Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4024774, 11 pages
https://doi.org/10.1155/2022/4024774

Research Article

Robust Keypoint Detection and Matching on Fisheye Images by Self-Supervised Learning

Wei Tian, Pei Cai, Yongkun Wen, and Xinning Chu

School of Automotive Studies, Tongji University, 201804 Shanghai, China

Correspondence should be addressed to Wei Tian; tian_wei@tongji.edu.cn

Received 23 July 2022; Revised 1 December 2022; Accepted 5 December 2022; Published 22 December 2022

Academic Editor: Anastasios D. Doulamis

Accurate image feature point detection and matching are essential to computer vision tasks such as panoramic image stitching and 3D reconstruction. However, ordinary feature point approaches cannot be directly applied to fisheye images due to their large distortion, which makes the ordinary camera model unable to adapt. To address such a problem, this paper proposes a self-supervised learning method for feature point detection and matching on fisheye images. This method utilizes a Siamese network to automatically learn the correspondence of feature points across transformed image pairs to avoid high annotation costs. Due to the scarcity of the fisheye image dataset, a two-stage viewpoint transform pipeline is also adopted for image augmentation to increase the data variety. Furthermore, this method adopts both deformable convolution and contrastive learning loss to improve the feature extraction and description of distorted image regions. Compared with traditional feature point detectors and matchers, this method has been demonstrated with superior performance on fisheye images.

1. Introduction

In recent years, visual feature extraction and keypoint matching have been widely applied in computer vision tasks, such as motion and behavior analysis [1, 2] and visual localization [3], which are essential to autonomous driving vehicles. In autonomous driving perception tasks, the traditional way to obtain environmental information is to use a narrow-angle pinhole camera, which yet has a limited field of view (FOV), and thus leads to a large range of blind spots. On the one hand, when the camera pose changes, the limited viewing angle can lead to the loss of feature points. On the other hand, the small FOV of the narrow-angle pinhole camera can be easily occupied by dynamic vehicles and pedestrians, resulting in incorrect pose estimation.

In contrast, the fisheye camera can perceive a wide range of a scene, and even obtain visual information about the hemispheric domain theoretically [4]. Figure 1 shows the visual difference between fisheye images and standard images. The middle part of the fisheye image protrudes and the part on the image boundary is compressed, leading to significantly varied resolution across the image. This distortion characteristic is a particular challenge for vision tasks such as keypoint matching and object detection. Standard images are with a consistent resolution and look closer to the real world. Usually, fisheye images should be rectified before applying conventional image-processing algorithms.

The large distortion in the fisheye image is attributed to the unconventional fisheye lens, which corresponds to a nonlinear projection as shown in Figure 2. In the pinhole projection model, the perspective projection of a point $\mathbf{P}$ from the 3D camera coordinate system X-Y-Z to the imaging plane u_s-v_s (denoted as u_I-v_I in the fisheye model) can be simply formulated by

$$\rho = f \cdot \tan\theta, \tag{1}$$

where ρ denotes the distance between the projected point $\mathbf{p}'$ on the imaging plane and the optical axis while f is the focal length. The angle of incident light is denoted as θ. However, the nonlinear projection of a fisheye lens is more complex and can be expressed by different mathematical models [4] according to the design and manufacturing, such as stereographic projection, equidistance projection, equisolid

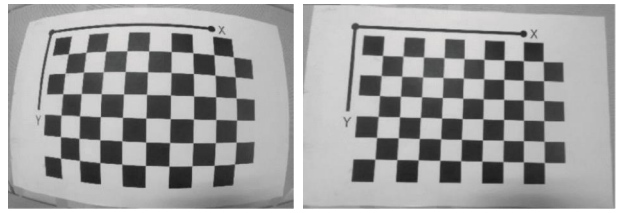

Figure 1: The significant visual distortion of the fisheye image (a) and compared to that of the standard image (b).

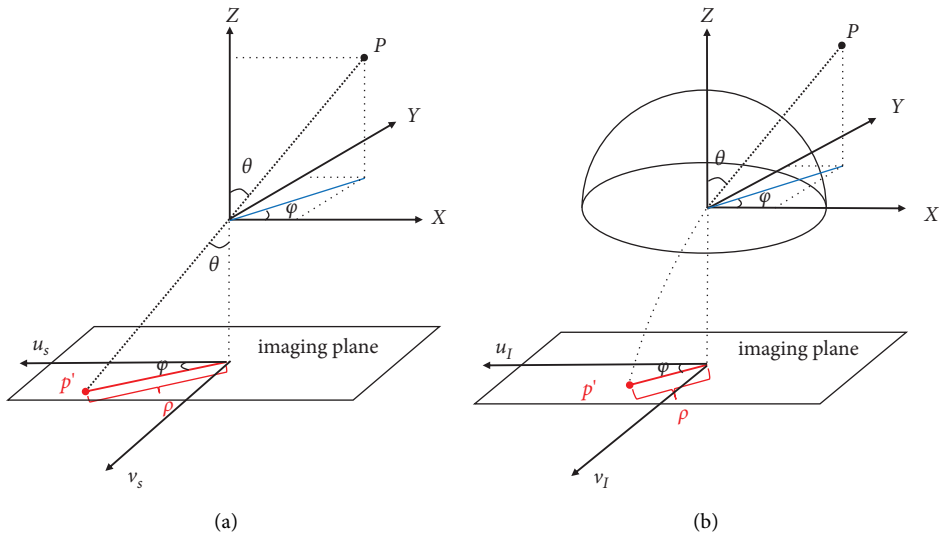

Figure 2: Projection models of the pinhole camera (a) and fisheye lens camera (b).

angle projection, and orthogonal projection, respectively, interpreted as follows:

$$\begin{aligned} \rho &= 2f \cdot \tan\frac{\theta}{2}, \\ \rho &= f \cdot \theta, \\ \rho &= 2f \cdot \sin\frac{\theta}{2}, \\ \rho &= f \cdot \sin\theta. \end{aligned} \tag{2}$$

The spatially varying distortion induced by the fisheye lens leads to strong appearance variations of the objects, especially for those in close-by surroundings [5]. Therefore, the processing algorithms for fisheye images are much more sophisticated, which are comparatively underexplored than those on standard images. However, the research about processing fisheye images is of great practical significance, as fisheye cameras have been widely applied in many fields such as navigation, road and tunnel inspection, and video surveillance, with details stated as follows. (1) Navigation: mobile robot navigation with panorama vision is one of the focuses of current researches. The perception module consisting of fisheye cameras can obtain a surround-view perception of the environment at a reduced number of perception sensors, and benefit the subsequent tasks such as trajectory tracking and navigation [6]. (2) Road and tunnel inspection. Health assessments of infrastructures are essential for construction tasks. For surface damage detection with a coverage of 360°, techniques with panorama vision such as fisheye cameras are prevalent [7–9], which helps to avoid serious incidents and thus ensure public safety. (3) Video surveillance: the hemispherical lens is commonly applied in modern surveillance devices [5] to provide a large FOV containing as much information as possible from the monitored environments. Fisheye cameras are also highly favored in tasks related to autonomous driving and 3D reconstruction, where accurate keypoint matching lays a solid foundation for follow-on vision tasks. However, due to significant distortion, general camera models (such as the pinhole model) and ordinary keypoint descriptors cannot be well applied in processing fisheye camera images (Figure 3).

Currently, research works on fisheye images mostly focus on undistortion schemes [10, 11]. In the image registration task, these schemes are utilized to undistort fisheye images, on which the keypoints are extracted and matched. However, the undistortion process in such methods will inevitably give rise to field-of-view loss and resampling artifacts [5]. Let alone, very few pioneer researches have explored keypoint detection and matching, which can directly apply to fisheye images. Additionally, uncertainties or noises in images can also influence the detection. Effective solutions are image preprocessing methods such as fuzzy logic-based ones [12, 13].

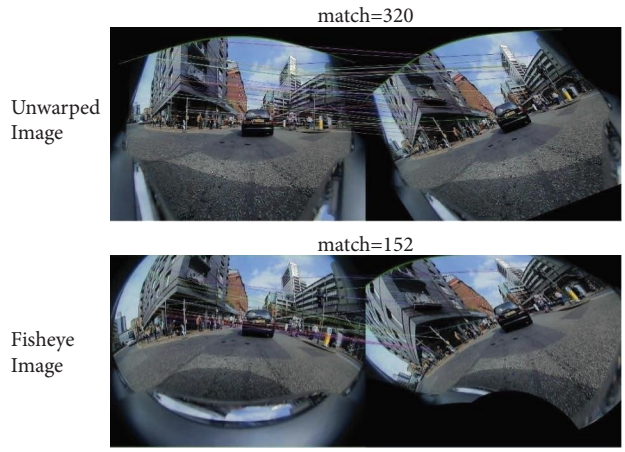

Figure 3: An example of image rotation. The number of matched points by SIFT on unwarped images is 320 (a), while on fisheye images (b), it is only 152, with a reduction of more than a half.

To date, keypoint models can be mainly categorized into traditional and deep learning-based methods. Compared to traditional ones, descriptors generated by deep learning can interpret much richer image information. Under the background that deep learning-based methods gradually occupy the mainstream, the research of fisheye images in this field currently encounters the following problems:

(i) Computer vision algorithms based on supervised learning require large-scale accurately annotated images. However, the scarcity of well-labeled fisheye image datasets limit the development of corresponding image-processing algorithms based on supervised learning.

(ii) The nonlinear projection of the fisheye lens leads to the large distortion of images. Therefore, image-processing algorithms based on the pinhole camera model cannot be directly applied to fisheye images. It is necessary to create algorithms to extract features according to the characteristics of fisheye images.

Considering the problems, we propose a self-supervised learning method for fisheye image keypoint detection and matching, whose performance surpasses the traditional models.

Our contributions are summarized as follows:

(i) We introduce a keypoint detection and matching approach for fisheye images based on self-supervision within one round of learning

(ii) We present an image transform pipeline to simulate the viewpoint change of fisheye images, which can help the self-supervised learning of keypoint correspondences across images

(iii) We integrate both the deformable convolution and the contrastive learning loss into the network to strengthen the feature learning on fisheye images

(iv) We conduct comprehensive evaluations on the WoodScape fisheye dataset and demonstrate that our method outperforms the baseline, as well as the traditional methods such as SIFT, SURF, ORB, BRISK, KAZE, and AKAZE.

The remainder of this work is organized as follows: Section 2 gives an overview of related work. Section 3 introduces the fisheye image viewpoint transform scheme, and the self-supervised learning approach for fisheye image keypoint detection and description. Section 4 shows the experimental results. Section 5 concludes this work.

2. Related Work

Here, research studies related to this work are reviewed in three aspects: (a) handcrafted keypoint models, (b) learning-based keypoint models, and (c) fisheye image undistortion approaches.

2.1. Handcrafted Keypoint Models. Traditional feature point detection methods include FAST [14], SIFT [15], SURF [16], ORB [17], KAZE [18], and AKAZE [19]. The FAST is a simple and efficient detector by comparison only with the surrounding pixels [14]. However, it cannot characterize feature points. Unlikely, the SIFT includes a descriptor of local image features that are invariant to rotation, scaling, and brightness changes, and also maintain a stability to a certain extent for angle changes, affine transforms, and noise [15]. However, its computational load is high. The SURF is a simplified version of SIFT with gradient approximation by Haar-like filters [16]. However, its advantages on runtime are still limited. The ORB algorithm is based on the directional FAST feature detection and the BRIEF feature description [17]. KAZE [18] and AKAZE [19] deploy approximations to speed up calculation in nonlinear scales. It enjoys a fast processing speed and can be applied in scenarios with high real-time requirements.

2.2. Learning-Based Keypoint Models. Simo-Serra et al. proposed a simple scheme of a Siamese network consisting of two same branches to learn the discriminating representation of a local patch [20]. By mining both positive and negative samples, they achieved high performance in the patch description. The LIFT [21] uses a spatial transformer layer to rectify the image patch for feature point detection, description, and orientation estimation. However, it is trained in multiple steps and requires the supervision from structure from motion (SFM) systems. The QuadNetworks [22] trains CNNs to rank points in a transform-invariant fashion. They can perform both single-modal and cross-modal interest point detection, yet without providing descriptors. The TILDE [23] selects keypoint candidates across multiple images from the same viewpoint to learn regressors, which are robust against drastic image changes by weather and lighting conditions. However, their approach is not explicitly trained for rotation and scaling invariance. The SuperPoint [24] built a self-supervised framework to train both detectors and descriptors for interest points, which are extracted from semidense grids. This method is first trained

on synthetic data and then on real images, resulting in two tedious rounds of training. The UnSuperPoint [25] was proposed as an improvement of the SuperPoint. It predicts keypoint locations by regression, and introduces a new loss function to train point detectors within a Siamese architecture in a self-supervised manner. It requires only one round of training and does not require the generation of pseudo ground truth points. Nevertheless, the above methods are mainly applied to pinhole camera images.

2.3. Fisheye Image Undistortion. The fisheye image undistortion is to correct distortions of the image induced by the nonlinear characteristics of the lens. The correction process starts from the optical imaging model, and reconstructs the incident ray using the camera parameters obtained by the calibration. Then, it builds a spatial mapping from the spherical perspective projection to the plane (or cylinder) projection [4]. Kannala and Brandt [26] proposed a flexible radially symmetric projection model with circular control points to improve the calibration accuracy. It is easy to expand and versatile and can be applied to cameras of both narrow and wide-angle lenses. Hartley and Kang [27] proposed a new scheme that does not establish any specific distortion model, but calibrates the radial distortion in a parameterless manner. However, this scheme is relative sensitive to noise. Wang et al. [28] proposed an extremely wide-angle camera model which complies with the equidistant projection principles. Based on that, it also gives four calibration methods that can be applied to a variety of application scenarios with high accuracy.

In this paper, we also propose a deep learning-based approach for feature point detection and description. Our approach is based on the UnsuperPoint [25] yet differs from it in three points. Firstly, based on the fisheye image undistortion, we adopt an image transform pipeline for data augmentation which is consistent with the viewpoint change of fisheye images, and thus beneficial for the learning of keypoint correspondences in real scenes. Furthermore, we integrate both deformable convolution and contrastive learning loss to enhance the feature learning on fisheye images, yielding more discriminative keypoint descriptors.

3. Proposed Approach

3.1. Fisheye Image Viewpoint Transform. As in [25], the self-supervised learning of keypoints requires transformed image pairs. However, the direct homography transform used by pinhole camera images cannot be applied to fisheye images due to their nonlinear projection characteristics. Therefore, we adopt a fisheye image viewpoint transform, as shown in Figure 4. The source fisheye image is firstly undistorted according to the projection model. A homography transform is then applied on the unwarped image. After that, the image is further warped into the target fisheye image, which can be considered as the source fisheye image undergoing viewpoint change.

More specific steps about this process are described here: we define the 2D spatial mapping from the fisheye image

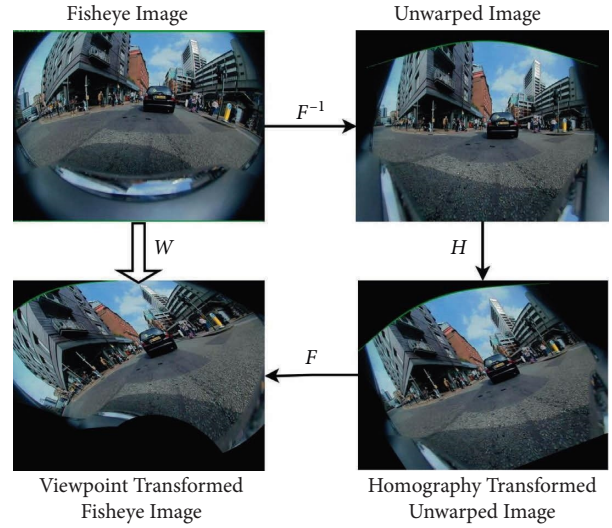

Figure 4: Overview of the fisheye image viewpoint transform.

domain $\mathbb{I}^2$ to the unwarped image domain $\mathbb{S}^2$ as: $\mathcal{F}$: $\mathbb{I}^2 \longrightarrow \mathbb{S}^2$. Thus, the inverse operation $\mathcal{F}^{-1}$ denotes the mapping from the unwarped image domain to the fisheye image domain: $\mathcal{F}^{-1}$: $\mathbb{S}^2 \longrightarrow \mathbb{I}^2$. The homography transform of an ordinary image $S \in \mathbb{S}$ is denoted as: $S_H = \mathcal{H}(S)$. With the operations described, we can generate a new fisheye image I' from the source I in following steps:

$$I' = \mathcal{W}(I) = \mathcal{F}^{-1}(\mathcal{H}(\mathcal{F}(I))). \tag{3}$$

The mapping $\mathcal{F}$ varies with the undistortion scheme. Through the mapping $\mathcal{W}$, we can obtain the paired fisheye images before and after the viewpoint transform. It should be noted that although the method is based on an undistortion scheme, the final output is still a fisheye image.

3.2. Image Warping Scheme. Here, we assume both extrinsic and intrinsic parameters of the fisheye camera are given. According to the spherical projection model, pixels on the fisheye images are firstly projected onto the spherical surface of a unit radius. Thus, points can be represented with 3D coordinates in the camera coordinate system. In a further step, the points are converted into the world coordinate system through the camera's extrinsic parameters. After that, the pinhole camera model is used to project the 3D points back to the ordinary image plane coordinates. In this way, the unwarped image after distortion correction can be obtained. Practically, to avoid image sparsity, each pixel on the new image is inversely transformed to the corresponding subpixel position on the original image, and the bilinear interpolation is used for sampling.

In this work, the camera is oriented in the horizontal direction. The image coordinate system is modified by locating its origin at the image center and changing the unit to the meter. Given a pixel with coordinates $\mathbf{p}_s = (u_s, v_s)$ on the unwarped image S_H, which has undergone the homography transform $\mathcal{H}$, we first use the pinhole camera model to project it onto the cylindrical surface and further convert it to a point $\mathbf{P}$ on a spherical surface with a unit radius.

According to [29], its 3D coordinates can be formulated as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{\sqrt{v_s^2 + f^2}} \begin{bmatrix} f \sin\theta_s \\ v_s \\ f \cos\theta_s \end{bmatrix}, \tag{4}$$

with $\theta_s = \arctan u_s / f$, and f denotes the focal length.

Then, we use the fisheye camera model to project the point from the 3D space back to the image coordinates $\mathbf{p}' = (u_I, v_I)$ on the new fisheye image I' [26]. The projection process in the fisheye camera model is shown in Figure 5. The coordinates of point $\mathbf{p}'$ can be calculated as follows:

$$\begin{bmatrix} u_I \\ v_I \end{bmatrix} = \frac{\rho(\theta)}{C} \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{5}$$

with

$$\begin{aligned} \rho(\theta) &= a_1\theta + a_2\theta^2 + \cdots + a_n\theta^n, \\ \theta &= \arccos\left(\frac{Z}{\sqrt{X^2 + Y^2 + Z^2}}\right), \\ C &= \sqrt{X^2 + Y^2}. \end{aligned} \tag{6}$$

The coefficients $a_1, \ldots, a_n$ can be provided by the fisheye camera projection model.

3.3. Self-Supervised Keypoint Learning. The fisheye viewpoint transform is incorporated into the self-supervised keypoint learning architecture as shown in Figure 6. This architecture utilizes a Siamese structure with a twin of branches. The input of branch A is the source image, while for branch B it is the viewpoint-transformed version of the source image by mapping $\mathscr{W}$. Both images undergo a random nonspatial transform such as color conversion or noising. Thereafter, a shared keypoint network is applied to predict keypoint scores, relative positions, and descriptors on both images. Prediction errors of the two branches are calculated in the loss function to guide the network training.

3.3.1. Keypoint Detection and Description Network. The keypoint detection and description network used in the self-supervised learning architecture is based on the work [16] and its parameters are listed in Table 1. This network consists of a backbone and three output heads. The RGB image is firstly fed into the backbone to generate a small feature map with a size of only 1/8 of the input image. The feature map is further processed by the subsequent heads to output three tensors with the same size, each in the representation of scores, relative positions, and descriptors of keypoints, respectively. As can be seen, each score, relative position, and descriptor in the output corresponds to an 8×8 region of the input image.

Since the visual features are nonuniformly scaled due to the distortion on the fisheye image, it will be inappropriate to apply the same convolutions on different image regions. Therefore, we apply the deformable convolution in the keypoint network based on the fact that it has a stronger adaptability than ordinary convolution to complex geometric deformation. Specifically, in the convolutional layers of both backbone and output heads, we adopt the deformable convolution so that the model can better learn the features in the distorted image.

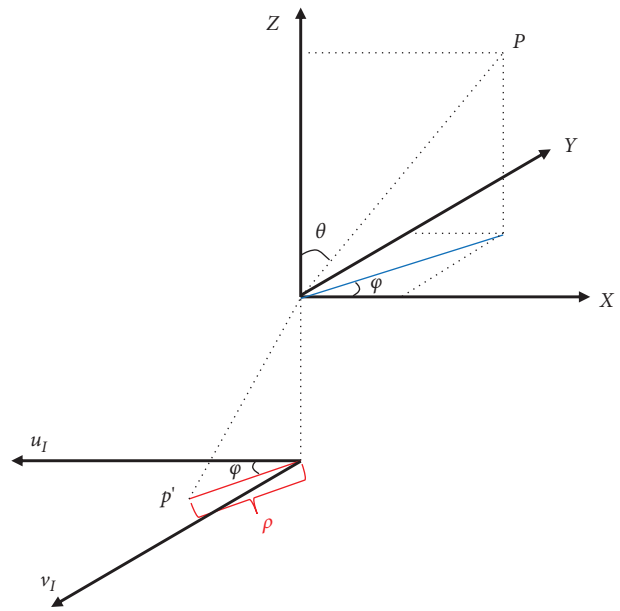

Figure 5: Fisheye camera projection model.

Additionally, for each convolutional layer, the stride is set to 1 and the kernel size equals 3. All convolutional layers are followed by batch normalization and an activation function of Leaky ReLU, except the last layer in each head.

3.3.2. Learning Loss. The learning loss considers the similarity of corresponding points on their positions, scores, and descriptors. Simultaneously, it encourages the spatially uniform distribution, repeatability of feature points, and decorrelation between nonidentical point descriptors, similar to [25]. The total loss can be decomposed into four parts: the self-supervised loss L_{ssp}, the uniform position distribution loss L_{uni}, the descriptor correspondence loss L_{desc}, and the descriptor decorrelation loss L_{decor}, interpreted as follows:

$$L = \alpha_{\text{ssp}} L_{\text{ssp}} + \alpha_{\text{uni}} L_{\text{uni}} + \alpha_{\text{desc}} L_{\text{desc}} + \alpha_{\text{decor}} L_{\text{decor}}, \tag{7}$$

where $\alpha_{\text{ssp/uni/desc/decor}}$ indicates the corresponding weight.

The self-supervised loss L_{ssp} can be further interpreted as follows:

$$L_{\text{ssp}} = \alpha_{\text{pos}} L_{\text{pos}} + \alpha_{\text{score}} L_{\text{score}} + \alpha_{\text{rep}} L_{\text{rep}}, \tag{8}$$

where the position loss L_{pos} is designed to minimize the Euclidean distance of paired points, thus ensuring that each pair corresponds to the same point in the original image. The score loss L_{score} is to ensure an identical score prediction for point pairs, specifically by minimizing the squared score

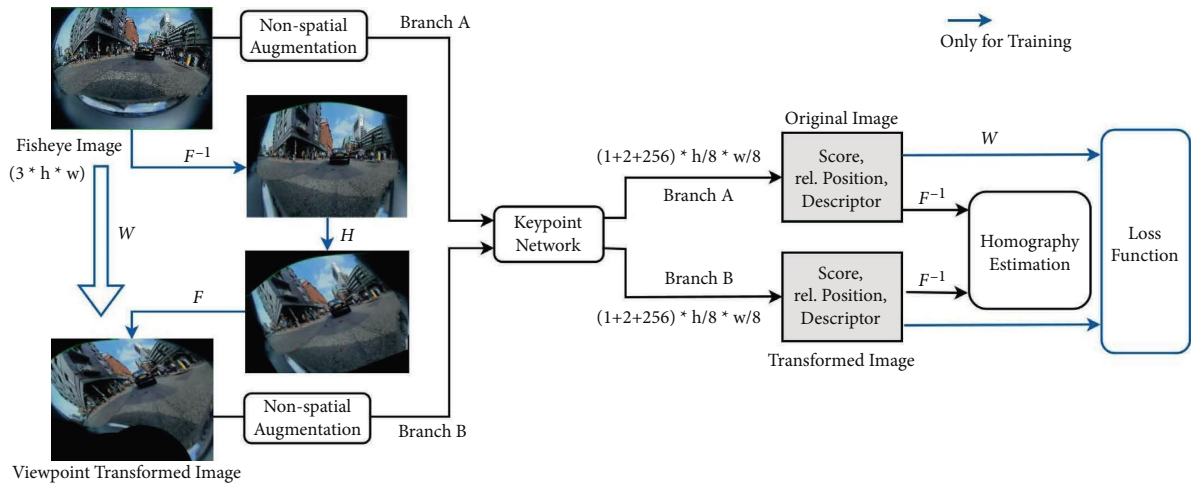

Figure 6: Overview of proposed self-learning architecture. The source fisheye image is firstly transformed into a viewpoint-changed version by undistortion, homography transformation, and warping, respectively. The keypoint network is applied on both source and transformed fisheye images to detect keypoints, interpreted by scores, relative positions, and descriptors. Based on the matching of keypoints, the homography transform between two fisheye images is further estimated and the losses are calculated (during training).

Table 1: Parameters of the keypoint network. "DConv" denotes the deformable convolution. All convolutional layers are followed by batch normalization and an activation function of leaky ReLU, except the last layer in each head.

	Module (kernel size)	Channel (in, out)	Stride
Backbone	2 × DConv (3 × 3)	(3, 32)	1
	1 × MaxPool (3 × 3)	(32, 32)	2
	2 × DConv (3 × 3)	(32, 64)	1
	1 × MaxPool (3 × 3)	(64, 64)	2
	2 × DConv (3 × 3)	(64, 128)	1
	1 × MaxPool (3 × 3)	(128, 128)	2
	2 × DConv (3 × 3)	(128, 256)	1
Head 1	1 × DConv (3 × 3)	(256, 128)	1
	1 × DConv (3 × 3)	(128, 256)	1
	1 × DConv (3 × 3)	(256, 1)	1
	1 × sigmoid	(1, 1)	1
Head 2	1 × DConv (3 × 3)	(128, 256)	1
	1 × DConv (3 × 3)	(256, 2)	1
	1 × sigmoid	(2, 2)	1
Head 3	1 × DConv (3 × 3)	(128, 256)	1
	1 × DConv (3 × 3)	(256, 256)	1

difference. The repeatability loss L_{rep} is to ensure that paired points with a close distance have a higher score, while pairs of far away points have a lower score. Given the predicted scores s_A and s_B by the twin branches A and B of the Siamese learning architecture for the i-th point pair, the loss L_{rep} can be calculated as follows:

$$L_{\text{rep}} = \sum_i \frac{s_A + s_B}{2}\left(d_i - \overline{d}\right), \tag{9}$$

where d_i indicates the distance between the i-th paired points, while $\overline{d}$ represents the mean distance of all point pairs.

The loss L_{uni} is to ensure a uniform distribution of predicted keypoints within the grid, rather than concentrating on the grid boundary. Thus, it is represented by summed differences between the distribution of predicted point coordinates and a uniform distribution. The loss L_{decor} aims to improve the compactness of descriptor by minimizing the correlation coefficients between nonidentical point descriptors within the same Siamese branch. The detailed calculation for L_{rep} and L_{decor} can be referred to [25].

Since the spatial relationship of feature point pairs is described by the complex mapping $\mathscr{W}$, the descriptor correspondence cannot be measured by linear operations. Inspired by the recent progress in contrastive learning of visual representation [30], we reinterpret the loss L_{des} as follows:

$$L_{\text{des}} = \sum_i -\log \frac{\exp\left(\text{sim}\left(\mathbf{f}_i^A, \mathbf{f}_j^B\right)\right)}{\sum_k 1_{[k!=j]} \exp\left(\text{sim}\left(\mathbf{f}_i^A, \mathbf{f}_k^B\right)\right)}, \tag{10}$$

with

$$\text{sim}\left(\mathbf{f}_i, \mathbf{f}_j\right) = \frac{\mathbf{f}_i^\top \cdot \mathbf{f}_j}{\tau \left\|\mathbf{f}_i\right\| \cdot \left\|\mathbf{f}_j\right\|}, \tag{11}$$

where $\mathbf{f}_i^A$ and $\mathbf{f}_j^B$ denote the i-th and j-th descriptor predicted by branch A and B, respectively. Here, $(\mathbf{f}_i^A, \mathbf{f}_j^B)$ is considered as a positive pair. The one-indicator $1_{[k!=j]}$ is only valid when k is not equal to j. Since there are 8×8 keypoints predicted for each image, a keypoint i on source image can only match one keypoint j on target image, while the rest 63 keypoints are considered as negatives for i. Thus, it ensures a nonzero denominator. The temperature τ is a hyperparameter, with a small value to reduce the impact of hard negative samples during the descriptor learning.

4. Experiment and Analysis

4.1. Experimental Setup

4.1.1. Dataset. The proposed self-learning architecture for keypoint detection and matching is evaluated on the released FV set of the WoodScape fisheye data [29], which consists of

2037 training images and 442 test images collected by the fisheye camera installed on one vehicle. The camera's intrinsic and extrinsic parameters are also calibrated. Therefore, fisheye images can be undistorted through the image unwarping process introduced in Sec. 3.2. On the WoodScape dataset, the polynomial $\rho(\theta)$ in equation (10) is set with an order of $n = 4$ with given coefficients $a_1 \sim a_4$.

4.1.2. Implementation. The proposed self-learning architecture is implemented with PyTorch on a desktop with an Intel Xeon CPU of 2.5 GHz and an Nvidia 2080Ti GPU. The network is pretrained on the ordinary images in MS COCO dataset [31] and further trained on the WoodScape fisheye images. During the pretraining, ordinary homography transforms are utilized to generate paired images. In further training, a random mapping $\mathscr{W}$ is applied for target fisheye image generation. The involved homography transform in mapping $\mathscr{W}$ consists of scaling, rotation, and perspective transform, which are uniformly sampled with a margin of 0.1, $\pi/2$, and 0.1, respectively. The weights for loss terms are empirically set to $\alpha_{\text{rep}} = 1$, $\alpha_{\text{pos}} = 1$, $\alpha_{\text{score}} = 2$, $\alpha_{\text{uni}} = 100$, $\alpha_{\text{des}} = 0.001$, and $\alpha_{\text{de cor}} = 0.03$. We adopt the ADAM as the optimizer. The whole model is trained for ten epochs with data shuffling, a batch size of 16, and a learning rate of 0.000025. All images are resized to a uniform size of 240×320 pixels for processing efficiency.

4.1.3. Metrics. The evaluation metrics adopted in experiments include the repeatability score (RS), the localization error (LE), the matching score (MS), and the homography accuracy (HA). The RS metric denotes the ratio between the number of points with correspondence and the total number of predicted points. A correspondence is established if points predicted from both images are located within the threshold $\varepsilon = 3$ by warping them into the same image plane. The LE metric is the mean distance between all matched point pairs according to the descriptors. The MS denotes the ratio between the number of good matches and the total number of points predicted in one image. A good match is defined as two corresponding points, which are also the nearest neighbors in descriptor space. To calculate HA, a source fisheye image is firstly unwarped by $\mathscr{F}^{-1}$. The average distance between the image corners transformed by the estimated homography, and those transformed by the ground truth homography is calculated and defined as Homography error (HE). The HA is the ratio between the number of estimated homographies under a specified HE threshold ($\varepsilon = 3$) and the total number of homographies.

4.2. Exploration on Hyperparameter τ. The temperature parameter τ has a large impact on the descriptor correspondence loss L_{des}. For hard negative samples, which can be easily classified as false positives, a smaller τ will reduce their weight during the learning. However, with an inappropriate small τ, true positives initialized with faraway positions can be neglected at the beginning of the training. To search for an appropriate temperature parameter, we train the network with different values of τ, and compare their test performance. The experimental results are reported in Table 2. As can be seen, with the setting of $\tau = 0.05$, the network achieves the best performance in terms of all metrics. Thus, we choose $\tau = 0.05$ as the optimal temperature parameter used in subsequent experiments.

4.3. Ablation Study on Model Setup. To verify the benefit of viewpoint transform (VT), deformable convolution (DC), and contrastive learning loss (CL), we conduct ablation studies on four different setups of the proposed network. The baseline (*B*) adopted in the experiment is the naive approach from work [25].

Test results are reported in Table 3. Obviously, by directly applying the baseline on fisheye images without viewpoint transform, the mean location error of corresponding points is relatively high, which is about 5 pixels and exceeds the default correspondence threshold ($\varepsilon = 3$). Integrated with the viewpoint transform of fisheye images, the mean location error is reduced by about 2 pixels. The contrastive learning loss further yields a promotion on other metrics within the range of 0.18 to 0.24. With all setups, the proposed architecture achieves the best performance in terms of all metrics, demonstrating their improvements over the baseline.

4.4. Comparison with Nonlearning-Based Approaches. Here, we compare our architecture with other nonlearning-based keypoint approaches including SIFT, SURF, ORB, BRISK, KAZE, and AKAZE. Evaluation metrics are the same as in previous experiments. For SIFT, SURF, ORB, BRISK KAZE, and AKAZE, we directly use their implementation provided by OpenCV. To explore the performance of compared approaches under different challenging scenarios, we also add the following preprocessing operations to test images, respectively.

(i) Contrast change: random change in image brightness, saturation, and hue with up to 40%, 40%, and 20%, respectively

(ii) Motion blur: blur filtering with a random filter size of up to 15 pixels

(iii) Random noise: Gaussian noise with a variance randomly sampled from 30 to 70

For fairness, the viewpoint transform applied on one test image is the same across all scenarios. Test results are reported in Tables 4–6, respectively.

From the experimental results, it is obvious that our proposed approach achieves the best matching score and homography accuracy in scenarios with contrast change and motion blur. It also achieves comparable results with the top-ranked ORB and BRISK in terms of location error and repeatability score metrics. Additionally, it can be seen that the repeatability of the proposed approach is relative sensitive to noise. We assume that the image noise affects the keypoint selection in the proposed approach to some extent. However, it still achieves the second best on the metric of homography accuracy and matching score, only with minor

Table 2: Test results of networks trained by a different temperature parameter τ. An up-arrow indicates that higher values are better. The best values are denoted in bold.

Temperature	RS ↑	LE ↓	HA ↑	MS ↑
$\tau = 0.03$	0.33	2.76	0.39	0.36
$\tau = 0.05$	**0.35**	**2.73**	**0.42**	**0.39**
$\tau = 0.1$	0.31	2.75	0.37	0.35
$\tau = 0.5$	0.26	2.79	0.31	0.24

Table 3: Ablation study on different configurations of the proposed approach. The superscript * denotes that the results are obtained at a threshold of 5 pixels. In the naive baseline, a hinge loss is adopted instead of the contrastive learning loss to learn descriptor correspondence. The best values are denoted in bold.

B	VT	CL	DC	RS ↑	LE ↓	HA ↑	MS ↑
✓				—	4.98*	—	—
✓	✓			0.17	2.83	0.24	0.15
✓	✓	✓		0.35	2.73	0.37	0.39
✓	✓	✓	✓	**0.41**	**2.53**	**0.41**	**0.43**

Table 4: Test of keypoint models under contrast change. Best and second best are denoted in bold and italics.

Model	RS ↑	LE ↓	HA ↑	MS ↑
SIFT	0.40	3.57	0.24	0.32
SURF	0.39	3.78	*0.26*	*0.34*
ORB	*0.45*	**2.46**	0.17	0.16
BRISK	**0.48**	2.98	0.20	0.27
KAZE	0.37	2.79	0.27	0.35
AKAZE	0.35	2.93	0.19	0.20
Ours	0.43	*2.59*	**0.38**	**0.39**

Table 5: Test of keypoint models under motion blur. Best and second best are denoted in bold and italics.

Model	RS ↑	LE ↓	HA ↑	MS ↑
SIFT	0.42	3.61	0.23	*0.40*
SURF	0.43	3.85	*0.25*	0.39
ORB	0.40	**2.56**	0.09	0.09
BRISK	*0.44*	2.69	0.12	0.17
KAZE	0.36	2.98	0.22	0.29
AKAZE	0.33	3.03	0.10	0.14
Ours	**0.45**	*2.68*	**0.34**	**0.41**

Table 6: Test of keypoint models under noise. Best and second best are denoted in bold and italics.

Model	RS ↑	LE ↓	HA ↑	MS ↑
SIFT	*0.41*	3.49	**0.37**	**0.39**
SURF	0.39	3.77	0.31	0.31
ORB	0.40	**2.55**	0.15	0.11
BRISK	**0.43**	2.88	0.19	0.15
KAZE	0.34	2.65	0.23	0.24
AKAZE	0.28	2.78	0.11	0.17
Ours	0.33	*2.61*	*0.33*	*0.32*

gaps to the top-ranked SIFT. It is also noted that the proposed approach achieves a much smaller location error (second best) than SIFT. Test examples in different scenarios are shown in Figure 7. Considering the comprehensive performance, the proposed approach shows a relatively high robustness against contrast change, motion blur, and noise.

Furthermore, we present the feature detection and description time of evaluated keypoint models in Table 7. As

Figure 7: Examples of qualitative results in scenarios of contrast change (1st column), motion blur (2nd column), and noise (3rd column). Correct matches are linked by green lines while false matches are in red.

Table 7: Feature detection and description time of compared keypoint models. Superscript * denotes utilization of GPU.

Model	SIFT	SURF	ORB	BRISK	KAZE	AKAZE	Ours*
Time (s)	0.2	0.18	0.06	0.27	0.4	0.072	**0.022**

can be seen, the ORB approach is the fastest among all handcrafted keypoint models, only requiring 0.06 second to process one frame. By running on the GPU platform, our proposed approach is also able to run in real time, with only 0.022 second per frame. Also, we calculate the value of FLOPs (floating point operations) and the number of parameters of our network, which are 7.4 G and 3.7 M, respectively, implying that our network is a relatively lightweight model.

5. Conclusions and Future Work

In this work, we propose a self-supervised learning architecture to address the challenging task of keypoint detection and matching on fisheye images. By integrating the viewpoint transform pipeline, the deformable convolution, and the contrastive learning loss, our method outperforms the baseline by a large margin. Through extensive experiments on challenging scenarios such as contrast change, motion blur, and noise, the comprehensive performance of the proposed approach is also demonstrated robust in terms of location error, homography accuracy, and matching score, compared to handcrafted models. As a direction of our future researches, we tend to integrate a more accurate and learnable undistortion scheme, which is free from the dependence on camera calibration parameters. Another direction is to include the multiscale image features to further improve the performance of the proposed approach.

Data Availability

All the data are available in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project was supported by the Shanghai Science and Technology Commission (No. 21ZR1467400) and the original research project of Tongji University (No. 22120220593).

References

[1] M. Jin, "Motion recognition based on deep learning and human joint points," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–10, Article ID 1826951, 2022.

[2] C. Flores-Munguía, J. C. Ortiz-Bayliss, and H. Terashima-Marín, "Leveraging a neuroevolutionary approach for classifying violent behavior in video," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1279945, 14 pages, 2022.

[3] C. Liu and J. Yao, "The robust semantic slam system for texture-less underground parking lot," *Journal of Advanced Transportation*, vol. 2022, Article ID 9681455, 11 pages, 2022.

[4] J. Wang, F. Shi, J. Zhang, and Y. Liu, "A new calibration model of camera lens distortion," *Pattern Recognition*, vol. 41, no. 2, pp. 607–615, 2008.

[5] V. Ravi Kumar, C. Eising, C. Witt, and S. Yogamani, "Surround-view fisheye camera perception for automated driving: overview, survey and challenges," 2022, https://arxiv.org/abs/2205.13281arXiv preprint arXiv:2205.13281.

[6] C. Yiakoumettis, N. Doulamis, G. Miaoulis, and D. Ghazanfarpour, "Active learning of user's preferences estimation towards a personalized 3d navigation of geo-referenced scenes," *GeoInformatica*, vol. 18, no. 1, pp. 27–62, 2014.

[7] G. Kim and J. Youn, "Production of a tunnel mosaic image using fisheye lens camera," *Sensors and Materials*, vol. 31, no. 10, pp. 3245–3259, 2019.

[8] C. Luo, L. Yu, J. Yan et al., "Autonomous detection of damage to multiple steel surfaces from 360 panoramas using deep neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 12, pp. 1585–1599, 2021.

[9] E. Protopapadakis, C. Stentoumis, N. Doulamis et al., "Autonomous robotic inspection in tunnels," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III-5, pp. 167–174, 06 2016.

[10] M. Lee, H. Kim, and J. Paik, "Correction of barrel distortion in fisheye lens images using image-based estimation of distortion parameters," *IEEE Access*, vol. 7, pp. 45723–45733, 2019.

[11] X. Wang, H. Bai, F. Wu, and X. Ye, "Fisheye lens distortion correction method based on improved spherical perspective projection," *Journal of Graphics*, vol. 39, no. 1, p. 43, 2018.

[12] M. Versaci, S. Calcagno, and F. Carlo Morabito, "Image contrast enhancement by distances among points in fuzzy hyper-cubes," in *Proceedings of the Computer Analysis of Images and Patterns*, pp. 494–505, Seville, Spain, August 2015.

[13] R. Manickavasagam and S. Selvan, "Automatic detection and classification of lung nodules in ct image using optimized neuro fuzzy classifier with cuckoo search algorithm," *Journal of Medical Systems*, vol. 43, no. 3, p. 77, 2019.

[14] E. Rosten and T. Drummond, "Machine learning for high speed corner detection," in *European Conference on Computer Vision*, vol. 1, pp. 430–443, ECCV, 2006.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] B. Herbert, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *Proceedings of the 2006 European Conference on Computer Vision (ECCV)*, pp. 404–417, Graz, Austria, May 2006.

[17] E. Rublee, R. Vincent, K. Kurt, and B. Gary, "Orb: an efficient alternative to sift or surf," in *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 2564–2571, Barcelona, November 2011.

[18] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Proceedings of the 2012 European Conference on Computer Vision (ECCV)*, Florence Italy, October 2012.

[19] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proceedings of the 2013 British Machine Vision Conference (BMVC)*, Bristol UK, September 2013.

[20] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proceedings of the 2015 IEEE international conference on computer vision*, pp. 118–126, Santiago, Chile, December 2015.

[21] Y. Kwang Moo, E. Trulls, V. Lepetit, and F. Pascal, "Lift: learned invariant feature transform," in *Proceedings of the 2016 European conference on computer vision (ECCV)*, pp. 467–483, Amsterdam, The Netherlands, October 2016.

[22] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys, "Quad-networks: unsupervised learning to rank for interest point detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1822–1830, Honolulu, Hawaii, July 2017.

[23] V. Lepetit, "Tilde: a temporally invariant learned detector," in *Procedings of th e 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–10, Boston, MA, USA, June 2015.

[24] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) workshops*, pp. 224–236, New Orleans, Louisiana, June 2018.

[25] C. Peter Hviid, K. Mikkel Fly, Y. Brodskiy, and H. Karstoft, "Unsuperpoint: end-to-end unsupervised interest point detector and descriptor," 2019, https://patrick-llgc.github.io/Learning-Deep-Learning/paper_notes/unsuperpoint. htmlarXiv preprint arXiv:1907.04011.

[26] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[27] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1309–1321, 2007.

[28] X. Wang, W. Feng, Q. Liu, B. Zhang, and Z. Cao, "Calibration research on fish-eye lens," *IEEE International Conference on Information and Automation*, pp. 385–390, 2010.

[29] S. Yogamani, C. Hughes, J. Horgan et al., "Woodscape: a multi-task, multi-camera fisheye dataset for autonomous driving," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), November 2019.

[30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 2020 International Conference on Machine Learning*, pp. 1597–1607, Vienna, Austria, July 2020.

[31] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft coo: common objects in context," in *Proceedings of the 2014 European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, September 2014.