EFFICIENTLY BUILDING 3D LINE MODEL WITH POINTS

Xinkai Gao, Qi Jia, Haojie Li, He Guo*

Dalian University of Technology
School of software
Dalian, Liaoning, China

ABSTRACT

3D modeling is a popular topic in the field of computer vision. Most of the existing methods based on the interest point matches or the line matches. The point-based methods are more mature, but usually less intuitive. Lines can provide more structural information and thus more intuitive, however, lines have more complex geometric properties, which makes the line-based methods usually time consuming. In this paper, we proposed a simple and effective approach of 3D modeling, which takes the intersections of the real lines and the virtual lines passing through interest points to construct the 3D models, and use a simple invariant to filter out wrong matches. Thus we can use the mature technology of point-based methods while the 3D point cloud maintain the structural information of lines. Experiments show our method can build meaningful 3D point cloud efficiently.

Index Terms— 3D modeling, virtual lines, intersection

1. INTRODUCTION

Building 3D models from images is an interesting and challenging task in computer vision. Most of the existing Structure form Motion(SfM) methods [1, 2]are based on interest points such as SIFT [3]. State-of-the-art point-based SfM methods can quickly and robustly get the camera poses and the 3D point cloud. However, the point clouds are usually sparse, with little structural information or even meaningless. As a result, some very time consuming subsequent Multi-View Stereo(MVS) pipelines such as PMVS [4] or SURE[5] are indispensable to create a meaningful dense point cloud.

Although lines can provide more structural information than points, the study of line matching methods[6, 7] and the 3D modeling methods based on line correspondences[7, 8, 9] is still far behind. This is because robust camera pose and motion estimation from line correspondences is much more difficult: The location of line segment endpoints are hard to predict due to unstable line detection methods and occlusion, and a line in one image may break into several parts in another image. The inaccurate correspondences of endpoints render direct computation of epipolar geometry impractical. So in the line-based SfM methods, many efforts are paid to reduce ambiguity, which makes line-based methods more complicate than point-based methods. In [10], at least 50 line correspondences are required to provide stable camera pose estimation, which are not always available. Bartoli and Sturm [11] proposed a full SfM pipeline based on line segments, including bundle adjustment steps. They represent 3D lines using Plücker coordinates and used a trifocal tensor, because two views cannot provide accurate camera motion if use lines only. Branislav Micusik et al[8] avoid to take endpoints into pose estimation, and proposed a novel error measuring method for bundle adjustment, but employ RANSAC for many times in the algorithm, which cause heavy computational burden.

Both points and lines are not perfect tools for 3D modeling, some researchers use more complex features such as planes [12, 13], but complex features often take more computation time. Many researchers also take the combination methods of lines and points or use special points to build 3D models. Alexander et al[14] proposed a linear time pose estimation method based on lines and points and is more accurate than traditional methods only based on points. Bartoli et al[14] proposed a method to solve the 3D reconstruction of lines by considering pencil of points(POPs) on lines. [7] utilize cross ratio constructed from points and lines to solve line matching problems. Moreover, they take the intersections of real lines and virtual lines formed by points to build 3D models. Which can directly use the existing point-based SfM to get 3D models with structural information from lines.

Inspired by[7], we propose a new approach of 3D modeling using intersections on lines: We first get interest point matches and line matches in the images and take the intersections of real lines and virtual lines passing through interest points to build the 3D model. In [7], the intersections must be taken during the line matching procedure, which means they can not take other line matching results to build the line models. Our approach, in another hand, does not rely on specific point or line matching method thus more flexible.

In this paper, we use a simple but effective geometric invariant to choose coplanar points and lines, and the intersection points on lines are used for 3D reconstruction. This approach can apply to existing state-of-the-art point-based methods, and only few number of matched interest points are required. Thus we can efficiently get 3D point clouds without losing the structural information of lines.

2. USING POINTS TO BUILD LINE MODELS

Most of the traditional line-based 3D modeling methods take the line as a whole. In this paper, we build the pixel-wise correspondence between lines through a simple approach, where lines are represented by many individual points, and the points will be used to build meaningful 3D point cloud using existing mature techniques.

2.1. Represent lines by points

The algorithm starts with point matching (such as SIFT[3]), and line matching(such as[6]). With the correspondences of lines and points, we can generate some virtual intersection points on the matched lines.

As illustrated in Fig. 1, $(A, A')$and$(B, B')$are two pairs of matched interest points in a pair of images, (a, a'),(b, b') and (c, c') are three pairs of matched lines. The dotted lines are the virtual lines jointing the interest points. The virtual lines intersect with real lines at some virtual points. Because the points and lines are matched in advance, we can get the correspondences of the intersection points in different images easily. These virtual points will server as basic elements rather than the original points or lines to build 3D models.

However, many of the virtual points can not be used to build 3D model mainly due to the following reasons: 1. Wrong point or line matching results. 2. The virtual lines may not be coplanar with the real line, which means the two lines do not really have intersection, in another word, the intersection point in the 2D image does not have a real corresponding 3D point. In Fig. 1, where virtual line $< A, B >$ is coplanar with a and b, so the virtual intersections (green dots)can be taken to build 3D model while the intersection of $< A, B >$ and c (red dots) should be abandoned as the intersection do not actually exist. In the manmade scene where lines are abundant, a line is usually located on a plane surface or is the intersection of two planes. Only if there are enough interest points on the same plane where the line is located, we can get enough intersections to represent the line. We show in our experiments that only few number of points are necessary to build the 3D model.

2.2. Getting real intersection points

In[15], researchers proposed a coplanar line-point affine invariant, which is constructed by one line and two points. We briefly introduce the construction of the invariant here: In Fig. 2, l is a line and A and B are two points coplanar with the line, d_1 and d_2 are the distances from the two points to the line, respectively. The ratio of d_1 and d_2 is proved to be affine invariant.

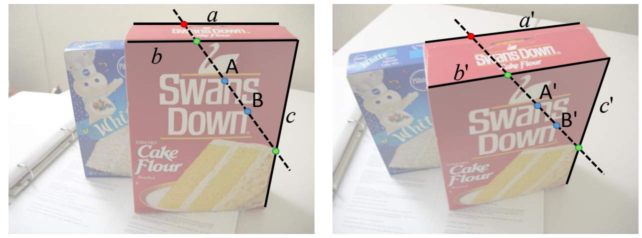


Fig. 1. Virtual lines and the intersections with real lines.

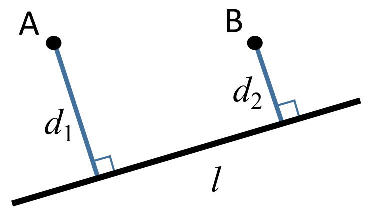


Fig. 2. The line-point invariant.

We use this invariant to filter out fake intersections resulting from wrong matches and non-coplanar line and points. For each pair of interest points and a line, we calculate the line-point invariant and the corresponding invariant in another image with the corresponding line and points. If the values of the two invariants in two images are not similar, we consider that the virtual line jointing the two points may not be coplanar with the line, and the intersection will not be calculated. Otherwise, the two intersections in two images will be taken as a pair of matched intersection points. Suppose the values of two invariants in the two images are V_1 and V_2, respectively, then the similarity of the two invariants is calculated by:

$$S = e^{-||V_1 - V_2||}.$$

Note that the invariant is calculated from the ratio of two distances. Consider in one image, we have two points with distances of 2 and 50, respectively, so we can get two different invariant values: 25 (50/2) or 0.04 (2/50). And the distances of the corresponding points in another image are 1.9 and 49, we can also get two invariants: 25.8 or 0.039. Thus we can get two very different similarities $e^{-0.8} = 0.449$ or $e^{-0.001} = 0.999$ of the two values only because of the different orders used. In this paper, we take a strict way to measure the similarity of the two values: We change the order of two distances to get two different ratios, and the minimum similarity are taken as the total similarity.

$$S_v = min(e^{-||V_1 - V_2||}, e^{-||1/V_1 - 1/V_2||})$$

If $S_v < 0.95$, we think that at least one of the points and the lines may not be coplanar, or there is at least one wrong match, and the intersections should be abandoned.

The invariants can filter out the intersections of non-coplanar lines, but some points may still significantly relocate from the true locations with inaccurate line directions, especially when two lines are nearly parallel. To restraint such errors, we set a threshold for the angle between the virtual line and the real line. In our experiment, if the angle is not greater than $\pi/8$, their intersection will also not be calculated.

Once we get the matched intersection points, we can use the existing point-based bundle adjustment methods for 3D reconstruction. The intersections are located on real lines, so the 3D point cloud will maintain the structure of lines. Moreover, point-based line models are more robust to line detection errors as they are not sensitive to inaccurate endpoints and line directions.

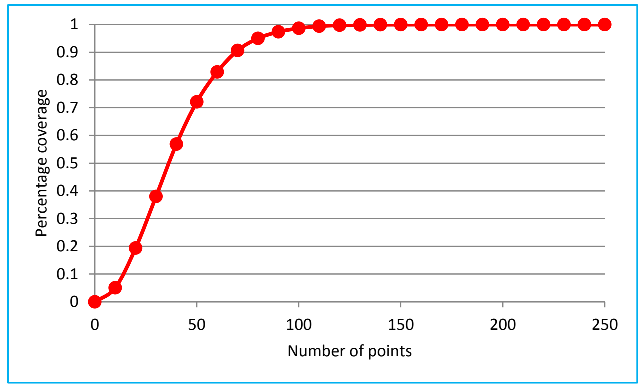


Fig. 3. Scene percentage coverage with differen number of interest points. The resolution of testing image is 920*620.

3. EXPERIMENTS

We show the experimental results in this section. First we calculate how many interest points are necessary. Then we show the reconstruction results of the methods proposed.

3.1. Scene coverage

The number of matched interest points and line in an image may be very large. Consider we have n pairs of points and m pairs of lines. The total number of virtual lines will be n^2, and the possible intersections will be n^2m. This number even can be larger than the total number of pixels in the image, so we do not need to take all the possible intersections into account. In this experiment, we show that only few number of points are necessary.

We assume the interest points are randomly distributed in the images, so we take n random points in the image , and check all the virtual lines jointing each pair of points. We also assume the width of a virtual line is 1 pixel and a pixel is covered by the interest points if it is located on any virtual line. We perform this experiment for 10 times and Fig. 3 shows the average percentage coverage against the total number of interest points. We can get that almost 99% of an image is covered with only about 100 points. In fact, even if only less than 50% of points on the lines are shown, we can still clearly identify the lines. Because not all the intersections can be used to represent the lines, in the following experiments, we take 200 matched interest points in each pair of image.

3.2. 3D reconstruction results

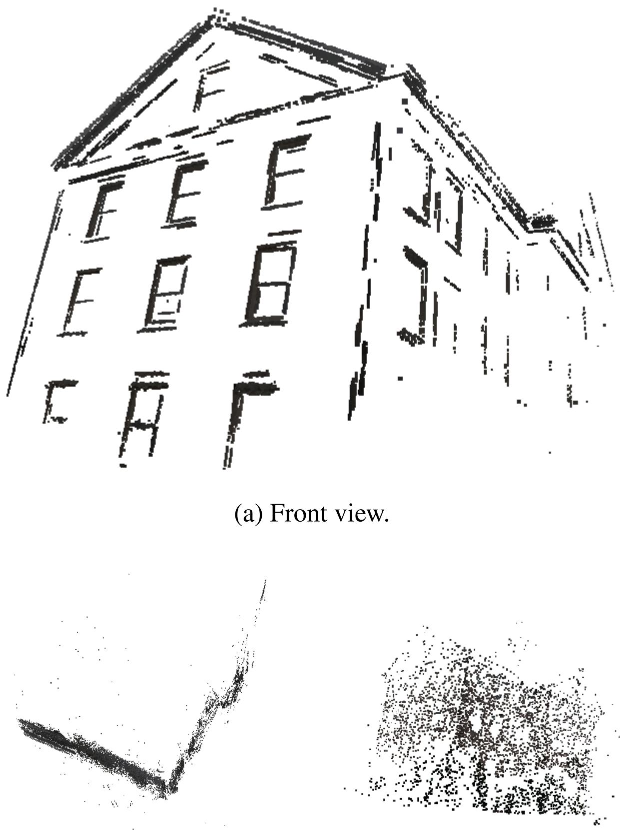
Fig. 4 shows the 6 images used to test our method, the images are are taken from[9]. The original points are matched by SIFT and the lines are matched by method in[6], and the 3D model is generated through VisualSFM[2, 16]. Fig. 5(a)shows the front view and Fig. 5(b)shows the top view of the 3D point cloud reconstructed by our method, and Fig. 5(c) gives the point cloud reconstructed from SIFT points. It is obvious that our approach can provide more structural information and semantically more meaningful 3D models than the traditional point-based method. Moreover, as we only take 200 interest points to calculate the intersections, the translation from lines to intersections is very fast.

Fig. 4. The testing images used in our experiments.

However, as we can see from the Fig. 4, there are many trees in the scene where no lines can be matched, so the line-based method failed to get any structures in that area. The SIFT based model get some points there although the trees are hardly to identify from the point cloud.

4. CONCLUSION

In this paper, we provide a simple and effective new approach to build 3D models. We take the intersections of virtual lines and real lines rather than the original points or lines to build 3D models. This strategy can be applied to existing state-of-the-art points based methods while maintain the structural information of line-based models. We take only very few matched points and use a simple invariant to filter out wrong intersection correspondences thus very fast and effective. Experiments show that our approach can get 3D point cloud with more meaningful structural information than traditional point-based method.



(a) Front view.

(b) Top view.

(c) SIFT model.

Fig. 5. 3D point cloud based on our method and SIFT points, we can see that our method can provide more meaningful result.(a) 3D point cloud of our method from the front view. (b) 3D point cloud of our method from the top view.(c)3D point cloud of SIFT points.

5. ACKNOWLEDGEMENTS

This work is partially supported by the Natural Science Foundation of China under grant Nos. 61402077, 61572096, 61272371, 51579035, 61632006 and 61572105.

6. REFERENCES

[1] Yasutaka Furukawa and Jean Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 8, pp. 1362–76, 2010.

[2] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M Seitz, "Multicore bundle adjustment," pp. 3057–3064, 2011.

[3] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] Y Furukawa, B Curless, S. M Seitz, and R Szeliski, "Towards internet-scale multi-view stereo," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, Ca, Usa, 13-18 June*, 2010, pp. 1434–1441.

[5] Mathias Rothermel, Konrad Wenzel, Dieter Fritsch, and Norbert Haala, "Sure: Photogrammetric surface reconstruction from imagery," in *LC3D Workshop*, 2012.

[6] Qi Jia, Xinkai Gao, Xin Fan, Zhongxuan Luo, Haojie Li, and Ziyao Chen, "Novel coplanar line-points invariants for robust line matching across views," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, 2016, pp. 599–611.

[7] Srikumar Ramalingam, Michel Antunes, Daniel Snow, Gim Hee Lee, and Sudeep Pillai, "Line-sweep: Cross-ratio for wide-baseline matching and 3d reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1238–1246.

[8] Branislav Micusik and Horst Wildenauer, "Structure from motion with line segments under relaxed endpoint constraints," *International Journal of Computer Vision*, vol. 1, pp. 1–15, 2016.

[9] Manuel Hofer, Michael Maurer, and Horst Bischof, "Efficient 3d scene abstraction using line segments," *Computer Vision & Image Understanding*, 2016.

[10] Zhengyou Zhang, "Estimating motion and structure from correspondences of line segments between two perspective images," vol. 17, no. 12, pp. 1129–1139, 1995.

[11] Adrien Bartoli and Peter Sturm, "Structure-from-motion using lines: representation, triangulation, and bundle adjustment," *Computer Vision & Image Understanding*, vol. 100, no. 3, pp. 416–441, 2005.

[12] Chelhwon Kim and Roberto Manduchi, "Planar structures from line correspondences in a manhattan world," in *ACCV*, 2014, pp. 509–524.

[13] Carolina Raposo, Michel Antunes, and Joao P. Barreto, "Piecewise-planar stereoscan:structure and motion from plane primitives," *Lecture Notes in Computer Science*, vol. 8690, pp. 48–63, 2014.

[14] Alexander Vakhitov, Jan Funke, and Francesc Moreno-Noguer, *Accurate and Linear Time Pose Estimation from Points and Lines*, pp. 583–599, Springer International Publishing, Cham, 2016.

[15] Bin Fan, Fuchao Wu, and Zhanyi Hu, "Robust line matching through line–point invariants," *Pattern Recognition*, vol. 45, no. 2, pp. 794–805, 2012.

[16] Changchang Wu, "Towards linear-time incremental structure from motion," in *International Conference on 3dv-Conference*, 2013, pp. 127–134.