Fast Dense 3D Reconstruction using an Adaptive Multiscale Discrete-Continuous Variational method

Zhuoliang Kang and Gérard Medioni
University of Southern California
Los Angeles, CA 90089
{zkang|medioni}@usc.edu

Abstract

We present a system for fast dense 3D reconstruction with a hand-held camera. Walking around a target object, we shoot sequential images using continuous shooting mode. High-quality camera poses are obtained offline using structure-from-motion (SfM) algorithm with Bundle Adjustment. Multi-view stereo is solved using a new, efficient adaptive multiscale discrete-continuous variational method to generate depth maps with sub-pixel accuracy. Depth maps are then fused into a 3D model using volumetric integration with truncated signed distance function (TSDF).

*Our system is accurate, efficient and flexible: accurate depth maps are estimated with sub-pixel accuracy in stereo matching; dense models can be achieved within minutes as major algorithms parallelized on multi-core processor and GPU; various tasks can be handled (*e.g. *reconstruction of objects in both indoor and outdoor environment with different scales) without specific hand-tuning parameters. We evaluate our system quantitatively and qualitatively on Middlebury benchmark and another dataset collected with a smartphone camera.*

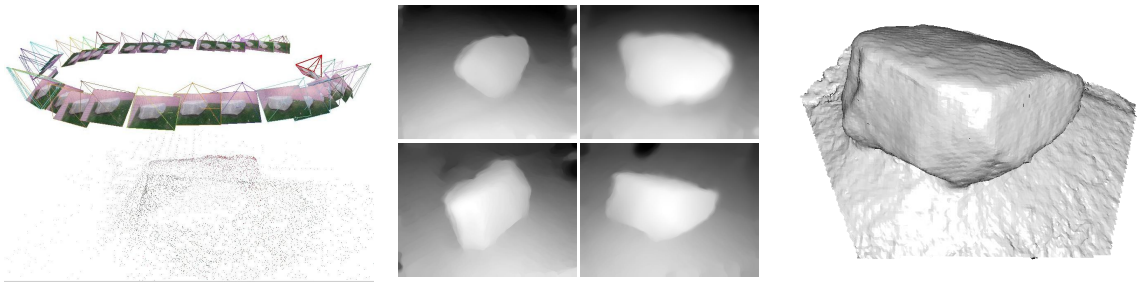
1. Introduction

Accurate and efficient dense 3D reconstruction with a hand-held camera has become possible due to the recent progress in computer vision technology and GPU hardware. Specifically, smartphone cameras have become a powerful at-hand source of images for everyone as smartphones getting popular. In this paper, we present a system for fast dense 3D reconstruction based on images captured using a smartphone camera.

Previously, there are mainly two ways to obtain images for 3D reconstruction using a hand-held camera. People can use either single shot mode to capture a sparse set of images click-by-click, or video mode to capture a video around the target. Recently, continuous shooting mode has become an internal function for many smartphone cameras. Under continuous shooting mode, users can capture sequential images in a quick succession by simply holding the shutter bottom down. It usually comes with a frame-rate lower than video (*e.g.*, can be adjusted between 1Hz to 10Hz). Compared with single shot mode, it provides enough images for dense reconstruction with one-click which makes the system user-friendly. Compared with video mode, the captured images suffer less motion blur as each image is captured by a quick open-close of the camera shutter. An example of several sequential images captured under continuous shooting mode is shown in Figure 1.

Figure 1. An example of sequential images captured walking around a target object, using a smartphone camera under continuous shooting mode.

Our system uses continuously shot images as input. Accurate camera calibration is achieved using structure-from-motion (SfM) algorithm with Bundle Adjustment [19]. A depth map for each image is then computed via multi-view stereo using a novel adaptive multiscale discrete-continuous variational method. In stereo matching, the sampling range is adjusted adaptively in a coarse-to-fine manner to ensure sub-pixel accuracy. Finally, obtained depth maps are fused into a 3D model using volumetric integration with truncated signed distance function (TSDF) [5]. All these modules are parallelized on multi-core processor and GPU, which makes the system fast. We evaluate our system on various datasets including targets in both indoor and outdoor environment with different scales. Results show that our system is flexible for different tasks and able to generate very accurate 3D models within minutes.



(a) (b) (c)

Figure 2. Pipeline of our system. (a). Images are captured around the target with $6D$ camera poses estimated using Bundle Adjustment. (b). An inverse depth map for each image is computed via multi-view stereo algorithm. (c). Estimated depth maps are fused into a 3D model via volumetric integration.

In the rest of this paper, we introduce related works in section 2, and explain each module of our system in section 3. In section 4, experimental results on various datasets are provided. Finally, we conclude our paper and present thoughts for future directions in section 5.

2. Related work

There are many works on dense 3D reconstruction have been proposed. In dense simultaneous-localization-and-mapping (SLAM), camera tracking and dense reconstruction are solved simultaneously based on video input [15, 14]. Another group of works aims to build dense 3D models from images. Most of them share two major parts: camera pose estimation and dense reconstruction. The state-of-the-art work for camera pose estimation is feature-based SfM with Bundle Adjustment. The pipeline is composed of three steps: feature points, such as SIFT [13], are detected and matched; relative pose methods, such as 5-point method [16], are used to generate initial camera poses with RANSAC scheme deal with false matches; finally, a global Bundle Adjustment is used to refine the results [19].

The state-of-the-art work on dense reconstruction is proposed by Furukawa and Ponce, which is a patch-based stereo algorithm generating dense models via image patch matching and expansion iteratively [8, 6]. Hiep *et al*. [10] proposed a pipeline for large-scale, detailed reconstruction in which a photometric consistent mesh is generated from an initial dense point cloud with variational optimization.

Several works have been proposed to couple camera pose estimation with dense reconstruction. Furukawa and Ponce proposed an approach to iteratively refine the camera poses and process patch-based reconstruction [7]. The camera pose refinement is achieved with a standard Bundle Adjustment involving all images. In [20], initial camera poses are refined iteratively with the feedback from optical flow between the texture on rendered surface and the actual images.

These works achieve sub-pixel accuracy by iteratively refining the obtained model, which is time-costly. We build the dense model efficiently by estimating a depth map for each image and fusing them together. Our novel adaptive multiscale discrete-continuous variational method is used to ensure sub-pixel accuracy in stereo matching.

3. Approach

3.1. Overview

The pipeline of our system is illustrated in Figure 2. In the beginning, images are captured sequentially using a smartphone camera walking around the target. Camera poses of obtained images are estimated using SfM algorithm with Bundle Adjustment applied to ensure the camera pose consistency (Figure 2(a)). With the obtained camera poses, a depth map of each image is computed using multi-view stereo algorithm (Figure 2(b)). In our system, a discrete-continuous variational method is used and parallelized in GPU. We also adaptively adjust the sampling range with a coarse-to-fine strategy to ensure sub-pixel accuracy in stereo matching. Finally, we integrate the computed depth maps into a 3D model via volumetric integration with TSDF [5](Figure 2(c)). In the rest of this section, we explain each part in detail.

3.2. Camera pose estimation

After obtaining sequential images around the target, the first step is to compute the $6D$ camera pose (position and orientation) for each image (Figure 2(a)). High-quality camera poses that are consistent in a global coordinate are essential for accurate dense reconstruction [15]. Several camera trackers, such as PTAM [11] and DTAM [14], have been proposed to do camera pose estimation for videos. They are able to provide camera poses for each video frame in real-time, with quality good enough for applications such

as augmented reality.

We aim to build accurate 3D models that are complete and consistent from different views, whereas these real-time camera trackers suffer drift problem due to the lack of a global optimization involving all cameras. In our system, we use feature-based SfM algorithm with Bundle Adjustment to generate high-quality camera poses. Bundle adjustment is achieved by minimizing the total reprojection error involving all cameras and a sparse set of 3D points [9]:

$$\mathbf{E}_{\theta_i, X_j} = \sum_i \sum_j d(\mathbf{P}(\theta_i, X_j), x_{ij})^2 \quad (1)$$

where θ_i is the camera parameters of i-th image, X_j is the 3D position of j-th point, x_{ij} is the actual position of j-th point on i-th image, $\mathbf{P}(\theta_i, X_j)$ is the predicated projection of j-th point on i-th image. $d(\mathbf{P}(\theta_i, P_j), x_{ij})$ denotes the geometric image distance between predicated position and actual position. By optimizing this energy function involving all cameras, estimated camera poses are ensured to be consistent in a global coordinate. With the recent progress in parallelized Bundle Adjustment algorithm [21], we are able to obtain accurate camera poses for dozens of frames in minutes.

3.3. Multi-view stereo

Given the camera poses, we now aim to estimate a metrically accurate depth map for each image using multi-view stereo algorithm. Finding dense correspondences between images in uncontrolled environment is difficult due to texture-less regions, occlusions and lack of contrast. In our system, we use a discrete-continuous variational method with a smoothness regularizer filling in texture-less regions. Photometric evidence from multiple views as well as epipolar geometry are used to constraint the problem and reduce ambiguities. Another challenge is to adapt the algorithm to various cases, such as targets in indoor and outdoor environment with different scales. The target depth range can vary significantly. We apply a coarse-to-fine strategy to adjust the sampling range adaptively so that sub-pixel accuracy of stereo matching is achieved for various cases.

3.3.1 Discrete-continuous variational method

We use the discrete-continuous variational method proposed in [14] for multi-view stereo matching. Given camera poses of an image I_0 and N neighboring images I_1 to I_N, we compute the inverse depth map of I_0 by minimizing an energy function including a non-convex photometric error term and a convex regularization term:

$$\mathbf{E}_{\mathbf{h}} = \sum_{\Omega} \lambda C(\mathbf{x}, \mathbf{h}(\mathbf{x})) d\mathbf{x} + \sum_{\Omega} \|\nabla \mathbf{h}(\mathbf{x})\|_{\epsilon} d\mathbf{x} \quad (2)$$

where $\mathbf{h}(\mathbf{x})$ represents the inverse depth of pixel $\mathbf{x}$. Ω is the 2D image domain of I_0. $C(\mathbf{x}, \mathbf{h}(\mathbf{x}))$ represents the photometric error term measuring the average intensity error across its neighboring views:

$$C(\mathbf{x}, h) = \frac{1}{N} \sum_{i=1}^{N} \left| I_0(\mathbf{x}) - I_i(\pi_i(\pi_0^{-1}(\mathbf{x}, h))) \right| \quad (3)$$

where $\pi_0^{-1}(\mathbf{x}, h)$ is the operator to compute the position of the 3D point back-projected from pixel $\mathbf{x}$ on I_0 when assigned to inverse depth value h, and $\pi_i(\pi_0^{-1}(\mathbf{x}, h))$ is the operator to compute the pixel location on I_i projected from this 3D point. $\|\nabla \mathbf{h}(\mathbf{x})\|_{\epsilon}$ is a Huber norm regularization term used to smooth the generated inverse depth map while reserving boundary discontinuities at the same time. Different from a pure Total-Variation regularization term, Huber norm allows smooth variation in small-scale to avoid the staircase effect.

To optimize Equation 2, we couple the photometric error term and regularization term with an auxiliary variable $\mathbf{h}'$:

$$\mathbf{E}_{\mathbf{h},\mathbf{h}'} = \sum_{\Omega} \left\{ \lambda C(\mathbf{x}, \mathbf{h}(\mathbf{x})) + \|\nabla \mathbf{h}'(\mathbf{x})\|_{\epsilon} \right\} d\mathbf{x} + \sum_{\Omega} \frac{1}{2\theta} (\mathbf{h}(\mathbf{x}) - \mathbf{h}'(\mathbf{x}))^2 d\mathbf{x} \quad (4)$$

which can be solved by optimizing $\mathbf{h}$ and $\mathbf{h}'$ alternatively.

When fixing $\mathbf{h}$, the problem becomes similar to the Total-Variation ROF image denoising problem [17], except that we are using a Huber norm as regularization term:

$$\mathbf{E}_{\mathbf{h}'} = \sum_{\Omega} \|\nabla \mathbf{h}'(\mathbf{x})\|_{\epsilon} d\mathbf{x} + \sum_{\Omega} \frac{1}{2\theta} (\mathbf{h}(\mathbf{x}) - \mathbf{h}'(\mathbf{x}))^2 d\mathbf{x} \quad (5)$$

which can be solved using a primal-dual approach continuously [4].

When fixing $\mathbf{h}'$, there is no coupling between neighboring pixels in the energy function any more:

$$\mathbf{E}_{\mathbf{h}} = \sum_{\Omega} \lambda C(\mathbf{x}, \mathbf{h}(\mathbf{x})) d\mathbf{x} + \sum_{\Omega} \frac{1}{2\theta} (\mathbf{h}(\mathbf{x}) - \mathbf{h}'(\mathbf{x}))^2 d\mathbf{x} \quad (6)$$

The optimal solution of $\mathbf{h}$ can be found pixel-wisely by searching exhaustively over a finite range of sampled inverse depth values. An $M \times N \times S$ discrete cost volume $\mathbf{C}$ is computed before optimization, containing photometric error for each pixel with linearly sampled inverse depth values between determined range h_{min} and h_{max}. $M \times N$ is the image resolution, and S is the number of points with linearly sampled inverse depth values. The photometric error is computed as defined in Equation 3. How to determine the sampling range of inverse depth values is explained in next section. More details of the primal-dual approach and discrete-continuous optimization can be found in [4, 14].

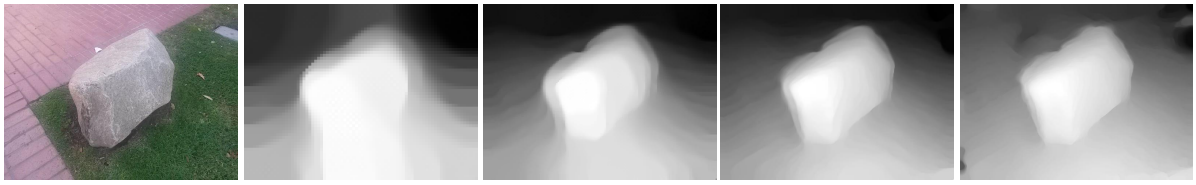


Figure 3. An example of the estimated inverse depth maps for the image pyramid with scale factor $\phi = 0.5$. From left to right: original image (size: 640×480), inverse depth map in level 3 (size: 80×60), level 2 (size: 160×120), level 1 (size: 320×240) and level 0 (size: 640×480).

Figure 4. Illustration of the adaptive adjustment of sampling range. Red line represents the sampling range along the epipolar line. Sampling range of finer level is adjusted to be centered around the estimated inverse depth value from coarser level. The sample step size is scaled by ϕ in each finer level, so that sub-pixel accuracy is achieved in each level with limited samples.

3.3.2 Adaptive sampling range adjustment

Choosing the proper sampling range of inverse depth values is a problem that needs to be addressed. The sampling range should be large enough to cover the entire target space. However, the number of sample points S is limited due to memory and computational cost in the exhaustive search step. The range should be narrow enough to ensure accurate stereo matching with limited sample points. In our system, we adjust the sampling range adaptively in a coarse-to-fine manner.

Specifically, we create an image pyramid with scale factor ϕ. The number of pyramid levels is defined so that the length of image diagonal in the coarsest level is smaller than S. The sampling range in the coarsest level is computed from the intersecting space of neighboring images. As an example illustrated in Figure 4, in each finer level, the sampling range of each pixel is adjusted to be centered around the estimated value from coarser level, with sampling step size scaled by ϕ. Linear sampling in the inverse depth leads to linear sampling along the epipolar lines in the image domain. By adjusting the sampling range in this coarse-to-fine manner, we ensure that sub-pixel accuracy is achieved in each level of stereo matching. As an example shown in Fig. 3, more structural details are added incrementally in each finer level. The choice of ϕ is given in section 3.5.

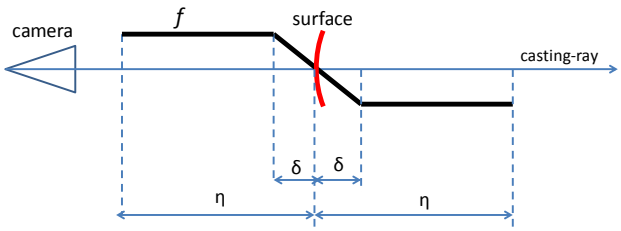


Figure 5. Illustration of the truncated signed distance function distributed along the casting ray. We define the distance as positive outside surface and negative inside surface.

3.4. Depth map fusion

After obtaining depth maps from different views, the last step is to fuse them into a 3D model. We use volumetric integration with truncated signed distance function (TSDF) to integrate the depth maps [5]. There are also robust volumetric integration methods have been proposed with regularization term to obtain smooth surfaces [23, 22]. In our system, the estimated inverse depth maps are accurate and smooth with the variational method. The integration method proposed in [5] works well for our cases without any further optimization over the obtained model.

A weighted signed distance function is defined in a discrete voxel grid $\Omega \in \mathbb{R}^3$. For each voxel $\mathbf{x} \in \Omega$, a cumulated signed distance value $D(\mathbf{x})$ and a cumulated weight $W(\mathbf{x})$ is assigned, where $D(\mathbf{x})$ represents its distance to the closest surface and $W(\mathbf{x})$ represents the confidence. During integration, each pixel of a depth map is converted into a truncated signed distance function $\mathbf{f}$ distributed along the casting ray to update the intersected voxels in Ω. As shown in Figure 5, the distance function $\mathbf{f}$ is cut in both side at distance η around the estimated surface to limit the distributing range of each pixel, which effectively reduces the impact of measurement error in the estimated depth maps. The same rule as in [5] is applied to incrementally update the signed distance value and weight for each voxel $\mathbf{x} \in \Omega$:

$$D_{i+1}(\mathbf{x}) = \frac{W_i(\mathbf{x})D_i(\mathbf{x}) + w_{i+1}d_{i+1}(\mathbf{x})}{W_i(\mathbf{x}) + w_{i+1}(\mathbf{x})} \qquad (7)$$

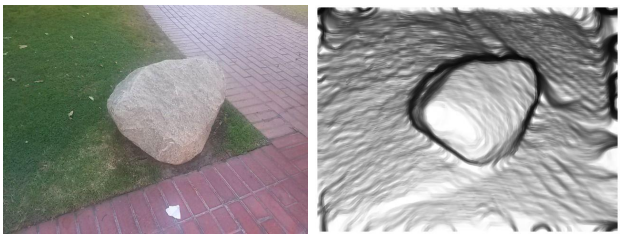


Figure 6. An example of weight map. From left to right: original image, computed weight map. Near-zero weight is assigned to depth discontinuities, which avoids the grazing ramps effectively.

$$W_{i+1}(\mathbf{x}) = W_i(\mathbf{x}) + w_{i+1}(\mathbf{x}) \tag{8}$$

where $D_i(\mathbf{x})$ and $W_i(\mathbf{x})$ are the signed distance value and weights after integrating i-th depth map. $d_{i+1}(\mathbf{x})$ and $w_{i+1}(\mathbf{x})$ is the signed distance value and weight distributed from $i+1$-th depth map. We compute the weight as the cosine of angle between the surface normal and optical ray to avoid grazing ramps. An example of computed weight map is shown in Figure 6. The color information of texture is fused into the voxel grid in the same way to ensure consistent texture appearance. After integration, a polygon mesh model can be extracted as the isosurface from the voxel grid using cube marching algorithm [12]. Voxels with a weight lower than a confidence threshold μ are treated as empty to remove noisy surfaces.

3.5. Parameter and implementation settings

The number of neighboring images N is set as 2 to reduce the impact of possible intensity variance across large baseline. The scale factor ϕ of image pyramids is set as 0.5. The number of points with sampled inver depth values S is set as 100. The parameters for discrete-continuous optimization is set according to [4, 14]. In optimization, we scale the inverse depth values with sampling step 1 so that a smoothness regularization weight $\lambda = 150$ works well across different datasets. The near-surface width δ is set to 1% of the voxel grid diameter, and the truncating distance $\eta = 3\delta$ as in [23]. The confidence threshold μ is set as 1.

We implement our system on a machine with Intel Xeon 3.6GHz quad-core processor and a GTX 590 graphics card. For camera pose estimation, we use VisualSFM [3] which is parallelized on multi-core processor and GPU with multi-core Bundle Adjustment [21]. The adaptive discrete-continuous variation method is also parallelized on GPU with CUDA implementation [2]. Volumetric integration is parallelized on multi-core processor.

4. Results

We provide results on two datasets. We quantitatively evaluate the performance of our system on Middlebury dataset, which is a benchmark with comparison to ground truth available [1, 18]. We also collect another dataset using a smartphone camera to demonstrate the performance and flexibility of our system in practical use.

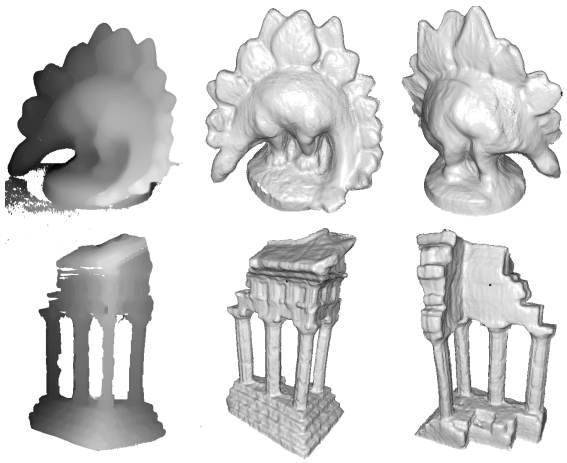


Figure 7. Results on Middlebury dataset. Top: results on "Dino ring" dataset. Bottom: results on "Temple ring" dataset. From left to right: an example of estimated inverse depth map, shaded model (frontal view), shaded model (back view)

4.1. Middlebury dataset

We evaluate our system on "Dino ring" and "Temple ring" from Middlebury multi-view dataset [1]. This dataset is captured and calibrated in well-controlled indoor environment, which is different from our target applications. However, the results is still meaningful as a performance evaluation. We implement our system with the parameter setting as described in section 3.5. The dark background pixels are thresholded out with a maximum intensity value 10. In volumetric integration, these background pixels are assigned a zero confidence value. The dataset description and run-time statics are listed in Table 1, and reconstruction results shown in Figure 7.

As a benchmark dataset, quantitative evaluation comparing to a ground truth model is also provided (refer to [18] for evaluation details). The results and computation statics of our system are uploaded in the evaluation page [1]. Because the smoothness constraint is used for the computation of depth maps, the reconstructed 3D models are smooth and complete without any optimization on voxel grid as in [23, 22], or refinement of the obtained polygon model as in [10, 20]. Our system is among the most efficient methods with 3 minutes run-time for each dataset.

For "Temple ring", we achieve the state-of-the-art performance with accuracy in 0.59mm and completeness of 97.9%. For "Dino ring", our system has an accuracy value of 0.69mm with a completeness value of 92.4%. The error are mainly contributed from the mismatches in the back views of "dino" due to the specular reflection, which is dif-

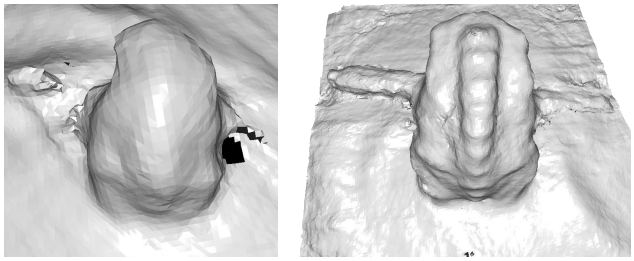


Figure 8. Close-up views of the reconstructed model with images captured from different distance. Left: results on "frog (far)". Right: results on "frog (near)". The structural details on the frog back are lost in "frog (far)" and well captured in "frog (near)" with a fixed image resolution (640×480 in this case).

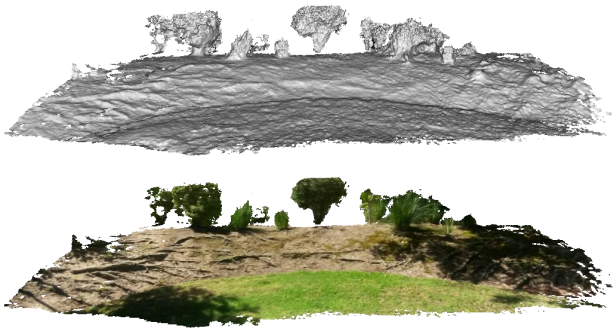


Figure 9. Results on "shrubs" dataset. Top: shaded polygonal model. Bottom: textured model.

ficult to solve without a time-consuming refinement of the obtained model as in [8].

4.2. Smartphone datasets

We collect another dataset using the camera on a Samsung Galxy S4 smartphone, including both indoor and outdoor targets with different scales. All the reconstruction results are obtained with the parameter setting as described in section 3.5. The dataset description and run-time statics are shown in Table 1.

Reconstruction of objects: We evaluate our system on objects under both indoor and outdoor environments. "plaster statue" is captured around a plaster head under indoor environment. "stone" and "table" are two sets of continuously shot images around target objects under outdoor environment. As results shown in Figure 10(a)(b)(c), objects in both indoor and outdoor environment can be handled well without any silhouette information available. The completeness of the reconstructed model depends on whether different views are covered by the captured images. In "stone" and "plaster statue", enough views are obtained to fully cover the object, which leads to a complete 3D model. In "table", the bottom side of the table was occluded, which leads to empty space in the reconstructed model.

Objects with different scales: To evaluate the flexibility of our system dealing with objects in difference scales, we collect two pairs of datasets targeting the same object captured from different distances. "iron statue (far)" and "iron statue (near)" is a pair of image sequences captured around a iron head statue under outdoor environment, with results shown in Figure 10(d)(e). "frog (far)" and "frog (near)" are captured around a wood frog under indoor environment with reconstruction results shown in Figure 10(f)(g).

With a limited image resolution, structural details are lost in large scale reconstruction with images captured from a decent distance, *e.g.*, the back of the wood frog as shown in Figure 8(Left). Under the same parameter setting, they can be well captured with images shot from a near distance as shown in Figure 8(Right). Our system provides the flexibility for structures in different scales without hand-tuning parameters: geometry structures in larger scale is available by capturing the images from a decent distance; detailed structures in small scale can be then added by shooting images around the target region from a near distance.

Scenes: The datasets mentioned above, including "Dino ring" and "Temple ring", are collected for object reconstruction. Most of them are captured while walking around the target object within a limited space. We collect another dataset to validate the performance of our system on reconstruction of a relatively larger scene. "shrubs" is captured while walking through shrubs with a distance $\sim$10 meters. The reconstruction result is shown in Figure 9.

4.3. Run-time analysis

The run-time statics for each dataset is shown in Table 1. "Dino ring" and "Temple ring" are two datasets with camera poses provided. The run-time of camera pose estimation are not included. In VisualSFM, feature detection, matching and Bundle Adjustment are parallelized on multi-core processor and GPU. The time ranges from seconds to a few minutes for each dataset in our experiments. The computational cost of multi-view stereo algorithm is proportional to the image size. The run-time of multi-view stereo for each image is around 2.5 seconds with an image size of 640×480. For "shrubs" dataset, we use a resolution 1280×720 in order to capture details for a larger scene. The run-time of multi-view stereo for each image is around 9.5 seconds. We use the integration algorithm proposed in [5] without any further optimization over the integrated volume. The time cost for volumetric integration varies from a few seconds to a minute in our experiments, depending on the voxel grid size.

As major algorithms parallelized on multi-core processor and GPU, our system is able to generate accurate dense models in minutes, which makes it very efficient compared with other methods where hours are usually needed.

Dataset statics				Run-time			
Dataset	Images	Image Size	Voxel Grid Size	Pose.	Depth.	Fusion.	Total
Dino ring	48	640 × 480	200 × 200 × 200	-	2m 39s	10s	2m 49s
Temple ring	47	640 × 480	200 × 220 × 200	-	2m 35s	10s	2m 45s
plaster statue	80	640 × 480	200 × 200 × 200	1m 59s	3m 15s	16s	5m 30s
stone	40	640 × 480	200 × 200 × 200	1m 07s	1m 35s	8s	2m 50s
table	55	640 × 480	200 × 200 × 200	58s	2m 12s	11s	3m 21s
iron statue(far)	20	640 × 480	200 × 200 × 200	22s	45s	4s	1m 11s
iron statue(near)	20	640 × 480	200 × 200 × 200	16s	45s	4s	1m 05s
frog(far)	52	640 × 480	200 × 200 × 200	1m 42s	2m 5s	11s	3m 58s
frog(near)	89	640 × 480	200 × 200 × 200	3m 15s	3m 38s	18s	7m 11s
shrubs	40	1280 × 720	650 × 300 × 400	4m 07s	6m 01s	1m 09s	11m 17s

Table 1. Dataset and run-time statics. Pose.: run-time for camera pose estimation. Depth.: time cost on estimation of depth maps using multi-view stereo algorithm. Fusion: cost on volumetric integration.

5. Conclusion and Future work

In this paper, we present a system for fast dense 3D reconstruction using images captured from a hand-held camera. Our system is accurate, efficient and flexible, and able to build accurate dense 3D models in minutes for objects with different scales. Currently, camera pose estimation is achieved using offline SfM algorithm with Bundle Adjustment to ensure accuracy. Our future work is to extend our system with real-time camera tracking utilizing the inertial measurement unit (IMU) within a smartphone. As the computation capacity of mobile processors and GPU grow, we are also exploring porting our system on a mobile platform.

6. Acknowledgments

This work was supported by grant DE-NA0001683 from the U.S. Department of Energy.

References

[1] Middlebury Multi-View Stereo Dataset. http://vision.middlebury.edu/mview/. 5

[2] NVIDIA CUDA. http://www.nvidia.com/cuda. 5

[3] VisualSFM. http://homes.cs.washington.edu/~ccwu/vsfm. 5

[4] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. 3, 5

[5] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 1, 2, 4, 6

[6] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards Internet-scale Multi-view Stereo. In *CVPR*, 2010. 2

[7] Y. Furukawa and J. Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *IJCV*, 84(3):257–268, 2009. 2

[8] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 32(8):1362–1376, 2010. 2, 6

[9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000. 3

[10] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *CVPR*, 2009. 2, 5

[11] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR*, 2007. 2

[12] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Siggraph Computer Graphics*, volume 21, pages 163–169. ACM, 1987. 5

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[14] R. Newcombe, S. Lovegrove, and A. Davison. DTAM: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2, 3, 5

[15] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *CVPR*, 2010. 2

[16] D. Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 26(6):756–770, 2004. 2

[17] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992. 3

[18] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 5

[19] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. *Vision algorithms: theory and practice*, pages 153–177, 2000. 1, 2

[20] B. Ummenhofer and T. Brox. Dense 3d reconstruction with a hand-held camera. In *Pattern Recognition*, pages 103–112. Springer, 2012. 2, 5

[21] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR*, 2011. 3, 5

[22] C. Zach. Fast and high quality fusion of depth maps. In *3DPVT*, 2008. 4, 5

[23] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *ICCV*, 2007. 4, 5

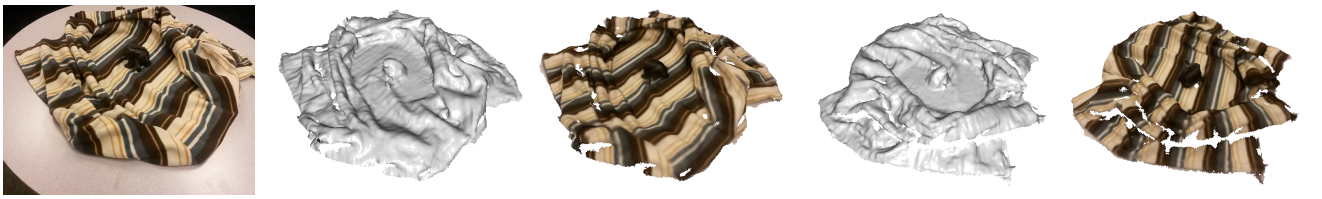
(a) plaster statue

(b) stone

(c) table

(d) iron statue (far)

(e) iron statue (near)



(f) frog (far)

(g) frog (near)

Figure 10. Results on smartphone dataset. From left to right: an example of input image, shaded model (view 1), textured model (view 1), shaded model (view 2), texture model (view 2).