FUNDAÇÃO GETULIO VARGAS
SCHOOL OF APPLIED MATHEMATICS

TOMÁS FERRANTI

SINGLE IMAGE 3D BUILDING RECONSTRUCTION USING ADJACENT RECTANGLES PARALLEL TO AN AXIS

Rio de Janeiro
2021

TOMÁS FERRANTI

SINGLE IMAGE 3D BUILDING RECONSTRUCTION USING ADJACENT RECTANGLES PARALLEL TO AN AXIS

Bachelor dissertation presented to the School of Applied Mathematics (FGV/EMAp) to obtain the Bachelor's degree of Applied Mathematics.

Study area: Computer vision.

Advisor: Paulo Cezar Pinto Carvalho

Rio de Janeiro

2021

Ficha catalográfica elaborada pela BMHS/FGV

Ferranti, Tomás
Single image 3D building reconstruction using adjacent rectangles parallel to an axis/ Tomás Ferranti. – 2021.
23f.

Bachelor Dissertation (Graduation) – School of Applied Mathematics.

Advisor: Paulo Cezar Pinto Carvalho.
Includes bibliography.

1. Computer vision. 2. 3D reconstruction. 2. Rectangles parallel to an axis. I. Carvalho, Paulo Cezar. II. School of Applied Mathematics. III. Single image 3D building reconstruction using adjacent rectangles parallel to an axis

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 work 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. Knowing the exact dimension of an object within the model allows the retrieval of its true scale. Results show that a good visualization of the scene can be created, using the proposed technique, from a single image.

Keywords: computer vision, building reconstruction, historic photographic collections.

Contents

1 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 two or more mutually orthogonal world directions, which must be annotated 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. Each rectangle acquires a texture by a bilinear interpolation within its bound projected area in the picture. When the length of an object is known, we can recover the true scale of the model by using proportionality.

Surveys to evaluate the state-of-the-art of 3D reconstruction are available in the literature. The review of (MUSIALSKI et al., 2013) 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., 2020) 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 to this problem, detailed in Section 2.5, 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., 2000) 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; CARVALHO; GATTASS, 2002) 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 similar steps for camera calibration, while covering a more general problem.

Analyzing others processes that handle more information as input allows some insights and ideas for our objective. Working with a set of still photographs, (DEBEVEC; TAYLOR; MALIK, 1996) employs photogrammetric modeling and view 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., 2016) deals with indoor point clouds by applying a volumetric parametric building model. Adding colors, (DORNELLES; JUNG, 2020) handles RGB-D sensors data and uses an iterative pose alignment procedure.

In relation to aerial data, (MAHMUD et al., 2020) 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; VOSSELMAN, 2004) makes use of a building reconstruction process similar to a search tree. Consisting of airborne image and laserscanner data, (ROTTENSTEINER et al., 2012) 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 defined image points, 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 reconstruction. This time includes both the processing and the evaluating process of which points to choose from the picture.

All the results available in this research can be reproduced through the open source tool available at (FERRANTI, 2021). This Web API has been developed in collaboration with *Instituto Moreira Sales* and Spatial Studies Lab at Rice University. Written in HTML and Javascript, it makes use of two external libraries: *Threejs* and *Nodejs*. The first handles all computer vision functions, such as setting up a 3D scene from a perspective camera, while the second operates creating a local server environment to save and load local files. This project final objective, yet to be achieved, is to create a 3D model of the city of *Rio de Janeiro* from previous centuries by making use of the ImagineRio repository.

The remaining of this work is organized as follows: Chapter 2 states the geometry

of the model and exposes the camera calibration, plane concatenation, texture extraction, and scale addition of our proposal. Chapter 3 shows four images examples of this model, discussing main results and problems. Chapter 4 concludes the dissertation, gathering all the findings and future works.

2 Scene Modelling

The model set out for this work 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 Figure 1:

- the World Coordinate System (WCS), centered in $\boldsymbol{C}$ and defined by the axes $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$. Indicates the objects coordinates in the world;
- the Camera Coordinate System (CCS), also centered in $\boldsymbol{C}$ but characterized by the axes $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$, where $\boldsymbol{W}$ perpendicular to the image plane. 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.

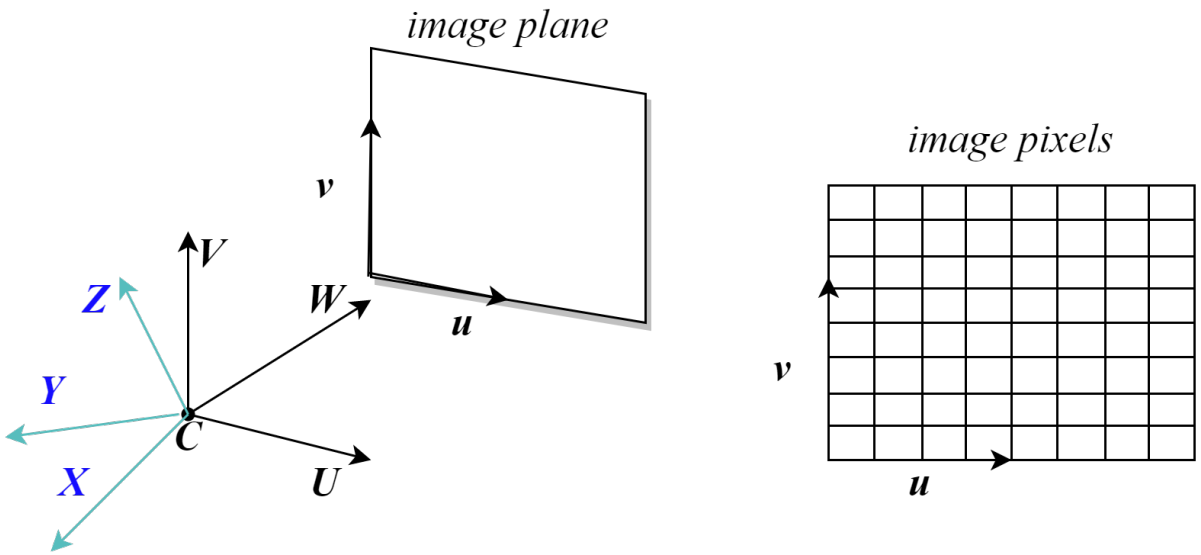


Figure 1 – Different types of coordinate systems in the model.

2.1 Projective Geometry

In this section we provide two main concepts of the projective geometry field: homogeneous coordinates and vanishing points.

2.1.1 Homogeneous Coordinates

When tackling with multiple coordinate systems of unequal dimensions, it is common to employ what we call homogeneous coordinates, also known as projective coordinates. Used in projective geometry, they make projective transformations to be easily represented by a matrix, which are simpler than their Cartesian formulas counterparts.

In such geometry, the points $[x : y : 1]$ and $[tx : ty : t]$ with homogeneous coordinates are considered to represent the same point (x, y), for all nonzero values of t. We can write this relation as

$$\begin{bmatrix} x \\ y \end{bmatrix} \iff \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} tx \\ ty \\ t \end{bmatrix} \tag{2.1}$$

and this allows us to write the projective transformations $(y_1, y_2, y_3) \mapsto (x_1, x_2)$ of form

$$x_1 = \frac{a_1 y_1 + a_2 y_2 + a_3 y_3}{a_7 y_1 + a_8 y_2 + a_9 y_3}, x_2 = \frac{a_4 y_1 + a_5 y_2 + a_6 y_3}{a_7 y_1 + a_8 y_2 + a_9 y_3} \tag{2.2}$$

as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \iff \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \cong \begin{bmatrix} tx_1 \\ tx_2 \\ t \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{2.3}$$

which makes calculations and notations much easier.

2.1.2 Vanishing Points

Call the projection of a point $\boldsymbol{P}$ into the image plane as the intersection of the line defined by the observer $\boldsymbol{C}$ and $\boldsymbol{P}$ with the image plane. When projecting the points of a line in the world into the image plane, the result will also represent a line in the image plane. We define the vanishing point $\boldsymbol{v}$ of a direction as the point of intersection of all parallel lines projections of same direction into the image plane. This result is shown in Figure 2.

This concept is studied in many aspects. Being present in art, it is used for one-point perspective, two-point perspective, and three-point perspective drawing styles. In our context, we will use the vanishing points to calibrate our camera.

2.2 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 very similar to the one in (ALVAREZ; CARVALHO; GATTASS, 2002).

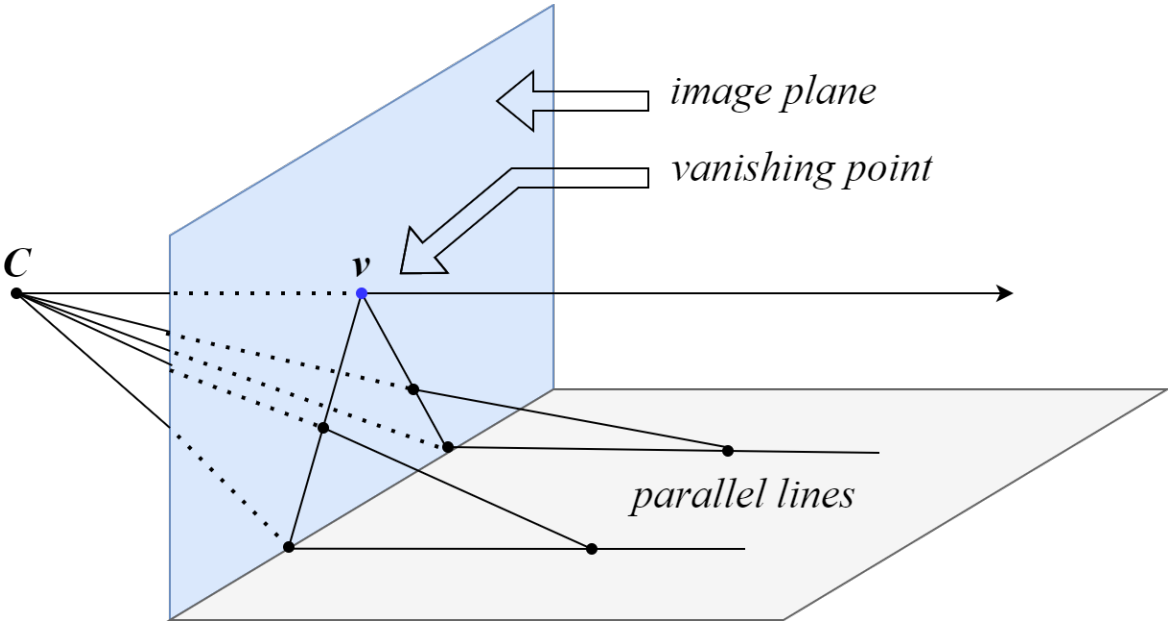


Figure 2 – Example of the vanishing point $\boldsymbol{v}$ corresponding to the direction of the parallel lines in the ground plane. In this scene we have the projection into the image plane in relation to observer $\boldsymbol{C}$.

2.2.1 Non-centered

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

$$\begin{aligned} \boldsymbol{CF_X} &= \boldsymbol{F_X} - \boldsymbol{C} \\ \boldsymbol{CF_Y} &= \boldsymbol{F_Y} - \boldsymbol{C} \\ \boldsymbol{CF_Z} &= \boldsymbol{F_Z} - \boldsymbol{C} \end{aligned} \tag{2.4}$$

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), an interesting observation is that $\boldsymbol{H}$ lies in the orthocenter of the triangle defined by $\boldsymbol{F_X}$, $\boldsymbol{F_Y}$, and $\boldsymbol{F_Z}$. This result is visualized through Figure 3 and can be easily checked as

$$\begin{aligned} (\boldsymbol{H} - \boldsymbol{F_X}) \cdot (\boldsymbol{F_Z} - \boldsymbol{F_Y}) &= ((\boldsymbol{C} - \boldsymbol{F_X}) + (\boldsymbol{H} - \boldsymbol{C})) \cdot (\boldsymbol{F_Z} - \boldsymbol{F_Y}) \\ &= (\boldsymbol{C} - \boldsymbol{F_X}) \cdot ((\boldsymbol{C} - \boldsymbol{F_Y}) + (\boldsymbol{F_Z} - \boldsymbol{C})) = 0 \end{aligned} \tag{2.5}$$

where $\cdot$ is the scalar product, therefore $(\boldsymbol{H} - \boldsymbol{F_X}) \perp (\boldsymbol{F_Z} - \boldsymbol{F_Y})$. By symmetry, similar formulas can be found for $(\boldsymbol{H} - \boldsymbol{F_Y}) \perp (\boldsymbol{F_Z} - \boldsymbol{F_X})$ and $(\boldsymbol{H} - \boldsymbol{F_Z}) \perp (\boldsymbol{F_X} - \boldsymbol{F_Y})$.

The transformation matrix is now recovered with these results. Denominating the distance between the image plane to the camera as $w_c = \|\boldsymbol{H} - \boldsymbol{C}\|$, we can find this value

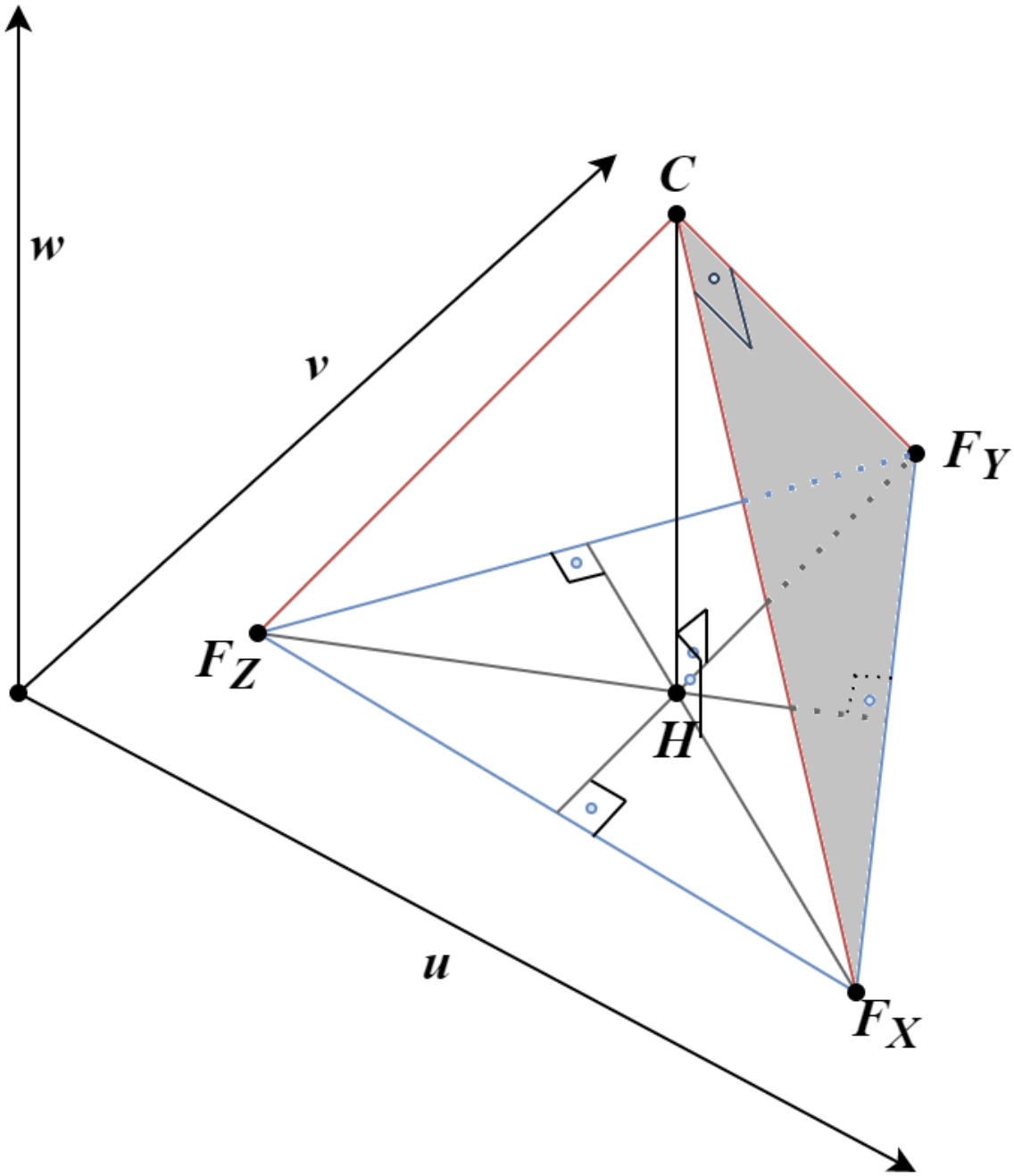


Figure 3 – Considering the triangle established by the vanishing points $\boldsymbol{F_X}$, $\boldsymbol{F_Y}$, and $\boldsymbol{F_Z}$, we have the intersection of their heights being $\boldsymbol{H}$, the projection of $\boldsymbol{C}$ into the plane. The new axis $\boldsymbol{w}$ is given by $-\boldsymbol{W}$.

without knowing $\boldsymbol{C}$ coordinates. By using Pythagoras Theorem on the three rectangular triangles $\triangle \boldsymbol{F_X H C}$, $\triangle \boldsymbol{F_Y H C}$, and $\triangle \boldsymbol{F_X C F_Y}$, we find that

$$\begin{cases} w_c^2 + \|\boldsymbol{F_X} - \boldsymbol{H}\|^2 = \|\boldsymbol{C} - \boldsymbol{F_X}\|^2 \\ w_c^2 + \|\boldsymbol{F_Y} - \boldsymbol{H}\|^2 = \|\boldsymbol{C} - \boldsymbol{F_Y}\|^2 \\ \|\boldsymbol{C} - \boldsymbol{F_X}\|^2 + \|\boldsymbol{C} - \boldsymbol{F_Y}\|^2 = \|\boldsymbol{F_Y} - \boldsymbol{F_X}\|^2 \end{cases} \implies \tag{2.6}$$
$$\implies w_c^2 = \frac{\|\boldsymbol{F_X} - \boldsymbol{F_Y}\|^2 - \|\boldsymbol{F_X} - \boldsymbol{H}\|^2 - \|\boldsymbol{F_Y} - \boldsymbol{H}\|^2}{2}$$

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

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} X_u & Y_u & Z_u \\ X_v & Y_v & Z_v \\ X_w & Y_w & Z_w \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{2.7}$$

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

$$(a - u_c, b - v_c, w_c) = k(u', v', w')$$
$$k = \frac{w_c}{w'} \implies \begin{cases} a &= \frac{w_c u' + u_c w'}{w'} \\ b &= \frac{w_c v' + v_c w'}{w'} \end{cases} \tag{2.8}$$

where k is a scalar. From Subsection 2.1.1, this leads to our projective transformation in homogeneous coordinates

$$\begin{bmatrix} a \\ b \end{bmatrix} \iff \begin{bmatrix} a \\ b \\ 1 \end{bmatrix} \cong \begin{bmatrix} ta \\ tb \\ t \end{bmatrix} = \begin{bmatrix} w_c & 0 & u_c \\ 0 & w_c & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_u & Y_u & Z_u \\ X_v & Y_v & Z_v \\ X_w & Y_w & Z_w \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{2.9}$$

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

Call this type of calibration non-centered, where we use the three vanishing points to find $\boldsymbol{H}$, u_c and v_c, and thus w_c. These values together state the transformation matrix.

2.2.2 Centered

Not all images have their three vanishing points well-defined. Sometimes, the only knowledge available are from two vanishing points. The problem of finding the third is impossible with only this information: it can be anywhere within the plane. Most of the time, if the picture is not cropped, the optical center $\boldsymbol{H}$ lies close to the center of the picture. Thus, assuming that its value is the center, finding the third vanishing point becomes possible.

For example, assume that we do not have the vanishing point $\boldsymbol{F_Z}$. Given that $\boldsymbol{H}$ is in the intersection of the heights, we can write $\boldsymbol{F_Z} = \boldsymbol{H} + j\boldsymbol{N}$, where j is an unknown scalar and $\boldsymbol{N}$ is any non-null vector orthogonal to $\boldsymbol{F_Y} - \boldsymbol{F_X}$. Applying the fact that $\boldsymbol{H} - \boldsymbol{F_X}$ is orthogonal to $\boldsymbol{F_Z} - \boldsymbol{F_Y}$, we have

$$\begin{aligned} 0 &= (\boldsymbol{H} - \boldsymbol{F_X}) \cdot (\boldsymbol{F_Z} - \boldsymbol{F_Y}) \\ &= (\boldsymbol{H} - \boldsymbol{F_X}) \cdot (\boldsymbol{H} + j\boldsymbol{n} - \boldsymbol{F_Y}) \\ &= \|\boldsymbol{H}\|^2 + j\boldsymbol{H} \cdot \boldsymbol{n} - \boldsymbol{H} \cdot \boldsymbol{F_Y} - \boldsymbol{H} \cdot \boldsymbol{F_X} - j\boldsymbol{F_X} \cdot \boldsymbol{n} + \boldsymbol{F_X} \cdot \boldsymbol{F_Y} \\ \iff j &= \frac{\|\boldsymbol{H}\|^2 - (\boldsymbol{F_X} + \boldsymbol{F_Y}) \cdot \boldsymbol{H} + \boldsymbol{F_X} \cdot \boldsymbol{F_Y}}{(\boldsymbol{F_X} - \boldsymbol{H}) \cdot \boldsymbol{n}} \end{aligned} \tag{2.10}$$

and therefore the coordinates of the third vanishing point $\boldsymbol{F_Z}$. A similar equation for the others vanishing points missing cases can be obtained by symmetry. After obtaining the three vanishing points, the same previous steps for camera calibration are followed. Call this type of calibration centered, employing only two vanishing points.

2.3 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 Figure 4, 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 Figure 5 we can see that

$$\begin{aligned} s(\boldsymbol{P_I} - \boldsymbol{C}) &= \boldsymbol{P} - \boldsymbol{C} \\ &= (\boldsymbol{P_0} - \boldsymbol{C}) + (\boldsymbol{P} - \boldsymbol{P_0}) \\ &= (\boldsymbol{P_0} - \boldsymbol{C}) + (p_1\boldsymbol{d} + p_2\boldsymbol{e}) \end{aligned} \tag{2.11}$$

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

$$\begin{aligned} s(\boldsymbol{P_I} - \boldsymbol{C}) \cdot \boldsymbol{n} &= (\boldsymbol{P_0} - \boldsymbol{C}) \cdot \boldsymbol{n} + p_1\boldsymbol{d} \cdot \boldsymbol{n} + p_2\boldsymbol{e} \cdot \boldsymbol{n} \\ s &= \frac{(\boldsymbol{P_0} - \boldsymbol{C}) \cdot \boldsymbol{n}}{(\boldsymbol{P_I} - \boldsymbol{C}) \cdot \boldsymbol{n}} \end{aligned} \tag{2.12}$$

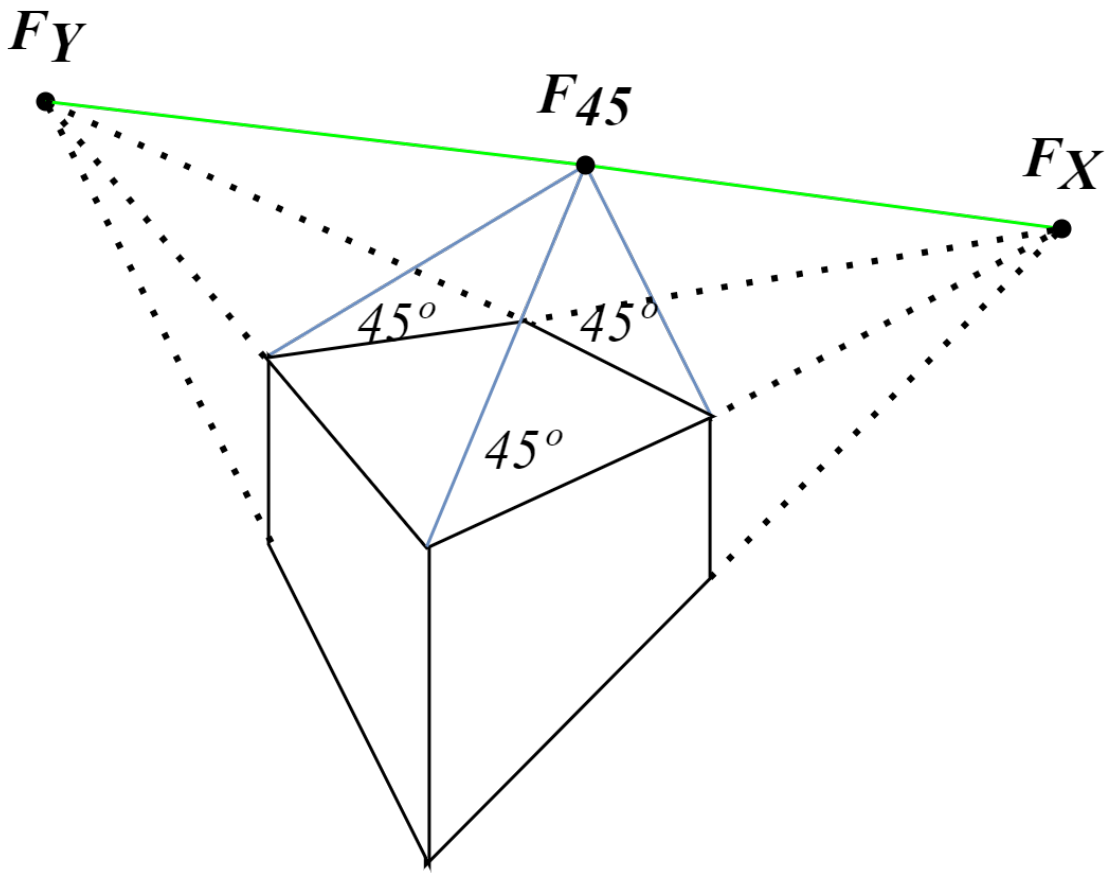


Figure 4 – Example of three blue segments parallel to axis $\boldsymbol{Z}$ with angle of 45 degrees with the other two axis. The vanishing point $\boldsymbol{F_{45}}$ lies on the green line. This green line is defined by the other two non-parallel axes vanishing points, $\boldsymbol{F_X}$ and $\boldsymbol{F_Y}$.

which specifies $\boldsymbol{P}$ coordinates. Hence, with the information of a point's unprojection belonging to a plane parallel to an axis, we can find its coordinates in the WCS by using its ICS coordinates. This process uses another point of this plane, its parallel axis, and a line segment orthogonal to this axis.

2.4 Planes Concatenation

The 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 these segments.

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

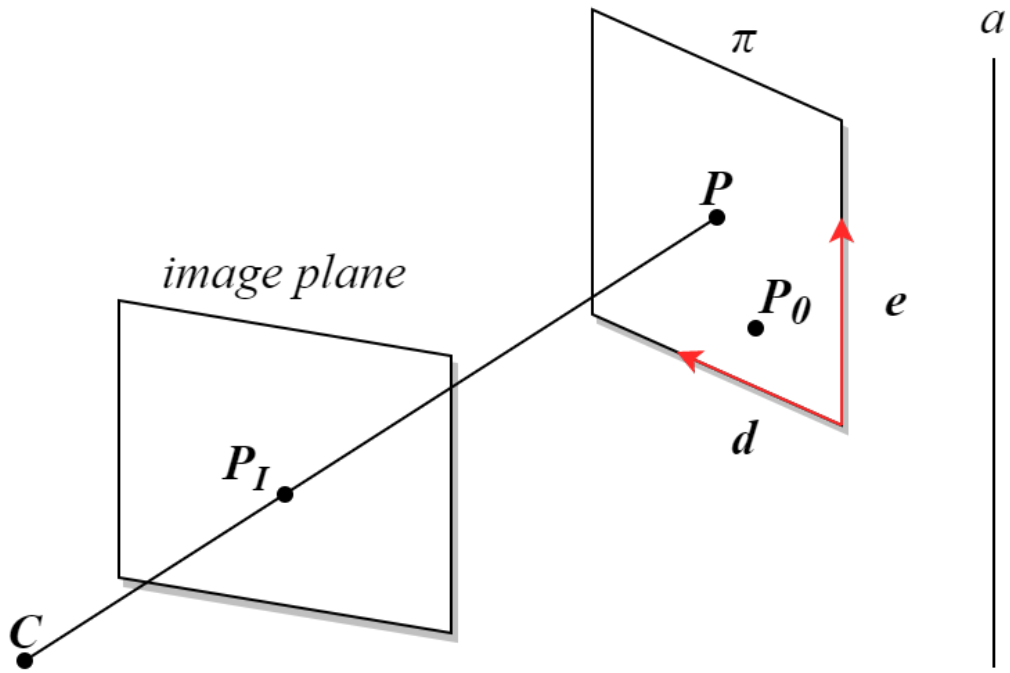


Figure 5 – The point $\boldsymbol{P_I}$ is known in the image and it is desired to find $\boldsymbol{P}$, its correspondent in the world. $\boldsymbol{P_0}$ and $\boldsymbol{P}$ belongs to π, which is parallel to the axis a. With the orthogonal vectors $\boldsymbol{d}$ and $\boldsymbol{e}$ the plane π is defined, allowing the calculation of $\boldsymbol{P}$ coordinates in the WCS.

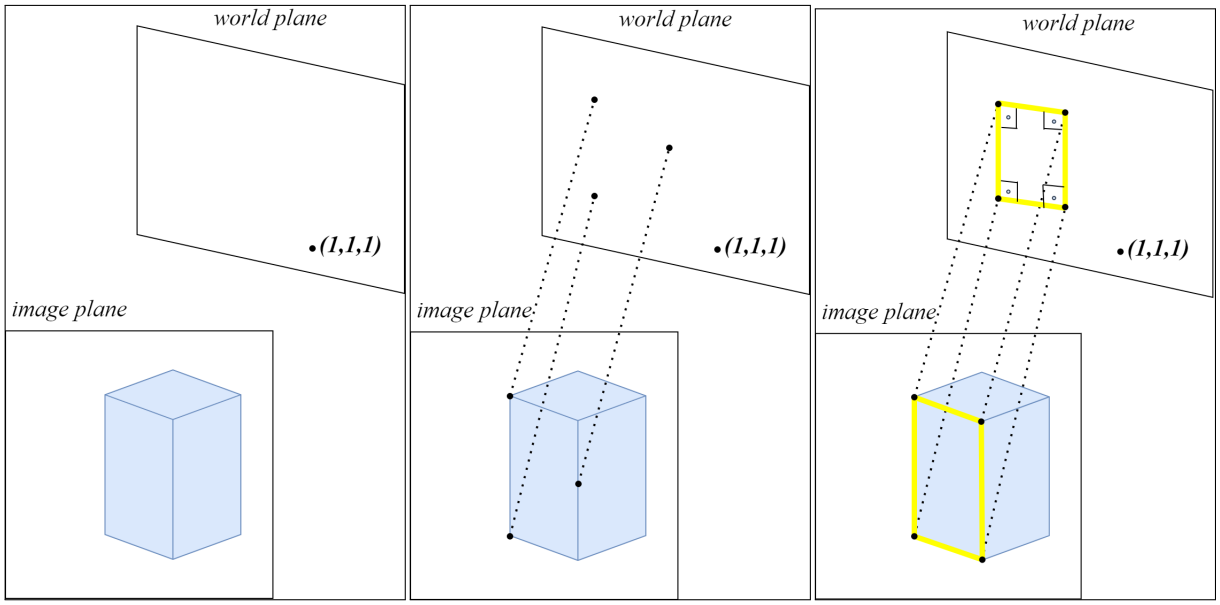
the plane type if needed.

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 plane type 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 only 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.

The indicated 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. This process for the first rectangle can be seen in Figure 6.

Adding other rectangles can be done in a similar way: consider the segment of the adjacent plane as the two initial vertices (with one of them being a point of the next plane) and another annotated extension point of the image. If the plane type of the next



(a) (b) (c)

Figure 6 – In (a) we have our desired object in the image plane and we know that its left face belongs to a known world plane (parallel to an axis). With this information, (b) shows their unprojection from the ICS to the WCS and (c) presents the final initial rectangle with its four corners.

rectangle is defined, we can follow the same previous steps.

2.5 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.

A partition is done to draft the texture in the rectangle area within the plane. Splitting it into $width \times height$ rectangles as in Figure 7, 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.

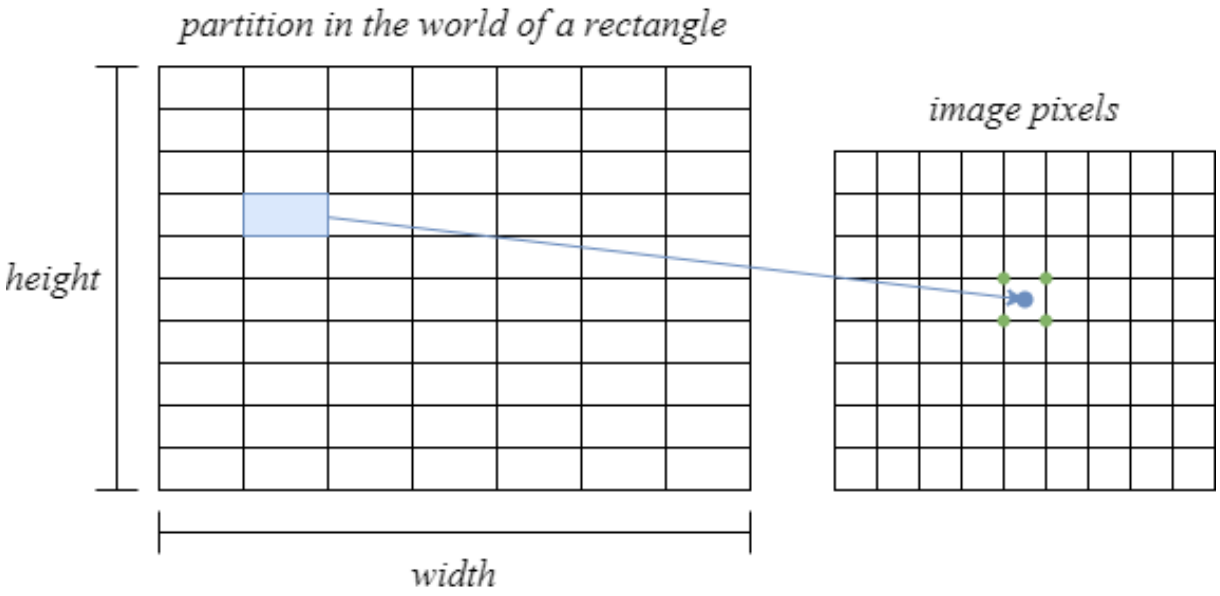


Figure 7 – The blue rectangle of the partition is projected in the image, resulting in decimal pixel value. To approximate its color, bilinear interpolation is done with the closest four points (represented by the green dots) using their RGBA values separately.

2.6 Adding a Scale

The created rectangles bound a certain area within the image. Any point inside of it can be unprojected following the plane type they belong to. Assume that we know the length p of a segment in the picture, defined by the projections of $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$, both projections contained within the region of one of the rectangles. With this information, it becomes possible to recover the scale of any desired segment on the model.

For example, one can find the height q of a door by annotating on the image the projection of two points, $\boldsymbol{Q_1}$ and $\boldsymbol{Q_2}$, representing the door side. After that, q is found by

$$q = \|\boldsymbol{Q_2} - \boldsymbol{Q_1}\| \frac{p}{\|\boldsymbol{P_2} - \boldsymbol{P_1}\|} \tag{2.13}$$

where we use the fact that each unit in the world is proportional to the scale of the known segment length.

3 Results and 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 parallel to two or more world axes. These are used to estimate the vanishing points of the axes directions and, consequently, the camera calibration. 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. After the model is finished, we can visualize it and add a scale to its dimensions. This workflow can be visualized in Figure 8.

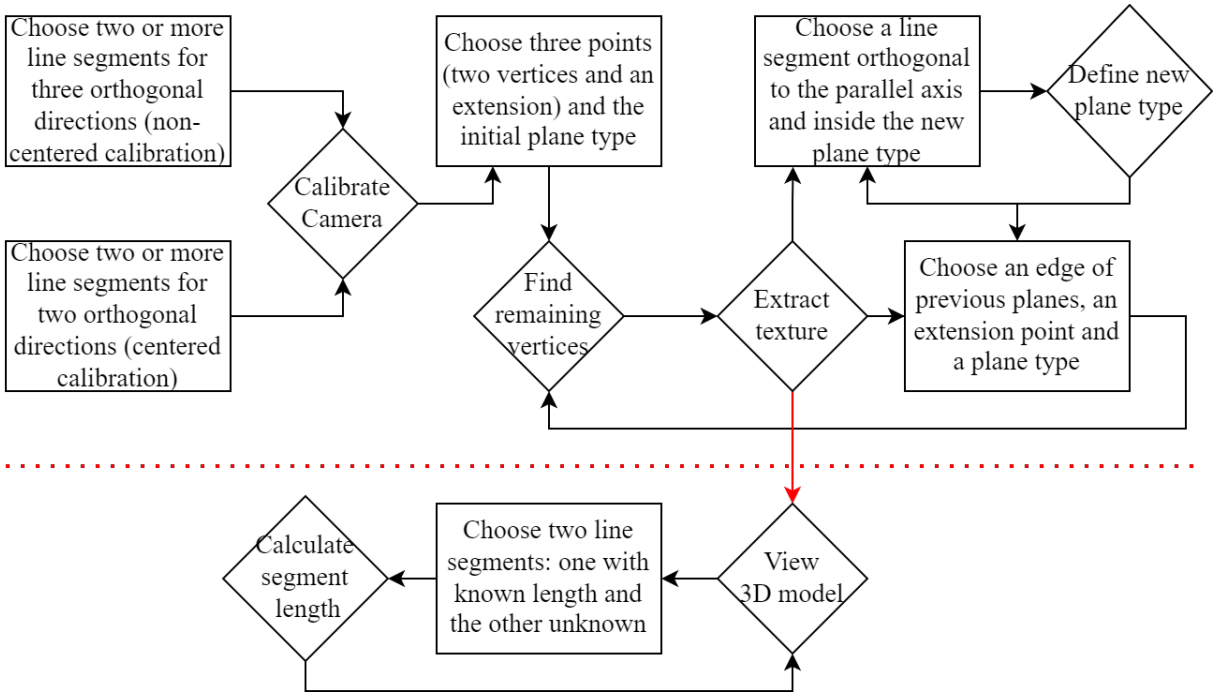


Figure 8 – Diagram showing the steps of creating the 3D model from a single image. Rectangles and rhombuses are inputs and internal calculations, respectively.

Four examples are given in Figure 9. These photos have two necessary characteristics: no lens distortion and two or more visible 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 $\boldsymbol{H}$ and new plane types orthogonal segments are defined by the pink segments;
- Stage (ii): the image superimposed by the reconstructed model, where the rectangles' projections are delimited by the yellow lines;

- Stage (iii): shows how the final model looks from a different 3D perspective of the camera;
- Stage (iv): speculating the length of an object dimension (brown segment), we can calculate a wanted height in the model (teal segment). Their values are shown in Table 1.

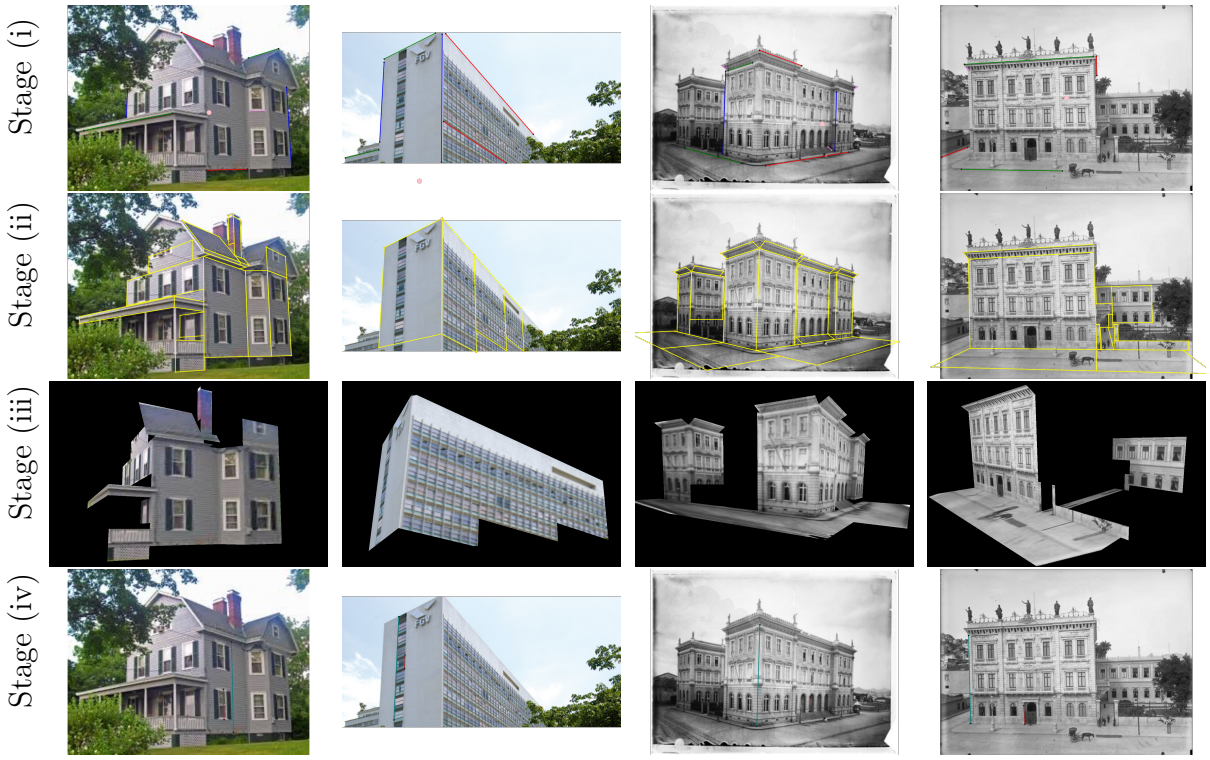
Many geometric aspects can be recovered from each model. Focusing only on Figure 9a, a non-centered calibration is used. The slanted wall angle with axis $\boldsymbol{X}$ is 0.57 radians or approximately 33 degrees. The same can be applied to the roof plane, having approximately 40 degrees angle with 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. We calculate the height of the side wall to be 4.7715m, given the estimated window height as 1.5m.

Changing our spotlight to Figure 9d, a centered calibration is employed by knowing only the vanishing points of axes $\boldsymbol{X}$ and $\boldsymbol{Y}$. Using the alley right next to the main edifice, we can build a path to the building on the back. This allows the model to be extended to its wall. Speculating the height of the front door to be 3m, we measure the height of the front building to be 14.924m.

For some images it may be difficult to obtain a precise calibration. This happens when the line segments for two or more axes of calibration are almost parallel, resulting in numerical problems when calculating the vanishing point. The limitation to rectangles 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.

	Segment length	
	Estimated brown segment	Model teal segment
Example A	1.5m	4.7715m
Example B	3m	23.2981m
Example C	2m	10.1843m
Example D	3m	14.924m

Table 1 – Value of the added scales to each model of Figure 9. By estimating the length of the brown segment at stage (iv), we can recover any desired segment length belonging to the model. In this case, the wanted segment length is represented by the teal color.



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

Figure 9 – Different cases of images in each stage of the process.

4 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. With a known segment length, the model acquires a scale, granting us the length of any desired object dimension. All of this is done through a set of specified points in the image.

The 3D model reconstructed can be visually inspected to be consistent with the building by superimposing the reconstructed model to the building's picture. Inheriting many intrinsic properties, the set of rectangles allows a good visualization of the scene. With additional data such as the correct width or height of a stated door or window, one can estimate sizes of objects and buildings with precision.

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.

References

ALVAREZ, Beatriz Silva Villa; CARVALHO, Paulo Cezar Pinto; GATTASS, Marcelo. 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. [S.l.: s.n.], 2002. P. 17–23.

BAI, K. et al. Survey of Learning Based Single Image Super-Resolution Reconstruction Technology. **Pattern Recognition and Image Analysis**, Pleiades Publishing Ltd, v. 30, n. 4, p. 567–577, Oct. 2020. DOI: 10.1134/s1054661820040045.

DEBEVEC, Paul E.; TAYLOR, Camillo J.; MALIK, Jitendra. 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. New York, NY, USA: Association for Computing Machinery, 1996. (SIGGRAPH '96), p. 11–20.

DORNELLES, T.; JUNG, C. Online frame-to-model pipeline to 3D reconstruction with depth cameras using RGB-D information. en. In: CONFERENCE on Graphics, Patterns and Images, 33. (SIBGRAPI), 2020, Virtual. Proceedings. Los Alamitos: IEEE Computer Society, 2020.

FERRANTI, Tomas. **TextureExtractor**. [S.l.]: GitHub, 2021. https://github.com/TomasFerranti/TextureExtractor.

GUILLOU, Erwan et al. Using Vanishing Points for Camera Calibration and Coarse 3D Reconstruction from A Single Image. **The Visual Computer**, v. 16, p. 396–410, Nov. 2000. DOI: 10.1007/PL00013394.

MAHMUD, Jisan et al. Boundary-Aware 3D Building Reconstruction From a Single Overhead Image. In: PROCEEDINGS of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], June 2020.

MUSIALSKI, Przemyslaw et al. A Survey of Urban Reconstruction. **Computer Graphics Forum**, v. 32, n. 6, p. 146–177, Sept. 2013. ISSN 01677055. DOI: 10.1111/cgf.12077.

OCHMANN, Sebastian et al. Automatic reconstruction of parametric building models from indoor point clouds. **Computers and Graphics**, v. 54, p. 94–103, 2016. Special Issue on CAD/Graphics 2015. ISSN 0097-8493. DOI: https://doi.org/10.1016/j.cag.2015.07.008.

ROTTENSTEINER, Franz et al. **The ISPRS benchmark on urban object classification and 3d building reconstruction**. en. [S.l.]: Göttingen : Copernicus GmbH, 2012. DOI: 10.15488/5042.

SUVEG, Ildiko; VOSSELMAN, George. Reconstruction of 3D building models from aerial images and maps. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 58, n. 3, p. 202–224, 2004. Integration of Geodata and Imagery for Automated Refinement and Update of Spatial Databases. ISSN 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2003.09.006.