Single Image 3D Building Reconstruction Using Rectangles Parallel to an Axis

Tomás Ferranti
School of Applied Mathematics
Fundação Getulio Vargas - FGV-EMAp
Rio de Janeiro, Brazil
tomas_ferranti_07@hotmail.com

Asla Medeiros e Sá
School of Applied Mathematics
Fundação Getulio Vargas - FGV-EMAp
Rio de Janeiro, Brazil
asla.sa@fgv.br

Paulo Cezar Pinto Carvalho
School of Applied Mathematics
Fundação Getulio Vargas - FGV-EMAp
Rio de Janeiro, Brazil
paulo.carvalho@fgv.br

***Abstract*—Historic photographic collections are valuable documents of urban evolution through time. Many historic buildings documented in such collections may have been demolished or changed over time. Digital modeling such buildings may be challenging due to the reduced amount of information available that may be limited to a few images and/or schematic drawings. This paper presents a method to create a 3D set of rectangles that approximates elements of a scene (such as walls, floors, and roofs) from a single image. Using a pinhole camera model, the extraction of geometry and texture of planes parallel to an axis can be obtained after a camera calibration step that recovers intrinsic parameters of the model. Results show that a good visualization of the scene can be created, using the proposed technique, from a single image.**

***Index Terms*—computer vision, building reconstruction, historic photographic collections**

I. Introduction

Computer vision is a broad field with many subareas that have been studied and developed in the last decades, where projective geometry is one of its most important tools. One of its subareas is three dimensional (3D) reconstruction which consists of the process of estimating the 3D characteristics of single or multiple objects, such as shape and appearance.

3D reconstruction can be achieved through various types of methods, using different inputs and outputs. Usual inputs can be images (one or multiple images), volumetric data, and point-cloud data. Common outputs are polygonal meshes, implicit functions, and voxel data. The work can be automated or semi-automated guided by user inputs through an interface.

Reconstructing scenes with little data such as a single image poses many challenges. In particular the result may be an inaccurate model representation, due to multiple objects overlapping in the image plane and/or possibly poor quality of the input image. Another issue is the fact that the perspective projection that produced the image is unknown.

In this work, we propose a method for reconstructing a 3D model of buildings from a single picture. The camera calibration, assumed to be a pinhole camera, is done by using the vanishing points of three mutually orthogonal world directions, which must be annotated by the user in the image. After the calibration step, we are able to create a chain of rectangles, situated in planes parallel to at least one of the world directions selected during calibration step. These planes usually represent structures such as walls, floors, and roofs. We do that by annotating, in the input image, points that correspond to corners of the target structures. The overall workflow is illustrated in Fig. 1.

The remaining of this paper is organized as follows: Section II presents an overview of different approaches to the model reconstruction problem, highlighting main similarities and differences to our proposal. Section III states the geometry of the model and exposes each step of our proposal, such as camera calibration, plane concatenation, and texture extraction. Section IV shows four images examples of this model, talking about main results and problems. Section V concludes the discussion, gathering all the findings and future work.

II. Related Work

Surveys to evaluate the state-of-the-art of 3D reconstruction are available in the literature. Musialski et al. [1] focuses on urban reconstruction, where the authors gives an overview of this vast field and details several workflows and methods. Considering the method's classification proposed by the authors, our work classifies as an interactive modelling using a camera model type, for instance, a pinhole camera model..

Focusing on techniques employed, Bai et al. [2] establishes the problem of image super-resolution reconstruction and classifies distinct approaches in many categories. Methods based on interpolation are one of these groups, in which our approach, detailed in subsection III-D, fits in.

Plenty of work involving camera models make use of calibration through vanishing points of multiple orthogonal directions. Using two directions and a single image, Guillou et al. [3] achieves the insertion of rectangular 3D boxes that are fit to objects by rotating, scaling, and translating. This process is later followed by a texture extraction for the model that includes locating and filling possible holes.

With one more direction and a picture of a building, Alvarez et al. [4] proposes an interactive system to insert 3D objects into the scene and evaluate their impact in the original image. Our proposal extends of their work following the same steps for camera calibration, while covering a more general problem.

Analyzing other data input types allows some insights and ideas for the workflow. Working with a set of still photographs, Debevec et al. [5] employs photogrammetric modeling and view

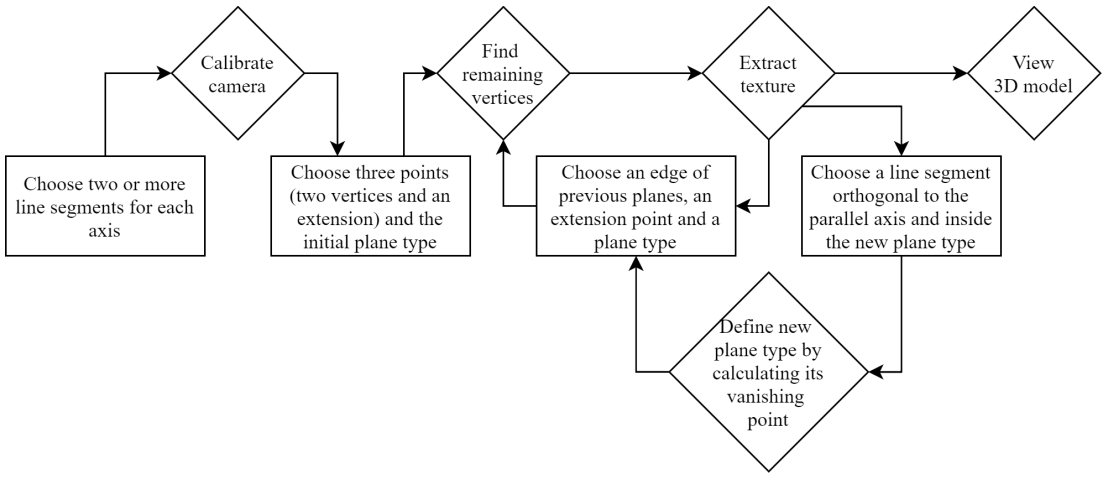


Fig. 1: Diagram showing the steps of creating the 3D model from a single image. Rectangles and rhombuses are user inputs and internal calculations, respectively.

dependent texture mapping to model and render architectural scenes. Our approach differs from it once we use as input a single image and user assisted vanishing points annotation. In the context of scanners, Ochmann et al. [6] deals with indoor point clouds by applying a volumetric parametric building model. Adding colors, Dornelles and Jung [7] handles RGB-D sensors data and uses an iterative pose alignment procedure.

In relation to aerial data, Mahmud et al. [8] creates a multi-task, multi-feature learning formulation from a single overhead image. Integrating with information from large-scale 2D Geographic Information System (GIS) databases, Suveg and Vosselman [9] makes use of a building reconstruction process similar to a search tree. Consisting of airborne image and laserscanner data, Rottensteiner et al. [10] presents a data set to evaluate the results of various submitted methods which afterwards are compared and analysed to identify promising strategies for urban object extraction.

Our proposal to solve 3D reconstruction problems from images differentiates from others by multiple aspects. Using only a single architectural picture, the entire process is guided by the user allowing a wide variety of 3D models to be created. Being simple and straightforward, each image may take from 5 to 10 minutes to create a reasonable final object.

III. Scene Modelling

The model set out for this paper is a pinhole camera model. A pinhole camera model can be very beneficial given its simplicity, such as the absence of lens distortion and a reasonable description of how a camera depicts a 3D scene. In these type of models, we typically have three coordinate systems.

These systems are illustrated in Fig. 2: the World Coordinate System (WCS), defined by the axes $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$, which indicates the objects coordinates in the world; the Camera Coordinate System (CCS), characterized by the axes $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$, centered in $\boldsymbol{C}$ and having $\boldsymbol{W}$ perpendicular to the image plane, which describes the object's position in relation to the camera position; and finally, the Image Coordinate System (ICS), having only two dimensions determined by the axes $\boldsymbol{u}$ and $\boldsymbol{v}$, it provides the pixels coordinates in the image.

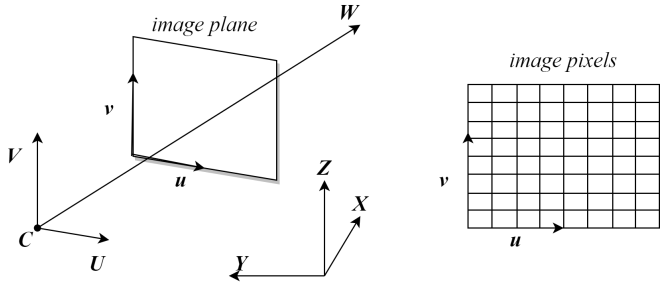


Fig. 2: Different types of coordinate systems in the model.

A. Camera Calibration

A transformation matrix portrays the transition between these three systems. The problem of identifying the transformation matrix that produced a given image is called camera calibration. One of the ways to recover this matrix is through calibration using vanishing points. The process adopted for this step is identical to the one in Alvarez et al. [4].

Consider three directions mutually orthogonal in the WCS. The camera calibration is achieved through the vanishing points relative to these directions vectors: $\boldsymbol{F_X}$, $\boldsymbol{F_Y}$, and $\boldsymbol{F_Z}$. Naming $\boldsymbol{C}$ the position of the camera, the segments

$$\begin{aligned} \boldsymbol{CF_X} &= \boldsymbol{F_X} - \boldsymbol{C} \\ \boldsymbol{CF_Y} &= \boldsymbol{F_Y} - \boldsymbol{C} \\ \boldsymbol{CF_Z} &= \boldsymbol{F_Z} - \boldsymbol{C} \end{aligned} \tag{1}$$

are also mutually orthogonal.

The image projection of a point in the WCS or CCS is the intersection of its line to $\boldsymbol{C}$ with the image plane. Designating $\boldsymbol{H}$ as the optical center (point of intersection between $\boldsymbol{W}$ axis

and image plane), [4] shows that $\boldsymbol{H}$ is exactly the orthocenter of the triangle defined by $\boldsymbol{F_X}$, $\boldsymbol{F_Y}$, and $\boldsymbol{F_Z}$ image projections.

The transformation matrix is now recovered with these results. Denominating the distance between the image plane to the camera as $w_c = \|\boldsymbol{H} - \boldsymbol{C}\|$, a notable outcome is

$$w_c^2 = \frac{\|\boldsymbol{F_X} - \boldsymbol{F_Y}\|^2 - \|\boldsymbol{F_X} - \boldsymbol{H}\|^2 - \|\boldsymbol{F_Y} - \boldsymbol{H}\|^2}{2} \tag{2}$$

where $\|.\|$ is the euclidean norm. Label the vectors (X_u, X_v, X_w), (Y_u, Y_v, Y_w), and (Z_u, Z_v, Z_w) as the normalized vectors of $\boldsymbol{CF_X}$, $\boldsymbol{CF_Y}$, and $\boldsymbol{CF_Z}$, respectively. The transition between WCS and CCS can be written as

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} X_u & Y_u & Z_u \\ X_v & Y_v & Z_v \\ X_w & Y_w & Z_w \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{3}$$

with (x', y', z') belonging to the WCS and (u', v', w') to the CCS. Call (a, b) and (u_c, v_c) the coordinates of (u', v', w') and $\boldsymbol{H}$ image projections in the ICS, respectively. Using the fact that (a, b) lies in the intersection of the image plane with the line defined by (u', v', w'), we have

$$\begin{aligned} (a - u_c, b - v_c, w_c) &= k(u', v', w') \\ k = \frac{w_c}{w'} \implies \begin{cases} a &= \frac{w_c u'}{w'} + u_c \\ b &= \frac{w_c v'}{w'} + v_c \end{cases} \end{aligned} \tag{4}$$

where k is a scalar. This leads to our transformation matrix in homogeneous coordinates

$$\begin{bmatrix} ta \\ tb \\ t \end{bmatrix} = \begin{bmatrix} w_c & 0 & u_c \\ 0 & w_c & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_u & Y_u & Z_u \\ X_v & Y_v & Z_v \\ X_w & Y_w & Z_w \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{5}$$

that takes a point from the WCS and finds its projection in the ICS.

B. Unprojecting Points From Planes

The problem of taking a projection point in the ICS and finding its corresponding point in the WCS is undetermined. There is an entire line that projects into the same point. To solve that, the point is assumed to belong to a certain plane. The only requirement for this plane is being parallel to at least one of the world axis.

Naming this plane π and his parallel axis a, consider the set of lines orthogonal to a within π. As illustrated in Fig. 3, these lines vanishing point will always lie in the segment defined by the other two axes vanishing points. A point of this plane, its parallel axis, and vanishing point are enough information to find the intersection with any line in the WCS.

Name $\boldsymbol{P_0}$ a point of π and $\boldsymbol{P_I}$ the desired point in the ICS. The normal of the plane is given by $\boldsymbol{n} = \boldsymbol{d} \times \boldsymbol{e}$, where $\boldsymbol{d}$ is the unprojection of the vanishing point direction and $\boldsymbol{e}$ is the parallel axis direction. Calling $\boldsymbol{P}$ the correspondent of $\boldsymbol{P_I}$ in the WCS, we have that $\boldsymbol{P}$ is in the line defined by $\boldsymbol{C}$ and $\boldsymbol{P_I}$ and also in the plane. Through Fig. 4 we can see that

$$\begin{aligned} s(\boldsymbol{P_I} - \boldsymbol{C}) &= \boldsymbol{P} - \boldsymbol{C} \\ &= (\boldsymbol{P_0} - \boldsymbol{C}) + (\boldsymbol{P} - \boldsymbol{P_0}) \\ &= (\boldsymbol{P_0} - \boldsymbol{C}) + (p_1\boldsymbol{d} + p_2\boldsymbol{e}) \end{aligned} \tag{6}$$

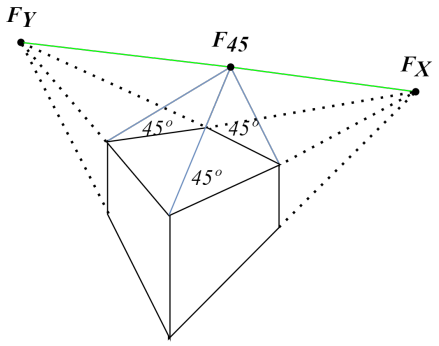


Fig. 3: Example of three blue segments parallel to axis $\boldsymbol{Z}$ with angle of 45 degrees with the other two axis. The vanishing point $\boldsymbol{F_{45}}$ lies on the green line. This green line is defined by the other two non-parallel axes vanishing points, $\boldsymbol{F_X}$ and $\boldsymbol{F_Y}$.

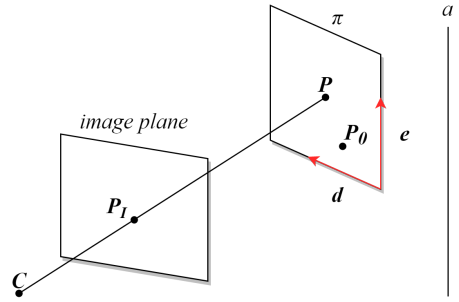


Fig. 4: The point $\boldsymbol{P_I}$ is known in the image and it is desired to find $\boldsymbol{P}$, its correspondent in the world. $\boldsymbol{P_0}$ and $\boldsymbol{P}$ belongs to π, which is parallel to the axis a. With the orthogonal vectors $\boldsymbol{d}$ and $\boldsymbol{e}$ the plane π is defined, allowing the calculation of $\boldsymbol{P}$ coordinates in the WCS.

where p_1, p_2, and s are scalars. When taking the dot product with $\boldsymbol{n}$ we have

$$\begin{aligned} s(\boldsymbol{P_I} - \boldsymbol{C}) \cdot \boldsymbol{n} &= (\boldsymbol{P_0} - \boldsymbol{C}) \cdot \boldsymbol{n} + p_1\boldsymbol{d} \cdot \boldsymbol{n} + p_2\boldsymbol{e} \cdot \boldsymbol{n} \\ s &= \frac{(\boldsymbol{P_0} - \boldsymbol{C}) \cdot \boldsymbol{n}}{(\boldsymbol{P_I} - \boldsymbol{C}) \cdot \boldsymbol{n}} \end{aligned} \tag{7}$$

which specifies $\boldsymbol{P}$ coordinates. Hence the unprojection residing in a plane of any point of the ICS is found by using any point of this plane, its parallel axis, and a line segment orthogonal to this axis.

C. Planes Concatenation

The three vanishing points used in the calibration step are estimated by locating in the image two or more line segments for each relative direction. Every direction then is processed individually, where the vanishing point coordinates is calculated through the mean of the pairwise intersections of the lines defined by the segments.

After calibration, the reconstruction of the scene on multiple planes is accomplished through user interaction. With the type of plane and three points in the boundary of a rectangle, the

initial rectangle is established. After that, any of the segments of previous planes can be chosen to expand the model, specifying an extension point and changing the plane type if needed.

Initially, three basic types of planes are defined: XZ, YZ, and XY. These are associated with one of the two axes and the already calculated vanishing point. The addition of a new type of plane is done by indicating its parallel axis and line segment in the image.

Given that we have no initial points, we can assume that $(1, 1, 1)$ belongs to the initial plane. This choice is arbitrary and reflects the fact that the true scale of the scene cannot be recovered with a single image. After the type of this first plane is defined, we can unproject the selected three points from the screen. The first two points establish two vertices of the rectangle, where the third is an extension point.

This third point is used to find the coordinates of the remaining two corners. Being a rectangle means that they have a mix of the two closer vertices coordinates. We can test every combination of coordinates. The correct one will have the lines defined by the segments going through the vanishing point related to its plane type.

The addition of other planes can be done in a similar fashion considering the segment of the adjacent plane as the two initial vertices and another user annotated extension point.

D. Texture Extraction

The quadrilateral formed in the image by the rectangle vertices projections bounds a certain region texture. The strategy adopted to map this texture to the rectangle follows two basic steps. Firstly, a width and height is needed to determine the size and number of pixels. Then a transformation between the image and the texture is done to fill the pixels colors.

Finding the width and height can be done through an aspect ratio test. Consider the rectangle aspect ratio in the WCS as $A = W/H$ and the quadrilateral aspect ratio in the ICS as $a = w/h$. If A is larger than a, the width and height are w and w/A, respectively. Otherwise, the width and height are hA and h, respectively.

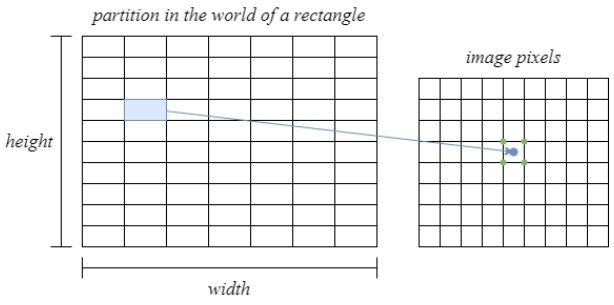


Fig. 5: The blue rectangle of the partition is projected in the image, resulting in decimal pixel value. To approximate its color, bilinear interpolation is done with the closest four points (represented by the green dots) using their RGBA values separately.

A partition is done to draft the texture in the rectangle area within the plane. Splitting it into $width \times height$ rectangles as in Fig. 5, we can project each one into the image and find a decimal value for its pixel position. An approximation of its value is calculated by doing bilinear interpolation for each value of the RGBA color using the four closest position-wise pixels in the image.

IV. Results & Discussion

Using the steps described previously, a workflow is created to process an image. Initially, we need to calibrate the camera. This is done with two or more line segments from the image for each world axis. These are used to estimate the vanishing points of the axes directions and, consequently, the transformation matrix. By specifying the first plane type and three points on the plane, an initial rectangle is created. From that, we can concatenate other rectangles from different plane types. This workflow can be visualized in Fig. 1.

Four examples are given in Fig. 6. These photos have two main needed characteristics: no lens distortion and three mutually orthogonal directions. For each image we have four pictures taken at the following stages:

- Stage (i): when the red, green, and blue segments are chosen as the calibration segments for the axes $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$, respectively. The pink point is $\boldsymbol{H_I}$ and new plane types segments are defined by the pink segments;
- Stage (ii): the image after the creation of the first rectangle and the expansion of the set to other types of planes. They are delimited by the yellow lines;
- Stage (iii)-(iv): shows how the final model looks from different points of view in 3D.

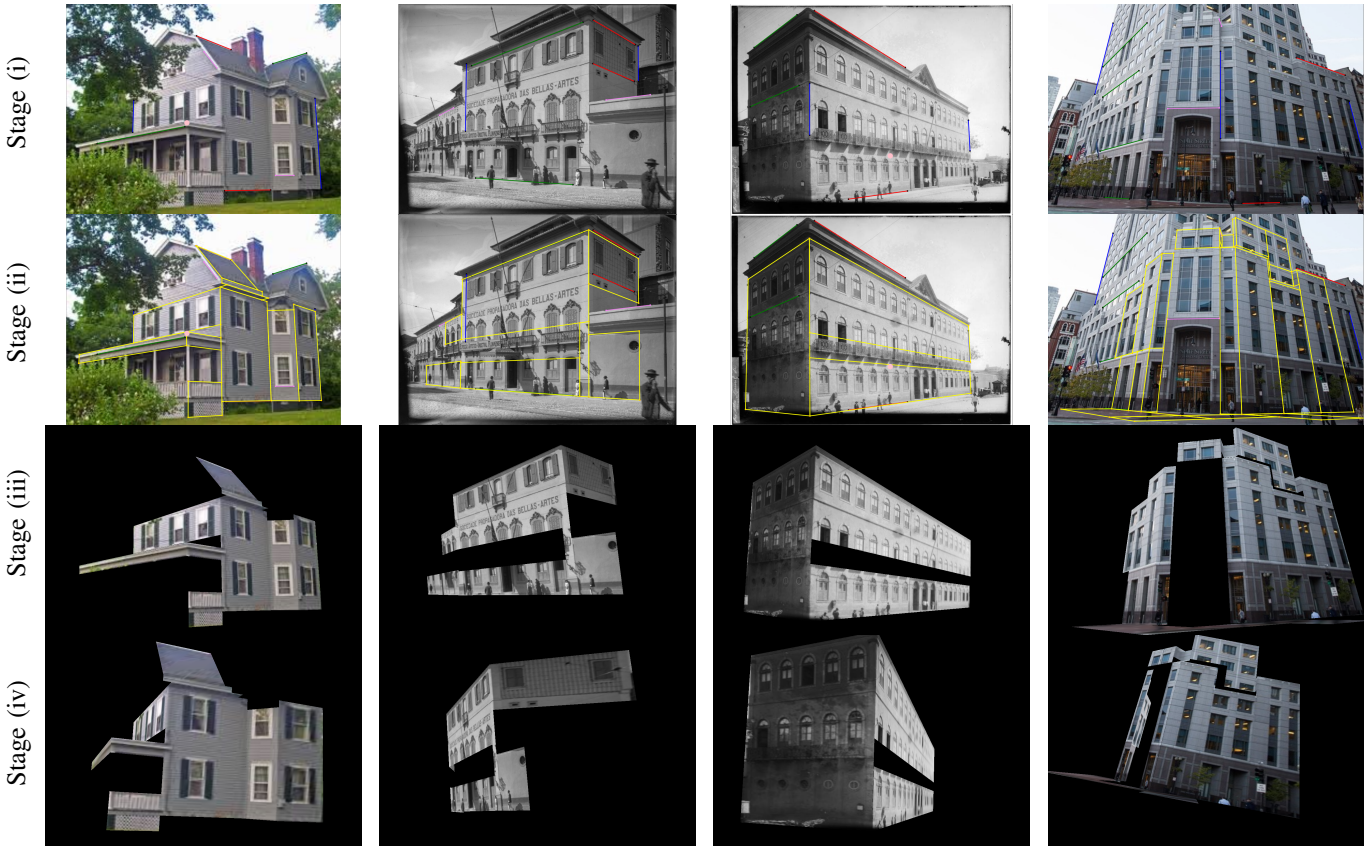
A lot of intrinsic parameters can be recovered from each model. Focusing only on Fig. 6a, the slanted wall angle with axis $\boldsymbol{X}$ is 0.57 radians or approximately 33 degrees. The same can be applied to the roof plane, having approximately 40 degrees angle with the axis $\boldsymbol{Z}$. A 3D overview allows the perception of size and distance, which would be unknown or badly guessed using only the image.

For some images it may be difficult to obtain a precise calibration. This happens when two or more line segments of an axis are almost parallel, resulting in numerical problems when calculating the vanishing point. The limitation to quadrilaterals makes so that other geometrical forms are badly represented in the model, such as pillars and balconies. Connecting two objects demands adjacent planes, which some images may not have.

V. Conclusion and Future Work

The proposed method of 3D reconstruction from a single image revealed, through examples, to be effective. Using the pinhole camera model leads to a fast and simple calibration through vanishing points. Unprojecting points from planes parallel to an axis creates a chain of concatenated rectangles of size limited by the image. The bilinear interpolation for each texture pixel allows the extraction of scene texture information. All of this is done through simple points specified by the user.

The 3D model reconstructed can be visually inspected to be consistent with the building by superimposing the reconstructed



(a) Example A (b) Example B (c) Example C (d) Example D

Fig. 6: Different cases of images and their produced 3D models.

model to the building's picture. Inheriting many intrinsic properties, the mesh allows a good visualization of the scene. With additional data such as the correct width and height of a stated object, one could estimate other objects dimensions.

Considering future work, there is room for plenty of improvements. Besides rectangles, a general polygonal or a more complex shape can be incorporated in a similar manner. Other types of calibration can also be added to deal with particular images problems. If multiple images are available, the creation of an approach to merge different sets of rectangles of the same scene would be valuable.

Acknowledgment

This research has been developed in collaboration with Instituto Moreira Sales and Spatial Studies Lab at Rice University.

References

[1] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer, "A Survey of Urban Reconstruction," *Computer Graphics Forum*, vol. 32, no. 6, pp. 146–177, september 2013.

[2] K. Bai, X. Liao, Q. Zhang, X. Jia, and S. Liu, "Survey of learning based single image super-resolution reconstruction technology," *Pattern Recognition and Image Analysis*, vol. 30, no. 4, pp. 567–577, Oct. 2020.

[3] E. Guillou, D. Meneveaux, E. Maisel, and K. Bouatouch, "Using vanishing points for camera calibration and coarse 3d reconstruction from a single image," *The Visual Computer*, vol. 16, pp. 396–410, 11 2000.

[4] B. S. V. Alvarez, P. C. P. Carvalho, and M. Gattass, "Insertion of three-dimensional objects in architectural photos," in *The 10-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2002, WSCG 2002, University of West Bohemia, Campus Bory, Plzen-Bory, Czech Republic, February 4-8, 2002*, 2002, pp. 17–23.

[5] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 11–20.

[6] S. Ochmann, R. Vock, R. Wessel, and R. Klein, "Automatic reconstruction of parametric building models from indoor point clouds," *Computers and Graphics*, vol. 54, pp. 94–103, 2016, special Issue on CAD/Graphics 2015.

[7] T. Dornelles and C. Jung, "Online frame-to-model pipeline to 3d reconstruction with depth cameras using rgb-d information," in *Conference on Graphics, Patterns and Images, 33. (SIBGRAPI), 2020, Virtual. Proceedings*. Los Alamitos: IEEE Computer Society, 2020, On-line.

[8] J. Mahmud, T. Price, A. Bapat, and J.-M. Frahm, "Boundary-aware 3d building reconstruction from a single overhead image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] I. Suveg and G. Vosselman, "Reconstruction of 3d building models from aerial images and maps," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 58, no. 3, pp. 202–224, 2004, integration of Geodata and Imagery for Automated Refinement and Update of Spatial Databases.

[10] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, "The isprs benchmark on urban object classification and 3d building reconstruction," 2012.