Instant Surface Reconstruction for Incremental SfM

Kichang Kim† Takayuki Sugiura† Akihiko Torii‡ Shigeki Sugimoto† Masatoshi Okutomi‡
†{ kichang.k, tsugiura, shige }@ok.ctrl.titech.ac.jp
‡{ torii, mxo }@ctrl.titech.ac.jp
Tokyo Institute of Technology, Tokyo, Japan

Abstract

We propose an instant reconstruction of 3D surface meshes for a set of images and their camera poses and sparse 3D points computed by structure from motion (SfM). We aim at the proposed method to be compatible with an incremental structure from motion system as it can immediately generate the 3D surface model seen from the latest view point recovered by SfM. The proposed method consists of four steps: reference and target image selection adaptive to so-far recovered surface models; triangular patch initialization by taking into account the scene structures; fast 3D pose estimation of each triangular patch using inverse compositional image alignment (ICIA); surface generation by robustly refining the 3D triangular patches and integrating into pieces of surface meshes. The surface reconstruction results are shown on real image datasets of indoor as well as outdoor scenes and compared with the state of the art approach.

1 Introduction

3D reconstruction of camera poses and points from a set of camera images becomes a feasible task according to the success of incremental structure from motion (SfM) pipelines [15, 19, 9]. The recovered camera poses and points are often used as the input for generating dense 3D points [5, 11] and 3D surfaces [8, 18] to provide richer scene visualization for entertainment applications as well as digital archiving of historical heritages.

For incremental SfM from collection of images, the one of the breakthroughs is Bundler [14, 15] which reconstructs the scene in a seed-and-grow manner by repeating the addition of new camera poses and points and bundle adjustment. In more detail, Bundler precomputes feature tracks using wide baseline matching [10] with RANSAC, then, initially recovers camera motion and 3D points by the five-point algorithm [12], grows by adding new camera poses by solving 3D-2D pose problem (DLT) [7] and refines by bundle adjustment [17, 16]. The efficiency is further improved in VisualSfM [19] by incorporating with the power of multicore CPU and GPU.

One of the successful and popular approaches for surface reconstruction combining with the sparse SfM is to compute denser point clouds by patch-based multiple view stereo (PMVS) [5], then apply Poisson surface reconstruction [8] and finally generate 3D meshes. Since PMVS provides robust 3D point clouds by conservatively starting from the stable sparse 3D points, the final surface models are very accurate. Vu *et al.* [18] recently proposed a surface reconstruction technique which provides 3D surface models with impressive quality. This method generates 3D surface meshes by extracting the boundary surface computing the intersections of viewing rays from cameras and the 3D Delaunay tetrahedras generated from 3D point clouds. The boundary extraction is formulated as a binary labelling problem where a unique solution can be efficiently found by a standard graph-cut algorithm. The 3D surface meshes are further refined by taking into account photo consistency. Since those recent state-of-the-art techniques aim at providing accurate and complete models for a fixed set of images, efficiency of computation and flexibility to the online input are out of focus.

In contrast, we are particularly interested in combining with the incremental SfM process as the system can output the surface models as soon as new image is processed. Such a system is suitable for recovering a midium-scale scene, *e.g.* a room and a building captured by a few users. Towards a system performing fully incremental and online 3D reconstruction, we propose the instant 3D surface reconstruction for a small subset of images and their camera poses and sparse 3D points. The main contribution is to instantly accomplish 3D surface reconstruction of the reference image based on the selection of the reference and target cameras adaptively w.r.t. the 3D surface models so-far reconstructed and on the fast estimation of 3D poses of each patches. Figure 2 briefly illustrates the pipeline of the proposed method.

2 Instant surface reconstruction

In this section, we propose the fast and robust surface reconstruction algorithm for generating 3D surface meshes from sparse input images. The proposed method consists of the folloiwng four steps:

(1) Reference and target images are adaptively determined w.r.t. the scene so-far recovered for suppressing computational cost.

(2) Small-triangle patches are generated from the reference image and 3D poses of the patches are initialized by using the sparse 3D points of SfM;

(3) The patch poses are refined by examining the photo consistency using the reprojection of the patch to the spatially neighboring target images;

(4) The patches are refined and integrated into larger pieces of surfaces.

2.1 Reference and target image selection

Every image having its camera pose recovered by incremental SfM is a candidate for the reference image to proceed surface reconstruction in our approach. A candidate image is accepted to be the reference image

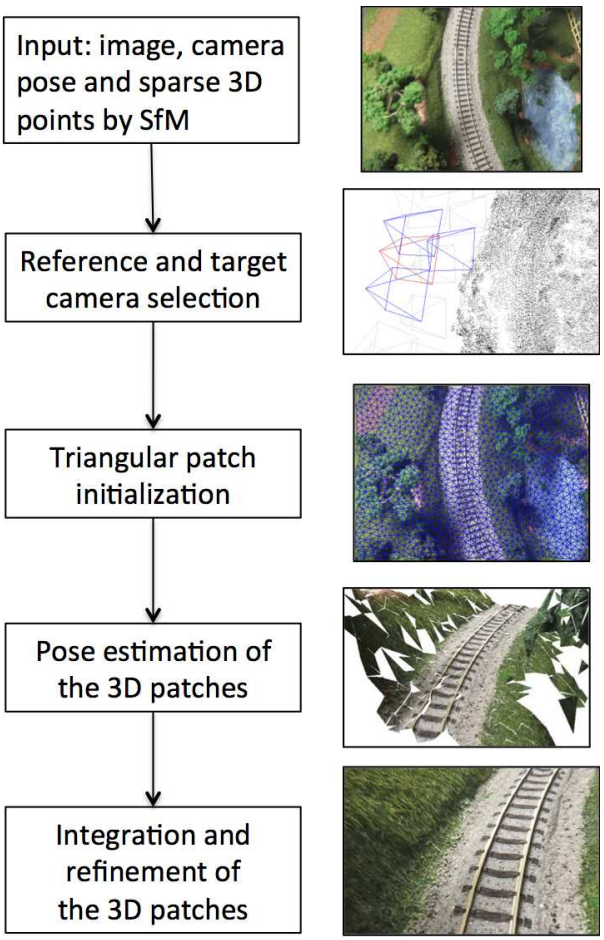


Figure 1. Partial pipeline of the proposed method

to proceed for the 3D surface generation if the image is not occupied by the reprojected surface meshes already reconstructed. This reference image selection is beneficial not only for computational cost but for visual quality since it avoids to generate overlays of duplicated 3D patches. The order of examining the candidate follows the order of incremental SfM reconstruction.

For the reference image, we pick the target images which are used for 3D patch pose estimation in Section 2.3. We select three images as the target images of examining the photo consistency according to the size of overlapping view fields with the reference image computed from the number of 3D points sharing in common. In more detail, the cameras already recovered by SfM sharing sufficient number of 3D points with the reference camera are sorted according to baselines to the reference as the distant camera is better ranked. We select three images as the target and if there are less than three satisfying the above condition, the reference image is cancelled and another one is selected in the candidate list.

2.2 3D triangular patch initialization

The reference image is divided into the pieces of triangular patches by using Delaunay triangulation for Harris points [6] detected on corners and edges. Note that we use both corner-like and edges-like features in contrast to popular tracking methods using corner-like features only. As a post-processing to remove too many features detected on edges, we subsample them as every distance among neighboring points is larger than a predefined threshold. After this subsampling, we use the Delaunay triangulation for these Harris points. There can be too sharp triangular patches, often vicinity to the image borders, due to the design of Delaunay triangulation. We remove such triangular patches since they are infeasible for estimating their 3D poses and give slow convergence in the iterative optimization stage.

After generating the triangular patches on the reference image, we estimate the 3D pose of each patch independently. By using the camera pose of the reference image, we compute the 3D poses of patches by estimating depths of three vertices of the triangle. In more detail, in order to estimate the initial depth of vertices, we use the (SURF) feature points in the reference image associated to the SfM 3D points. For the patch containing at least one feature point associated to the 3D point, we compute the initial depth of vertices by averaging depth of the SfM feature points. For the patch not containing any SfM feature point, we give the value of average depth of nearby patches. We repeat this initialization until every patch has its initial depth.

2.3 3D triangular patch estimation

After the initialization, the depths of vertices of the triangular patches are further refined by examining the photo consistency using the reprojection of the patches to its spatially neighboring target images. Using the target images, the depths of vertices are optimized by minimizing the cost function,

$$E(\mathbf{d}) = \sum_{n} \sum_{\mathbf{u} \in \mathtt{P}} \left(I_R(\mathbf{u}) - I_n(T_n(\mathbf{u}, \mathbf{d})) \right)^2 \qquad (1)$$

where $\mathbf{d}$ is a 3-dimension vector composed by inverse depths of three vertices of each triangular patch, $\mathbf{u}$ is a 2D image coordinates of a patch $\mathtt{P}$ on the reference image, n indicates the index of target image, $T()$ is a mapping function of a 2D point on the reference image to the target image. $I_R()$ and $I_n()$ are image intensities of reference and target images as a function of 2D image coordinates. This minimization problem is solved for each patch independently by using Gauss-Newton method which can be efficiently implemented with ICIA technique [1]. Since there can be erroneous 3D pose estimation due to weak texture, occlusions and illumination changes, we cancel the depth estimation of such patches by examining whether the optimization is converged in a predefined maximum iteration and later recover them using the depths of the neighboring patches stably estimated in Section 2.4.

2.4 3D patch refinement and integration

First, we seek for the patch with no depth surrounded by three patches whose depths are successfully estimated. The depths of vertices of this patch are recovered by averaging the depths of each pair of corresponding vertices (Figure 2 (a)). Next, we seek for the patch with no depth surrounded by two patches patches successfully recovered. If these two patches lie on a plane, the depths of vertices of this patch are computed as they fit on it (Figure 2 (b)). We repeat these

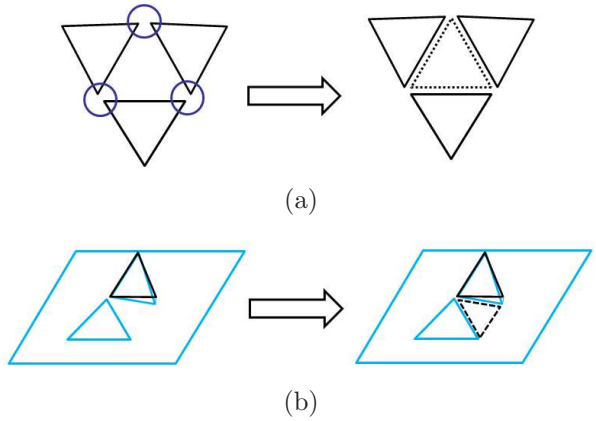


Figure 2. Patch refinement. (a) Missing patch recovered by three surrounded patches. (b) Missing path recovred by two neighboring patches lying on a 3D plane.

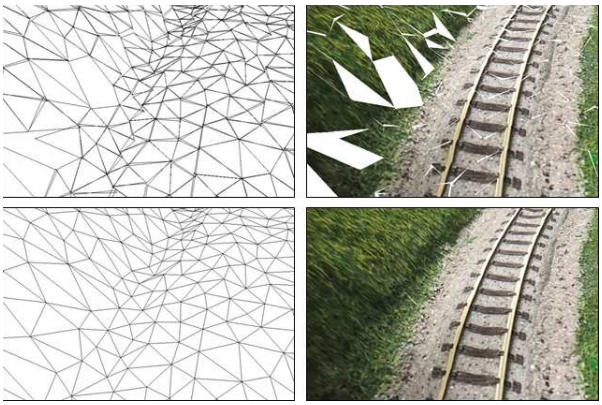


Figure 3. Example of 3D patches before refinement and integration (top) and after (bottom).

two processes until no new patch is updated accordingly.

Finally, we merge the common vertices of neighboring patches on the reference image and integrate into pieces of surface meshes if the distances among the vertices in 3D are under the threshold computed using the mean depth of all feature points (Figure 3).

As a whole system, when the processes of surface generation for a reference image are finished, we wait for a new image, camera pose and 3D points computed by SfM and then start running the surface generation by setting the new image as a reference image.

3 Experimental results

We implemented the 3D surface reconstruction algorithms by C++ using open source libraries: `OpenCV` [2] as a basic computer and S-Hull [13] for Delaunay triangulation. All the experiments are run on a standard PC composed of Intel Core i7 3.33GHz CPU and 32GB RAM.

The camera poses and sparse 3D points are computed by VisualSfM [19] and used as the input data for surface reconstruction. We use PMVS2 [4, 5] both for comparing the resulted 3D models quantitatively and for evaluating the computational efficiency. As a comparison of surface models, it is also possible to use non-textured 3D surface meshes obtained by Poisson surface reconstruction bundled in Meshlab [3] but visual comparison of non-textured wire-frame models require some skill. Even though PMVS2 provides dense point clouds only, we take the advantage of output with colors.

Dataset	Desk	Vienna	DiTrevi
#Images	84	82	168
PMVS (sec.)	127	225	278
Proposed (sec.)	18	30	18

Table 1. Computational time.

Figure 4 shows, from left to right, an example of input images, camera poses and sparse 3D points by SfM, 3D surface mesh models obtained by the proposed method, and colored dense 3D points obtained by PMVS. Table 1 summarizes the number of input images, computational time on the proposed method and PMVS for each dataset. The improvement on computational time is significant for every dataset while the visual quality of resulted model is comparable or even better. Furthremore, another advantage of the proposed method is on the online visualization of the surface models, *i.e.* we can visualize them as soon as new reference image is processed.

4 Concluding remarks

We proposed a method for instantly reconstructing 3D surfaces from images with camera poses and sparse 3D points obtained by SfM. The reference camera selection so as to have minimum overlaps and the fast computation of 3D triangular patches by using ICIA bring a significant improvement on computational efficiency.

The further comparisons with other state-of-the-art surface reconstruction and quantitative evaluations are left as future works. We are currently working on the system level integration to the online SfM algorithm in order to achieve a fully online 3D reconstruction system.

Acknowledgment This work was partly supported by Grants-in-Aid for Scientific Research (21240015) from Japan Society for the Promotion of Science.

References

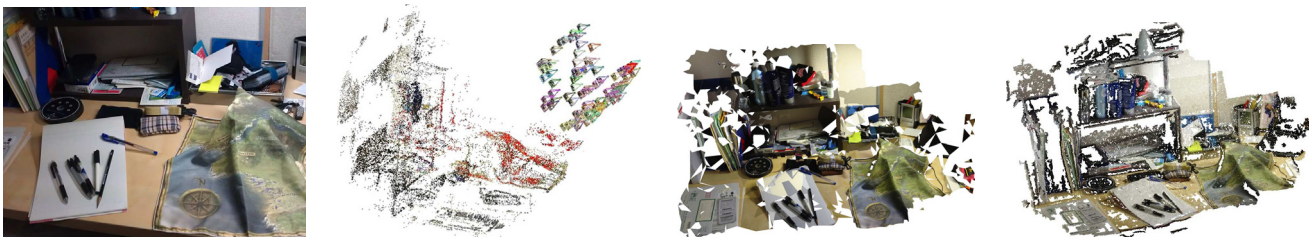
[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] P. Cignoni, M. Corsini, and G. Ranzuglia. Meshlab: an open-source 3d mesh processing system. *ERCIM News*, 2008(73), 2008.

[4] Y. Furukawa. Patch-based multi-view stereo software (PMVS - version 2), 2010.

[5] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 32:1362–1376, 2010.



(a) Desk

(b) Vienna

(c) DiTrevi

Figure 4. Examples

[6] C. G. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[7] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.

[8] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *In Proc. of Fourth SGP*, SGP '06, pages 61–70. Eurographics Association, 2006.

[9] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *In Proc. Sixth IEEE and ACM ISMAR'07*, Nara, Japan, November 2007.

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, pages 384–393, 2002.

[11] Richard A. Newcombe, S. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, pages 2320–2327, 2011.

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

[13] D. A. Sinclair. S-hull: a fast radial sweep-hull routine for delaunay triangulation, 2010.

[14] N. Snavely, S.M. Seitz, and R.S. Szeliski. Photo tourism: Exploring image collections in 3D. In *Proc. SIGGRAPH*, pages 835–846, 2006.

[15] N. Snavely. Bundler: Structure from motion (sfm) for unordered image collections. `http://phototour.cs.washington.edu/bundler/`, 2008.

[16] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[17] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment: A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[18] H. Vu, P. Labatut, J. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE TPAMI*, 34(5):889–901, 2012.

[19] C. Wu. VisualSFM: A visual structure from motion system. `http://homes.cs.washington.edu/~ccwu/vsfm/`, 2011.