Descriptor Free Visual Indoor Localization with Line Segments*

Branislav Micusik
AIT Austrian Institute of Technology
branislav.micusik@ait.ac.at

Horst Wildenauer
Zeno Track GmbH, Austria
horst.wildenauer@zenotrack.com

Abstract

We present a novel view on the indoor visual localization problem, where we avoid the use of interest points and associated descriptors, which are the basic building blocks of most standard methods. Instead, localization is cast as an alignment problem of the edges of the query image to a 3D model consisting of line segments. The proposed strategy is effective in low-textured indoor environments and in very wide baseline setups as it overcomes the dependency of image descriptors on textures, as well as their limited invariance to view point changes. The basic features of our method, which are prevalent indoors, are line segments. As we will show, they allow for defining an efficient Chamfer distance-based aligning cost, computed through integral contour images, incorporated into a first-best-search strategy. Experiments confirm the effectiveness of the method in terms of both, accuracy and computational complexity.

1. Introduction

The visual localization problem stands for estimation of the 3D location of a query image in a given 3D model from visual information only. In recent years the localization problem has been attracting ever-increasing attention in the computer vision community. Yet, most of the proposed methods assume that the 3D model consists of a sparse set of 3D points associated with their image descriptors. On the algorithmic level, the localization problem is mostly a matching challenge, *i.e*. given 2D point features with their descriptors extracted from a query image, searching for tentative 2D-3D correspondences such that re-sectioning PnP algorithms can be applied.

Standard matching procedures based on comparison of image descriptors across images have a known bottleneck. The problem is rooted in the limited descriptor invariance to certain 3D transformations. Most descriptors are invariant to affine transformations and assume planarity of the supporting image regions from which they are computed. In practice, these relaxations were shown to be feasible and many successful techniques solving the localization problem on large city scale have emerged [18].

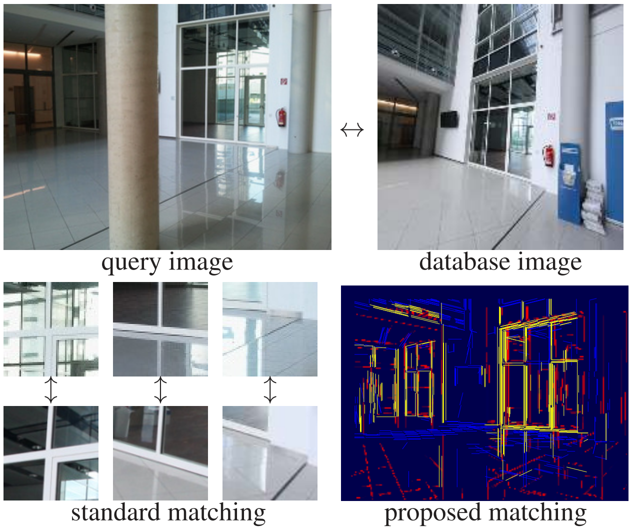

Figure 1. Top row: Two images in correspondence, however, difficult to be matched and localized in 3D by standard approaches based on point features and their local descriptors. Bottom left: Standard matching with descriptors. Three corresponding patch pairs, however, due to very different appearance with unmatchable descriptors. Bottom right: Proposed line segment alignment-based matching. Line segments in red are those detected in the query, in blue those projected from the 3D model, and in yellow those which are mutually matched.

Most of the techniques are employed on outdoor scenery, but application in indoor scenarios typically results in a significant performance drop. The reason is that indoor scenes often exhibit a lot of windows, wiry structures, reflections and repetitions, as well as limited texture, see Fig. 1. All this causes standard procedures based on image descriptors to poorly perform indoors.

In this paper, we introduce a novel proof-of-concept approach to the indoor localization problem. Aside from ubiquitous interest point + descriptor methods and representing scenes by 3D point clouds, we build a 3D model consisting

[*] This research received funding from the Austrian Research Promotion Agencys (FFG) projets LOLOG 840168, LARAH 4586620, and PAMON 835916.

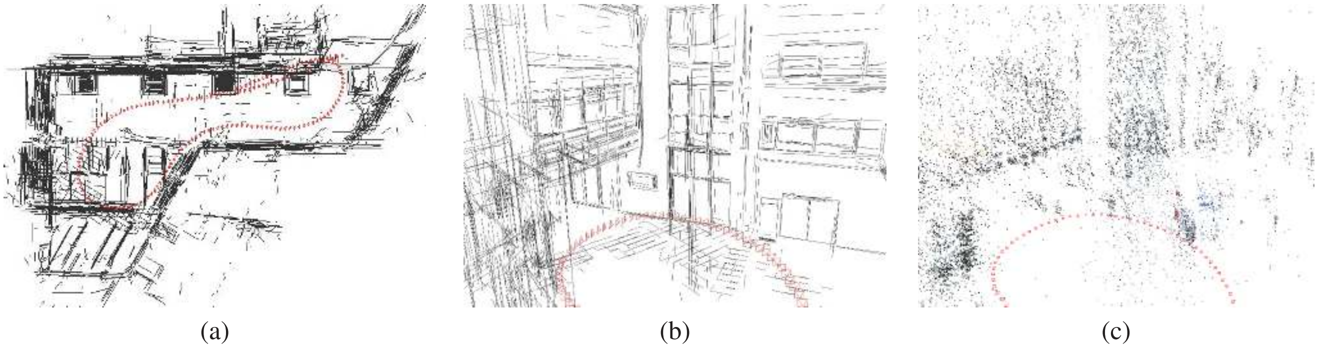

Figure 2. 3D model of an indoor scene. (a) Top view on the 3D line model as an output of a line segment-based SfM. (b) Inside view. (c) Same inside view, but on the 3D point model as an output of a standard point-based SfM. Notice that the point-based 3D model allows almost no semantic perception compared to the 3D line model, where all the structures nicely pop up.

of 3D line segments, as shown in Fig. 2. This allows the matching of the query image to the 3D model to be cast as an alignment problem between two sets of line segments. This strategy avoids a use of explicit, rather vague, line descriptors like in [2, 23]. Instead, our method harnesses geometric information provided by lines, and, despite the simplicity, copes well with extreme baseline changes that break other approaches. We show that evaluating alignment of two line sets in two images can be computed very efficiently with Chamfer matching through the integral contour strategy. Moreover, the proposed alignment cost allows for an effective tree-based search strategy to be employed, resulting in low computation time.

Compared to interest point related work, a rather limited amount of attention has been devoted to the topic of line-based localization. Despite the prevalence of lines in indoor scenery, research on lines in context of structure from motion and visual localization has not yet reached the same level of maturity as almost two decades of research on interest points has. Unfortunately, line-based methods have somewhat fallen out of the vision community's favor and making up the leeway will take time. We hope that our work, by showcasing the potential of lines in visual localization, will rekindle the interest in this topic.

2. Related Work

The visual location problem has been widely studied, mainly from the perspective of 2D-2D or 2D-3D matching. State-of-the-art approaches like [9, 17, 18, 13, 22] rely on point features with their associated descriptors. They try to effectively match a query image to the dataset of images, given a 3D model as a cloud of 3D points. The large scale nature of the problem requires techniques like visual words, kd-based approximated nearest neighbors, co-occurrence statistics of the image descriptors, *etc*., to be involved to retrieve the most likely matches for the subsequent RANSAC-based geometry validation step. However, most of the techniques work reasonably only in outdoor scenarios and fare poorly when dealing with indoor scenes.

Other primitives than point features have been studied for specific problems, *e.g*. line segments with their descriptors for two view matching of low texture images [2, 23], or image contours for object recognition [21, 19, 4, 15]. Intensity edges were shown to be powerful in this context and significant speed-ups through the integral image concept with Chamfer matching were achieved. Integral images as intermediate image representations for fast calculation of region sums were introduced in [24] and for linear sums along line contour segments in [4]. This concept was adapted for effective computation of distance transform integrals for fast directional Chamfer edge matching in [15]. Chamfer matching itself is a well known strategy for measuring distance between contours [1], and has been significantly enhanced in [21] by encoding edge directions into the criterion function in order to allow for more accurate matching.

In this paper we show that indoor scenes can be modeled as a piece-wise wiry structure since they usually exhibit a rich population of line segments. Line segments, as linear representation of edgels, fit well into contour-based matching frameworks and we show how the matching cost can be designed to obtain a highly effective tree-based retrieval algorithm. The proposed concept does not replace point-based localization strategies, but offers a complementary strategy for scenes where they are insufficient.

Our method has been designed to allow for inconsistencies in detection of line segments across different views. In practice, it is impossible to guarantee a perfect, repeated detection of the same line segment in different views, because the endpoints are rather unstable. That is, long line segments are often broken up into several shorter pieces. This is for example caused by illumination differences across images, imperfect radial image undistortion, and occlusions due to viewpoint changes. Thus, insensitivity of the matching, *i.e*. the comparison of image line segments to their 3D model counterparts, to endpoint misalignments and location

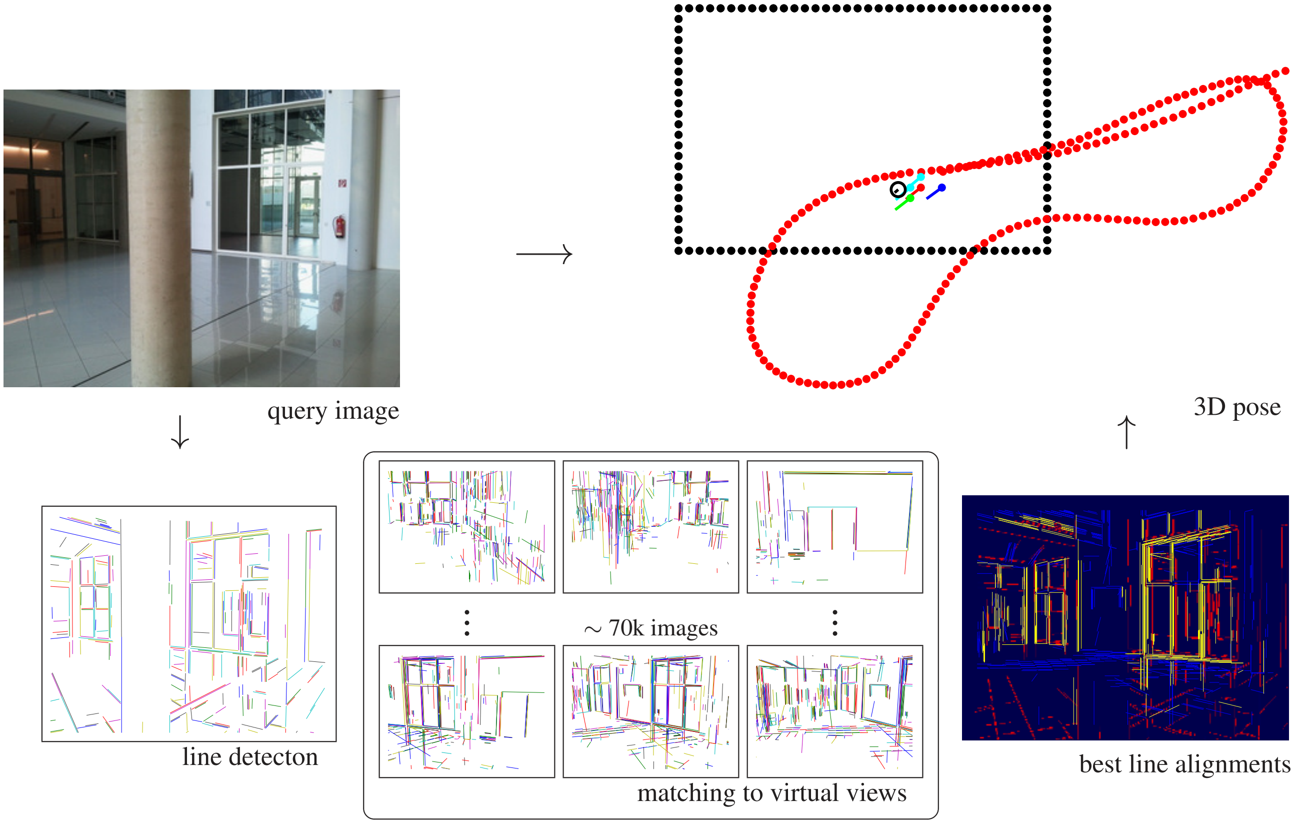

Figure 3. Concept of the localization. Bottom from left: Line segments are detected in the query image and are used to match it to virtual views by effectively evaluating their mutual alignments. Top right: The best alignment yields 3D pose of the query in the sampled cube whose edges are shown as black dotted rectangle. The best matches are depicted according to better score as a red, green, blue and cyan short line with a dot. 3D trajectory of a mapping sequence is shown in red, the ground truth pose of the query as a small black circle.

of the splitting points is very important practical feature.

3. Image localization

We understand visual localization as the process of estimating the position and orientation of a query camera image in a given 3D model. Here, the 3D model consists of line segments with no explicit image descriptors. Pose estimation is cast as an alignment problem, see Fig. 3, such that the projected line segments from the 3D model align with the edges in the query image. This is achieved by dense sampling of the 3D model, producing so called virtual views. These virtual views are then compared based on the proximity of their line edges to the line edges of the query image. Edges underlying line segments are shown to facilitate efficient evaluation of alignment quality, which allows for a large number of virtual views to be tested. Knowing the internal calibration of the camera of a query image is advantageous, but is not necessary.

3.1. Building 3D Model of Line Segments

The key component of the proposed localization framework is the representation of the scene by a reconstructed set of 3D line segments, as shown in Fig. 2. Numerous techniques for building such 3D models from a sequence of images taken by a calibrated camera exist. There are basically two groups of methods. First, techniques which simultaneously reconstruct line segments and camera poses [5, 20, 7, 16]. Second, techniques utilizing point features for estimating the poses of the camera followed by a guided line segment reconstruction [25, 10, 8].

In our work, we use the technique of [16] belonging to the first group. Preliminary experiments in different indoor environments showed us that there are usually enough point features for incremental pose estimation when a wide-angle lens and small baselines are employed. However, for scene representation the points are insufficient as the reconstructed set of 3D points is rather sparse. Employing a guided line segment reconstruction given the poses provides a rich and versatile scene representation which, as we will demonstrate, serves as a base for our novel localization

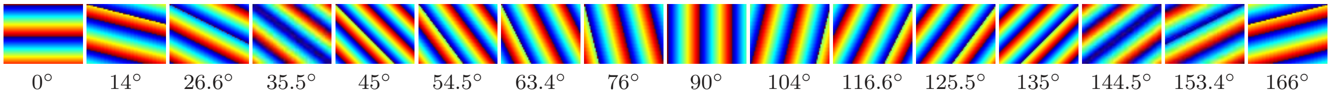

Figure 4. Sampled directions for integral contour computation. Color is used to make the directions visible and to show no holes in the rasterization of the direction.

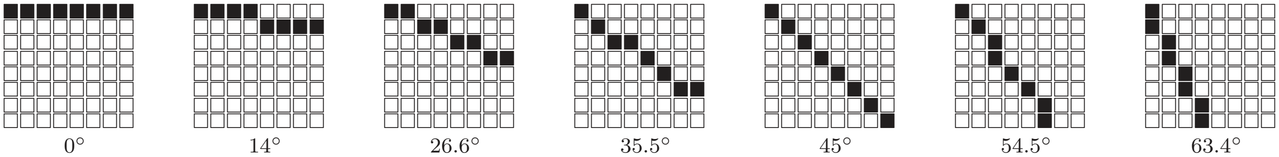

Figure 5. Discretization of the sampled directions.

strategy.

3.2. Virtual Cameras

Given the 3D model consisting of line segments, we sample the space of possible locations and orientations of the query image. At each sampled location and orientation we project the line segments into the virtual view, as shown in Fig. 3. We assume that the coordinate system of the 3D model is metric and has one axis perpendicular to the ground plane. This allows to create slices of sampled planes and to restrict the amount of sample positions.

We sample the location in each slice in steps of 30 cm, and orientation in pan, tilt and roll angle in steps 10 degrees. Projections indexed by tilt, roll, and focal length are stored separately. A priori knowledge of any of the angles or the focal length allows to significantly reduce the computational burden. Moreover, in the retrieval stage, this hash table strategy directly reads only virtual cameras with the closest tilt, roll, and focal length to the query image, and brings efficient memory management.

3.3. Matching by Alignment

Matching by alignment solves the following problem. Given a pair of images containing line segments, evaluate the cost of aligning one set of line segments with the second set. This yields a quadratic complexity as each line from one set needs to be compared to all lines in the second set. Now, query image needs to be compared to all virtual views whose number might reach 100k. Exact nearest neighbor search is simply prohibitive, even when some hash table speed-up strategies are involved. Furthermore, preliminary experiments indicated that also approximated nearest neighbor using kd-trees performs poorly.

Instead, we propose to incorporate so called integral contour strategy with use of Chamfer distance for evaluating the alignment. Chamfer distance [1] measures the discrepancy of two contours, and in its basic form is defined as the cost of aligning two edge maps $\mathcal{E} = \{\mathbf{x}_{ei}\}_{i=1}^{N_\mathcal{E}}$ and $\mathcal{T} = \{\mathbf{x}_{ti}\}_{i=1}^{N_\mathcal{T}}$ as

$$d(\mathcal{E},\mathcal{T}) = \frac{1}{N_\mathcal{T}} \sum_{\mathbf{x}_t \in \mathcal{T}} \min_{\mathbf{x}_e \in \mathcal{E}} \|\mathbf{x}_e - \mathbf{x}_t\|. \tag{1}$$

The min operation in Eq. (1) can be replaced by a look-up in the distance transform (DT) image $\mathtt{I}_{\mathrm{DT}\mathcal{E}}$ of the edge map $\mathcal{E}$ which can be computed in linear time. To achieve robustness against noise in edgels, it is a standard practice to truncate the DT image

$$d(\mathcal{E},\mathcal{T}) = \frac{1}{N_\mathcal{T}} \sum_{\mathbf{x}_t \in \mathcal{T}} \mathtt{I}_{\mathrm{DT}\mathcal{E}}(\mathbf{x}_t, \gamma), \tag{2}$$

where γ is a threshold for truncating the DT values, typically set to about 10 pixels.

A popular extension to basic Chamfer matching is to divide the edge map and the template into discrete orientation channels θ_i and sum the individual Chamfer scores, as done in context of hand detection in [21]. In our work, we relax the edge maps to line segment maps. Edgels in $\mathcal{E}$ are split into piece-wise linear segments, using standard line detection algorithms, *e.g.* we found the one described in [3] suitable. We consider 16 discrete orientation channels, 0, 14, 26.6, ..., 166°, and assign to each edgel the discrete orientation closest to the orientation of the fitted line segment. The values for discrete orientations come from the rasterization of scan-lines in the image, as shown in Fig. 4 and in Fig. 5 such that the scan lines fully span the image without holes. We compute sixteen DT images $\mathtt{I}^{\theta_i}_{\mathrm{DT}\mathcal{E}}$ on filtered binary images. A filtered binary image is an image composed of edgels from the original edge map $\mathcal{E}$ whose assigned discrete orientations are θ_i .

The sum operation in Eq. (2) is expensive, considering the number of edgel pixels. We therefore adopt the concept of integral contours. An integral contour image sums pixels along a respective scan line, as shown in Fig. 4 and in Fig. 5. To compute it off-line, it needs one pass through the query image. By this we produce so called integral distance transform (IDT) images $\mathtt{I}^{\theta_i}_{\mathrm{IDT}\mathcal{E}}$, analogously to [15]. Evaluating the sum for one line segment becomes now $\mathcal{O}(1)$, *i.e.* just two arithmetic operations, see Eq. (4). Instead of summing along hundreds of pixels of the edgels one just looks up into the $\mathrm{I}^{\theta_i}_{\mathrm{IDT}\mathcal{E}}$ at the endpoints of the line segment and divides by its length. This yields a significant saving in computations, resulting in a dramatic speed-up during the matching.

For the template map $\mathcal{T}$, in our case corresponding to a virtual view, we split the projected lines of the 3D model into smaller ones and snap them into the discrete orientations, as shown in Fig. 6. The splitting into the smaller line segments is controlled by requiring a sufficiently accurate rasterized representation of the original line segment. The template map becomes a map of line segments expressed by starting and end points of the line segment, $\mathcal{T}^l = \{\mathbf{l}_i\}_{i=1}^{N_{\mathcal{T}^l}}$, where $\mathbf{l}_i = [\mathbf{x}^s_i; \mathbf{x}^e_i]$ is a four element vector with two x, y coordinates. The distance of the two edge maps can be then finally evaluated as

$$d(\mathcal{E}, \mathcal{T}) = \frac{1}{N_{\mathcal{T}^l}} \sum_{\mathbf{l} \in \mathcal{T}^l} d(\mathcal{E}, \mathbf{l}). \tag{3}$$

with

$$d(\mathcal{E}, \mathbf{l}) = \left(\mathrm{I}^{\theta(\mathbf{l})}_{\mathrm{IDT}\mathcal{E}}(\mathbf{x}^e, \gamma) - \mathrm{I}^{\theta(\mathbf{l})}_{\mathrm{IDT}\mathcal{E}}(\mathbf{x}^s, \gamma)\right)/l_{len}. \tag{4}$$

The function $\theta(\mathbf{l})$ assigns a discrete orientation to a line segment $\mathbf{l}$ as its closest snapping orientation. Length of the line is denoted as l_{len} and is calculated as $\|\mathbf{x}^e - \mathbf{x}^s\|_2$.

The matching is commonly cast as searching for the template $\mathcal{T}$ from the set of templates $\mathcal{T}_{set}$ which minimizes the distance d as

$$\mathcal{T}^* = \arg \min_{\mathcal{T} \in \mathcal{T}_{set}} d(\mathcal{E}, \mathcal{T}).$$

We propose to use different strategy where the best template is chosen as

$$\mathcal{T}^* = \arg \max_{\mathcal{T} \in \mathcal{T}_{set}} c(\mathcal{E}, \mathcal{T}). \tag{5}$$

with

$$c(\mathcal{E}, \mathcal{T}) = \frac{1}{N_{\mathcal{T}^l}} \sum_{\mathbf{l} \in \mathcal{T}^l} \delta\big(d(\mathcal{E}, \mathbf{l}) < \gamma\big), \tag{6}$$

where $\delta(d(\mathcal{E}, \mathbf{l}) < \gamma)$ is a binary function returning 1 if the inequality is fulfilled, 0 otherwise. $N_{\mathcal{T}^l}$ is the number of line segments in the template map $\mathcal{T}^l$. The threshold γ controls the width of the strip around the line in which a matching line can lie, set to 10 pixels in all our experiments. In words, the matching cost in Eq. (6) selects the template map which has the highest number of matching lines. The same counting strategy is commonly utilized in RANSAC-based estimation procedures across different CV tasks. As we further show, this form of the matching cost allows to use a very effective tree-based search. The cost $c(\mathcal{E}, \mathcal{T})$ gets values in the $\langle 0 \;\; 1 \rangle$ interval, with 0 meaning that none of the line segments matches, 1 that all line segments match, respectively.

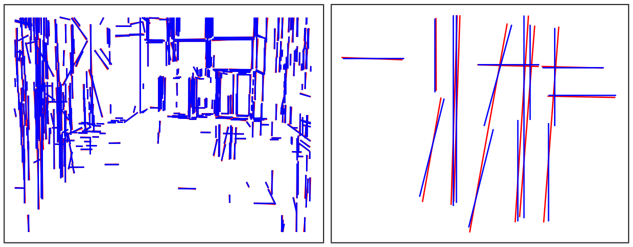

Figure 6. Snapping and splitting lines of virtual views to the 16 sampled directions. Red lines are the projected 3D lines. Blue lines are the snapped and split approximated lines further used for alignment calculations. Left: A virtual view. Right: A zoom.

3.4. Search Optimization

The form of the cost function in Eq. (6) facilitates a best-first search strategy, used *e.g.* in A^* algorithm. It allows to reach the maximum without evaluating all the summands of the sum operation. In experiments we found that to retrieve the 1-best candidate template map, only 50% of all calculations are needed to be compared to the fully evaluated sum.

The implementation of the search procedure is given in Algorithm 1. In principle, all template maps (virtual views) are proposed to be processed simultaneously, each line after the other, switching between the template maps, depending on the actual upper bound of the cost

$$\bar{c}(\mathcal{E}, \mathcal{T}_{1:i}) = \frac{N_{\mathcal{T}^l} - i + i * c(\mathcal{E}, \mathcal{T}_{1:i})}{N_{\mathcal{T}^l}}, \tag{7}$$

where $\mathcal{T}_{1:i}$ stands for the template map with the first ith line segments. Eq. (7) reflects the best case cost if all remaining line segments match. The best-first search of Algorithm 1 can be efficiently realized using a priority queue which *e.g.* is a standard container adaptor of C++. We retrieve the 10 best matches in all our experiments, which results on average in 60% evaluations of all line segments.

3.5. Search Reduction

The search for the best aligning virtual view to the query can be reduced by extracting additional information from the query image. For example, the tilt and roll angle of the query camera can be estimated from a vertical vanishing point. The vertical vanishing point corresponds to the direction of gravity and can be reliably detected, independent from the location in a building. We do not consider approaches assuming three orthogonal vanishing points, the so called Manhattan world assumption, where a full rotation in the global coordinate system can be estimated. In modern buildings, the horizontal vanishing directions often do not fulfill the Manhattan world assumption, and therefore it is not safe to rely on them.

We use the strategy for estimating the vertical vanishing point and the radial distortion from [27]. It allows us to directly fix the roll and tilt angle and compare the query image

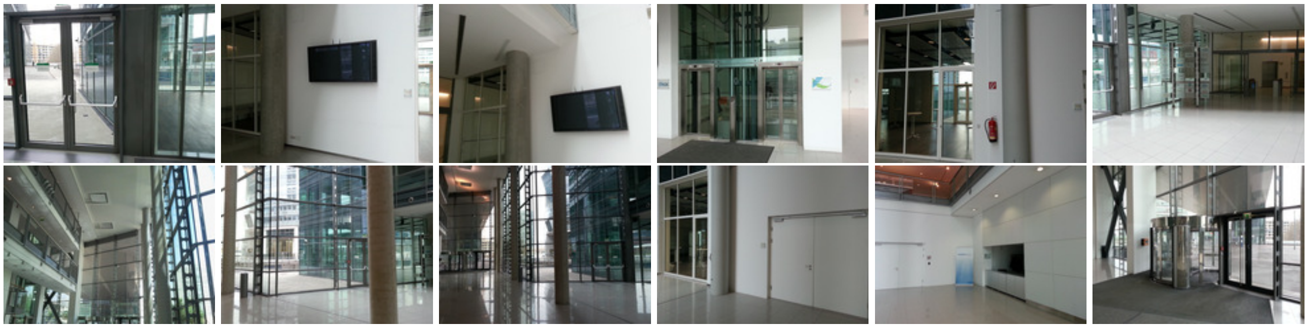

Figure 7. Some of the testing query images taken with a mobile phone camera.

Algorithm 1 Search for the K best alignments to a query

```
detect line segments in query, giving a set $\mathcal{E}$
compute integral contour images $\mathtt{I}^{\theta}_{\mathrm{IDT}\mathcal{E}}$ at 16 orientations of $\theta$
set upper bounds $\bar{c}(\mathcal{E}, \mathcal{T}) \leftarrow 1$ for all virtual views $\mathcal{T} \in \mathcal{T}_{\mathrm{set}}$
set counters of processed lines $i_{\mathcal{T}} \leftarrow 1$ for all $\mathcal{T}$
set counter of number of best matches $k \leftarrow 0$
push $\bar{c}(\mathcal{E}, \mathcal{T})$ of all $\mathcal{T}$ into the priority queue
repeat
    pop $\mathcal{T}$ from the priority queue with highest $\bar{c}(\mathcal{E}, \mathcal{T})$
    evaluate $i_{\mathcal{T}}$th line by scoring $\bar{c}(\mathcal{E}, \mathcal{T}_{1:i_{\mathcal{T}}})$ in Eq. (7)
    $i_{\mathcal{T}} \leftarrow i_{\mathcal{T}} + 1$
    if $i_{\mathcal{T}} = N_{\mathcal{T}^l}$ then
        $k \leftarrow k + 1$
    else
        push $\bar{c}(\mathcal{E}, \mathcal{T}_{1:i_{\mathcal{T}}})$ into the priority queue
until $k = K$
$K$ best matches found
```

against the virtual views with the closest of the two angles. The tilt and roll angle can be alternatively obtained from the Inertial Measurement Unit (IMU) which is a common part of mobile phones or smart cameras nowadays.

The same strategy can be applied for the focal length which can be estimated from at least two perpendicular vanishing points or is simply known [6, 26]. After fixing the roll and tilt angles, the focal length, and the radial distortion, the remaining pan angle and the location are the subjects to be estimated.

4. Experiments

Most of the datasets which are used for evaluation of visual localization algorithms are from outdoor large scale city [9, 17, 18, 13] or well textured indoor lab environments [14]. In this paper we tackle poorly textured indoor environments and are primarily interested in localization accuracy under 1m. To be able to judge the results in sense of such accuracy we created our own dataset from indoors with 50 images and laboriously created Ground Truth.

We chose an entrance foyer of a modern building 20m long and 10m wide with many glass walls, repetitive structures, reflective tiles as a representative difficult scene, shown in Fig. 7. For accurate 3D modeling of the scene, we used a camera equipped with a 180 degree field of view fisheye lens and made a loop closed trajectory with 200 images. We applied standard point- and our line-based Structure from Motion pipeline of [16]. We obtained a 3D cloud of points and line segments, respectively, shown in Fig. 2. For establishing Ground Truth of a query image we manually established correspondences between the query and the moving sequence which is used for SfM estimate. We applied the P3P re-sectioning algorithm of [12] to estimate full pose of the camera, followed by a bundle adjustment.

Mobile camera We used a mobile phone camera with known internal calibration to acquire 50 test images with VGA resolution. The time between the acquisition of the 3D model and the images of a mobile phone was roughly two years. This makes the localization harder because of some changes in the environment and different lighting conditions.

The scene is split into three sectors and sampled as three slightly overlapping 3D blocks. The 3D block consists of three parallel 10m x 6m rectangles sampled with 30cm grid, one of them shown in Fig. 3, at three different heights, shifted by 30cm. At each sampled location the virtual camera orbits with the pan angle of 10 degs. This results in 70k virtual views, as depicted in Fig. 3. Some of the query images are shown in Fig. 7.

We compared our approach to two other approaches, *i.e.* a naive point-based approach and to [18]. The slow naive approach takes a query and finds the best match via matching it to all dataset images consecutively, validated through a RANSAC-based estimation with the P3P re-sectioning algorithm of [12]. The approach of [18] is a sped up version of the naive approach to handle large scale problems. It is mainly suited for large outdoor datasets where the speed of inference is of concern. To make the approach effective in-

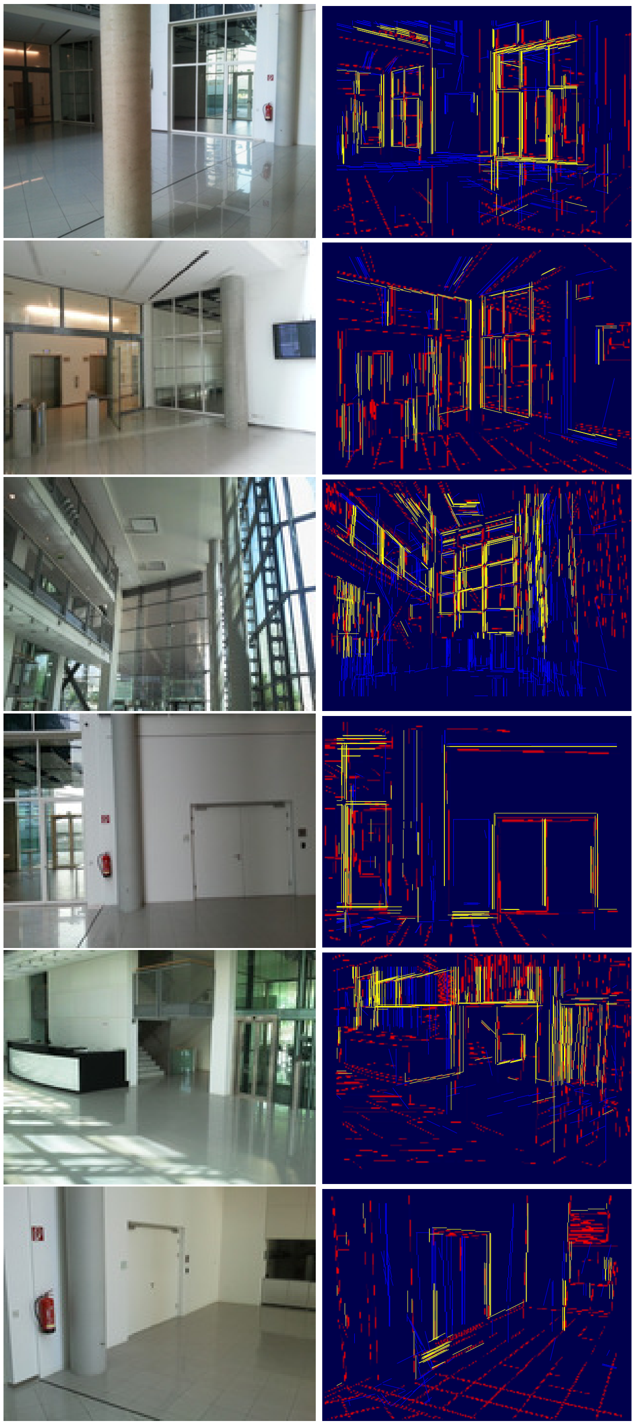

Figure 8. Query images with the best aligned virtual views. In red are depicted the lines of the query image, in blue the lines of the best matching virtual view. In yellow are the matching lines. The images are best seen in color.

door we learned new vocabulary trees from indoor datasets.

As can be seen from the distribution of pose errors in Fig. 9, within an accuracy of 1 meter only 27 cameras out of 50 can be localized with the naive point-based approach and 12 with the approach of [18]. Our line-based approach outperforms both with 41 successfully localized cameras. The peak of the histogram for the proposed method is more to the left with majority of the estimated poses within 30 cm. For the point-based naive method, the peak is around 60 cm and the distribution is more flat. For the point-based method of [18], the peak is around 1 meter. The proposed line-based approach clearly outperforms the two point-based counterparts. Some of the successful alignments are depicted in Fig. 8.

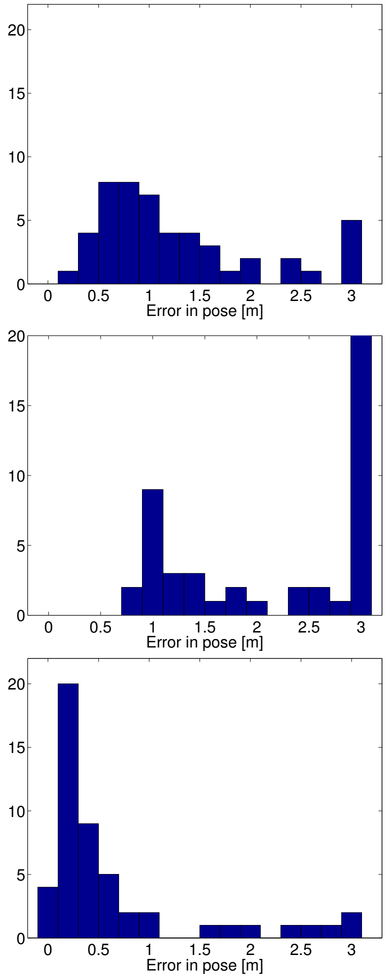

Figure 9. Distribution of the error in camera centers for the point-based naive (top) and of Sattler *et al*. [18] (middle) method. Our line-based is shown at the bottom. Those images which cannot be localized or are localized with error greater than 3 m, are all put into the bucket of 3 m.

Surveillance cameras We applied the proposed method to surveillance cameras installed in the building at the height of roughly 5 meters from the ground plane. The cameras were internally calibrated by the automatic method of [27] using vanishing points. This experiment emphasizes the strength of the proposed method. The surveillance cameras look at the scene from very different viewpoints compared to the moving camera which was used for building the 3D model. The change in viewpoint exceeds the limits of invariance of image descriptors. As a result none of the cameras can be localized with the point features and their descriptors with the naive approach and with the approach described in [18].

On the contrary, the proposed localization succeeded even under such difficult conditions, demonstrating the advantage of the descriptor free concept. Fig. 10 shows three surveillance cameras with the best aligning virtual cameras. As can be seen, the fully automatically estimated poses all lie around the position which was achieved from manually established correspondences and the P4P re-sectioning algorithm of [11]. Virtual cameras were sampled in a block 4 x 4 x 2 meters around rough positions of the cameras.

Calibration of non-overlapping cameras w.r.t. to a given 3D model is of great importance for surveillance applications. However, it is still very challenging due to the nature of indoor environments and the large baseline differences when matching the images of surveillance cameras to a 3D model. The proposed method can fill the gap here as the presented descriptor free localization method can handle much wider baselines and successfully cope with strong illumination changes. Moreover, in such an application scenario, there is typically no requirement on real time processing and the rough locations of the cameras are known which is in favor of the presented method.

Performance We implemented the localization Algorithm 1 in C++, resulting in the following performance statistics. To find 10 best matches takes 0.1 sec on 5k virtual cameras and 1.5 sec on 60k virtual cameras with, in average, 300 lines per virtual camera, on Intel Core i7 CPU 970 @3.2GHz. Computation and storing the virtual views is done off-line. In many application scenarios a priori knowledge about camera's approximate location can be safely assumed (*e.g.* wifi triangulation in smart devices, camera calibration in surveillance networks) such that pose computation on a "room-level" should be sufficient. Or, a robot moving in a known environment would utilize pose predictions which considerably shrink the search space. A way of memory saving for price of increasing of the computational time, is usage of dedicated hardware (GPUs) which would make on-demand rendering of virtual views possible, *i.e.* only the 3D reconstruction needs to be stored.

Considering the computational cost and the scalability, the presented method surely cannot compete with the state-of-the-art in point-based methods. This is not surprising as the research on interest points/descriptors spans well over several decades where the community drew heavily from machine learning and CBIR developments.

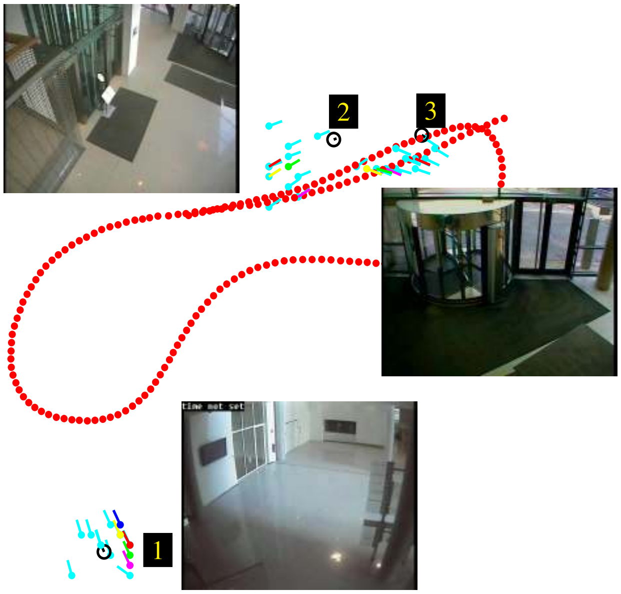

Figure 10. Localization of surveillance cameras. Small circles depict positions estimated from manually established point correspondences. Short lines with dots are the best fifteen (in order red, green, blue, magenta, yellow, cyan) returned virtual cameras obtained with the proposed method.

5. Conclusion

We presented a novel approach to the indoor visual localization problem, demonstrating that the adoption of line segments for indoor scene representation and localization compares favorably to the state-of-the-art. Line segments represent richer features than points, which allows for a matching strategy that fully avoids the need of image descriptors. Our descriptor-free matching handles scenes with low texture and query images with wider baseline to the modeling sequence far better than state-of-the-art point-based methods. Furthermore, we showed that despite the high number of virtual views, line segments allow for efficient evaluation of the matching cost. This results in an effective localization strategy, complementing existing point-based methods.

We hope that the promising results will spark interest and encourage further investigation of the utility of line segments in structure from motion and localization. A future research direction is the unified treatment of lines and points to produce practical solutions that can overcome the weaknesses of the complementary techniques. Furthermore, although we argued for purely descriptor free matching from a conceptual point of view, we acknowledge that the use of weak line descriptors to filter out severe mismatches could contribute towards enhancing the performance.

References

[1] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI*, pages 659–663, 1977.

[2] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *Proc. CVPR*, 2005.

[3] J. C. Bazin and M. Pollefeys. 3-line ransac for orthogonal vanishing point detection. In *Proc. IROS*, 2012.

[4] C. Beleznai and H. Bischof. Fast human detection in crowded scenes by contour integration and local shape estimation. In *Proc. CVPR*, 2009.

[5] R. Deriche and O. Faugeras. Tracking line segments. In *Proc. ECCV*, 1990.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[7] K. Hirose and H. Saito. Fast line description for line-based slam. In *Proc. BMVC*, 2012.

[8] M. Hofer, A. Wendel, and H. Bischof. Incremental line-based 3D reconstruction using geometric constraints. In *Proc. BMVC*, 2013.

[9] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *Proc. CVPR*, June 2009.

[10] A. Jain, C. Kurz, T. Thormählen, and H.-P. Seidel. Exploiting global connectivity constraints for reconstruction of 3D line segments from images. In *Proc. CVPR*, 2010.

[11] K. Josephson and M. Byröd. Pose estimation with radial distortion and unknown focal length. In *Proc. CVPR*, 2009.

[12] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proc. CVPR*, 2011.

[13] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3D point clouds. In *Proc. ECCV*, 2012.

[14] H. Lim, S. N. Sinha, M. Cohen, and M. Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *Proc. CVPR*, pages 1043–1050, 2012.

[15] M.-Y. Liu, C. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *Proc. CVPR*, 2010.

[16] B. Micusik and H. Wildenauer. Structure from motion with line segments under relaxed endpoint constraints. In *Proc. 3DV*, 2014.

[17] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *Proc. ICCV*, 2011.

[18] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *Proc. ECCV*, 2012.

[19] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *PAMI*, 30(7):1270–1281, July 2008.

[20] P. Smith, I. Reid, and A. Davison. Real-time monocular slam with straight lines. In *Proc. BMVC*, 2006.

[21] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *PAMI*, 28(9):1372–1384, 2006.

[22] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate localization and pose estimation for large 3D models. In *Proc. CVPR*, 2014.

[23] B. Verhagen, R. Timofte, and L. Van Gool. Scale-invariant line descriptors for wide baseline matching. In *Proc. Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[24] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.

[25] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. In *Proc. ECCV*, 2002.

[26] H. Wildenauer and A. Hanbury. Robust camera self-calibration from monocular images of Manhattan worlds. In *Proc. CVPR*, 2012.

[27] H. Wildenauer and B. Micusik. Closed form solution for radial distortion estimation from a single vanishing point. In *BMVC*, 2013.