Bundled Depth-Map Merging for Multi-View Stereo

Jianguo Li, Eric Li, Yurong Chen, Lin Xu, Yimin Zhang
Intel Labs China, Haidian, Beijing, 100190
{jianguo.li, eric.q.li, yurong.chen, lin.x.xu, yimin.zhang}@intel.com

Abstract

Depth-map merging is one typical technique category for multi-view stereo (MVS) reconstruction. To guarantee accuracy, existing algorithms usually require either sub-pixel level stereo matching precision or continuous depth-map estimation. The merging of inaccurate depth-maps remains a challenging problem. This paper introduces a bundle optimization method for robust and accurate depth-map merging. In the method, depth-maps are generated using DAISY feature, followed by two stages of bundle optimization. The first stage optimizes the track of connected stereo matches to generate initial 3D points. The second stage optimizes the position and normals of 3D points. High quality point cloud is then meshed as geometric models.

The proposed method can be easily parallelizable on multi-core processors. Middlebury evaluation shows that it is one of the most efficient methods among non-GPU algorithms, yet still keeps very high accuracy. We also demonstrate the effectiveness of the proposed algorithm on various real-world, high-resolution, self-calibrated data sets including objects with complex details, objects with large area of highlight, and objects with non-Lambertian surface.

1. Introduction

Multi-view stereo (MVS) has received more and more attention recently. Great progress has achieved since the Middlebury evaluation was provided [23]. Although some algorithms have achieved very high accuracy and some algorithms are very efficient, efforts are still valuable to design a method with a combination of high accuracy and high efficiency. Study of the application gamut is another valuable topic for MVS algorithms.

According to the taxonomy of Seitz et al [23], MVS algorithms can be divided into four categories: 3D volumetric based approaches [16, 29, 28, 24], surface evolution techniques [11, 33, 5, 22, 15, 4], depth-map merging based methods [7, 1, 26, 20, 32, 19, 3], and featured-region growing and expansion based algorithms [31, 6, 9, 18, 14]. Among these four categories, depth-map merging based method is highly flexible due to its capacity to integrate any available stereo matching techniques. This method typically has two steps. First, depth-maps are calculated from stereo image pairs. Second, the generated depth-maps are merged to produce 3D models. However, algorithms of this category are crucial to the accuracy of the estimated depth-map. They either require sub-pixel stereo matching accuracy [1], or require multiple or continuous depth-map estimation [3, 19]. It remains a challenging problem to merge inaccurate or even erroneous depth maps.

In this paper, we proposed a bundle optimization algorithm for high quality depth-map merging. First, we tailored a stereo matching module based on DAISY features [27]. As is known, DAISY is a local descriptor for dense stereo matching, and shown superior to normalized-cross-correlation (NCC) based method. Second, we designed a two-stage bundle optimization to merge depth-maps. The first stage groups the connected stereo matches among all stereo pairs into tracks. We built tracks in an optimized way to filter unreliable ones and obtain initial 3D points. The second stage optimizes the position and normals of initial 3D points using photo-consistency. Finally, 3D geometric model is generated from the optimized 3D point cloud using surface reconstruction techniques. The major contributions of this paper are as follows:

(1) We introduce bundle optimization for depth-map merging. Although the back-end techniques are already used in some other applications, we are the first to use bundle optimization for depth-map merging. Furthermore, the proposed bundle optimization is robust to noise since it incorporates unreliable tracks filtering within the optimization procedure.

(2) We study the application gamut of the proposed method on various real-world objects, including objects with complex details and non-Lambertian/highlight surface. The experiments demonstrate the effectiveness of the proposed method. This is one pioneer work in this field.

(3) We introduce DAISY features for multi-view stereo, and speed up the epipolar-line search with kd-tree.

Suppose the epipolar-line has n pixels, the complexity of our method is $O(n \log n)$ in each line, while that of NCC based method is $O(n^2)$.

(4) We design the algorithm to be naturally parallel processable and scalable to the complexity of problem and the number of processors/cores. The algorithm is shown one of the most efficient non-GPU methods.

The rest of paper is organized as follows. We first compare with related works in Section 2, and present the algorithm framework in Section 3. We then describe the tailored stereo matching modules in Section 4, and details of bundle optimization in Section 5. Experimental results on Middlebury data sets and real-world data sets are shown in Section 6. Some discussions are presented in Section 7, and conclusions are drawn in Section 8.

2. Comparison to Previous Works

There are four major kinds of related works. First, several depth-map merging methods should be mentioned. Campbell [3] proposes multiple depth hypotheses for each pixel, and globally optimizes them to guarantee depth-map smoothness and accuracy. Strecha [26] uses generative model to jointly solve depth and visibility for accurate depth-map generation. These two methods focus more on accurate depth-map generation. Goesele [7] uses NCC based pixel window matching techniques to produce depth-maps and merges them with volumetric integration. Bradely [1] enhances that work with sub-pixel scaled-window matching and dedicated outlier filtering algorithm. Liu [19] introduces continuous depth-map by variational optical flow. Our method is close related to these three methods, but differs from them in two aspects. First, we did not require sub-pixel level matching accuracy. Second, we merge depth-map using a bundle optimization technique, while all these three methods adopt a simple winner-take-all (WTA) scheme to merge depth-maps.

Second, besides the WTA scheme, several optimization based algorithms cannot be ignored. Merrell [20] presents the use of visibility consistence to merge depth-map on GPU. Zach [32] introduces total variation regularization and L_1 norm to measure data fidelity for depth-map merging. These methods require relatively clean point cloud, while our method filters unreliable matches and points within the bundle optimization framework.

Third, photo consistence based optimization has been used in several MVS algorithms, which take the distortion of matching window into account. Furukawa [31] considers 3D points as a patch, reprojecting pixels in matching window to 3D patches and further to the matched views for the evaluating of photo-consistence. This procedure alleviates the distortion of rectangle matching window derived from surface projection. More important, the optimization of photo-consistence leads to the optimal point position and normals. Habbecke [9] finds that the tangent plane of a 3D patch induced a homography which can directly map pixels from one view to another. With this property, the optimization can be done very efficient with approaches like forward-additive and inverse compositional image alignment algorithms [8]. We borrow the idea of homography mapping from them. However, our method is based on DAISY features, while all these methods are based on NCC. In addition, we formulate a bound-constrained optimization for the problem instead of unconstrained optimization in existing algorithms.

Fourth, in the area of local features for stereo reconstruction, Harris points and DoG corner have been adopted by [31]. But they obviously work poor for textureless regions, and thus rely on region growing and expansion techniques [31, 9]. Vu [30] introduces even more interest points such as SIFT, regular grid points, etc. DAISY is a dense local descriptor recently proposed by Tola et al [27]. It is shown superior to NCC based window matching method for stereo matching. There is still absence of DAISY based MVS system. Our practice shows that DAISY works well on MVS problem from both accuracy and efficiency perspectives.

3. Algorithm Framework

We aim at building an efficient and accurate image-based 3D reconstruction system. Figure 1 illustrates the system framework. Basically, the system can be divided into two main modules: stereo matching and bundle optimization.

In the stereo matching part, we first did the stereo-pair selection according to the given camera parameters. Given a selected stereo pair, we rectified them and then did the correspondences search on epipolar-line using DAISY features. Suppose right image is the reference view, we adopt kd-tree to accelerate the search by indexing the DAISY features in the epipolar-line of right image. For each pixel in the corresponding epipolar line of left image, we returned matches in the reference image by kd-tree search.

The bundle optimization module aims to obtain high quality oriented point cloud. It consists of two stages:

(1) **Bundled track optimization**: A track is a set of connected stereo matches from all stereo pairs. Each track corresponds to a set of pixels and associating views. We optimized tracks with reprojection error. When the reprojection error of a view surpasses a threshold, the view is removed from the track. When the number of views in a track is less than a threshold (i.e., 3), the track is not reliable and thus deleted. Initial 3D point cloud is generated from reliable tracks.

(2) **Position refinement and normal estimation**: After track optimization, each point in the initial point cloud corresponds to a reliable track of views. We optimized

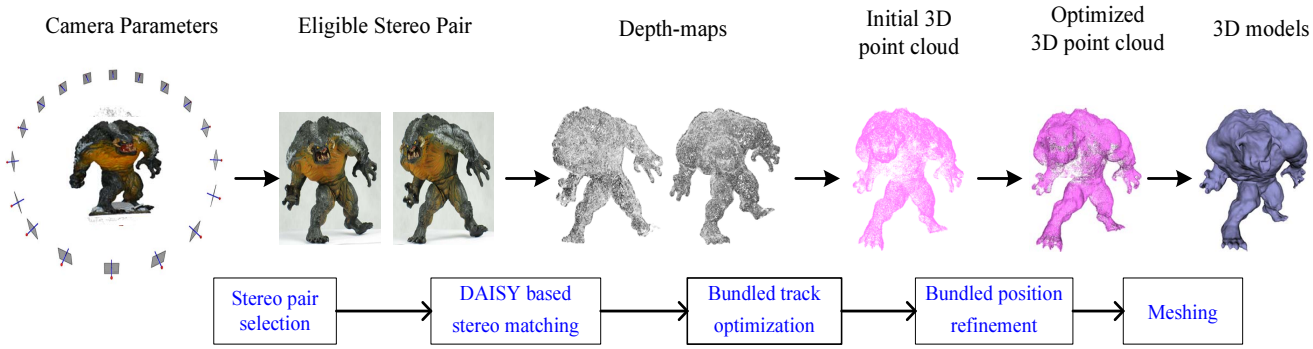

Figure 1. System framework of our approach

the total photo-consistency among the track of views to obtain accurate 3D point position and normals.

Once optimized 3D point clouds are available, watertight mesh may be generated using surface reconstruction algorithm like Poisson surface reconstruction (PSR) [21] and touch-expand algorithm [17][1]. In the following, we will focus on the two main modules in details.

4. Stereo Matching

Stereo matching is crucial to the accuracy of MVS system. In our case, the reason is not due to the precision (sub-pixel level) of matches, but due to the stability or reliability of matches. Even if the matches does not have sub-pixel precision, they could be refined with our bundle optimization algorithm. Hence, we still put some efforts on stereo matching to improve the reliability of matches. The goal is achieved from two sub-modules: eligible stereo pair selection and tailored dense matching.

4.1. Stereo-pair selection

Not all image pairs are eligible for stereo matching, and the selection of stereo pair will not only improve the accuracy of final MVS results, but also impact the time performance of the system.

Given calibrated camera parameters, the pair selection is based on two statistics from i-th possible image pair: angle θ_i between principal view directions and distance d_i between camera optical centers. Eligible stereo pairs are determined by the following rules sequentially:

(1) θ_i satisfied $5^\circ < \theta_i < 45^\circ$.

(2) For those pairs passed the first rule, suppose the median of d_i is $\bar{d}$, pair-i will be removed if $d_i > 2\bar{d}$ or $d_i < 0.05\bar{d}$.

(3) Each view has at most two stereo pairs according to the metric $\theta_i \cdot d_i$ in ascending order.

[1]Both can give good results, while PSR will give smoother results.

Suppose the number of images is n, the number of eligible stereo pair is less than $2n$.

When cameras are self-calibrated, we further utilized available stable sparse matches from structure-from-motion procedure to refine θ_i. As each match corresponds two viewing rays, the average angle between viewing-rays over all matches is used to replace θ_i.

4.2. Dense matching

For each eligible stereo pair, the image views are first rectified such that epipolar line corresponds to scan-line in images. Suppose the right image is the reference view, for each pixel in the left image, stereo matching find the closed matching pixel on the corresponding epipolar-line in the right image. The matching is based on DAISY features, which is shown superior to NCC based method in dense stereo matching [27].

We adopted kd-tree to accelerate the epipolar-line search. First, we extracted DAISY features for each pixel on the scan-line of right image, and indexed these features using kd-tree. For each pixel on the corresponding line of the left image, we returned top-K (K=10 in our practice) candidates in the right image by the kd-tree search. After the whole scan-line is processed, we further optimized intra-line results by dynamic programming within the top-K candidates. This scan-line optimization guarantees no duplicated correspondences within scan-line.

In fact, this idea is related to Campbell [3], which extracts multiple depth hypothesis for each image pixel, and then does global optimization. The differences are two folds. First, we focused on local consistence in this step and improved the results with the following bundle optimization step, while his method puts all pixels and corresponding hypothesis in a global optimal framework. Second, processing unit of our method is pixels in one scan-line, while that of his method is pixels of the whole reference image. The multiple depth hypothesis of every pixel yields larger memory consumption for his method.

The DAISY feature extraction on the scan-line can be

performed in parallel. Besides, the computation complexity is greatly reduced from NCC based method. Suppose the epipolar-line contains n pixels, the complexity of NCC based matching is $O(n^2)$ in one scan-line, while the complexity of our case is $O(2n \log n)$ since the kd-tree building complexity is $O(n \log n)$, and the kd-tree search complexity is $O(\log n)$ per query.

For the consideration of running speed on high resolution images, we defined a sampling step s (=1,2,$\cdots$) for the scan-line of left image, while keep searching every pixel in the corresponding line of reference image. For instance, s=2 means that we only find correspondences for every two pixels in the scan-line of left image.

When depth-maps are ready, we will filter unreliable matches. In details, we firstly filtered matches which the angle between viewing-rays falls outside the range $5^\circ \sim 45^\circ$. Second, we filtered matches that the cross-correlation of DAISY features is less than a certain threshold such as α = 0.8. Third, if optional object silhouettes are available, we enforced them to further filter unnecessary matches.

There are several parameters in DAISY features: local region radius R, number of ring Q, number of histograms in a ring T and the number of bins in each histogram H. We fixed Q=3, T=8, and H=4 in our experiments, while tuned R according to data sets. Note that H=8 may bring somewhat accurate results, but lead to more memory and time consuming. Hence, we did not use it in our experiments.

5. Bundle Optimization

This section will describe two stages of bundle optimization separately.

5.1. Track optimization

First, we gave a mathematical definition of track. Given n images, suppose x_1^k is a pixel in the 1st image, it matches to pixel x_2^k in the 2nd image, and further x_2^k matches to x_3^k in the 3rd image, $\cdots$. The set of matches t_k=$\{x_1^k, x_2^k, x_3^k, \cdots\}$ is called a track, which should correspond to the same 3D point. We imposed each track must contain pixels coming from at least β views (β = 3 in all our experiments). This constraint can ensure the reliability of tracks.

We first collected all possible tracks in the following way. Starting from 0-th image, given a pixel in this image, we recursively traversed connected matched pixels in all the other n-1 images. During the procedure, we marked every pixel with a flag when it has been collected by a track. This flag can avoid redundant traverse. We looped over all pixels in the 0-th image in parallel. When finished the 0-th image, we repeated the recursive traversing procedure on unmarked pixels in left images.

When tracks are built, we optimized each of them to get an initial 3D point cloud. Since some tracks may contain erroneous matches, direct triangulation will introduce outliers. We penalized views which have reprojection error surpassing a threshold γ (=2 pixels in all our experiments), and define the objective function for the k-th track t_k as follows:

$$\min \sum_{x_i^k \in t_k} w(x_i^k) \| x_i^k - P_i^k \hat{X}^k \|, \tag{1}$$

where x_i^k is a pixel from i-th view, P_i^k is the projection matrix of i-th view, $\hat{X}^k$ is the estimated 3D point of the track, and $w(x_i^k)$ is a penalty weight defined as follows

$$w(x_i^k) = \begin{cases} 1 & \text{if } \| x_i^k - P_i^k \hat{X}^k \| < \gamma, \\ 10 & \text{otherwise.} \end{cases}$$

The objective is minimized with the Levenberg-Marquardt algorithm. When the optimization is finished, we checked each track for the number eligible view, i.e., #($w(x_i^k)$==1). A track t_k is reliable if #($w(x_i^k)$==1) $\geq \beta$. We created initial 3D point clouds from reliable tracks.

The objective minimization is parallel performed on track-level. The whole track optimization scales well on multi-core platform.

5.2. Position refinement and normal estimation

Although the initial 3D point cloud is reliable, there are two problems. First, the point positions are still not quite accurate since our stereo matching does not have sub-pixel level precision. More important, the point cloud does not have normals. The second stage focuses on the problem of point position refinement and normal estimation.

As pointed out in [10], given a 3D point X and projection matrix of two views $P_1 = K_1[I, 0]$ and $P_2 = K_2[R, \mathbf{t}]$, the point X and its normal $\mathbf{n}$ form a plane $\pi : \mathbf{n}^T X + d = 0$, where d can be interpreted as the distance from the optical center of camera-1 to the plane. This plane is known as the tangent plane of the surface at point X. One pretty property is that this plane induces a homography

$$H = K_2(R - \mathbf{t}\mathbf{n}^T/d)K_1^{-1}.$$

As a result, distortion from matching of rectangle window can be eliminated via a homography mapping. Given 3D points and corresponding reliable track of views, we can compute total photo-consistence of the track based on homography mapping as

$$E_k = \sum_{i,j \in t_k} \| DF_i(x) - DF_j(H_{ij}(x; \mathbf{n}, d)) \|, \tag{2}$$

where $DF_i(x)$ means the DAISY feature at pixel x in view-i, and $H_{ij}(x; \mathbf{n}, d)$ is the homography from view-i to view-j with parameters $\mathbf{n}$ and d.

H_{ij} is solved as follows. Given two projection matrix $P_i = K_i[R_i, \mathbf{t}_i]$ and $P_j = K_j[R_j, \mathbf{t}_j]$, we transferred 3D

point X from world coordinate to the camera's coordinate by $X^i_{cam} = R_i X + \mathbf{t}_i$ and $X^j_{cam} = R_j X + \mathbf{t}_j$. Since $X = R_i^{-1}(X^i_{cam} - \mathbf{t}_i) = R_j^{-1}(X^j_{cam} - \mathbf{t}_j)$. With simple deduction, we obtain

$$H_{ij} = K_j\big(R_{ij} + \frac{\mathbf{t}_{ij}\mathbf{n}_{ij}^T}{\mathbf{n}_{ij}^T X^i_{cam}}\big)K_i^{-1}, \tag{3}$$

where $R_{ij} = R_j R_i^{-1}$, $\mathbf{t}_{ij} = \mathbf{t}_j - R_{ij}\mathbf{t}_i$ and $\mathbf{n}_{ij}$ is the normal of X^i_{cam} in the coordinate of camera-i.

Minimization E_k yields the refinement of point position and accurate estimation of point normals. In practice, the minimization is constrained by two items: (1) the reprojection point should be in a bounding box of original pixel; (2) the angle between normal $\mathbf{n}$ and the view ray $\overrightarrow{XO_i}$ (O_i is the center camera-i) should be less than 60° to avoid shear effect. Therefore, the objective is defined as

$$\begin{aligned} \min \quad & E_k, \\ s.t. \quad & (1)\ |\hat{x}_i - x_i| < \gamma, \\ & (2)\ \mathbf{n}_{ij} * \overrightarrow{X^i_{cam}O_i}/\|\overrightarrow{X^i_{cam}O_i}\| > 0.5, \end{aligned} \tag{4}$$

where $\hat{x}_i$ is the reprojection point of pixel x_i.

This optimization problem is solved by the bound constrained BFGS algorithm [2]. Bound constrained optimization is much more efficient than unconstrained algorithm since it requires much fewer objective function evaluation. In practice, we found the speedup may be more than 5x without loss of result accuracy.

To improve convergence of the minimization, the normal direction can be initialized with a rough normal estimation methods as [12, 13]. This paper adopts the grid search method suggested in [13]. Suppose $\mathbf{n}_0 = \overrightarrow{X^i_{cam}O_i}/\|\overrightarrow{X^i_{cam}O_i}\|$ is the starting point, we transfer $\mathbf{n}_0$ to spherical coordinates as $(\theta_0, \phi_0, 1)$. The rough gird is designed as $\theta_k = \theta_0 \pm k \cdot 10^\circ$ and $\phi_k = \phi_0 \pm k \cdot 10^\circ$, where $k = 0, 1, 2, 3, 4$.

Note that the optimization is parallel performed on each 3D point and associated track.

6. Experiments and Results

6.1. Implementation details

We implemented the proposed approach in C/C++ with dedicated performance optimization on X86 multi-core platform. The DAISY feature extraction is based on the source code by the authors [27]. The Levenberg-Marquardt algorithm is based on MINPACK. The bounded BFGS algorithm is based on L-BFGS-B [2]. Fortran codes are transferred to C code by f2c. We revised these packages to make them friendly to parallel processing. The proposed algorithm is naturally parallel processable. We used OpenMP to realize data-level parallelism. The 'data' here may indicate pixels in scan-line during stereo matching, tracks during track-optimization, and initial 3D points during position refinement and normal estimation. Besides, we wrote SIMD codes for some hotspots in modules like DAISY feature extraction and kd-tree search. Moreover, we did a lot of engineering works on memory management so that the system is able to handle high-resolution images and large-size data sets. For instance, the system allows compute DAISY feature in different resolutions, and provide strategy to cache some least-used data during bundle-optimization to disk and invoke them on demand. In summary, our system performs very efficient, and is shown one of the fastest non-GPU methods in the Middlebury evaluation.

[2] http://www.eecs.northwestern.edu/∼nocedal/lbfgsb.html

Table 1. Summary of parameters used in the experiments

Parameters	Value
sampling-step in scan-line of left image, s	1,2,$\cdots$
local region radius in DAISY, R	6∼15
DAISY parameters (Q/T/H)	3/8/4
correlation threshold of DAISY feature, α	0.8
top-K match candidates by kd-tree, K	10
minimum number of views in a track, β	3
maximum allowed reprojection error, γ	2 pixels

All our experiments have been carried out on an Intel Xeon workstation with 8-core 3.0GHz CPU and 8GB RAM.

Parameters used in our algorithm are summarized in Table 1. Most parameters are fixed except sampling-step s and the radius of DAISY region R. In practice, we set s = 1 for low resolution images, and s = 2 for high resolution images to speed up the whole system. The determining of R is related to two factors: the baseline of stereo pair and the property of texture on the objects. Generally, large baseline requires large R, and object with more textureless regions requires larger R.

6.2. Evaluation on Middlebury datasets

We demonstrated the effectiveness of the proposed algorithm on a number of datasets. First, the algorithm is quantitatively evaluated on two Middlebury datasets: *dinoRing* and *templeRing* [23]. In our experiments, we set R = 15 for *dinoRing*, and R = 9 for *templeRing*. Table 2 shows the quantitative results of our algorithms. Figure 2 further illustrates the reconstruction results of these two datasets.

Table 2. Results on Middlebury datasets. Note that the time is measured in seconds

Dataset	Accuracy	Completeness	Time (s)
dinoRing	0.43mm	99.7%	354
templeRing	0.64mm	98.2%	213

It shows that the result on *dinoRing* ranking within top 5 at the time of writing. This also shows that our method works fine on objects with textureless surface. The result

Figure 2. Reconstruction results of Middlebury data sets *templeRing* and *dinoRing*.

on *templeRing* is satisfied not only on the accuracy but also on the reconstructed fine details such as pillars of the temple. Nevertheless, the accuracy may be further improved since we did not tweak the parameters much during the evaluation. Besides, we realized that the radius R in bundle optimization can be different from that in stereo-matching. Furthermore, it would be better making R in bundle optimization adaptive to the size of local tangent plane at each 3D point. Our future work may take this point into account.

6.3. Results on real-world self-calibrated datasets

We also evaluated the proposed method on three real-world datasets. These datasets are captured with a Canon IXUS 970 digital camera. The image resolution ranges from 2Mpix to 6Mpix. The height of target objects ranges from 20cm to 2.5m. Specially, the camera parameters for each dataset are self-calibrated with a structure-from-motion tool developed by ourselves with technique similar to [25].

The target objects are well chosen to study the application gamut of the proposed algorithm. In summary, the data sets contains several dimension of properties: object with fine details and complex surface, object with non-Lambertian surface, and object with large area of highlight. The three datasets are *Monster*, *SculptFace*, and *IronLion*. Their properties are listed in Table 3. These datasets are fairly challenging. This application gamut study on them will verify whether our method is widely applicable.

Note that the background of *SculptFace* and *IronLion* is clutter, we only cropped a bounding box of the target for each image instead of making accurate silhouette. This may introduce noise in stereo matching, but are also able to verify the capability of our algorithm for outlier removing.

Table 3. Real-world datasets

Dataset	*Monster*	*SculptFace*	*IronLion*
#Image	34	14	17
Image size	1600×1200	2816×2112	2816×2112
Object height	∼20cm	∼1.8m	∼2.5m
Dimension	complex	highlight	non-Lambertian
Runtime(s)	440	410	570

The reconstruction results and example images of *Monster*, *SculptFace*, and *IronLion* are shown in Figure 3, 4, 5 separately [3]. For the *Monster* dataset, many fine details on the back of the *Monster* have been reproduced. For the *SculptFace* dataset, although there is not only large-area of highlight but also non-Lambertian surface, the reconstruction recovers major details of the target. This example also demonstrates that our algorithm is able to handle relief-like structures. In addition to non-Lambertian surface, the *IronLion* dataset has much larger base line than the other two. Our approach is still able to recover major detail of the object. There are some errors on the top of head and back of the lion due to insufficient photos available at these certain views. Considering that camera self-calibration must introduce some error, these results look quite well.

Both of the *SculptFace* dataset and the *IronLion* dataset contain highlight areas of metallic surface. The proposed algorithm shows the ability to handle highlight areas. This is due to three reasons. First, DAISY feature has more powerful capability to capture small difference than NCC based method as validated in [27]. Second, the scan-line optimization in dense-matching makes important compensation for mis-matching in highlight area. Third, the bundle optimization module can remove noise/outlier matches via the check of the reliability of tracks.

The proposed algorithm scales well on these high resolution datasets. The thread load balance in stereo matching is even better since there are more pixels in one scan-line. All these three datasets can be reconstructed within 10 minutes on the test platform as shown in Table 3. More specifically, the dense matching takes about 70% of runtime on these typical datasets, and bundle optimization constitutes most of the rest. Compared to these two modules, the time for surface reconstruction is almost negligible (<5%).

7. Discussion

There are two major modules (i.e., stereo matching and bundle optimization) in the proposed method. Regarding the replacement of DAISY with NCC based method in stereo matching, the results are two folds. First, when a

[3] For more details, please refer to our supplementary video.

Figure 3. Example images and reconstruction results of the *Monster* dataset.

Figure 4. Example images and reconstruction results of the *SculptFace* dataset.

simplest NCC based matching method is adopted, it will degrade the final accuracy since DAISY outperforms NCC in stereo matching. Second, when a tailored version like [1] is used to replace DAISY, it will largely increase the computing complexity as this method requires sub-pixel scale-window matching.

Regarding the replacement of bundle optimization with a simple merging strategy as in [20], this will bring even more critical problem because Merrell's method does not provide position refinement and normal estimation.

In summary, bundle optimization is more important in terms of 3D modeling quality because it not only removes outlier/noise matches, but also refines the position of 3D points and estimates their normals. However, DAISY based stereo matching is also an indispensable component in optimizing the quality. The whole pipeline takes both accuracy and efficiency into account.

8. Conclusion

In this paper, we presented a novel two-stage bundle optimization algorithm for depth-map merging based multi-view stereo reconstruction. We tailored DAISY features for fast stereo matching. Although the generated depth-maps do not have sub-pixel level precision or even erroneous, the two-stage optimization is able to simultaneously remove outliers and produce high-quality point clouds. Another major advantage is that the proposed approach is amenable to parallelization on multi-core system. With data-level parallelism, the approach is shown to be one of the most efficient methods among non-GPU algorithms. The accuracy of the proposed method is satisfied according to the Middlebury evaluation. Furthermore, robustness is guaranteed by the application gamut study on various real-world high resolution datasets, including object with complex details, object with non-Lambertian material, and object with large highlight area. Future work will focus on research like adaptive DAISY radius during bundle optimization and many-core/GPU based implementation.

Acknowledgements

We thank S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski for the setup of the Middlebury evaluation, and D. Scharstein for the great help of the evaluation of our results. Thanks also go to Intel colleagues Jim Hurley and Horst Haussecker for their supports to our project.

References

[1] D. Bradley, T. Boubekeur, and T. Berlin. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *CVPR*, 2008.

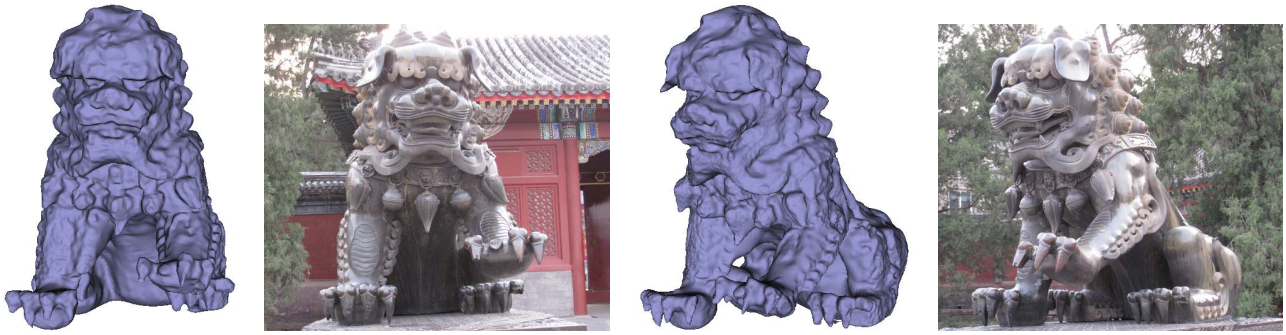

Figure 5. Example images and reconstruction results of the *IronLion* dataset.

[2] R. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208, 1995.

[3] N. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008.

[4] A. Delaunoy, E. Prados, P. Gargallo, J.-P. Pons, and P. Sturm. Minimizing the multi-view stereo reprojection error for triangular surface meshes. In *BMVC*, 2008.

[5] O. Faugeras and R. Keriven. surface evolution, PDE's, level set methods and the stereo problem. *IEEE TIP*, 1998.

[6] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M.Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007.

[7] S. Goesele, B. Curless, and S. Seitz. Multi-view stereo revisited. In *CVPR*, 2006.

[8] M. Habbecke and L. Kobbelt. Iterative multi-view plane fitting. In *VMV*, 2006.

[9] M. Habbecke and L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *CVPR*, 2007.

[10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, chapter 13. Cambridge press, 2nd edition, 2003.

[11] C. Hernandez and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 96, 2004.

[12] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH*, 1992.

[13] M. Jancosek and T. Pajdla. Effective seed generation for 3D reconstruction. In *Computer vision winter workshop*, 2008.

[14] M. Jancosek and T. Pajdla. Segmentation based multi-view stereo. In *Computer Vision Winter Workshop*, 2009.

[15] K. Kolev, M.Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3D reconstruction. *IJCV*, 2009.

[16] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV*, 2002.

[17] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *ICCV*, 2007.

[18] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE PAMI*, 27, 2005.

[19] Y. Liu, X. Cao, Q. Dai, and W. Xu. Continuous depth estimation for multi-view stereo. In *CVPR*, 2009.

[20] P. Merrell, A. Akbarzadeh, and et al. Real-time visibility-based fusion of depth maps. In *ICCV*, 2007.

[21] M. M.Kazhdan and H.Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, 2006.

[22] J.-P. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *CVPR*, 2005.

[23] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and et al. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. http://vision.middlebury.edu/mview/.

[24] S. N. Sinha, P. Mordohai, and M. Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *ICCV*, 2007.

[25] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 2007.

[26] C. Strecha, R. Fransens, and L. Gool. Combined depth and outlier estimation in multi-view stereo. In *CVPR*, 2006.

[27] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *CVPR*, 2008.

[28] S. Tran and L. Davis. 3D surface reconstruction using graph cuts with surface constraints. In *ECCV*, 2006.

[29] G. Vogiatzis, C. H. Esteban, P. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE PAMI*, 2007.

[30] H. Vu, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *CVPR*, 2009.

[31] Y.Furukawa and J.Ponce. Accurate,dense, and robust multi-view stereopsis. In *CVPR*, 2007.

[32] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust TV-L1 range image integration. In *ICCV*, 2007.

[33] A. Zaharescu, E. Boyer, and R. Horaud. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In *ACCV*, 2006.