Learning to Identify Correct 2D-2D Line Correspondences on Sphere

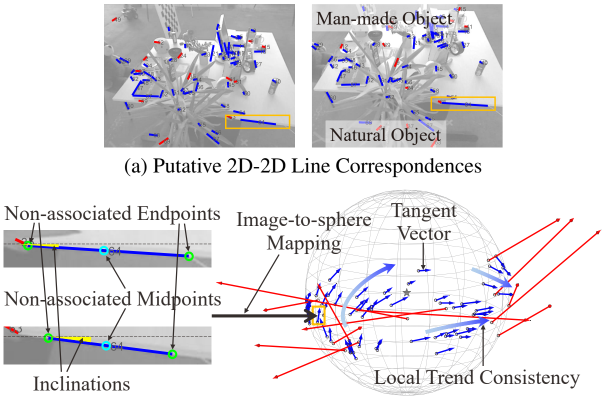
Haoang Li[1] Kai Chen[1] Ji Zhao[2] Jiangliu Wang[1] Pyojin Kim[3] Zhe Liu[4] Yun-Hui Liu[1]
[1]The Chinese University of Hong Kong, Hong Kong, China [2]TuSimple, China
[3]Sookmyung Women's University, South Korea [4]University of Cambridge, United Kingdom

Abstract

Given a set of putative 2D-2D line correspondences, we aim to identify correct matches. Existing methods exploit the geometric constraints. They are only applicable to structured scenes with orthogonality, parallelism and coplanarity. In contrast, we propose the first approach suitable for both structured and unstructured scenes. Instead of geometric constraint, we leverage the spatial regularity on sphere. Specifically, we propose to map line correspondences into vectors tangent to sphere. We use these vectors to encode both angular and positional variations of image lines, which is more reliable and concise than directly using inclinations, midpoints or endpoints of image lines. Neighboring vectors mapped from correct matches exhibit a spatial regularity called local trend consistency, regardless of the type of scenes. To encode this regularity, we design a neural network and also propose a novel loss function that enforces the smoothness constraint of vector field. In addition, we establish a large real-world dataset for image line matching. Experiments showed that our approach outperforms state-of-the-art ones in terms of accuracy, efficiency and robustness, and also leads to high generalization.

1. Introduction

2D-2D correspondences of points and lines[1] are the basis of numerous computer vision algorithms [1, 4, 34]. Putative correspondences can be obtained by various methods [26, 41, 15]. In practice, these correspondences consist of correct matches, i.e., inliers and mismatches, i.e., outliers. Outliers are caused by viewpoint differences and repetitive patterns. Since outliers drastically affect the algorithm robustness, it is important to identify inliers. Identifying inlier *point* correspondences has been widely studied. Most existing methods are based on geometric constraint [31, 38] or spatial regularity [44, 42]. The geometric constraint-based methods leverage the fact that all the inliers can be fitted by the same parametric model, e.g., essential matrix [12]. The spatial regularity-based methods generate 2D displacement vectors, i.e., optical flow [27] by connecting point correspondences. They leverage the fact that vectors generated by inliers are regular. The above methods are all applicable to both structured (typically man-made) scenes with orthogonality, parallelism and coplanarity [21], and unstructured (typical natural) scenes [40].



(a) Putative 2D-2D Line Correspondences

(b) Zoom View of Fig. 1(a)

(c) Spatial Regularity of Vectors

Figure 1. (a) A pair of lines with the same number represents a 2D-2D line correspondence. Putative correspondences consist of inliers (blue) and outliers (red). (b) Baseline methods use the inclinations, midpoints, or endpoints to encode the angular, positional, or both angular and positional variations of image lines, respectively. (c) We map line correspondences into vectors tangent to sphere. Neighboring vectors mapped from inliers exhibit a local trend consistency (analogous to "a school of fish").

Compared with the above point problem, identifying inlier *line* correspondences (see Fig. 1(a)) is more challenging and has not been well studied. Existing geometric constraint-based methods [9, 43] are only suitable for structured scenes since inliers are only geometrically constrained in these scenes. A spatial regularity-based method can theoretically handle both structured and unstructured scenes, but how to design such a method remains an open question. Specifically, we express the spatial regularity of line correspondences by both angular and positional variations of image lines. As shown in Fig. 1(b), it is straightforward to use the inclinations, midpoints, or endpoints to encode the angular, positional, or both angular and positional variations

*Pyojin Kim and Yun-Hui Liu are co-corresponding authors.

[1] We use "line" to represent "line segment" for writing simplification.

of image lines, respectively. However, as will be shown in Sections 3 and 6, these baseline methods may be affected by non-association and ambiguity problems, and thus result in unsatisfactory accuracy. Overall, a method suitable for both structured and unstructured scenes does not exist.

We propose the first approach, which is applicable to both structured and unstructured scenes, to identify inlier line correspondences. Instead of geometric constraint, we leverage the spatial regularity on sphere. Specifically, we propose to map line correspondences into vectors tangent to sphere. We use these vectors to encode both angular and positional variations of image lines, which is more reliable and concise than directly using inclinations, midpoints or endpoints of image lines. As shown in Fig. 1(c), neighboring vectors mapped from inliers (but not outliers) exhibit a spatial regularity called local trend consistency, regardless of the type of scenes. To encode this regularity, we design a neural network and also propose a novel loss function that enforces the smoothness constraint of vector field [29]. In addition, existing datasets for image line matching are either small [24, 32] or only provide synthetic images [17]. To solve this problem, we establish a large real-world dataset composed of 11,934 image pairs, and provide a tool to automatically extend it. Our main contributions are:

- We propose a novel image-to-sphere mapping to generate vectors tangent to sphere. These vectors solve the non-association and ambiguity problems, and inlier vectors exhibit a spatial regularity.
- We propose a novel loss function to learn the spatial regularity of vectors tangent to sphere. Accordingly, our approach is the first one applicable to both structured and unstructured scenes.
- We establish a large real-world dataset for image line matching. Our dataset and tool for dataset extension are publicly available.[2]

Experiments showed that our approach outperforms state-of-the-art ones in terms of accuracy, efficiency and robustness, and also leads to high generalization.

2. Related Works

Image Line Extraction. Many methods, e.g., LSD [36] and Linelet [7] exploit the image gradient. They first detect edges based on gradient magnitude and then group these edges into lines based on gradient orientation. However, they are prone to resulting in incomplete and ambiguous line extraction. In contrast, state-of-the-art deep learning-based method [37] avoids edge detection. It reformulates line extraction as a region coloring problem and solves this problem by learning the regional attraction.

Image Line Matching. This work generates putative line correspondences, i.e., the input of inlier identification approaches introduced in the next paragraph. Most existing methods measure the similarity between two lines in terms of visual appearance around these lines. Visual appearance can be encoded by either handcrafted features or the features learned by neural networks. For example, LBD descriptor [41] is a widely used handcrafted feature. It computes the gradient histogram around an independent line. RPR descriptor [43] is another handcrafted feature describing an area defined by a pair of lines. DLD descriptor [17] is learned by a neural network. It is more suitable for low-textured environments than the above handcrafted features.

Identifying Inlier Line Correspondences. A spatial regularity-based method does not exist since endpoints of a line correspondence may be non-associated. In the following, we introduce existing geometric constraint-based methods. We classify them into two categories in terms of application scenarios, i.e., two or three images. On two images, geometric constraints rely on particular structures. Specifically, relative image pose geometrically constrains the inliers of 2D-2D line correspondences associated with 1) orthogonal and parallel 3D lines [9], or 2) coplanar 3D lines [43]. The integration of a relative pose estimation method [9, 43] and RANSAC [10] computes the optimal pose fitting the largest number of correspondences, and also treats these correspondences as inliers. The above integrations are not very practical for two reasons. First, they are only applicable to structured scenes. Second, their used correspondences satisfying the structure assumptions are relatively difficult to sample. In contrast, our approach is suitable for both structured and unstructured scenes. Moreover, it provides higher accuracy, efficiency and robustness than the above integrations, as will be shown in the experiments.

On three images, trifocal tensor [11] geometrically constrains the inliers of 2D-2D-2D line correspondences. Micusik et al. [30] used the integration of a trifocal tensor estimation method and RANSAC to compute trifocal tensor and identify inliers. However, computing trifocal tensor commonly requires more than 9 inliers [18]. Sampling such inliers requires a large number of iterations, resulting in unsatisfactory efficiency and robustness. Another inlier identification method [20] begins with using the known poses of the first two images to triangulate 3D lines. Then it matches these 3D lines against 2D lines in the third image whose pose is unknown. Given these putative 3D-2D line correspondences, the integration of an absolute image pose estimation method and RANSAC computes the pose and identifies inliers. Since this method requires the known poses of the first two images, its generality is low.

3. Image-to-sphere Mapping

To explore the spatial regularity of line correspondences, we propose to map line correspondences into vectors tangent to sphere. We use these vectors to encode both angular

[2] https://sites.google.com/view/haoangli/projects/inlier_line_corres

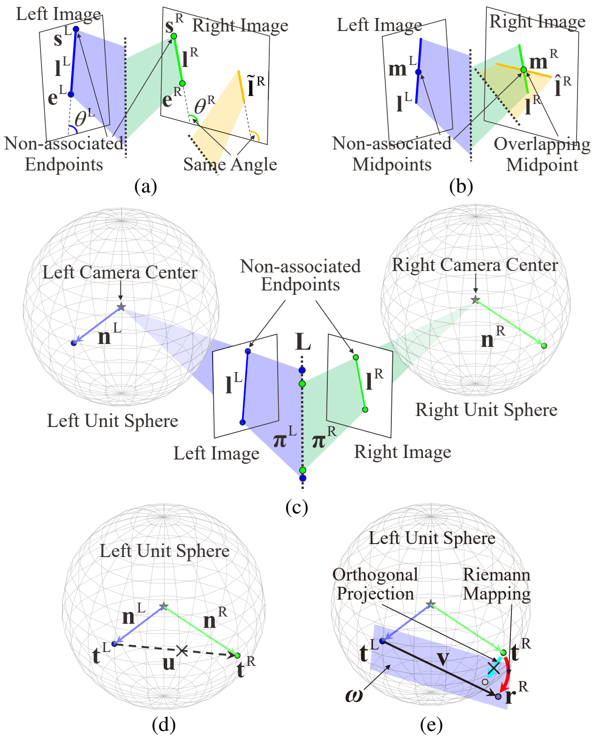

Left Image
$\mathbf{s}^{\mathrm{L}}$
$\mathbf{l}^{\mathrm{L}}$
$\mathbf{e}^{\mathrm{L}}$
θ^{L}
$\mathbf{s}^{\mathrm{R}}$
Right Image
$\mathbf{l}^{\mathrm{R}}$
$\mathbf{e}^{\mathrm{R}}$
θ^{R}
$\tilde{\mathbf{l}}^{\mathrm{R}}$
Non-associated Endpoints
Same Angle
(a)
Left Image
$\mathbf{m}^{\mathrm{L}}$
$\mathbf{l}^{\mathrm{L}}$
Right Image
$\mathbf{m}^{\mathrm{R}}$
$\hat{\mathbf{l}}^{\mathrm{R}}$
$\mathbf{l}^{\mathrm{R}}$
Non-associated Midpoints
Overlapping Midpoint
(b)
Left Camera Center
Non-associated Endpoints
Right Camera Center
$\mathbf{n}^{\mathrm{L}}$
$\mathbf{L}$
$\mathbf{n}^{\mathrm{R}}$
$\mathbf{l}^{\mathrm{L}}$
$\mathbf{l}^{\mathrm{R}}$
Left Unit Sphere
Right Unit Sphere
$\boldsymbol{\pi}^{\mathrm{L}}$
$\boldsymbol{\pi}^{\mathrm{R}}$
Left Image
Right Image
(c)
Left Unit Sphere
$\mathbf{n}^{\mathrm{L}}$
$\mathbf{n}^{\mathrm{R}}$
$\mathbf{t}^{\mathrm{L}}$
$\mathbf{u}$
$\mathbf{t}^{\mathrm{R}}$
(d)
Left Unit Sphere
Orthogonal Projection
Riemann Mapping
$\mathbf{t}^{\mathrm{L}}$
$\mathbf{v}$
$\mathbf{t}^{\mathrm{R}}$
$\boldsymbol{\omega}$
$\mathbf{r}^{\mathrm{R}}$
(e)

Figure 2. (a, b) Baseline methods in the image lead to ambiguity and/or non-association problems. (c) Given a 2D-2D line correspondence $(\mathbf{l}^{\mathrm{L}}, \mathbf{l}^{\mathrm{R}})$, we compute the unit projection plane normals $\mathbf{n}^{\mathrm{L}}$ and $\mathbf{n}^{\mathrm{R}}$. (d) We move $\mathbf{n}^{\mathrm{R}}$ to the left sphere and keep its coordinates unchanged. (e) The plane $\boldsymbol{\omega}$ is tangent to sphere at the point $\mathbf{t}^{\mathrm{L}}$. We map the point $\mathbf{t}^{\mathrm{R}}$ into the point $\mathbf{r}^{\mathrm{R}}$ lying on $\boldsymbol{\omega}$, and then connect $\mathbf{t}^{\mathrm{L}}$ and $\mathbf{r}^{\mathrm{R}}$ to define the vector $\mathbf{v}$ tangent to sphere.

and positional variations of image lines. In the following, we first introduce the motivation of our mapping by illustrating the limitations of various baseline methods.

Baseline Methods in Image. As shown in Fig. 2(a), BL-Ang uses a pair of inclinations $(\theta^{\mathrm{L}}, \theta^{\mathrm{R}})$ to encode the angular variation of an inlier line correspondence $(\mathbf{l}^{\mathrm{L}}, \mathbf{l}^{\mathrm{R}})$. However, this inlier and an outlier $(\mathbf{l}^{\mathrm{L}}, \tilde{\mathbf{l}}^{\mathrm{R}})$ provide similar angular variations. Such an ambiguity problem results in some indistinguishable outliers. As shown in Fig. 2(b), BL-Pos uses a pair of midpoints $(\mathbf{m}^{\mathrm{L}}, \mathbf{m}^{\mathrm{R}})$ to encode the positional variation of an inlier line correspondence $(\mathbf{l}^{\mathrm{L}}, \mathbf{l}^{\mathrm{R}})$. However, this inlier and an outlier $(\mathbf{l}^{\mathrm{L}}, \hat{\mathbf{l}}^{\mathrm{R}})$ provide similar positional variations. Moreover, the midpoints $(\mathbf{m}^{\mathrm{L}}, \mathbf{m}^{\mathrm{R}})$ may be non-associated due to incomplete image line detection [36]. Such ambiguity and non-association problems result in some indistinguishable outliers. As shown in Fig. 2(a), BL-Ang-Pos uses two pairs of endpoints $(\mathbf{s}^{\mathrm{L}}, \mathbf{s}^{\mathrm{R}})$ and $(\mathbf{e}^{\mathrm{L}}, \mathbf{e}^{\mathrm{R}})$ to encode both angular and positional variations of an inlier line correspondence $(\mathbf{l}^{\mathrm{L}}, \mathbf{l}^{\mathrm{R}})$. However, a pair of endpoints may be non-associated, which is similar to the above non-associated midpoints. Moreover, compared with our vectors tangent to sphere (introduced below), two pairs of endpoints are redundant and thus increase the difficulty of learning. As will be shown in the experiments, the above baseline methods provide unsatisfactory accuracy.

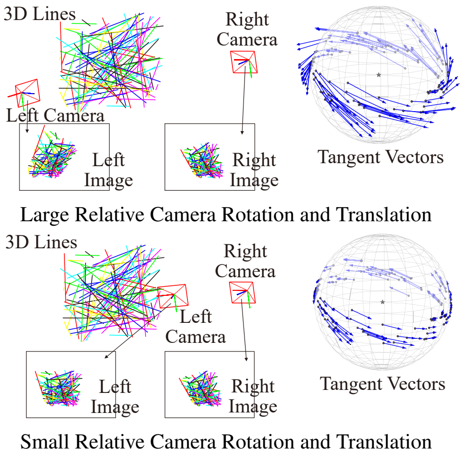


Figure 3. Given a set of 2D-2D line correspondences, we generate vectors tangent to sphere. These vectors exhibit the spatial regularity, regardless of the type and magnitude of camera motion.

Our Method on Sphere. To overcome the limitations of the above baselines, we propose a novel image-to-sphere mapping. Fig. 2(c) shows that a line correspondence $(\mathbf{l}^{\mathrm{L}}, \mathbf{l}^{\mathrm{R}})$ is associated with an unobservable 3D line $\mathbf{L}$. Let us consider the line $\mathbf{l}^{\mathrm{L}}$ in the left image to illustrate the projective geometry. Specifically, $\mathbf{l}^{\mathrm{L}}$ and the left camera center define a projection plane $\boldsymbol{\pi}^{\mathrm{L}}$. We compute it by $\boldsymbol{\pi}^{\mathrm{L}} = \left(\mathbf{K}\ [\mathbf{I} \mid \mathbf{0}]\right)^{\top} \mathbf{l}^{\mathrm{L}}$, where $\mathbf{K}$ denotes the known intrinsic matrix obtained by calibration [12], $\mathbf{I}$ denotes a 3×3 identity matrix, $\mathbf{0}$ denotes a 3×1 zero vector. We compute the unit normal $\mathbf{n}^{\mathrm{L}}$ of the projection plane $\boldsymbol{\pi}^{\mathrm{L}}$. Similarly, we obtain the unit normal $\mathbf{n}^{\mathrm{R}}$ of the projection plane $\boldsymbol{\pi}^{\mathrm{R}}$. The normals $\mathbf{n}^{\mathrm{L}}$ and $\mathbf{n}^{\mathrm{R}}$ are independent of the endpoints of lines $\mathbf{l}^{\mathrm{L}}$ and $\mathbf{l}^{\mathrm{R}}$, respectively. Therefore, we solve the problem that endpoints of lines $\mathbf{l}^{\mathrm{L}}$ and $\mathbf{l}^{\mathrm{R}}$ are non-associated.

As shown in Fig. 2(d), we move the normal $\mathbf{n}^{\mathrm{R}}$ to the left sphere and keep its coordinates unchanged. Accordingly, the terminal points $\mathbf{t}^{\mathrm{L}}$ and $\mathbf{t}^{\mathrm{R}}$ of the normals $\mathbf{n}^{\mathrm{L}}$ and $\mathbf{n}^{\mathrm{R}}$ both lie on the left sphere. Note that we do not treat the vector $\mathbf{u}$ defined by $\mathbf{t}^{\mathrm{L}}$ and $\mathbf{t}^{\mathrm{R}}$ as our displacement vector. The reason is that $\mathbf{u}$ passes through the sphere, and thus exploring its spatial regularity is difficult.[3] Instead, as shown in Fig. 2(e), we define our displacement vector by the vector $\mathbf{v}$ tangent to sphere. Specifically, we treat the point $\mathbf{t}^{\mathrm{L}}$ as the initial point of vector $\mathbf{v}$. We exploit Riemann mapping [8] to project the point $\mathbf{t}^{\mathrm{R}}$ as the point $\mathbf{r}^{\mathrm{R}}$ lying on the tangent plane $\boldsymbol{\omega}$, and treat $\mathbf{r}^{\mathrm{R}}$ as the terminal point of vector $\mathbf{v}$. Note that we do not use orthogonal projection to project the point $\mathbf{t}^{\mathrm{R}}$ to the plane $\boldsymbol{\omega}$. The reason is that orthogonal projection may result in ambiguity, i.e., $\mathbf{t}^{\mathrm{R}}$ and another point on sphere may lead to the same orthogonal projection on the plane $\boldsymbol{\omega}$.

[3] Intuitively, this problem is analogous to some cases that Euclidean distance is less appropriate than geodesic distance on sphere [3, 22].

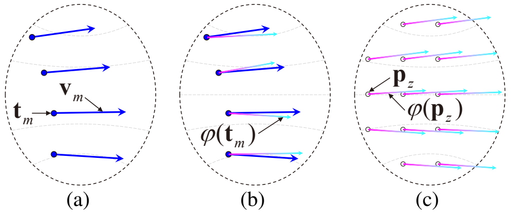


Figure 4. Illustration of local vector field fitting. (a) Outlier-free vectors $\{\mathbf{v}_m\}_{m=1}^M$. (b) We find a local vector field φ that minimizes the difference between the observations $\{\mathbf{v}_m\}_{m=1}^M$ and predictions $\{\varphi(\mathbf{t}_m)\}_{m=1}^M$. (c) We use φ to interpolate the vectors $\{\varphi(\mathbf{p}_z)\}_{z=1}^Z$ starting at some sampled points $\{\mathbf{p}_z\}_{z=1}^Z$.

4. Learning Spatial Regularity on Sphere

Based on the above image-to-sphere mapping, given N putative line correspondences, we generate N tangent vectors. By leveraging the spatial regularity of these vectors, we propose a neural network to predict the inlier probability of each vector. Our main contribution is a novel loss function to learn the spatial regularity.

4.1. Loss Function

Local Trend Consistency (LTC) Loss. As shown in Fig. 1(c), neighboring vectors generated by inliers exhibit the spatial regularity on sphere. We call this regularity LTC. As shown in Fig. 3, we follow [19] to synthesize 3D lines and cameras with various relative rotation and translations. LTC is valid regardless of the type and magnitude of camera motion. In addition, we observe that a scene with significant depth variation may lead to multiple sets of vectors that correspond to different LTCs but overlap with each other, which is similar to the point problem [28]. Therefore, we recommend using our LTC for the images with relatively small depth variations.

a) Basis of Modeling LTC. We introduce the local vector field fitting that is the basis of modeling LTC. In our context, intuitively, a local vector field is composed of numerous vectors starting at different positions in a local area of sphere. Mathematically, we parametrize a local vector field by a vector-valued function whose input is an arbitrary position in a local area and output is a 3D vector [29, 23]. To facilitate understanding, let us consider an outlier-free case to illustrate the local vector field fitting. Fig. 4(a) shows M vectors $\{\mathbf{v}_m\}_{m=1}^M$ starting at the points $\{\mathbf{t}_m\}_{m=1}^M$ in a local area. We use these vectors to fit a local vector field φ. Specifically, as shown in Fig. 4(b), we aim to find the optimal function $\varphi(\cdot)$ that minimizes the difference between the observations $\{\mathbf{v}_m\}_{m=1}^M$ and predictions $\{\varphi(\mathbf{t}_m)\}_{m=1}^M$, i.e.,

$$\min_{\varphi} \sum_{m=1}^{M} p_m \cdot \|\mathbf{v}_m - \varphi(\mathbf{t}_m)\|. \tag{1}$$

where p_m denotes the predicted inlier probability whose ground truth value is 1 in this outlier-free case.

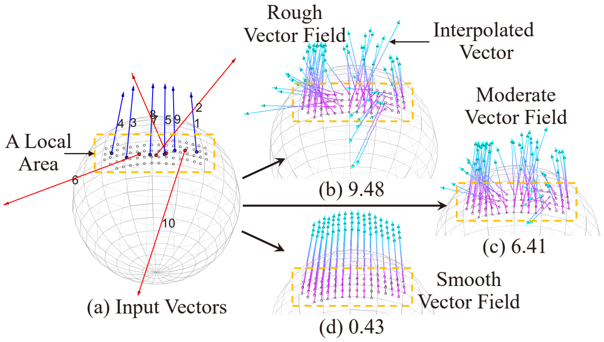


Figure 5. Effectiveness of our LTC loss. (a) Input vectors consist of inliers $\{\mathbf{v}_i\}$ (blue) and outliers $\{\mathbf{v}_j\}$ (red). (b,c,d) We assign these vectors with different inlier probabilities $(p_i, p_j) \in \{(0.5, 0.5), (0.8, 0.2), (0.99, 0.01)\}$ to fit local vector fields. A number below each sphere represents our LTC loss in Eq. (4).

We span the local vector field φ by M unknown-but-sought basis vectors $\{\mathbf{c}_m\}_{m=1}^M$ [29]. Accordingly, we reformulate fitting a local vector field φ as computing its basis vectors $\{\mathbf{c}_m\}_{m=1}^M$. Specifically, we first linearly express the prediction $\varphi(\mathbf{t}_m)$ in Eq. (1) by unknown-but-sought basis vectors $\{\mathbf{c}_m\}_{m=1}^M$. For example, we express $\varphi(\mathbf{t}_1)$ by

$$\varphi(\mathbf{t}_1) = \sum_{m=1}^{M} e_{1\leftarrow m} \cdot \mathbf{c}_m. \tag{2}$$

The coefficient $e_{1\leftarrow m}$ encodes the effect of the basis vector $\mathbf{c}_m$. Intuitively, $e_{1\leftarrow m}$ is inversely proportional to the distance from $\mathbf{t}_m$ to $\mathbf{t}_1$. To define this coefficient, we use Gaussian kernel [39]. Then we substitute the above linear expressions of $\{\varphi(\mathbf{t}_m)\}_{m=1}^M$ (e.g., Eq. (2)) into Eq. (1), obtaining a linear system to compute the basis vectors $\{\mathbf{c}_m\}_{m=1}^M$ [29]:

$$(\mathbf{P}^{-1} + \mathbf{E})\underbrace{[\mathbf{c}_1, \mathbf{c}_2, \cdots \mathbf{c}_M]^\top}_{\mathbf{C}\,\in\,\mathbb{R}^{M\times 3}} = [\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_M]^\top, \tag{3}$$

where $\mathbf{P}$ is an $M \times M$ diagonal matrix composed of the inlier probabilities $\{p_m\}_{m=1}^M$ in Eq. (1), and $\mathbf{E}$ is an $M \times M$ matrix composed of the coefficients in Eq. (2) (for example, its first row is $[e_{1\leftarrow 1}, e_{1\leftarrow 2}, \cdots e_{1\leftarrow M}]$). In the following, we use the computed basis vectors $\{\mathbf{c}_m\}_{m=1}^M$ to model LTC and define our LTC loss.

b) Modeling LTC and Defining LTC Loss. Intuitively, a high-level LTC of vectors corresponds to a high-level "smoothness" of the local vector field φ. Accordingly, we model LTC by this smoothness that can be evaluated by the norm in reproducing kernel Hilbert space $\mathcal{H}$ [29] as

$$\|\varphi\|_{\mathcal{H}} = \frac{1}{M} \sum_{w=1}^{3} \mathbf{C}_w{}^\top \mathbf{E} \mathbf{C}_w \rightarrow L^{\mathrm{LTC}}, \tag{4}$$

where $\mathbf{C}_w$ denotes the w-th column of the matrix $\mathbf{C}$ in Eq. (3), and $\mathbf{E}$ is defined in Eq. (3). The smaller norm $\|\varphi\|_{\mathcal{H}}$ is, the higher-level LTC is. Therefore, we treat $\|\varphi\|_{\mathcal{H}}$ as our

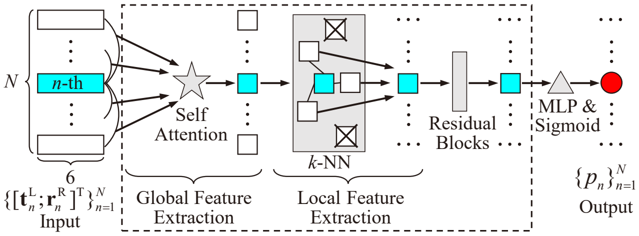

N
n-th
6
$\{[\mathbf{t}_n^{\mathrm{L}};\mathbf{r}_n^{\mathrm{R}}]^{\mathrm{T}}\}_{n=1}^{N}$
Input
Self Attention
Global Feature Extraction
k-NN
Local Feature Extraction
Residual Blocks
MLP & Sigmoid
$\{p_n\}_{n=1}^{N}$
Output

Figure 6. Pipeline of our network. Each input data is defined by a vector tangent to sphere. We take the n-th data for example to illustrate how our network predicts the inlier probability p_n.

LTC loss. In addition to quantitative evaluation of our LTC loss, we provide a visual evaluation. Specifically, as shown in Fig. 4(c), we uniformly sample some positions $\{\mathbf{p}_z\}_{z=1}^{Z}$ in the local area. At each position $\mathbf{p}_z$, we use the computed basis vectors $\{\mathbf{c}_m\}_{m=1}^{M}$ to interpolate a vector $\varphi(\mathbf{p}_z)$ (similar to Eq. (2)). The more regular vectors $\{\varphi(\mathbf{p}_z)\}_{z=1}^{Z}$ are, the smaller our LTC loss is.

c) An Example. As shown in Fig. 5(a), let us consider 10 vectors in a local area to illustrate the effectiveness of our LTC loss. The vectors $\{\mathbf{v}_i\}$ ($i \in \{1,3,4,5,7,9\}$) are inliers, and $\{\mathbf{v}_j\}$ ($j \in \{2,6,8,10\}$) are outliers. We vary their inlier probabilities p_i and p_j within $[0,1]$ to mimic different values predicted by our network. We use these vectors and probabilities to compute our LTC loss by Eqs. (3) and (4). We also visually evaluate our LTC loss by interpolation (similar to Fig. 4(c)). We report some representative results as follows. In Fig. 5(b), $(p_i, p_j) = (0.5, 0.5)$, i.e., predictions significantly deviate from ground truth values $(1, 0)$. Accordingly, our LTC loss is high. In Fig. 5(c), $(p_i, p_j) = (0.8, 0.2)$, i.e., predictions approach ground truth values. Accordingly, our LTC loss decreases. In Fig. 5(d), $(p_i, p_j) = (0.99, 0.01)$, i.e., predictions are nearly equal to ground truth values. Accordingly, our LTC loss is low.

Binary Cross Entropy (BCE) Loss. The purpose of using BCE loss [5] is to guarantee satisfactory and consistent accuracy (that may not be high enough). Specifically, a small number of vectors may not be clustered into any local area, and thus are not constrained by our LTC loss. In contrast, BCE loss enforces the constraint on each vector $\mathbf{v}_n$.

Total Loss. Our total loss is the combination of the above LTC and BCE losses. Given N vectors, we employ robust spectral clustering [25] to group their initial points into S local areas (other algorithms can be used instead). A set of points in the same cluster constitute a local area. For each area, we compute our LTC loss based on Eq. (4). Moreover, for each of N vectors, we compute BCE loss L_n^{BCE}. Our total loss is given by

$$L^{\text{total}} = \frac{1}{S}\sum_{s=1}^{S} L_s^{\mathrm{LTC}} + \lambda \cdot \frac{1}{N}\sum_{n=1}^{N} L_n^{\mathrm{BCE}}, \tag{5}$$

where λ controls the trade-off between two losses. Collaboration between LTC and BCE losses is analogous to data and smooth terms in well-known graph cut [6]. Specifically, using only BCE loss may lead to a relatively rough vector field due to over-fitting. LTC loss can enforce smoothness constraint to correct errors.

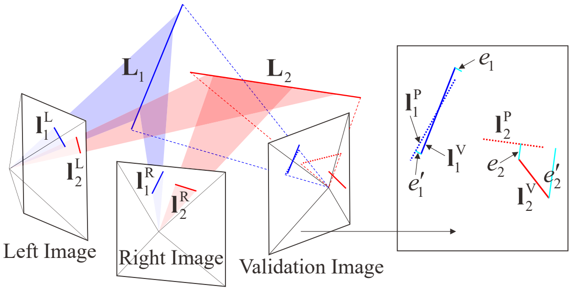

$\mathbf{L}_1$
$\mathbf{L}_2$
$\mathbf{l}_1^{\mathrm{L}}$
$\mathbf{l}_2^{\mathrm{L}}$
$\mathbf{l}_1^{\mathrm{R}}$
$\mathbf{l}_2^{\mathrm{R}}$
Left Image
Right Image
Validation Image
e_1
$\mathbf{l}_1^{\mathrm{P}}$
$\mathbf{l}_1^{\mathrm{V}}$
e_1'
$\mathbf{l}_2^{\mathrm{P}}$
e_2
e_2'
$\mathbf{l}_2^{\mathrm{V}}$

Figure 7. Illustrating how we automatically label a 2D-2D line correspondence $(\mathbf{l}_n^{\mathrm{L}}, \mathbf{l}_n^{\mathrm{R}})$ as an inlier (blue) or outlier (red) for our LRW dataset establishment.

4.2. Network

As shown in Fig. 6, we express each tangent vector $\mathbf{v}_n$ by concatenating its 3×1 initial and terminal points (see Fig. 2(e)) vertically, i.e., $\mathbf{v}_n \leftarrow [\mathbf{t}_n^{\mathrm{L}}; \mathbf{r}_n^{\mathrm{R}}]$. Accordingly, N vectors $\{\mathbf{v}_n^{\top}\}_{n=1}^{N}$ constitute an $N \times 6$ input matrix. Inspired by [42], we first extract the *global* feature for each vector $\mathbf{v}_n$. Specifically, we compute the pairwise similarities between $\mathbf{v}_n$ and all the other vectors by a self-attention module [35]. We integrate this similarity information into $\mathbf{v}_n$ to enrich the expression of $\mathbf{v}_n$. Then we extract the *local* feature for $\mathbf{v}_n$. Specifically, we exploit k-nearest neighbor (k-NN) search [5] to retrieve k neighboring vectors of $\mathbf{v}_n$. We aggregate the enriched expressions of these k vectors to obtain a local feature of $\mathbf{v}_n$. We further extract features by a series of residual blocks [13]. Finally, we process the above features by a multi-layer perceptron (MLP) and a Sigmoid activation, obtaining N numbers within $[0,1]$. The n-th number corresponds to the predicted inlier probability p_n of the vector $\mathbf{v}_n$. If $p_n \geqslant 0.5$, we treat $\mathbf{v}_n$ as an inlier. Otherwise, we treat $\mathbf{v}_n$ as an outlier.

5. Our Dataset

Existing datasets for image line matching are either small [24, 32] or only provide synthetic images [17]. In contrast, we establish a large real-world (LRW) dataset composed of 11,934 pairs of images. We obtain these images from 8 sequences of TUM dataset [33]. We associate each image pair with a set of putative line correspondences whose ground truth labels are known. In our experiments, we select 80% and 20% of image pairs for training and testing, respectively.

We provide a tool that can automatically generate the putative 2D-2D line correspondences and label them. Specifically, on each sequence, we continuously sample three consecutive images using different sampling resolutions. The

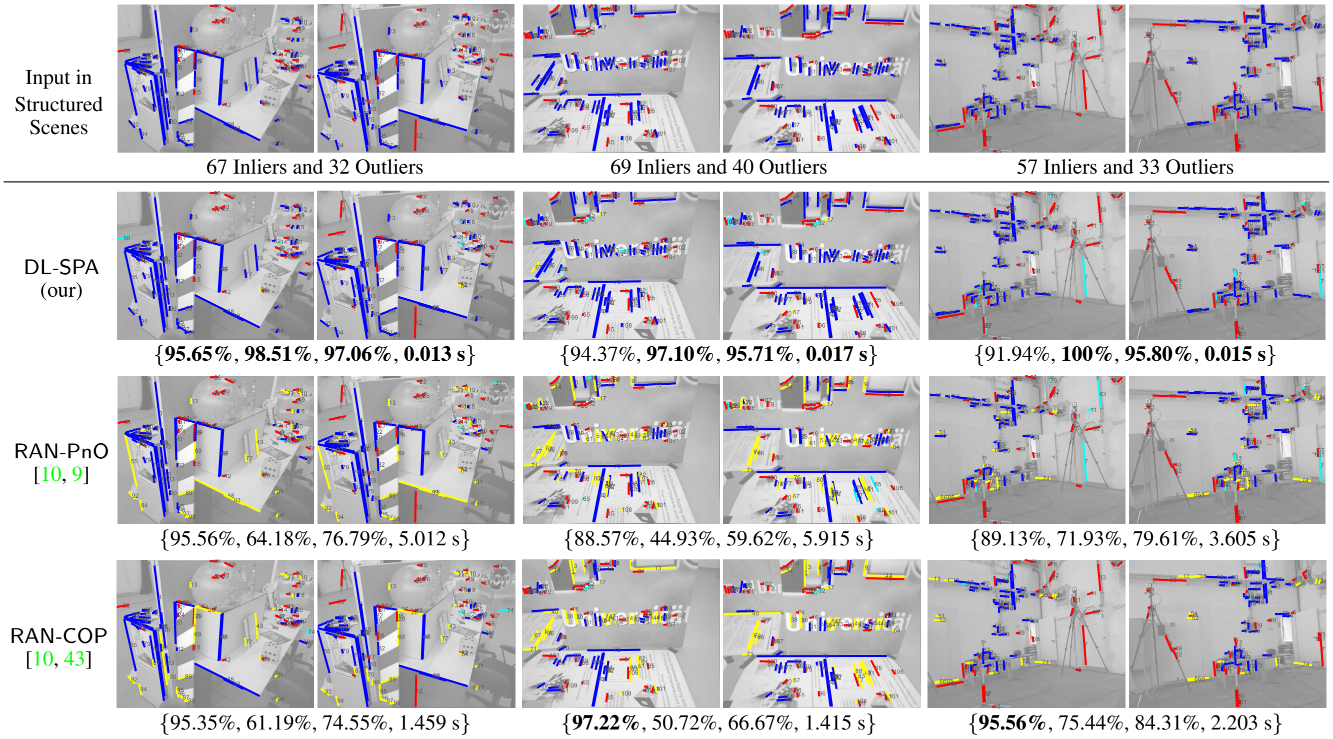

Input in Structured Scenes
67 Inliers and 32 Outliers
69 Inliers and 40 Outliers
57 Inliers and 33 Outliers
DL-SPA (our)
{**95.65%**, **98.51%**, **97.06%**, **0.013 s**}
{94.37%, **97.10%**, **95.71%**, **0.017 s**}
{91.94%, **100%**, **95.80%**, **0.015 s**}
RAN-PnO [10, 9]
{95.56%, 64.18%, 76.79%, 5.012 s}
{88.57%, 44.93%, 59.62%, 5.915 s}
{89.13%, 71.93%, 79.61%, 3.605 s}
RAN-COP [10, 43]
{95.35%, 61.19%, 74.55%, 1.459 s}
{**97.22%**, 50.72%, 66.67%, 1.415 s}
{**95.56%**, 75.44%, 84.31%, 2.203 s}

Figure 8. Accuracy and efficiency comparisons on representative image pairs of our LRW dataset. The first row shows putative line correspondences. Ground truth inliers and outliers are shown in blue and red, respectively. The second, third and fourth rows present the inlier identification results. True positive and true negative are shown in blue and red, respectively; False positive and false negative are shown in cyan and yellow, respectively. A quadruplet of numbers below each image pair represents {precision, recall, F_1-score, runtime}.

purpose of sampling three (instead of two) images is to solve the problem that 2D-2D line correspondences are not geometrically constrained. Fig. 7 shows a sampled image triplet. We extract image lines by LSD [36] and match these lines across three images by LBD descriptor [41], generating putative 2D-2D-2D correspondences $\{(\mathbf{l}_n^{\mathrm{L}}, \mathbf{l}_n^{\mathrm{R}}, \mathbf{l}_n^{\mathrm{V}})\}_{n=1}^{N}$. We use each 2D-2D correspondence $(\mathbf{l}_n^{\mathrm{L}}, \mathbf{l}_n^{\mathrm{R}})$ to triangulate a 3D line $\mathbf{L}_n$, and then project $\mathbf{L}_n$ to the validation image as $\mathbf{l}_n^{\mathrm{P}}$. We compute the reprojection errors [2], i.e., the orthogonal distances e_n and e'_n between the observation $\mathbf{l}_n^{\mathrm{V}}$ and the endpoints of projection $\mathbf{l}_n^{\mathrm{P}}$. If $(e_n + e'_n)/2$ is smaller than 5 pixels, we label the correspondence $(\mathbf{l}_n^{\mathrm{L}}, \mathbf{l}_n^{\mathrm{R}})$ as an inlier. Otherwise, we label $(\mathbf{l}_n^{\mathrm{L}}, \mathbf{l}_n^{\mathrm{R}})$ as an outlier.

6. Experiments

We denote our deep learning-based approach that leverages the spatial regularity on sphere by DL-SPA. We compare our DL-SPA with state-of-the-art methods in Section 6.1. We conduct ablation studies of our image-to-sphere mapping and loss function in Section 6.2. We evaluate the generalization of our DL-SPA in Section 6.3.

Implementation Details. We set the parameter λ in Eq. (5) as 2, and the parameter k of k-NN search as 8. We use Adam [16] to train our network. The learning rate is 10^{-3}, batch size is 32, and number of epochs is 30.

Evaluation Criteria. We call a correctly identified inlier "true positive (TP)", and wrongly identified inlier "false positive (FP)". We call a correctly identified outlier "true negative (TN)", and wrongly identified outlier "false negative (FN)". We follow [42] to evaluate the algorithm accuracy by precision, recall and F_1-score. Specifically, precision $= \frac{\delta(\mathrm{TP})}{\delta(\mathrm{TP})+\delta(\mathrm{FP})}$ and recall $= \frac{\delta(\mathrm{TP})}{\delta(\mathrm{TP})+\delta(\mathrm{FN})}$ where $\delta(\cdot)$ denotes the cardinality. F_1-score $= \frac{2\cdot\mathrm{precision}\cdot\mathrm{recall}}{\mathrm{precision}+\mathrm{recall}}$.

6.1. Comparison with State-of-the-art Methods

As introduced in Section 2, existing methods to identify inliers of 2D-2D line correspondences exploit the geometric constraints. We compare our DL-SPA with the state-of-the-art ones:

- Integrating RANSAC [10] with the camera pose estimation method that requires a "2D-2D line correspondence triplet" [9]. This triplet is projected from a 3D line triplet whose two lines are mutually parallel and orthogonal to the third. We denote this integration by RAN-PnO.
- Integrating RANSAC [10] with the camera pose estimation method that requires a "2D-2D line correspondence pair" [43]. This pair is projected from two coplanar 3D lines. We denote this integration by RAN-COP.

Table 1. Accuracy and efficiency comparisons on five sequences (SQ) of our LRW dataset. We present the mean of each metric.

	Outlier Ratio	DL-SPA (our)				RAN-PnO [10, 9]				RAN-COP [10, 43]			
		Precision	Recall	F_1-score	Runtime	Precision	Recall	F_1-score	Runtime	Precision	Recall	F_1-score	Runtime
SQ1	31.73%	**94.53%**	**97.54%**	**95.96%**	**0.013 s**	91.42%	71.16%	79.23%	4.847 s	93.15%	59.51%	71.86%	1.549 s
SQ2	32.91%	**93.19%**	**97.68%**	**95.33%**	**0.016 s**	83.68%	66.54%	74.13%	9.340 s	92.51%	68.73%	78.12%	2.546 s
SQ3	36.61%	91.62%	**98.30%**	**94.76%**	**0.014 s**	90.29%	60.87%	71.04%	5.213 s	**95.20%**	59.80%	72.38%	1.475 s
SQ4	25.98%	91.63%	**98.72%**	**94.95%**	**0.011 s**	82.85%	65.60%	73.16%	6.457 s	**95.60%**	67.18%	78.42%	1.854 s
SQ5	29.48%	**93.57%**	**98.00%**	**95.64%**	**0.011 s**	82.58%	63.94%	71.42%	6.753 s	91.19%	73.98%	81.03%	1.610 s

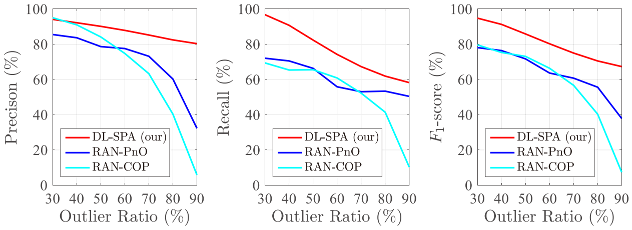


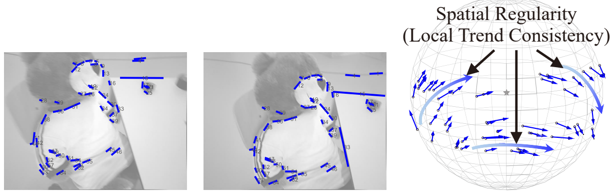
Figure 9. Robustness comparisons with respect to outlier ratio on five sequences of our LRW dataset. We present the evolutions of average precision, recall and F_1-score.

Note that RAN-PnO and RAN-COP are only applicable to structured scenes. For a fair comparison, from our LRW dataset, we select five sequences obtained in structured scenes. Fig. 8 and Table 1 present the comparisons in terms of accuracy and efficiency.

Accuracy. Our DL-SPA achieves the highest F_1-score, demonstrating that learning the spatial regularity on sphere is effective. In contrast, RAN-PnO and RAN-COP lead to unsatisfactory F_1-scores that are mainly affected by recalls. The reason is that sampling a "valid" 2D-2D line correspondence triplet or pair is relatively difficult. Specifically, correspondences of a valid triplet or pair should not only be inliers, but also exactly satisfy the above structure assumption. For example, RAN-PnO samples three inlier correspondences that are not associated with orthogonal or parallel 3D lines, i.e., an invalid triplet. These correspondences result in wrong relative pose estimation, which affects the precision. Moreover, they cannot be fitted by the correct pose estimated by a valid triplet, which affects the recall.

Efficiency. Our DL-SPA achieves the highest efficiency, demonstrating the superiority of deep learning in this task. In contrast, RAN-PnO and RAN-COP are time-consuming for two main reasons. First, they quasi-exhaustively use three or two line correspondences to generate numerous candidate correspondence triplets or pairs. Second, as mentioned above, sampling a valid correspondence triplet or pair is relatively difficult. Accordingly, these methods require a large number of iterations (higher than 1000 in general) based on the probabilistic guarantee [12].

Robustness. We train our DL-SPA on the original LRW dataset whose average outlier ratio is 31.34%. We vary the outlier ratio from 30% to 90% by perturbing inlier correspondences. We test all the methods under different outlier ratios. As shown in Fig. 9, our DL-SPA is relatively robust (especially in terms of precision) since inlier vectors always exhibit LTC regardless of the outlier ratio. In contrast, the accuracies of RAN-PnO and RAN-COP drastically decrease as the outlier ratio increases. The reason is that a high outlier ratio is prone to resulting in invalid sampling.



(a) 48 Inliers in an Unstructured Scene (b) 3D Tangent Vectors

Figure 10. (a) Ground truth inlier line correspondences on a representative image pair of our LRW dataset (we do not show outliers on purpose). (b) Based on these inlier correspondences, we generate 3D tangent vectors by our image-to-sphere mapping.

Table 2. Comparing our DL-SPA with various baseline methods on all the test data of our LRW dataset.

	F_1-score
BL-Ang	82.73%
BL-Pos	84.45%
BL-Ang-Pos	88.80%
DL-SPA (our)	**93.71%**

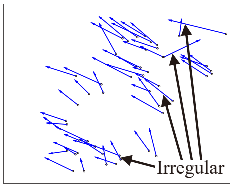
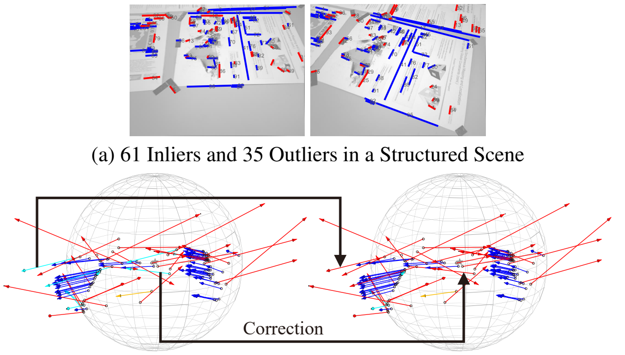


Figure 11. 2D displacement vectors generated by connecting the midpoints of image line correspondences in Fig. 10(a).

6.2. Ablation Studies

Image-to-sphere Mapping. As introduced in Section 3, we map line correspondences into vectors tangent to sphere. Fig. 10 shows a representative test. Tangent vectors generated by inliers all exhibit LTC. In the following, we compare our DL-SPA leveraging the LTC of tangent vectors with various baseline methods in the image (see Section 3). Except for the input, the network architectures of various baselines are the same as the architecture of DL-SPA (see Fig. 6). We use the same image pairs of our LRW dataset to train and test various baselines and our DL-SPA. Accordingly, the inputs of BL-Ang, BL-Pos, BL-Ang-Pos and our DL-SPA are $N \times 2$, $N \times 4$, $N \times 8$ and $N \times 6$ matrices, respectively. Table 2 shows that our DL-SPA is more accurate than various baselines. The reason is that our 3D tangent vectors are



(a) 61 Inliers and 35 Outliers in a Structured Scene

{87.14%, 98.39%, 92.42%}
(b) BCE Loss

{95.31%, 98.39%, 96.83%}
(c) BCE & LTC Losses

Figure 12. (a) Putative line correspondences on a representative image pair of our LRW dataset. Ground truth inliers and outliers are shown in blue and red, respectively. (b,c) We compare our DL-SPA trained by only BCE loss, and the combination of BCE and LTC losses. True positive and true negative are shown in blue and red, respectively; False positive and false negative are shown in cyan and yellow, respectively. A triplet of numbers below each sphere represents {precision, recall, F_1-score}.

Table 3. Comparing our DL-SPA trained by only BCE loss, and the combination of BCE and LTC losses on all the test data of our LRW dataset. We present the average F_1-score.

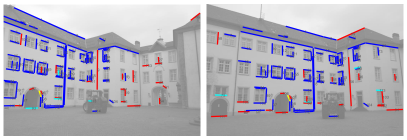
	BCE Loss	BCE & LTC Losses
SQ1	93.64%	**95.96%** (↑ 2.32%)
SQ2	94.25%	**95.33%** (↑ 1.08%)
SQ3	92.23%	**94.76%** (↑ 2.53%)
SQ4	92.84%	**94.95%** (↑ 2.11%)
SQ5	94.82%	**95.64%** (↑ 0.82%)
SQ6	94.01%	**95.92%** (↑ 1.91%)
SQ7	94.09%	**95.34%** (↑ 1.25%)
SQ8	93.87%	**94.76%** (↑ 0.89%)

regular and also solve the ambiguity, non-association and redundancy problems (see Fig. 11).

Loss Function. Recall that our total loss in Eq. (5) is the combination of BCE and LTC losses. On our LRW dataset, we train our DL-SPA by only BCE loss, and the combination of BCE and LTC losses, respectively. As mentioned in Section 4.1, we do not train our DL-SPA by only LTC loss. Fig. 12 and Table 3 show that our DL-SPA trained by only BCE loss provides relatively high accuracy since BCE loss is suitable for inlier/outlier identification, a binary classification problem. Our DL-SPA trained by the combination of BCE and LTC losses improves the accuracy. The reason is that our LTC loss enforces the smoothness constraint of vector field and thus corrects some false positives.

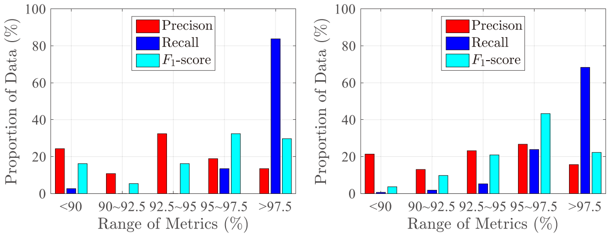
6.3. Generalization Evaluation

Recall that we train our DL-SPA on our LRW dataset. In this section, we test its generalization on a new dataset, i.e., EPFL dataset [32]. Ground truth inlier and outlier line correspondences of EPFL dataset are provided by Line3D++ [14]. As shown in Figs. 13 and 14, the accuracy of our DL-SPA on EPFL dataset is similar to that on our LRW dataset for two main reasons. First, the spatial regularity on sphere is independent of image appearance. Instead, it depends on relative motions between two images. On our LRW dataset, we use various motions to generate the training data. These motions nearly cover the motions on EPFL dataset. Second, the spatial regularity on sphere is independent of image resolution. Specifically, the length of an image line does not affect the vector generated by our image-to-sphere mapping. Therefore, while images of EPFL and our LRW datasets have different appearances and resolutions, our DL-SPA can handle both datasets well.



{95.60%, 98.86%, 97.21%, 0.014s}

Figure 13. Evaluating the generalization of our DL-SPA on a representative image pair of EPFL dataset [32]. Denotations are the same as those in Fig. 8.



{91.89%, 97.88%, 94.68%}
(a) EPFL Dataset

{93.19%, 98.03%, 95.46%}
(b) Our LRW Dataset

Figure 14. Evaluating the generalization of our DL-SPA by comparing the results on all the test data of (a) EPFL dataset [32] and (b) our LRW dataset. A triplet of numbers below each image represents the means of {precision, recall, F_1-score}.

7. Conclusions

We propose a novel approach to identify inliers of putative 2D-2D line correspondences. Our approach is the first one suitable for both structured and unstructured scenes. To achieve this goal, we leverage the spatial regularity on sphere. We propose a novel image-to-sphere mapping to generate vectors tangent to sphere. Moreover, we propose a novel loss function to learn LTC of vectors. In addition, we establish a large real-world dataset for image line matching. Experiments showed that our approach outperforms state-of-the-art ones in terms of accuracy, efficiency and robustness, and also leads to high generalization.

Acknowledgments. This work is supported in part by the Hong Kong Centre for Logistics Robotics, RGC via grant 14207119, the CUHK VC Discretionary Fund, and Shenzhen Municipal Government via the Shenzhen-HK Collaboration Zone Project.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven Seitz, and Richard Szeliski. Building Rome in a day. In *ICCV*, 2009. 1

[2] Adrien Bartoli and Peter Sturm. The 3D line motion matrix and alignment of line reconstructions. *IJCV*, 2004. 6

[3] Jean-Charles Bazin, Cedric Demonceaux, Pascal Vasseur, and Inso Kweon. Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. *IJRR*, 2012. 3

[4] Alexander Berg, Tamara Berg, and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005. 1

[5] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 5

[6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 2001. 5

[7] Nam-Gyu Cho, Alan Yuille, and Seong-Whan Lee. A novel linelet-based representation for line segment detection. *TPAMI*, 2018. 2

[8] Manfredo Perdigao do Carmo. *Riemannian geometry*. Birkhäuser, 1992. 3

[9] Ali Elqursh and Ahmed Elgammal. Line-based relative pose estimation. In *CVPR*, 2011. 1, 2, 6, 7

[10] Martin Fischler and Robert Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 2, 6, 7

[11] Richard Hartley. Lines and points in three views and the trifocal tensor. *IJCV*, 1997. 2

[12] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003. 1, 3, 7

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[14] Manuel Hofer, Michael Maurer, and Horst Bischof. Efficient 3D scene abstraction using line segments. *CVIU*, 2017. 8

[15] Qi Jia, Xinkai Gao, Xin Fan, Zhongxuan Luo, Haojie Li, and Ziyao Chen. Novel coplanar line-points invariants for robust line matching across views. In *ECCV*, 2016. 1

[16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[17] Manuel Lange, Fabian Schweinfurth, and Andreas Schilling. DLD: A deep learning based line descriptor for line feature matching. In *IROS*, 2019. 2, 5

[18] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *CVPR*, 2017. 2

[19] Haoang Li, Jian Yao, Xiaohu Lu, and Junlin Wu. Combining points and lines for camera pose estimation and optimization in monocular visual odometry. In *IROS*, 2017. 4

[20] Haoang Li, Ji Zhao, Jean-Charles Bazin, Wen Chen, Kai Chen, and Yun-Hui Liu. Line-based absolute and relative camera pose estimation in structured environments. In *IROS*, 2019. 2

[21] Haoang Li, Ji Zhao, Jean-Charles Bazin, and Yun-Hui Liu. Quasi-globally optimal and near/true real-time vanishing point estimation in Manhattan world. *TPAMI*, 2020. 1

[22] Haoang Li, Ji Zhao, Jean-Charles Bazin, and Yun-Hui Liu. Robust estimation of absolute camera pose via intersection constraint and flow consensus. *TIP*, 2020. 3

[23] Haoang Li, Ji Zhao, Jean-Charles Bazin, Lei Luo, Junlin Wu, and Jian Yao. Robust camera pose estimation via consensus on ray bundle and vector field. In *IROS*, 2018. 4

[24] Kai Li, Jian Yao, Mengsheng Lu, Heng Yuan, Teng Wu, and Yinxuan Li. Line segment matching: A benchmark. In *WACV*, 2016. 2, 5

[25] Zhenguo Li, Jianzhuang Liu, Shifeng Chen, and Xiaoou Tang. Noise robust spectral clustering. In *ICCV*, 2007. 5

[26] David Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 1

[27] Bruce Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981. 1

[28] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *TIP*, 2014. 4

[29] Charles Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Computation*, 2005. 2, 4

[30] Branislav Micusik and Horst Wildenauer. Structure from motion with line segments under relaxed endpoint constraints. *IJCV*, 2016. 2

[31] David Nister. An efficient solution to the five-point relative pose problem. *TPAMI*, 2004. 1

[32] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008. 2, 5, 8

[33] Jurgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012. 5

[34] William Thompson, Pamela Lechleider, and Elizabeth Stuck. Detecting moving objects using the rigidity constraint. *TPAMI*, 1993. 1

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[36] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *TPAMI*, 2010. 2, 3, 6

[37] Nan Xue, Song Bai, Fu-Dong Wang, Gui-Song Xia, Tianfu Wu, Liangpei Zhang, and Philip Torr. Learning regional attraction for line segment detection. *TPAMI*, 2019. 2

[38] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018. 1

[39] Lyubomir Zagorchev and Ardeshir Goshtasby. A comparative study of transformation functions for nonrigid image registration. *TIP*, 2006. 4

[40] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a Non-Manhattan world. In *CVPR*, 2016. 1

[41] Lilian Zhang and Reinhard Koch. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *JVCIR*, 2013. 1, 2, 6

[42] Chen Zhao, Zhiguo Cao, Chi Li, Xin Li, and Jiaqi Yang. NM-Net: Mining reliable neighbors for robust feature correspondences. In *CVPR*, 2019. 1, 5, 6

[43] Ji Zhao, Laurent Kneip, Yijia He, and Jiayi Ma. Minimal case relative pose computation using ray-point-ray features. *TPAMI*, 2019. 1, 2, 6, 7

[44] Ji Zhao, Jiayi Ma, Jinwen Tian, Jie Ma, and Dazhi Zhang. A robust method for vector field learning with application to mismatch removing. In *CVPR*, 2011. 1