



Robust multi-view stereo synthesized by various parameters model ^{☆,☆☆}



Yongming Nie, Tao Yue^{*}, Hao Zhu, Sidan Du^{*}, Xun Cao

School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, PR China

ARTICLE INFO

Article history:

Received 1 September 2016

Revised 31 October 2016

Accepted 11 November 2016

Available online 11 November 2016

Keywords:

Multi-view stereo

Variational method

Matrix splitting

Low-rank matrix

Accelerated proximal gradient

ABSTRACT

In this paper, we have developed a novel and robust framework of combining a matrix splitting with multi-view stereo reconstructions to separate reconstruction inaccuracies from a various parameters model for high-accuracy multi-view stereo reconstruction. Instead of performing the fixed parameters reconstruction procedure, we apply the variational based 3D reconstruction algorithm multi-times with various parameters to derive a set of hypothetic 3D models, and then synthesized the final result by formulating the problem as a low-rank matrix splitting problem. Benefited from the matrix splitting formulation, the outliers and bad matches, which are treated as the noise in the synthesized model, are effectively removed and thus lead to a 3D reconstruction with higher accuracy than the existing fixed parameters reconstructions. Constrained convex optimization is introduced for matrix splitting with an accelerated proximal gradient (APG) algorithm integrated for fast convergence. Both the experiments on the Middlebury and real-world data sets have demonstrated the effectiveness of the proposed method.

© 2016 Published by Elsevier Inc.

1. Introduction

Multi-View Stereo (MVS) reconstruction algorithm [1–11], is used to construct 3D model from multiple calibrated images of a realistic object. It has drawn intensive interests in practical applications, such as industrial design, characters modeling for films, electronic games, cultural relics preservation, and commercial advertisement.

Generally, 3D object model reconstruction can be divided into two steps. First of all, all point clouds are computed from the calibrated image pairs of adjacent viewpoints. Many techniques have been proposed to generate the point clouds from the image pairs. Campbell et al. [12] proposed to compute the point clouds using discrete Markov Random Field (MRF). Gargallo and Sturm [13] developed a Bayesian model for the point clouds generation. Liu et al. [14] applied the variational approach to estimate continuous depth map, which is verified to be more accurate than other solutions [15–18]. Secondly, point clouds are merged together properly to produce the 3D model. Therefore, there are many works about point cloud fusion strategies concentrated on how to reduce the redundancy and improve the accuracy of the fused point cloud.

Merrel et al. proposed a fast algorithm for depth map fusion [19]. They considered selecting the most representative depth information by projecting it to the original calibrated images with the least distance. Then, the redundancy among the rough point cloud is reduced by clustering. Liu et al. [14] and Bradley et al. [10] took a similar techniques to reduce the redundancy point cloud using normalized cross correlation (NCC).

However, both point clouds estimation strategy and merging step have two prominent drawbacks:

1. The point clouds estimation is usually nonadaptive, in other words, the quality of the estimated point clouds highly depend on the parameter tuning. Analytical 3D object models generally include a data and smooth term when we apply the variational method to derive the point cloud, and the reconstructed 3D object would vary in details (i.e., texture, edges, etc.) when adapting different value of the terms' weights. Typically, reconstructed objects will have smoother surface if we use larger weighting smooth term; oppositely, it will have richer texture details if we use smaller weighting smooth term.
2. The 3D locations of a single scene point computed from different image pairs are often misaligned due to the point clouds corruption, resulting in severe accuracy damage.

To address these issues, a various parameters models synthesized 3D reconstruction framework is proposed in this paper. To preserve both smooth surface and texture details, we have experimented various parameter sets to construct multiple point clouds

[☆] This paper has been recommended for acceptance by M.T. Sun.

^{☆☆} This paper is a collaborative effort.

^{*} Corresponding authors.

E-mail addresses: nju323nym@gmail.com (Y. Nie), yuetao@nju.edu.cn (T. Yue), zhuhao@gmail.com (H. Zhu), coff128@nju.edu.cn (S. Du), caoxun@nju.edu.cn (X. Cao).

(for the same object). Then we stack all these point clouds to derive the low-rank matrix for further splitting so as to product the optimized model for more accurate 3D object reconstruction. As shown in Fig. 1, the initial point clouds are computed by the variational depth map estimation method with various parameters of a large range. The synthesized models of the same object (i.e., the initial point clouds) with different details and corruption errors are formulated into a series of column vectors, and then a matrix splitting model is employed to extract the final 3D model from the error corrupted hypotheses matrix. The *Contributions* of our work can be summarized as follows:

1. The idea of combining a matrix splitting (robust PCA) with multi-view stereo reconstructions to separate reconstruction inaccuracies from a various parameters model for high-accuracy multi-view stereo reconstruction is proposed firstly.
2. For extracting the final model from the hypotheses, a low-rank matrix splitting problem, which is more suitable for the sparse error distribution is applied.
3. By using a convex optimization of the nuclear and l_1 -norm combined objective, the proposed algorithm can be computed efficiently and accurately.

Benefiting from the above three main contributions, the proposed algorithm improves the robustness and the quality of results greatly than the existing framework straightly combining the variational based points clouds estimation and naive clouds merging. It is worth noting that the initial point clouds estimation in the proposed method can be easily replaced by any other binocular stereo algorithms, so that our method is of much flexibility with the development in the 3D reconstruction research realm.

The reminder of this paper is organized as follows: the related work is given in Section 2. The various parameters 3D reconstruction framework is illustrated in Section 3 and the implementation details are presented in Section 4. Experiments is performed to demonstrate the effectiveness of our scheme in Section 5 using the Middlebury and real-world data sets. We conclude this paper in Section 6.

2. Related work

In this section, we review the previous works about multi-view stereo reconstruction, especially on depth maps fusion techniques.

Kolmogorov and Zabih [3] computed a set of depth maps using multi-baseline stereo with graph cuts, then merge the point clouds into a voxel volume by computing the intersections of occluded volumes from each view-point. Vogiatzis [4] proposed a correlation cost volume in the neighborhood of the visual hull. A minimum cost surface is then computed using volumetric min-cut. Furukawa and Ponce [20] used wide-baseline stereo matching to recover the 3D coordinates of salient feature points, then shrink a visual hull model so that the recovered points lie on its surface, and refine the result using energy minimization. Goesele et al. [21] computed a depth map from each viewpoint and merged the results using volumetric integration.

Noise reduction (i.e., outlier and bad matches removal) is an important research topic for in 3D object model reconstruction. In the literature, a large number of 3D point cloud fusion approaches has been proposed which can be roughly grouped into two main classes: filtering and optimization methods. The former using the physical properties (the distance of each point) of point clouds to reduce the noise, while the latter builds an appearance model of the object and formulates the fusion problem as a mathematical optimization problem. Liu [14] and Bradley [10] took advantages of the NCC metric to refine the depth maps. The method reviewed above can be categorized as filtering techniques since they take advantages of the physical properties of point clouds to remove the noise in the generated depth maps. But, the 3D models of [10,14] have shown filtering only is often not effective. This is because, as detailed in [14], the NCC metric is limited to remove those isolated patches from the surface and refined point cloud still include a large quantity of errors. Campbell et al. [12] enhanced the quality of depth maps by extracting multiple depth candidates for each pixel, and then imposed global optimization algorithm to remove outliers and achieve point cloud smoothness simultaneously. Zach et al. [22] used total variation with l_1 norm regularizer to compute the point clouds, making sure that outliers and errors could be reduced. Deng et al. [23] have further introduced noisy depth maps fusion for multi-view stereo via matrix completion. The aforementioned methods [12,22,23] can be categorized as optimization since they using mathematical formulate to optimize the object. They could further reduce the redundancy and improve the accuracy of the estimated point clouds, but the models' complexity makes the optimization procedure difficult to find the optimal solutions(hard to overcome the

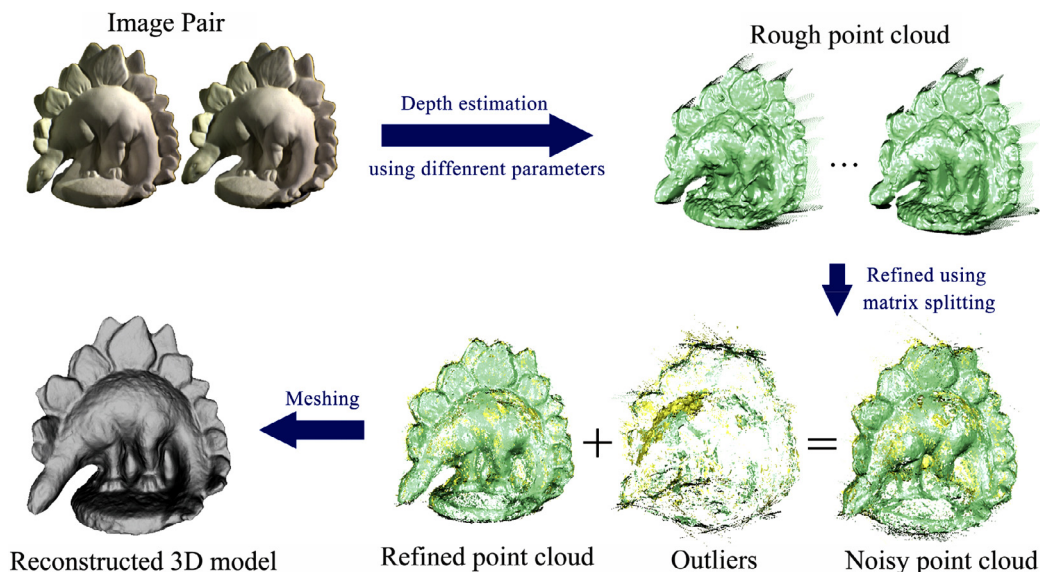


Fig. 1. Reconstruction pipeline of our proposed method. Firstly, rough point cloud were estimated using variational method with different parameters. Then, refined point cloud were computed by matrix splitting method. Finally, accurate 3D model was meshed using refined point cloud.

local minimum). So, these optimization methods not enough robust for multi-view stereo reconstruction problem.

3. Synthesized 3D reconstruction

In this section, we will propose our robust multi-view stereo method by using the various parameters models synthesized 3D reconstruction framework. As is shown in Fig. 4, the initial rough point clouds are computed from the input image pairs with different parameters. Depending on the parameter settings, a series of rough point clouds with different levels of outlier errors and smoothness are derived. By arranging these initial hypotheses into a sequence of column vectors, a matrix splitting model is used in this paper to handle the outlier errors extraction. In the following we will give the details of the proposed framework.

3.1. Various parameters point clouds estimation

Among all the depth map estimate algorithms, the variational-based method has demonstrated its strength by presenting high accuracy in developing continuous correspondence between motion images [12,24–26]. Thus, we use the variational based method [14] to estimate the hypothetic point clouds in this paper. Considering the self-containing and convenience to expression, we briefly introduce the method here. (We refer the readers to [14] for details.)

According to [14], the depth map $d(w)$ can be computed by minimizing the target energy function, which includes both the data terms and the smoothness terms. w denotes the shift from the pixel position on the target image to the optimal correspondence on the epipolar line in the reference image, named disparity. The projection of w on the epipolar line is inversely proportional to the depth $d(w)$. It is formulated as:

$$E(w) = E_D(w) + \alpha E_S(w) \quad (1)$$

where

$$E_D(w) = \int_{\Omega} \psi_D(|I_r(p + d(w)) - I_t(p)|^2 + \gamma(|\nabla I_r(p + d(w)) - \nabla I_t(p)|^2) dx dy$$

and

$$E_S(w) = \int_{\Omega} \psi_S(|\nabla w|^2) dx dy$$

Here, I_r and I_t denote the reference image and the target image respectively. $\psi(s^2) = \sqrt{s^2 + \epsilon^2}$ is a robust operator, where ϵ is a small positive constant. ψ can be applied separately to each term. $\nabla I(x, y)$ represents the gradient of the $I(x, y)$ and $\nabla = (\partial_x, \partial_y)^T$, where x and y represents image spatial direction. α is a positive

weighting parameter that trades off the data terms versus the smoothness terms.

For simplicity, we use the following abbreviations for derivatives and differences: $I_x = \partial_x I_r(p + d(w))$; $I_{xy} = \partial_{xy} I_r(p + d(w))$; $I_y = \partial_y I_r(p + d(w))$; $I_{yy} = \partial_{yy} I_r(p + d(w))$; $I_z = I_r(p + d(w)) - I_t(p)$; $I_{xz} = \partial_x I_r(p + d(w)) - \partial_x I_t(p)$; $I_{xx} = \partial_{xx} I_r(p + d(w))$; $I_{yz} = \partial_y I_r(p + d(w)) - \partial_y I_t(p)$. The problem of finding w that minimize the target variational energy function can be translated to the minimization of Euler-Lagrange equation [24]. Using the aforementioned abbreviations, Euler-Lagrange equation is shown as:

$$\begin{aligned} \psi'_D(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)) \times (I_z I_x i + \gamma I_{xz} I_{xx} i + \gamma I_{yz} I_{xy} i) \\ - \alpha \text{div}(\psi'_S(|\nabla w|^2) \nabla w) = 0 \\ \psi'_D(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)) \times (I_z I_y j + \gamma I_{xz} I_{xy} j + \gamma I_{yz} I_{yy} j) \\ - \alpha \text{div}(\psi'_S(|\nabla w|^2) \nabla w) = 0 \end{aligned} \quad (2)$$

Here, $(i, j)^T$ is the unit normal vector of direction of epipolar line corresponding to pixel p .

As seen, both data and smoothness terms in Eq. (2) are non-linear functions. In order to remove the non-linearity in Eq. (2), two fixed point iterations are applied. In the first step, Eq. (2) is converted to a linear system by a multi-resolution strategy with two nested fixed point iterations. In order to implement a multi-scale approach, a coarse-to-fine strategy with down-sampling factor η which can be more close to the global optimum of the energy function is used. It is proposed here to use an arbitrary $\eta \in (0, 1)$. We assume that w_0 is the initialization of w for each pixel in the target image, and w_0 was pre-computed from the visual hull (file generated by the EPVH tool [27]). Finally, the linear system of equations about w is solved using multi-grid method [28]. In contrast to other numerical methods, multi-grid algorithm is able to damp both high frequency and low frequency errors effectively.

As soon as a fixed point in w^k is reached, we can calculate the point clouds at each pixel according to the known camera parameters. With the help of epipolar geometry, optical flow map of the adjacent views can be derived from the function above. Then the partial point clouds can be computed from optical-flow maps and calibration parameters. Finally, 3D model point clouds are generated by combining the partial point clouds of each view point.

Fig. 2 illustrates the point clouds of both Middlebury Data sets *Dino* and *Temple*, from which we could see outliers (from the corrupted estimation in optical flow) and bad matches (from conflicting estimation of the points captured by different cameras). As a result, the key problem is how to remove these noise for point clouds refinement so as to form an accurate and complete 3D object mode reconstruction. We formulate the problem as an optimization based low-rank matrix splitting shown in the following sections.

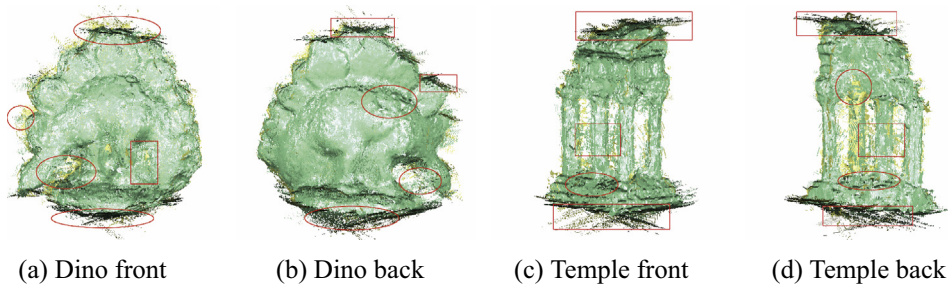


Fig. 2. Rough point clouds of DinoSparseRing data sets and TempleSparseRing data sets, the red rectangles and circles mark the outliers and the bad matched respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Outlier extraction using matrix splitting

A hypothetic point clouds D can be modeled as the combination of the ground truth data A and the outliers E . It is formulated as:

$$D = A + E \quad (3)$$

where D is the matrix representation of the rough point clouds which are estimated using variational method, A is the matrix denoted accurate point clouds and E is the matrix representation of the noise data. We demonstrate the key module of removing outliers and bad matches and how we cast the problem of 3D model reconstruction as a problem of splitting a low-rank matrix.

The matrix $D \in R^{m \times n}$ is constructed by stacking all the vectorized rough point clouds $vec(d_i)$ as:

$$D = [vec(d_1), \dots, vec(d_i)]$$

where $vec(d_i)$ ($i = 1, \dots, n$) is vectorized rough point as illustrated in Fig. 4. d_i represents different point clouds generated using different parameters listed in Table 1 variational method of the same object. Especially, n indicates the number of generated rough point clouds using variational method. We set $n = 16$ in our simulations. In other words, ten similar 3D models that contains a few outliers are generated by variational method. To clearly readers how to determine which points from different views should be put in the same column. We denote the estimated 3-D point corresponding to pixel $I_x^{(k)}$ by d_x^k . Projecting d_x^k to another image $I^{(l)}$, its 2-D correspondence in that image can be derived. Considering depth estimation error and calibration inaccuracy, this corresponding point may deviate from the real one. Generally, this corresponding point in the neighboring region of $I_p^{(l)}$. If the distance between d_x^k and $d_x^{k,l}$ is less than a specific threshold (determined by experiments), these two points are regarded as describing the same surface point and will be placed in the same column of the reconstructed matrix. In practice, the nearest point is stacked in the reconstructed matrix column if there are multiple points.

D is stacked using the same model point clouds which are generated by different parameters in variational method. The data matrix D obeys the low-rank model and the rank of D is one. In practice, there are only few points in each row were corrupted and majority of the points are correct, hence most entries in E will be zero, i.e., E is a sparse matrix. Candès et al. [29] was introduced

to recover a low rank matrix with sparse noise on the correct points. Thus, the problem can be expressed as a constrained optimization problem:

$$\begin{aligned} \min_{A,E} \quad & rank(A) + \lambda \|E\|_0 \\ \text{s.t.} \quad & D = A + E \end{aligned} \quad (4)$$

where $\|\cdot\|_0$ denotes the ℓ_0 -norm (number of non-zero entries in the matrix) and λ is a positive weighting parameter that trades off the rank of the A versus the sparsity of the error E .

While Eq. (4) follows from our formulation, it is not tractable since both $rank(\cdot)$ and ℓ_0 -norm are non-convex and discontinuous functions. Solving this optimization problem efficiently will be the topic of discussion in Section 4.

4. Implementation details

As discussed above, the outlier extraction is formulated as a matrix splitting optimization to remove outliers and bad matches, as shown in Fig. 3. As seen, the optimization problem given in Eq. (4) is NP-hard [11]. In this section, we propose to solve it efficiently based on convex relaxation of matrix rank minimization [29–32].

4.1. Convex relaxation

In the paper [33], Wright et al. have proposed that the problem in Eq. (4) can be solved by replacing the cost function with its convex relaxation, provided that the rank of the matrix A is not too high and the number of non-zero entries in the matrix E is sparse. Candès et al. [29] have proposed that the cost function was replaced by $\min_{A,E} \|A\|_* + \lambda \|E\|_1$. This convex relaxation, named Robust Principal Component Analysis (RPCA) in [29], replaces $rank(\cdot)$ with the nuclear norm (sum of the singular values of the matrix) and the ℓ_0 -norm with the ℓ_1 -norm (sum of the absolute values of all entries of the matrix). Therefore, the optimization problem given in Eq. (4) can be transformed as:

$$\begin{aligned} \min_{A,E} \quad & \|A\|_* + \lambda \|E\|_1 \\ \text{s.t.} \quad & D = A + E \end{aligned} \quad (5)$$

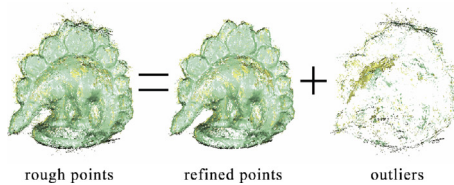
where $\|\cdot\|_*$ ($\|A\|_* = \sum_{i=1}^r \sigma_i(A)$) represents the nuclear norm and $\|\cdot\|_1$ is ℓ_1 -norm, λ is a trade-off parameter, and $\sigma_i(X)$ is the i -th singular value of the matrix X . $\|\cdot\|_1$ is defined as $\|E\|_1 = \sum_{i,j=1}^n |E_{ij}|$, which is the sum of the absolute value of matrix entries. It has been proved in [29]. Eq. (4) has the same optimal solution as Eq. (5) and $\lambda = C/\sqrt{\max(m,n)}$, where C is a constant.

4.2. Fast algorithm using accelerated proximal gradient

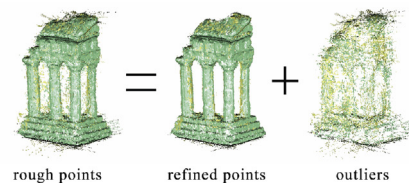
There are many works on developing for nuclear-norm and ℓ_1 norm minimization [32,33]. In the next, We show how the Accelerated Proximal Gradient (APG) can be adapted to efficiently solve

Table 1
Parameters used in the experiments.

Parameter	Value
Smooth parameter α	2:2:8
The weight factor γ	85:5:100
Multi-gride iteration	5
Outer fixed point iteration	5
Inner fixed point iteration	10
Reduction factor η	0.7



(a) Point cloud data splitting of Dino



(b) Point cloud data splitting of Temple

Fig. 3. Rough point clouds of DinoSparseRing datasets and TempleSparseRing datasets are split into two parts, refined points and outliers respectively.

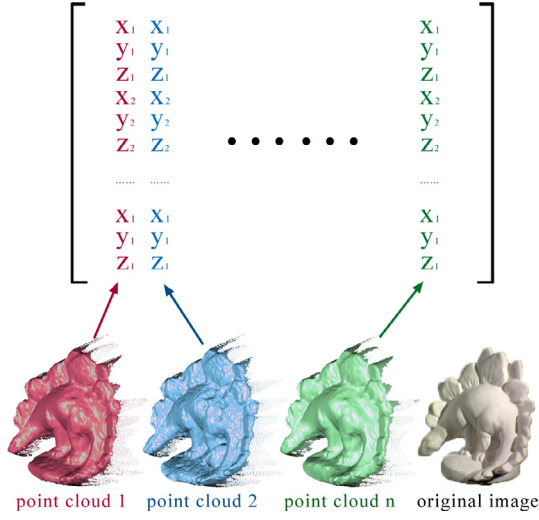


Fig. 4. The formation of low-rank matrix D is stacked using the same model point cloud which is generated by different parameters in variational method.

Eq. (5). Relaxation of the optimization problem in Eq. (5) can be formulated as

$$\min_{A,E} F(A,E) = \mu \|A\|_* + \mu \lambda \|E\|_1 + \frac{1}{2} \|D - A - E\|_F^2 \quad (6)$$

Table 2

Results for Middlebury datasets.

Methods	Dino Sparse Ring			Temple Sparse Ring		
	A. [mm]	C. [%]	T. [min]	A. [mm]	C. [%]	T.[min]
Deng [23]	0.49	97.8	13	0.74	96.1	12
Liu2 [14]	0.51	98.7	20	0.65	96.9	16
Bradley [10]	0.38	94.7	7	0.48	93.7	4
Furukawa 3 [7]	0.37	99.2	152	0.63	99.3	129
Ours	0.37	99.2	21	0.62	95.2	18

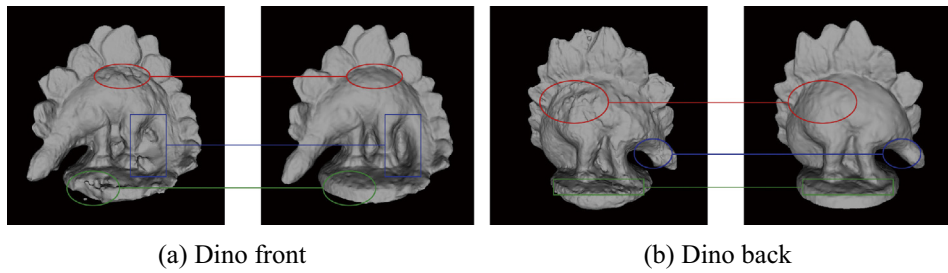


Fig. 5. Comparison results of DinoSparseRing data sets between the coarse models using NCC method and the refined models using our method, which demonstrate the strength of our refine model.

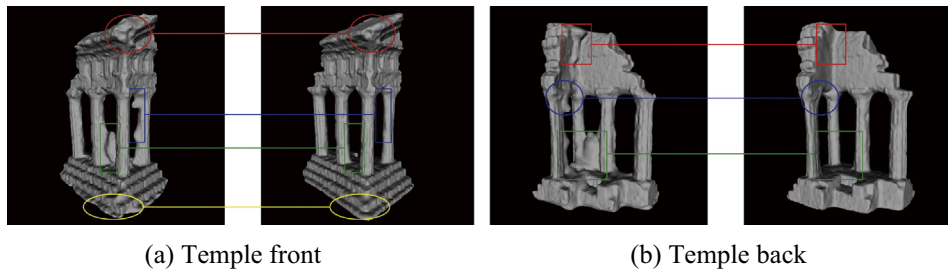


Fig. 6. Comparison results of TempleSparseRing data sets between the coarse models using NCC method and the refined models using our method, which demonstrate the strength of our refine model.

where μ is a small positive scalar and $\|\cdot\|_F$ is the Frobenius norm. Accelerated proximal gradient (APG) algorithm [34] is utilized to reach the solution of A in an iterative fashion, where the basic APG iteration is given by

$$\begin{aligned} A_{k+1} &= S_{\mu/L_f} \left(Y_k^A - \frac{1}{L_f} (Y_k^A + Y_k^E - D) \right) \\ E_{k+1} &= S_{\mu/L_f} \left(Y_k^E - \frac{1}{L_f} (Y_k^E + Y_k^A - D) \right) \end{aligned} \quad (7)$$

where Y_k is updated using the following rule for fast convergence:

$$\begin{aligned} Y_k^A &= A_k - \frac{t_{k-1} - 1}{t_k} (A_k - A_{k-1}) \\ Y_k^E &= E_k - \frac{t_{k-1} - 1}{t_k} (E_k - E_{k-1}) \\ t_{k+1}^2 - t_{k+1} &\leq t_k^2 \end{aligned} \quad (8)$$

Meanwhile, A_k and E_k are updated following

$$\begin{aligned} A_{k+1} &= US_{\frac{\mu_k}{2}}[S]V^T \\ E_{k+1} &= S_{\frac{i\mu_k}{2}}[G_k^E] \end{aligned} \quad (9)$$

where USV^T represents the singular decomposition of G_k^E , as defined in Algorithm 1. $S_\epsilon[x]$ is a shrinkage operator shown below:

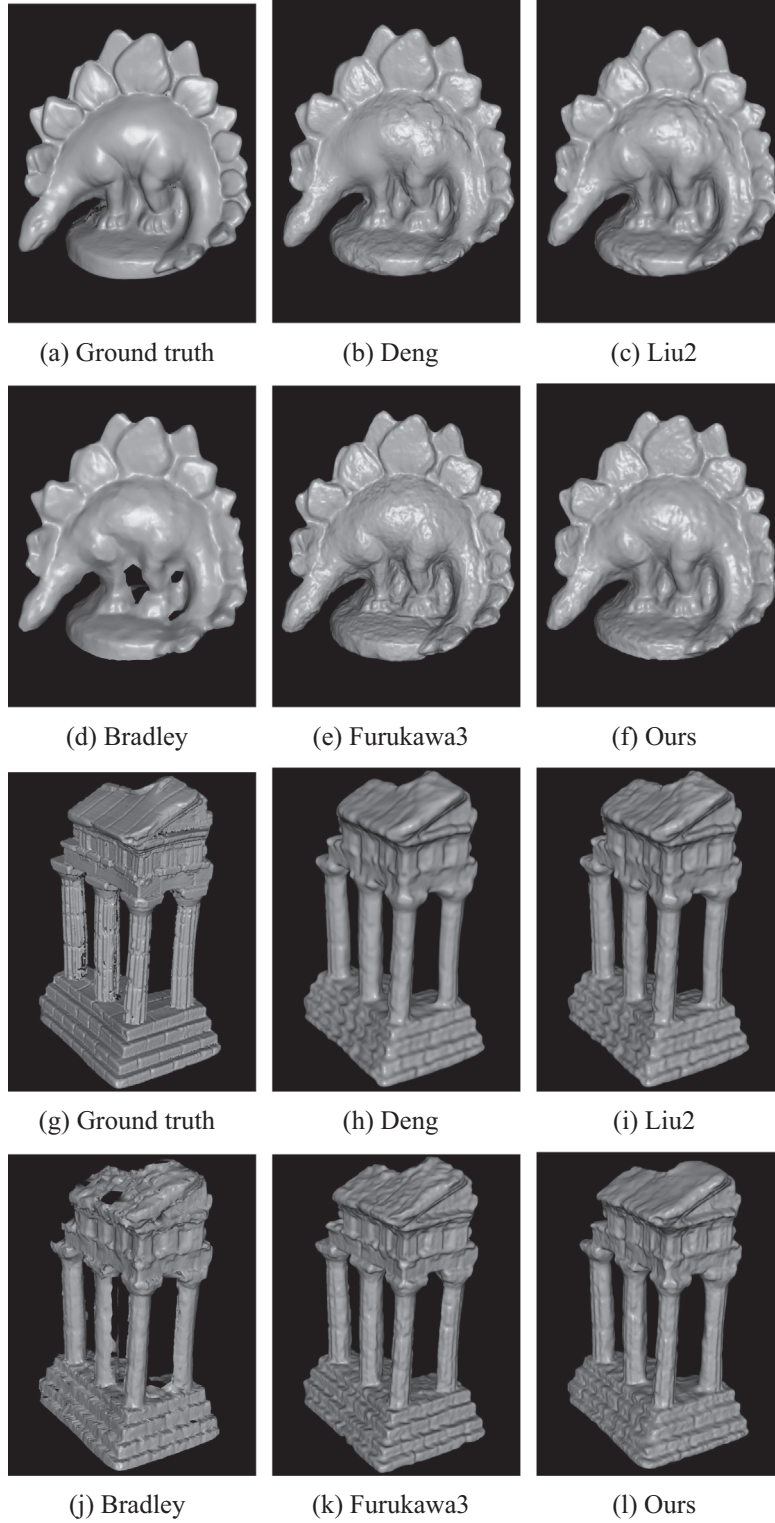


Fig. 7. The comparison of reconstruction results on both DinoSparseRing and TempleSparseRing data sets.

$$\mathcal{S}_\epsilon[x] =: \begin{cases} x - \epsilon & \text{if } x > \epsilon \\ x + \epsilon & \text{if } x < -\epsilon \\ 0 & \text{otherwise} \end{cases}$$

where $x \in R$ and $\epsilon > 0$.

μ is a small positive scalar. A classical technique [30], which varies μ , starting from a large initial value μ_0 and decreasing it geometrically for each iteration until it reaches the floor μ , can speed up the convergence dramatically. μ_k is a monotonically increasing positive sequence. The entire algorithm to solve Eq. (6) has been summarized as Algorithm 1.

Algorithm 1. Data matrix splitting via APG

Input : Data matrix $D \in R^{m \times n}$.
Initialization: $A_0, A_{-1} \leftarrow 0$; $E_0, E_{-1} \leftarrow 0$; $t_0, t_{-1} \leftarrow 1$; $h\mu \leftarrow \delta\mu_0$.
repeat
2 $Y_k^A \leftarrow A_k + \frac{t_{k-1}-1}{t_k}(A_k - A_{k-1}), Y_k^E \leftarrow E_k + \frac{t_{k-1}-1}{t_k}(E_k - E_{k-1});$
3 $G_k^A \leftarrow Y_k^A - \frac{1}{2}(Y_k^A - Y_k^E - D);$
4 $(U, S, V) \leftarrow \text{svd}(G_k^A), A_{k+1} = US_{\frac{\mu_k}{2}}[S]V^T;$
5 $G_k^E \leftarrow Y_k^E - \frac{1}{2}(Y_k^A + Y_k^E - D);$
6 $E_{k+1} = S_{\frac{\mu_k}{2}}[G_k^E];$
7 $t_{k+1} \leftarrow \frac{1+\sqrt{4t_k^2+1}}{2};$
8 $\mu_{k+1} \leftarrow \max(\eta\mu_k, h\mu);$
9 $k \leftarrow k + 1.$
until convergence;
Output : $(A, E) \leftarrow (A_k, E_k).$

5. Experiments

To evaluate the performance of the matrix splitting algorithm, we tested this method using data sets from the Middlebury Vision Evaluation Website [35] and Real-World data sets [36]. We implemented our algorithm using C++ with OpenCV library. Firstly the rough point clouds are estimated using variational method, then the matrix splitting is applied to refine the point clouds. 3D models are generated from the rough point clouds and the refined point clouds using Poisson surface reconstruction [37] separately. We compare the two 3D models to demonstrate that our method removes the noise and errors effectively.

In these experiments, we verify the effectiveness of our method quantitatively. We compare our results with other state-of-the-art

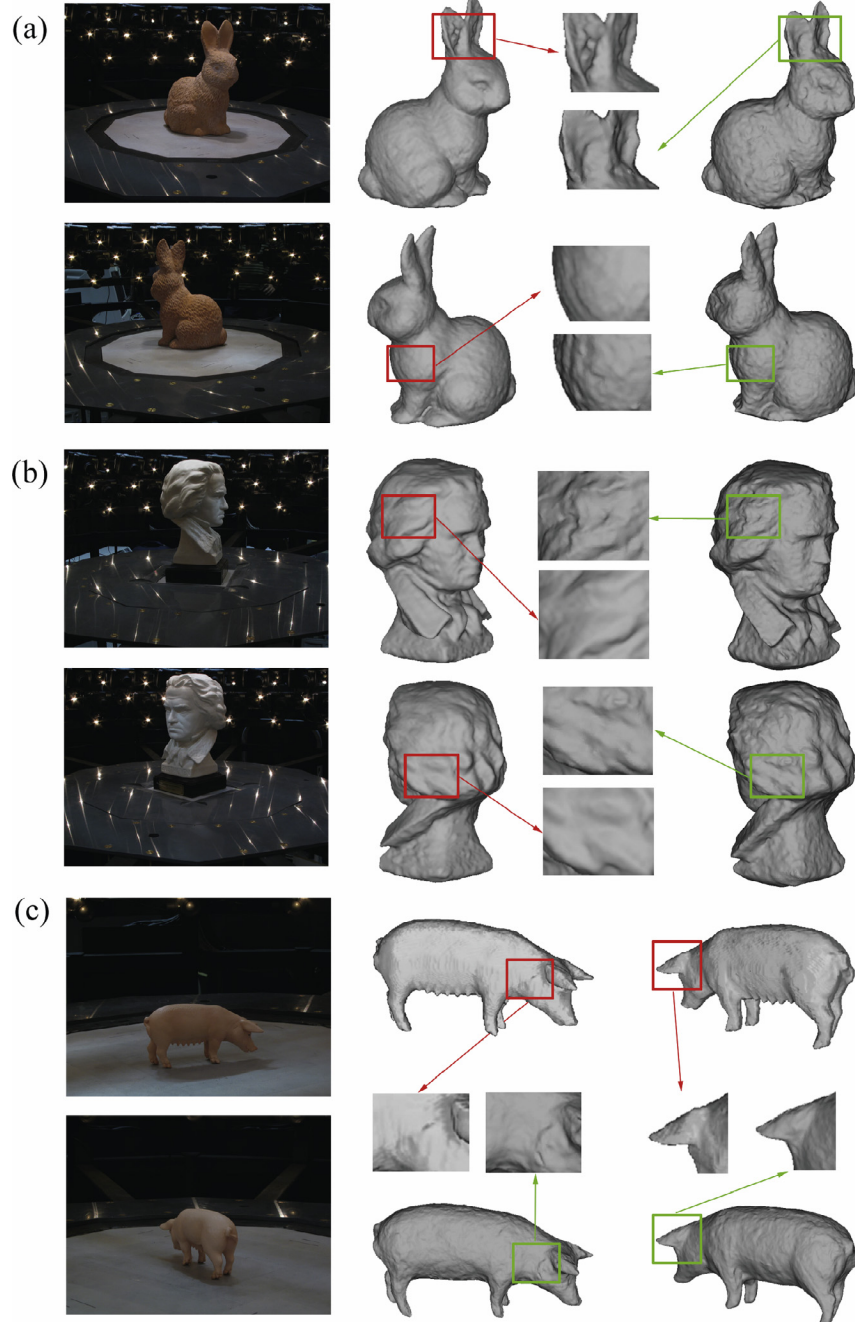


Fig. 8. Images and 3D reconstruction results of several real datasets: (a) bunny (b) beethoven (c) pig. The red box represent of the results of [24], green box represent of our results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reconstruction algorithms. The comparison is reported in Table 2, where we use A to represent the reconstruction accuracy, C to indicate the completeness and T is the processing time at minute scale.

5.1. Hypothetic point clouds estimation using variational method

Firstly, *DinoSparseRing* and *TempleSparseRing* datasets are used to generate the rough point clouds using variational method. The data sets can be obtained from the website [35], which contain 16 images and a calibration parameters file. Left parts of Fig. 5 shows the rough 3D model of *Dino* using filter in PMVS [7]. In Figs. 5 and 6, the colored rectangles and circles mark the regions that are corrupted severely. The rough point clouds estimation module is implemented based on Brox's optical flow method [24].

5.2. Outlier extraction using matrix splitting

It can be seen from Fig. 6 that the reconstruction accuracy of variational method vibrates over different parts of the model. By contrast, a matrix splitting method as Algorithm 1 can refine the point clouds robustly in this experiments. In the above experiments, λ is a important weighting parameter. We have fixed the value of the parameter $\lambda = 1/\sqrt{n}$, as suggested by [29]. n is the number of all rough points, i.e., dimensions of the vector $vec(d_i)$ ($i = 1, \dots, n$). From Fig. 6 we can see that the proposed refined method could dramatically remove outliers and bad matches.

The quantitative results of our framework are shown in Table 2. At the moment when the results were submitted to the Middlebury Vision Evaluation Website, the accuracy and completeness measurements of the results about two data sets both rank top, especially, the 3D model of *Dino* was in top three. Fig. 7 show refined 3D models, which are more accurate and complete than existing state-of-art methods [7,10,14,23].

5.3. Middlebury datasets

According to the results, we can conclude that the algorithm achieves the best accuracy among all the reconstruction algorithm applied in *DinoSparseRing* data sets. In the condition of the same accuracy and completeness, Furukawa's [7] reconstruction algorithm spent 10 more times of computational cost than our method. Bradley's method [10] achieves higher accuracy about *TempleSparse* data sets, but results of them have low scores in completeness since they can't generate closed models. Overall, our method is among the best reconstruction accuracy on both evaluation data sets, as well as a attractive balance on both reconstruction completeness and computational efficiency.

5.4. Real-world data sets

To further evaluate the proposed method, we applied our algorithm to several real-world data sets: *beethoven* datasets [36], *bunny* datasets [36], *pig* datasets [36]. Ground truth for these data sets were not available, and thus qualitative evaluation of appearance of the reconstruction results was adopted in the experiments.

The *beethoven*, *bunny* and *pig* data sets include 33, 36, 26 images respectively. They were captured by a set of synchronized multiple cameras. The *beethoven* data sets present textureless surface, and the *pig* data sets present smooth surface. Our experiment results of these data sets are illustrated in Fig. 8. The proposed method is able to handle noise and outliers effectively, even for the textureless situation. The *bunny* data sets include only 36 images of highly textured images. Unlikely, some details of wrings of the bunny were not kept. The reason may be that these small details be affect

by the sooth data factor when rough points evaluation using variational method.

6. Conclusions

In this paper, we propose a novel technique for accurate reconstruction of 3D geometrical shape from multiple calibrated images of the same object. The rough point clouds are generated by integrating the silhouette information and epipolar constraint into the variational method. To remove outliers and bad matches effectively, We first introduce a low-rank matrix splitting on 3D model reconstruction which can be solved efficiently using accelerated proximal gradient algorithm. Compared with the existing state-of-art methods, we have shown the top performance in both accuracy and completeness measurements using Middlebury and real-world data sets.

Exploring the advantage of the low-rank minimization, we consider that the proposed technique can be extended to handle similar problems in other low-level vision tasks in the future.

Acknowledgments

This work was partially supported by Grant Nos. 61300157, 61201425, 61371166, 61422107 from National Science Foundation of China. We thank S. Seitz, B. Curless, J. Diebel, D. Scharstein and R. Szeliski for the *Temple* and *Dino* data sets and evaluations.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.jvcir.2016.11.005>.

References

- [1] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, A comparison and evaluation of multi-view stereo reconstruction algorithms, 2006 IEEE Computer Society Conference on Computer vision and pattern recognition, vol. 1, IEEE, 2006, pp. 519–528.
- [2] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *Int. J. Comput. Vision* 47 (1–3) (2002) 7–42.
- [3] V. Kolmogorov, R. Zabih, Multi-camera scene reconstruction via graph cuts, in: *Computer Vision–ECCV 2002*, Springer, 2002, pp. 82–96.
- [4] G. Vogiatzis, C. Hernández, P.H. Torr, R. Cipolla, Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (12) (2007) 2241–2246.
- [5] M. Habbecke, L. Kobbelt, A surface-growing approach to multi-view stereo reconstruction, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
- [6] M. Goesele, N. Snavely, B. Curless, H. Hoppe, Multi-view stereo for community photo collections, in: *IEEE International Conference on Computer Vision, ICCV, 2007*, pp. 1–8.
- [7] Y. Furukawa, J. Ponce, Accurate, dense, and robust multiview stereopsis, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (8) (2010) 1362–1376.
- [8] C. Hernández Esteban, F. Schmitt, Silhouette and stereo fusion for 3d object modeling, *Comput. Vision Image Understand.* 96 (3) (2004) 367–392.
- [9] S. Tran, L. Davis, 3d surface reconstruction using graph cuts with surface constraints, in: *Computer Vision–ECCV 2006*, Springer, 2006, pp. 219–231.
- [10] D. Bradley, T. Boubekeur, W. Heidrich, Accurate multi-view reconstruction using robust binocular stereo and surface meshing, in: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, 2008, pp. 1–8.
- [11] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, *Found. Comput. Math.* 9 (6) (2009) 717–772.
- [12] N.D. Campbell, G. Vogiatzis, C. Hernández, R. Cipolla, Using multiple hypotheses to improve depth-maps for multi-view stereo, in: *Computer Vision–ECCV 2008*, Springer, 2008, pp. 766–779.
- [13] P. Gargallo, P. Sturm, Bayesian 3d modeling from images using multiple depth maps, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 2, IEEE, 2005, pp. 885–891.
- [14] Y. Liu, X. Cao, Q. Dai, W. Xu, Continuous depth estimation for multi-view stereo, in: *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009*, IEEE, 2009, pp. 2121–2128.
- [15] M.R. Oswald, J. Stühmer, D. Cremers, Generalized connectivity constraints for spatio-temporal multi-view 3d reconstruction, in: *European Conference on Computer Vision*, 2014, pp. 32–46.

- [16] K. Kolev, T. Brox, D. Cremers, Fast joint estimation of silhouettes and dense 3D geometry from multiple images, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (3) (2012) 493–505.
- [17] D. Cremers, K. Kolev, Multiview stereo and silhouette consistency via convex functionals over convex domains, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (6) (2011) 1161–1174.
- [18] K. Kolev, D. Cremers, Continuous ratio optimization via convex relaxation with applications to multiview 3d reconstruction, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. CVPR 2009, 2009, pp. 1858–1864.
- [19] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.M. Frahm, R. Yang, D. Nister, M. Pollefeys, Real-time visibility-based fusion of depth maps, in: *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.
- [20] Y. Furukawa, J. Ponce, High-fidelity image-based modeling, *Urbana* 12 (2006) 1825–1860.
- [21] M. Goesele, B. Curless, S.M. Seitz, Multi-view stereo revisited, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2402–2409.
- [22] C. Zach, T. Pock, H. Bischof, A duality based approach for realtime $tv - l_1$ optical flow, in: *Pattern Recognition*, Springer, 2007, pp. 214–223.
- [23] Y. Deng, Y. Liu, Q. Dai, Z. Zhang, Y. Wang, Noisy depth maps fusion for multiview stereo via matrix completion, *IEEE J. Sel. Top. Signal Process.* 6 (5) (2012) 566–582.
- [24] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: *Computer Vision-ECCV 2004*, Springer, 2004, pp. 25–36.
- [25] K. Kolev, M. Klodt, T. Brox, D. Cremers, Continuous global optimization in multiview 3d reconstruction, *Int. J. Comput. Vision* 84 (1) (2009) 80–96.
- [26] N. Slesareva, T. Bühler, K.U. Hagenburg, J. Weickert, A. Bruhn, Z. Karni, H.-P. Seidel, Robust variational reconstruction from multiple views, in: *Image Analysis*, Springer, 2007, pp. 173–182.
- [27] J.-S. Franco, E. Boyer, et al., Exact polyhedral visual hulls, *British Machine Vision Conference (BMVC'03)*, vol. 1, 2003, pp. 329–338.
- [28] R.E. Bank, T.F. Dupont, H. Yserentant, The hierarchical basis multigrid method, *Numer. Math.* 52 (4) (1988) 427–458.
- [29] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, *J ACM (JACM)* 58 (3) (2011) 11.
- [30] E.J. Candès, Y. Plan, Matrix completion with noise, *Proc. IEEE* 98 (6) (2010) 925–936.
- [31] E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted l_1 minimization, *J. Fourier Anal. Appl.* 14 (5–6) (2008) 877–905.
- [32] Z. Lin, M. Chen, Y. Ma, The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-rank Matrices. Available from: <1009.5055>.
- [33] J. Wright, Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization, *Adv. Neural Inform. Process. Syst.* 22.
- [34] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, *SIAM J. Optimiz.* (2016) (submitted for publication).
- [35] Middlebury Vision Evaluation Website. <<http://vision.middlebury.edu/mview/data/>>.
- [36] Real World Datasets. <<http://vision.in.tum.de/data/datasets/3dreconstruction/>>.
- [37] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson surface reconstruction, in: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 2006, pp. 61–70.