and completeness threshold was set to 90% in these evaluations. The accuracy and completeness of our results for *temple ring, dino ring, temple sparse ring*, and *dino sparse ring* data sets are listed in Table 2. All these evaluation results are available publicly on the Middlebury evaluation page [37]. We also compared the proposed method with a variety of methods[2] listed in evaluation page. Due to the limited space, we only show the results for *dino sparse ring* and *dino ring* in Fig. 9. The good performance of dinosaur dataset demonstrated our method is suitable for objects with low and smooth textures. Furthermore, we selected stereo-silhouette based methods in the evaluation list and compared them with our method as shown in Table 2. Since some reconstruction results were not reported by authors on certain datasets, we labeled them as '×'. From the comparison results, it can be seen that the proposed method achieves good performance on accuracy and completeness among all the datasets. For *dino sparse ring* and *temple sparse ring* datasets, the proposed method outperforms other methods. As shown in Table 2, more observed images can generate more accurate reconstruction. Thus, the accurate 3D reconstruction using *dino sparse ring* and *temple sparse ring* dataset are more challenging. The high ranks of our method on the *dino sparse ring* and *temple sparse ring* datasets indicate that this method is very competitive in 3D reconstruction using datasets with sparsely sampled viewpoints.

To demonstrate the influence of the *silhouette reprojection error*, we removed the energy term of *silhouette reprojection error* from the energy function for the two phases of the optimization by setting λ to 0, respectively. For the first phase optimization, we assumed that the initial surface was the visual hull. As shown in Fig. 10, with the increase of iteration

[2]These methods that we used in the experiment were reported in the evaluation website by the time of this paper submission.

steps, the surface was obviously over-smoothed. For the second phase optimization, we assumed that a good initial surface had already been obtained from the first phase optimization. The reconstructed results were compared with the results which includes energy term of *silhouette reprojection error* in the optimization procedure, as shown in Fig. 11.

4.2. Other datasets

To further evaluate the proposed method, we applied our algorithm to several other public datasets: *twins* [25], *Beethoven* [38], *bunny* [38], *bird* [38] and *captain* [20]. The final experimental results with these datasets are illustrated in Fig. 12. Groundtruths for these datasets were not available, and thus qualitative evaluation of appearance of the reconstruction results was adopted in the experiments.

Each of the *twins* and *captain* datasets include 36 high resolution images of a compact, smooth and low textured object which was captured by using a fixed camera and a turntable where the object was posed. The experimental results in Fig. 12(a) and (b) indicate that the depth concavity and small protrusion on the objects have been well recovered. The *Beethoven*, *bunny* and *bird* datasets include 35, 36, 21 images, respectively. They were captured by a set of synchronized cameras. The *Beethoven* dataset presents textureless surface, and the *bunny* dataset presents homogenous/smooth surface. The results on these two datasets are shown in Fig. 12(c-d). The proposed method integrates surface regularization in the energy function, and thus is able to handle noise and outliers effectively, even for the textureless cases. The *bird* dataset includes only 21 images of highly textured images (Fig. 12(e)). The small protrusion such as the wings, claws and head of bird model affect the border of predicted silhouette images, and thus the *silhouette reprojection error* prevents them from being over-smoothed. On the contrary, some details of feathers were not kept. The reason may be that these small details do not affect the consistency of silhouette thus minimizing the *silhouette reprojection error* cannot prevent them from being smoothed.

4.3. Limitations and future work

There are some limitations of the TwGREM. First, the main assumption of TwGREM is that silhouettes have been accurately segmented. However, for some scenes, such as outdoor scenes, it is probably difficult to accurately segment silhouettes based on images. The inaccuracy of silhouettes will affect the quality of reconstruction. In the future, we will put the algorithms of silhouette segmentation and 3D reconstruction into one unified framework. In other words, we use the reconstructed 3D model to guide the segmentation of silhouettes and then use the refined silhouettes for better 3D reconstruction. Second, the main time consume of the algorithm is on the second phase optimization. To refine a high-resolution model, the each evolution step should be small and thus convergence of algorithm is slow. For *dino ring*, TwGREM takes 20 minutes on the first phase optimization and takes about 70 minutes on the second phase optimization. Because the computation step for each vertex is independent, it is feasible to perform the computation of each vertex on GPU kernels using CUDA parallel strategy.

5. Conclusion

In this paper, we proposed a two-phase optimization method for generalized reprojection error minimization (TwGREM). First, a new energy function is formulated based on minimizing the *generalized reprojection error*, which integrates both stereo and silhouette cues. This function can be effectively minimized using a convex relaxation of the model on 3D volumetric grids, and then refined using surface evolution on a triangle mesh. TwGREM can reconstruct regions with concavities and protrusions, and is robust against initialization. When compared with the state-of-the-art methods, TwGREM is especially competitive in data with smooth textures and sparsely sampled views. Moreover, due to both implicit and explicit representations of surface are used in the optimization procedure, TwGREM is able to produce high-resolution and high-quality 3D reconstruction effectively, adaptive to surface topology and suitable for GPU implementation with a low memory requirements.

Acknowledgements

The authors thank Dr. Daniel Scharstein for evaluating our results on Middlebury datasets, Dr. Carlos Hernández and Dr. Peng Song providing us the *twins* and the *captain* datasets, respectively. This work was supported by National Institutes of Health Grant No. R01CA165255 of the United States, and the National Natural Science Foundation of China under Grant No. 61173086 and 61271093.

Appendix A

A.1. Deriving evolution equation defined in Eq (17)

The surface estimated from the first phase optimization of TwGREM is still a coarse approximation of the true surface of 3D object. This is because of the limitation of volumetric method. It has a large memory requirements for the high-quality surface reconstruction. On the contrary, mesh-based method can reconstruct the coarse surface in a high-quality with much less memory requirements, and is ideal for surface refinement. The purpose of our evolution equation is to iteratively deform the current surface to the true 3D surface. The deformation of surface can be implemented by minimizing the generalized reprojection errors in gradient decent manner. In this process, we first estimate the derivative of the energy function (Eq. (7)) with respect to 3D surface point $\mathbf{x}$:

$$\frac{\partial E}{\partial \mathbf{x}} = \frac{\partial \left(\int_{\Psi_1} \tilde{\rho}\left(\mathbf{p}\right) d\mathbf{p} \right)}{\partial \mathbf{x}} + \lambda \frac{\partial \left(\int_{\Psi_2} \tilde{\tau}\left(\mathbf{p}\right) d\mathbf{p} \right)}{\partial \mathbf{x}} + \kappa \frac{\partial \left(\int_{S} d\sigma \right)}{\partial \mathbf{x}}.$$

Then based on the Euler-Lagrange equation, the evolution equation can be derived as:

$$\frac{\partial S}{\partial t}\left(\mathbf{x}\right) = -\kappa \boldsymbol{\omega} + \left(\mu + \lambda\beta\right)\mathbf{n}$$

where μ, β and $\boldsymbol{\omega}$ are derivative of stereo reprojection error term, silhouette reprojection error term and smoothness term, respectively. Because the surface S is represented using triangle mesh M, the evolution equation can be rewritten as:

$$\frac{\partial M}{\partial t}(\mathbf{x}_{v_i}) = -\kappa\boldsymbol{\omega}_{v_i} + (\mu_{v_i} + \lambda\beta_{v_i})\,\mathbf{n}_{v_i}$$

where $\mathbf{x}_{v_i}$ is the vertex of triangle mesh M. To add the stability of evolution, we replace the μ_{v_i} with $\mu_{v_i}^{min}$, which is defined in Eq. (18).

$$\mu_{v_i}^{min} = \arg\min_{\mu} \left(\sum_{v_j \in \mathcal{N}(v_i)} W_{v_j} \left| \boldsymbol{\xi}_{v_j} - \mu \right| \right)$$

where $v_j \in \mathcal{N}(v_i)$ is a vertex in the neighborhood of vertex v_i on the triangle mesh M. W_{v_j} is the weight for each vertex $v_j \in \mathcal{N}(v_i)$. Eq. (18) is a weighted L_1 minimization problem which can be efficiently estimated by weighted median.

References

1. Bradley D, Popa T, Sheffer A, Heidrich W, Boubekeur T. Markerless garment capture. ACM Trans. Graph. 2008; 27(3)
2. Lafarge F, Keriven R, Bredif M, Vu H. Hybrid multiview reconstruction by Jump-Diffusion. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2010:350–357.
3. Yan P, Khan SM, Shah M. 3D model based object class detection in an arbitrary view. Proceedings of the 11th International Conference on Computer Vision. 2006:1–6.
4. Kushal A, Ponce J. Modeling 3D objects from stereo views and recognizing them in photographs. Proceedings of the 9th European Conference on Computer Vision. 2006:563–574.
5. Seitz S, Curless B, Diebel J, Scharstein D, Szeliski R. A comparison and evaluation of multiview stereo reconstruction algorithms. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2006:519–526.
6. Strecha C, von Hansen W, Van Gool L, Fua P, Thoennessen U. On benchmarking camera calibration and multiview stereo for high resolution imagery. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2008:1–8.
7. Mada SK, Smith MK, Smith LN, Midha S. An overview of passive and active vision techniques for hand-held 3D data acquisition. Opt. Eng. 2010; 39(1):10–22.
8. Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis. 2002; 47(1/2/3):7–42.
9. Yoon K, Kweon I. Adaptive support-weight approach for correspondence search. IEEE Trans. Pattern Anal. Mach. Intell. 2006; 28(4):650–656. [PubMed: 16566513]
10. He K, Sun J, Tang X. Guided image filtering. Proceedings of the 11th European Conference on Computer Vision. 2010:1–14.
11. Rhemann C, Hosni A, Bleyer M, Rother C, Gelautz M. Fast cost-volume filtering for visual correspondence and beyond. IEEE Conf. Comput. Vis. Pattern Recognit. 2011
12. Yang Q. A non-local cost aggregation method for stereo matching. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2012:1402–1409.
13. Faugeras O, Keriven R. Variational principles, surface evolution, PDEs, level set methods, and the stereo problem. IEEE Trans. Image Process. 1998; 7(3):336–344. [PubMed: 18276253]
14. Goldlücke B, Ihrke I, Linz C, Magnor M. Weighted minimal hypersurface reconstruction. IEEE Trans. Pattern Anal. Mach. Intell. 2007; 29(7):1194–1208. [PubMed: 17496377]
15. Szeliski R. Rapid octree construction from range sequences. Comput. Vis. Graph. Image Process. 1993; 58(1):23–32.
16. Matusik W, Buehler C, McMillan L. Polyhedral visual hulls for real-time rendering. Proc. 12th Eurographics Workshop on Rendering. 2001:115–125.

17. Tarini M, Callieri M, Montani C, Rocchini C, Olsson K, Persson T. Marching intersections: an efficient approach to shape-from-silhouette, Proceedings of 7th International Fall Workshop on Vision. Modeling, and Visualization. 2002:283–290.
18. Vogiatzis G, Hernández C, Torr PHS, Cipolla R. MultiView stereo via volumetric Graph-Cuts and occlusion robust photo-consistency. IEEE Trans. Pattern Anal. Mach. Intell. 2007; 29(12):2241–2246. [PubMed: 17934232]
19. Lempitsky V, Boykov Y, Ivanov D. Oriented visibility for multiview reconstruction. Proceedings of the 9th European conference on Computer Vision. 2006:226–238.
20. Song P, Wu X, Wang M. Volumetric stereo and silhouette fusion for image-based modeling. Vis. Comput. 2010; 26(12):1435–1450.
21. Sinha S, Mordohai P, Pollefeys M. Multiview stereo via graph cuts on the dual of an adaptive tetrahedral mesh. Proc. IEEE Int. Conf. Comput. Vis. Oct.2007
22. Kolev K, Klodt M, Brox T, Cremers D. Continuous global optimization in multiview 3D reconstruction. Int. J. Comput. Vis. 2009; 84(1):80–96.
23. Chan TF, Esedoglu S, Nikolova M. Algorithms for finding global minimizers of image segmentation and denoising models. SIAM J. Appl. Math. 2006; 66(5):1632–1648.
24. Bresson X, Esedoglu S, Vandergheynst P, Thiran JP, Osher S. Fast global minimization of the active contour/snake model. J. Math. Imaging Vis. 2007; 28(2):151–167.
25. Hernandez C, Schmitt F. Silhouette and stereo fusion for 3D object modeling. Comput. Vis. Image Underst. 2004; 96(3):367–392.
26. Cremers D, Kolev K. Multiview stereo and silhouette consistency via convex functionals over convex domains. IEEE Trans. Pattern Anal. Mach. Intell. 2011; 33(6):1161–1174. [PubMed: 20820076]
27. Kolev K, Pock T, Cremers D. Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. Proceedings of the 11th European Conference on Computer Vision. 2010:538–551.
28. Gargallo P, Prados E, Sturm P. Minimizing the reprojection error in surface reconstruction from Images. Proc. Int. Conf. Comput. Vis. 2007:1–8.
29. Delaunoy A, Prados E, Gargallo P, Pons JP, Sturm P. Minimizing the multiview stereo reprojection error for triangular surface meshes. Proc. Br. Mach. Vis. Conf. 2008:1–10.
30. Pons JP, Keriven R, Faugeras O. Multiview stereo reconstruction and scene flow estimation with a global image-based matching score. Int. J. Comput. Vis. 2007; 72(2):179–193.
31. Zaharescu A, Boyer E, Horaud RP. TransforMesh: a topology-adaptive mesh-based approach to surface evolution. Proc. Asian Conf. Comput. Vis. 2007:166–175.
32. Chambolle A. An algorithm for total variation minimization and applications. J. Math. Imaging Vis. 2004; 20(1–2):89–97.
33. Rother C, Kolmogorov V, Blake A. GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 2004; 23(3):309–314.
34. Dyken C, Ziegler G, Theobalt C, Seidel HP. High-speed marching cubes using HistoPyramids. Comput. Graph. Forum. 2008; 27(8):2028–2039.
35. Segal M, Korobkin C, Widenfelt R, Foran J, Haeberli P. Fast shadows and lighting effects using texture mapping. Comput. Graph. 1992; 26(2):249–252.
36. Rauh A. A fast weighted median algorithm based on Quickselect. Proceedings of the17th IEEE International Conference on Image Processing (ICIP). 2010:105–108.
37. http://vision.middlebury.edu/mview/eval/
38. http://vision.in.tum.de/data/datasets/3dreconstruction
39. Liu Y, Cao X, Dai Q, Xu W. Continuous depth estimation for multi-view stereo. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2009
40. Tanskanen P, Kolev K, Meier L, Camposeco F, Saurer O, Pollefeys M. Live metric 3D reconstruction on mobile phones. Proc. Int. Conf. Comput. Vis. 2013
41. Ummenhofer B, Brox T. Point-based 3D reconstruction of thin objects. Proc. Int. Conf. Comput. Vis. 2013

42. Jenseny R, Dahly A, Vogiatzis G, Tolax E, Aanæs H. Large scale multi-view stereopsis evaluation. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2014

43. Kolev K, Tanskanen P, Speciale P, Pollefeys M. Turning mobile phones into 3D scanners. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2014

44. Delaunoy A, Pollefeys M. Photometric bundle adjustment for dense multi-view 3D modeling. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2014

45. Pradeep V, Rhemann C, Izadi S, Zach C, Bleyer M, Bathiche S. MonoFusion: real-time 3D reconstruction of small scenes with a single web camera. IEEE Int. Symp. Mixed Augmented Real. 2013:83–88.

46. Micušík B, Košecká J. Multi-view superpixel stereo in urban environments. Int. J. Comput. Vis. 2010; 89(1):106–119.

47. Xu C, Zhang D, Zhang Z, Feng Z. BgCut: automatic ship detection from UAV images. Sci. World J. 2014

48. Rothganger F, Lazebnik S, Schmid C, Ponce J. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. Int. J. Comput. Vis. 2006; 66(3):231–259.

Highlights

- A generalized reprojection error is proposed to fuse stereo and silhouette cues.
- The method composes of convex optimization and mesh-based surface refinement.
- Insensitive to initialization and scalable for high-resolution reconstruction
- Good performance for data with smooth texture and sparsely sampled viewpoints

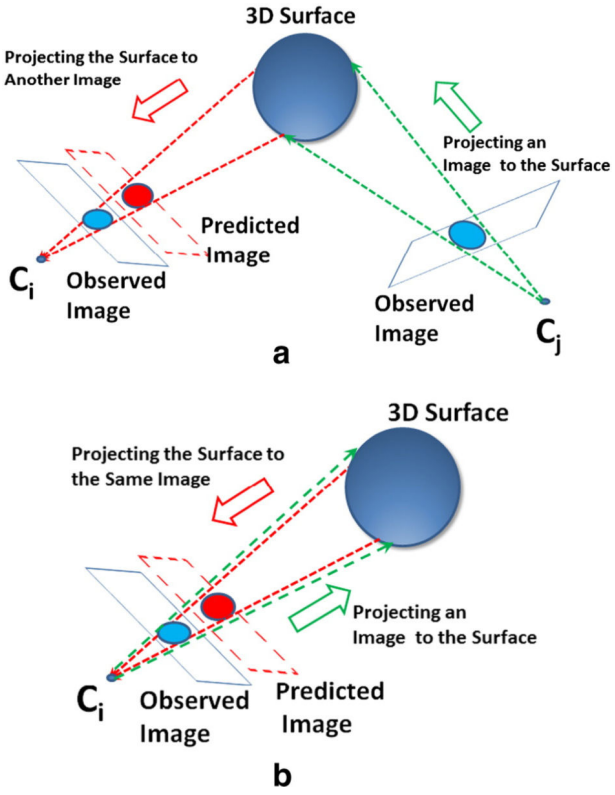


Fig. 1.

Definition of *generalized reprojection error*. (a) Reprojection error defined using different viewpoints (*stereo reprojection error*). (b) Reprojection error defined using a single viewpoint. (*silhouette reprojection error*).

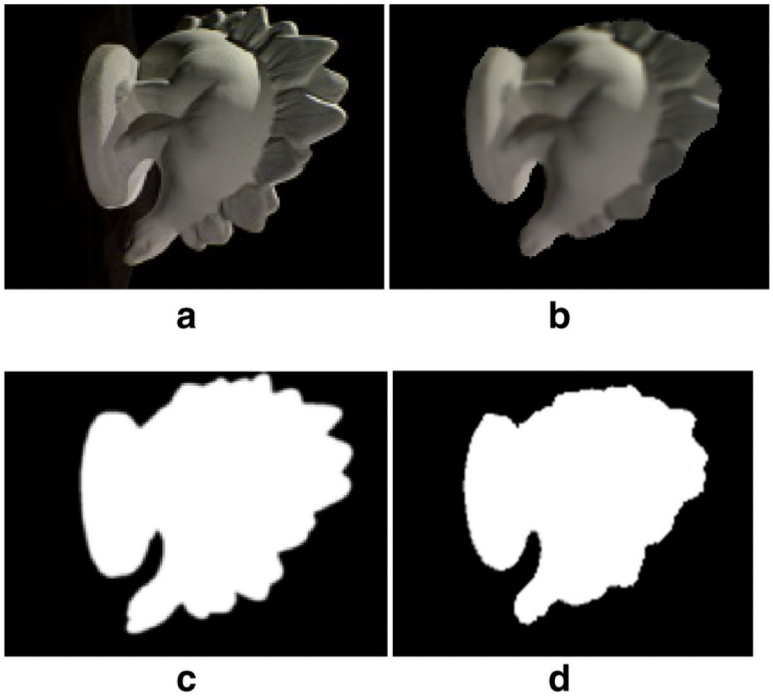


Fig. 2.

Illustration of *silhouette reprojection error*. (a) Observed image. (b) Predicted image generated by projecting (a) to an over-smoothed surface. (c) Silhouette image of (a) via image segmentation. (d) Binarized image of (b). The main difference between (a) and (b) can be observed on the boundary regions of 3D surface projection. The comparison between (c) and (d) shows a significant inconsistency of (a) and (b).

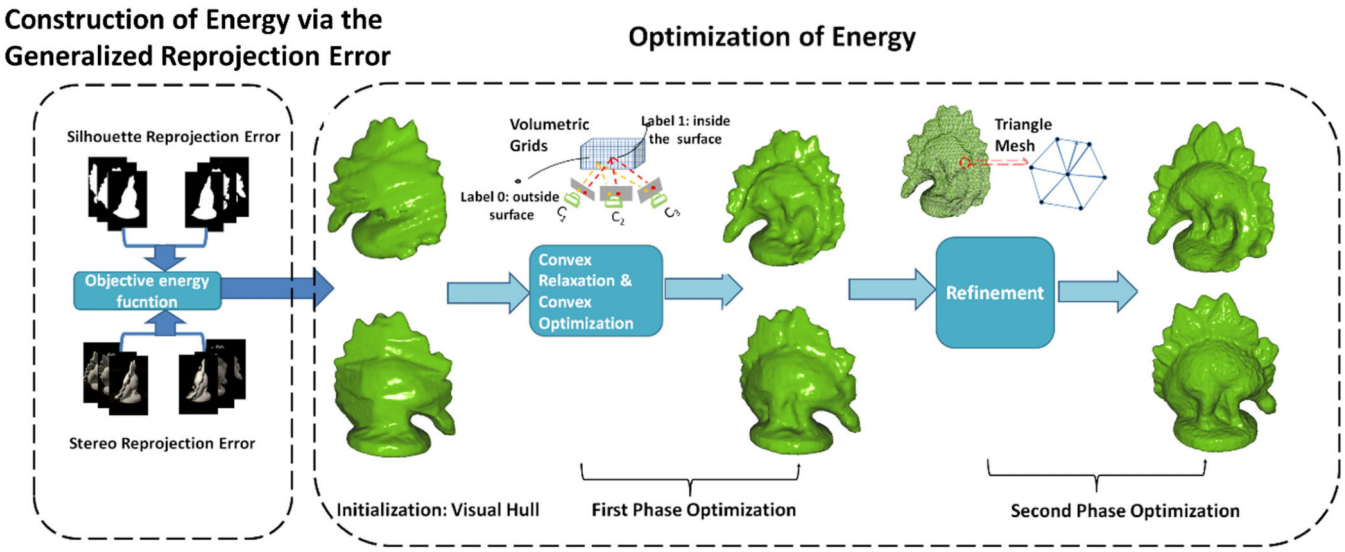


Fig. 3.
Overview of TwGREM.

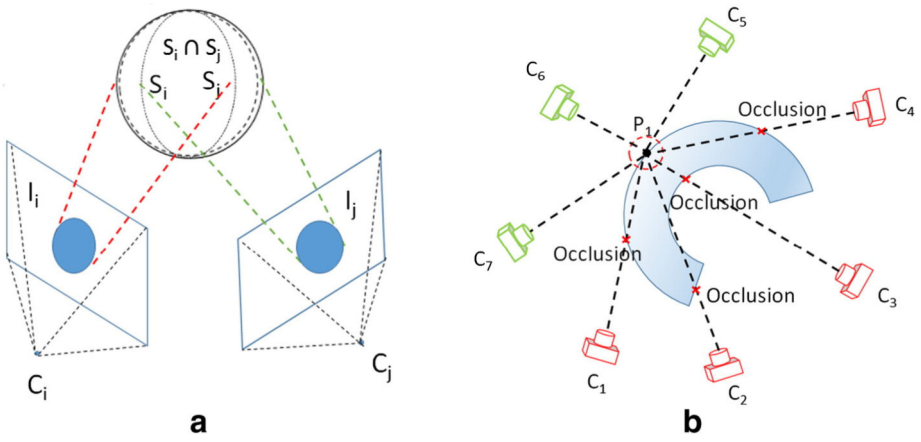


Fig. 4.

Illustration of surface visibility. (a) Visible parts of surface for each image: S_i for image I_i and S_j for image I_j. $S_i \cap S_j$ is the shared visible part for image I_i and I_j. (b) One particular part of the object (e.g., P_1) can only be seen by some of the cameras (e.g., C_5, C_6, C_7) due to occlusion.

Fig. 5.
Middlebury benchmark: dinosaur and temple.

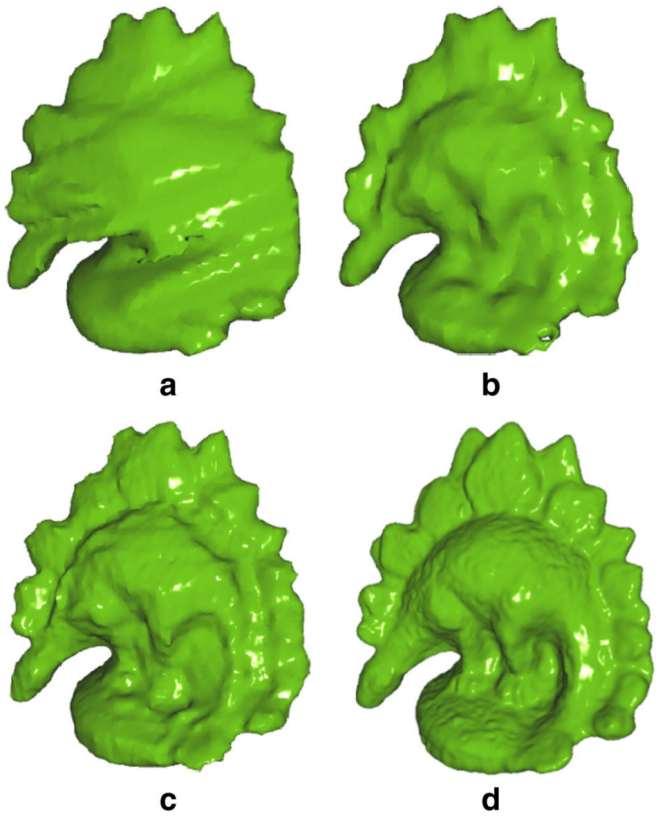


Fig. 6.

Experimental results on the *dino sparse ring*. (a) The visual hull. (b) First phase optimization on a 64 × 64 × 64 volumetric grids. (c) First phase optimization on a 128 × 128 × 128 volumetric grids. (d) Second phase optimization on triangle mesh.

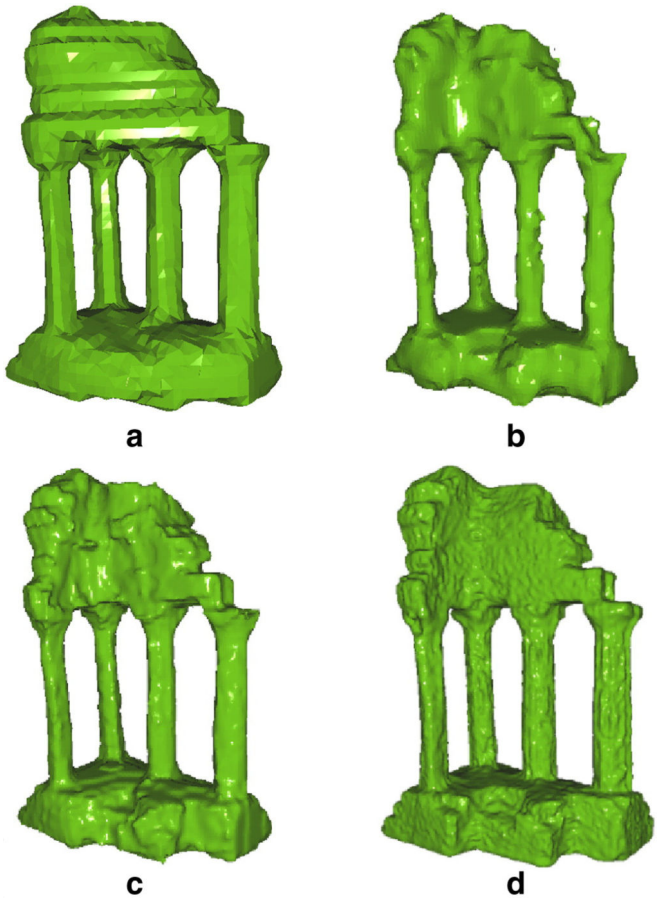
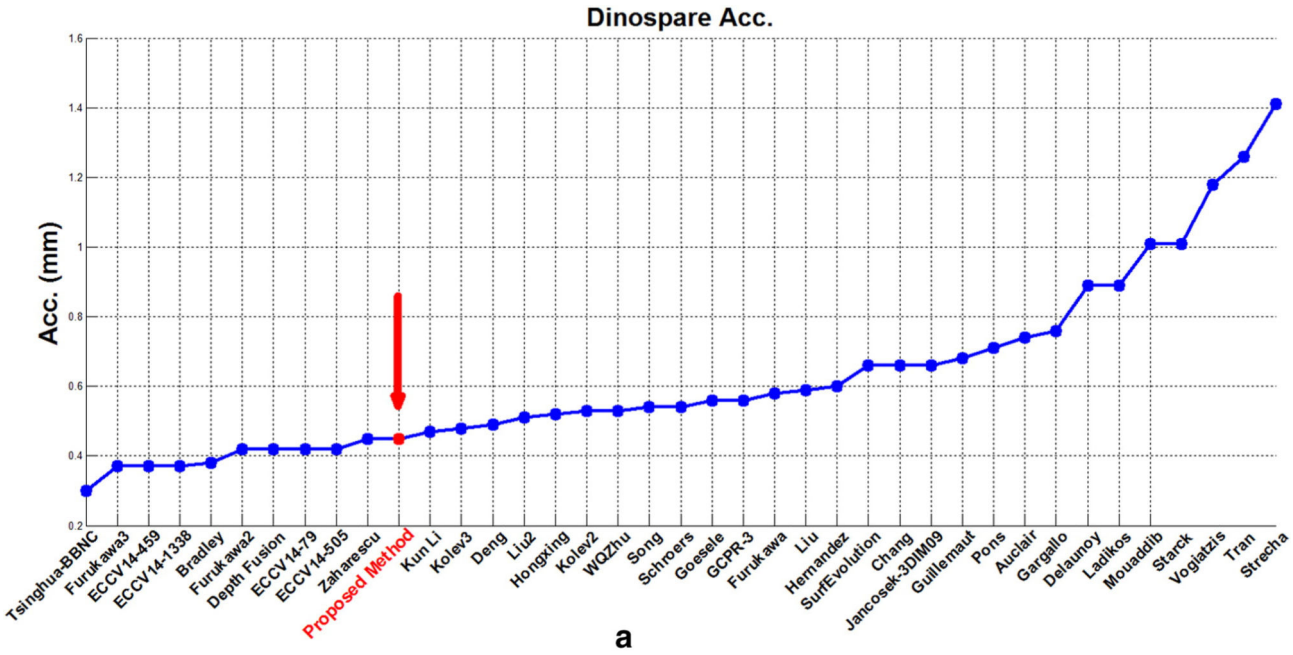


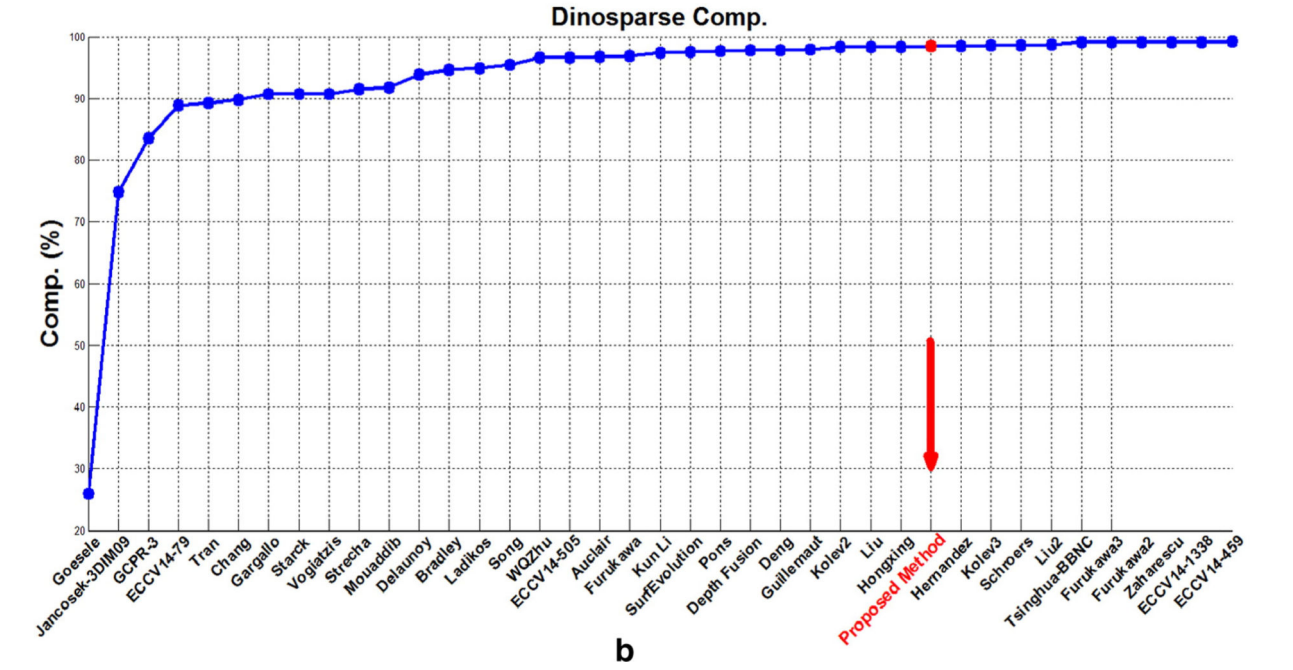
Fig. 7.

Experimental results on the *temple sparse ring*. (a) The visual hull. (b) First phase optimization on a 64 × 64 × 64 volumetric grids. (c) First phase optimization on a 128 × 128 × 128 volumetric grids. (d) Second phase optimization on triangle mesh.

Fig. 8.
Reconstruction results on *dino ring* and *temple ring*.



a



b

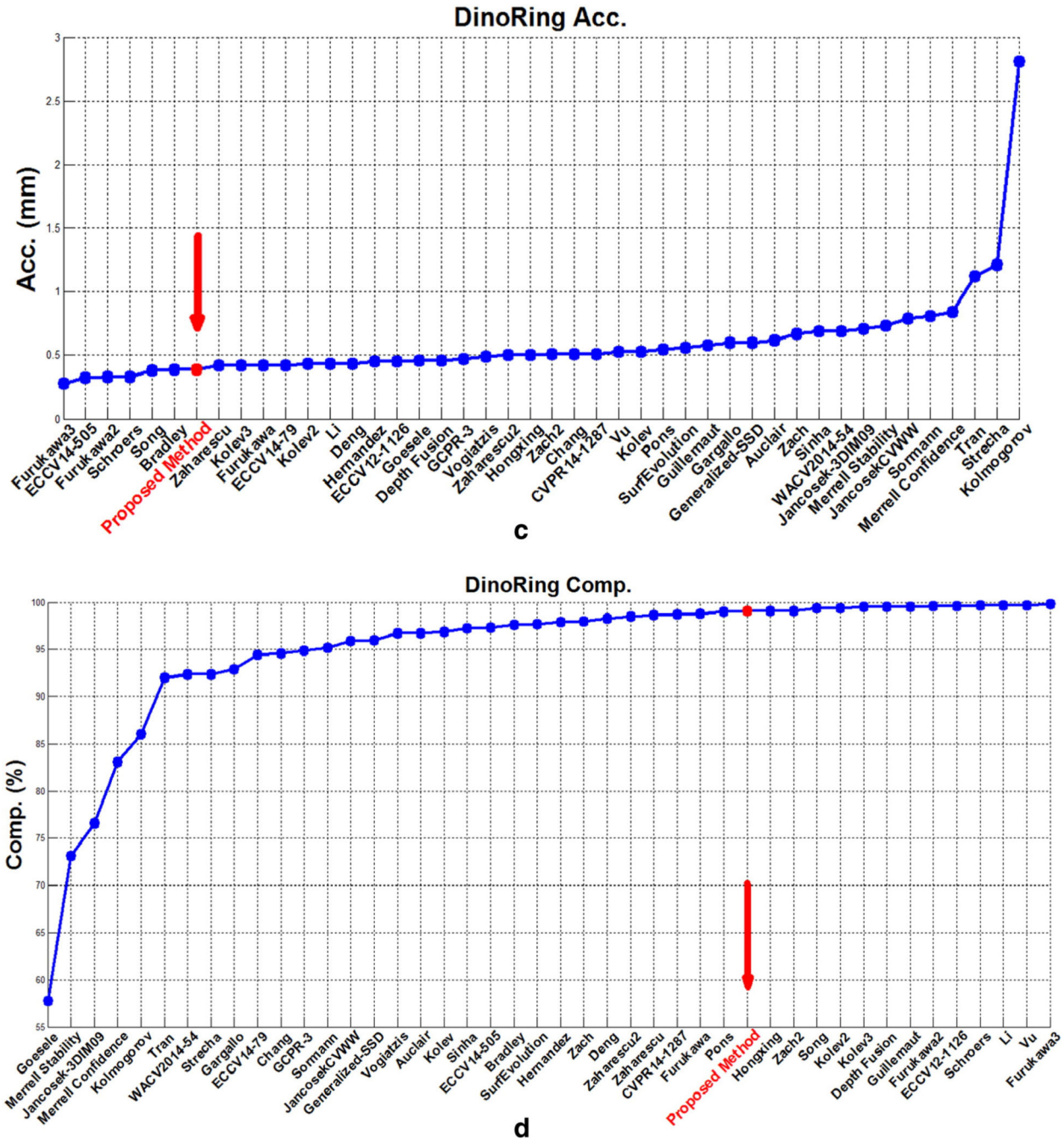


Fig. 9.

Comparison of accuracy (acc.) and completeness (comp.) with methods listed on the Middlebury evaluation page (until April, 2014) for *dino sparse ring* (a-b), *dino ring* (c-d), respectively.

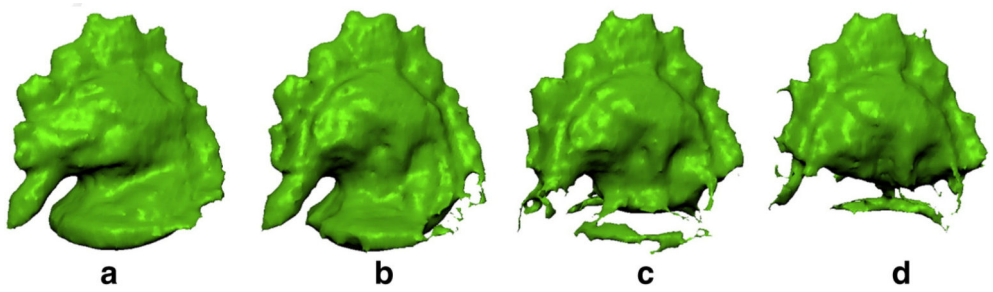


Fig. 10.

Influence of *silhouette reprojection error* in the first phase optimization. The energy term of *silhouette reprojection error* was removed by setting λ to 0. With the visual hull as an initial surface, the surface was over-smoothed while the number of iteration *m* increases. (a-d) are the reconstructed surface of *dino sparse ring* when *m* = 500, 1000, 2000, 4000, respectively.

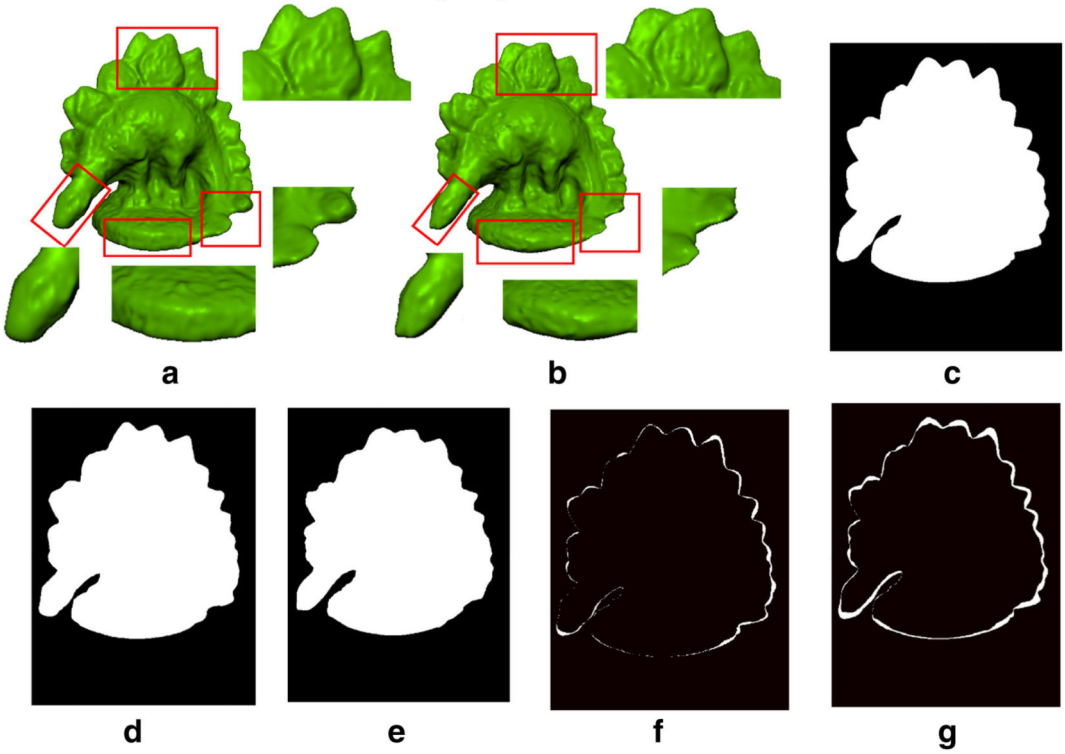


Fig. 11.

Influence of *silhouette reprojection error* in the second phase optimization. (a) is the reconstruction result of *dino sparse ring* by including the *silhouette reprojection error* in the energy function. (b) is the reconstruction result of *dino sparse ring* without using the term of *silhouette reprojection error* by setting λ to 0. (c) One of silhouette images in 16 viewpoints. (d) Predicted silhouette image generated via surface (a). (e) Predicted silhouette image generated via the surface (b). (f) The inconsistent regions of (d) and (c). (g) The inconsistent regions of (e) and (c).

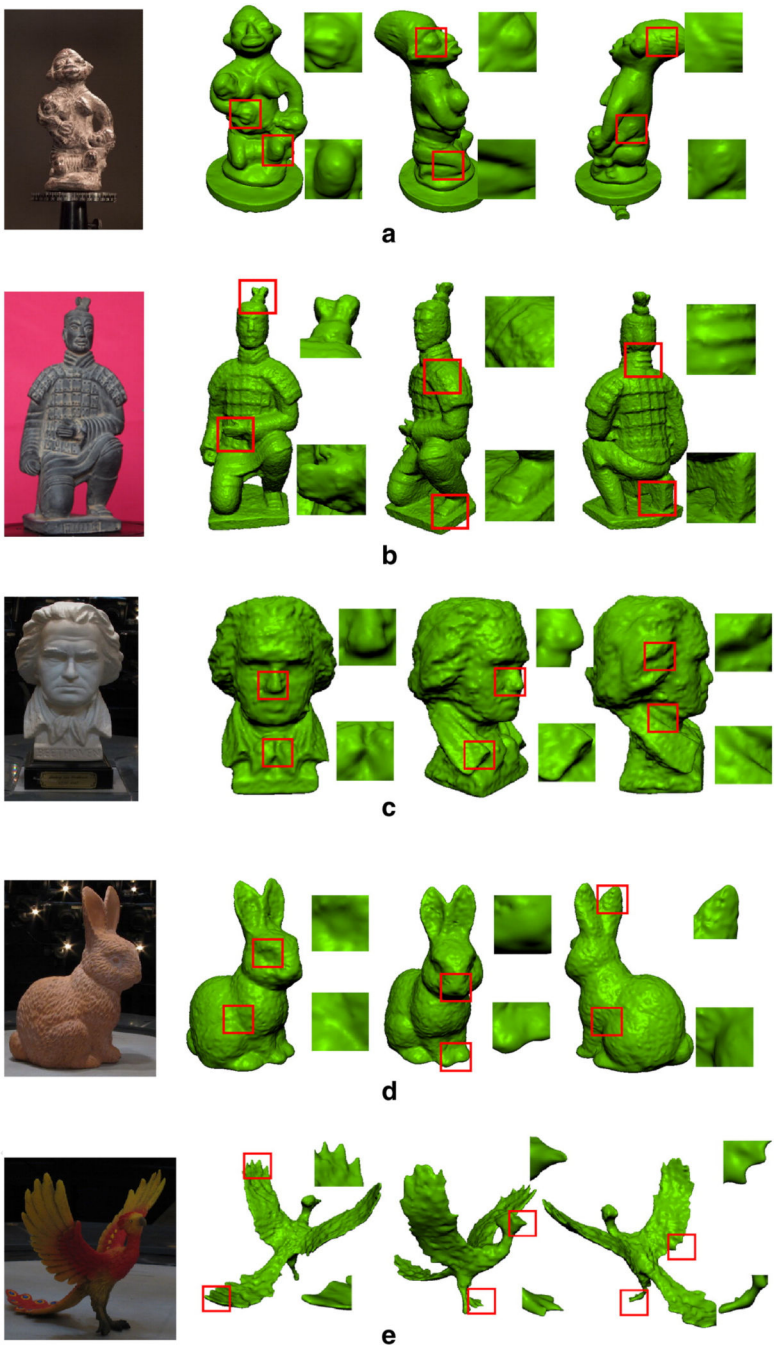


Fig. 12.

Images and 3D reconstruction results of several public datasets: (a) *twins*, (b) *captain*, (c) *Beethoven*, (d) *bunny*, and (e) *bird*.

Algorithm 1

First phase optimization of TwGREM.

Input: a set of observed images and silhouette images.

Output: the reconstructed surface.

Initialization: $u \leftarrow \phi$, $\mathbf{n}$, g and f are calculated according to the visual hull.

Do

1. Minimize Eq. (11), according to Eqs. (14)-(16)

2. Extract a triangle mesh to represent reconstructed surface using GPU-based marching cubes algorithm.

3. Generate predicted images according to reconstructed surface, and update g and f.

While $|E(S^{m+1}) - E(S^m)| < \varepsilon$, where ε is a small positive threshold.

Algorithm 2

Second phase optimization of TwGREM.

Input: a set ofobserved images and silhouette images, an initial surface from estimate of first phase optimization.

Output: the reconstructed surface.

Initialization: $\mathbf{n}_{v_i}$, $\mu_{v_i}^{\min}$ and β_{v_i} are calculated according to initial surface.

Do

1. Evolve the triangle mesh with a small time step t according to Eqs. (17)-(19).

2. Generate predicted images according to reconstructed surface, and update $\mathbf{n}_{v_i}$, $\mu_{v_i}^{\min}$ and β_{v_i} for each vertex of triangle mesh.

While $|E(S^{m+1}) - E(S^m)| < \varepsilon$, where ε is a small positive threshold.

Table 1

Datasets used in experiment.

Dataset	Number of images	Resolution	Time (minute)
Dino sparse ring	16	640 × 480	41
Temple sparse ring	16	640 × 480	49
Dino ring	48	640 × 480	90
Temple ring	47	640 × 480	113
Twins	36	2008 × 3040	140
Beethoven	33	1024 × 768	75
Bunny	36	1024 × 768	75
Bird	21	1024 × 768	63
Captain	36	3088 × 2056	150

Table 2

Quantitative comparison with the related methods on Middlebury datasets.

Methods	Datasets	Dino sparse ring		Dino ring		Temple sparse ring		Temple ring	
		Acc.(mm)	Comp. (%)	Acc.(mm)	Comp. (%)	Acc.(mm)	Comp. (%)	Acc.(mm)	Comp. (%)
TwGREM		***0.45* (*1st*)**	98.5 (2nd)	0.39 (2nd)	99.1 (4th)	***0.68* (*1st*)**	94.7 (3rd)	0.66 (3rd)	98.0 (4th)
Hernandez [25]		0.6	98.5	0.45	97.9	0.75	95.3	***0.52***	***99.5***
Song [20]		0.54	95.5	***0.38***	99.4	×	×	0.61	98.3
Gargallo [28]		0.76	90.7	0.6	92.9	1.05	81.9	0.88	84.3
Delaunoy [29]		0.89	93.9	×	×	0.73	***95.9***	×	×
Vogiatzis [18]		1.18	90.8	0.49	96.7	2.77	79.4	0.76	96.2
Kolev2 [22]		0.53	98.3	0.43	99.4	1.04	91.8	0.72	97.8
Kolev3 [27]		0.48	***98.6***	0.42	***99.5***	0.97	92.7	0.7	98.3
Sinha [21]		×	×	0.79	94.9	×	×	0.69	97.2