Facades Modeling from a Ground-View Video with Map Constraints

LIU Ruijun
Ecole Centrale de Nantes, CERMA-IRSTV
1 Rue de la Noë, 44321 Nantes, France
ruijun.liu@ec-nantes.fr

Myriam Servières
Ecole Centrale de Nantes, CERMA-IRSTV
1 Rue de la Noë, 44321 Nantes, France
myriam.servieres@ec-nantes.fr

Guillaume Moreau
Ecole Centrale de Nantes, CERMA-IRSTV
1 Rue de la Noë, 44321 Nantes, France
guillaume.moreau@ec-nantes.fr

Abstract

We present a 3D modeling approach to reconstruct urban facades with map constraints. Our main constribution is to use GIS data to filter the points cloud to improve the reconstruction process. The method's input is a ground-view video sequence captured by a common webcam. We extract original feature points cloud by using structure-from-motion technique. First, the footprint from GIS enables us to achieve filtering process, which is a key step to generate a quality reconstruction result in our method. Secondly, our method reconstructs the final facades based on the footprint. We show a simple and accurate facades modeling system to reconstruct building facades.

1 Introduction

With the development of computer vision and geographical information system (GIS), 3D-GIS is applied to more and more fields. 3D-GIS face two challenges: data acquisition and visualization, which can be seen as a closed loop, i.e. images are used to acquire information for 3D-GIS as well as 3D-GIS is reprensented by digital images. The virtual facade modeling plays an important role in the 3D-GIS reconstruction process. Facade modeling often uses a set of images from a video sequence as input, based on several previous techniques: feature detection and matching, structure from motion (SfM), Multi-View Stereo (MVS) ([1], [2] and [3]) and so on. There are several excellent platforms which are able to achieve facade modeling, such as Google Sketchup and ArcScene platform. Google Sketchup can reconstruct facades of the building friendly, but it requires abundant interactive operations with user. Researchers in computer vision and computer graphics field attempt to do facade modeling much more rapidly, accurately and easily.

This paper is organized as follows. In Section 2, we will review the related works; in Section 3, we will present our method; the results are shown in Section 4; the last part is our conclusion in Section 5.

2 Related works

Facades modeling from images or video has received tremendous interest in the computer vision field, and there is a lot of prior works. Here, we simply classify facade modeling from a video sequence as either detailed dense reconstruction approaches that are often based on MVS techniques, or coarse sparse reconstruction approaches that are based on SfM techniques.

Some approachs employ MVS techniques, which is based on the idea of 3D surface information reconstruction from a set of images from different views, for example, [1], [2], [3] and [4].

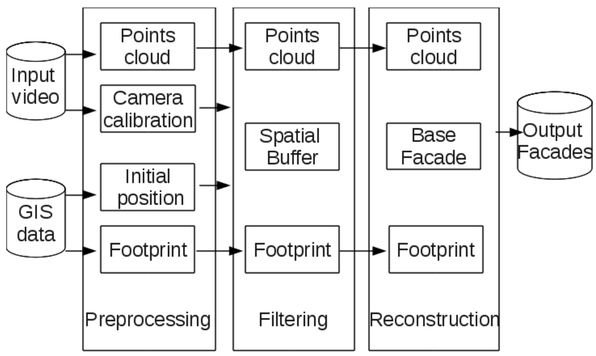

Figure 1. Pipeline of our facade modeling method.

Furukawa et al. [1] and Goesele et al. [2] have developed an automatic method to reconstruct architecture, textureless and non-Lambertian surfaces, which makes the reconstructed architectural facades appear non-flat and full of bumps and hollows. Based on [1], Furukawa et al. [4] describe an internet-scale MVS system. The authors state a novel cluster algorithm to divide the unorganized images from Internet. This method also achieves an automatic reconstruction, but is time consuming. Pollefeys et al. [3] present a real-time 3D reconstruction system from video. It can create ground-view 3D model. Their MVS approach is based on a huge frames video as input, and is also time consuming. Seitz et al. [5] give us a summary of many MVS approaches.

R. Newcombe et al. [6] present a rapid and dense reconstruction method requiring a handy camera. They achieve a real-time reconstruction by estimating an optical flow field to update a depth map. The authors use a multi-scale approach in the reconstruction process [7]. Stüehmer et al. [8] also focus on geometric reconstruction in real-time using simple device. They provide a robust depth map estimation method to generate reconstruction result. These two methods both

Figure 2. Some keyframes from a video sequence (*Château des Ducs de Bretagne*)

uses SfM technique to obtain points cloud as a reconstruction input, but they only reconstruct an indoor scene, and will not be suitable for facade modeling in outdoor environment.

In contrast to these methods, our method aims at a reconstruction process in outdoor environment, named facades modeling. Combined with GIS data, we propose a novel filtering algorithm to reconstruct facades in the outdoor environment. Comparing with MVS methods, our reconstruction process is easily implementable.

3 Method

3.1 Overview

In this subsection, we provide the pipeline of our facades modeling method (Figure 1). This approach uses a video sequence (one example is shown in Figure 2, *Château des Ducs de Bretagne* in Nantes) and GIS data as input. There are three main steps to reconstruct facades: preprocessing (Section 3.2), filtering (Section 3.3) and reconstruction process (Section 3.4). First, we use SfM method to implement preprocessing, obtaining several information, such as keyframe, points cloud, camera parameters and initial position. We also get the building footprint from GIS. Next, to avoid bad effect from noisy points, a constraint method will be applied to filter noise. We generate a set of spatial buffers to remove bad points. At last, facades model is generated by using our reconstruction method.

3.2 Preprocessing

Before filtering and reconstruction process, we do some preprocessing. We start from a set of extracted SfM points and estimate the camera initial position for the filtering process.

3.2.1 Structure from Motion

Structure from motion is used to estimate the camera parameters (extrinsic and intrinsic parameters), and to find the object's structure by analyzing local information. Regarding SfM technique, there are several classic feature trackers as for example Kanade-Lucas-Tomasi (KLT) feature tracker [9] and MonoSLAM system [10]. Some methods do not adapt to real-time process. Therefore, we use a state-of-the-art real-time SfM method, Parallel Tracking and Mapping system (PTAM) [11], to obtain a points cloud and camera parameters. PTAM updates the map using bundle adjustment, and extracts thousands of feature points. Eventually, we are able to compute the normal of each points as one input of the reconstruction process.

3.2.2 Initial Position and Footprint

With the SfM methods, both camera's pose and its intrinsic parameters could be estimated. But in our application, we make the assumption that we know the initial position clearly in GIS. We could also use other method to calculate the initial position, for example Bioret's method [12]. Then the local footprint of building can be extracted from 2D GIS data.

3.3 Filtering

Quality feature points are the key to the accurate reconstruction result. But the 3D points extracted from SfM technique are noisy. We design a filter to optimize them. Our filter is a spatial buffer generated with map constraint from GIS. This section describes the filtering process to remove noisy points from the points cloud.

3.3.1 Matching Footprint and Spatial Buffer

Based on the preprocessing, we obtain the initial position of the camera (camera extrinsic parameters). The geo-referencing process enables us to extract the right footprint from GIS. Then, we can align the 2D footprint with 3D points. Figure 3 shows us the matching process between the points cloud from top view and the building footprint. The blue curve is part of the footprint of Chateau des Ducs de Bretagne. We note that there are many points (feature points) near the footprint and several sparse points (noisy points) far from the footprint.

Therefore, a 2D buffer has been created based on the 2D footprint. This 2D buffer can be considered as a 3D spatial buffer with infinite height, as shown in Figure 4. Figure 4 (a) shows us one spatial buffer: value b is buffer width and the height is infinite. We use the spatial buffer to discard the outside points (blue points in Figure 4 (b)).

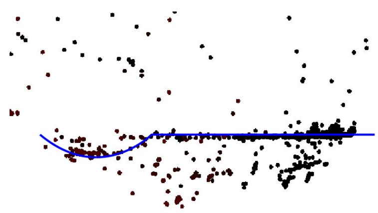

Figure 3. Matching process between points cloud and building footprint.

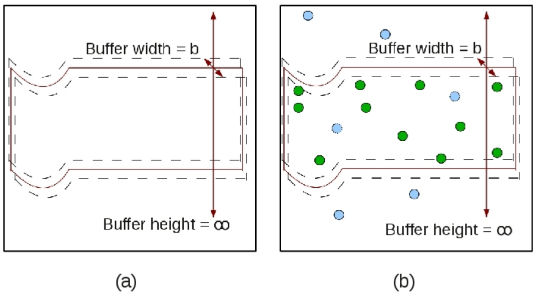

Figure 4. (a) spatial buffer, (b) 3D points in the spatial buffer

3.3.2 Establishing Coordinate and filtering Process

To simplify the filtering process, we project the 3D points into a 2D coordinate system. As the initial position is known, a cartesian coordinate system could be established, as shown in Figure 5. Initial position is projected as the origin, and XY-axes are parallel to the XY-axes of world coordinate system.

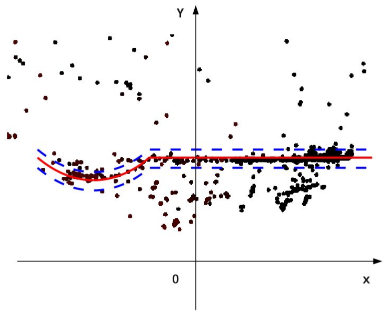

Figure 5. Filtering noisy points with footprint constraint in Cartesian coordinate.

Each 3D point is projected to this coordinate as a 2D point from the top view, we define the function

$$point(i) = P(x, y)$$

meanwhile, the footprint is represented as function

$$y = f(x)$$

and then, we choose a constant factor δy as a constraint parameter. If the variable y of the point overtops the scope *[f(x)-δy, f(x)+δy]*, we will discard this point. We add a label function *L(point(i))* to each point depending on its coordinates. If the point is discarded, the label *L(point(i))* is 0; otherwise, the label is 1. Therefore, regarding *point(i) = P(x,y)*, using the function *ρ(point(i))* as :

$$\rho(point(i)) = \frac{1}{\delta y}\|point(i) - f(x)\|,$$

the label function is defined as :

$$L(point(i)) = \begin{cases} 0 & \rho(\text{point(i)}) < 1 \\ & \cap\ \lambda(\text{point(i)}) \geq \rho(\text{point(i)}) * \text{a} \\ & \\ 1 & \rho(\text{point(i)}) < 1 \\ & \cap\ \lambda(\text{point(i)}) < \rho(\text{point(i)}) * \text{a} \\ & \\ 0 & \rho(\text{point(i)}) \geq 1 \end{cases}$$

where *λ(point(i))* is the outlier count of *point(i)*, as defined by PTAM, and a is the outlier threshold. Experiments allow us to set $\delta y = 2$ and $a = 9$.

The footprint filtering result is shown in Figure 6.

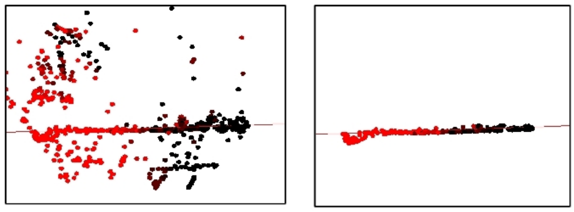

Figure 6. Left: original 3D points, Right: 3D points after footprint filtering algorithm

3.4 Reconstruction

We assume that all facades have the same ground height and are vertical to the ground plane. Then we define each facade as a plane, which has 3 parameters: left boundary, right boundary and height. Left boundary and right boundary are first estimated using the footprint. We only need to calculate the facade height. As we know the filtered 3D points, the height is computed by the max of a set of points in the buffer, fitting that facade. Figure 7 displays the details of one facade reconstruction. Therefore, we can generate facade simply and automatically using this method.

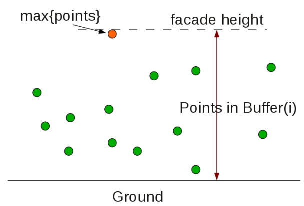

Figure 7. One facade modeling process: the orange point is the max one in this group

We repeat this method for every visible facade in our input video.

4 Results

The following results have been implemented in C++ code and on a PC with Intel Core i7-870 CPU processor, Nvidia Quadro FX 580 graphics card. The entire system consists of three components: preprocessing, filtering and modeling. We embedded our preprocessing code into PTAM [11] to obtain the 3D points clouds and outlier count of each point. The input video sequence is captured by a webcam Logitech QuickCam Pro 9000, which was put in a vehicle.

The pipeline of one example is shown in Figure 8, the left is the input data (GIS data and video including nearly 300 frames); the central process of our approach consists of extraction and filtering process. In the final part, 7 facades are reconstructed, as shown in the right part of Figure 8.

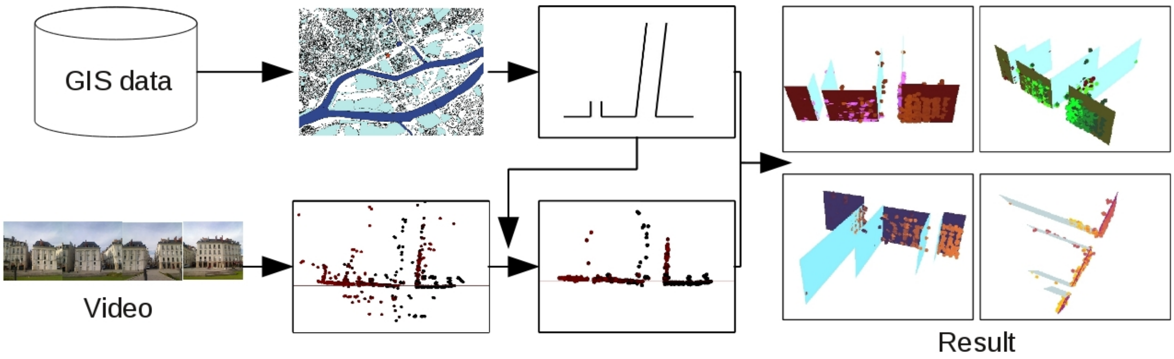

Figure 8. Pipeline and reconstruction results

5 Conclusions

In this paper, we have presented a facades modeling approach that uses a video and map constraints extracted from GIS. Further, our facades result can be used in the data acquisition process for 3D-GIS. Our contribution is twofold: a new facades modeling scheme and a map constraint filtering algorithm.

One limitation is that our method now only reconstructs off-line and off-site. It needs improvement to be applied in Augmented Reality (AR) environment. Therefore, our future work includes reconstruction in real time and an AR display on site.

References

[1] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[2] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz, "Multi-view stereo for community photo collections," in *IEEE 11th International Conference on Computer Vision(ICCV) 2007*, 2007.

[3] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles, "Detailed real-time urban 3D reconstruction from video," *International Journal of Computer Vision (IJCV)*, vol. 78, no. 2-3, pp. 143–167, 2008.

[4] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *CVPR*, 2010.

[5] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, (Washington, DC, USA), pp. 519–528, IEEE Computer Society, June 2006.

[6] R. A. Newcombe and J. Andrew, "Live dense reconstruction with a single moving camera davison," *CVPR*, 2010.

[7] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions," in *Proceedings of the Shape Modeling International 2003*, (Washington, DC, USA), IEEE Computer Society, 2003.

[8] J. Stuehmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Pattern Recognition (Proc. DAGM)*, (Darmstadt, Germany), pp. 11–20, September 2010.

[9] C. Tomasi and T. Kanade, "Detection and tracking of point features," tech. rep., CMU-CS-91-132, Carnegie Mellon University, 1991.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 6, pp. 1052–1067, 2007.

[11] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, (Nara, Japan), November 2007.

[12] N. Bioret, G. Moreau, and M. Servières, "Outdoor localization approach based on image/GIS correspondence using a simple 2D building layer," in *2nd workshop on Mobile Geospatial Augmented Reality. Quebec*, Lectures notes in geoinformation and cartography, Springer-Verlag, 2008.