SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2016, Vol. 59 092103:1–092103:11
doi: 10.1007/s11432-015-0957-4

Robust dense reconstruction by range merging based on confidence estimation

Yadang CHEN[1,3], Chuanyan HAO[2,3]*, Wen WU[3] & Enhua WU[3,4]

[1]*School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing* 210044*, China;*
[2]*School of Education Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing* 210023*, China;*
[3]*Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau* 999078*, China;*
[4]*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing* 100864*, China*

Received January 18, 2016; accepted March 7, 2016; published online August 18, 2016

Abstract Although the stereo matching problem has been extensively studied during the past decades, automatically computing a dense 3D reconstruction from several multiple views is still a difficult task owing to the problems of textureless regions, outliers, detail loss, and various other factors. In this paper, these difficult problems are handled effectively by a robust model that outputs an accurate and dense reconstruction as the final result from an input of multiple images captured by a normal camera. First, the positions of the camera and sparse 3D points are estimated by a structure-from-motion algorithm and we compute the range map with a confidence estimation for each image in our approach. Then all the range maps are integrated into a fine point cloud data set. In the final step we use a Poisson reconstruction algorithm to finish the reconstruction. The major contributions of the work lie in the following points: effective range-computation and confidence-estimation methods are proposed to handle the problems of textureless regions, outliers and detail loss. Then, the range maps are merged into the point cloud data in terms of a confidence-estimation. Finally, Poisson reconstruction algorithm completes the dense mesh. In addition, texture mapping is also implemented as a post-processing work for obtaining good visual effects. Experimental results are presented to demonstrate the effectiveness of the proposed approach.

Keywords stereo matching, 3D reconstruction, textureless regions, outliers, details loss, range map

Citation Chen Y D, Hao C Y, Wu W, et al. Robust dense reconstruction by range merging based on confidence estimation. Sci China Inf Sci, 2016, 59(9): 092103, doi: 10.1007/s11432-015-0957-4

1 Introduction

Three-dimensional reconstruction has been a hot topic for many years, and it is also widely used in a variety of fields such as virtual reality and augmented reality especially. The 3D information of a scene can be obtained by cameras with especial hardware support [1] such as infrared scanning or others such technology. With the rapid development of computer vision [2, 3], most people prefer to recover 3D

* Corresponding author (email: hcy@njupt.edu.cn)

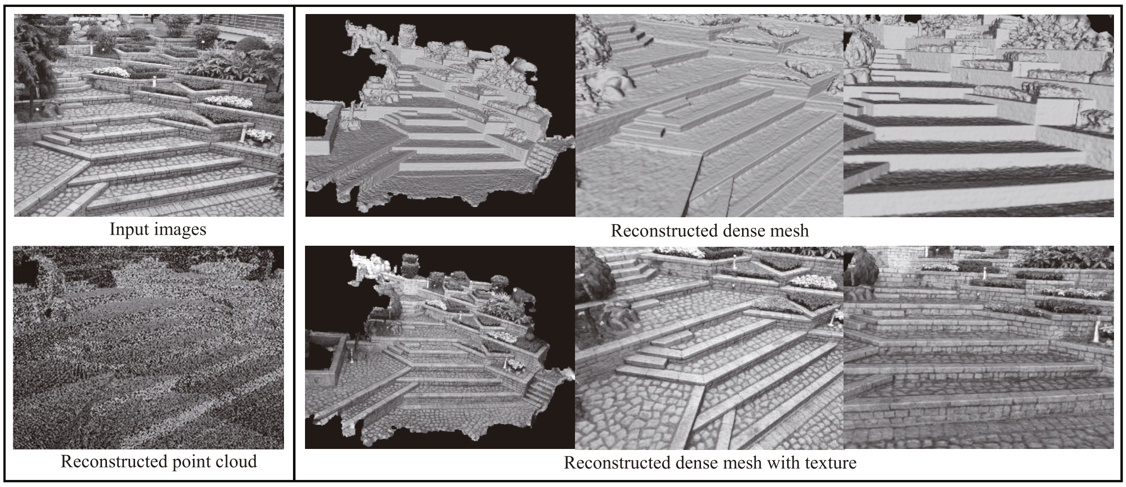
Input images
Reconstructed dense mesh
Reconstructed point cloud
Reconstructed dense mesh with texture

Figure 1 An example of our work: "Stone-step". The input of the work is a number of images from different camera views, and the output is a reconstructed dense mesh.

information readily from 2D materials alone, such as from images [4, 5] or videos [6, 7], that can be taken easily with a normal camera. Certainly, video-based reconstruction is somewhat similar to image-based reconstruction, as a video is composed exactly of a sequence of images.

In fact, recovering high dimensional data exactly from low dimensional samples exactly has been an ill-conditional problem without an exact solution so far, because there must be information missing for the reconstruction of data having higher dimensions. Nevertheless, there has been much excellent albeit struggling work on the topic in the past such as those methods proposed in [8–11]. However, most of them are not effective for dealing with difficult cases such as textureless regions, outliers, and detail loss. The specifics of these challenging points are described as follows:

(1) Textureless regions: the pixels within the region have nearly the same color information.

(2) Outliers: occlusion, object movements in the matching views, violations of constant illumination and image noise.

(3) Detail loss: object details are lost during the reconstruction.

As we know, the key point of stereo matching is to try to find the best matched points between two views. Consequently it is relatively difficult to find the ground truth matched point from a textureless region, because all the candidates have almost the same color as the matching one. On the other hand, the outliers that may be caused by different cases will violate the epipolar geometrical knowledge on account of those interferences listed above. Detail loss is another annoying problem as well. In summary, all of these are necessary issues that need to be taken into account for 3D reconstruction. The main contributions in this work include accurate and smooth depth maps that are generated under three restrictions for each reference frame. The depth maps are merged into a dense mesh by using a form of error-cloud optimization based on the confidence. Figure 1 gives an example of our work.

2 Related work

Multi-view 3D reconstruction techniques have been well studied for many years. In this section, we discuss the relevant previous work from the following aspects.

Outlier handling. Seitz et al. [5] proposed a form of the voxel coloring method to cope with large changes in visibility and the ambiguous modeling of an intrinsic scene color and texture information. Outliers are also taken full account of in their method by dividing the 3D space into the several "near-to-far" layers. However, the reconstruction needs prior knowledge of the bounding box. Outlier estimation is introduced in some methods [12–14]. These usually estimate a visibility map at first, which is commonly used to indicate whether a pixel in one image is also visible in another image. Each pixel in the map has a value of 0 or 1, indicating being occluded or not, respectively. Yet unfortunately, this preprocessing operation appears inconvenient for most of the reconstruction work. If it is assumed that the outliers occur in only a few images, then the problem can be solved to a certain extent in a simplified way [1,15], where the authors select a subset of the matching results from multiple views rather than all of them in order to eliminate the negative interference from outlier matching. This artifice indeed achieves some progress with less computational cost, but it cannot always be effective in all cases.

Textureless regions and details loss handling. Textureless regions present another challenge for multi-view stereo matching. There is a well-known reconstruction software called CVMS [8,9] proposed by Furukawa et al. In this work, the system generates a dense 3D reconstruction by expanding the initially sparse 3D feature points. Though this work can produce good results on objects with a rich texture, such as buildings and sculptures, it is not applicable to textureless regions because there are no detected feature points from which to expand in a large textureless region. A much more widely used method is to solve the textureless regions by treating multi-view stereo as a problem with the form of a Markov random field (MRF) [15–18]. As is widely known, there are two basic items that describe a MRF problem: the data item and the smooth item. For textureless regions, the data item will be confused by all candidates with the same color. Therefore, the relationship between two neighboring pixels is used for reference by combining the smooth item. Most of the methods use this idea to handle textureless regions, while ignoring the fact that more details would be smoothed by this "content-unaware" item. Even though some of them turn the content-unaware item into a color-sensitive one, it is still less than helpful to the complicated cases.

Dense reconstruction. Until now, it has been surprising to find that many dense reconstruction methods have been proposed based coincidentally on range (depth) map fusion [19–23]. Specifically, Ref. [22] is based on minimizing an energy functional consisting of a total-variation-regularization force and an L1 data-fidelity term. They present a novel and efficient numerical scheme, which combines the duality principle for the TV term with a point-wise optimization step. Stühmer et al. [20] present a novel variational approach to estimate dense depth maps from multiple images in real-time by using robust penalty factors for both the data term and regularizer. The work in [18] is based on the construction of a hierarchical signed distance field represented in an incomplete primal octree by incrementally adding triangulated depth maps.

The most important question for all the above depth-fusion-based methods is how to generate accurate depth maps, because the quality of a final dense reconstruction lies in the quality of the depth maps. Most methods get the depth maps by simple means, such as the Plane Sweep [4], resulting in rough depth maps being generated. In order to remove the noise of the reconstruction results from multiple rough depth maps, they tried to minimize a type of non-deterministic convex energy function iteratively, for instance by truncated distance fields in a voxel space: $\Omega \subseteq \mathbb{R}^3$ [19, 22]. However, the consumption of both the run-time and memory space was hardly affordable in a real-time system. Thus, the accuracy is usually obtained at the cost of low efficiency. Directed to solving this problem, we focused our efforts on achieving better depth results with higher efficiency, and also to accelerate the algorithm for depth fusion.

3 Overview

From the above discussion, we give the overview of our reconstruction pipeline in Figure 2, which can be divided into four parts for simplicity. Considering the positive points of the depth-fusion-based methods for dense reconstruction, we also construct the depth map for each input view. Here the estimation of a depth map can be seen as a bridge between the 2D image and the 3D reconstructed geometrical model. In a similar way to other reconstruction work, structure-from-motion is the first stage in the pipeline used to extract the camera poses and some sparse feature points. There has been much successful SFM work that is good enough for reconstruction such as [24–26], so it is sensible to use this as the starting point in our work. Depth computation and confidence estimation are then carried out. A form of confidence-

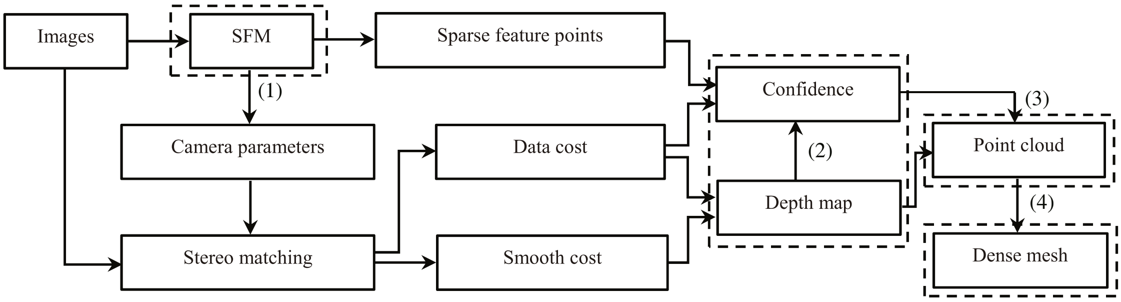

Figure 2 Overview of our reconstruction pipeline. Depth-map estimation can be seen as a bridge between the 2D image and the 3D mesh. (1) Structure from motion; (2) depth map computation (Figure 3(e)) and confidence estimation (Figure 3(c)); (3) point cloud (Figure 3(f)); (4) surface reconstruction (Figure 6).

based depth-fusion algorithm is proposed to merge all the depth maps into cloud data. Finally, a Poisson reconstruction algorithm [27] is combined to finish the reconstruction. In addition, texture mapping is also implemented as a post-processing job for a good visual effect. Of all the procedures in our work, the depth-map construction and confidence estimation are the most important, because we try to solve the problem of textureless regions, outliers and detail loss together during this step in a more mitigated way. Our purpose is to try to preserve the details in the constructed depth map with high confidence, and to mark out the outliers in the confidence map as well. In the rest of this paper, the details of the depth computation and confidence estimation are given in Subsection 4.1, and the details of the range-merging algorithm follow in Subsection 4.2. The experimental results and their comparisons can be seen in Section 5.

4 Algorithms

First, we adopt the MBA algorithm [25] interleaved with repeated bundle adjustment in an optimized parallel implementation on the basis of the frame-rate-pose estimation to achieve an accurate estimation for the positions of the scene points and poses of the camera frames. Then, in this section, we focus on describing the two parts of the work shown as (2) and (3) in Figure 2, which are also main contributions of this paper. What is more, the sampling precision of the depth computation and the confidence estimation are exactly the same as the image resolution.

4.1 Depth computation and confidence estimation

In general, depth-map computation can be equivalent to minimizing a form of energy equation that can be solved by many software packages such as total variation 1 norm optimization (TV-L1) [19,21] or the belief-propagation algorithm [16]. In this paper, we improve the equation as follows:

$$D_{I_i}(x) = \begin{cases} \arg\min_{d_x \in \boldsymbol{D}} \{E_c(d_x, I_i, \boldsymbol{N}_i) + \mu E_s(d_x, I_i)\}, & x \notin \boldsymbol{S}_f, \\ d_f(x), & x \in \boldsymbol{S}_f, \end{cases} \tag{1}$$

where $\boldsymbol{S}_f$ is the set of feature points from the MBA algorithm, and $d_f(x)$ is the disparity value of the point that can been seen as the ground truth for guiding the rest of the computation. Thus, if the SFM algorithm can produce more feature points at first, Eq. (1) will preserve more details of the object for the reconstruction. The term E_c indicates the data cost, E_s indicates the smooth cost and μ is usually set as a fixed value to control the weight between the two costs.

4.1.1 *Details of data cost and smooth cost*

More specifically here, the set of disparity candidates $\boldsymbol{D}$ of each pixel (disparity is the reciprocal of depth) is defined first by us as a discrete enumeration in the range of $[d_{\min}, d_{\max}]$ in order to reduce the computational cost. In addition, I_i denotes the ith image and $\boldsymbol{N}_i$ denotes the set of all the other projected images (not including the outside projection ones). The data cost is defined as

$$E_c(d_x, I_i, \boldsymbol{N}_i) = \sum_{x \in I_i} 1 - \mathrm{norm}(D_c(x, d_x, R_i, \boldsymbol{N}_i)), \tag{2}$$

where norm is the normalization function. The D_c function is written as

$$D_c(x, d_x, I_i, \boldsymbol{N}_i) = \sum_{f \in \boldsymbol{N}_i} L_c(x, d_x, f, I_i), \tag{3}$$

and

$$L_c(x, d_x, f, I_i) = \frac{\varepsilon_c}{\varepsilon_c + \|I_i(x) - I'_f(x', d_x)\|}. \tag{4}$$

Here, ε_c controls the shape of the differentiable robust function, and $\|I_i(x) - I'_f(x', d_x)\|$ is the color distance by L-2 norm, where $I_i(x)$ denotes the color of pixel x in image I_i and $I'_f(x', d_x)$ denotes the color of pixel x' in the matching frame f. The homography process of x' projected from x is defined as

$$x' = \boldsymbol{K}_f \boldsymbol{R}_f \boldsymbol{R}_{I_i}^{\mathrm{T}}(\boldsymbol{K}_{I_i}^{-1} x - d_x \boldsymbol{T}_{I_i}) + d_x \boldsymbol{K}_f \boldsymbol{T}_f, \tag{5}$$

where $\boldsymbol{K}$, $\boldsymbol{R}$ and $\boldsymbol{T}$ are camera intrinsic and extrinsic parameters. The terms x and x' are the homogeneous coordinates of pixel x and x' respectively. Moreover, we adopt the same optimization mechanism to decrease the negative effect from outliers as that introduced by Kang et al. [28]. We assume that if the outlier problem happens in a few images, the sum of the color difference in Eq. (3) will be disturbed by the few "bad" comparisons even for the ground truth. Therefore, the problem can be solved by selecting only the better half of the comparison results rather than all of them, as seen in the improved equation:

$$D_c(x, d, I_i, \boldsymbol{N}_i) = \sum_{f \in \frac{\boldsymbol{N}_i}{2}} L_c(x, d, f, I_i), \tag{6}$$

where $\frac{\boldsymbol{N}_i}{2}$ denotes the better half of n comparison results. The smooth cost is defined as

$$E_s(d_x, I_i) = \sum_{x \in I_i} \mathrm{norm}\left(\sum_{y \in \boldsymbol{N}(x)} \lambda(x, y)|d_x - d_y| \right), \tag{7}$$

where $\boldsymbol{N}(x)$ is the set of neighbors of pixel x in I_i, and $\lambda(x, y)$ is the "content-aware" item that is sensitive to color distance:

$$\lambda(x, y) = \frac{\varepsilon_g}{\varepsilon_g + \parallel I_i(x) - I_i(y) \parallel}, \quad y \in \boldsymbol{N}(x), \tag{8}$$

by which the weight of a smooth item will be reduced to zero by the relatively large color distance between the two pixels, where ε_g controls the shape of the differentiable robust function.

As is well known, the quality of the final reconstructed results depends on the quality of the depth maps. Compared with other methods, our method can generate a more accurate depth result, as seen in Figure 3. The details of the scene are preserved mostly according to Eq. (1), and the depth computation for the textureless regions can be guided in the appropriate way by Eq. (7). Moreover, our depth computation is also robust to outliers problem by Eq. (6), nevertheless the result cannot always be true for every case in the real world. Accordingly, we make confidence estimation for each depth computation in the final reconstruction.

4.1.2 *Confidence estimation*

A confidence estimation for each depth map is both important and necessary. In more detail, we follow certain rules as follows in order to define the confidence for each depth pixel:

(1) At first, the confidences of feature points in Eq. (1) should have the highest value because they indicate the ground truth.

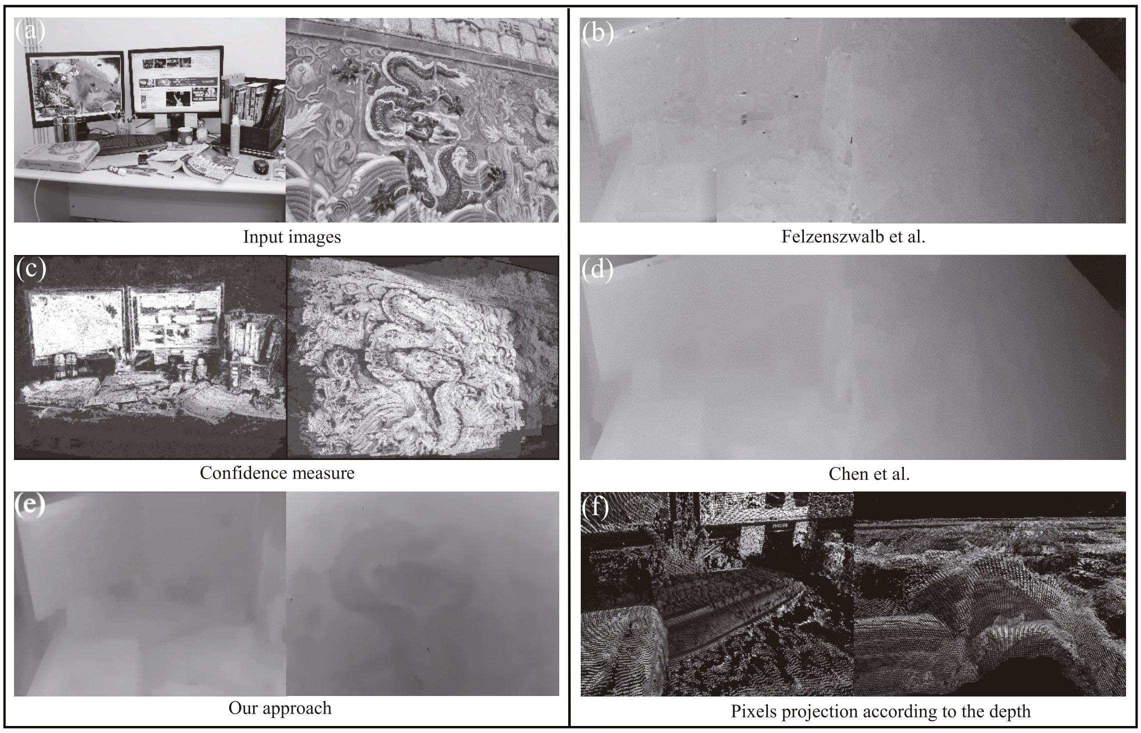

Figure 3 Depth map and confidence estimation for two groups of input. (a) The input sample. (b) depth computation by Felzenszwalb et al. [16]. (c) confidence estimation. (d) depth computation by Chen et al. [7]. (e) our approach for depth computation. (f) the pixels projection according to the depth map of (e).

(2) Then the pixels near the feature points should have relatively high confidence, whereas the pixels far from these points should have a lower confidence.

(3) The final rule is defined in relation to the possible number of the outliers during the multi-views stereo matching. More outliers should have less confidence, and vice versa.

We use the following mathematical equation to encode the above rules:

$$F_{I_i}(x)=\begin{cases}\dfrac{\max\left(\exp\left(-\frac{\|x-y(x)\|^2}{2\sigma_f^2}\right),M_f\right)}{N_i^o/N_i^p+\varepsilon_f}, & x\notin \boldsymbol{S}_f,\\ 1, & x\in \boldsymbol{S}_f,\end{cases}\tag{9}$$

where $y(x)$ is the closest feature point to x, and $\|x-y(x)\|$ is the Euler distance between $y(x)$ and x. N_i^p is the total number of projected images for image I_i, and N_i^o is the possible number of outliers. For those pixels $x\notin \boldsymbol{S}_f$, the confidence will be small for two cases:

(1) x and $y(x)$ are separated by a large distance, which is used to penalize the pixels far from the feature points.

(2) There is a large number outliers during the stereo matching, which is used to penalize the outliers. Here, M_f, σ_f and ε_f are all adjustable parameters used to map the confidence onto a reasonable arrange, which is set in an experiential way in our experiment. What is more, once the best candidate d_x is chosen as the disparity result after the depth-map computation, then N_i^o can be obtained approximately by counting the bad matches that have a large color distance computed by $\parallel I_i(x)-I'_f(x',d_x)\parallel$. We define bad matching as having a color distance of more than 20 (within the range [0, 256]) in our experiment.

The results for both the depth map and the confidence can be seen in Figure 3. In order to prove that our method can make a better depth map for each input, we compare the result of our approach with the traditional belief-propagation algorithm based on a Markov random field (MRF) [16], and also with an improved algorithm [7], which are shown as Figure 3(e), (b) and (d) respectively. Figure 3(c) gives us the confidence-estimation map for the depth map, where a white pixel means the highest confidence, and

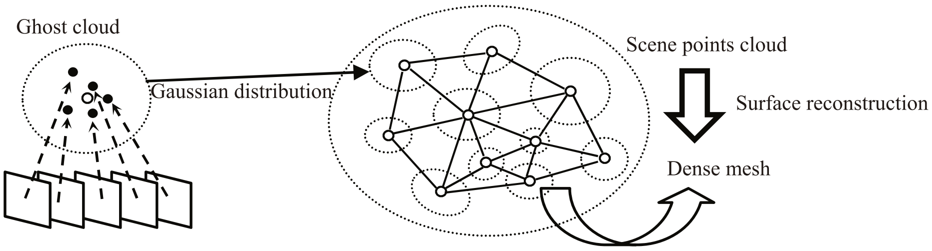

Figure 4 Depth fusion into the point cloud data.

black means the lowest. The confidence map is used for the following depth-fusion work along with the depth map, for the purpose of marking out the outliers. Figure 3(f) shows the point cloud data projected from one input image according to the depth map; the major details of the object have been kept well through our method.

4.2 Depth fusion into point cloud data

Here, we focus on how to transform the depth maps into the dense mesh. We usually consider the depth map as 2.5D rather than the pure 3D. The simplest way to finish the conversion is to project all the pixels into 3D space according to their depths, and then the dense mesh can be obtained by any surface-reconstruction algorithm based on the point cloud data. The basic idea of the depth fusion is just as discussed above, but the situation is not that simple here. As we know, the depth bundle includes multiple depth maps rather than only one, where some of the pixels on different depth maps are likely to denote the same point in 3D space. Ideally, the pixels that denote the same 3D point, even if they are on different depth maps, should be projected to the same position in 3D space, while that is impossible for reality. In this section, we introduce a depth-fusion method similar to the projection classification algorithm proposed by Chen et al. [7], by which we can find all the ghost clouds. The distribution of each ghost cloud satisfies the rule that the closer to the ground truth location, the more concentrated in the distribution of the deviated projection, and vice versa. So the expectation of the ground truth location is calculated by treating the ghost cloud as a Gaussian distribution. Finally, each ghost cloud becomes one reconstructed point in 3D space making up the final point cloud data of the scene, as shown in Figure 4. The difference between their work and ours is that we use a type of confidence-based score mechanism to record the grade for each candidate point x, as follows:

$$s_{x'_i} = F_{I_i}(x')\left(\frac{\varepsilon_{sc}}{\varepsilon_{sc} + \|c_x - c_{x'_i}\|} + \varphi_d \frac{\varepsilon_{sd}}{\varepsilon_{sd} + |u_{x\to x'_i} - u_{x'_i}|}\right), \tag{10}$$

where $\|c_x - c_{x'_i}\|$ is the color-difference L-2 norm between x and x'_i and $|u_{x\to x'_i} - u_{x'_i}|$ is the distance L-1 norm. $s_{x'_i}$ denotes the score of x'_i projected from x, where x'_i belongs to the rest of the matching images. ε_{sc} and ε_{sd} control the shape of the differentiable functions, and φ_d controls the weight. From Eq. (10), the score is re-weighted to the range $[0, 1]$ in terms of the confidence.

Once all the ghost clouds have been solved, each solution point is connected with its neighbors to form triangular faces by a surface-reconstruction algorithm. The associated vertex normal vector can be computed here as $\boldsymbol{n}_v = \triangle\boldsymbol{v}_x \times \triangle\boldsymbol{v}_y$. Texture mapping is carried out by the Gourand interpolation method as the final task for a good visual effect.

5 Experiments and comparisons

All the test input images in our experiment are resized into 2048×1730 resolution, and the run-time analysis is tested on a desktop PC equipped with a 3.4 GHz Intel 8 Core processor, and 8 GB of RAM. The actual CPU occupancy rate during the reconstruction achieves 70% at the most by 6 threads. In theory, any scene in the image sequence can be reconstructed. However, we find that the more completely

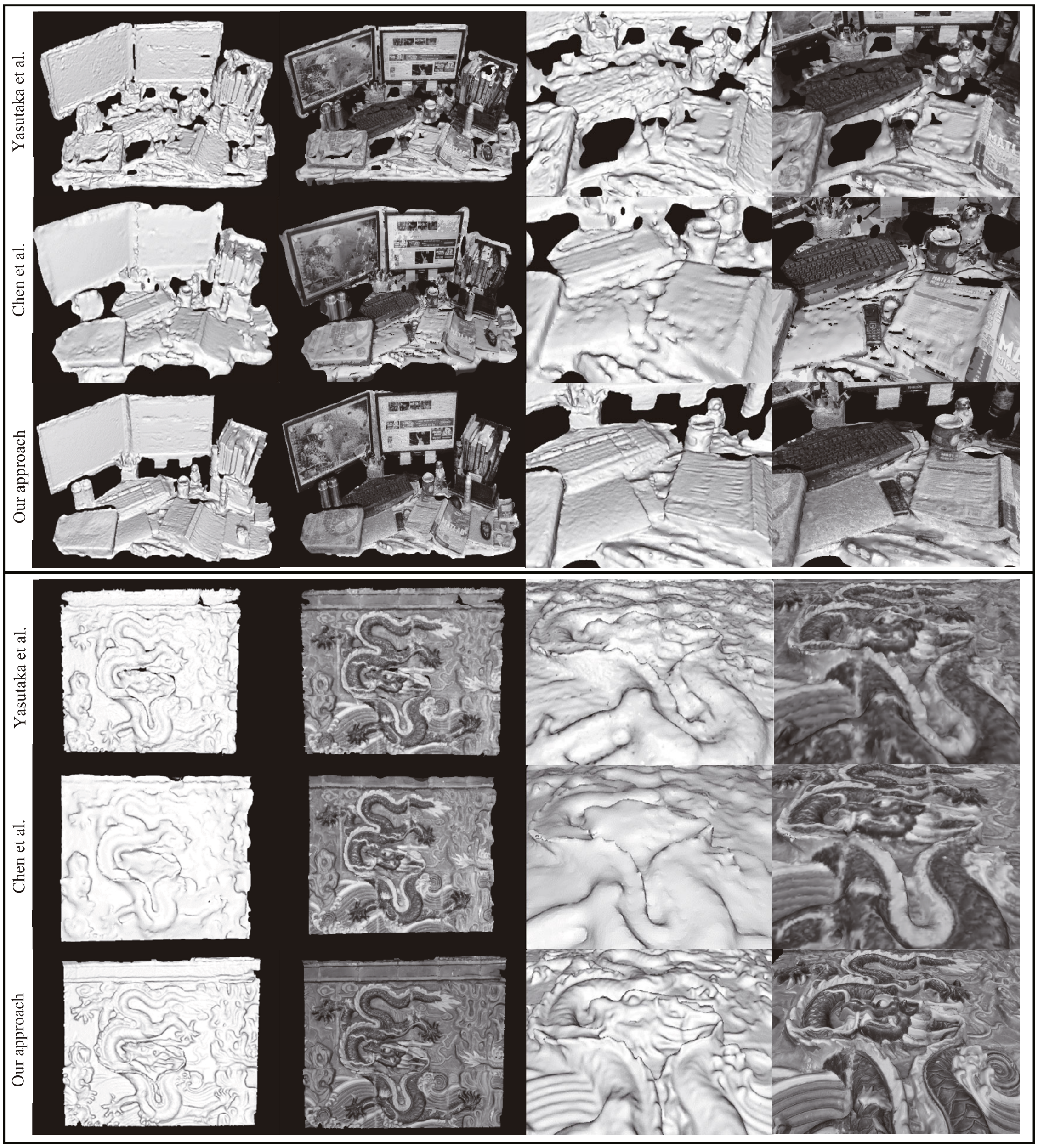

Figure 5 Some comparative reconstruction results (“Office” and “Dragon”) reconstructed by Yasutaka et al. [9], Chen et al. [7] and ours.

a scene is captured in the camera, the better the reconstructed result will be, because there are fewer view data lost during the reconstruction. In the experiment, the parameters mentioned in our paper were set to $\mu = 0.5$, $\varepsilon_c = \varepsilon_g = 10$, $\sigma_f = 20$, $M_f = 100$. The other parameters vary with different scenes and camera coordinates.

Figure 3 gives the recovered depth results between our study and other methods. To further evaluate the effectiveness of our approach, we made some comparative evaluations of the final reconstructed results. All of the three methods in Figure 5 are finished into a dense mesh from the point cloud by Poisson surface reconstruction. We note that the Poisson reconstruction algorithm can fill in the holes by adjusting the locations of the points in the cloud. Therefore, in order to prove that our approach can produce better point cloud data, we have deleted the rectified faces in Figure 5, while trying to maintain the original appearance of the reconstructed point cloud. Certainly all the existing or future surface-reconstruction

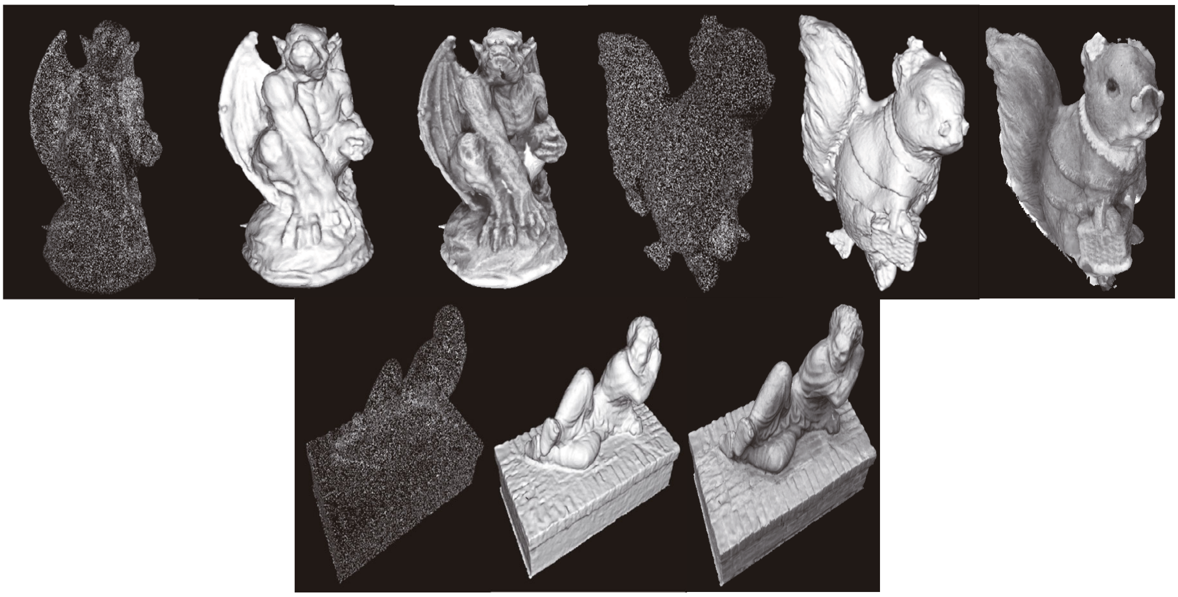

Figure 6 More reconstruction results: “Gargoyle”, “Squirrel” and “Derhass”. The above test image sequences are downloaded from http://www.gris.informatik.tu-darmstadt.de/projects/multiview-environment/datasets.html and http://www-scf.usc.edu/ zkang/software.html.

Table 1 Run-time records for different cases

Sequence	Sum of images	Run-time (min)		
		Yasutaka	Chen	Ours
Step	25	4.1	5.2	5.9
Office	27	4.2	5.2	6.1
Dragon	9	1.5	1.8	2.0
Gargoyle	23	3.7	4.8	5.3
Squirrel	23	3.8	4.9	5.4
Derhass	79	13.6	16.3	18.1

methods can be replaced here, and our intention in this paper is to produce better-reconstructed point cloud data as far as possible.

From the results, it can be concluded that Yasutaka’s method [9] is incapable of dealing with those textureless regions (e.g., notebooks with uniform color) because no feature points are detected as the seeds used to expand the reconstruction. While for Chen’s method [7], the details of the object are smoothed too much by their feature-mesh restriction. The above two problems have been handled well by our approach: all the textureless regions, the details of the keyboard and the dragon wall are recovered sufficiently well in comparison with the other two methods. It is also worth mentioning that our reconstruction is robust to the outliers due to the contribution from the confidence estimation, which removes the outlier-noise from the range map. There are more test results shown in Figure 6 as well, and the more image-views will give a better reconstruction result. The detailed image number of each sequence and the run-time have been recorded in Table 1.

All the run-time records are measured during the process between SFM and producing the point cloud data (not including SFM). In order to be more intuitive, we also illustrate the data of the table as a graph in Figure 7. It is obvious that Yasutaka’s method has the fastest run-time because they reconstruct the object directly with many patches from multiple images. Both our approach and that proposed by Chen et al. have to compute the depth map first and then merge the range maps. In our experiments, we find that the run-time for this depth-based method is mostly taken up by the depth computation. What is more, the confidence map is estimated further in our work, which takes up more time. However, we can

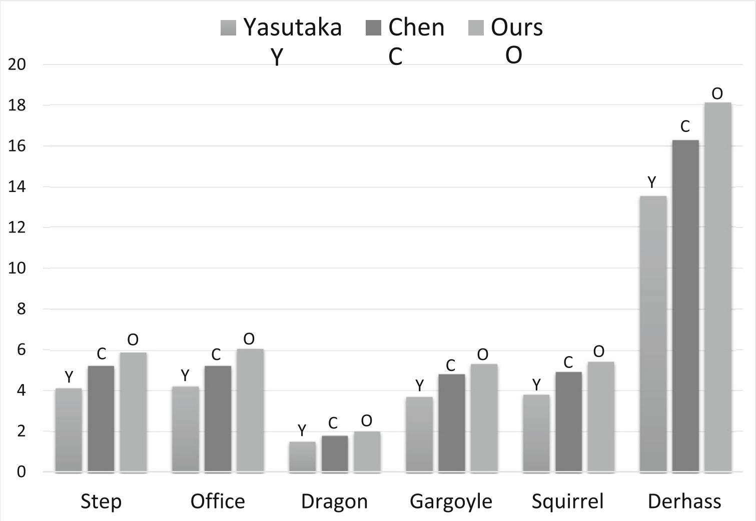

Figure 7 Run-time illustration.

see from the reconstruction results that the additional computational cost is meaningful and necessary.

6 Conclusion and limitations

Conclusion. We present a method of 3D reconstruction from the input of an image sequence. First of all, we review some state-of-the-art work about video reconstruction. Then we list the shortages that exist in other methods. In order to address these issues, we design our reconstruction system to undergo 4 stages, where the positions of the camera and sparse 3D points are first estimated by the SFM algorithm, then the depth and confidence maps are computed for each image. After that, all the depth maps are merged into the point cloud data according to the confidence of the depth, and we adopt the Poisson reconstruction algorithm to finish the process. Additionally, texture mapping is also implemented as a post-processing job for a good visual effect. The related results have been demonstrated in the experimental section of our work.

Limitation. In our experiments, we find that the efficiency of our system is affected mostly at the stages of the depth and confidence computations. Thus, those parts of our work might be bottlenecks of the efficiency of this system.

Future work. We exploit such a complicated system with a very limited capacity and there must be many details in our codes that can be improved greatly in future work in order to improve the efficiency. What is more, there are many parallel calculations involved in this work, which implies that our system can be transplanted to the GPU platform for acceleration.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61272326) and Startup Foundation for Introducing Talent of NUIST (Grant No. 2243141601013).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Chen J W, Bautembach D, Izadi S. Scalable real-time volumetric surface reconstruction. ACM Trans Graphic, 2013, 32: 1–16

2 Chen Y D, Hao C Y, Wu W, et al. Recursive video segmentation (in Chinese). Sci Sin Inform, 2014, 44: 1361–1369

3 Hao C Y, Chen Y D, Wu W, et al. Image completion with perspective constraint based on a single image. Sci China Inf Sci, 2015, 58: 092109

4 Collins R T. A space-sweep approach to true multi-image matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, 1996. 358–363

5 Seitz S M, Dyer C R. Photorealistic scene reconstruction by voxel coloring. Int J Comput Vision, 1999, 35: 151–173

6 Newcombe R A, Davison A J. Live dense reconstruction with a single moving camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, 2010. 1498–1505

7 Chen Y D, Hao C Y, Wu W, et al. Live accurate and dense reconstruction from a handheld camera. Comput Animat Virtual Worlds, 2013, 24: 387–397

8 Furukawa Y, Ponce J. Accurate, dense, and robust multiview stereopsis. IEEE Trans Pattern Anal Mach Intell, 2010, 32: 1362–1376

9 Furukawa Y, Curless B, Seitz S M, et al. Towards internet-scale multi-view stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, 2010. 1434–1441

10 Wang Y, Ji X Y, Dai Q H. Key technologies of light field capture for 3d reconstruction in microscopic scene. Sci China Inf Sci, 2010, 53: 1917–1930

11 Agarwal S, Furukawa Y, Snavely N, et al. Building rome in a day. Commun ACM, 2011, 54: 105–112

12 Sun J, Li Y, Kang S B, et al. Symmetric stereo matching for occlusion handling. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, 2005. 399–406

13 Strecha C, Fransens R, van Gool L. Combined depth and outlier estimation in multi-view stereo. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, 2006. 2394–2401

14 Ryan K, Camillo J T. Optical flow with geometric occlusion estimation and fusion of multiple frames. In: Proceedings of the 10th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition, Hongkong, 2015. 364–377

15 Zhang G F, Jia J Y, Hua W, et al. Robust bilayer segmentation and motion depth estimation with a handheld camera. IEEE Trans Pattern Anal Mach Intell, 2011, 33: 603–617

16 Felzenszwalb P F, Huttenlocher D P. Efficient belief propagation for early vision. Int J Comput Vision, 2006, 70: 41–54

17 Woodford O, Torr P, Reid I, et al. Global stereo reconstruction under second-order smoothness priors. IEEE Trans Pattern Anal Mach Intell, 2009, 31: 2115–2128

18 Fuhrmann S, Goesele M. Fusion of depth maps with multiple scales. In: Proceedings of the SIGGRAPH Asia Conference. New York: ACM, 2011. 1–8

19 Zach C, Pock T, Bischof H. A globally optimal algorithm for robust TV-L1 range image integration. In: Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, 2007. 1–8

20 Stühmer J, Gumhold S, Cremers D. Real-time dense geometry from a handheld camera. In: Proceedings of the Conference on Pattern Recognition, Lecture Notes in Computer Science, Berlin, 2010. 11–20

21 Graber G, Pock T, Bischof H. Online 3d reconstruction using convex optimization. In: Proceedings of IEEE International Conference on Computer Vision Workshops, Barcelona, 2011. 708–711

22 Bischof H. Dense reconstruction on-the-fly. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, 2012. 1450–1457

23 Shan Q, Curless B, Furukawa Y, et al. Occluding contours for multi-view stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, 2014. 4002–4009

24 Snavely N, Seitz S M, Szeliski R. Photo tourism: exploring photo collections in 3D. In: Proceedings of the ACM SIGGRAPH, New York, 2006. 835–846

25 Wu C C, Agarwal S, Curless B, et al. Multicore bundle adjustment. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, 2011. 3057–3064

26 Dong Z L, Zhang G F, Jia J Y, et al. Efficient keyframe-based real-time camera tracking. Comput Vision Image Und, 2014, 118: 97–110

27 Kazhdan M, Hoppe H. Screened Poisson surface reconstruction. ACM Trans Graphic, 2013, 32: 1–13

28 Kang S B, Szeliski R. Extracting view-dependent depth maps from a collection of images. Int J Comput Vision, 2004, 58: 139–163