Robust and Accurate Line- and/or Point-Based Pose Estimation without Manhattan Assumptions

Yohann Salaün[1,2], Renaud Marlet[1], and Pascal Monasse[1(✉)]

[1] LIGM, UMR 8049, École des Ponts, UPE, Champs-sur-marne, France
{yohann.salaun,renaud.marlet,pascal.monasse}@enpc.fr
[2] CentraleSupélec, Châtenay-Malabry, France

Abstract. Usual Structure from Motion techniques based on feature points have a hard time on scenes with little texture or presenting a single plane, as in indoor environments. Line segments are more robust features in this case. We propose a novel geometrical criterion for two-view pose estimation using lines, that does not assume a Manhattan world. We also define a parameterless (*a contrario*) RANSAC-like method to discard calibration outliers and provide more robust pose estimations, possibly using points as well when available. Finally, we provide quantitative experimental data that illustrate failure cases of other methods and that show how our approach outperforms them, both in robustness and precision.

1 Introduction

Structure from Motion (SfM) techniques are now able to reliably recover the relative pose of cameras (external calibration) in many common settings, enabling 3D reconstruction from images as well as robotic navigation (SLAM). However, they still have a hard time in a number of practical situations, in particular in indoor environments, where surfaces are mainly planar with little or no texture. The fact is SfM techniques are mostly based on the detection of salient points, and such points are scarce in indoor settings and may occur in degenerate configurations, on a single plane. As a result, camera calibration can fail or yield inaccurate pose estimation.

Furthermore, a number of 3D reconstruction applications call for a reduced number of images to lower the acquisition burden. For instance, when a whole building is to be captured to generate a building information model (BIM), being able to only take a few pictures per room is more cost effective. It may even be compulsory for renovation companies, that have only a short and limited access to a building before submitting a tender. In this commercial stage, they do not look for the most accurate 3D information but for one that is easy to

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46478-7_49) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 801–818, 2016.
DOI: 10.1007/978-3-319-46478-7_49

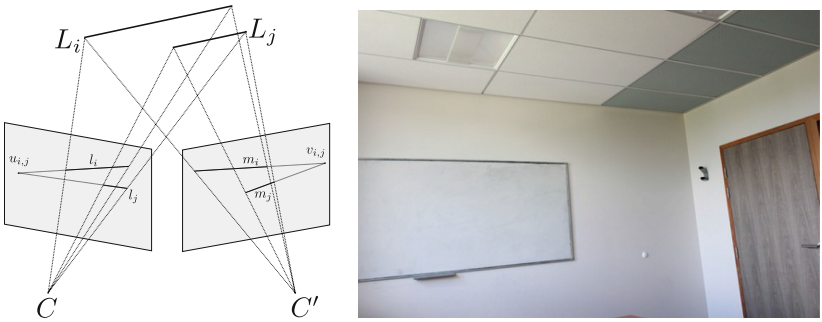

Fig. 1. To register two images, we use the relation between reprojected parallel 3D lines. It allows a more robust and accurate calibration in indoor scenes when points fail to calibrate.

capture and reliable enough to construct a sound bid. Some other companies also propose 3D tools and services to rethink the layout of rooms, possibly placing furniture advertisement too. For private individuals not to be dissuaded to run into this process, it must be easy for them to get a well approximated 3D view of their accommodation using only a few pictures. But lowering the number of images means that the baseline and view angle between two consecutive images are wider, and the image overlap is reduced. As a result, there are fewer salient points that are visible from at least two images, some matches are missed due to distorted feature descriptors, mismatch rate is higher due to matching threshold relaxation, and match location is less accurate due to perspective distortion at detection.

To circumvent these shortcomings, line segments have been proposed as robust features for camera calibration. In fact, line segments can be detected even in textureless images. Besides, while at least 5 points are required for motion estimation (essential matrix computation in the non-planar case), only 3 lines are enough under some conditions. Last, many lines segments correspond to actual 3D edges or to lines drawn on a planar surface, and are thus robust to strong viewpoint changes. Note however that only the line *direction* is actually robust, neither the segment *end points* in 2D nor the line position in 3D. Indeed, end points are not accurately detected in images and are often significantly wrong, as over-segmentation is common due to weak gradients and noise. Moreover, although the occluding edges of rounded objects such as round pillars and trees (visible edges of a cylinder) have a different location under different views, the 3D direction of the corresponding line segments stays the same.

A straightforward approach to camera rotation estimation with lines is to estimate vanishing points (VPs) in images based on detected line segments, to match these VPs, and to define the rotation between the two images as the rotation that best sends each 3D VP direction in one image to its corresponding 3D VP direction in the other image. However, such a calibration has a poor accuracy. The reason is that existing methods for VP detection are not assessed

on motion estimation; they are generally tuned regarding line clustering capacities as well as zenith and horizon estimation [1,2], with arbitrary ground truths. Moreover, VPs are theoretical constructs; they are not real. They abstract the fact that actual object lines that are more or less parallel, more or less converge to the same area. But lines on objects are never perfectly parallel and objects, including buildings, are never perfectly parallel one to another. (The same goes for orthogonality.) The fact is multiple VPs are often detected for a single "logical" vanishing direction. This is in contrast with actual 3D points on objects, which exist per se, independently of other points, although locating identical points on different images may be inaccurate.

Elqursh and Elgammal [3] proposed a 3-line approach for camera pose estimation. They consider a triplet of 3D lines L_0, L_1, L_2 such that $L_0 \parallel L_1$, having a 3D direction d_1, and $L_2 \perp L_0, L_1$, having a 3D direction d_2. Given a reprojection of these 3D lines on a image as (l_0, l_1, l_2), d_1 can be recovered as the vanishing point corresponding to (l_0, l_1), given by the intersection of l_0 and l_1, and d_2 as the direction orthogonal to d_1 that belongs to l_2 when seen as a 2D point. Considering similar reprojections on a second image, a camera motion can be computed as the rotation that best sends (d_1, d_2) estimated from image 1 to (d'_1, d'_2) estimated from image 2. We show in this paper that it leads to more accurate rotations than using the average VPs. Our interpretation is that VPs prematurely aggregate vanishing lines. On the contrary, the 3-line approach considers the contribution of each vanishing line independently (actually by pairs of orthogonal directions), which is less sensitive to coarse parallelism. Besides, filtering triplets with a RANSAC-like procedure to only keep inliers within an angular threshold of the rotation considered as model leads to an even more accurate rotation.

Yet, this method has a number of drawbacks. First, it assumes that some vanishing directions are orthogonal and that there are enough triplet samples of line segments of this kind for a significant group of inliers to emerge. Second, to estimate the translation, which requires points contrary to rotation estimation, this method only considers line intersection information (assuming lines are coplanar), which are poor cues; it does not exploit detected points when some are available, although they could improve the calibration. Third, the final refinement stage, a Levenberg-Marquardt optimization with free rotation and translation, only involves error measures of points, which can lead to degenerate cases and degrade the rotation estimation when points are mostly planar.

In this paper, we propose a novel approach for two-view pose estimation using lines, without Manhattan-world assumptions (Fig. 1). The key idea is to consider pairs of supposedly parallel lines. Each pair identifies a vanishing direction, and two such pairs are thus enough to define a rotation, without any orthogonality constraint. This formulation generalizes well to robust estimation. Our contributions are the following:

– We present a line-based orthogonality-free geometric criterion for pose estimation.

– We turn it into a robust method, possibly combining with point detections if any.
– We construct a parameterless (*a contrario*) version of this robust method.
– We provide quantitative experimental data that illustrate failure cases of calibrating with points only or with the 3-line method, and show that our approach consistently outperforms other methods, providing the best of both the line and point worlds.

2 Related Work

Lines alone do not provide enough information about the relative pose between two images [4,5]. That is why the usual scheme for line-based calibration uses the trifocal tensor [6], relying on triplets of pictures to estimate relative poses. In this setting, 3D lines can then be reconstructed and refined with the motion [7], possibly under some Manhattan-world assumption [8]. In [9], the authors developed a whole framework to calibrate a scene from lines only. However, this approach needs 13 triplets of matched lines between pictures. This requires in practice a large overlap between pictures, and the presence of many inliers. In contrast, our method can calibrate a pair of images for either small baselines (e.g., for SLAM) or wide baselines (e.g., for 3D reconstruction).

Using a device to shoot two stereo pictures at a time, calibrating image pairs using lines becomes simpler than with the trifocal tensor [10]. In this setting, points too have been used in addition to lines, but with a small baseline (SLAM) and again with trifocal constraints [11]. This category of methods does not apply to an arbitrary set of pictures.

To get rid of the requirements imposed by the trifocal tensor, other approaches assume additional constraints on lines. The main such constraint is a Manhattan-world assumption [12]. However, it reduces the applicability to specific (although common) scenes as it requires that at least 3 dominant directions are found and that all these directions are orthogonal. Elqursh and Elgammal [3] only use local parallelism and orthogonality hypotheses to estimate motion from lines, but it remains a theoretical and practical limitation. Besides their method disregards points that could be detected, which misses an opportunity for greater accuracy. Their refinement stage may also degrade the solution as it gives little importance to pure line constraints.

Assuming that matched line segments in two images overlap in 3D (as opposed to just defining a common direction) and supposing that this overlap is maximal, it is possible to recover both the motion and the 3D line structure [5]. However, as mentioned in the introduction, over-segmentation is frequent and the overlap constraint is thus too strong for practical cases. A related approach, based on segment midpoints constrained to move only along the line direction has been proposed [13], not requiring overlap but still sensitive to over-segmentation. Besides, both approaches require non planar scenes.

Another family of approaches use junctions at line intersections, reducing to a point problem [14,15]. However, as shown below, these points are not accurately

located. And again, it does not address the degenerate case of points lying on a single plane.

Non-Manhattan scenes have been addressed too, but in a setting with a very small baseline (e.g., for SLAM) where the motion can be predicted from one frame to another [16]. Besides, the estimated motion is based on estimated VPs, which has a lower accuracy than directly using lines, as argued above and shown below. Still, a common approach to estimate the camera motion is to map vanishing points in one image to vanishing points in the other image, possibly assuming there exist three mutually orthogonal vanishing points [8,17], possibly in conjunction with points [18,19].

3 Pose Estimation from Lines

We consider a set of 3D lines viewed by two cameras and, when available, a set of 3D points also viewed by the cameras. We write $\hat{l}$ and $\hat{p}$ the projections on the first camera of a 3D line L and a 3D point P. We note $\hat{m}$ and $\hat{q}$ their projection on the second camera. We use homogeneous coordinates to represent the lines and points.

Without loss of generality, we suppose the first and second camera projection matrices are respectively $K[I|0]$ and $K'[R|t]$, where R and t are the rotation and the translation direction we want to estimate. We assume the internal parameters K, K' of the cameras are known and we note C, C' the camera centers. Given a 3D line L, we consider the normal $l = K^{-T}\hat{l}$ to the plane passing through C and L; and given a 3D point P, we consider the 3D point direction $p = K^{-1}\hat{p}$.

3.1 Vanishing Point of Two Parallel Lines

Let L_i, L_j be two parallel 3D lines. Their 2D projection l_i, l_j intersect at a vanishing point $l_i \times l_j$, that can be seen both as a 2D point in the image and as the common 3D direction of lines L_i, L_j. The normalized direction of this vanishing point is $u_{ij} = \frac{l_i \times l_j}{||l_i \times l_j||}$. Similarly, the projections m_i and m_j on the second image intersect at vanishing point $v_{ij} = \frac{m_i \times m_j}{||m_i \times m_j||}$, which also represents the common direction of lines L_i, L_j. The orientation of cameras being related by rotation R, we thus have:

$$Ru_{ij} = s_{ij}\, v_{ij}, \tag{1}$$

where $s_{ij} = \pm 1$, as the direction is not oriented.

3.2 Rotation Estimation from Two Pairs of Parallel Lines

We consider two such pairs of parallel lines, with corresponding VP directions u_1, u_2 for the first camera and v_1, v_2 for the second camera. The rotation R satisfies

$$Ru_1 = s_1 v_1 + \epsilon_1 \qquad Ru_2 = s_2 v_2 + \epsilon_2 \qquad \text{with } s_1, s_2 = \pm 1, \tag{2}$$

and $\epsilon_1 = \epsilon_2 = 0$. Due to noise, no rotation may achieve these conditions. Still, the rotation $\hat{R}$ that satisfies at best (2), in the sense that it minimizes $\|\epsilon_1\|^2 + \|\epsilon_2\|^2$, can be computed as $\hat{R} = AB^T$ where $A\Sigma B^T$ is the singular value decomposition (SVD) of 3×3 matrix $M = s_1 v_1 u_1^T + s_2 v_2 u_2^T$ [20]. Getting only an approximate rotation matters little here because it is just to be used in a RANSAC framework to select inliers, from which a refined rotation is then computed.

As signs s_1, s_2 are unknown, 4 rotation matrices are possible solutions. The rotation matrix to retain can be chosen either with a geometric criterion (e.g., the rotation that has the largest number of features in front of both cameras) or an angular criterion (e.g., the rotation whose angle is less than 90°). As degenerate cases can occur if the parallel lines used to compute $\hat{R}$ belong to the same or to close vanishing points, a practical heuristics is to check that vanishing directions u_1 and u_2 differ by at least a given angle (5° in our experiments), and likewise for v_1 and v_2.

3.3 Translation Estimation

The translation t can only be computed up to a scale factor, and its direction cannot be estimated just from lines without extra constraints. Still, once the rotation R is computed, the translation direction can be estimated from two point correspondences.

Two non-parallel coplanar 3D lines intersect at a 3D point, and their 2D projections intersect at corresponding 2D points. Given two such 2D points p_1, p_2, with correspondence q_1, q_2 in the second image, we should have $(Rp_i \times q_i)^T t = 0$. A translation direction can thus be defined as $\arg\min_{\|t\|_2=1} \sum_{i=1}^{2} ((Rp_i \times q_i)^T t)^2$ and computed as the vector associated to the lowest singular value of the 3×3 matrix $\sum_{i=1}^{2} (Rp_i \times q_i)(Rp_i \times q_i)^T$.

Instead of relying only on points corresponding to line intersections hypotheses as in [3], we also consider using detected feature points when available. The fact is that even in low-textured scenes, a few good points can often been detected and matched. Besides, point correspondences originating from detections are often more accurate than line intersections, and there are less mismatches than when considering the intersection of any two lines, as they are not necessarily coplanar. Experiments show that detected points, if any, contribute to a better accuracy (see Sect. 7.3)

4 Robust Pose Estimation

4.1 Robust Rotation Estimation

For a robust rotation estimation, we use a RANSAC method where we sample line pairs. At each iteration, we randomly pick 2 different VPs, and then 2 lines for each VP, which are thus likely to be 3D-parallel. These two line pairs define a rotation R, as described in Sect. 3. Any other line pair (l_i, l_j), with matches (m_i, m_j), is then considered as an inlier for model R iff the following angle

$$d_{\mathsf{lines}}(l_i, l_j, m_i, m_j) = \angle(R(l_i \times l_j), m_i \times m_j) \tag{3}$$

is less than a given threshold (2° in our experiments). This angle measures the discrepancy between the two vanishing directions defined by (l_i, l_j) and (m_i, m_j).

Given the rotation hypothesis $\tilde{R}$ maximizing the number n of inliers, a better rotation $\bar{R}$ can then be re-estimated from all inliers. For this, we rely on the same tool used in [3]: considering that each inlier (l_i, l_j, m_i, m_j) defines two vanishing directions u_{ij}, v_{ij} which should be equal up to R, cf. (1), the best rotation can be defined as

$$\bar{R} = \underset{R^T R = I}{\arg\min} \|RU - V\|_F \tag{4}$$

where $\|.\|_F$ is the Frobenius norm, U and V are the $3 \times n$ concatenations of the column vectors u_{ij} and $s_{ij} v_{ij}$, and s_{ij} is the sign that best sends u_{ij} to $s_{ij} v_{ij}$, i.e., with the lowest $\angle(\tilde{R} u_{ij}, s_{ij}\, v_{ij})$. The solution to this orthogonal Procrustes problem can be obtained as the projection of the 3×3 matrix $M = VU^T$ onto the set of orthogonal matrices, easily derived from the SVD $M = A\Sigma B^T$, as $\bar{R} = AB^T$ [21]. (This estimation actually generalizes the estimation used in the two line pairs case, cf. Sect. 3.2.)

4.2 Robust Translation Estimation

For a robust translation estimation, given an estimated rotation, we use a RANSAC method where we sample points. A point can be picked by sampling two lines belonging to two different VP clusters (as their intersection is meaningless for translation if they belong to the same VP), or by sampling a detected point, if any (see Sect. 3).

For a more homogeneous treatment of points w.r.t. lines, we do not use the standard point reprojection error, measured in pixels. We rely on a new distance function that provides an angular error measure, as defined below. The threshold to decide whether a point is an inlier w.r.t. a translation hypothesis can thus be the same as for line pairs with a rotation hypothesis (2° in our experiments).

For any point p_i in the first image, with correspondence q_i in the second image, $Rp_i + t$ should be equal to q_i up to a scale factor. As the magnitude of t cannot be known, this relation is better exploited by considering the cross product with t, i.e., $Rp_i \times t$ should be collinear to $q_i \times t$. This can be seen as the equality of the normals or the epipolar planes $CC'p_i$ and $CC'q_i$. This leads to the following angle error measure:

$$d_{\mathsf{p,oints}}(p_i, q_i) = \angle(Rp_i \times t, q_i \times t) \tag{5}$$

The robust translation direction $\tilde{t}$ maximizes the number of point inliers w.r.t. $d_{\mathsf{p,oints}}$.

4.3 Non-Linear Refinement

As for the rotation, the best translation hypothesis $\tilde{t}$ can be re-estimated using all inliers. We actually refine simultaneously both R and t w.r.t. found inliers. For

this, we define an energy combining homogeneously line and point constraints, based on angular errors:

$$\mathcal{C}_{\mathsf{lines}}(R) = \sum_{ij \text{ inlier line pair}} \|Ru_{ij} \times s_{ij}v_{ij}\|^2 \tag{6}$$

$$\mathcal{C}_{\mathsf{p,oints}}(R,t) = \sum_{i \text{ inlier point}} \left\| \frac{(Rp_i \times t)^T}{||Rp_i \times t||} \times \frac{q_i \times t}{||q_i \times t||} \right\|^2 \tag{7}$$

We then use the Levenberg-Marquardt (LM) algorithm to minimize the sum of these two functions, starting from the estimated motion $\bar{R}$ and $\tilde{t}$, to obtain a refined calibration:

$$(R^\star, t^\star) = \arg\min_{R,t} \mathcal{C}_{\mathsf{lines}}(R) + \mathcal{C}_{\mathsf{p,oints}}(R,t) \tag{8}$$

In [3], the authors also use a refinement process, but it mainly takes points into account and only soft constraints for lines. Experiments have shown that it tends to deteriorate the solution in scenes where points cannot calibrate (see Sect. 7.4).

5 Robust Pose Estimation from Lines and Points

Feature points, when detected, are not only useful for translation estimation (cf. Sects. 3.3 and 4.2). They can also be used for the whole motion estimation, as in traditional SfM. In fact, since points generally lead to very accurate calibrations and as lines are more robust to degenerate cases, an appropriate use of both kinds of features should lead to more robustness and higher accuracy.

To benefit from the best of both worlds, we consider a mixed method where models are alternatively sampled as follows and we keep the pose that maximizes the total number of inliers:

- We draw 2 line pairs to estimate a rotation (cf. Sect. 4.1), then 2 points to estimate a translation (cf. Sect. 4.2). These points may indifferently correspond to line intersections or to detected points as they are drawn from a single merged set.
- Or we draw 5 points to get an essential matrix, thus a rotation and translation [22].

6 Robust Parameterless *a Contrario* Pose Estimation

To avoid the burden of having to explicitly choose a distance threshold in RANSAC, which is data-specific, we use the *a contrario* theory (AC). In this setting, line and possibly point inliers are automatically selected without having to set specific thresholds. Moreover, Moulon *et al.* [23] have shown that such a parameterless AC-RANSAC performs better than standard RANSAC, not only because it relies on optimal thresholds but also because it can adapt to data variation within a single dataset.

6.1 Number of False Alarms (NFA)

In the framework of the *a contrario* theory, the accuracy of a set of inliers w.r.t. a model is measured using the Number of False Alarms (NFA). This NFA is an upper-bound approximation of the expected number of models of equivalent accuracy obtained with all possible combinations drawn from n random features following a given background model. In this setting, models with high expectations are considered less meaningful than models with low expectations because they are more likely to be found with random features that have no real meaning.

The AC theory has been applied to robust Fundamental matrix estimation from point features [24,25], and generalized to any geometric model [26]. The general formula is:

$$\mathrm{NFA}(n,k,\epsilon) = N_{\mathsf{outcomes}}(n - N_{\mathsf{sample}})\binom{n}{k}\binom{k}{N_{\mathsf{sample}}}\mathrm{p}(\epsilon)^{k-N_{\mathsf{sample}}} \tag{9}$$

where N_{sample} is the number of samples needed to estimate one model, N_{outcomes} is the maximum number of models estimated from a given sampling, k is the number of hypothesized inlier correspondences, and $\mathrm{p}(\epsilon)$ is the probability for a random feature following the background model to be at a distance lower than ϵ of the estimated model.

6.2 NFA for Rotation Estimation from Lines

As in [24], we suppose that our background model consists of uniform and independent random lines distributed in the image. The estimated model is a rotation matrix, and because of sign ambiguity, $N_{\mathsf{outcomes}} = 4$ different rotations can be obtained from a sample of $N_{\mathsf{sample}} = 4$ lines, treated as two line pairs (cf. Sect. 3.2). Thus:

$$\mathrm{NFA}_{\mathsf{lines}}(n_{\mathsf{lines}},k_{\mathsf{lines}},\epsilon) = 4(n_{\mathsf{lines}}-4)\binom{n_{\mathsf{lines}}}{k_{\mathsf{lines}}}\binom{k_{\mathsf{lines}}}{4}\mathrm{p}_{\mathsf{line}}(\epsilon)^{k_{\mathsf{lines}}-4}. \tag{10}$$

Given a single random line, there is not enough information to compute its actual distance to the model. Yet, we can approximate such a distance from the distance, to the model, of 3D-parallel line pairs (Eq. (3)) that contain this line. We actually define the error of a single line l_i and its match m_i w.r.t. a rotation model R as:

$$d_{\mathsf{line}}(l_i,m_i,R) = \min_{L_j \parallel L_i} d_{\mathsf{lines}}(l_i,l_j,m_i,m_j). \tag{11}$$

Here, $\mathrm{p}_{\mathsf{line}}(\epsilon)$ is the probability for a random 3D line to have the correct direction up to an angular error of ϵ. This probability can be expressed as the relative area of two spherical caps of the unit sphere:

$$\mathrm{p}_{\mathsf{line}}(\epsilon) = \mathbb{P}(d_{\mathsf{line}}(l_i,m_i,R) \le \epsilon) = \frac{2\ \mathcal{A}_{\mathsf{cap}}(\epsilon)}{\mathcal{A}_{\mathsf{sphere}}} = \frac{4\pi(1-\cos\epsilon)}{4\pi} = 1-\cos\epsilon. \tag{12}$$

At each RANSAC iteration, we evaluate the distance of every line to the estimated rotation and sort them by increasing distance. The NFA of this rotation is given by:

$$\mathrm{NFA}(R) = \min_{k \in [4, n_{\mathsf{lines}}]} \mathrm{NFA}_{\mathsf{lines}}(n_{\mathsf{lines}}, k, d_{\mathsf{line}}(l_k, m_k, R)), \tag{13}$$

where (l_k, m_k) is the k-th line after distance sorting. The final rotation is the rotation with the lowest NFA after all RANSAC iterations.

6.3 NFA for Motion Estimation from Lines and Points

We now combine lines and point features into a unified AC framework for pose estimation. More precisely, we provide a parameterless AC variant of the method in Sect. 5, where we alternatively draw 2×2 lines plus 2 points, or just 5 points.

As mixing heterogeneous samplings is complex, we make a number of approximations. We merge all line and point features into a single set, and consider the event "randomly pick a motion that has k ϵ-accurate inliers (lines and/or points) among a total of $n = n_{\mathsf{lines}} + n_{\mathsf{p,oints}}$ features". We are interested in samples of $N_{\mathsf{sample}} = 6$ features, considering additionally that the only samples from which we can build a valid model consist either of 2×2 lines plus 2 points, or at least 5 points. All other kinds of samples (with another proportion of lines and points among 6 features) are disregarded and treated as if no model could be constructed from them. In this setting, the maximum number of possible motions that are compatible with the sample is $N_{\mathsf{outcomes}} = \max(4, 10)$ and we define the following approximate NFA:

$$\mathrm{NFA}(n, k, \epsilon) = 10(n-6)\binom{n}{k}\binom{k}{6}\mathrm{p}(\epsilon)^{k-6}. \tag{14}$$

Here, we consider that we still have $\mathrm{p}(\epsilon) = 1 - \cos\epsilon$ since we compare the 3D directions of either 3D lines or epipolar plane normals.

At each RANSAC iteration, we evaluate the angular distances of every line and point to the estimated pose (R, t) and sort them by increasing value. The NFA is here:

$$\mathrm{NFA}(R, t) = \min_{k \in [6, n]} \mathrm{NFA}(n, k, d(f_k, g_k, R, t)), \tag{15}$$

where f_k is the k-th feature after distance sorting, and g_k its match. The final pose is the one that has the lowest NFA after all RANSAC iterations.

7 Experiments

To compare and assess the different methods, we experiment on real and synthetic data. (More details on the experiments are provided in the supplementary material.)

Fig. 2. Sample of pictures from datasets (left to right) *Building*, *Car* and *Office*.

Datasets. Synthetic datasets are described below. Our real datasets (Fig. 2) consist of:

- *Office*: an office room with more or less Manhattan directions and little texture,
- Strecha *et al.*'s dataset [27]: several outdoor scenes (e.g., castle courtyard), which is a de facto standard for evaluating camera calibration methods,
- *Building*: a V-shaped building featuring some non-orthogonal lines,
- *Car*: the close view of a car in the street, with no particular line alignments.

A ground-truth motion is available for all datasets.

For all experiments, line segments are detected with a variant of LSD [28] that detects lines at different scales and reduces over-segmentation [29]. They are then matched with LBD [30]. We also detect and match possible feature points with SIFT [31]. Matches are then filtered with K-VLD [32]. In this setting, the resulting line segments and points contain few mismatches. We cluster lines using the vanishing point detector of Almansa *et al.* [33]. Line clusters are then further merged with a greedy strategy if their vanishing directions are similar up to a given threshold (5° in our experiments).

Compared methods. We consider the following methods for comparison:

- **Best VP**: a rotation estimation only, based on detected vanishing points. VPs in different images are greedily matched based on line matches (largest number of matches in common). As it is common that some VPs are inaccurate or under-represented w.r.t. support lines, there are often only two reliable VPs. For this reason, we consider rotations originating only from a pair of VPs. Besides, to provide a strong baseline to compare with other methods, we actually consider all rotations estimated from all VP pairs and keep the one that is the closest to the ground truth.
- **5-point**: pose estimation via essential matrix computation from 5 points [22], using AC-RANSAC as it performs better than standard (fixed-threshold) RANSAC [23].

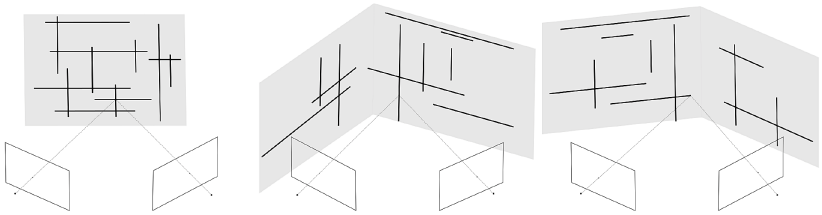

Fig. 3. Configurations: *Planar* (left), *Manhattan* (middle), *Quasi-Manhattan* (right).

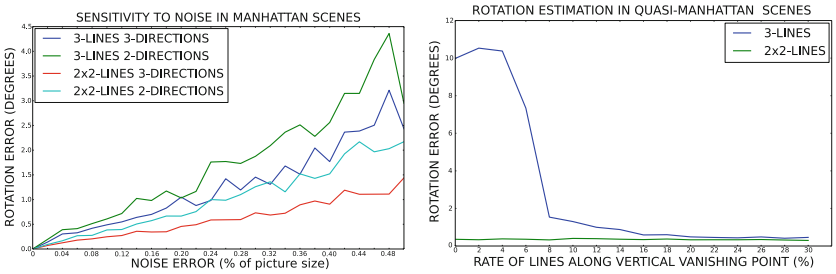

Fig. 4. Left: impact of noise in Manhattan scenes. Right: impact of the rate of orthogonal lines.

- **4-point**: pose estimation via homography matrix computation from 4 points in an *a contrario* framework [26], a method supposed to better deal with planar scenes.
- **3-line**: the method developed in [3], using arrangements of line triplets to estimate the pose. The RANSAC error threshold for rotation estimation is 2° as in the paper.
- **3-line + SIFT**: a variant of 3-line where SIFT detections are added to line intersections for translation estimation.
- 2×2**-line**: our method based on pairs of parallel lines to estimate the rotation, with possible SIFT points for translation (cf. Sect. 4). RANSAC threshold is 2° too.
- **mixed**: the combination of our 2×2-line method and the 5-point method in a classical RANSAC framework (cf. Sect. 5).
- **AC-mixed**: our *a contrario* variant of the mixed method (cf. Sect. 6).

We use the angular refinement presented in Sect. 4.3 implemented with the Ceres library [34]. Note that for points methods, our experiments have shown that the angular refinement is as accurate as the epipolar distance refinement.

7.1 Sensitivity to Noise

To study the impact of noise on pose estimation, we resort to synthetic data. We test only line-based methods as it is not clear how to relate noise models for

lines and for points. We consider the following realistic common configurations (see Fig. 3):

– *Planar*: lines along 2 orthogonal directions on a single plane,
– *Manhattan*: lines along 3 orthogonal directions on multiple planes.

For each scene, we generate 100 random 3D line segments on the planes with a uniform distribution on the different directions, and we generate 2 camera positions on a circle around the scene with an angle of 45° between them. Each line is projected on both views and we add a Gaussian noise with standard deviation σ to shift all line end points. Line matches and VP clusters are given as input to each method to avoid any bias. For each configuration, we randomly generate 100 such scenes to get average results.

We study in Fig. 4 the rotation accuracy as a function of noise σ, varying between 0 % and 0.5 % of the image size. The 2×2-line method is more accurate than the 3-line method, even in the Manhattan configuration for which the 3-line method is designed.

7.2 Sensitivity to Manhattan-Ness

Figure 3 studies the impact of Manhattan-ness, in another synthetic configuration:

– *Quasi-Manhattan*: lines along 3 directions d_1, d_2, d_3 such that $d_1 \perp d_2, d_3$ and $\widehat{d_2, d_3} = 120°$, on multiple planes.

The noise σ is set here to a medium value (0.2 % of image size), and we vary the rate of lines in the vertical direction d_1, between 0 % and 30 % of the 100 sampled lines, for 100 random scenes. As observed in Fig. 4, the 3-line method is not robust to a low rate of orthogonal lines, whereas our 2×2-line method is unaffected.

As for real data, we consider the *Building* dataset, which is analogous to the above synthetic test configuration, due to the V shape of the building, as well as the *Car* dataset, which is inherently non-Manhattan and which features only weak vanishing points. Results are shown on the left of Fig. 5. The 3-line method fails on both datasets, with rotation errors mostly above 5°, sometimes higher than 20°. In constrast, our 2×2-line method is robust, even on the *Car* dataset, with errors mostly under 2°.

7.3 Line Intersections Vs Detected Feature Points

We now study the impact of points on translation accuracy. As argued in Sect. 3.3, we propose to use points, if detected, together with line intersections to get a good tradeoff: benefiting from accurate point detections when available, but always having line intersections as backup in case no point is detected and matched.

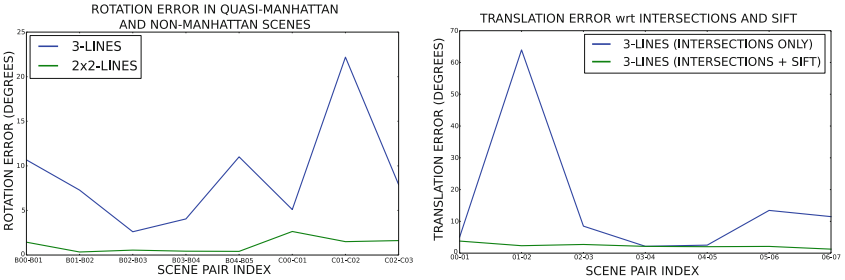

Fig. 5. Left: rotation error on image pairs of datasets *Building* (index $B_{i,i+1}$) and *Car* (index $C_{j,j+1}$), in a row. Right: translation error on image pairs of the *Office* dataset.

We consider the 3-line method, that originally only uses line intersection points [3]; we just add detected points to intersections for translation estimation. We experiment with *Office*, a low-textured dataset featuring only a few SIFT points (about 30 on average). Results are shown on the right of Fig. 5. The combination of detections with line intersections yields a far more robust and accurate estimation than with intersections alone — even though these detected points alone are not enough to provide a good calibration (cf. Table 1). Results are similar with the 2×2-line method.

7.4 Sensitivity to Motion Refinement

To study the sensitivity of the final motion estimation refinement (LM), we compare the 3-line method [3], whose refinement uses hard point constraints and (indirectly) soft line constraints, with the refinement in our 2×2-line method, which balances equally line and point constraints. For this, we compute estimation errors before and after refinement. We also compare with the 5-point method to illustrate the deterioration effect observed with the 3-line method refinement. We use *Office*, a dataset that points alone fail to calibrate well because of the lack of texture, and Strecha's *Herz-Jesu-P8* scene [27], where calibration based on points succeeds very well.

Figure 6 shows the results. Refining using line intersections only, as in the original 3-line method [3], often provides a poor calibration. As the refinement in [3] uses only points, the 3-line+SIFT method tends to behave as the 5-point method, which reduces the interest of using lines: it improves or deteriorates the initial estimation, depending on the ability of points to calibrate the image pair. In *Herz-Jesu-P8*, as the 5-point method is extremely accurate, refinement is better for 3-line+SIFT than for 2×2-line. However, for scenes not calibrated by points such as *Office*, the refinement for 3-line+SIFT degrades the original solution whereas our refinement always improves it.

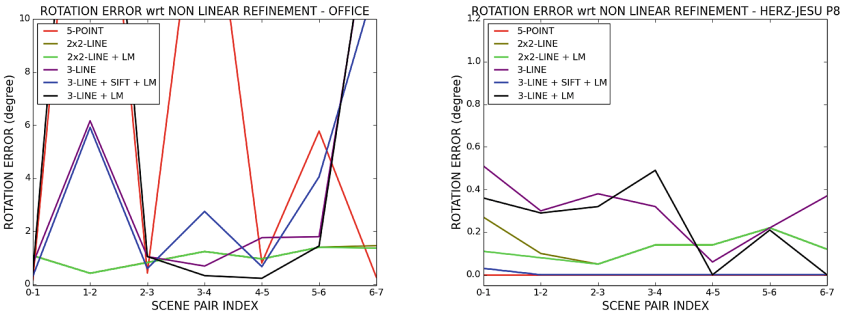

Fig. 6. Rotation error for image pairs in datasets *Office* (left) and Strecha's Herz-Jesu-P8 (right). Reported methods do not use their final refinement stage (Levenberg-Marquardt, LM) unless otherwise mentioned. The curve for "2×2-lines + LM" is often hidden by the "2×2-line" curve.

Table 1. On the left, the angular error of rotation estimation, without refinement. Best results are shown in **bold**. Unreliable results, with an average error over 5°, are shown in red. On the right, the percentage of line hypothesis kept at the end of RANSAC for hybrid methods.

Method / Dataset	VP-based	Point-based		Line-based	
	Best VP	5-point	4-point	3-line	2×2-line
Strecha	1.29	**0.05**	1.86	0.35	0.28
Office	**0.65**	8.63	25.37	3.93	1.05
Building	0.54	**0.35**	1.00	8.04	0.56
Car	14.93	**0.23**	3.19	24.27	2.41

Method / Dataset	mixed	AC-mixed
Strecha	54%	10%
Office	63%	43%
Building	70%	20%
Car	0%	0%

Table 2. Average error of rotation and translation estimation, including non-linear refinement. Best results are shown in **bold**. Unreliable ones, with an average error over 5°, are shown in red.

Method / Dataset		5-point	3-line	3-line+ SIFT	2×2-line	mixed	AC-mixed
Strecha	R	**0.02**	0.46	0.05	0.25	0.19	**0.02**
	t	**0.18**	3.37	0.36	1.03	0.80	0.21
Office	R	6.88	6.45	3.67	1.03	1.01	**0.57**
	t	27.19	20.38	16.26	3.26	3.13	**1.44**
Building	R	0.23	6.68	3.73	0.49	0.24	**0.21**
	t	**0.31**	37.63	18.72	1.57	0.83	0.45
Car	R	**0.19**	24.25	13.81	2.37	0.75	0.24
	t	**0.20**	69.47	30.46	18.03	0.89	0.28

7.5 Robustness and Accuracy of Rotation Estimation

As translation estimation is very sensitive to rotation errors, we study rotation estimation. Table 1 confirms that (*i*) VPs alone do not provide reliable results, alternating accurate and poor estimations. (*ii*) Point methods are very accurate but lack of robustness in low-textured scenes. (*iii*) Line methods are robust to the lack of texture. However, the 3-line method is mostly limited to Manhattan scenes, contrary to our 2×2-line method, which systematically outperforms the 3-line method, even on Manhattan scenes.

7.6 Robustness and Accuracy of Pose Estimation

In Table 2, all methods are refined, using also SIFT matches (except the 3-line method). Experiments show the benefit of combining line and point features: the mixed methods are robust to the variety of scenes, and their accuracy is better than or about the same as point-only methods. Moreover, although the RANSAC method already gives good results, its *a contrario* variant is far more accurate and does not need any parameter.

8 Conclusion

We presented a new framework for line-based camera pose estimation. Unlike [3], the approach does not require orthogonality. It is also less sensitive to noise. Besides being compatible with wide baseline, not requiring overlaps in three views, it requires a low number of features, i.e., 2×2 lines, which is good for RANSAC when the outlier rate is high (higher than the 3-line method [3], but much less than 13-line methods [4,9]).

We also define a proper way to combine line and point information into a robust and accurate calibration method that leverages on both kinds of features. Our refinement balances well lines and points, contrary to [3]. It is a significant improvement over methods that only turn lines into points with an extra coplanarity assumption [3,14,15].

We thouroughly study the behavior of our approach in different settings, comparing to other existing methods. Our experiments show that our AC-mixed method is at least as robust and accurate as other methods in any context, most often outperforming them, including point-based methods. As it is based on the *a contrario* theory, our method does not need any parameter tuning and automatically adapts to data variation, including within a single dataset, which is important for the robustness of an SfM pipeline.

Future work includes extending our approach to multiple views. Moreover, even if low texture does not impede the calibration, dense reconstruction do not work well in this case; we aim at leveraging on lines to enable reconstruction of such scenes.

Acknowledgements. This work was carried out in IMAGINE, a joint research project between ENPC and CSTB. It was partly supported by Bouygues Construction.

References

1. Xu, Y., Oh, S., Hoogs, A.: A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments. In: IEEE Computer Vision and Pattern Recognition (CVPR 2013), pp. 1376–1383, June 2013
2. Lezama, J., Grompone von Gioi, R., Randall, G., Morel, J.M.: Finding vanishing points via point alignments in image primal and dual domains. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), June 2014
3. Elqursh, A., Elgammal, A.: Line-based relative pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), pp. 3049–3056, June 2011
4. Weng, J., Huang, T.S., Ahuja, N.: Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI 1992) **14**(3), 318–336 (1992)
5. Zhang, Z.: Estimating motion and structure from correspondences of line segments between two perspective images. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI 1995) **17**(12), 1129–1139 (1995)
6. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, New York (2004). ISBN: 0521540518
7. Bartoli, A., Sturm, P.F.: Structure-from-motion using lines: representation, triangulation, and bundle adjustment. Comput. Vis. Image Underst. (CVIU 2005) **100**(3), 416–441 (2005)
8. Schindler, G., Krishnamurthy, P., Dellaert, F.: Line-based structure from motion for urban environments. In: Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT 2006), Computer Society, pp. 846–853. IEEE, Washington, DC (2006)
9. Zhang, L., Koch, R.: Structure and motion from line correspondences: representation, projection, initialization and sparse bundle adjustment. J. Vis. Commun. Image Representation **25**(5), 904–915 (2014)
10. Chandraker, M., Lim, J., Kriegman, D.: Moving in stereo: efficient structure and motion using lines. In: IEEE 12th International Conference on Computer Vision (ICCV 2009), pp. 1741–1748 (2009)
11. Pradeep, V., Lim, J.: Egomotion using assorted features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), pp. 1514–1521, June 2010
12. Koseck, J., Zhang, W.: Extraction, matching, and pose recovery based on dominant rectangular structures. Comput. Vis. Image Underst. (CVIU 2005) **100**(3), 274–293 (2005)
13. Montiel, J., Tardós, J., Montano, L.: Structure and motion from straight line segments. Pattern Recogn. (PR 2000) **33**(8), 1295–1307 (2000)
14. Bay, H., Ferrari, V., Van Gool, L.: Wide-baseline stereo matching with line segments. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005), Computer Society, pp. 329–336. IEEE, Washington, DC (2005)
15. Draréni, J., Keriven, R., Marlet, R.: Indoor calibration using segment chains. In: Mester, R., Felsberg, M. (eds.) DAGM 2011. LNCS, vol. 6835, pp. 71–80. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23123-0_8
16. Lee, J.K., Yoon, K.J.: Real-time joint estimation of camera orientation and vanishing points. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), June 2015

17. Cipolla, R., Drummond, T., Robertson, D.: Camera calibration from vanishing points in images of architectural scenes. In: British Machine Vision Conference (BMVC 1999), pp. 382–391 (1999)
18. Rother, C., Carlsson, S.: Linear multi view reconstruction and camera recovery using a reference plane. Int. J. Comput. Vis. (IJCV 2002) **49**(2–3), 117–141 (2002)
19. Sinha, S.N., Steedly, D., Szeliski, R.: A multi-stage linear approach to structure from motion. In: Kutulakos, K.N. (ed.) ECCV 2010. LNCS, vol. 6554, pp. 267–281. Springer, Heidelberg (2012). doi:10.1007/978-3-642-35740-4_21
20. Markley, F.L.: Attitude determination using vector observations and the singular value decomposition. J. Astronaut. Sci. **36**(3), 245–258 (1988)
21. Schönemann, P.: A generalized solution of the orthogonal Procrustes problem. Psychometrika **31**(1), 1–10 (1966)
22. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI 2004) **26**(6), 756–777 (2004)
23. Moulon, P., Monasse, P., Marlet, R.: Adaptive structure from motion with *a Contrario* model estimation. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7727, pp. 257–270. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37447-0_20
24. Moisan, L., Stival, B.: A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. Int. J. Comput. Vis. (IJCV 2004) **57**(3), 201–218 (2004)
25. Moisan, L., Moulon, P., Monasse, P.: Fundamental matrix of a stereo pair, with a contrario elimination of outliers. Image Process. On Line (IPOL) **6**, 89–113 (2016). http://dx.doi.org/10.5201/ipol.2016.147
26. Moisan, L., Moulon, P., Monasse, P.: Automatic homographic registration of a pair of images, with a contrario elimination of outliers. Image Process. On Line (IPOL) **2**, 56–73 (2012). http://dx.doi.org/10.5201/ipol.2012.mmm-oh
27. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008). pp. 1–8, June 2008
28. von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a line segment detector. Image Process. On Line (IPOL 2012) **2**, 35–55 (2012). doi:10.5201/ipol.2012.gjmr-lsd
29. Salaün, Y., Marlet, R., Monasse, P.: Multiscale line segment detector for robust and accurate SfM. In: 23rd International Conference on Pattern Recognition (ICPR) (2016)
30. Zhang, L., Koch, R.: An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. J. Vis. Commun. Image Representation **24**(7), 794–805 (2013)
31. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV 2004) **60**(2), 91–110 (2004)
32. Liu, Z., Marlet, R.: Virtual line descriptor and semi-local matching method for reliable feature correspondence. In: British Machine Vision Conference (BMVC 2012), United Kingdom, pp. 16.1–16.11, September 2012
33. Almansa, A., Desolneux, A., Vamech, S.: Vanishing point detection without any a priori information. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI 2003) **25**(4), 502–507 (2003)
34. Agarwal, S., Mierle, K. et al.: Ceres solver. http://ceres-solver.org