IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Line as a Visual Sentence: Context-aware Line Descriptor for Visual Localization

Sungho Yoon[1] and Ayoung Kim[2*]

***Abstract*—Along with feature points for image matching, line features provide additional constraints to solve visual geometric problems in robotics and computer vision (CV). Although recent convolutional neural network (CNN)-based line descriptors are promising for viewpoint changes or dynamic environments, we claim that the CNN architecture has innate disadvantages to abstract variable line length into the fixed-dimensional descriptor. In this paper, we effectively introduce Line-Transformers dealing with variable lines. Inspired by natural language processing (NLP) tasks where sentences can be understood and abstracted well in neural nets, we view a line segment as a sentence that contains points (words). By attending to well-describable points on a line dynamically, our descriptor performs excellently on variable line length. We also propose line signature networks sharing the line's geometric attributes to neighborhoods. Performing as group descriptors, the networks enhance line descriptors by understanding lines' relative geometries. Finally, we present the proposed line descriptor and matching in a Point and Line Localization (PL-Loc). We show that the visual localization with feature points can be improved using our line features. We validate the proposed method for homography estimation and visual localization.**

***Index Terms*—Localization, SLAM, Deep Learning for Visual Perception.**

I. Introduction

VISUAL features are widely utilized in many robotics and computer vision (CV) applications such as visual tracking, simultaneous localization and mapping (SLAM), and structure-from-motion (SFM). While the keypoint features are well-studied and a base of many applications, keypoints that are not well distributed in the image may result in unstable and inaccurate pose estimation.

Recent research has reported that SLAM performance can be enhanced by using both points and lines [1, 2, 3, 4] even for low-textured environments. For example, line band descriptor (LBD) [5] is one of the widely-used line descriptors in SLAM. Reliable performance of LBD has been reported for continuous frames, whereas the performance degrades under a wide baseline, which prevented the line-based approach from adapting line features directly in visual localization [6, 7]. Tackling this limitation, approaches leveraged convolutional neural network (CNN) to learn the representation of line descriptors [8, 9, 10] yielding superior performance. Still, CNN has innate difficulties in dealing with variable line lengths because the dimension of CNN should be predefined by its architecture. Another issue of line occlusion is also related to this length variation and was examined in SOLD^2 [11].

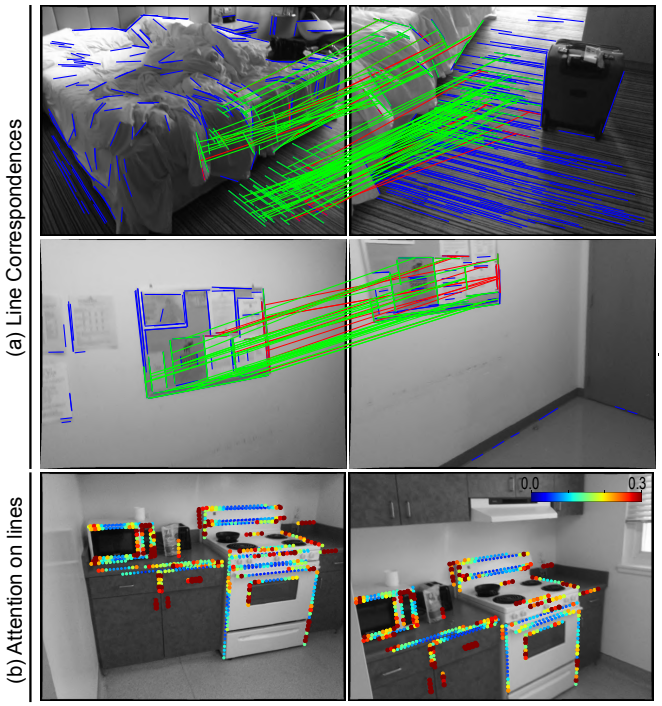


Fig. 1: (a) For both low and high-textured environments, our line descriptor performs accurate matching. (b) Attention scores for points in a line with matched lines having similar attentional contexts.

In our work, we overcome this fixed length requirement and achieve flexibility to the length variation by interpreting a line segment from the natural language processing (NLP) point of view. We view a line consisting of points as a sentence consisting of words, thereby allowing various lengths for lines. The word2vec in NLP learns vector representations that encapsulate each distinct word, leveraging them as base inputs of RNN, LSTM, or transformers for later tasks such as text classification and text summarization. Similarly, we utilize a learned descriptor map as a transition from RGB pixels to point vectors.

Based on this key idea, we propose a novel line descriptor using transformers [12]. The transformers in our model learn the context of the point vectors within a line segment to summarize it in the form of a line descriptor. Unlike SOLD^2, a line is not just a simple sequence of points but is handled with attention to its context dynamically, enabling reliable performance even under occlusion. The attributes of the proposed

Manuscript received: April 29, 2021; Revised August 2, 2021; Accepted September 7, 2021. This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was fully supported by [Localization in changing city] project funded by Naver Labs Corporation.

[1]S. Yoon is with the Robotics Program, KAIST, Daejeon, S. Korea sungho.yoon@kaist.ac.kr

[2]A. Kim is with the Department of Mechanical Engineering, SNU, Seoul, S. Korea ayoungk@snu.ac.kr

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

method can be summarized below.

- We present a novel line segment descriptor using a transformer architecture by considering line segments as sentences and points as words. Leveraging NLP for line descriptor, we can handle lines with various lengths.
- The proposed line descriptor understands the line segment's context by paying attention to more meaningful points on a line. It effectively abstracts various length lines to a fixed-sized descriptor.
- We suggest line signature networks that share the message of line attributes (e.g., position, angle, length, and descriptors) between neighborhoods. Serving as a grouped line descriptor, the line descriptors can learn geometric attributes of their neighborhoods.

II. Related Work

A. Line Descriptors

As a handcrafted line descriptor, Wang et al. [13] proposed a mean-standard deviation line descriptor (MSLD) observing gradients around line segments. LBD [5] enhanced the precision and computing time by investigating gradients on bands that are parallel to a line. Recently, Vakhitov and Lempitsky [8] presented a learnable line descriptor (LLD) for visual SLAM in a deep-learning manner. They first built a full-sized descriptor map using CNN and uniformly split a line segment into a fixed-number of subsegments. Lange et al. proposed a learning-based line descriptor named DLD [9]. They also split a line segment and trained the CNN with a self-generated dataset. Extended works from the same group [10] adapted wavelets to extract more meaningful information from an image, referred to as the Wavelet-based line descriptor (WLD). Pautrat et al. [11] introduced SOLD^2 for joint line detection and description. To find a line correspondence, SOLD^2 sampled uniformly-spaced point descriptors on a line and performed the sequence matching.

B. Point Descriptors

The recent point descriptor also focused on learning-based methods. Like a handcrafted feature SIFT [14], learned descriptors often utilized the patch as their inputs as in L2-Net [15] and HardNet [16]. Examining keypoint detection and description together, LIFT [17] introduced an end-to-end learning pipeline implementing keypoint detection, orientation estimation, and description simultaneously. SuperPoint [18] proposed self-supervised learning for detector and descriptor using synthetic pretraining and homography adaptation.

C. Point-and-Line SLAM

Exploiting both point and line for SLAM has strong advantages, robustly performing in challenging environments such as low-textured, motion blurred, and defocused cases [1, 2, 3, 4]. PL-SVO [1] extended the SVO method by adapting its semi-direct approach to line segments. PL-SLAM [2] presented line features on the monocular ORB-SLAM. In [3], stereo camera-based PL-SLAM was presented to improve the bag-of-words method using points and lines.

In this paper, we utilize visual localization to evaluate line descriptors. Visual localization [6, 7] is a problem of estimating a camera pose given a query image and a prior 3D feature map, which is similar to relocalization in SLAM. It differs from the previous works in considering only discrete scenes, including large viewpoint changes, and it is effective for evaluating the robustness of line descriptors in various situations.

D. Transformers in NLP and CV

We briefly summarize recent studies connecting NLP and CV. Vaswani et al. proposed the transformer for language translation [12], and it became a base architecture of many state-of-the-art methods in NLP tasks. Bidirectional Encoder Representations from Transformers (BERT) is one of the widely used models utilizing transformer encoders [19]. One of the recent trends in CV is the adoption of the transformers [20, 21]. Dosovitskiy et al. proposed a Vision Transformer (ViT) [20] in which 16x16 patches from an image are used as inputs of a standard transformer for image classification. Visual Transformer (VT) [21] adopted a transformer using semantic tokens from convolutions and spatial attention so as to treat an image focussing more on important regions.

III. Method

The proposed Line-Transformers aim to build a line descriptor given points located on a line segment (Fig. 2).

A. Line Transformers

1) Line Tokenizer: We briefly illustrate NLP terminologies and explain our method based on a similar concept. In NLP, word tokenization is a process used to divide a sentence into smaller words called tokens. The tokens are utilized as minimum units for model inputs. Special tokens are also used to perform special tasks. For example, the classification token `[CLS]` is applied to aggregate sequence representation for classification tasks and the separator token `[SEP]` is used to differentiate two sentences. These tokens are converted into vector representations by allocating words with similar meaning into a similar vector space. The vector representation of tokens is called word embedding, and NLP models exploit them to understand natural languages effectively. In our problem, we formulate a point-and-line segment as the relationship between a word and a sentence in the natural language.

As in Fig. 3, the line tokenizer aims to generate point tokens and their embeddings that describe a line segment. After detecting line segments from an image [22], we uniformly select points $\mathbf{p}$ on a line segment. The position of point $\mathbf{p}$ consists of its image coordinates and keypoint confidence: $\mathbf{p_i} = (x, y, c)_i$. The inter-points interval (v) is then determined depending on the level of discriminability. When the interval of points is small, the model can receive more information to describe a line but needs larger computations and memory. The number of points (n) on a line is $n = \lfloor \ell / v \rfloor + 1$ when a line length is ℓ. As the `[CLS]` token in BERT [19], we introduced a special line token `[LINE]` to aggregate contextual information of

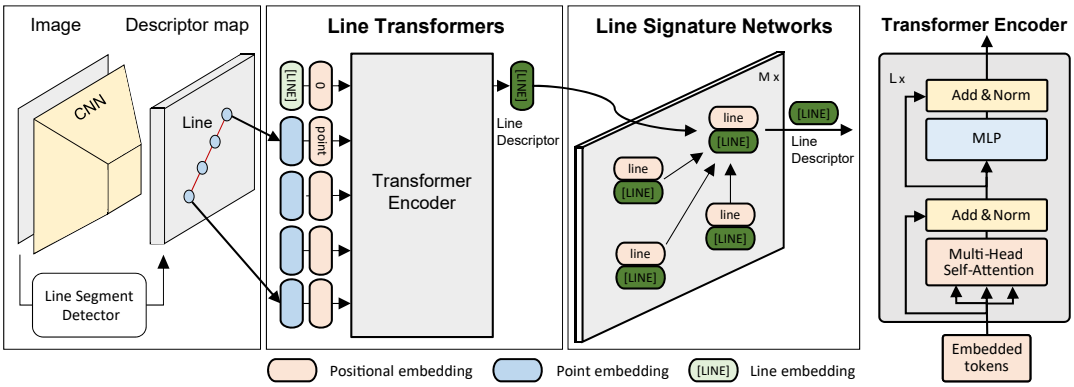


Fig. 2: Line-Transformers consist of two major components: *line transformers* and *line signature networks*. The first component uses a *line tokenizer* to extract point tokens and embeddings from a line segment. Considering the context of the point embeddings, transformers summarize it into a line embedding, or a line descriptor. The second component enhances the line descriptor by sharing lines' positional context with its neighborhoods.

point tokens. The line token is prepended on the series of point tokens.

We encode each point token into a vector representation, point embedding $\mathbf{E} \in \mathbb{R}^{1\times D}$, where D is the descriptor dimension. This is done by associating each point token with a vector that matched in a dense descriptor map [18] encoded with a descriptor vector at each pixel. The embedding of the special token `[LINE]`, $\mathbf{E_{line}} \in \mathbb{R}^{1\times D}$, expresses the initial state of a line descriptor, and its weight values are learned during the training process. Finally, we add the embeddings to the positional embeddings $\mathbf{E_{pos}} \in \mathbb{R}^{(\mathbf{n+1})\times \mathbf{D}}$ which is obtained by multilayer perceptron (MLP) using each point's position.

2) Transformer: Given the token embeddings, we model a line descriptor using transformers [12]. The transformer encoder is composed of two sublayers: multi-head self-attention (MSA) layers and MLP layers, whereas each sub-layer has a residual connection and layer normalization (LN).

We stack the transformer L times as in (1). Here, the token embeddings $\mathbf{z_0}$ serve as the encoder inputs. The line embedding $\mathbf{E_{line}}$ is located at the first element of $\mathbf{z_0}$, and is expressed with the superscript 0. Then, at the last layer $\mathbf{z}_L$, the line embedding contains the line context (i.e., $\mathbf{E_{line}} = \mathbf{z_0^0}$ and a line descriptor $\mathbf{d} = \mathbf{z}_L^0$).

$$
\begin{aligned}
\mathbf{z_0} &= [\mathbf{E_{line}}; \mathbf{E_1}; \mathbf{E_2}; ...; \mathbf{E}_n] + \mathbf{E_{pos}}, \\
\mathbf{z'_{i-1}} &= \mathrm{LN}(\mathrm{MSA}(\mathbf{z_{i-1}}, \mathbf{mask_0}) + \mathbf{z_{i-1}}), \\
\mathbf{z}_i &= \mathrm{LN}(\mathrm{MLP}(\mathbf{z'_{i-1}}) + \mathbf{z'_{i-1}}), i = 1...L \\
\mathbf{d} &= \mathbf{z}_L^0
\end{aligned}
\tag{1}
$$

Each line segment has a various number of point tokens based on the length ℓ. To handle the various sizes at once, we use a mask vector $\mathbf{mask_0}$ for the MSA to reject the unrelated point embeddings inside of scaled dot-product attention. The mask has the size $1 \times n_{max}$.

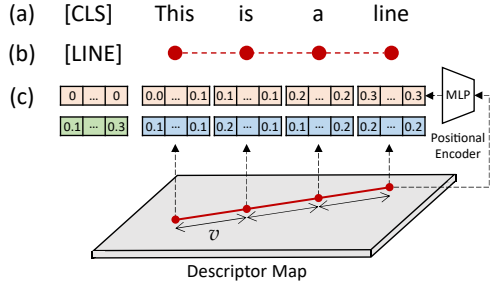


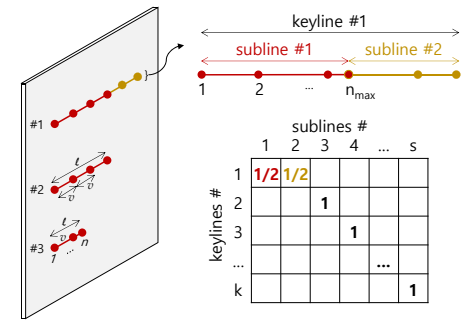
Fig. 3: (a) In BERT [19], a sentence is tokenized, and a `[CLS]` is prepended for a sentence classification task. (b) Similar to BERT, a line segment is tokenized, and a special line token `[LINE]` is used for aggregating points information. (c) The point embedding is a descriptor vector located on each point token on a descriptor map.

B. Line Signature Networks

Inspired by line signatures [23] and message-passing [24, 25] in graph neural networks (GNN), the proposed deep line signature networks are designed for line signatures' messages to be shared with each line segment using graph attention networks. The line signature is originally proposed as a grouped line descriptor. It clusters neighbor line segments as a group and takes relative positions by a series of angles and length ratios. While the line signature needs to define the clustering range of neighbor line segments and the man-defined attributes of lines, we exploit a graph attention network that can implicitly assign neighbor line segments and pass the messages of their descriptions including positions.

$$
\begin{aligned}
\mathbf{d'}_i &= \mathbf{d}_i + \mathrm{MLP}(\mathbf{x}_i, \mathbf{y}_i, \ell_i, \cos\theta_i, \sin\theta_i) \\
\mathbf{s_0} &= [\mathbf{d'_1}; \mathbf{d'_2}; ...; \mathbf{d'}_m] \\
\mathbf{s}_j &= \mathbf{s_{j-1}} + \mathrm{MLP}\Big(\mathbf{s_{j-1}} || \mathrm{MSA}(\mathbf{s_{j-1}})\Big), j = 1...M
\end{aligned}
\tag{2}
$$

We first make an attribute embedding by feeding the line's attributes such as midpoints (x, y), angle θ, and length ℓ to MLP. Then, we add it on descriptor $\mathbf{d_i}$ and share the messages with each line segment using the message-passing network as in (2). The operator $||$ represents concatenation, and m denotes the number of line descriptors within an image. We also stack the message-passing layers M times. Finally, we normalize line descriptors in $\mathbf{z}_M$ after feeding them to another MLP.



(a) Keylines & Sublines (b) Adjacency matrix

Fig. 4: Keylines, sublines, and adjacency matrix. Points are selected with the interval v on a line segment. When the number of points n is greater than n_{max}, the keyline is divided into two sublines as in (b). The adjacency matrix describes relationship between keylines and sublines.

C. Sublines to Keylines

For transformers, the number of input tokens is limited by its maximum size (n_{max}). Overcoming the limited text length of transformers, the longer sequences were often truncated in NLP. Instead of truncation, we handle long line segments utilizing the keyline and subline concepts. Let us call the original line segment a *keyline*. When the keyline is longer than the maximum ($\ell > n_{max} \times v$), we divide it into multiple *sublines*. In doing so, we can build an adjacency matrix $\mathrm{A_{img}}$ via the relationship between keylines and sublines. The values of the adjacency matrix are one divided by the number of sublines, as presented in Fig. 4(b). Then, the distance matrix of sublines $\mathrm{C_{sublines}}$ can be transformed into a distance matrix of keylines $\mathrm{C_{keylines}}$ as follows:

$$\mathrm{C_{keylines}} = \mathrm{A_{img1}} \cdot \mathrm{C_{sublines}} \cdot \mathrm{A_{img2}}^{\top},$$

where the adjacency matrices of two images are $\mathrm{A_{img1}}$ and $\mathrm{A_{img2}}$. The distance matrix of sublines includes the distance between descriptors from two images, which can be calculated by a matcher such as nearest neighbor.

Similar to the ensemble averaging, (III-C) averages the multiple sublines' distances with respect to a keyline via matrix multiplication of adjacency matrix with distance matrix. For example, the distance between keylines in `image1` and sublines in `image2` can be represented in $\mathrm{A_{img1}} \cdot \mathrm{C_{sublines}}$.

D. Loss Function

We use a triplet loss function with semi-hard negative sampling strategy [26]. The basic idea of triplet loss is to make the distance between an anchor descriptor $\mathbf{a_i}$ and its matched (positive) descriptor $\mathbf{p_i}$ closer and to simultaneously further the distance with a nonmatched (negative) descriptor $\mathbf{n_i}$. In line segment matching, one line in one image can be matched with more than two lines in another image and this means that a single anchor line can have multiple positive lines. In our implementation, we choose the most overlapped line as the positive $\mathbf{p_i}$. The overall loss function is given as:

$$\mathcal{L} = \frac{1}{n}\sum_{i}^{n} \max(0, \alpha + \|\mathbf{a}_i - \mathbf{p}_i\|^2 - \left\|\mathbf{a}_i - \mathbf{n}_i'\right\|^2), \quad (3)$$

where the semi-hard negative sample $\mathbf{n}_i'$ is chosen to the hard negative further away from the positive $\mathbf{p}_i$. We observed that semi-negative sampling helped converging loss values stably. The margin value α provides the capacity to increase the negative distances, and we set it to 1.

E. Implementation Details

To detect line segments on images, we used Line Segment Detector (LSD) [22] for its high generalizability over various environments. We selected SuperPoints [18] for our front-end descriptor map. Because its raw descriptor map is sized $H/8 \times W/8$ given an image sized $H \times W$, we set 8 to the interval of points on a line. We limited the number of point tokens greater than 2 and smaller than 21 for a subline. A line descriptor, key, query, and value in MSA have the same dimension $\mathbf{D} = 256$ as the SuperPoint. The MSA has four heads, and the line transformers and line signature networks have $L = 12$ and $M = 7$ layers, respectively. Our networks contain 14M parameters, and they run at an average speed of 20 ms on an NVIDIA GTX 2080Ti GPU for 256 line descriptors in an image. It is implemented in Pytorch using the Adam optimizer with a learning rate of 0.001.

IV. Experiments

We evaluate our line descriptor in terms of homography estimation and visual localization performance. For two test scenarios, we compare the proposed method against SuperPoint, handcrafted line descriptor LBD [5], learning-based line descriptors LLD [8], WLD [10], and SOLD^2 [11]. We use a nearest neighbor (NN) matcher for LBD, LLD, and WLD, and SOLD^2 by means of its own line matcher. We include SuperPoint as a reference for the point-based matching. We used a maximum of 512 and 1,024 points at each dataset. For lines, we extracted 256 line segments in the longer order. Further results can be seen in the video `line-as-a-visual-sentence.mp4`.

A. Homography Estimation

1) Datasets: For homography estimation, we use Oxford and Paris dataset [27]. The dataset includes 5K (Oxford) and 6K (Paris) images and we split them into the training, validatation, and test sets. For self-supervised learning, we augment images using synthetic homographies and photometric distortions [18, 24, 28].

To establish ground-truth line correspondences from an image pair, we first detect line segments from both the original image and its augmented image. Then, we project two endpoints of each line onto one another using a known homography matrix. The criteria for the true correspondence are (*i*) overlap existence, (*ii*) reprojection error (< 4 px), and (*iii*) angle difference ($< 2°$). The resulting true correspondences were expressed as an overlap-similarity matrix. The overlap similarity between two lines measures the ratio between overlapped line length and smaller line length as $O(L_1, L_2) = L_1 \cap L_2 / \min(L_1, L_2)$. The overlapped line length $L_1 \cap L_2$ is the distance between two middle endpoints

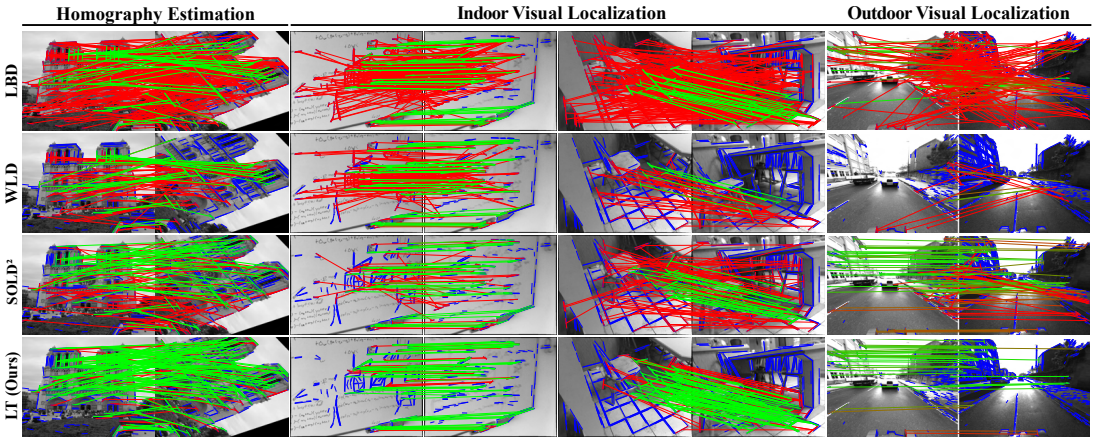


Fig. 5: Qualitative line segment matches for homography estimation and visual localization. The last three columns represent blurring, viewpoint, and illumination changes. More correct matches (green) and fewer wrong matches (red) indicate better performance. Unmatched lines are colored in blue.

TABLE I: Homography estimation evaluation. The best-performing values are in bold font.

	Homography AUC			P	R	F
	AUC@5px	AUC@10px	AUC@20px			
(SuperPoint)	(38.5)	(43.3)	(46.3)	(37.6)	(38.6)	(39.6)
LBD	2.2	7.6	17.5	20.6	55.2	30.1
LLD	0.7	2.6	6.6	5.9	13.6	8.3
WLD	16.7	35.2	54.5	48.0	51.3	49.6
SOLD^2	**31.8**	51.5	67.1	41.1	45.8	43.3
LT (Ours)	29.5	**52.1**	**69.4**	**57.7**	**61.5**	**59.5**

among four endpoints of two lines. For SuperPoint, true point correspondences are defined by the point-wise reprojection errors (< 4 px) following [24].

2) Metrics: We implement a line-based homography matrix estimation [29]. With the homography estimation utilizing a random sample consensus (RANSAC) of 2,000 iterations, we compute average reprojection errors of the four image corners and report the area under the cumulative error curve (AUC) at the thresholds (5, 10, and, 20 pixels). We also compute matching precision (P) and recall (R) based on the ground-truth matches.

3) Results: Table. I lists the quantitative comparison. Our Line-Transformers outperform other line descriptor methods in terms of F-score by a large margin (10.1%). Our method also outperforms every homography estimation metric except AUC under five pixels. Compared to SuperPoint, the Line-Transformers yielded more stable performance at the AUC under 10 and 20 pixels. LLD has low performances on this dataset because it originally trained in continuous frames without large viewpoint changes.

The precision and recall are a direct and explicit measure for line-matching performance depending solely on the number of correct/wrong matches. More implicit performance could be captured from homography estimation when the performance depends on the number, distribution, and quality of the matches. In that sense, the proposed method fulfilled the quantity and quality of the reliable matches. We discuss the matching evaluation in more detail in §IV-D.

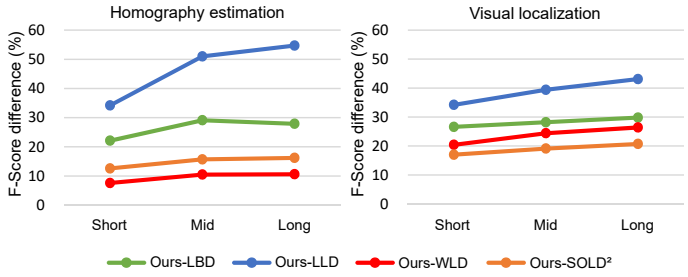
Fig. 5 shows qualitative results for the homography estimation-based line matching. Line-Transformers have better performance by producing more correct matches and fewer false matches than other line descriptors. It also shows that LBD in particular has many incorrect matches, resulting in lower matching precision.

B. Visual Localization

Next, we evaluate line descriptors by estimating a camera pose in 3D line maps. The evaluations were performed using the ScanNet [30] and Oxford Radar RobotCar [31] datasets for indoor and outdoor experiments. For indoor environments, we trained our networks in supervised learning, and initialized them with the homography estimation's weights. Then, the weights trained using ScanNet were applied directly to outdoor localization to observe our networks' generalizability.

1) Indoor: The ScanNet dataset [30] includes 2.5M views of RGB-D data and 3D camera poses in 1,513 indoor scenes. We generate ground-truth line correspondences and 3D line maps. Similarly, as in the research [32, 24], we selected sufficiently overlapping image pairs (40–80%) by investigating the depth maps. Due to the potential inaccuracy in the depth maps, we decomposed a line into point arrays and examined the intermediate points to validate the line correspondences. Then, we randomly select 350 image pairs from each of the four overlapped scenes for the test set, a total of 1,400 image pairs. Because depth maps are not sufficiently accurate to find all ground-truth line pairs, some correct line correspondences may be counted as a false negative. Therefore, we mitigate this issue by checking scene depth quality, as well as observing the ratio between the numbers of all extracted line segments and the validated line segments during projection/unprojection procedures.

2) Outdoor: The Oxford Radar RobotCar dataset [31] is built upon the Oxford RobotCar dataset [33]. We selected two sequences (2019-01-11-12-26-55 and 2019-01-16-13-09-37) for our reference and query sets. Then, we randomly selected 300 query images in a sequence and performed visual place recognition [34] to identify corresponding reference images that have a 3D line feature map. Instead of using the global positioning system (GPS) potential unreliable signals, we have computed the ground-truth relative pose between query and reference images using their point clouds via Iterated Closest



(a) Performance difference between Line-Transformers and other descriptors.

		Precision			Recall			F-score		
		Short	Mid	Long	Short	Mid	Long	Short	Mid	Long
Homography etimation	LBD	10.9	20.8	28.0	40.5	54.9	55.0	17.1	30.2	37.1
	LLD	3.3	6.0	8.1	10.3	13.3	14.1	5.0	8.3	10.3
	WLD	24.9	43.5	53.9	43.3	55.4	55.0	31.6	48.8	54.4
	SOLD²	24.3	39.6	50.0	29.5	48.5	47.7	26.6	43.6	48.8
	LT (Ours)	**35.3**	**57.5**	**67.5**	**44.1**	**61.3**	**62.7**	**39.2**	**59.3**	**65.0**
Visual localization	LBD	11.3	17.7	27.1	42.3	47.9	53.4	17.8	25.9	35.9
	LLD	6.8	9.8	16.5	26.3	29.1	35.9	10.2	14.7	22.6
	WLD	18.6	25.6	38.7	33.8	35.3	39.9	24.0	29.7	39.3
	SOLD²	24.7	30.8	45.7	30.8	40.4	44.4	27.4	35.0	45.0
	LT (Ours)	**34.7**	**45.1**	**59.5**	**61.6**	**67.6**	**73.3**	**44.4**	**54.1**	**65.7**

(b) Matching performances by line length.

Fig. 6: Performance difference by line length. The figure (a) illustrates that the overall graph has an upward trajectory, thus showing that our method performs better than other CNN-based line descriptors when line segments extend.

TABLE II: Visual localization

	Visual Localization AUC (Indoor) 0.25m/10°		0.50m/10°		1.00m/10°		P	R	F
(SuperPoint)	(83.6 PnP)		(86.4 PnP)		(86.8 PnP)		(49.5)	(69.1)	(57.7)
	PnL	PnPL	PnL	PnPL	PnL	PnPL			
LBD	15.7	38.4	19.0	45.9	19.3	46.6	18.5	49.8	27.0
LLD	12.0	35.8	16.1	41.6	16.4	42.5	10.5	31.8	15.8
WLD	28.0	55.9	35.4	61.6	36.1	62.4	27.9	37.3	31.9
SOLD²	46.4	73.5	57.4	77.9	59.2	78.6	35.0	40.5	37.5
LT (w/o LS)	47.4	76.9	59.8	81.6	61.4	82.5	42.4	62.2	50.4
LT (Ours)	**53.0**	**79.2**	**65.0**	**83.3**	**66.6**	**83.9**	**49.4**	**68.4**	**57.3**
	Visual Localization AUC (Outdoor) 0.25m/2°		0.50m/5°		5.00m/10°		P	R	F
(SuperPoint)	(37.1 PnP)		(63.6 PnP)		(90.8 PnP)				
	PnL	PnPL	PnL	PnPL	PnL	PnPL			
LBD	1.9	4.5	2.6	18.5	9.1	59.2	-	-	-
LLD	1.9	7.5	6.4	30.2	20.0	71.7	-	-	-
WLD	9.4	21.5	29.4	45.7	54.7	83.4	-	-	-
SOLD²	21.5	28.7	54.0	53.2	82.3	90.2	-	-	-
LT (w/o LS)	21.9	29.4	55.1	58.9	87.9	91.7	-	-	-
LT (Ours)	**26.8**	**30.9**	**57.7**	**61.1**	**90.2**	**91.3**	-	-	-

Point (ICP). In the final evaluation sets, we excluded query-reference image pairs with poor ICP fitness. The 3D line maps were generated following procedures similar to those for our indoor evaluation. Unlike the indoor RGB-D camera, however, the projected LiDAR points are so sparse that 2D line segments can be difficult to find their corresponding depth value. To alleviate this, we utilized a depth completion [35].

3) Metrics: We report the percentage of correctly localized query images while using different thresholds (i.e., 0.25m, 10/ 0.5m, 10/ 1.0m, 10 for indoor and 0.25m, 2/ 0.5m, 5/ 5.0m, 10 for outdoor). We estimated the camera pose by Perspective-n-Point-Line (PnPL) [36] with a RANSAC of 20 iterations. For SuperPoint, we leveraged Perspective-n-Point (PnP). We analyze the pose estimation results that use points, lines, and points-and-lines, respectively. We also report matching precision (P) and recall (R) based on the ground-truth matches at indoor.

4) Results: For both indoor and outdoor tests, Line-Transformers achieve the highest performance among other line descriptors in visual localization and precision-recall metrics (Table. II). The qualitative results in Fig. 5 illustrate that the Line-Transformers perform robustly in imaging changes such as blurring, viewpoints, and illuminations. Our method can be generalized well performing reliably using the same weights trained in ScanNet datasets.

Unlike in the homography estimation, the point-based method with PnP outperforms all line-based methods. One of the reasons is the small number of 3D line inliers during depth validation. While the 3D feature point is directly determined by its corresponding depth pixel, some 3D line features are filtered out during depth linearity's validation in RANSAC. Thus, within our mapping method, line-based localization is prone to performance degradation than point-based. However, as will be discussed in §IV.F, the line features can complement the performance of the points, especially when point feature numbers are small or biased.

TABLE III: Ablation of Line-Transformers.

Line-Transformers	Line-based Visual Localization 0.25m/10°	0.50m/10°	1.00m/10°	P	R	F
No Line Signature Net (LS)	47.4	59.8	61.4	42.4	62.2	50.4
No Positional encoding in LS	49.4	62.8	64.3	44.3	66.4	53.1
No mid-point in LS	51.1	62.6	64.9	47.7	69.0	56.4
No length, angle in LS	51.9	64.2	65.8	44.0	68.4	53.6
Full	**53.0**	**65.0**	**66.6**	**49.4**	**68.4**	**57.3**

C. Performance by Line Length

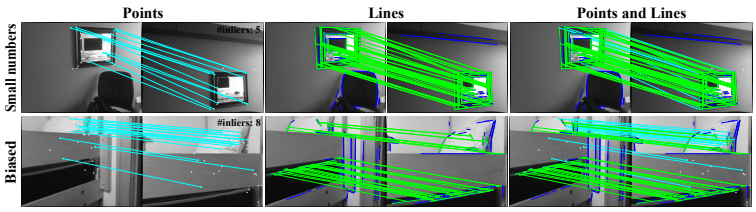
To examine the robustness of our model with respect to various line lengths, we further investigate performance comparisons against CNN-based methods while varying line length. From each dataset, we evenly divide line segment sets into three groups (i.e., short, mid, and long) by their lengths. Each group has 33% numbers of all line segments, and the mid-length group ranged from about 30 to 50 pixels.

As presented in Fig. 6(b), the proposed Line-Transformers outperform other descriptors for all line lengths. Furthermore, as shown in Fig. 6(a), the matching performance difference between Line-Transformers and other CNN-based line descriptors (LLD and WLD) is increased by line length. This tendency indicates that the performance of the proposed method is preserved even with longer line segments, unlike the handcrafted descriptor and CNN-based descriptor.

D. Discussion on Evaluation Metrics

Unlike point features that assume one-to-one matching, line matching is a many-to-many problem because a line detector tends to break the same line segment into small lines differently at each image pair. For example, two non-overlapping lines may originate from a single line due to the occlusion and segmentation; thus, they should be considered the correct correspondences semantically.

Evaluation of line segment matches often depends on human inspections [37, 10, 9, 5], assuming that line matches are one-to-one correspondences [5], not many-to-many. Therefore, when a matcher finds only the closest matching pair, the precision and recall should be carefully considered cautioning



(a) Point-and-line features

	Indoor Visual Localization			
	#inliers	0.25m/10°	0.50m/10°	1.00m/10°
SuperPoint	-	83.6	86.4	86.8
PL-Loc	<20	83.6	86.5	87.2
	<5	**84.5**	**87.4**	**88.0**
Best Feature (%)	P/L/PL	61.0 / 6.3 / 32.7		

(b) Point-and-Line Visual localization

Fig. 7: Points and lines are complementary for better localization, especially when keypoints are biased or in small numbers.

many-to-many correspondences. In that sense, the precision-recall metrics may be limited because they cannot consider non-overlapping line correspondences.

Visual localization and homography estimation could be more proper metrics in this aspect. In visual localization, matching with the non-overlapped but semantically same line is also considered a good match because a Perspective-n-Line (PnL) algorithm does not consider the endpoint's positions to ensure the robustness for changing endpoints. Similarly, the line-based homography estimation does not consider endpoints [29] but is limited to planar scenes in the real 3D world. Hence, we found that line-based visual localization is a better alternative that can inspect both overlapped and non-overlapped line matches with large perspective changes.

E. Understanding Line-Transformers

To examine the attention scores of Line-Transformers, we visualize the maximum attention score of each multi-heads within a line segment (Fig. 8). The point embeddings on a line are not considered equally for a line descriptor, but they have their own attention patterns. We also observe that the matched lines have similar attention patterns and the line descriptors tend to refer to the endpoints of a line.

The Fig. 8 (b) illustrates the attention between line descriptors in line signature networks. We observe that the attentions gradually focus on a small number of neighbor lines at a later layer. To study the quantitative effect of the line signature networks, we evaluate our models without line signature networks. As presented in Table. II and Table. III, the line signature networks improve the localization results from 47.4% to 53.0% under the threshold of 0.25m and 10.

F. Visual Localization Using Feature Points and Lines

Despite the generally better performance in Table. II, the point-based localization may be deteriorated due to the small number of points (e.g., feature-less environment) or biased feature distribution as in the sample cases in Fig. 7. This section examines how the line-based approach can enhance point-based localization in a complementary fashion.

For the analysis, we define point-based localization failure using the reprojection errors of 3D features and count the inlier when the reprojection error is less than four pixels. Then, the PL-Loc was additionally performed for the cases when the number of inliers is less than 5 or 20 (i.e., when point-based inliers drop).

As presented in the table in Fig. 7(b), the PL-Loc provides additional enhancements to visual localization. More interestingly, the point outperformed over the line for 61% of cases, which indicates the remaining 39% of cases are with potential to be improved with lines. This also implies the proper combination of point and line would improve the overall localization performance. For example, better metrics followed by a strong model selection could be examined to complete robust PnPL in future work.

V. Conclusion

This paper presented a novel line descriptor handling variable line length effectively using attention mechanisms, inspired by NLP tasks handling sentences and paragraphs of various lengths. We also presented a PL-Loc pipeline leveraging keypoints and keylines simultaneously for visual localization. Our experiments demonstrated that our line descriptor achieved state-of-the-art performance in homography estimation and visual localization datasets.

References

[1] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "Pl-svo: Semi-direct monocular visual odometry by combining points and line segments," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2016, pp. 4211–4216.

[2] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2017, pp. 4503–4508.

[3] R. Gomez-Ojeda, F. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "Pl-slam: A stereo slam system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, pp. 734–746, 2019.

[4] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang, "Visual-inertial odometry with point and line features," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2019, pp. 2447–2454.

[5] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *J. of Visual Comm. and Img. Rep.*, vol. 24, pp. 794–805, 2013.

[6] C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla, F. Kahl, and T. Sattler, "Long-term visual localization revisited," *IEEE Trans. Pat. Anal. and Mach. Intell.*, pp. 1–1, 2020.

[7] P. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2019, pp. 12 716–12 725.

[8] A. Vakhitov and V. Lempitsky, "Learnable line segment descriptor for visual SLAM," *IEEE Access*, pp. 1–1, 2019.

[9] M. Lange, F. Schweinfurth, and A. Schilling, "Dld: A deep learning based line descriptor for line feature matching," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2019, pp. 5910–5915.

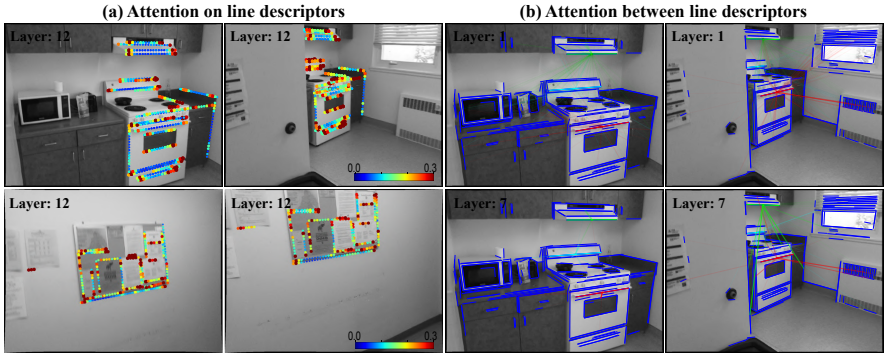


Fig. 8: Visualizing attention scores. (a) Attention patterns in lines describe how much the points embeddings contribute to building line descriptors. The matched lines follow similar attention patterns. (b) Attention scores between line descriptors are initially low and widely spread, and they are gradually converged onto a small number of neighbor lines at a later layer.

[10] M. Lange, C. Raisch, and A. Schilling, "WLD: A Wavelet and Learning based Line Descriptor for Line Feature Matching," in *Proc. Intl. Symp. on Vis., Mod., and Vis.*, 2020, pp. 39–46.

[11] R. Pautrat, L. Juan-Ting, V. Larsson, M. R. Oswald, and M. Pollefeys, "Sold Self-supervised occlusion-aware line description and detection," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2021.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Sys. Conf.*, 2017, pp. 5998–6008.

[13] Z. Wang, F. Wu, and Z. Hu, "MSLD: A robust descriptor for line matching," *Pattern Recognition*, pp. 941–953, 2009.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Comput. Vision*, pp. 91–110, 2004.

[15] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2017, pp. 6128–6136.

[16] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Advances in Neural Information Processing Sys. Conf.*, 2017, pp. 4826–4837.

[17] K. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *Proc. European Conf. on Comput. Vision*, 2016, pp. 467–483.

[18] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog. Workshops*, 2018, pp. 224–236.

[19] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. of the North American Chapter of the Assoc. for Comput. Linguistics: Human Lang. Tech.*, 2019, pp. 4171–4186.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *Proc. Intl. Conf. on Learning Representations*, 2021.

[21] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, M. Tomizuka, K. Keutzer, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," *CoRR*, vol. abs/2006.03677, 2020.

[22] R. G. von Gioi, J. Jakubowicz, J. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pat. Anal. and Mach. Intell.*, vol. 32, pp. 722–32, 2010.

[23] L. Wang, U. Neumann, and S. You, "Wide-baseline image matching using line signatures," in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2009, pp. 1311–1318.

[24] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2020, pp. 4937–4946.

[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Intl. Conf. on Learning Representations*, 2018.

[26] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2015, pp. 815–823.

[27] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2018, pp. 5706–5715.

[28] J. Revaud, P. Weinzaepfel, C. R. de Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2D2: repeatable and reliable detector and descriptor," in *Advances in Neural Information Processing Sys. Conf.*, 2019, pp. 12 405–12 415.

[29] E. Dubrofsky and R. Woodham, "Combining line and point correspondences for homography estimation," in *Proc. Intl. Symp. Visual Computing*, vol. 5359, 2008, pp. 202–213.

[30] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2017, pp. 2432–2443.

[31] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," *arXiv: 1909.01300*, 2019.

[32] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable CNN for joint detection and description of local features," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2019, pp. 8092–8101.

[33] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *Intl. J. of Robot. Research*, vol. 36, no. 1, pp. 3–15, 2017.

[34] J. Revaud, J. Almazán, R. S. Rezende, and C. R. de Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2019, pp. 5106–5115.

[35] J. Park, K. Joo, Z. Hu, C. Liu, and I. S. Kweon, "Non-local spatial propagation network for depth completion," in *Proc. European Conf. on Comput. Vision*, 2020, pp. 120–136.

[36] S. Agostinho, J. Gomes, and A. D. Bue, "Cvxpnpl: A unified convex solution to the absolute pose estimation problem from point and line correspondences," *CoRR*, 2019.

[37] K. Li, J. Yao, M. Lu, Y. Heng, T. Wu, and Y. Li, "Line segment matching: A benchmark," in *Proc. IEEE Winter Conf. on Appl. of Computer Vision*, 2016, pp. 1–9.