Chapter

# Build 3D Abstractions with Wireframes

*Roi Santos Mateos, Xose M. Pardo and Xose R. Fdez-Vidal*

## Abstract

This chapter serves as an introduction to 3D representations of scenes or Structure From Motion (SfM) from straight line segments. Lines are frequently found in captures of man-made environments, and in nature are mixed with more organic shapes. The inclusion of straight lines in 3D representations provide structural information about the captured shapes and their limits, such as the intersection of planar structures. Line based SfM methods are not frequent in the literature due to the difficulty of detecting them reliably, their morphological changes under changes of perspective and the challenges inherent to finding correspondences of segments in images between the different views. Additionally, compared to points, lines add the dimensionalities carried by the line directions and lengths, which prevents the epipolar constraint to be valid along a straight line segment between two different views. This chapter introduces the geometrical relations which have to be exploited for SfM sketch or abstraction based on line segments, the optimization methods for its optimization, and how to compare the experimental results with Ground-Truth measurements.

**Keywords:** structure from motion, 3D abstraction, SLAM, 3D sketch, 3d reconstruction

## 1. Introduction

Most of the methods for environment abstraction from multiple views are just relying on points and ignoring other basic shapes like lines. Line based Structure from Motion methods based on lines create an abstraction based on straight line segments from a set of images. Analogously to point based abstraction methods like SIFT, in order to estimate the three-dimensional coordinates of lines in an spatial representation, the correspondences between lines among multiple images must be obtained by using detection and matching. The matching process for lines across the different views will return correspondences that can be exploited using 3D geometric relations. The matched features (points or lines) among views are used to estimate the position of cameras, referred to as extrinsic parameters. From the camera poses, the features are forward projected in the 3D abstraction or sketch.

The 3D line abstraction methods based on straight line segments that are most frequently found in the literature are designed to work altogether with detailed point-based reconstructions [1], therefore employing the camera extrinsic parameters obtained from these point rotation and translation invariants. This permits higher accuracy in the 3D reconstructions than using solely straight lines. A different approach employs only straight line segment correspondences in the reconstruction, independently of point based 3D reconstructions [2]. This approach has been proved advantageous over the first one in scenarios where not enought feature points can be accurately put in correspondence between the different views. There are few publications about uncertainty analysis in 3D line reconstructions based on lines. One of the most recent ones explains the state of the art for these metrics [3].

Oppositely to points, straight lines have a direction, and this dimensionality can be exploited geometrically. The intersection of coplanar straight lines reveal geometrical information. Likewise, groups of segments will also indicate the location of the most probable vanishing points from a camera plane [4]. These geometric properties are not offered by points, therefore lines can be a good complement when performing a spatial reconstruction [5, 6]. Additionally, pairs of straight line segments are often related by the strong constraints of parallelism and orthogonality [7, 8]. This allows to combine individual similarities of pairs of segments altogether with the coplanarity constraints [9]. A recent work employed 3D lines to reconstruct surfaces [10].

### 1.1 SfM lines carry higher complexity

In literature, research about straight segments have always been developed after works related to feature points. Lines have often been left as a complement for applications of these works devoted to feature points. There are reasons for the line based SfM to be more complex than a feature point based one:

1. Detection of points is restrained to sole coordinates in images, while line detection extends to several pixels that are ideally adjacent to other pixels of the line. Nevertheless, in practice, detecting the limits of a straight line segment is not trivial in real images, due to digital noise, occlusions or changes in illumination. **Algorithms describing different continuity criteria must be employed in order to obtain a reliable edge detection in an image**. Moreover, as an straight line means a special case of an edge, detected edges have to be fit to straight lines. Fitting edges to straight segments can be accomplished by applying linear regression for the points comprising an edge in the image. Finally, the method has to find the endpoints of straight line segments, accounting for fragmentation or occlusions.

2. A set of pictures of the same scene may feature different kinds of viewpoint changes among captures, including camera rotations, zooms and translations. These **changes in the camera viewpoint produce a morphological transformation of the primitives in the captured frame**, which translates into displacements of the detected primitives, changes on their shape, distortions, fragmentation or even the impossibility to detect the same primitive in another image by employing the same operations that served to detect it in one of the pictures. Some of these transformations are not applicable to points, for instance a fragmentation: A point is either fully present or not, but it should not be such a thing as a detected fragmented point. Therefore, **there are more morphologic transformations that can affect 2D line segments than the ones that can affect points, due to camera viewpoint change**. Generally, prominent viewpoint transformations increase the difficulty in matching primitives, because the greater the transformation of the same primitives among different images of the scene, the greater the difficulty to match them.

3. Matching primitives between images is not always accurate, specially when dealing with line segments. When finding counterparts for primitives detected in other images, it is common to come out with several mismatched primitives. These wrongly matched primitives are referred to as **matching outliers** [11]. Matching outliers can produce that the description of the structure of groups of primitives can not be correctly compared to others, and employing inaccurate structure descriptions to propagate the matching to other images may cause problems when computing the final 3D abstraction. Some of the sources of the difficulties matching lines are because **line segments are subject to more morphological transformations than points. The description of individual lines are therefore more subject to these transformations, and less truthfully at the end. This fact forces line matching methods to rely more on structure of neighborhoods than points.** The accuracy of the description of these neighborhoods compared with the real morphological transformation of the lines it comprises are highly dependent on the ratio of matching outliers.

4. In the frame of 3D reconstruction from relations between feature points, known the relative position of two cameras and the position of one point on the first image, **there is a constraint that forces the counterpart of this point on the second image to lay on a line. It is called epipolar constraint. But a single infinite 2D line represented in two images does not feature epipolar constraint. The only point-to-point valid correspondences in matched segments under a viewpoint change are their endpoints.** For this case of a line segment, in order to estimate its position in 3D, is required to detect in the images both endpoints of the line segment. In some cases it may be difficult to accurately detect the end of a line segment in an image. For instance, a segment can end by merging with another edge under a different slope, progressively dimming until it vanishes, by intermittent occlusions, or being abruptly fragmented. Moreover, one or both segment endpoints may lay in the limits of the frame, and in this case it will not be possible to extract the 3D pose of the line.

The above mentioned tasks portrait the main differences between lines and points raised out during the engineering of a complete line-based 3D sketch generation method from images. For each stage of the method,specific tasks and problems have to be solved in the state-of-the art: detection of borders, matching lines over pairs of views, comparing the line matching performance against competition, relate the matched primitives among sets of more than two images, estimation of spatial lines, optimizing the abstraction and exploiting the resulting 3D structure.

## 2. Estimate 3D straight line segments

A 3D line can be thought as a multi-view entity that relates a perceivable line segment in the real world to its counterparts in images, given that these have been correctly detected and matched. The process of generating a 3D representation from different pictures of the scene is visually represented in **Figure 1**.

For the SfM problem, the poses of the cameras that took the pictures are not provided, and it is up to the SfM algorithm to simultaneously estimate the poses for the cameras and primitives. In the present case, SfM has to estimate the pose of the lines in space, relative to the cameras. The first requirement for the method is the calibration matrix $K$ for each camera, which provides the transformation between

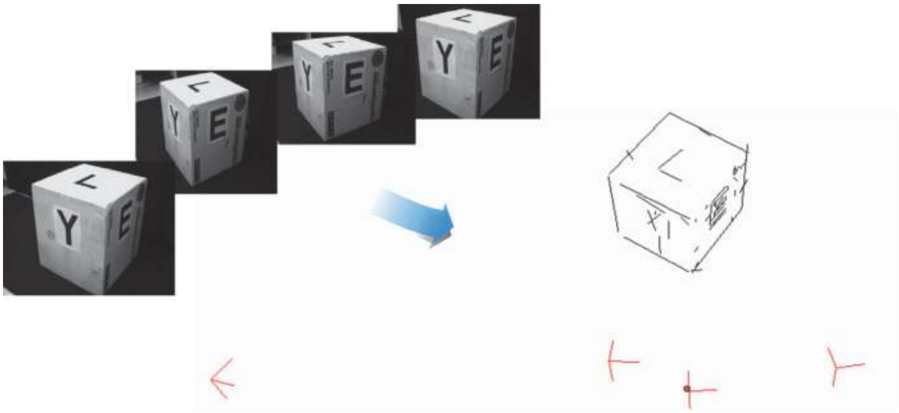


**Figure 1.**
*Visual representation from [2]. It depicts the challenge of converting a set of 4 pictures into a 3D sketch featuring the line segments and camera axis. The 4 cameras are represented as three axis reference frame in red.*

each point in one image, in homogeneous coordinates, to a ray in Euclidean three-dimensional space. Secondly, SfM has to estimate the projection matrices $P$ for the cameras, representing a map from 3D to 2D:

$$\mathbf{x} = \mathbf{P}\mathbf{X}, \tag{1}$$

where $\mathbf{x}$ is a 2D point on the image, and $\mathbf{X}$ its projection in 3D space. $\mathbf{K}$ is intrisic to each camera, while $\mathbf{P}$ is extrinsic and embeds the 3D translation and rotation of the camera's image plane. The estimated translation is valid up to scale.

A common space can be built to host the cameras and spatial lines. For this new common space the camera that took the first processed picture takes the place of the origin, and for the rest of cameras $\mathbf{P}$ can be estimated from the lines matched between the captured images. Alternatively, camera poses can also be retrieved from a feature-point based SfM pipeline and these cameras be employed for the estimation of spatial lines. For instance, the feature-point descriptor SIFT [5] can be used to match points in images with a low ratio of outliers. These feature point relations are obtained both in the foreground and background. A set of relations between points or lines in two images allows to estimate the homography constraints between both views by applying the 5-point algorithm [12] using the points or the segment endpoints. A purge of outliers can be performed employing RANSAC [13] for robust estimation. Therefore, a set of stereo 3D projections is obtained combining the available images pairwisely, and each stereo system featuring both camera poses and a point cloud. The objective is to obtain an unique 3D point cloud sketch, embedding all cameras and point matches. Hence, camera poses are sequentially stacked, relative to each other, in the new spatial reference frame. And the 3D estimations for the feature points in the new 3D space can be computed as the center of gravity for their position relative to the common camera in both stereo systems. Finally a sparse bundle adjustment [14] is used to minimize the pixel distance of the back-projected 3D point and the original observation of this point on each image in homogeneous coordinates. These reprojection errors on the planes of the cameras are minimized employing the Levenberg-Marquardt algorithm. The resulting keypoint-based 3D reconstruction contains the optimized 3D estimations for the cameras and the point cloud.

Several straight segment matching methods are based on texture descriptors [15, 16], coloring [17] or in keypoint-line projective invariants [18, 19]. Under these conditions, matching results will be influenced by the level of texture in the images. In the case that a low number of detected segments can be distinguished by employing image texture based descriptor, or in case that a low number of feature points are identified throughout the set of images, the resulting set of matched lines will not be satisfactory. On the other hand, if line matching is rooted on weak epipolar constraints [1], line matching will be highly dependent on the accuracy of the camera poses.

Extrinsic parameters for cameras are needed to project the matched lines into space. Having the same segment completely detected and without fragmentation for both views under viewpoint change, endpoints are the only points in a segment with known exact counterpart in the other image. Unfortunately, segment detection is not accurate in the location of the endpoints. Therefore, the most accurate abstractions will be the ones built rooted on camera extrinsics obtained from a dense feature point based SfM. As written above, known the projection matrices $P$ of two cameras, a point on an image projects as a 3D ray in Euclidean space. And this 3D ray projects like an infinite 2D line on any plane different than the one that contains the point. Therefore, each 3D point $X_p$ will have its image into an epipolar line $e_p$ contained in the image. As the unknown point is constrained into a line in the other image plane, analogously a segment will be constrained between both epipolar lines corresponding to the segment endpoints. This weak epipolar constraint can be employed for matching segments between images [1].

## 3. Geometric relations

A *3D abstraction method* estimates the position of 3D line segments $\Gamma = \{\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_N\}$, from an unordered sequence of images, taking from cameras with planes $\Upsilon = \{\Upsilon^1,\Upsilon^2,\Upsilon^3,\ldots,\Upsilon^M\}$. Straight lines are detected in the original images, put in correspondence among them, forward projected into space, and rewritten in homogeneous coordinates.

The 3D line based sketch $\{\Upsilon,\Gamma\}$ is built from the knowledge of correspondences among line projections $l$ on camera planes, and the intrinsics of all the cameras. The following paragraphs explain the linear triangulation of these observations, as performed from scale-space images. This allows to discriminate and weight down lines that have been detected on two or more scales with a different slope. The practical consequence is that prior to any 3D extrapolation of the observed lines, matching inliers with inconsistent endpoint location among scales on both images can be avoided, as these lines might introduce uncertainty in the estimation for the pose of the camera.

The camera poses $\mathbf{P}$ are estimated from the endpoint correspondences of $l$. The Essential matrix $\mathbf{E}$ is computed from the camera pairs, by using the Five-Point Algorithm [12], and RANSAC [13] for hypotheses generation. Having $\mathbf{E}$ and $l$, the relative camera rotation and translation among the first pair of cameras $\boldsymbol{P}^j = [\mathbf{R}|\mathbf{t}]$ are estimated using cheirality check and discarding the triangulated endpoints that are not in front of the cameras. The left camera is chosen to have the pose $\boldsymbol{P}^1 = [\mathbf{I}|0]$, and the newly added cameras are stacked from this position in the unique reference frame.

The forward projection of lines in 3-space is described in the page 196 of Hartley and Zisserman's book [20]. The 3D forward projection $\boldsymbol{\Gamma}_\mathbf{i}$ of a line, bundled in the same reference frame, can be obtained using the *DLT* method on the set of stereo 3D camera back-projections. This is performed in homogeneous coordinates because it allows to consider line endpoints in the infinite. Therefore, from now on, when a 2D point is mentioned it will be supposed homogeneous coordinates. There exists a $3 \times 3$ matrix $\mathbf{E}$, known as the essential matrix, such that if $u$ and $u'$ are a pair of matched points, then $u'\mathbf{E}u = 0$. If a sufficient number of matched points are known, the matrix $\mathbf{E}$ may be computed as the solution of an overdetermined set of linear equations. For the present problem, the internal calibration of the cameras is known, therefore it is possible to determine from $\mathbf{E}$ the relative placement of the cameras and hence the relative locations of the 3D points corresponding to the matched points. A linear triangulation method is projective-invariant because only camera and line distances are minimized.

The above described DLT method for lines starts with the segments on the pair of cameras $\left\{\Upsilon^a, \Upsilon^b\right\}$ with the highest inlier ratio. Based on this first triangulation, the other cameras are appended to the 3D abstraction: The next camera $\Upsilon^c$ is chosen according to the higher inlier ratio of line matching with $\Upsilon^a$ and $\Upsilon^b$. Analogously, the following camera $\Upsilon^n$ is picked among the ones with the higher inlier ratio of line matching with previously selected cameras. The detection of 2D lines $l$ in the original images carry an uncertainty for the position of these observations. This uncertainty implies that no 3D point $\boldsymbol{X}$ will satisfy that their projections on cameras $\Upsilon^1$ and $\Upsilon^2$ are $\boldsymbol{x_1} = \boldsymbol{P}^1\boldsymbol{X}$, $\boldsymbol{x_2} = \boldsymbol{P}^2\boldsymbol{X}$ respectively. Moreover, the image points do not satisfy the epipolar constraint $\boldsymbol{x_2}\boldsymbol{F}\boldsymbol{x_1} = 0$. Therefore, a method that only minimizes the distances on the image from the estimations to the observations is required: A projective-invariant triangulation method. A linear triangulation [20] method does not depend on the projective frame in which $\boldsymbol{X}$ is defined.

The forward projection from a normalized 2D line observed on the camera image plane $m$, denoted by $\boldsymbol{l}_{\boldsymbol{i}}^{\boldsymbol{m}}$, is the plane $\boldsymbol{P_m}^T\boldsymbol{l}_{\boldsymbol{i}}^{\boldsymbol{m}}$, so the condition for a point $\boldsymbol{X}_a$ to be in this plane is:

$$\left(\boldsymbol{l}_{\boldsymbol{i}}^{\boldsymbol{m}}\right)^T \boldsymbol{P_m}\boldsymbol{X}_a = 0. \tag{2}$$

Each point $\boldsymbol{X}_a$ returns a linear equation in the entries of $\boldsymbol{P_m}$. Denoting by $\boldsymbol{x}_{m,E}^i$ and $\boldsymbol{x}_{m,F}^i$ the forward projection of the endpoints of $\boldsymbol{l}_m^i$, named $\boldsymbol{X}_E^i$ and $\boldsymbol{X}_F^i$, under $\boldsymbol{P}_m$, then any other 3D point on the line $\boldsymbol{X^i}(\mu) = \boldsymbol{X}_E^i + \mu\boldsymbol{X}_F^i$ projects to a point:

$$\boldsymbol{x}_i^m(\mu) = \boldsymbol{P}_m(\boldsymbol{X}_{i,E} + \mu\boldsymbol{X}_{i,F}) = \boldsymbol{x}_{i,E}^m + \mu\boldsymbol{x}_{i,F}^m, \tag{3}$$

which is on the line segment $\boldsymbol{l}_{\boldsymbol{i}}^{\boldsymbol{m}}$.

In the described method, an unique reference frame is built. The world reference system is fixed onto the first camera, hence its camera matrix, $\boldsymbol{P_E}$, is computed with $\boldsymbol{R_E} = \boldsymbol{I}$ and $\boldsymbol{T_E} = \boldsymbol{0}$. The extrinsics for the partner camera $\boldsymbol{P_m}$ on the baseline is obtained from the essential matrix by using RANSAC. Before the subsequent DLT triangulations with a new camera, its extrinsics are estimated also by RANSAC from the 2D-3D results of the already computed DLT. From here, new cameras will be added incrementally, just one per DLT iteration, in order to avoid DLTs between two uninitialized camera projection matrices.

For DLT it is required a set of observed line correspondences, $\boldsymbol{l}_j^m$ to $\boldsymbol{l}_j^n$, matched among images. The projection on the image plane of camera $m$ of an endpoint $\boldsymbol{X}_{i,E}$ of the spatial line $\boldsymbol{\Gamma}_j$ is denoted as $\boldsymbol{x}_{j,E}^m = \boldsymbol{P}_m\boldsymbol{X}_{\ j,E}$. This point on the $m$-th camera plane is matched to its counterpart on the $n$-th camera $\boldsymbol{x}_{j,E}^n = \boldsymbol{P}_n\boldsymbol{X}_{\ j,E}$. Both equations can be combined into $\boldsymbol{AX}_{\ j,E} = 0$, where $\mathbf{A}$ is the matrix of equation coefficients. It is built from the matrix rows $\boldsymbol{A}_r$ contributed from each correspondence, whose resemble the movement of each line between both views. $\boldsymbol{X_{j,E}}$ contains the unknowns for the endpoint position.

By using the cross product on the $m$-th camera: $\boldsymbol{l}_j^m \times \left(\boldsymbol{P}_m\boldsymbol{X}_{\ j,E}\right) = 0$,

$$x_m\left(\boldsymbol{p}_{\boldsymbol{m}}^{\boldsymbol{3T}}\boldsymbol{X}_{\boldsymbol{j,E}}\right) - \left(\boldsymbol{p}_{\boldsymbol{m}}^{\boldsymbol{1T}}\boldsymbol{X}_{\boldsymbol{j,E}}\right) = 0, \tag{4}$$

$$y_m\left(\boldsymbol{p}_m^{3T}\boldsymbol{X}_{\ j,E}\right) - \left(\boldsymbol{p}_m^{2T}\boldsymbol{X}_{\ j,E}\right) = 0, \tag{5}$$

$$x_m\left(\boldsymbol{p}_m^{2T}\boldsymbol{X}_{\ j,E}\right) - y_m\left(\boldsymbol{p}^{1T}\boldsymbol{X}_{\ j,E}\right) = 0 \tag{6}$$

where $(x_m, y_m)$ and $(x_n, y_n)$ are the coordinates of $\boldsymbol{x}^m_{j,E}$ and $\boldsymbol{x}^n_{\ j,E}$ respectively. $\boldsymbol{p}_m^{rT}$ is the $r$-th row of $\boldsymbol{P}_m$. It can be decomposed similarly for $\boldsymbol{P}_n$, and compose the equation of the form $\boldsymbol{AX}_{\ j,E} = 0$. Solving:

$$\boldsymbol{A} = \begin{bmatrix} x_m\boldsymbol{p}_m^{3T} - \boldsymbol{p}_{\boldsymbol{m}}^{\ \ 1T} \\ y_m\boldsymbol{p}_m^{3T} - \boldsymbol{p}_m^{2T} \\ x_n\boldsymbol{p}_n^{3T} - \boldsymbol{p}_n^{1T} \\ y_n\boldsymbol{p}_n^{3T} - \boldsymbol{p}_n^{2T} \end{bmatrix}. \tag{7}$$

The solution for the 4 equations of the over-determined problem (four equations for four homogeneous variables) is only valid up to scale. The set of points in space mapping to a 3D line $\boldsymbol{\Gamma_j}$ via $\boldsymbol{P_m}$, is the plane $\boldsymbol{P}_m\boldsymbol{\Gamma}_j$.

The result of the linear triangulation process is $\boldsymbol{\Gamma}_{\mathbf{i}}$ and $u^{\boldsymbol{j}}$, represented in cartesian coordinates.

Every 3D segment $\Gamma_i$ is estimated as the center of gravity of the estimations for the same line for each par of images. The set of line projections observed in $\Upsilon$ is represented as $l$ = $\{l_1^1, l_2^1, ..., l_N^1, ..., l_N^M\}$. A Line Feature is defined as a subgroup of projections from $l$ of the same 3D line $\Gamma_i$. The Line Features are noted as $L$ = $\{L_1, L_2, ..., L_N\}$. The 3D lines $\Gamma$ are obtained by forward projecting the endpoints of $l$ from pairs of camera planes of $\Upsilon$, by using linear triangulation, analogously to Direct Linear Transformation (*DLT*) [20]. The cameras $\Upsilon$ are sequentially bundled in the same reference frame. The new ones are stacked according to the $L$-to-$\Gamma$ correspondences, computed in the previous stereo pair of cameras. The merged estimations for 3D lines $\{\Gamma_i\}$ are computed as the center of gravity of the spatial lines.

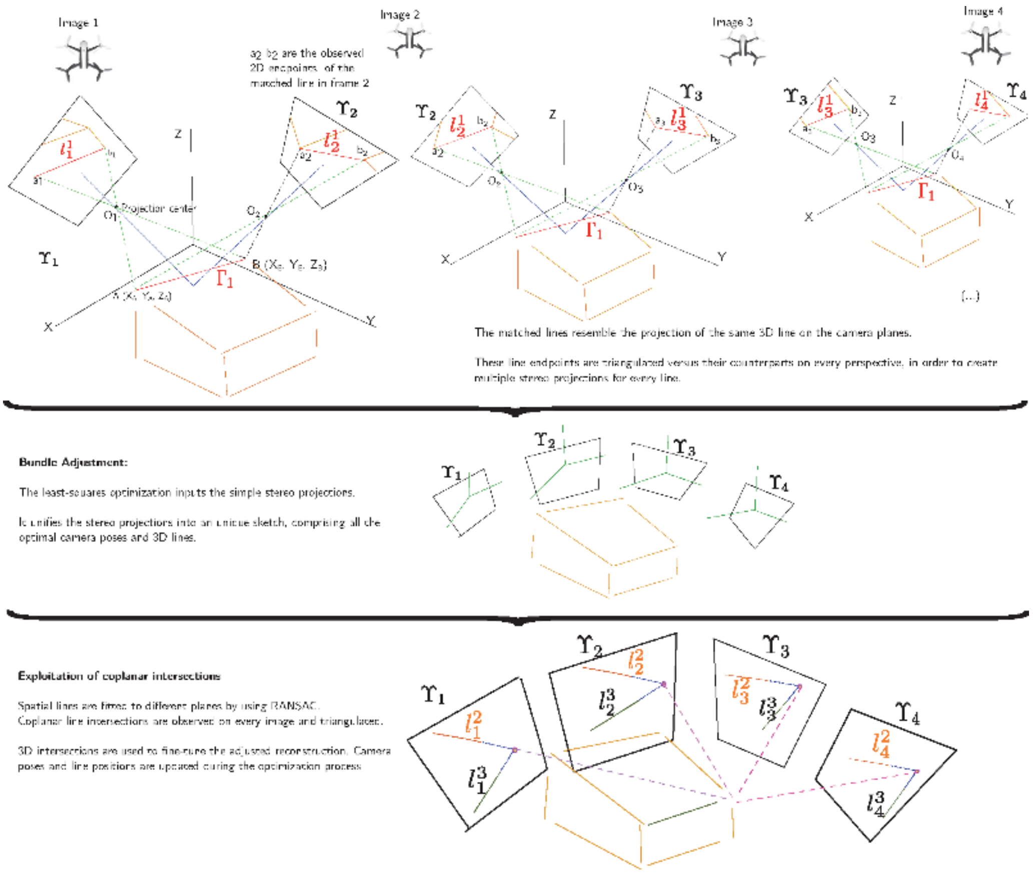
The 3D sketch $\{\Upsilon, \Gamma\}$ generated by linear triangulation is used as input for an optimization algorithm. The least-squares optimization named Sparse Bundle Adjustment (SBA) [14] is based on the Levenberg–Marquardt algorithm, and uses as input the estimated camera extrinsics $\Upsilon$ and the set $\Gamma$, now containing unique estimations for each 3D line [21].

The 3D estimations for lines and cameras are drawn in the same spatial sketch, altogether with the cameras. Next, these spatial line segments $\Gamma$ are fit to different different planes $\mathcal{P}$. $\Gamma$ is therefore segmented into different groups according to the planes $\mathcal{P}$, and so is done with their projections $L$. The group of Line Features fitted to the plane $\mathcal{P}_t$ is noted as $F_t$. The intersections of the coplanar lines $F_t$ on the camera plane $\Upsilon^j$ are the spatial points $\mathcal{T}_t^j$. Therefore, the algorithm can go back to the original images, now known which line segments are coplanar. The intersections of these coplanar lines on the images are described similarly as a feature point. Following this analogy, the descriptor for this feature point will be the pair of two coplanar lines drawing it. We have the correspondences of the straight lines accross images, so we can extrapolate these correspondences to their intersection for the cases where they are coplanar. Secondly, known the correspondences between these intersections, they can be triangulated analogously as it was performed in the first routine with the endpoints of $l$. The correspondences in $\mathcal{T}_t^j$ are then fed into the linear triangulation algorithm, in order to create initial estimates for the 3D intersections by forward projecting $\mathcal{T}_t^j$. The set of estimations for the 3D points resembling the intersections is a sparse cloud, and it is denoted as $\mathcal{R}$. Finally, and same as with the endpoints, the 3D intersections $\mathcal{R}$ enter the least-squares optimization. The SBA returns the new optimized estimations for $\Upsilon$, and the optimal 3D intersections $\mathcal{R}$. The spatial line and camera pose estimations are corrected by forward projecting them from the newly estimated camera planes $\Upsilon$. This returns the final sketch $\{\Upsilon,\Gamma\}$. The high level diagram on **Figure 2** shows the process described in this section.

### 3.1 Bundle adjustment for line segments

In the case of feature points, the final position of the projected features relative to the camera poses is estimated throughout an optimization process. As a part of most SfM pipelines, bundle adjustment [14] is based on Levenberg-Marquardt, and it rearranges the poses of the cameras and 3D points. The cost function of this optimization process is engineered to find the minimum distance error between the reprojection of every 3D point onto each camera plane and their original observation. A limit value for the residual is usually set to stop the iterative process for the event of convergence, while another threshold is set to end the optimization when reaching a maximum number of iterations.

Along matched segments under a viewpoint change, the only point-to-point valid correspondences are their endpoints. Segment's endpoint location are noticeably less accurate than a rotation and scale invariant feature point. Employing line endpoints as the sole set of geometrical constraints in the adjustment might not be



The matched lines resemble the projection of the same 3D line on the camera planes.

These line endpoints are triangulated versus their counterparts on every perspective, in order to create multiple stereo projections for every line.

**Bundle Adjustment:**

The least-squares optimization inputs the simple stereo projections.

It unifies the stereo projections into an unique sketch, comprising all the optimal camera poses and 3D lines.

**Exploitation of coplanar intersections**

Spatial lines are fitted to different planes by using RANSAC.
Coplanar line intersections are observed on every image and triangulated.

3D intersections are used to fine-tune the adjusted reconstruction. Camera poses and line positions are updated during the optimization process

**Figure 2.**
*Figure from [2]. Graphic representation of the 3D abstraction layer of the method. The different cameras are represented as drones.*

adequate to improve the 3D sketch. Some of the reasons for this are that recurrent segment mismatches, fragmentation or the inaccurate placement of counterparts may prevent the convergence of the optimization. It is possible to perform a line-based Bundle Adjustment by converting the primitives into Plücker coordinates [20, 21] within the cost function of the optimisation process. This allows a reduction in the number of parameters and the computational cost.

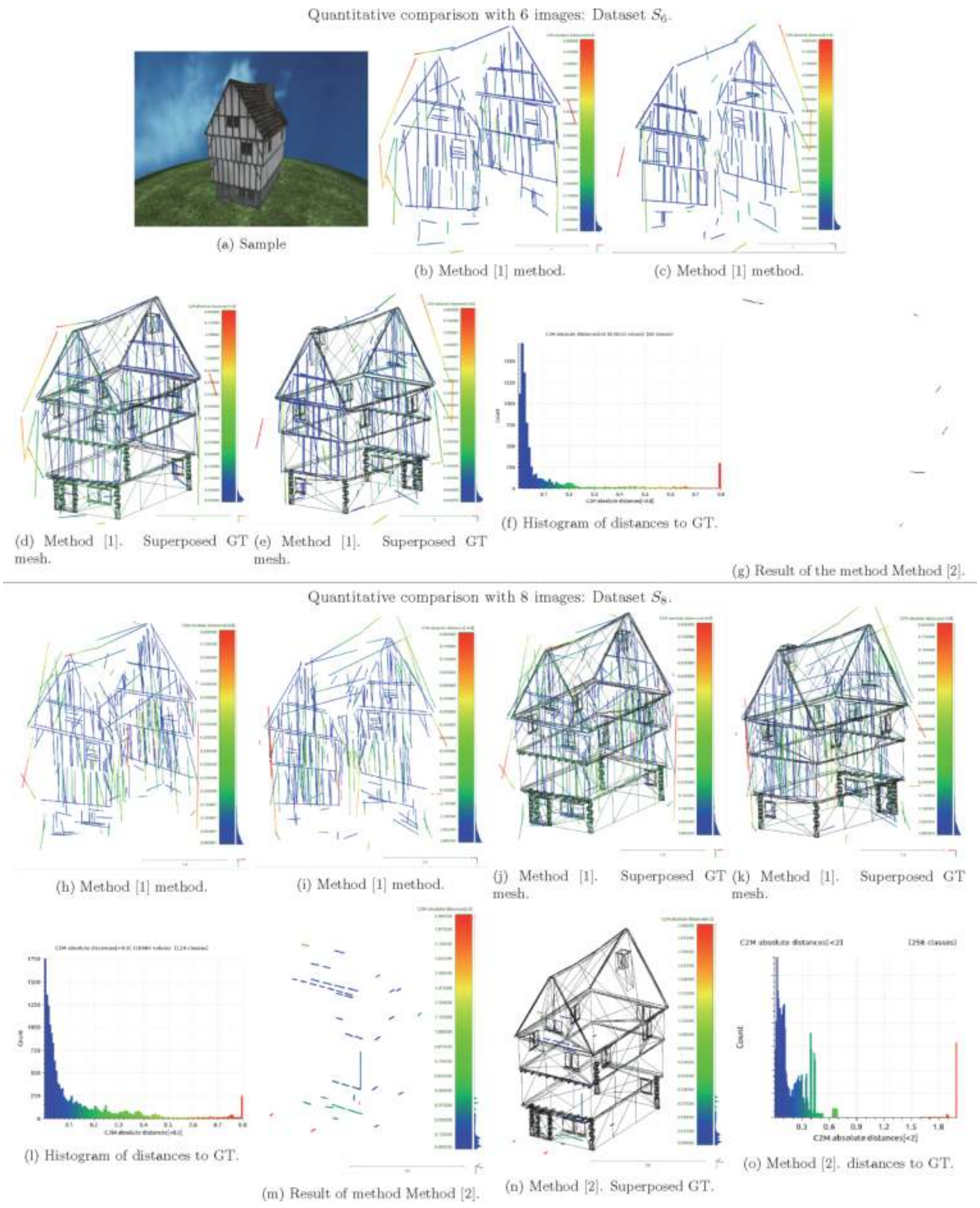
### 3.2 How to compare the results with Ground Truth meshes

In order to prove the validity of a 3D abstraction method, it has to be benchmarked against a Ground Truth dataset for SfM, which includes both intrinsic and extrinsic parameters for the cameras. These are built with synthetic images from 3D models [22], or with real pictures [23] teamed with 3D model data including the pose of the cameras and the measurements from 3D scanning or Lidar. Both synthetic and real Ground-Truth datasets include a 3D model. The resulting point cloud is aligned with the Ground Truth mesh. The normal distance between the surface of the mesh and the points is computed. In order to assess how the generated sketch fits the Ground Truth model, the Mean Square Error of the distance between both spatial shapes is computed, because it acts as the natural loss function of a Gaussian distribution. In the case of 3D line sketch, in order to compare the sketch with the Ground truth mesh, the 3D straight segments must be discretized into points. To measure the difference in proportions between the generated 3D sketch and the Ground Truth mesh, the normal distance between the surface of the mesh and the discretized points on the lines is computed. Using the obtained error in the distances, discretized points on the lines are coloured to account how far they are from the surface of the mesh. There are several variables that condition the resulting 3D sketch number of images: Firstly, the number of images showing common elements of the scene is one of them. Secondly, the number of segments that can be matched between images. Thirdly, the transformation between both images might condition the matching inlier ratio, and hence, the number of segments correctly projected into space.

For 3D line sketching methods, the length of the final 3D lines will depend on the fragmentation of the detected lines, and its number is closely related to the number of line correspondences between the images. Therefore, results of 3D reconstructions will unavoidably depend on the performance of the method for stages before the spatial projection. Quantitative measurements for 3D abstraction are performed on Ground Truth datasets. The proportions of the generated sketch is measured based on the distance between the segments and the Ground Truth mesh.

Employing a feature-point based abstraction method is profitable for datasets with a sufficient number of pictures featuring textured surfaces, so a dense 3D point cloud can be created. For these 3D abstractions, cameras are located accurately due to the precision of the point rotation and translation invariants. This is the case of the results obtained by abstraction methods working altogether with SIFT pipelines [1, 22], but requiring dozens of high definition pictures with textured surfaces for SIFT to be able to accurately estimate the camera extrinsics.

There are real world applications of Computer Vision that does not always permit to obtain high definition pictures, in textured environments, without blurring and digital noise. For these applications it can be advantageous to estimate the camera extrinsics independently of any feature point 3D reconstruction [2]. **Figure 3** shows a quantitative comparison of the methods [2] and [1] with just 6 and 8 images chosen from the dataset. **Figure 4** increases the number of images to 10 and 12. The test cases are labeled as $S_6$, comprising image numbers


Quantitative comparison with 6 images: Dataset $S_6$.
Quantitative comparison with 8 images: Dataset $S_8$.
C2M absolute distances[<2]
[258 classes]
0.3 0.6 0.9 1.2 1.5 1.8
C2M absolute distances[<2]
Count

(a) Sample

(b) Method [1] method.

(c) Method [1] method.

(d) Method [1]. Superposed GT mesh.

(e) Method [1]. Superposed GT mesh.

(f) Histogram of distances to GT.

(g) Result of the method Method [2].

(h) Method [1] method.

(i) Method [1] method.

(j) Method [1]. Superposed GT mesh.

(k) Method [1]. Superposed GT mesh.

(l) Histogram of distances to GT.

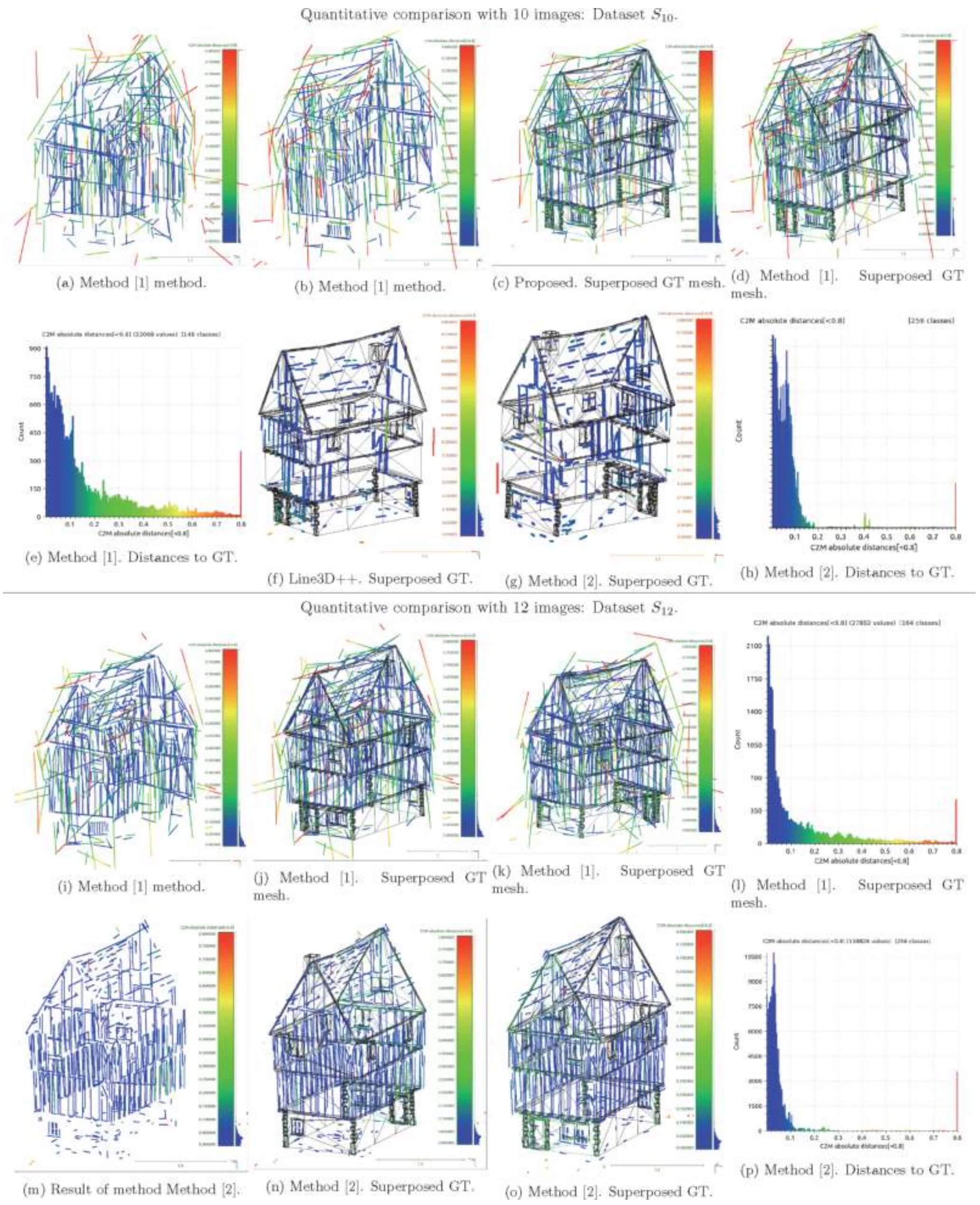
(m) Result of method Method [2].

(n) Method [2]. Superposed GT.

(o) Method [2]. distances to GT.

**Figure 3.**
*Figure from [2]. Quantitative comparison using the sets $S_6$ and $S_8$. This figure is better viewed on a screen with a 4x zoom. (a) Sample of the set. (b) and (c) [2] against $S_6$, resulting in 175 lines. (d) and (e) Same superposed onto the Ground Truth mesh. (f) Histogram of distances to Ground Truth with [1] method. The maximum distance to be accounted, is set to be 0.8, already considered as outlier. (g) Sparse atomic lines returned by [1] method. (h) to (l) [1] against the set $S_8$, with 294 segments. (m) and (n) same measurements for the result by [1]. (o) shows the histogram for this latter result.*

{6,9,86,46,49,126} from [22], $S_8$ further add two more images {89,129} to the list, $S_{10}$ includes {8,10,12,88,90,48,50,52,128,130}, and $S_{12}$ further adds images {92,132} to the latter. The resulting 3D line sketches from both sides of the house are aligned by using common lines. This completed sketch is finally aligned to the Ground Truth in order to measure the precision. Note that this experiment takes into account just a the variation of the number of images in the dataset [22]. The results show that the method [2] obtains more usable results for a low number of images, and the results of method [1] are only more adequate than method [2] when the number of images gets close to a dozen. The spatial lines are colored attending to its distance to the surface of the Ground Truth mesh.



Quantitative comparison with 10 images: Dataset $S_{10}$.

(a) Method [1] method. (b) Method [1] method. (c) Proposed. Superposed GT mesh. (d) Method [1]. Superposed GT mesh.

(e) Method [1]. Distances to GT. (f) Line3D++. Superposed GT. (g) Method [2]. Superposed GT. (h) Method [2]. Distances to GT.

Quantitative comparison with 12 images: Dataset $S_{12}$.

(i) Method [1] method. (j) Method [1]. Superposed GT mesh. (k) Method [1]. Superposed GT mesh. (l) Method [1]. Superposed GT mesh.

(m) Result of method Method [2]. (n) Method [2]. Superposed GT. (o) Method [2]. Superposed GT. (p) Method [2]. Distances to GT.

**Figure 4.**
*Figure from [2]. Quantitative comparison using the sets $S_{10}$ and $S_{12}$. This figure is better viewed on a screen with a 4x zoom. (a), (b) and (c) [2] against $S_{10}$. The obtained 475 lines have been discretized in points. The distance from each point in the cloud to the surface of the Ground Truth mesh is represented in colors. (d) and (e) Same superposed onto the Ground Truth mesh. (f) Histogram of distances to Ground Truth with the [2] method. The maximum distance to be accounted is set to be 0.8, already considered as outlier. (g) Sparse atomic lines returned by the method [1]. It has been aligned with the Ground Truth mesh. (h) to (l) Same for the method [2] against the set $S_{12}$, with 556 segments. (m) and (n) same measurements for the result by [1]. (o) shows the histogram for this latter result.*

## 4. Conclusions

A 3D abstraction method receives as input the camera intrinsic parameters and several pictures of the scene. There are two different approaches: The first one does not require the camera extrinsics estimated from an external SfM pipeline, nor the Ground Truth camera poses [2]. It sources the line correspondences from a line matching method, and is able to generate 3D sketches from sets of pictures. This kind of approaches get an edge against datasets with low number of images, or when these present corrupted texture, blurring, and low definition images where the feature point descriptor fails to detect a fair number of keypoints. The reduced number of correspondences limit the thickness of the point cloud generated by the SfM pipelines, and therefore the accuracy of the estimated camera extrinsics. With inaccurate estimations for the cameras, exploiting homography constraints is not adequate to source line correspondences. Oppositely, [2] is able to reconstruct simple line-based sketches with fair precision and number of lines. It required lower number of images to obtain more complete abstractions than method [1]. The range of scenarios where it is advantageous to use method [1] for 3D abstraction includes sets of pictures of simple objects, with low texture, poor illumination, low resolution, blurring or under other conditions that make difficult the success of a point based algorithm. In these scenarios it outperforms the competition in terms of quantity of lines, precision and completeness of the abstraction. Another conclusion is that camera extrinsics are unavoidably required for 3D abstractions featuring many lines, because the estimation for the camera poses will not be accurate if the line matching method returns matching outliers or line fragmentation.

On the other hand, for datasets with moderate number of images, which clear textures, the second approach can be profitable. In this case, the geometric relations from the related points among the images will permit the feature point based pipeline to generate a moderately dense 3D point cloud. In this case, the poses of the cameras obtained by the point based pipeline can be trusted, and used as basis for line matching and linear projection to generate the 3D sketch. The results obtained with method [1] with datasets of hundreds of images are very good. An abstraction using this method will team perfectly with a dense reconstruction.

Both approaches are valid for their range of applications. The first one is valid for difficult datasets with noise and low number of images. The second approach will shine with datasets with high texture and many pictures, because it will will profit of the high precision obtained from feature point based 3D reconstruction pipelines for locating the cameras.

## Acknowledgements

The original research work [2] received financial support from the Xunta de Galicia through grant ED431C 2017/69 and Xunta the Galicia (Centro singular de investigación de Galicia accreditation 2016-2019) and the European Union (European Re gional Development Fund - ERDF) through grant ED431G/08.

## Conflict of interest

The author declares no conflict of interest.

## Author details

Roi Santos Mateos*, Xose M. Pardo and Xose R. Fdez-Vidal
University of Santiago de Compostela, Spain

*Address all correspondence to: roi.santos@usc.es

IntechOpen

## References

[1] Manuel Hofer, Michael Maurer, and Horst Bischof. Improving sparse 3d models for man-made environments using line-based 3d reconstruction. In 3D Vision (3DV), 2014 2nd International Conference on, volume 1, pages 535–542. IEEE, 2014.

[2] Santos, Roi, Xose M. Pardo, and Xose R. Fdez-Vidal. "Scene wireframes sketching for Unmanned Aerial Vehicles." Pattern Recognition 86 (2019): 354-367.

[3] Zhou, Hang, et al. "Uncertainty Analysis of 3D Line Reconstruction in a New Minimal Spatial Line Representation." Applied Sciences 10.3 (2020): 1096.

[4] Nieto, Marcos, and Luis Salgado. "Real-time robust estimation of vanishing points through nonlinear optimization." Real-Time Image and Video Processing 2010. Vol. 7724. International Society for Optics and Photonics, 2010.

[5] Lowe, David G. "Object recognition from local scale-invariant features." Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. Ieee, 1999.

[6] Alcantarilla, Pablo Fernández, Adrien Bartoli, and Andrew J. Davison. "KAZE features." European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012.

[7] Micusik, Branislav, and Jana Kosecka. "Piecewise planar city 3D modeling from street view panoramic sequences." 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009.

[8] Criminisi, Antonio, Ian Reid, and Andrew Zisserman. "A plane measuring device." Image and Vision Computing 17.8 (1999): 625-634.

[9] Kim, Hyunwoo, and Sukhan Lee. "Simultaneous line matching and epipolar geometry estimation based on the intersection context of coplanar line pairs." Pattern Recognition Letters 33.10 (2012): 1349-1363.

[10] Langlois, Pierre-Alain, Alexandre Boulch, and Renaud Marlet. "Surface reconstruction from 3d line segments." 2019 International Conference on 3D Vision (3DV). IEEE, 2019.

[11] Santos, Roi, Xose M. Pardo, and Xose R. Fdez-Vidal. "Outlier Detection for Line Matching." Iberoamerican Congress on Pattern Recognition. Springer, Cham, 2018.

[12] Nistér, David. "An efficient solution to the five-point relative pose problem." IEEE transactions on pattern analysis and machine intelligence 26.6 (2004): 756-770.

[13] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24.6 (1981): 381-395.

[14] Triggs, Bill, et al. "Bundle adjustment—a modern synthesis." International workshop on vision algorithms. Springer, Berlin, Heidelberg, 1999.

[15] Zhang, Lilian, and Reinhard Koch. "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency." Journal of Visual Communication and Image Representation 24.7 (2013): 794-805.

[16] Wang, Zhiheng, Fuchao Wu, and Zhanyi Hu. "MSLD: A robust descriptor for line matching." Pattern Recognition 42.5 (2009): 941-953.

[17] Bay, Herbert, Vittorio Ferraris, and Luc Van Gool. "Wide-baseline stereo matching with line segments." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

[18] Lourakis, Manolis IA, Spyros T. Halkidis, and Stelios C. Orphanoudakis. "Matching disparate views of planar surfaces using projective invariants." Image and Vision Computing 18.9 (2000): 673-683.

[19] Jia, Qi, et al. "Novel coplanar line-points invariants for robust line matching across views." European Conference on Computer Vision. Springer, Cham, 2016.

[20] Hartley, Richard, and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.

[21] Bartoli, Adrien, and Peter Sturm. "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment." Computer vision and image understanding 100.3 (2005): 416-441.

[22] Jain, Arjun, et al. "Exploiting global connectivity constraints for reconstruction of 3D line segments from images." 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010.

[23] Strecha, Christoph, et al. "On benchmarking camera calibration and multi-view stereo for high resolution imagery." 2008 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, 2008.