URBAN STRUCTURE ESTIMATION USING PARALLEL AND ORTHOGONAL LINES

An Undergraduate Research Scholars Thesis

by

RUI LIU

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Research Advisor: Dr. Dezhen Song

May 2015

Major: Computer Engineering

TABLE OF CONTENTS

Page

ABSTRACT

Urban Structure Estimation Using Parallel and Orthogonal Lines. (May 2015)

Rui Liu
Electrical and Computer Engineering Department
Texas A&M University

Research Advisor: Dr. Dezhen Song
Computer and Electrical Engineering Department

In urban areas, buildings will block the line-of-sight of GPS signals while mobile robots are navigating through an environment which will result in localization errors of the robot. Robot localization using camera sensor is another way for exploring the surrounding area. Depth uncertainty in camera images could affect the accuracy of the estimated structure of the surrounding environment. Man-made structures contain many parallel and orthogonal lines. In this study, I use parallel and orthogonal lines in the scene to improve the estimation of the environment. The proposed method improves structure estimation compared with the standard reprojection error method.

ACKNOWLEDGEMENTS

I would like to thank Joseph Lee and Dr. Dezhen Song for their tremendous help for this research project. Their teaching and support motivate me to make continuous development to this thesis and my personal improvement.

SECTION I

INTRODUCTION

In vision-based robot navigation, the robot keeps updating its location and map by measuring the distance from itself to surrounding objects. In man-made environment, building reconstruction based on multiple images can be used to help robots to locate its position and construct its 3D map for its surrounding environment.

Man-made objects usually have many line segments. If properly used, line segments can be used to help the building reconstruction process. Urban buildings generally have two geometric properties: line segments on a plane can be parallel or orthogonal. The following figures were taken in Texas A&M University campus; there are many orthogonal and parallel line segments on different building planes. This study introduces the method used to improve 3D structure reconstruction by using orthogonality and parallelism of coplanar line segments.

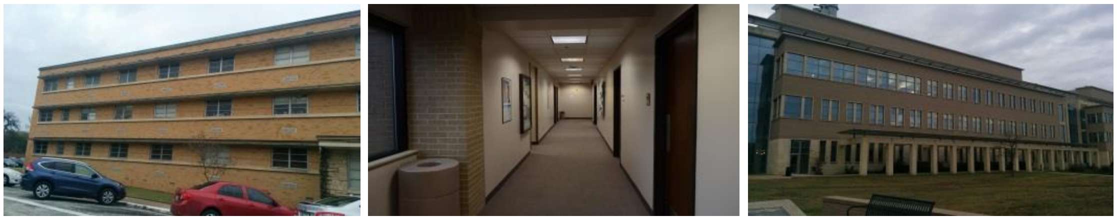


Fig. 1 Man-made environments have many parallel and orthogonal lines

SECTION II

RELATED WORK

Scene recovery based on line segments is a popular topic in computer vision. Previous work has been done to study reconstruction problems by using lines. In [1], a method that uses line features from image correspondences of multiple views to do line-based scene reconstruction by linearizing the Plücker constraint is proposed. Roberto Manduchi in [2] develops an algorithm to reconstruct the planar structures and relative camera pose in a Manhattan world by using matched lines from two images taken from distinct viewpoints. Both reprojection error method and line segments based method are used in [3] to explore a new algorithm to do planar building facade segmentation. Ali Elqursh and Ahmed Elgammal in [4] proposed a new algorithm to calibrated camera relative pose estimation by using lines, especially orthogonal and parallel line segments in physical indoor and urban environments. Having all present features in distinct views of a same scene been accurately detected and exactly matched can be very difficult, which finally make reconstruction be a tough task.

SECTION III

PROBLEM STATEMENT

1. Notations and Assumptions

Notations:

- Let I and I' represent a pair of images from different viewpoints. Prime symbol $(')$ represents the entities of the second image. I denote a pair of lines by $\mathbf{l} = (a, b, c)$ and $\mathbf{l'} = (a', b', c')$.
- Let $\mathbf{P}$ and $\mathbf{P}'$ represent the camera matrix. Let $\mathbf{y}$ represent the intersection point of two lines. Let $\mathbf{x}_i$ represent the i-th measured image coordinate in image I. Let $\hat{\mathbf{x}}_i$ represent an estimated point for $\mathbf{x}_i$, and let $\mathbf{x}'_i$ represent the correspondence point in image I'.

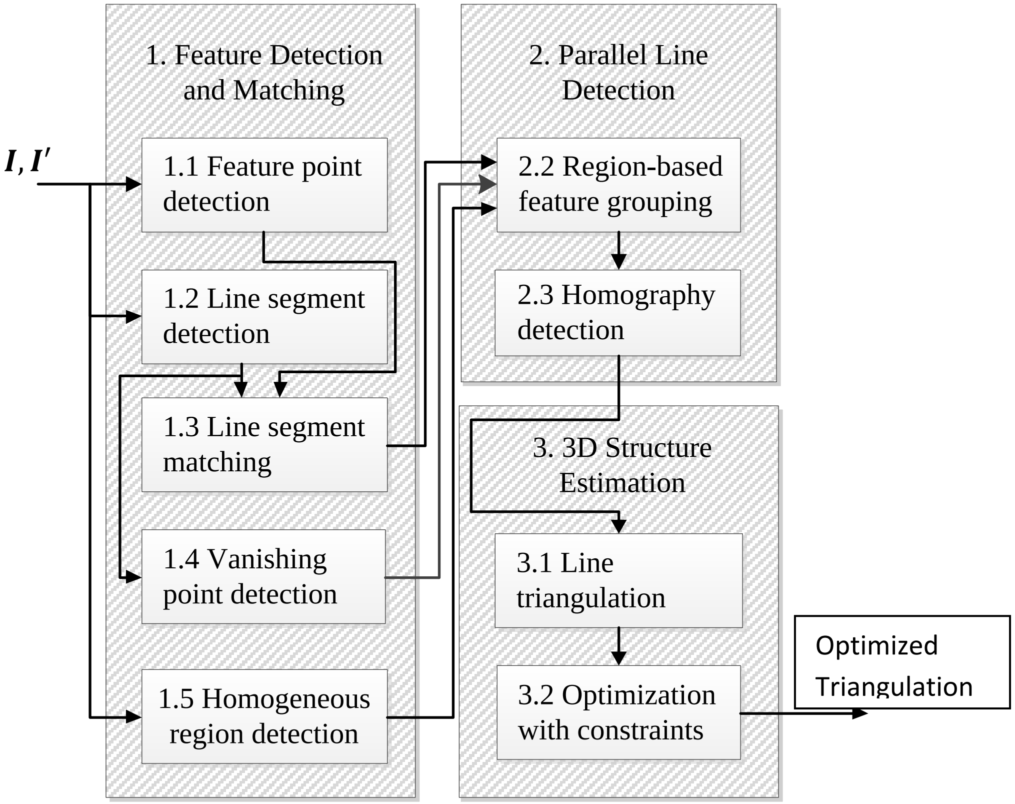
2. Problem Definition

Given two image views I and I', detect matched parallel and orthogonal line segments and estimate the 3D structure of the scene.

SECTION IV

SYSTEM DESIGN

Fig. 2 shows a high level overview of our system, which consist of three main blocks: (1) feature detection and matching, (2) Parallel Line Detection, and (3) 3D structure estimation.



I, I'
1. Feature Detection and Matching
1.1 Feature point detection
1.2 Line segment detection
1.3 Line segment matching
1.4 Vanishing point detection
1.5 Homogeneous region detection
2. Parallel Line Detection
2.2 Region-based feature grouping
2.3 Homography detection
3. 3D Structure Estimation
3.1 Line triangulation
3.2 Optimization with constraints
Optimized Triangulation

Fig.2 System diagram

Given two input images I and I', Fig. 2 Block 1 performs SIFT and line segment detection and matching. After matching line segments from Block 1, Block 2 performs initial coplanar parallel line detection with image segmentation. The homography computed from point correspondence is used to identify coplanar line segments. From the grouped coplanar parallel line information, Block 3 performs line triangulation and applies the geometric constraints.

SECTION V

METHODS

1. Feature Detection and Matching

We choose scale-invariant feature transform (SIFT) points and line segments as the features here and we use existing software to accomplish these tasks.

1. Feature Point Detection

SIFT, designed by David Lowe in [5], is the algorithm used to detect and describe local features in images. It can effectively extract many featured points from images. Here is an example by using SIFT for two image views with matched feature points shown.

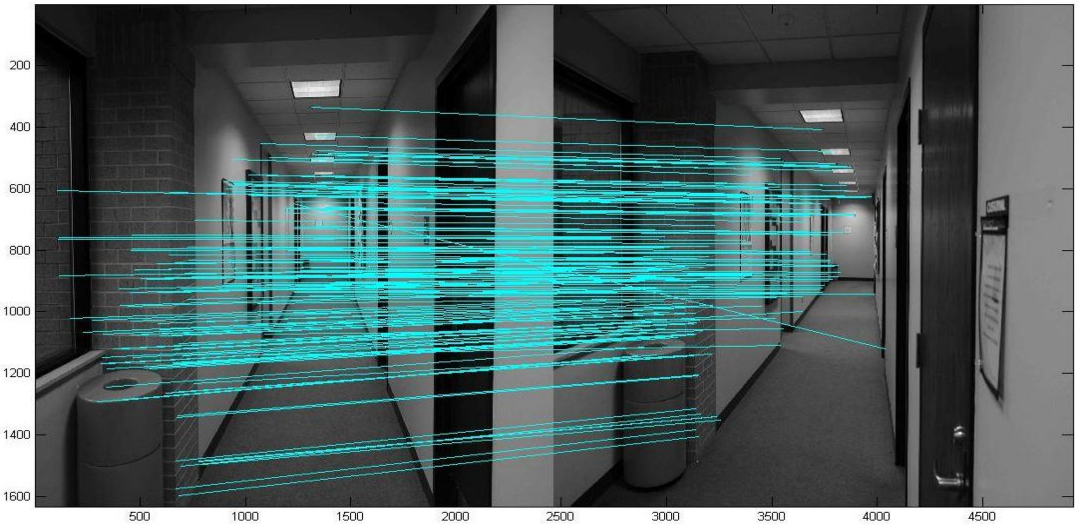


Fig. 3 182 matched SIFT points are shown

2. Line Segment Detection

Based on the SIFT, line segment as another local feature can be detected by using [6]. Each line segment is expressed by recording two end points. Fig. 4 shows an example with detected line segments plotted.

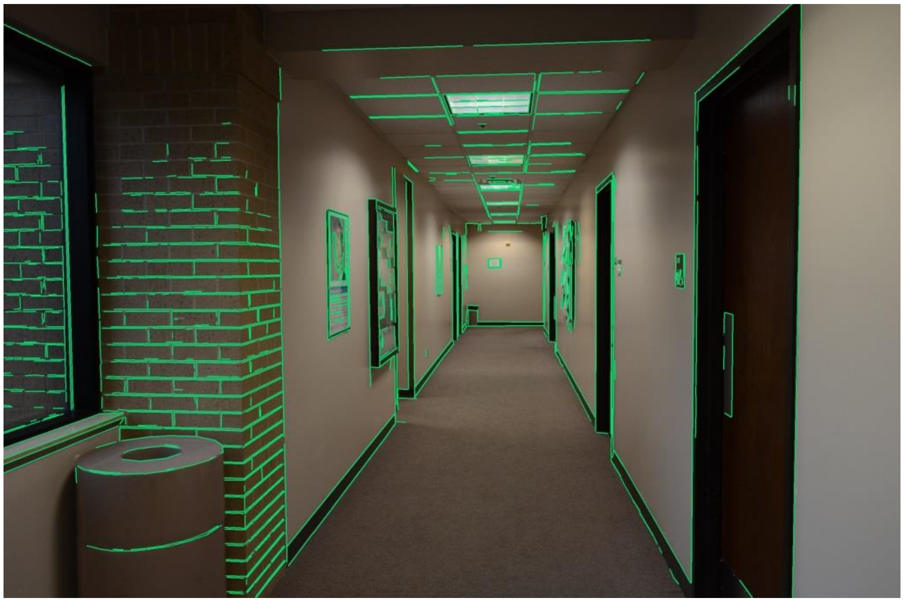


Fig.4 1038 line segments are detected and plotted

3. Line Segment Matching

After detecting line segments for two image views, they are matched by using the method proposed in [7]. This method uses feature point matches in Section 1.1 to match line segments. Fig. 5 shows an example with matched line segments plotted with same index number and color.

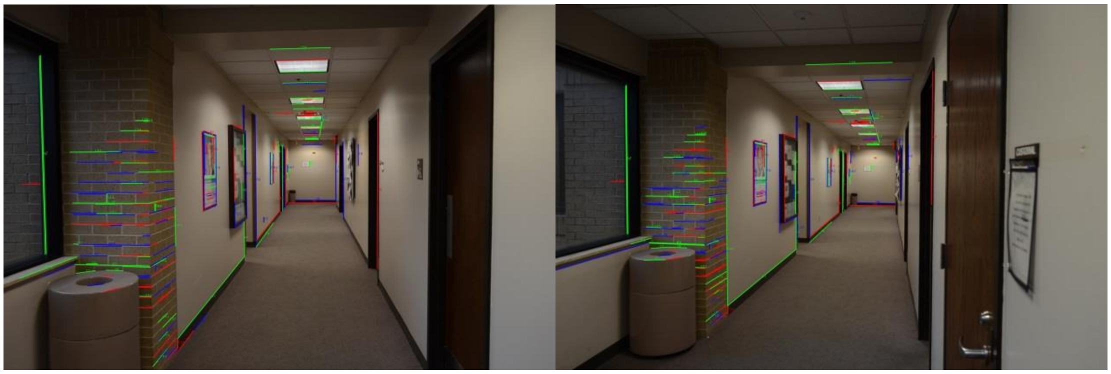


Fig. 5 175 line segments are matched for two image views for same scene

4. Homogeneous Region Detection

Planes usually have homogeneous appearance. The optimization we proposed will be applied to coplanar line segments on the same physical planes. Thus it is important to detect the physical planes in a given scene. We use the image segmentation method proposed in [8]. Fig. 6 is a result for the two images view for same scene.

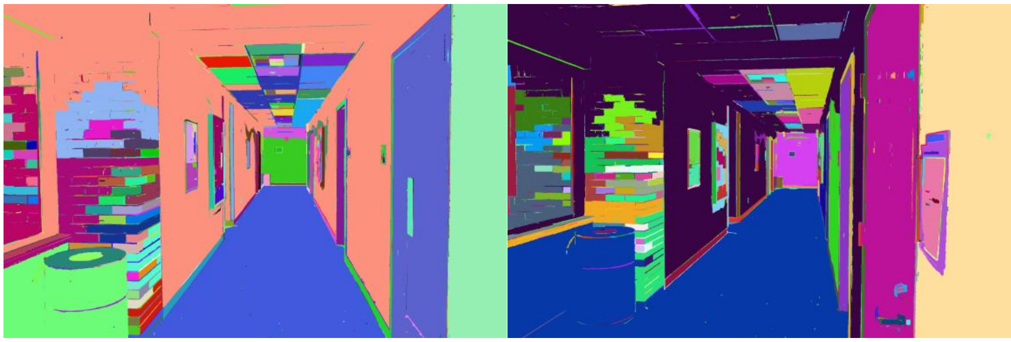


Fig. 6 Image segmentation using method [8]

5. Vanishing Point Detection

We can identify if two line segments as parallel if they pass the same point at infinity. For each line **l** and **l′** **(**each line is expressed with 3 element column vector**)**, the intersection point can be computed by

$$\mathbf{y} = \mathbf{l} \times \mathbf{l}' = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \mathrm{a} & \mathrm{b} & \mathrm{c} \\ \mathrm{a}' & \mathrm{b}' & \mathrm{c}' \end{bmatrix} \qquad \mathbf{(1)}$$

If **y** is near the vanishing point we determine **l** and **l′** parallel. Vanishing point detection is the first step we do to explore coplanar parallel line segments. Vanishing point can be detected by using [9]. Fig. 7 is an example with vanishing point plotted. Parallel line segments are plotted with same color.

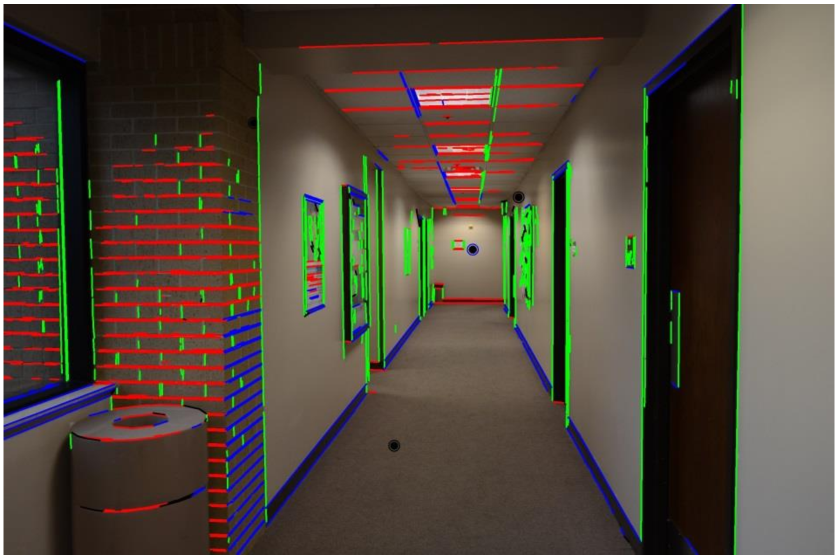


Fig. 7 4 Vanishing points are found and plotted with method [9]

2. Parallel Line Detection

We have two steps for the coplanar parallel line segments detection as shown in the system diagram.

6. Region-based feature grouping

   We first find the boundary for each single color area by finding their contours. Fig. 8 shows the homogeneous region contours.

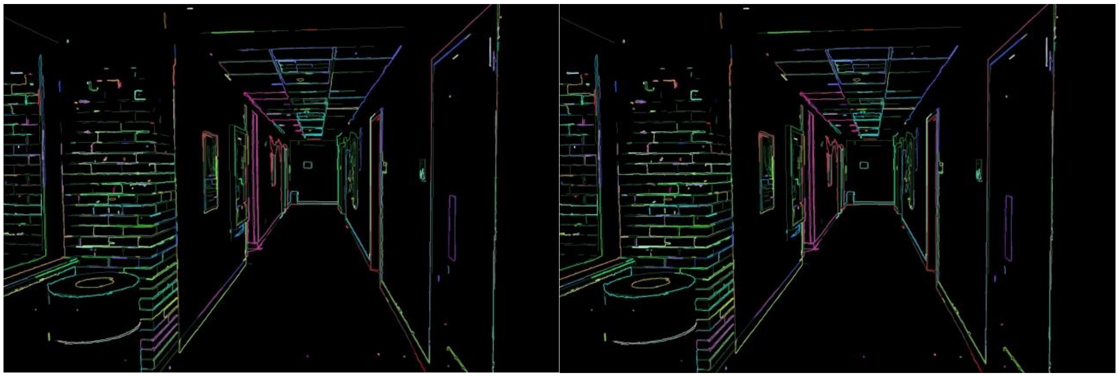


Fig. 8 Contours for each homogeneous region is shown

We set a criterion for grouping coplanar line segments on each homogeneous region. We use two criterions: (a) those line segments surrounding a same contour are in the group represented by the contour and (b) a line segment can belong to multiple contours.

7. Homography Detection

   A single color covered area may not represent a single physical plane. If four points are on the same plane, and any three of them are not collinear, then we can compute a homograph matrix H, which represents the transformation from point $\mathbf{x}_i$ on image I to point $\mathbf{x}'_i$ on image I'. By using this theory, we can verify coplanar line segments. For a sequence of four matched point pairs of two image views, Random sample consensus algorithm (RANSAC) is used to compute the homography. Then, we remove the detected homography plane and continue to use RANSAC for the rest.

3. 3D Structure Estimation

To estimate the 3D structure, we follow two steps: (1) initialization using line triangulation and (2) using orthogonality, parallelism, and the reprojection error constraint to refine the structure.

1. Line Triangulation

   In two views, a line is the intersection of two planes, where each plane is generated with the camera center and two points on the image plane of the camera center. Given the camera matrix $\mathbf{P}$ and the two image lines $\mathbf{l}$ and $\mathbf{l}'$, we use the following span representation to represent the line segment L generated from the two intersected planes according to the formula provided from [10]. Fig. 9 from [10] illustrates this theory.

$$L = \begin{bmatrix} \mathbf{l}^{\mathrm{T}}\mathrm{P} \\ \mathbf{l}'^{\mathrm{T}}\mathrm{P}' \end{bmatrix} \qquad \textbf{(2)}$$

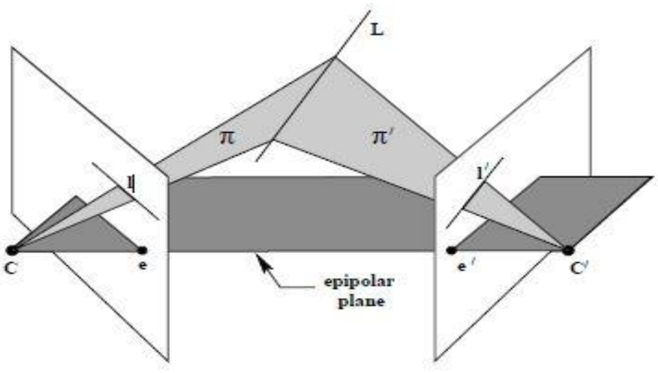


Fig. 9 Line triangulation

Planes are defined as $\pi = \mathrm{P}^{\mathrm{T}}\mathbf{l}$ and $\pi' = \mathrm{P}'^{\mathrm{T}}\mathbf{l}'$, P and P′ represent the camera matrixes for the two cameras.

2. Optimization with geometric constraints

Fig. 10 shows how we apply the parallelism constraint to two line segments. Two parallel lines in 3D space intersect at ideal point on a plane located infinitely from the camera. Back-projection from that intersection on infinite plane to image plane gives us the vanishing point. We minimize the distance from vanishing point to the measured one.

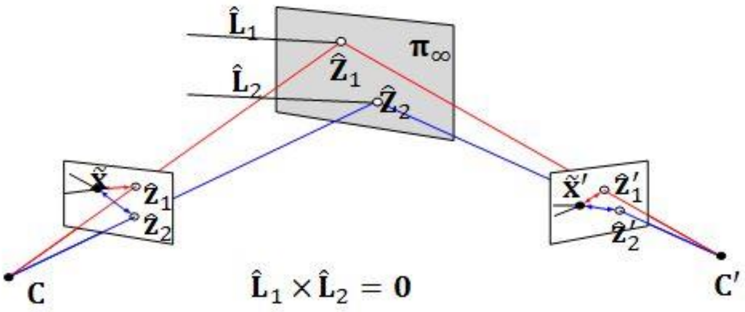


Fig. 10 The parallelism constraint

3. Triangulation Experiments

We applied most of the previous theories and methods to triangulate the corridor scene shown in figure 3. The short line segments make a lot noise for the whole reconstruction result, thus we removed the 124 short line segments and mismatched line segments. Figure 11 is shows the colored line segment we want to triangulate. After figure 11, the following pictures show how our program plots the result line by line, and how the noise comes to the triangulation.

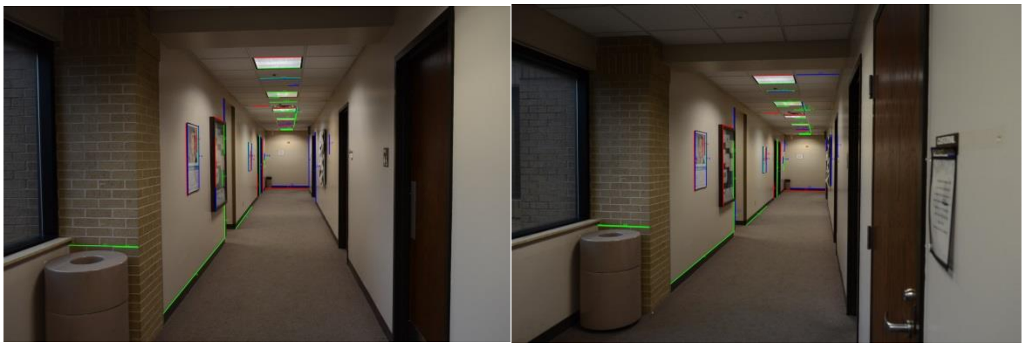


Fig. 11 The plotted 51 colored line segments will be triangulated

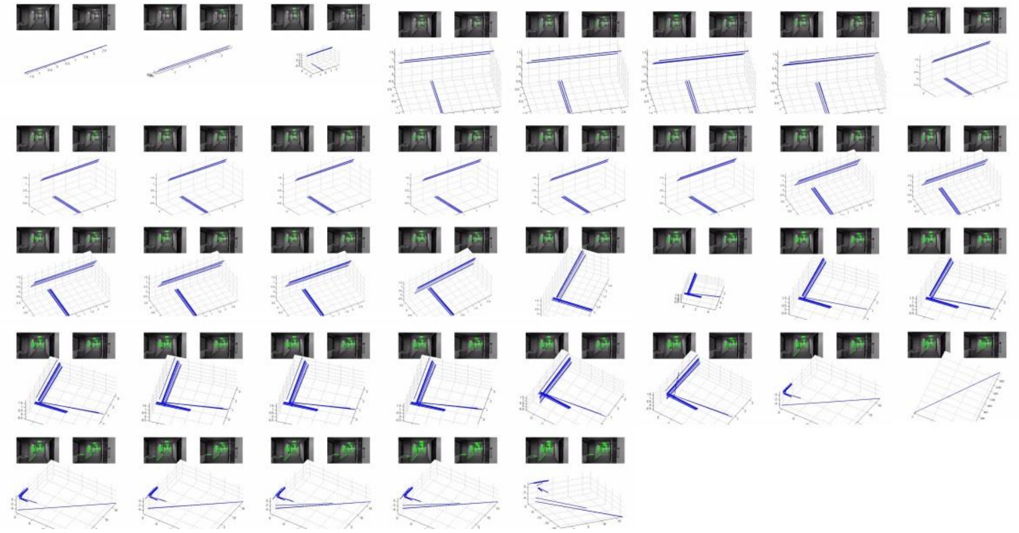


Fig. 12 Process of drawing each triangulated line and noise appearance

As we can see, the result shows good orthogonality and parallelism at the beginning, however, noise comes in when we take in more line segments. The reason for abnormal result can be due to the bad selection of the scene (we get mismatches for line segments and points) or the failure of visualization software (MATLAB).

SECTION VI

CONCLUSION AND FUTURE WORK

A lot previous work has been done in line segments related topics, but there are still a lot unsolved issues as well as potential improvements like exact line segment matching. This project is aimed at improving line triangulation by enforcing orthogonality and parallelism, however, only the triangulation part has been done and the result tells us that triangulation we have done here should be improved to make the basic scene be seen. Half-year long research for this topic is pressing, but we do obtain a lot related knowledge in computer vision research area. This knowledge could be very useful for future research. For the next several months, continuous work will be done on applying orthogonality and parallelism constraints to improve the performance of line triangulation. Meanwhile, we will collect more ground truth data sets to test the performance of the optimized triangulation. We will consider using new visualization software (C++ and OpenGL) and programming language (C++ and OpenCV) to do the visualization job.

REFERENCES

[1]. Bartoli, A.; Sturm, P., "Multiple-view structure and motion from line correspondences," *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on,* vol., no., pp.207,212 vol.1, 13-16 Oct. 2003

[2]. Kim, Chellhwon; & Manduchi, Roberto, "*Planar Structures from Line Correspondences in a Manhattan World.*" Computer Vision, UC Santa Cruz. 2014.

[3]. Joseph Lee, Yan Lu, and Dezhen Song, "*Planar Building Facade Segmentation and Mapping Using Appearance and Geometric Constraints",* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, US, Sept., 2014

[4]. Elqursh, A.; Elgammal, A., "Line-based relative pose estimation," *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* , vol., no., pp.3049,3056, 20-25 June 2011.

[5]. David G. Lowe, **"**Distinctive image features from scale-invariant keypoints,**"** *International Journal of Computer Vision,* 60, pp. 91-110, February 2004.

[6]. R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "*LSD: a line segment detector*," Image Processing On Line, 2012.

[7]. B. Fan, F. Wu, and Z. Hu, "*Robust line matching through line-point invariants*" Pattern Recognition, vol. 45, no. 2, pp. 794-805, Feb. 2012.

[8]. Grundmann, M.; Kwatra, V.; Mei Han; Essa, I., "*Efficient hierarchical graph-based video segmentation,*" *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* , vol., no., pp.2141,2148, 13-18 June 2010.

[9]. M. Nieto and L. Salgado, "Real-time robust estimation of vanishing points through nonlinear optimization," in IS&T/SPIE Int. Conf. on Real-Time Image and Video Processing, SPIE vol.7724, 772402, 2010. (DOI 10.1117/12.854592)

[10]. R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, 2004.