Creating Textured 3D Models from Image Collections using Open Source Software

Ali Alouache
School of Automation
Beijing Institute of Technology
Haidian District 100081,
Beijing
People's Republic of China

Xiaolan Yao
School of Automation
Beijing Institute of Technology
Haidian District 100081,
Beijing
People's Republic of China

Qinghe Wu
School of Automation
Beijing Institute of Technology
Haidian District 100081,
Beijing
People's Republic of China

ABSTRACT

The creation of digital three dimensional (3D) spatial models for the real scenes using image collections is one of the interesting applications of computer vision. In this paper we present some experimental results for creating textured 3D models from image collections using open source software packages (i.e. VisualSFM, CMVS, SURE, MeshLab, Cloud Compare). The images can be picked up with different types of cameras, or using the different imaging systems like UAVs and Satellites. At first, we use VisualSFM which is a robust Structure from Motion software estimating the calibration parameters of all the images, and a sparse 3D point cloud. We present two alternative softwares for multi view dense stereo reconstruction (CMVS and SURE). CMVS and SURE are effective tools and can operate on the common desktop PCs. The obtained results with CMVS and SURE are visualized with Meshlab and Cloud Compare respectively. Again we used MeshLab for mesh generation and texture mapping based on Poisson's filter for surface reconstruction and texturing tools available in Meshlab.

General Terms

Image processing, Computer Vision, 3D World Modelling From Image Collections.

Keywords

3D Reconstruction, Structure From Motion, Multiple View Dense Stereo Reconstruction Mesh Generation, Texture Mapping, VisualSFM, CMVS, SURE, MeshLab, Cloud Compare.

1. INTRODUCTION

The 3D spatial information of real scenes can be obtained using several expensive technological devices like laser scanner or LIDAR [1], [2]. However sometimes the use of active sensors is not possible for example 3D Reconstruction of very large terrains and towns [3], [4], therefore the cameras can be used as alternative sensors. The cameras are advantageous because they are chip devices, easy to use for capturing images and offer extra information like color and texture. Hence the creation of digital 3D models for the real world using images has attired a lot the intention of researchers in the last years [5], [6]. On the other hand, the image processing task requires much complex algorithms and programming tools.

In the past years, the 3D information was mainly needed for robotics applications like mobile robot autonomous navigation and guidance [7] [8], and architectural modelling [9], [10], but today this is shifted to other domains like: gaming, movies, computer graphics, augmented reality, industrial and medical applications [11], [12], [13]. The cameras today are everywhere and everyone's hand. For instance, people want to keep realistic 3D models in their digital devices for many things in the daily life and use phones for accurate navigation as shown in [14]. In practice there exist several tools for creating digital 3D models from a collection of images like commercial software, web services and open source implementations [15], [16]. In this paper we present some experimental results for creating textured 3D models from image collections using open source software packages (i.e. VisualSFM, CMVS, SURE, MeshLab, Cloud Compare). The images can be picked up with different cameras, using the imaging systems such as UAVs and Satellites.

Our aim is to create realistic textured 3D models from a collection of uncalibrated images. At first, we describe the full pipeline for creating digital 3D models from image collections. The pipeline does not work automatically, but it just shows the steps we followed in our experiments for creating textured 3D models using image collections.

VisualSFM is a robust Structure from Motion software used to estimate the calibration parameters of all the images and generate a sparse 3D point cloud. The output of the VisualSFM is saved as (.PLY) for the sparse 3D point cloud and (.txt) for the parameters of the cameras.

Multi View Dense Stereo Reconstruction needs as input the images with the calibration parameters. We use two alternative software packages for multiple view dense stereo which are CMVS and SURE. The dense models generated with CMVS and SURE are visualized with Meshlab and Cloud Compare respectively. The software CMVS and SURE are effective tools which can operate fast and on the common desktop PCs.

Again we used MeshLab for mesh generation, coloring and texture mapping, hence a textured 3D model is generated from the dense point cloud. The results could be exported in different 3D files formats depending on the requirements of the user.

An overview of the organization of the rest of this paper is given as follows: in section 2 we present the full pipeline for creating textured 3D models from image collections and the used software packages. In section 3 we demonstrate the obtained experimental results. Finally Some Conclusions concerning the work are given in section 4.

2. THE FULL PIPELINE FOR 3D MODEL CREATION FROM IMAGE COLLECTIONS

Fig 1 illustrates the full pipeline for creating textured 3D models from a collection of multiple images. A set of uncalibrated images are inserted as input to Structure from Motion SfM (sometimes called Structure and Motion (SaM) estimation.

As indicated in fig.1, the outputs of the SfM are the cameras' matrices file (.txt) and a sparse 3D point cloud file (.PLY). Once a sparse 3D structure and cameras matrices are recovered by SfM, then Multiple Dense View Stereo (MVDS) reconstruction is proceed to build dense 3D model.

MVDS utilizes the images, the cameras' matrices for generating a dense 3D model. The dense 3D point cloud and the images are used for mesh generation and texture mapping for creating a digital textured 3D models in different formats. The sparse 3D point cloud is useful too, where it can be used with the images for generating textured models but with less accuracy and clearness compared to the dense 3D model.

2.1 VisualSFM

VisualSFM is an open source software GUI for Structure from Motion estimation using a collection of uncalibrated images. VisualSfM integrates and improves the following algorithms: an incremental SFM system, SIFT on the GPU, and the Multicore Bundle Adjustment.

In addition to the sparse reconstruction, VisualSFM provides an interface to run PMVS/CMVS tool that is used for 3D dense reconstruction. VisualSFM needs other packages: Gtk, SiftGPU, CUDA, GLEW, GLUT, DevIL.

Fig. 2 shows the VisualSFM interface and the numbered steps as following

1. Adding some images from the computer.
2. SIFT features detection and matching.
3. 3D sparse reconstruction,
4. Dense 3D reconstruction based on the software CMVS.

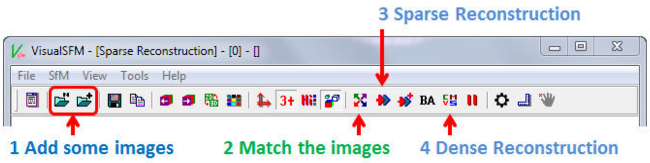


Fig .2 The VisualSFM interface of 3D Reconstruction

2.2 CMVS

CMVS is a development of the software PMVS. It is a multi-view stereo software that takes a set of images and camera parameters as input and reconstructs the 3D structure of the scene as a dense points cloud. CMVS decomposes the images into a set of image clusters of small size, allowing a MVS software to be used to process each cluster independently and in parallel, ensuring that the union of reconstructions from all the clusters should not miss any details that can be otherwise obtained from the whole image set. CMVS should be used in conjunction with an SFM software (Bundler).

2.3 SURE

SURE is available for free for non-commercial use at: http://www.ifp.unistuttgart.de/publications/software/sure/index.html. The dense image matching software SURE has been developed by the Institute for Photogrammetry at the University of Stuttgart. It uses a multi-view stereo (MVS) approach, where first stereo pairs are matched against each other. This stereo matching step is based on the library libTSGM, which implements a modified version of the Semi Global Matching (SGM) algorithm i.e. enabling the determination of 3D information for almost each pixel. The modification uses a hierarchical approach, which enables the processing of complex scenes with large depth variations with short processing time and low memory consumption. Within a second step, the results of image matching are fused by triangulating rays for multiple stereo models at once. This improves the precision of object points, but also enables the rejection of outliers as well as the determination of quality values for each 3D point. The whole implementation is parallelized and optimized for scalability. Thus, large datasets regarding image size and image count can be processed on common desktop PCs.

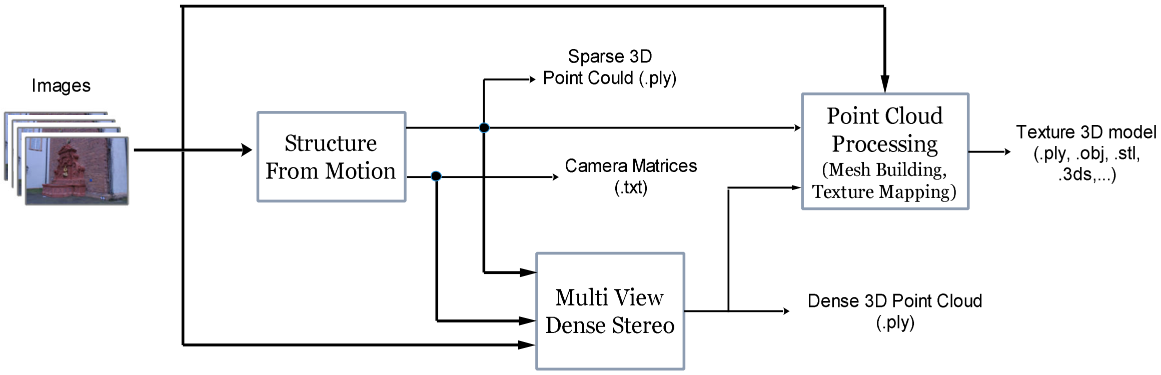

Images
Structure From Motion
Sparse 3D Point Could (.ply)
Camera Matrices (.txt)
Point Cloud Processing (Mesh Building, Texture Mapping)
Texture 3D model (.ply, .obj, .stl, .3ds,...)
Multi View Dense Stereo
Dense 3D Point Cloud (.ply)

Fig .1 Pipeline for 3D model creation from multiple images

2.4 MeshLab

MeshLab is an open source software widely used for 3D point cloud processing such in our case: selecting, smoothing and coloring meshes, surface reconstruction and texture mapping .etc. MeshLab supports different input/output files formats as follows

Import: PLY, STL, OFF, OBJ, 3DS, COLLADA, PTX, V3D, PTS, APTS, XYZ, GTS, TRI, ASC, X3D, X3DV, VRML, ALN.

Export: PLY, STL, OFF, OBJ, 3DS, COLLADA, VRML, DXF, GTS, U3D, X3D.

2.5 Cloud Compare

Cloud Compare is a 3D point cloud processing software which provides a set of basic tools for manually editing and rendering 3D points clouds and triangular meshes. It also offers various advanced processing algorithms, among which methods for performing: registration, segmentation, geometric features estimation .etc.

3. EXPERIMENTAL RESULTS

This section demonstrates the obtained experimental results using a collection of images for a fountain as shown in figure 3.

As indicated in figure 3, the 11 images are labelled as follows Im0, Im1,…, Im10.

The fountain datasets can be downloaded directly from the following website which contains several datasets (http://cvlabwww.epfl.ch/data/multiview/fountain_dense.html).

3.1 Structure from Motion Estimation

VisualSFM employs SIFT Features detector for detecting the interest features from the images.

Table 1 shows the obtained results with all the images, note that all the images have the same size.

The images shown in fig. 3 are textured and detected a large number of SIFT features.

The VisualSFM software extracts the EXIF data from all the images, and the estimated focal length for each camera is given in Table 1.

After detecting the SIFT features in all the images, features matching is done among all the possible pair of images. The results of features matching between all the pairs of images are summarized in table 2, note that the numbers of matched features in each row of table 2 is decreasing until images that don't have much similar parts due to the large change of the cameras' positions.

Once features matching is accomplished, a good pair of images is automatically selected for initializing the SfM process (here the initial pair of images are Im3 and Im5 is selected automatically for initializing the process of the 3D reconstruction), therefore the parameters of the cameras are calculated from the essential matrix, and the 3D points are estimated by triangulation using the matched points (see Table 2).

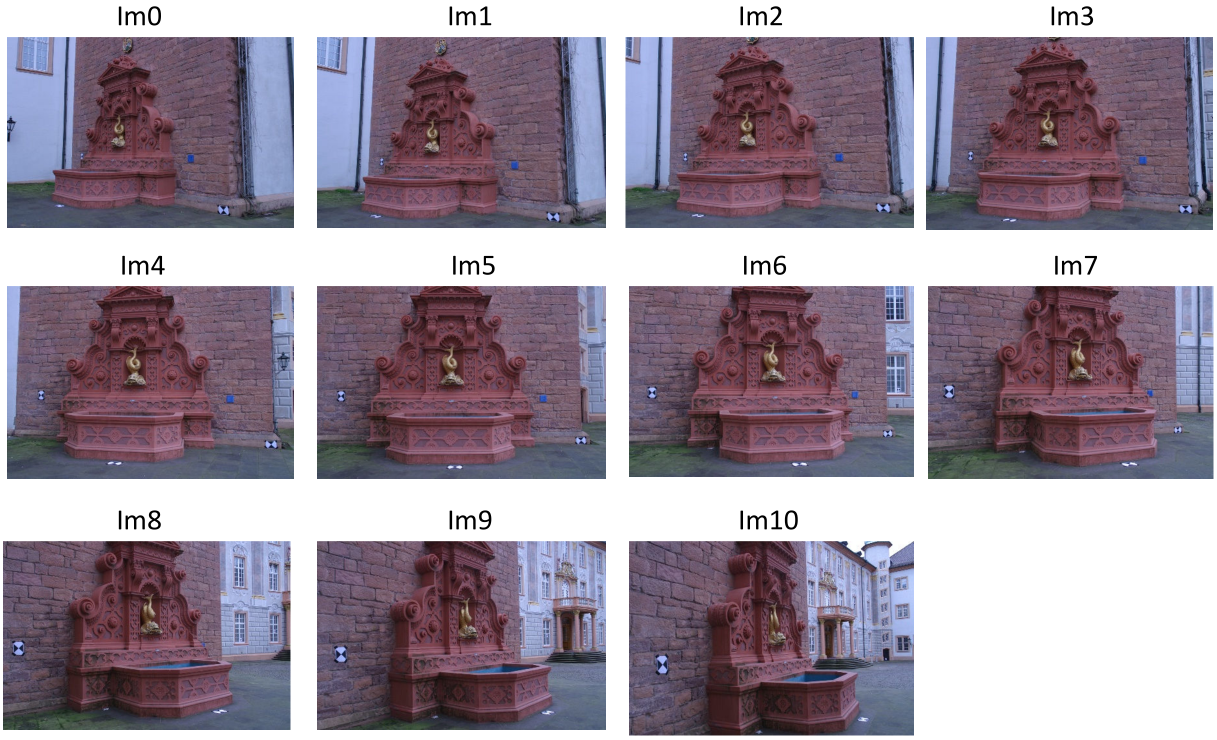


Fig .1 The collection of 11 images of a fountain

Table 1. The results of features detection and focal length estimation

Images	Size (pixels)	SIFT Features	focal length
Im0	3072 X 2048	12212	2754.03564453
Im1	3072 X 2048	8276	2753.55664063
Im2	3072 X 2048	8279	2756.69116211
Im3	3072 X 2048	9039	2760.10205078

Im4	3072 X 2048	8822	2762.359375
Im5	3072 X 2048	9127	2767.328125
Im6	3072 X 2048	9398	2772.5703125
Im7	3072 X 2048	9648	2773.20068359
Im8	3072 X 2048	9961	2777.75268555
Im9	3072 X 2048	10645	2779.01147461
Im10	3072 X 2048	10657	2753.64208984

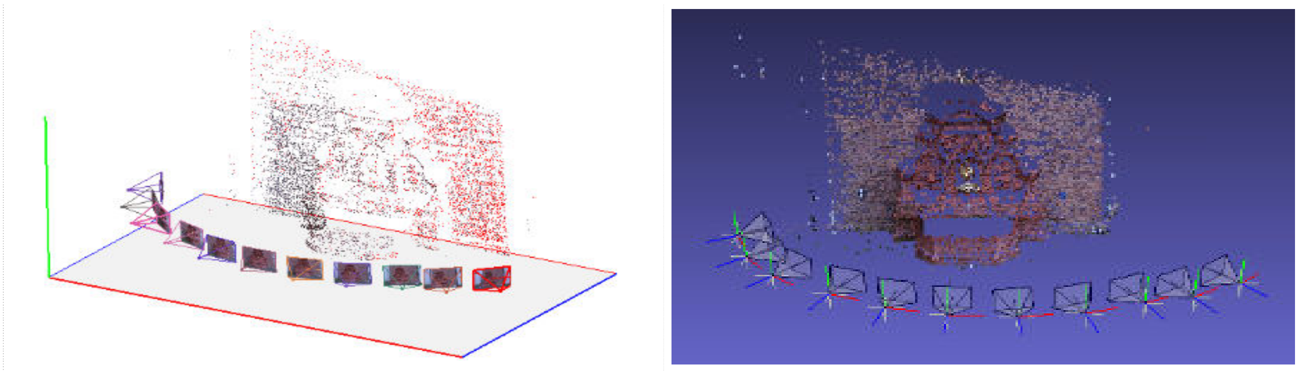
Table 2. The number of matched features between all the possible pairs of images

	Im1	Im2	Im3	Im4	Im5	Im6	Im7	Im8	Im9	Im10
Im0	3733	2373	1420	925	597	375	230	100	54	44
Im1		4097	2659	1591	986	596	303	112	63	56
Im2			3879	2443	1445	818	391	136	73	70
Im3				3853	2428	1452	677	219	139	95
Im4					3812	2358	1251	460	284	142
Im5						3826	2235	947	535	212
Im6							3640	1685	898	346
Im7								2973	1640	548
Im8									3371	1147
Im9										2954

The initial 3D point cloud obtained from this initial pair of images (Im3 and Im5) contains (1833) 3D points.

The rest of the images are added in a sequential manner. The camera's matrix of the added image is calculated and the recovered 3D points are added to the initial point cloud. As a final step, the estimation of the cameras' matrices and the 3D structure are refined with Bundle Adjustment to minimize the reprojection errors given by Levenberg–Marquardt algorithm. The final point cloud recovered from the correspondence features of table 3 contains (13574) 3D points.

Fig. 4 demonstrates the obtained results i.e. the 3D point cloud with the initial positions of the 11 cameras.



Visualization with VisualSfM

Visualization with MeshLab

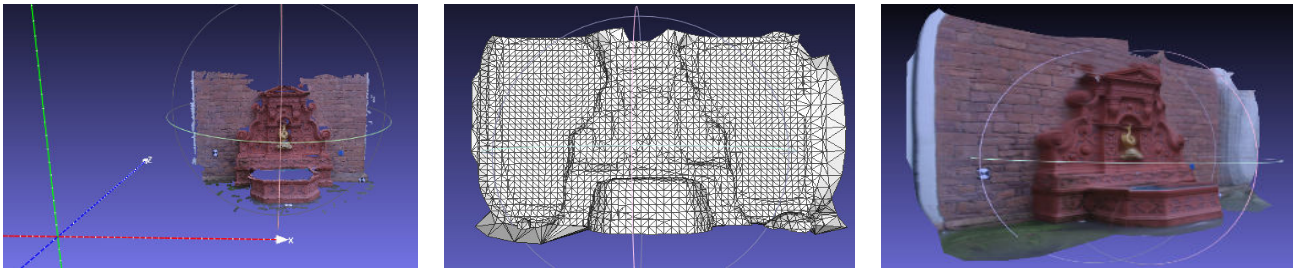
Fig. 4 3D sparse scene with initial positions cameras

The output of the VisualSFM is saved as a file called (bundle.rd.out), which contains the sparse 3D as digital item of extension (.PLY), and the cameras' matrices as (.txt).

3.2 Multi View Dense Scene Reconstruction with CMVS

CMVS exploits the images, the cameras parameters (.txt) and the sparse 3D point could (.PLY) to perform a Multiview Dense Stereo Reconstruction.

Fig. 5 shows the results generated with CMVS visualized with MeshLab, the point cloud contains (325935) 3D points with color information (Red, Blue, Green). The poisson's surface reconstruction filter is applied for generating a 3D mesh from the point cloud. Fig 5 illustrates the results of mesh generation with poisson's surface reconstruction, and the obtained 3D textured model.



Dense 3D model 3D Mesh Textured 3D model

Fig. 5 3D Dense point cloud generated with CMVS

3.3 MVDS Reconstruction Using SURE

In this section we use the software SURE for multiple view dense stereo reconstruction instead of CMVS. We need to provide the path of images and the project files generated by VisualSFM for SURE (.exe) software.

Figure 6 illustrates the obtained with the software SURE dense 3D model with different views. The results are visualized with the software CloudCompare.

As illustrated in fig 6, the second view (right image of the figure) of the 3D point cloud showed clearly the window and the wall recovered from the images. Could Compare offers some functionalities for processing the 3D point cloud like evaluating the accuracy of the 3D point cloud, but can't do mesh building and texture mapping therefore the 3D point cloud is exported again as (.PLY) to MeshLab for mesh building and texture mapping.

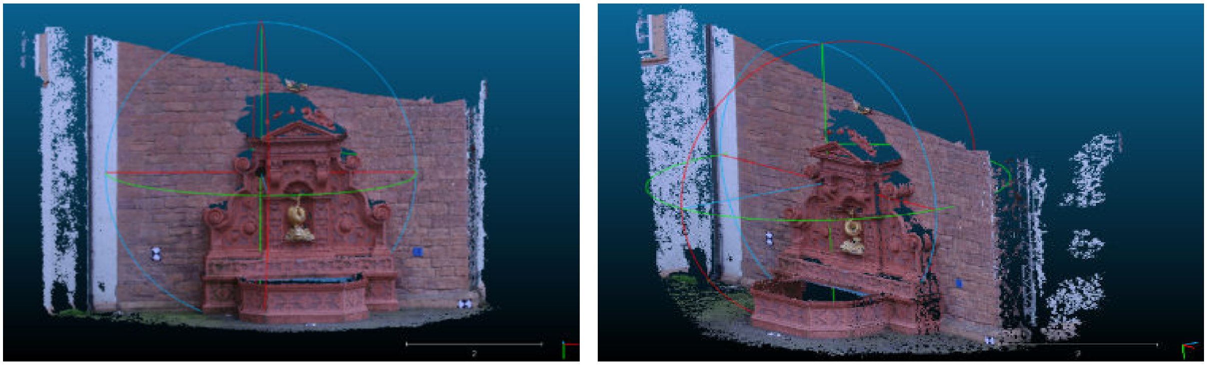
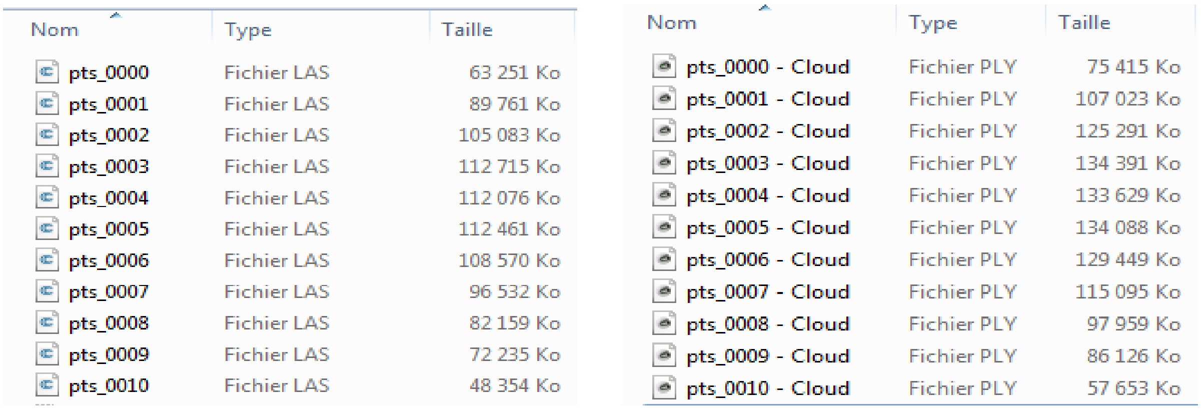


Fig. 6 Different views of the dense 3D point cloud generated with SURE



Nom	Type	Taille
pts_0000	Fichier LAS	63 251 Ko
pts_0001	Fichier LAS	89 761 Ko
pts_0002	Fichier LAS	105 083 Ko
pts_0003	Fichier LAS	112 715 Ko
pts_0004	Fichier LAS	112 076 Ko
pts_0005	Fichier LAS	112 461 Ko
pts_0006	Fichier LAS	108 570 Ko
pts_0007	Fichier LAS	96 532 Ko
pts_0008	Fichier LAS	82 159 Ko
pts_0009	Fichier LAS	72 235 Ko
pts_0010	Fichier LAS	48 354 Ko

Nom	Type	Taille
pts_0000 - Cloud	Fichier PLY	75 415 Ko
pts_0001 - Cloud	Fichier PLY	107 023 Ko
pts_0002 - Cloud	Fichier PLY	125 291 Ko
pts_0003 - Cloud	Fichier PLY	134 391 Ko
pts_0004 - Cloud	Fichier PLY	133 629 Ko
pts_0005 - Cloud	Fichier PLY	134 088 Ko
pts_0006 - Cloud	Fichier PLY	129 449 Ko
pts_0007 - Cloud	Fichier PLY	115 095 Ko
pts_0008 - Cloud	Fichier PLY	97 959 Ko
pts_0009 - Cloud	Fichier PLY	86 126 Ko
pts_0010 - Cloud	Fichier PLY	57 653 Ko

(.LAS) files Generated with SURE Conversion into (.PLY) files

Fig. 7. The conversion of (.LAS) files into the format (.PLY)

In fact, SURE performed dense matching between the images which enabling the determination of 3D information for almost each pixel in the image, therefore it generated 11 dense point clouds from the 11 images, each point cloud is stored with 3D format (.LAS) as indicated in fig. 7, the digital files are opened and visualized with CloudCompare and converted to (.PLY) such that they can be opened with MeshLab. Note in fig. 7, the size of the converted (.PLY) digital files is bigger than the (.LAS) files.

Opening all the files once generate a huge colored 3D point cloud. Merging all the point cloud into a single point cloud demand a powerful calculator, moreover the poisson's surface reconstruction filter is demanding powerful calculator. These digital files containing the point clouds may be uploaded to a web service for further processing.

4. CONCLUSIONS

In this paper we have presented some experiments for creating textured 3D models from a collection of uncalibrated images. The problem is decomposed into a number of tasks; each task is being performed with existing techniques and combined to create realistic textured 3D models. We have used appropriate open source software packages (i.e. VisualSFM, CMVS, SURE, MeshLab).

VisualSFM is a robust software that is used for structure from motion estimation, it is running fast by exploiting multicore parallelism for features detection, matching, and bundle adjustment. However some limitations of the VisualSFM are the following

- The features based reconstruction only works with textured surfaces, which means VisualSFM will not work if too few features are detected e.g. white wall, uniform color objects (note that the used fountain dataset's images are textured).
- The incremental SfM method has some limitations, like sensitivity to initialization (as seen in experiments) and sometimes the user need to try manual initialization.

Multi View Dense Stereo open source software tools i.e. CMVS and SURE are used for generating dense 3D model based on the output of the VisualSFM. MeshLab is used for 3D mesh generation and texture mapping.

The scale of the digital 3D models is not verified in this work, hence in the future we want to apply scaling filters of MeshLab for scaling the 3D model to have the realistic distances of the scene. As a general conclusion, this work demonstrated that there exist a good open source software for 3D reconstruction similar to the best commercial software or web services.

5. REFERENCES

[1] R. Furukawa, H. Kawasaki, R. Sagawa, and Y. Yagi. Shape from grid pattern based on coplanarity constraints for one-shot scanning. IPSJ Transaction on Computer Vision and Applications, pages:139–157, 2009.

[2] 3D Building Reconstruction From Lidar Point Clouds By Adaptive Dual Contouring. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume II-3/W4, 2015, PIA15+HRIGI15 – Joint ISPRS conference 2015, 25–27 March 2015, Munich, Germany.

[3] Klingner, B. Martin, D. Roseborough, J. "Street View Motion-from-Structure-from-Motion,". (ICCV), 2013 IEEE International Conference on , vol., no., pp.953,960, 1-8 Dec. 2013.

[4] Jianxiong Xiao Tian Fang Peng Zhao Maxime Lhuillier∗ Long Quan. Image-based Street-side City Modeling. Proceeding of ACM SIGGRAPH ASIA 2009. Volume 28 Issue 5, December 2009, Article No. 114.

[5] MULLER, P., ZENG, G., WONKA, P., AND GOOL, L. V. 2007. Image-based procedural modeling of fac¸ades. ACM Transactions on Graphics (Aug.), 85:1–85:10.

[6] S., MERRELL, P., SALMI, C., SINHA, S., TALTON, B.,WANG, L., YANG, Q., STEWNIUS, H., YANG, R., WELCH, G., AND TOWLES, H. 2008. Detailed real-time urban 3D reconstruction from video. International Journal of Computer Vision 78, 2, 143–167.

[7] Murray, D. & Little, Using Real-Time Stereo Vision for Mobile Robot Navigation. J.J. Autonomous Robots (2000) 8: 161. doi:10.1023/A:1008987612352.

[8] Pratik Agarwal Wolfram Burgard Luciano Spinello. Metric Localization using Google Street View. IEEE Proceeding of the International Conference on Intelligent Robots and Systems (IROS), 2015.

[9] Sinha, S.N., Steedly, D., Szeliski, R., Agrawala, M., Pollefeys, M.: Interactive 3D Architectural Modeling from Unordered Photo Collections. In: SIGGRAPH, Asia (2008).

[10] Larsen, C., & Moeslund, T. B. (2011). 3D Reconstruction of Buildings with Automatic Facade Refinement. Lecture Notes in Computer Science, 6938 , 451-460. 10.1007/978-3-642-24028-7_42.

[11] Rashmita Khilar. 3D image reconstruction: Techniques, applications and challenges. Proceeding of the IEEE International Conference on Optical Imaging Sensor and Security (ICOSS), 2013.

[12] Liping Zheng ; Guangyao Li ; Jing Sha. The survey of medical image 3D reconstruction. Proceeding. SPIE 6534, Fifth International Conference on Photonics and Imaging in Biology and Medicine, (May 01, 2007); doi:10.1117/12.741321.

[13] Ying-mei WEI, Lai KANG, Bing YANG, Ling-da WU, (May 2013) Applications of structure from motion: a survey. Journal of Zhejiang University-SCIENCE C (Computers & Electronics) (Online). Available: www.zju.edu.cn/jzus; www.springerlink.com.

[14] P. A. Zandbergen and S. J. Barbeau. Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. Journal of Navigation, 64(03):381–399, 2011.

[15] Adam Strupczewski. 3D reconstruction software comparison for short sequences. Proc. SPIE 9290, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, 2014; doi: 10.1117/12.2075922.

[16] J.C. Torres, G. Arroyo, C. Romo, J. De Haro. 3D Digitization using Structure from Motion. CEIG - Spanish Computer Graphics Conference, 2012.