SEMI-AUTOMATIC 3D RECONSTRUCTION OF PIECEWISE PLANAR BUILDING MODELS FROM SINGLE IMAGE

Chen Feng, Graduate Research Assistant,
School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China
Department of Civil and Environmental Engineering, University of Michigan, 2350 Hayward St., Ann Arbor, MI 48109, USA
simba.forrest@gmail.com

Fei Deng, Associate Professor,
School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China
fdeng@sgg.whu.edu.cn

Vineet R. Kamat, Associate Professor
Department of Civil and Environmental Engineering, University of Michigan, 2350 Hayward St., Ann Arbor, MI 48109, USA
vkamat@umich.edu

ABSTRACT: *This paper presents a novel algorithm that enables the semi-automatic reconstruction of man-made structures (e.g. buildings) into piecewise planar 3D models from a single image, allowing the models to be readily used for data acquisition in 3D GIS or in other virtual or augmented reality applications. Contrary to traditional labor intensive but accurate Single View Reconstruction (SVR) solutions that are based purely on geometric constraints, and recent fully automatic albeit low-accuracy SVR algorithms that are based on statistical inference, the presented method achieves a compromise between speed and accuracy, leading to less user input and acceptable visual effects compared to prior approaches. Most of the user input required in the presented approach is a line drawing that represents an outline of the building to be reconstructed. Using this input, the developed method takes advantage of a newly proposed Vanishing Point (VP) detection algorithm that can simultaneously estimate multiple VPs in an image. With those VPs, the normal direction of planes which are projected onto the image plane as polygons in the line drawing can be automatically calculated. Following this step, a linear system similar to traditional SVR solutions can be used to achieve 3D reconstruction. Experiments that demonstrate the efficacy and visual effects of the developed method are also described.*

KEYWORDS*: Computer Vision, J-linkage, Vanishing Point, Single View Reconstruction, Image Based Modeling, Virtual Reality, Augmented Reality*

1. INTRODUCTION

Single view reconstruction (SVR), as one of the image based modeling (IBM) techniques, has been extensively studied from both the side of computer graphics and computer vision. It could help us in the situation when we want to recover a 3D scene while having only one image at hand, which means the traditional multiple view reconstruction approaches in either close-range photogrammetry or computer vision cannot be applied.

In the past, the main stream of SVR algorithms focused purely on the geometric structural information that one can infer from a single view of the scene as apriori knowledge. The key idea of these approaches is that through this knowledge provided by users, a scene’s 3D structure can be calculated by geometry theorems. "Tour into the picture (TIP)", proposed by computer graphic researchers (Youichi et al., 1997), is probably the earliest solution taking advantage of vanishing point (VP) to recover 3D structures, though the assumption that the picture has only one VP limits its application. Almost at the same time, computer vision researchers from University of Oxford conducted a series of research works on single view metrology (Criminisi et al., 2000, Liebowitz and Zisserman, 1999), introducing the theory of projective geometry which laid a solid mathematical foundation for SVR. Later, researchers from INRIA proposed a SVR algorithm based on user-provided constraints such as coplanarity and perpendicularity to form a linear system (Sturm and Maybank, 1999). Compared with other similar methods (Grossmann and Santos-Victor, 2005, van den Heuvel, 1998), Sturm's algorithm is regarded as one of the most flexible ones and will be the basis of SVR method in this paper.

Recently, a group of computer vision researchers have shifted their attention from geometry to machine learning to develop new SVR algorithms. Arguing the traditional SVR to be labor-intensive, Hoiem et al. from Carnegie

Mellon University proposed a fully automatic algorithm that folds the image segments into a pop-up model (Hoiem et al., 2005). Similar algorithms include dynamic Bayesian network SVR of indoor image (Delage et al., 2006) and Markov Random Field (MRF) SVR (Delage et al., 2007, Saxena et al., 2006). Although these algorithms can achieve full automation, their reconstructed 3D model's visual effects still need to be improved for virtual reality or augmented reality applications.

In this research, inspired by the idea of utilizing machine learning algorithms to deduce some of this geometric structural information so as to reduce a part of labor burden on users, we integrate a newly proposed line segment detector(LSD) (Grompone et al., 2010) and a robust multiple structures estimator (Toldo and Fusiello, 2008) into Sturm's SVR algorithm. It will first be compared with traditional methods in section 2 and then be explained in detail in section 3. Some of the experimental results are given in section 4, followed by summarization of this paper's contributions and a conclusion.

2. OVERVIEW

2.1 Traditional SVR

Figure 1 presents the general schema of traditional SVR algorithms (*FIG. 1*).

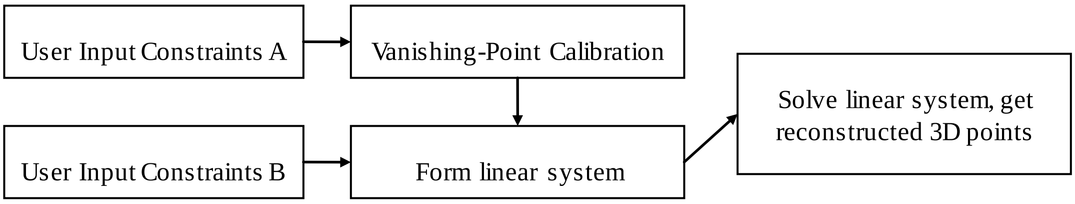
User Input Constraints A
Vanishing-Point Calibration
User Input Constraints B
Form linear system
Solve linear system, get reconstructed 3D points

FIG. 1: General schema of traditional SVR algorithms

Constraints A provided by users are usually parallel constraints, i.e. which image line segments' 3d object space correspondences are with the same direction. In fact, this process equals to a manual classification on line segments - line segments whose 3D correspondences with the same direction are grouped into a same class. Each group of line segments' extended lines should intersect (ideally if without measurement errors) at the same point in the image plane, i.e. the vanishing point. If three vanishing points are found whose corresponding 3D directions are perpendicular to each other, the camera's principle point and focal length can then be calculated (Caprile and Torre, 1990).

Constraints B are mainly coplanar constraints in Sturm's methods. By specifying which image points' 3D correspondences lie on the same 3D plane, whose normal direction is also specified through combination of any two vanishing points, a linear system could then be formed and solved. The solution of that linear system contains each image point's depth,which also means the 3D structure of the scene.

2.2 Semi-auto SVR

Compared with traditional SVR approaches, our method tries to minimize the user input by taking advantage of a multiple structures estimator called j-linkage (*FIG. 2*).

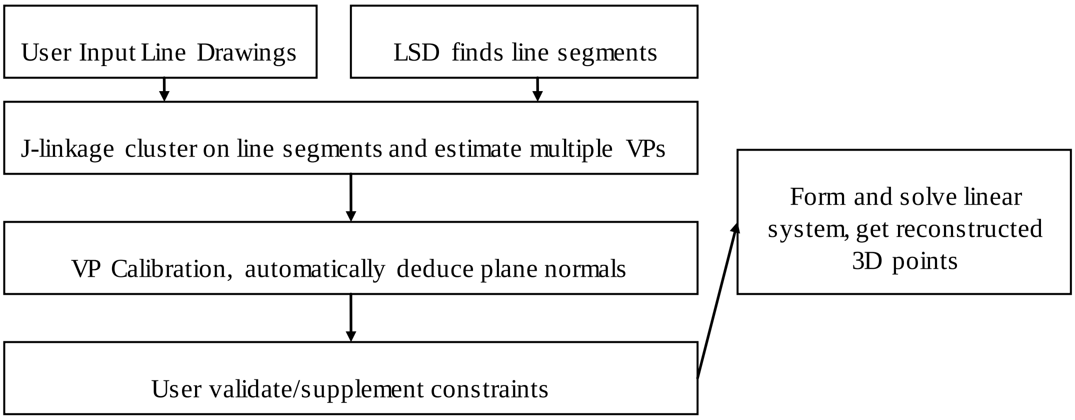
User Input Line Drawings
LSD finds line segments
J-linkage cluster on line segments and estimate multiple VPs
VP Calibration, automatically deduce plane normals
User validate/supplement constraints
Form and solve linear system, get reconstructed 3D points

FIG. 2 Schema of our semi-automatic SVR algorithms

As we can see from *FIG. 2*, the user input constraints A and B in *FIG. 1* are replaced with "user input line drawings" and "user validate/supplement constraints". This means the parallel constraints and 3D plane's normal directions will be automatically deduced instead of being specified by users. Thus users will only need to sketch out the building to be reconstructed from a single image, leave all the computation to the algorithms, then check and validate the constraints reasoned by the algorithms, and supplement other constraints if necessary. This will then allow the reconstructed 3D model to be manipulated on the computer.

2.3 Global Assumptions

Before we explain our semi-automatic SVR algorithm in detail, there are several global assumptions that must be addressed:

- *No radial distortion.* The image used in our algorithm should already be corrected for radial distortion, or the radial distortion parameter must be small enough to be ignored. Generally, this assumption can be easily met, as long as we do not use special lens (such as wide-angle lens or fish-eye lens) and the building to be reconstructed lies in the middle of the image.

- *Camera's principle point is located at the image center, its aspect ratio is 1 and skew factor is 0.* This assumption means the calibration matrix of the camera has the form (with known image width W and height H, while focal length f as the only unknown parameter to be calibrated)

$$\mathbf{K} = \begin{bmatrix} f & 0 & W/2 \\ 0 & f & H/2 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

  Although the assumption that principle point locates at the center of the image seems to be too strong, considering the manufacturing quality of digital consumer cameras, experiments have shown that this error will not induce much effect on the reconstructed model's visual effects.

- *Camera coordinate system is our world coordinate system.* This means we ignore all the six exterior parameters, i.e. the translation and rotation of the camera, so the projection matrix of the camera will be of the following form:

$$\mathbf{P} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}].$$

  Also our reconstructed 3D point is up to a scale factor, meaning we are only concerned about its shape but ignore its size. In many virtual reality applications, this is sufficient.

- *Manhattan World Assumption (Coughlan and Yuille, 1999).* This assumption is saying that a natural reference frame is given in most indoor and outdoor city scenes, for they are based on a Cartesian coordinate system. Under this assumption, we could use vanishing-point calibration algorithm to recover the focal length of the camera.

3. ALGORITHM MODULES

Our proposed SVR algorithm consists of six sub-procedures, as can be seen from *FIG. 2*. Each module will be described in detail in the following sections.

3.1 User Input

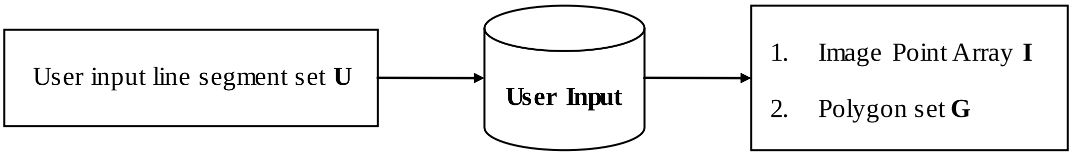

FIG. 3 Input and output(I/O) of user Input module

Most of the user interactions in our method are handled in this module. It enables users to sketch out the skeleton of the building with a set of line segments $\mathbf{U} = \{l_i = (x_i^s, y_i^s; x_i^e, y_i^e), i = 1, 2, \cdots, N_u\}$, in which a line segment l is represented by its two end points $(x^s, y^s), (x^e, y^e)$.

The data structure in this module enables applying computational geometry algorithms, in order to output an image point array $\mathbf{I} = \{p_i = (x_i, y_i) \mid \forall i \neq j, p_i \neq p_j\}$, whose elements all come from end points of line segments in $\mathbf{U}$, and a set of polygons $\mathbf{G}$, in which each polygon is represented by an ordered index array of image point array $\mathbf{I}$.

To reconstruct the output image point array's 3D correspondence will be our SVR method's final objective. Along with the topological information stored in polygon set $\mathbf{G}$, one can easily get the building's 3D model.

3.2 Line Segment Detector

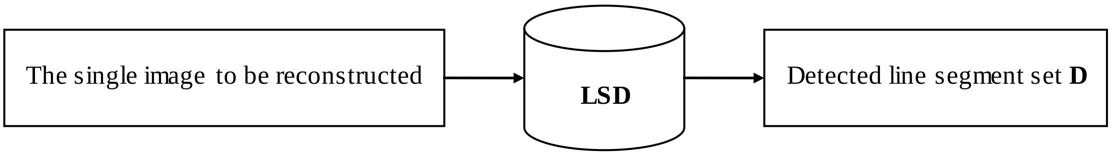
The single image to be reconstructed
LSD
Detected line segment set **D**

FIG. 4 I/O of LSD module

Similar to line segment set $\mathbf{U}$ in *FIG. 3*, LSD module's output $\mathbf{D}$ are also line segments represented by end points.

However, different from the traditional edge detection method which first uses Canny edge detector followed by a series of complicated post-processing (Tardif, 2009), the newly proposed Line Segment Detector (LSD) (Grompone et al., 2010) provides us a fast, simple and easy-to-use interface which also gives accurate results yet requires no parameter tuning.

3.3 J-linkage

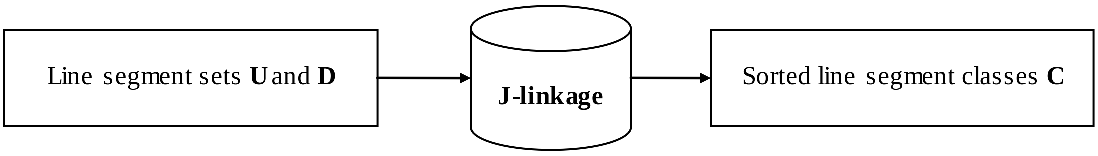
Line segment sets **U** and **D**
J-linkage
Sorted line segment classes **C**

FIG. 5 I/O of J-linkage module

J-linkage module wraps a recently proposed robust multiple structures estimator (Toldo and Fusiello, 2008), taking as input line segment sets $\mathbf{U}$ and $\mathbf{D}$ in the first two modules, outputting sorted line segment classes $\mathbf{C} = (C_1, C_2, \cdots, C_{Nc}), \forall i < j, |C_i| > |C_j|$, a ordered array of line segment sets sorted by their sizes, in which each element C_i is a set of line segments coming from $\mathbf{U}$ and $\mathbf{D}$ where the operator $|C_i|$ represents the number of elements of the set. Ideally, each class of line segments should correspond to a vanishing point and hence a 3D direction in object space.

The j-linkage estimator was carefully designed to robustly estimate models with multiple instances in a set of data points. This leads us to the Hough Transform. However, quantization of the parameter space, the basis of Hough Transform, will inevitably cause many of its shortcomings such as inaccuracy and the choice of parameterization of models. Enlightened by a popular parameter estimation approach in computer vision, RANSAC (Fischler and Bolles, 1981), and the conceptual representation from pattern recognition, the j-linkage estimator also needs no parameter tuning. Besides estimation of multiple model instances, it could classify all data points according to the best model instance they fit, which will be of great use in our normal deduction module.

In our algorithm, the "data points" for J-linkage are line segments, from both user drawn and LSD detection, and the "model instances" are vanishing points. Line segments through different vanishing points are classified into different line segment groups. In order to apply j-linkage estimator, three functions have to be defined:

1. function W that solves model parameters from minimal number of data points

2. function F that estimates the distance (or fitness) of a given model and a data point

3. distance function D of a pair of data points

For function W, one can easily figure out that the minimal number of data points (i.e. line segments) needed to solve the model parameters (vanishing point's coordinate in image plane) is two. By using homogeneous coordinates, it can be written as (operator $\times$ means cross product of two 3D vectors)

$$W(l_i, l_j) = \frac{l_i \times l_j}{\|l_i \times l_j\|}, l_k = (x_k^s, y_k^s, 1) \times (x_k^e, y_k^e, 1), k = i, j. \tag{2}$$

Function F, as the discussion in literature (Tardif, 2009), could be well approximated by the distance of the line segment's end point and the line through the vanishing point and the mid-point of the line segment (*FIG 6*), as below (v is homogeneous coordinate of a vanishing point, l is a line segment represented by two end points and $dist$ is the distance function of 2d point to line):

$$F(v, l) = dist(m, (x^s, y^s)), m = v \times (\frac{x^s + x^e}{2}, \frac{y^s + y^e}{2}, 1). \tag{3}$$

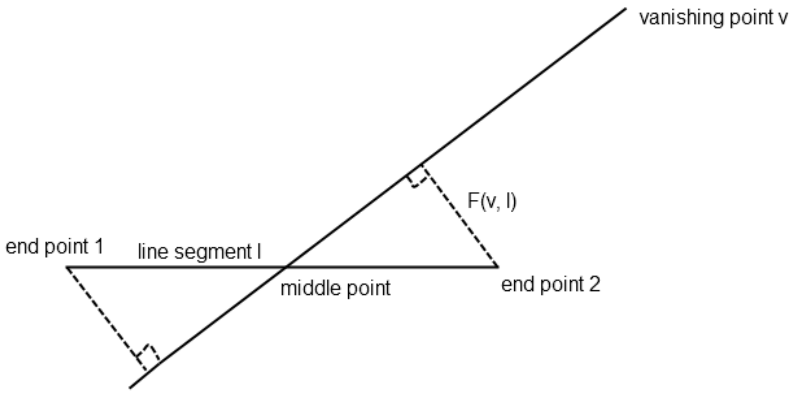

FIG 6 Approximation of fitness function F

Function D, to be used at the random sampling step in j-linkage estimator, was not described in the literature (Tardif, 2009). According to the key idea of j-linkage and our experiments, it could also be well approximated as the distance of two line segments' middle points:

$$D(l_i, l_j) = \|m_i - m_j\|, m_k = (\frac{x_k^s + x_k^e}{2}, \frac{y_k^s + y_k^e}{2}), k = i, j \tag{4}$$

3.4 Vanishing-point Calibration

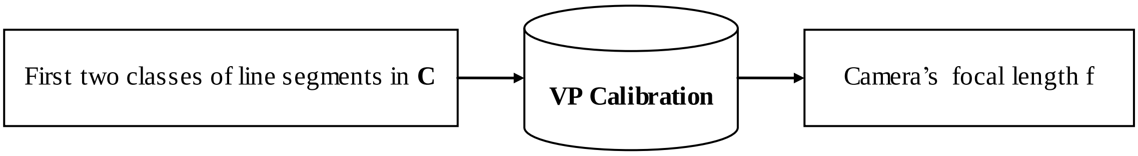

FIG 7 I/O of vanishing-point calibration module

In this module, based on the above mentioned Manhattan world assumption, we further assume that the first three largest line segment classes in size should correspond to the three coordinate basis directions in the Manhattan reference frame, which is to say their corresponding 3D directions are perpendicular with each other. With this assumption which is often valid in most of the urban outdoor and indoor scenes, there is no need for users to specify which three classes of line segments form a orthogonal coordinate system.

Hence the vanishing point calibration could be automatically completed. Firstly, for each class of line segments, estimate the best fit vanishing point through:

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \vdots & \vdots & \vdots \\ a_n & b_n & c_n \end{bmatrix} \cdot \begin{bmatrix} v^x \\ v^y \\ v^w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, (a_i, b_i, c_i) = \frac{(x_i^s, y_i^s, 1) \times (x_i^e, y_i^e, 1)}{\|(x_i^s, y_i^s, 1) \times (x_i^e, y_i^e, 1)\|}, i = 1, 2, \cdots, n, \tag{5}$$

In equation (5), (v^x, v^y, v^w) is the homogeneous coordinate of the vanishing point v, and could be solved using the singular value decomposition (SVD) algorithm. After the first two vanishing points v_1, v_2 are estimated from the two classes, the camera focal length f could be calculated by the equation (Caprile and Torre, 1990):

$$f = \sqrt{-(x_1 - \frac{W}{2})(x_2 - \frac{W}{2}) - (y_1 - \frac{H}{2})(y_2 - \frac{H}{2})}, \tag{6}$$

In (6) $x_i = v_i^x / v_i^w, y_i = v_i^y / v_i^w, i = 1, 2.$

3.5 Normal Deduction

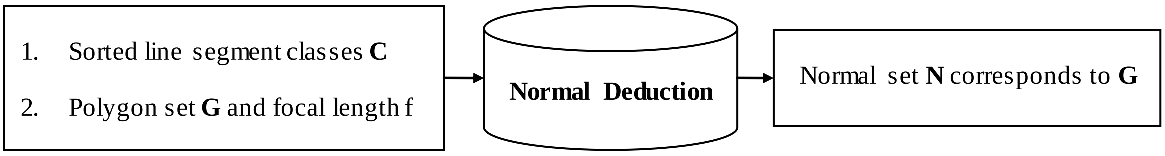
1. Sorted line segment classes **C**
2. Polygon set **G** and focal length f
Normal Deduction
Normal set **N** corresponds to **G**

FIG. 8 I/O of normal deduction module

Normal deduction is essential in many SVR algorithms (Grossmann and Santos-Victor, 2005, Sturm and Maybank, 1999). One of our semi-automatic SVR algorithm's features is that, in this module it could automatically calculate and assign normal directions for each of the 3D planes which are projected onto the image plane as polygons, while in traditional SVR algorithms they has to be specified all by users manually.

The basic idea of this module is the fact that with two known 3D directions parallel to a 3D plane, the normal direction of the plane could be calculated, i.e. their cross product. While in camera geometry, 3D directions correspond to vanishing points in image plane, thus one can get the following equation (Sturm and Maybank, 1999):

$$\mathbf{n} = \frac{\mathbf{K^T l}}{||\mathbf{K^T l}||}, \mathbf{l} = \mathbf{v_1} \times \mathbf{v_2}, \tag{7}$$

In (7), $\mathbf{n}$ is the unit normal direction, $\mathbf{K}$ is camera calibration matrix from equations (1) and (6) and $\mathbf{v_1}, \mathbf{v_2}$ are homogeneous coordinates of two different vanishing points whose corresponding 3D directions are parallel to the plane with normal $\mathbf{n}$.

Once we know how to calculate the normal direction from the 3D plane's two different vanishing points, the only computation remaining is how to automatically find two vanishing points of a 3D plane (projected onto the image plane as polygon). With the help of some simple computational geometry algorithms, and the assumption that each 3D plane has plenty of parallel lines with at least two different directions, our normal deduction algorithm could be described in the following pseudo-code:

```
For each polygon g in polygon set G
    From within line segment set U and D, put all line segments that lie within g into a new line segment set T
    Find two line segment classes $C_i, C_j$, such that among all classes, $|C_k \cap T|, k = i, j$ are the two largest
    Estimate two vanishing points $v_i, v_j$ from $C_i, C_j$ by equation (5)
    Calculate the unit normal direction $\mathbf{n}$ by equation (7), which is the deduced normal for polygon g
```

Certainly, under some special cases, those assumptions do not hold, so errors may happen and some of the calculated normal directions may go wrong. That is why there needs to be a validation step for users to check those errors (*FIG. 2*), and this is still much easier than the traditional method.

3.6 Sturm's SVR

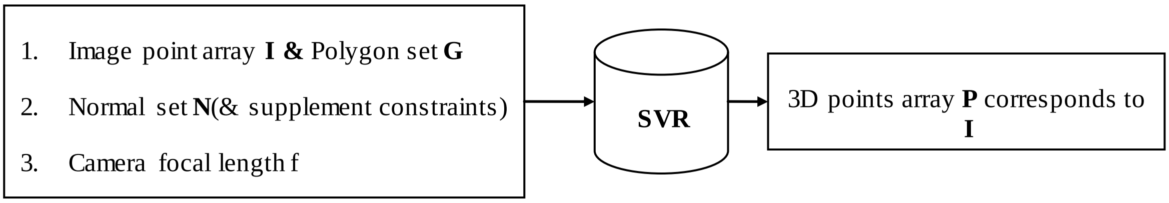
1. Image point array **I** & Polygon set **G**
2. Normal set **N**(& supplement constraints)
3. Camera focal length f
SVR
3D points array **P** corresponds to **I**

FIG. 9 I/O of SVR module

This module is basically the same as Sturm's SVR algorithm (Sturm and Maybank, 1999). However, we add another kind of constraint into their original linear system---parallelogram constraint. The purpose of this is to make our algorithm more flexible, for parallelograms are easy to find on buildings and there is no need to use normal information when adding this constraint into the system.

The parallelogram constraint is based on the geometry fact that if four 3D points $\mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}, \mathbf{P_4}$ could successively form a parallelogram, they must satisfy the equation

$$\mathbf{P_1} + \mathbf{P_3} = \mathbf{P_2} + \mathbf{P_4}. \tag{8}$$

Using the same parameterization as Sturm's method, if there are N image points to be reconstructed, M polygons, and K parallelograms, the linear system should be

$$\begin{pmatrix} \underset{M\times M}{\mathbf{D}} & \underset{M\times N}{\mathbf{C}} \\ \underset{N\times M}{\mathbf{C}^T} & \underset{N\times N}{\mathbf{L}} \\ \underset{3K\times M}{\mathbf{0}} & \underset{3K\times N}{\mathbf{E}} \end{pmatrix} \begin{pmatrix} \underset{M\times 1}{\mathbf{d}} \\ \underset{N\times 1}{\lambda} \end{pmatrix} = \begin{pmatrix} \underset{M\times 1}{\mathbf{0}} \\ \underset{N\times 1}{\mathbf{0}} \\ \underset{3K\times 1}{\mathbf{0}} \end{pmatrix}, \tag{9}$$

in which $\mathbf{E}\lambda = \mathbf{0}$ expresses the parallelogram constraint and matrices $\mathbf{D}, \mathbf{C}, \mathbf{L}$ have the same meaning as Sturm's method.

4. EXPERIMENTS

We implement the above semi-automatic SVR algorithm in Windows XP platform using C++. The LSD module is available provided by its author at http://www.ipol.im/pub/algo/gjmr_line_segment_detector/. The original j-linkage module is also provided by its author at http://www.toldo.info/roberto/?page_id=46. Some of the experiment results are shown below.

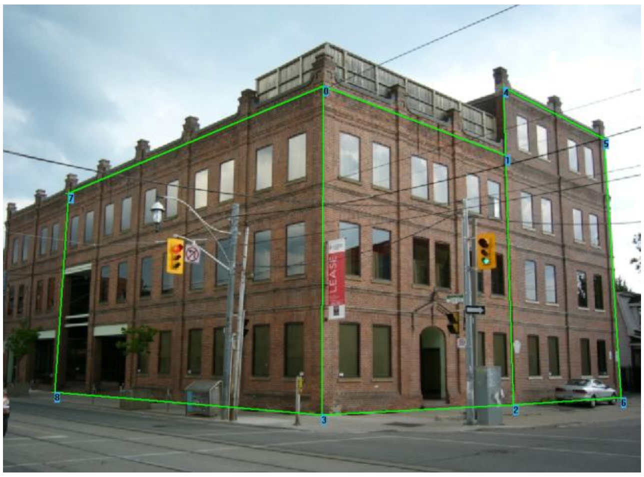

FIG. 10 User inputs a set of line segments (drawn in green)[1](Denis et al., 2008)

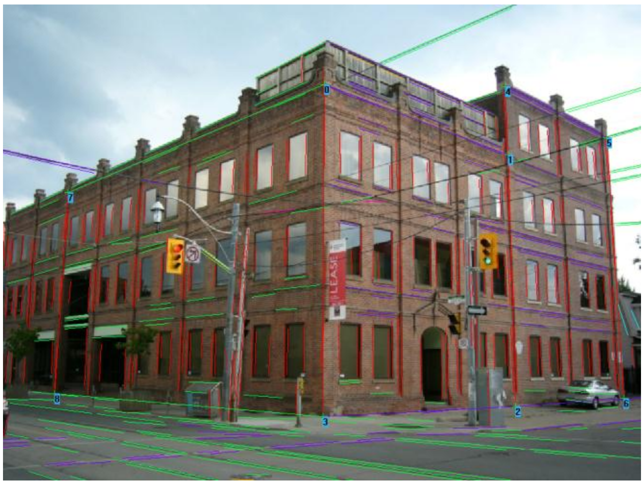

FIG. 11 LSD and J-linkage output, the first three classes are drawn in red, green and purple respectively

[1] This picture comes from York Urban Database provided by Denis et al. 2008

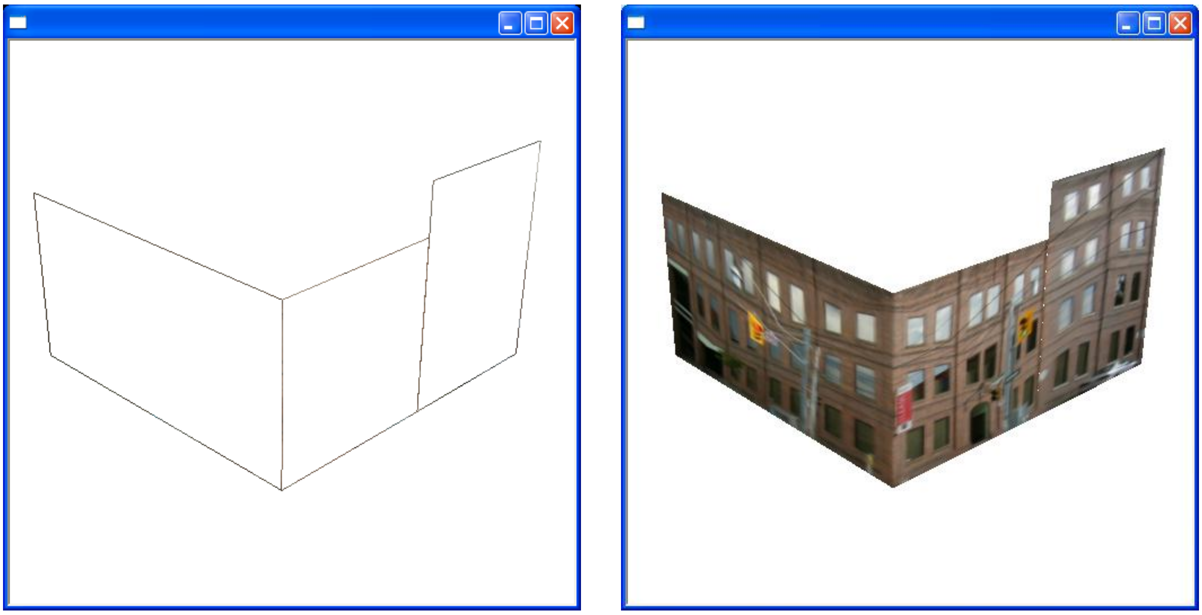

FIG. 12 Reconstructed 3D model in wire-frame and surface mode

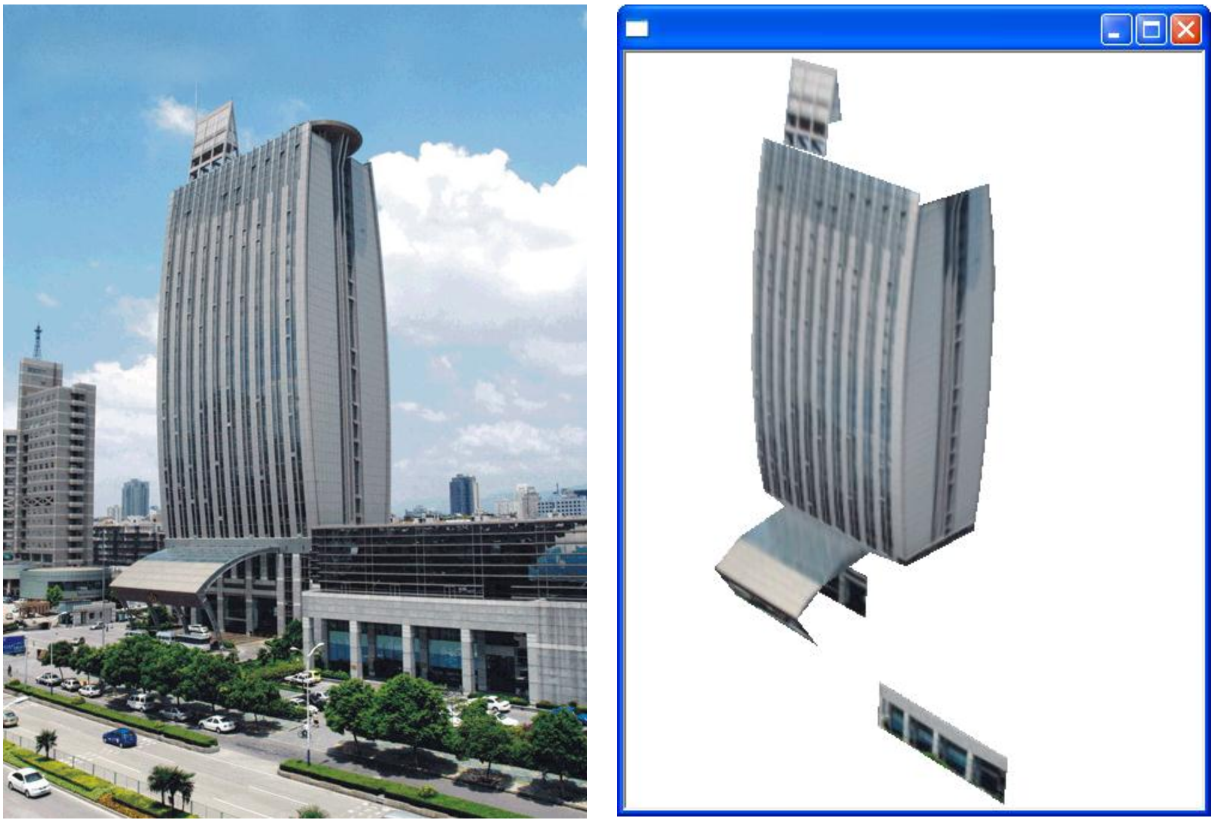

FIG. 13 Another reconstructed 3D model

5. CONTRIBUTIONS

The main contributions of this paper are summarized as follows:

- Introducing LSD and J-linkage algorithms into SVR, under certain assumptions, the automation of vanishing point calibration and 3D plane normal deduction are made possible.

- Taking advantage of a new kind of constraint—parallelogram, integrating it into the Sturm's SVR linear system, our SVR algorithm becomes more flexible.

6. CONCLUSIONS

This paper presented a novel SVR algorithm. By utilizing a newly proposed line segment detector and a robust multiple structures estimator, we introduced automatic vanishing point calibration and 3D plane normal deduction into the algorithm, thereby reducing much of the user interaction burden. Also, we extended the traditional SVR algorithm by adding parallelogram as a new kind of constraint, which does not need normal direction to form the SVR linear system. In the future we plan to consider additional approaches to make this automation more robust.

7. REFERENCES

Caprile, B. and Torre, V. (1990). Using vanishing points for camera calibration. *International Journal of Computer Vision,* 4**,** 127-139.

Coughlan, J. M. and Yuille, A. L. (1999). Manhattan World: compass direction from a single image by Bayesian Inference. International Conference on Computer Vision(ICCV), 1999.

Criminisi, A., Reid, I. and Zisserman, A. (2000). Single view metrology. *International Journal of Computer Vision,* 40**,** 123-148.

Delage, E., Lee, H. and Ng, A. (2006). A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. Computer vision and pattern recognition(CVPR).

Delage, E., Lee, H. and Ng, A. (2007). Automatic single-image 3d reconstructions of indoor manhattan world scenes. *Robotics Research***,** 305-321.

Denis, P., Elder, J. and Estrada, F. (2008). Efficient edge-based methods for estimating manhattan frames in urban imagery. European Conference on Computer Vision, 2008.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM,* 24**,** 381-395.

Grompone, G., Jakubowicz, J., Morel, J. and Randall, G. (2010). LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 32**,** 722.

Grossmann, E. and Santos-Victor, J. (2005). Least-squares 3D reconstruction from one or more views and geometric clues. *Computer vision and image understanding,* 99**,** 151-174.

Hoiem, D., Efros, A. A. and Hebert, M. (2005). Automatic photo pop-up. *ACM Transactions on Graphics (TOG),* 24**,** 584.

Liebowitz, D. and Zisserman, A. (1999). Combining scene and auto-calibration constraints. The 7th International Conference on Computer Vision, Kerkyra, Greece 1999.

Saxena, A., Chung, S. and Ng, A. (2006). Learning depth from single monocular images. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS,* 18**,** 1161.

Sturm, P. and Maybank, S. J. (1999). A method for interactive 3d reconstruction of piecewise planar objects from single images. British Machine Vision Conference, Nottingham, England, 1999.

Tardif, J. P. (2009). Non-Iterative Approach for Fast and Accurate Vanishing Point Detection. International Conference on Computer Vision(ICCV), 2009.

Toldo, R. and Fusiello, A. (2008). Robust multiple structures estimation with J-Linkage. European Conference on Computer Vision(ECCV), 2008.

Van Den Heuvel, F. A. (1998). 3D reconstruction from a single image using geometric constraints. *ISPRS Journal of Photogrammetry and Remote Sensing,* 53**,** 354-368.

Youichi, H., Ken-Ichi, A. and Kiyoshi, A. (1997). Tour into the picture: using a spidery mesh interface to make animation from a single image. Proceedings of the 24th annual conference on Computer graphics and interactive techniques.