PHOTO- AND VIDEO-BASED 3D RANGING AND MODELING

Yuan-Fang Wang
Department of Computer Science
University of California
Santa Barbara, CA 93106
yfwang@cs.ucsb.edu

Abstract

In this paper, we present our research on photo- and video-based 3D ranging and modeling. We have constructed such a 3D ranging and modeling system, PhotoModel3D, that was made available for free, non-commercial use over the Web. The system has received over a hundred thousands Web visits and thousands of use in the past two years alone. Currently, we demo 900 3D models thus constructed using photos and videos contributed from anonymous users all over the world. Here, we describe the algorithms used in the 3D pipeline and present the results of a comparison study and an accuracy analysis of its performance.

Keywords: photo, video, 3D, model, reconstruction, vision

1 Introduction

In this paper, we present our research on photo- and video-based 3D ranging and modeling. Image-based 3D modeling is widely considered an ill-posed, inverse problem in computer vision that is difficult to solve efficiently, robustly and accurately [7, 11, 21]. Furthermore, photo- and video-based 3D modeling is complicated, as it comprises a pipeline of intertwined components, touching upon many facets of computer vision, e.g., 2D feature analysis and tracking, localized 2D to 3D structure and motion inference, global numerical optimization, 3D surface generation, and multi-view texture mapping. A complete 3D pipeline must successfully address all these problems and more.

Recently, the rapid maturity and wide adoption of some crucial infrastructure and hardware technologies has greatly facilitated such image-based, 3D ranging research. In 2011 alone, over 1.4 billion camera phones and another 100 million digital cameras were sold worldwide. A commodity PC these days comes with a 2-, 4-, or 6-core CPU with gigabytes or terabytes of disk storage for processing and storing image and video data. Wired, wireless, and cellular networks abound that allow easy upload and download of videos and photos. The technological confluence is enabling "rubber-meets-the-road" validation of over 40 years of 3D image-based ranging and modeling research, and help pushing the academic research into the real-world consumer and military markets.

We have developed such a photo- and video-based 3D ranging and modeling software that we call PhotoModel3D [28]. PhotoModel3D employs a photo-based and photo-only analysis paradigm known as either structure from motion (SfM) in the computer-vision and computer-graphics communities [7, 11, 21, 19, 25, 9] or simultaneous localization and mapping (SLAM) in the robotics communityy [5, 13, 22, 17]. Regardless of the nomenclature, the general principles of such a 3D modeling system are to exploit the motion parallax effect exhibited in multiple images taken by a travelling camera to infer the 3D scene structures and the camera poses. PhotoModel3D (1) works with both discrete images and continuous videos taken by a consumer-market digital camera, camcorder, or camera phone of any make and model, (2) uses no special equipment (e.g., lens and tripod), active projection, artificial lighting, prior camera calibration, and man-made markers and registration patterns, (3) requires no user training (just point and shoot), (4) is fully automated and end-to-end (from photographs to fully colored and textured 3D models) without manual intervention or data-specific parameter tuning, (5) is a software-based solution that runs on commodity Linux and

.

Windows servers without the need of special hardware (GPU, DSP, etc.) acceleration, (6) has been demonstrated in an unbiased study to outperform many state-of-the-art 3D modeling pipelines based on a similar SfM principle [29], (7) has been shown to infer 3D models of high fidelity, with an average 3D structure error less than 0.2% measured against ground-truthed 3D LIDAR models, (8) has been deployed on the web allowing free, non-commercial use for more than 2 years; receiving over 100 thousands web visits and thousands of use, and (9) has successfully constructed thousands of 3D models of a large variety of 3D scenes using images and videos contributed from anonymous users all over the world.

The remainder of the paper will describe our 3D processing flow and algorithms used in PhotoModel3D. The discussion will be followed by the presentation of an comparion and an accuracy study. Finally, we will summarize the current status of our research and development in a concluding remark.

2 Technical Description

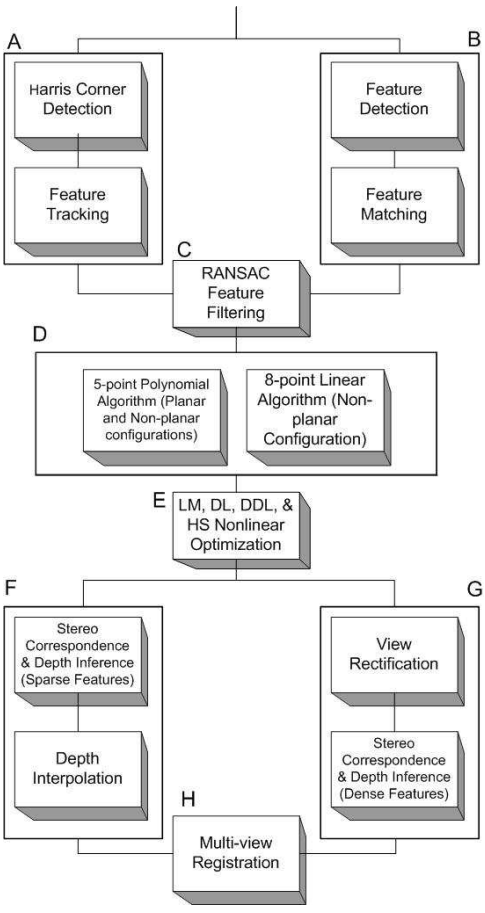


Figure 1: Flowchart of Visualsize's 3D modeling pipeline.

Here we present the architecture of our 3D pipeline using the flow chart depicted in Fig. 1. Different modules in Fig. 1 and their functions are described in more detail below.

◇ *Feature selection, tracking, and matching* (Boxes A and B): This step identifies prominent and semi-invariant features and matches these features across multiple images to establish their correspondences. We use two complementary paradigms: continuous tracking for videos (Box A) and discrete matching for images (Box B).

When continuous videos are captured at a high video frame rate, there is often trifling change in the appearance and position of image features in adjacent frames. We therefore detect prominent features (using the Harris corner detector [10]) and track their locations in images through a localized search operation. While a large number of trackers are available, we have opted to use our FFT-based tracker [14] that is accurate and achieves real-time performance for reasonably complex scenes.

If only a few isolated snapshots of the scene are acquired, changes in a feature's pose and appearance in these snapshots can be significant to render tracking infeasible due to large changes in perspective and illumination, and a limited search-window size. Instead, we compute advanced features (similar to SIFT [15] or SURF [1]) which are insensitive to scale, rotation, and color changes in images. We match these features in two images, regardless of their locations, to establish feature correspondences.

◇ *Robust camera motion inference* (Boxes C, D, and E): This step uses matched image features in two views to infer the camera's movement in between the views. The core process is either a 5-point polynomial algorithm or an 8-point linear algorithm [18, 7, 12, 11]. The 5-point algorithm handles both planar and non-planar 3D configurations, and hence, is more general than the 8-point counterpart that fails if the 3D scene is planar. However, 8-point runs faster than 5-point.

The names of the inference algorithms refer to the minimum numbers of pairs of matched image features in two views that are needed for deducing the camera's motion parameters. In reality, we track and match significantly more features than just five or eight. Furthermore, tracking/matching results are necessarily imprecise due to noise and

.

image quantization. Catastrophic failure in tracking (loss of trajectory) and matching (erroneous pairing assignments) do happen occasionally. To improve the robustness in camera motion inference, we use a nonlinear selection and filtering strategy called RANSAC [8] to better condition the feature matching process.

Finally, nonlinear optimization (Box E) is used to give a final "polish" to the best result from RANSAC and the 5-point or 8-point linear algorithms. We have used the Levenberg-Marquart (LM) [11], Dog Leg (DL), Double Dog Leg (DDL), and Hook-Step (HS) algorithms [2, 6], which are different variances of procedures for optimally combining the Gauss-Newton method and the gradient-descent method. While LM, DL, DDL, and HS are nonlinear iterative optimization procedures, convergence is fast because of a good initial guess has been obtained (Boxes C and D).

◇ *Stereo rectification, matching, and depth inference* (Boxes F and G): This step is to infer 3D surface depth and construct a 3D model that captures both structure and appearance information. We consider two different approaches. In one approach (Box F), only the depths of the tracked/matched image features (Boxes A and B) are explicitly computed to form a sparse depth map. Depths of the intermediate pixels are estimated through bi-linear interpolation from those of the tracked/matched features. This approach is computationally efficient and works well if the scene structure is smooth.

A more accurate 3D model can be constructed by computing pixel disparity and inferring 3D depth at *each and every* pixel in the images (Box G). To efficiently and reliably perform stereo analysis, the image pair should be in a standard side-by-side configuration. If not, we must either identify the corresponding epipolar lines in the two images, or rectify the two images to rearrange the image pixels in such a way that the corresponding pixels in the two images lie on the same image scan lines [20]. We then apply a stereo matching algorithm based on dynamic programming [3], which takes into consideration pixel-, neighborhood-, and globally-based similarity criteria in matching.

◇ *Multi-view registration* (Box H): The final step is for registering partial 3D models constructed from multiple 2-view analyses into a more complete 3D model. We treat each partial 2-view model as a cloud of 3D points, and these point clouds are related by rigid-body transforms in space. We solve the registration problem by finding the rigid-body registration parameters to match 3D point clouds with one another using least-square.

Fig. 2 shows sample 3D models. Note that depending on the requirement of a particular application, we can generate both texture-mapped models and point-cloud models, though only texture-mapped models are shown. A lot more results, currently comprising 900 3D models of all kinds of objects: human faces and others, soft and hard objects, smooth and rough appearance, large and small targets, nature and man-made scenes, complete (360^o all-around) and partial 3D models, indoor and outdoor settings, ground and airborne photos, short (as few as 5 images) and long (as many as 800 images) sequences, are available at our website http://www.visualsize.com.

3 Experimental Results

We present two studies here: one is a performance comparison of five 3D modeling systems based on the same SfM principle, and the other is an accuray study to learn how faithful our computer models can be to the ground-truthed LiDAR models of the same 3D scenes.

3.1 Performance Comparison

We present here a comparison study of five 3D modeling systems based on the SfM principles (Bundler [25], Bundler + PMVS2 [9], Project Photofly from Autodesk, ARC 3D Web Service [27], and our own [29]). The usage scenario we try to emulate in this study is that of a commercial 3D modeling system that accepts 3D modeling requests from clients (cell phones, tablets, PCs, etc.) over the Web, executes the 3D modeling pipeline on a back-end server, and returns the 3D model as a result. The users (1) are not computer vision experts and cannot provide additional information other than the photos themselves, (2) are not willing to go through lengthy training, or purchase expensive cameras or specialized photography equipment for building 3D models, (3) may be cost conscientious especially when connecting to the back-end server through a mobile device where the user may have to pay for the bandwidth usage (and hence, no uploading large photos that tie up Web links for a long time), and (4) are accustomed to the "instant gratification"

.

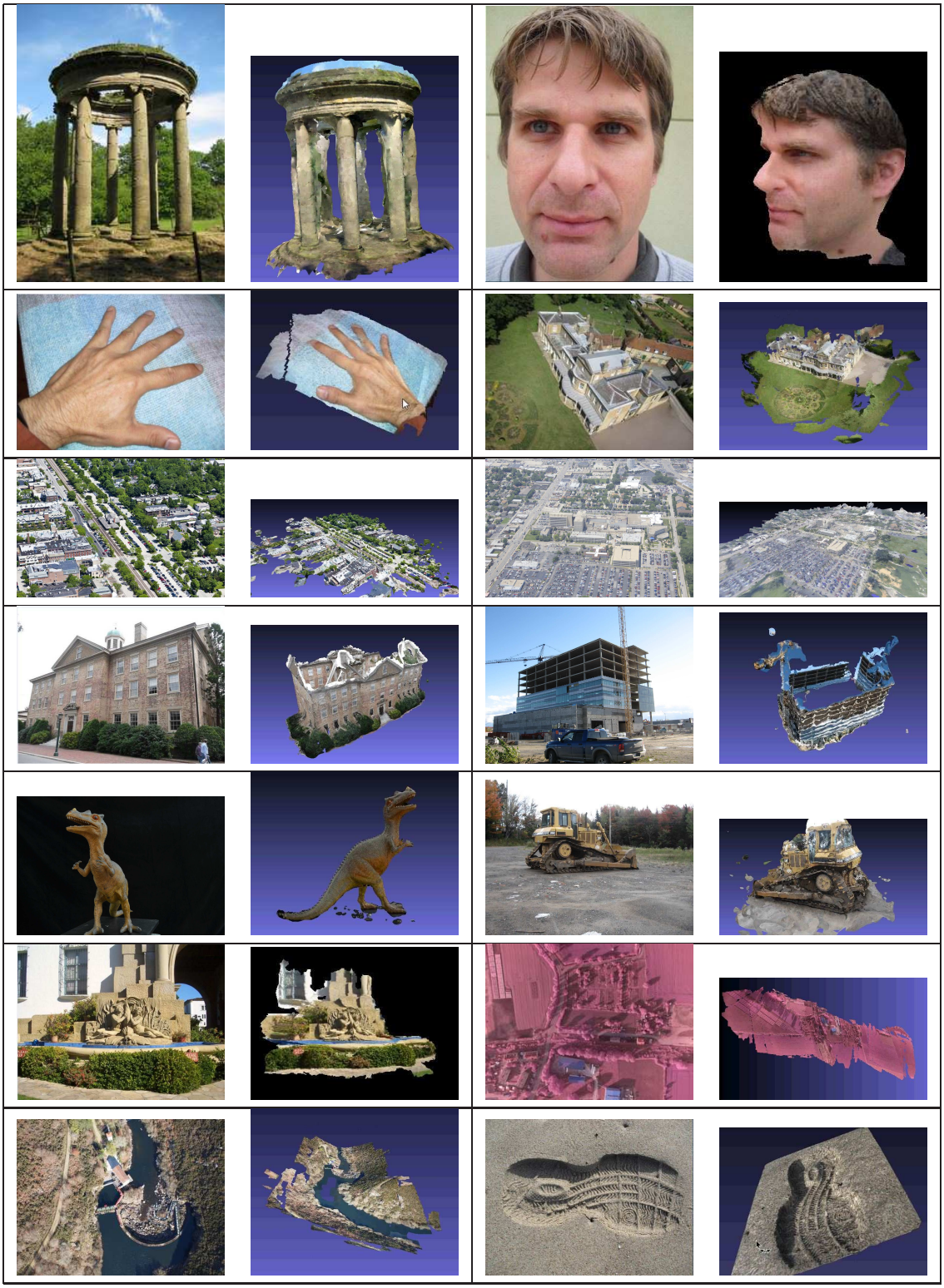


Figure 2: Sample results of our 3D modeling system. Each data set is represented by one input image (left) and one image of the 3D model (right). Short movies of the 3D models of 900 test data sets are available for viewing on the Web at: http://www.visualsize.com/3ddemo/index.php.

.

Web experience, and hence, are impatient to get the results back.

While similar performance comparison has been attempted before [23, 24, 26], our study stands out by performing "rubber-meets-the-road" validation tests that closely mimic what a commercial 3D modeling system needs to accomplish in the real world. The novelty of our comparison study is three-fold:

(1) The comparison was performed by exercising the full 3D modeling pipelines, from input images all the way to 3D models, instead of testing some isolated components in a 3D pipeline [24],

(2) In addition to the ground-truthed 3D data provided by [26], we have used over 100 data sets (122 to be exact), with over three thousand images, representing a variety of 3D scenes, collected from a large number of consumer-market digital cameras and camera phones of many makes/models, and contributed by anonymous users all over the world. Furthermore, these images were shot without prior camera calibration, use of special equipment (tripod, lens, etc.) and lighting (laser and structured light projection), and user training in image acquisition. In contrast, [23, 24, 26] have used small, calibrated data sets, and

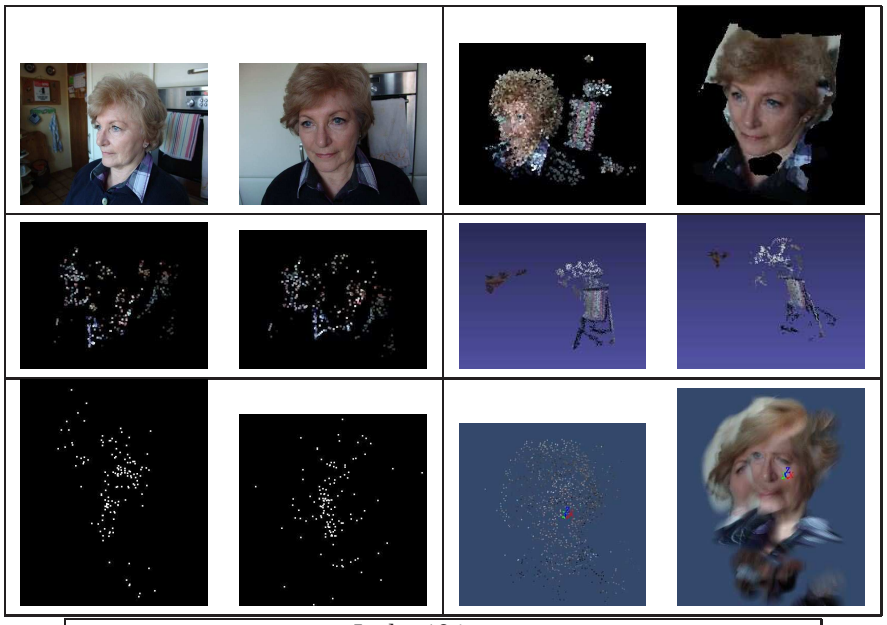
(3) To ensure that the comparison is fair and the results do not depend on the details of implementation, we have included only those 3D modeling systems that are available for use on the Web or locally in a binary format; comprise a complete, fully-automated 3D pipeline that leads from input images to 3D models—without any user intervention and without data-dependent parameter tuning; and are able to perform the feats using images of a reasonable size. Furthermore, a diligent Web search has unearthed no other 3D modeling system that fits the comparison requirements, and hence, our selection is believed to be comprehensive and provides a holistic view of the state of the art.

The test platform was a PC with a 2.8Ghz Intel Core 2 Duo CPU, 4G RAM, running Windows 7. The experimental procedures were extremely straightforward: As all these 3D modeling systems need were input images, all we did was to provide them with the input images and then waited for the computation to finish. We felt that the simplicity of the procedures better ensured fairness. We (1) have used the binary releases of these programs so we could not have compiled them incorrectly, (2) we have used the default execution scripts supplied with the releases without modification so we could not have tuned the parameters wrongly, and (3) we have run these programs on the same machine using the same number of CPU cores, the same amount of memory, and under similar runtime conditions. For Project Photofly and ARC 3D Web Service, the images were uploaded to their servers on the Web and there was no end-user tunable parameters on their GUI.

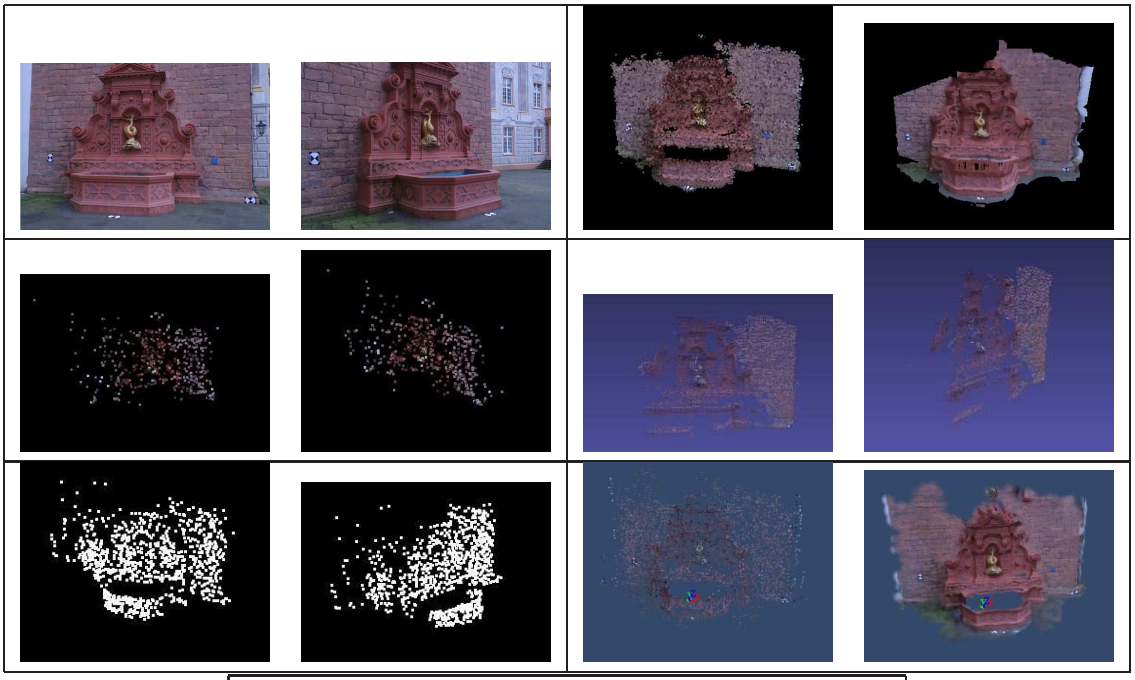
Fig. 3 presents three such results of running the five modeling systems on ten sample data sets. The two images at top left are sample input images, the two at top right are our results, the two at middle left are results of Bundler, the two at middle right are results of Bundler + PMVS2, the two at bottom left are results of ARC 3D Web Service, and the two at bottom right are results of Autodesk's Project Photofly. The table below the graphic results shows the name and size of the data set, how many pictures were processed, how many 3D points were generated, and the runtime for Bundler, Bundler+PMVS2, ARC 3D Web Service, and our system. Runtime of ARC 3D and Photofly was not included as data sets were processed on their cloud servers. Photofly used a proprietary point-cloud format so the cloud density information was not available either.

Results of Bundler, Bundler+PMVS2, and ARC 3D Web Service are presented in discrete point-cloud format as these programs did not generate 3D texture-mapped models. We supplement texture-mapped results for Project Photofly and our system if one point-cloud picture is enough to illustrate the density and quality of such a discrete structure. As page limit does not allow us to show all these examples, and the quality and accuracy of a 3D model is best evaluated by viewing the model in 3D—instead of just a few screen shots, we strongly urge interested readers to browse our Web site for more information [29]. In terms of cloud density and quality, and the chance of success, the test data indicated that ours outperformed Project Photofly, which outperformed Bundler+PMVS2, which outperformed Bundler, and which outperformed ARC 3D. This observation also confirms our experience with the much larger, over 100 data set ensemble.

.



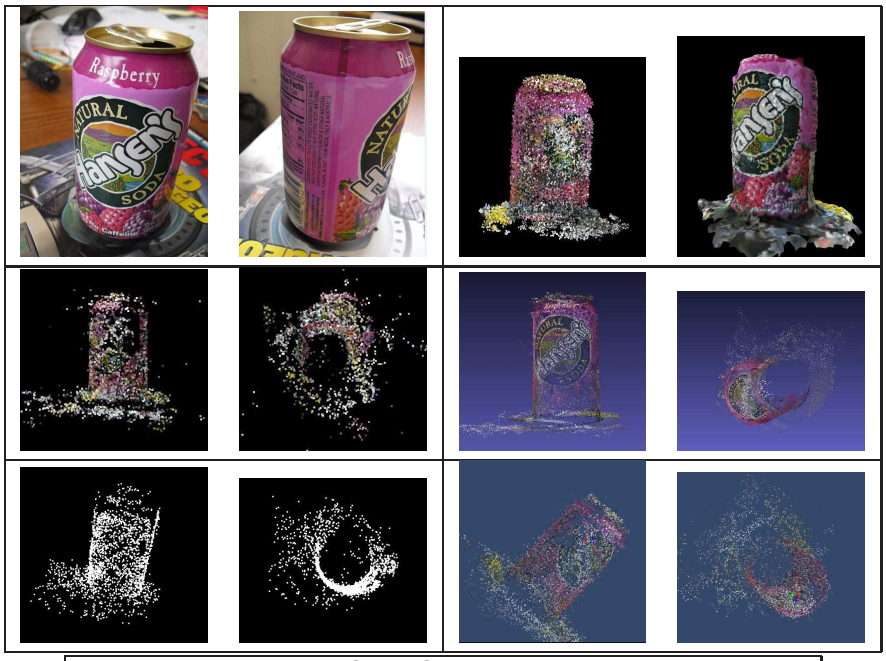
Lady: 10 images				
	Bundler	Bundler + PMVS2	ARC 3D	Ours
Pic processed	4	4	6	10
# 3D points	467	2,428	148	10,810
Run time	0:39	0:58	-	2:00



Fountain: 11 images				
	Bundler	Bundler + PMVS2	ARC 3D	Ours
Pic processed	5	5	11	11
# 3D points	604	7,221	1,025	132,670
Run time	0:55	1:30	-	6:05

Figure 3: The two images at top left are sample input images, the two at top right are our results, the two at middle left are results of Bundler, the two at middle right are results of Bundler + PMVS2, the two at bottom left are results of ARC 3D Web Service, and the two at bottom right are results of Autodesk's Project Photofly. For each data set, the table below the graphic results shows the name and size of the data set, how many pictures are processed, how many 3D points are generated, and the runtime for Bundler, Bundler+PMVS2, ARC 3D Web Service, and our system.

.



Soda Bottle: 46 images				
	Bundler	Bundler + PMVS2	ARC 3D	Ours
Pic processed	18	18	35	46
# 3D points	6,214	27,221	2,473	75,275
Run time	5:48	11:07	-	11:18

Figure 3 continued

3.2 Accuracy Study

Here, we attempt to answer the question that is likely to be in a reader's mind: How accurate is our 3D modeling system? Accuracy analysis requires comparison with the ground truth. However, as mentioned before, our test datasets were collected from a large number of consumer-market cameras and phones, and no ground-truth 3D profiles were available. The Dino and Temple data sets used in [24] were gathered using the Stanford Spherical Gantry, which provided the ground truth in the camera poses, but not in the 3D structures. Middlebury Stereo Datasets [4, 23] comprise only short sequences (up to 7 images) using a fixed linear camera translation, and hence, are not that interesting to us. To our best knowledge, [26] provides the only publicly available 3D data sets with ground-truthed 3D profiles that used a general camera motion, and were specifically generated to validate 3D modeling algorithms using the SfM principle. Ground-truth 3D profiles (gathered using a LIDAR system) for two data sets, Fountain-P11 and Herz-Jesu-P25, were used in our accuracy analysis. These data sets are available for download at: http://cvlabwww.epfl.ch/data/multiview, and were used in the following academic paper [26].

We tried to emulate—as faithfully as possible—what a commercial 3D modeling system needs to accomplish for a client submitting 3D modeling tasks through a Web-service model. To this end:

• We have used two different spatial resolutions: the VGA size (640×480) and a higher 2150×1434 resolution, for upload and processing. This is for testing the ability of the system for handling both low-res and high-res imagery.

• We did not use any camera calibration data generated externally (i.e., we did not use the intrinsic or the extrinsic camera parameters supplied with the images). Again, in the real-world application scenarios, such intrinsic and extrinsic camera parameters are not available. Most, if not all, consumer-market digital cameras and phones are not calibrated. A 3D modeling pipeline must be able to automatically calibrate the intrinsic and extrinsic camera parameters using nothing but the input images, without any outside assistance.

.

Table 1: Accuracy Evaluation results

Data set	Herz-Jesu-1	Fountain-1	Herz-Jesu-2	Fountain-2
# of images	25	11	25	11
spatial resolution	640 × 480	640 × 480	2150 ×1434	2150 ×1434
Runtime	11min 01sec	4min 30sec	59min 48sec	7min 12sec
# of 3D points	342,949	140,785	1,437,984	1,072,139
# of faces	735,419	315,195	2,781,106	2,107,543
max error	4.74%	5.07%	4.42%	1.92%
median error	0.62%	0.62%	0.44%	0.23%
average error	0.41%	0.44%	0.26%	0.17%

• We have concentrated on the "end results" and ignored the "by products" of such 3D processing. That is, our comparison is on the faithfulness of the 3D models, not on the accuracy of the recovered intrinsic and extrinsic camera parameters. We believe that in the consumer market, the end users are mainly interested in the 3D models. Furthermore, we believe that the situation where a modeling system estimated erroneous camera parameters but somehow still obtains correct 3D structures fortuitously is extremely unlikely (we have never observed such a phenomenon).

After our 3D pipeline finished generating 3D models from the input VGA images, these models were aligned with the ground-truth models in a two-stage procedure: We first used the manual alignment process provided by Meshlab [16] to roughly align our 3D models with the ground-truth models. The alignment process consisted of loading both models into Meshlab, manually specifying a small number of corresponding points in the two models to establish a rough alignment, and then allowing Meshlab to refine the initial manual alignment using an iterative ICP algorithm.

After the models were roughly aligned as in the previous step, we loaded both models into our own display program that allowed small x, y, z rotations and translations applied to the ground-truth models. We applied such small translations and rotations interactively and eye-balled the display for the best qualitative alignment results.

After models had been aligned, we computed an absolute error measure for each and every 3D point in our 3D models. This error was the minimum distance from a 3D point in our models to the closest points in the corresponding ground-truth models. We then computed a percentage error by dividing the absolute error distance by the largest dimension of the ground-truth models in the x, y, or z direction.

Our modeling pipeline ran on a Windows 7 desktop with an Intel Core i-3 3.3GHz processor, 6GB of memory, and 1TB of disk space. For each data set, we used two different spatial resolutions: VGA and 2150×1434. The runtime, density and accuracy statistics are summarized in Table 1 and the 3D models are depicted graphically in Figs. 4 to 5. One can also download or view these models in 3D at http://www.visualsize.com/3ddemo/comparison/accuracy.html. As can be seen in Table 1, our modeling algorithm was able to construct 3D models which confirmed with the ground-truth models very well. The average error was less than 0.5%—and this error was computed over hundreds of thousands of recovered 3D points. Increasing spatial resolution improved the accuracy only marginally, but can lengthen the computation time significantly in the case of Herz-Jesu.

4 Concluding Remarks

There is obvious trade-off of using photos and videos for 3D ranging and modeling. The impact of a system like PhotoModel3D on the society is that the system enables anyone and everyone with a digital camera, camcorder, and phone (over three billions such devices are in circulation worldwide today) to become a 3D content producer without any training in science and engineering. PhotoModel3D system has been deployed for about 2 years and has received more than 100,000 Web visits and thousands of uses since 2011. However, it is theoretically impossible to determine the absolute distance and scale using photos or videos alone. So some dimension (distance and size) measurements must be known of certain scene entities to make absolute ranging possible.

.

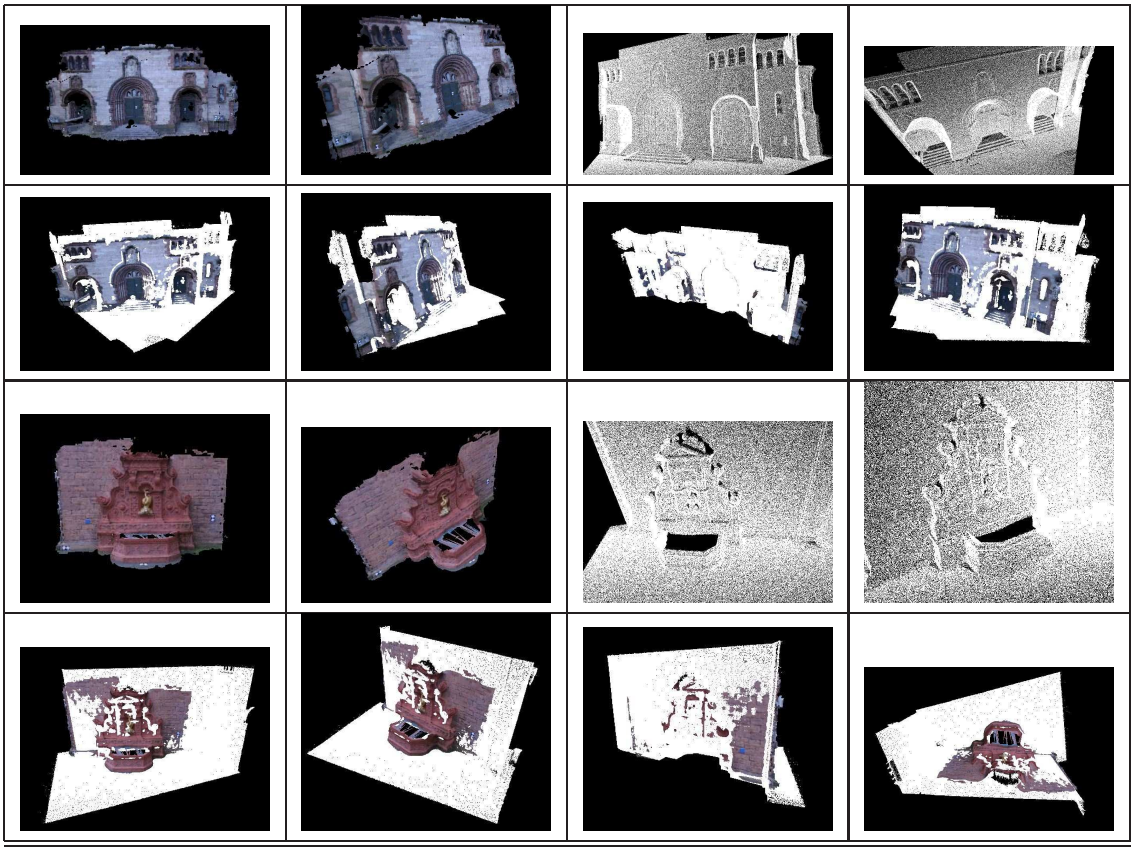


Figure 4: Herz-Jesu-P25 (top) and Fountain-P11 (bottom) processed at the VGA resolution. For each data set, left two on the 1st row: our results. Right two on the 1st row: ground truth. The 2nd row displays both our model and the ground truth registered in a common reference frame.

References

[1] H. Bay, T. Tuytelarrs, and L. V. Gool. SURF: Speeded Up Robust Features. In *Proc. European Conference Computer Vision*, pages 404–417, 2006.

[2] A. Conn, N. Gould, and P. Toint. *Trust Region Methods, MPS-SIAM Series On Optimization*. SIAM, Philadelphia, PA, 2000.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. MIT Press, Cambridge, MA, 1990.

[4] D. Scharstein and R. Szeliski. http://vision.middlebury.edu/stereo/, 2002.

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera Slam. *IEEE Trans. Pattern Analy. Machine Intell.*, 29, 2007.

[6] J. Dennis and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA, 1996.

[7] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, MA, 1993.

[8] M. Fischler and R. Bolles. RANdom Sample Consensus: A Paradigm for Modeling Fitting with Application to Image Analysis and Autoamted Cartography. *Communications of ACM*, 24:381–395, 1981.

[9] Y. Furukawa and J. Ponce. http://grail.cs.washington.edu/software/pmvs/, 2010.

[10] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Fouth Alvey Vision Conference*, pages 147–151, 1988.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Camridge, MA, 2003.

[12] R. I. Hartley. In Defense of the Eight-Point Algorithm. *IEEE Trans. Pattern Analy. Machine Intell.*, 19, 1997.

[13] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007.

[14] D. Koppel, Y. F. Wang, and H. Lee. Image-Based Rendering and Modeling in Video-Endoscopy. In *Proc. of IEEE Int. Symp. on Biomedical Imaging*, pages 272–279, Arlington, VA, April 2004.

[15] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60:91–100, 2004.

.

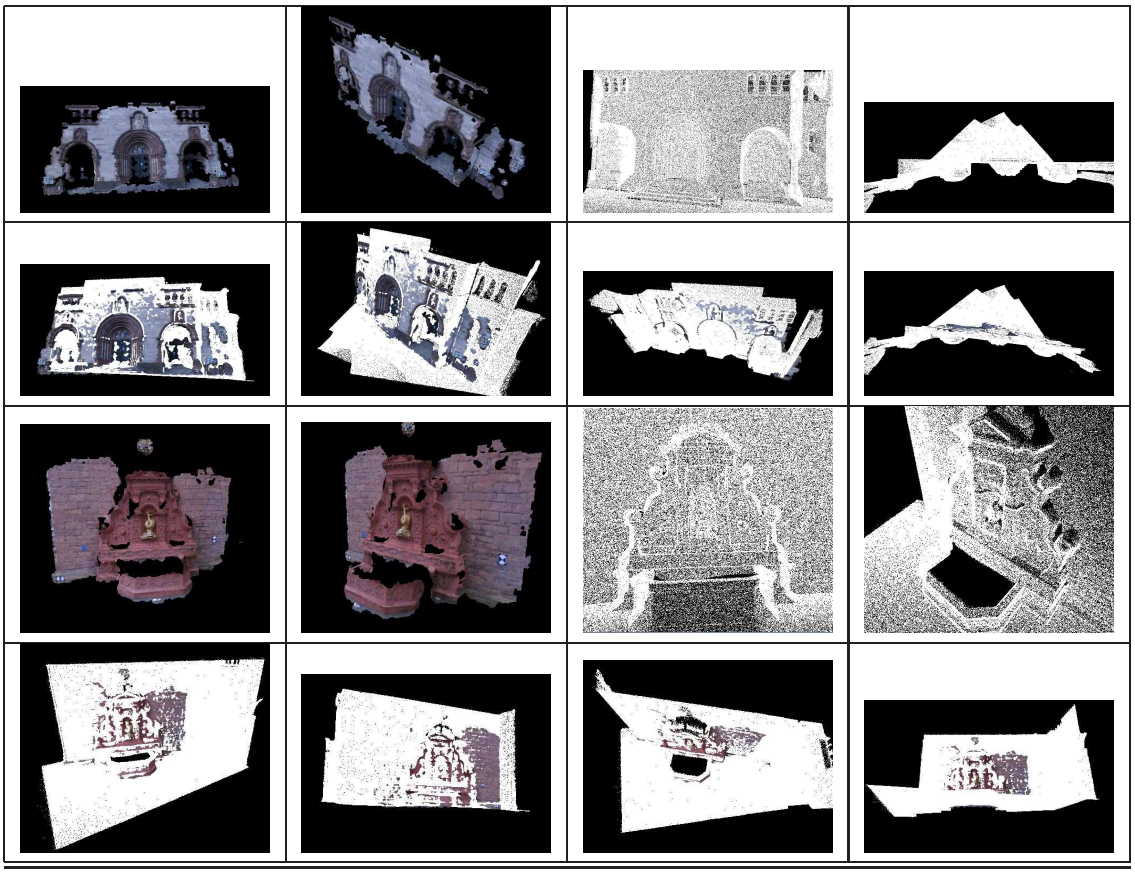


Figure 5: Herz-Jesu-P25 (top) and Fountain-P11 (bottom) processed at the 2150 $\times$1434 resolution. For each data set, left two on the 1st row: our results. Right two on the 1st row: ground truth. The 2nd row displays both our model and the ground truth registered in a common reference frame.

[16] Meshlab. http://meshlab.sourceforge.net/, 2010.

[17] R. A. NewCombe and A. J. Davison. Live Dense Reconstruction with a Single Moving Camera. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, 2010.

[18] D. Nister. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Analy. Machine Intell.*, pages 756–770, 2004.

[19] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual Modeling with a Hand-held Camera. *Int. J. Comput. Vision*, 59:207–232, 2004.

[20] M. Pollefeys, R. Koch, and L. V. Gool. A Simple and Efficient Rectification Method for General Motion. In *Proc. Int. Conf. Comput. Vision*, 1999.

[21] L. Quan. *Image-Based Modeling*. Springer, 2010.

[22] S. J. L. R. A. NewCombe and A. J. Davison. DTAM: Dense Tracking and Mapping in Real Time. In *Proc. Int. Conf. Comput. Vision*, Barcelona, Spain, Nov. 2011.

[23] D. Scharstein and R. Seliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Int. J. Comput. Vision*, 47:7–42, 2002.

[24] S. Seitz, B. Curless, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, 2006.

[25] N. Snavely, S. Seitz, and R.Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *ACM SIGGRAPH Conf. Proc.*, pages 332–338, 2006.

[26] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, 2008.

[27] M. Vergauwen and L. V. Gool. Web-based 3D Reconstruction Services. *Machine Vision and Applications*, 17:411–426, 2006.

[28] VisualSize Inc. http://wwww.visualsize.com, 2007-.

[29] Y. F. Wang. A Comparison Study of Five 3D Modeling Systems Based on the SfM Principles, Visualsize Inc., 2011.