This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PanoPoint: Self-Supervised Feature Points Detection and Description for 360° Panorama

Hengzhi Zhang[1], Hong Yi[1], Haijing Jia[1], Wei Wang[1], Makoto Odamaki[2]
[1]Ricoh Software Research Center (Beijing) Co., Ltd. Beijing, China
[2]Ricoh Company, Ltd. Japan
{Hengzhi.Zhang, Hong.Yi, Haijing.Jia, Wei.Wang5}@cn.ricoh.com, makoto.odamaki@jp.ricoh.com

Abstract

We introduce PanoPoint, the joint feature point detection and description applied to the nonlinear distortions and the multi-view geometry problems between 360° panoramas. Our fully convolutional model operates directly in panoramas and computes pixel-level feature point locations and associated descriptors in a single forward pass rather than performing image preprocessing (e.g. panorama to Cubemap) followed by feature detection and description. To train the PanoPoint model, we propose PanoMotion, which simulates the representation between different viewpoints and generates warped panoramas. Moreover, we propose PanoMotion Adaptation, a multi-viewpoint adaptation annotation approach for boosting feature point detection repeatability instead of manual labelling. We train on the annotated synthetic dataset generated by the above method, which outperforms the traditional and other learned approaches and achieves state-of-the-art results on repeatability, localization accuracy, point correspondence precision and real-time metrics, especially for panoramas with significant viewpoint and illumination changes.

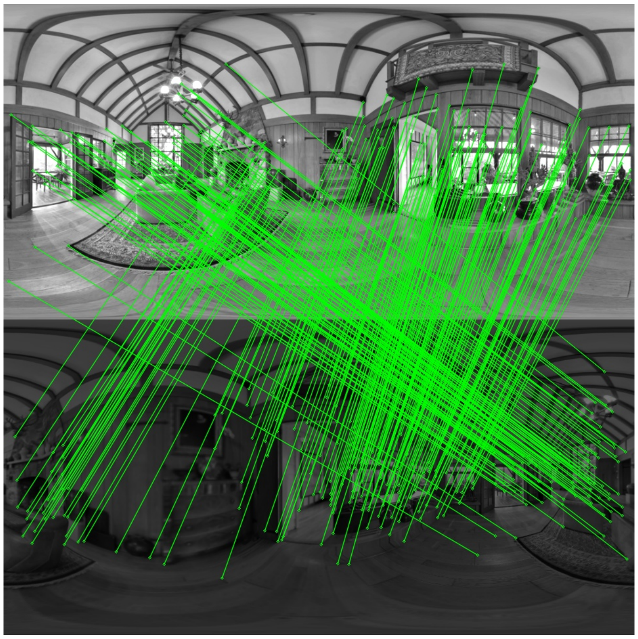


Figure 1. **Point Correspondences by PanoPoint.** We present a fully-convolutional neural network that computes feature point locations and associated descriptors in a single deep network, even under significant viewpoint and illumination changes.

1. Introduction

Extracting feature points from images captured under different illumination and viewpoints has attracted the attention of computer vision and graphics researchers in the last decade. Feature points detection and description is a critical technology in many applications, such as robot navigation [35], augmented and virtual reality (AR/VR) [20], Visual Simultaneous Localization and Mapping (VSLAM) [21], and Structure-from-Motion (SFM) [41].

In the past decade, many data-driven learning-based feature points detection and description methods have emerged [6,9,10,36,48–50], replacing the traditional manual annotation methods [1, 26, 37]. These methods generally use planar images captured by traditional cameras. The limited field of vision of traditional cameras leads to the need for more feature points. The omnidirectional view 360° camera lets capture the entire scene simultaneously, rather than taking multiple shots from different angles [17,54,56]. Nevertheless, there are severe geometric distortions of objects near the spherical projection poles in panoramas captured by 360° cameras [24]. Therefore, feature point detection and description in panoramas is a great challenge.

This work proposes a novel self-supervised learning network that addresses the challenge of training on panorama datasets without explicit annotations. Our approach leverages a synthetic dataset of simple geometric shapes to generate pseudo-ground truth feature points. We employ a con-

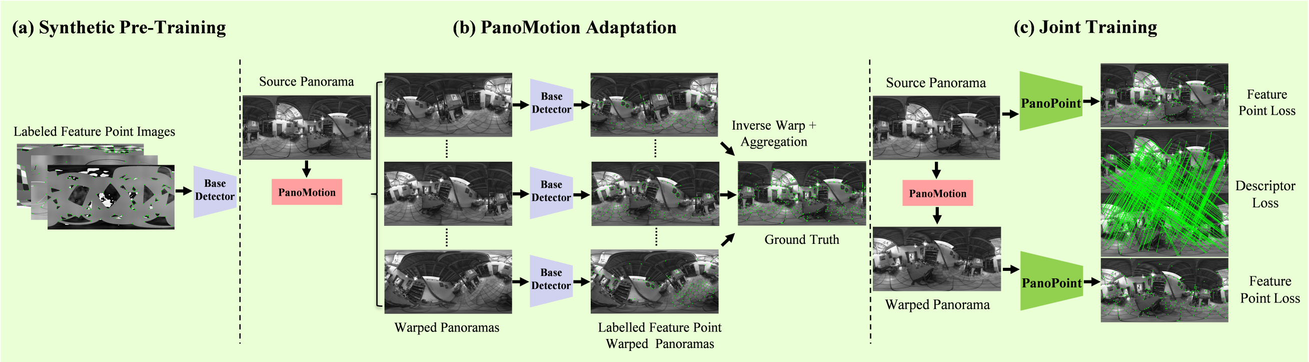


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, (a) our detector network is pre-trained on a synthetic dataset with known ground truth and performs (b) PanoMotion Adaptation to generate pseudo-ground truth for feature points. Finally, the generated pseudo ground truth (c) trains a fully convolutional network to jointly extract feature points and descriptors on panoramas.

volutional neural network Base Detector trained on the synthetic dataset to achieve this. However, the Base Detector could miss potential feature point locations. To mitigate this issue, we propose a PanoMotion Adaptation approach (see Section 3.3) which facilitates self-supervised training (see Figure 2) of PanoPoint, which is a feature point detection and description model. Figure 1 shows that the feature points extracted using PanoPoint have strong repeatability and perfect description even under significant viewpoint and illumination changes. Unlike the Base Detector, the PanoPoint comprises two sub-networks: the detection head and the description head (see Section 4.2). Additionally, we propose PanoMotion (see Section 3.2), a representation that simulates camera viewpoint changes by repeatedly warping the source panorama. Through PanoMotion, we can generate pseudo-ground truth feature points by detecting feature point locations in each warped panorama using the Base Detector and inverse-warping these locations reproject to the source panorama. Further, we can also obtain pseudo-ground truth feature points for each warped panorama through PanoMotion. For training, we input both the source and warped panoramas into the PanoPoint and compare them with their corresponding pseudo-ground truth feature point locations. Concurrently, we compare the descriptors generated by PanoPoint at the corresponding points on the source and warped panorama. We create the world's first dense point-to-point annotated panorama dataset based on the approaches above. Overall, our contributions can be summarized as follows:

1) We propose PanoPoint, to compute pixel-level feature point locations and associated descriptors in panoramas in a single forward pass, even under significant viewpoint and illumination changes.
2) We propose PanoMotion, a new representation approach that simulates point-to-point correspondences between different viewpoints and generates warped panoramas.
3) We create the world's first dense point-to-point annotated panorama dataset by PanoMotion Adaptation, a multi-viewpoint adaptation approach for boosting feature point detection repeatability instead of manual labelling. The PanoPoint can repeatedly detect rich feature points (see Section 6.2) when trained on the dataset using PanoMotion Adaptation.

2. Related Works

Feature point detection and description on the planar images generally adopt local-based methods to overcome the interference caused by global factors (*e.g.*, lighting, rotation, noise). Descriptors represent the local features of feature points. The same feature points should be repeatedly detected in different viewpoints and have similar descriptors. On the contrary, the difference between feature points and non-feature point descriptors should be significant enough. Traditional detection methods [1, 26, 37] often detect features based on their gradient or intensity information. However, the robustness of the traditional methods in complex scenarios is not good enough [18]. In the past few years, researchers have used learning-based methods [6, 9, 10, 36, 48, 50] to extract feature points and generate descriptors. For example, Tian *et al.* [50] propose a fully convolutional structure to generate dense descriptors. Dusmanu *et al.* [10] uses the response value of the descriptor feature vector to find feature points. DeTone *et al.* [9] propose a self-supervised learning method to extract feature points and descriptions at the same time. Revaud *et al.* [36] believe that model learning must make feature points repeatable and the descriptions distinguishable. Recently, Christiansen *et al.* [6] used a regression method to predict locations. Based on [6], Tang *et al.* [48] generates optimal inlier sets from possible corresponding point-pairs using a neurally-guided outlier-rejection scheme. However, these methods are all designed for planar images, which are unsatisfactory for panoramas. There are a few researches on feature point detection and description of panoramas [11,57,58]. Generally, to solve the panorama distortion, the researcher converts the panorama to the planar image through the cube projection [58] and performs feature points detection and description on the planar image. In addition, some research [11,57] has focused on subdivided icosahedral sampling methods, which can effectively alleviate panoramic distortion. For example, Eder *et al*. [11] proposes tangent images, which render images onto an oriented pixel grid tangent to a subdivided icosahedron. Although the above methods mitigate nonlinear distortion, they require additional processing. We design a novel end-to-end network model to predict the locations of feature points and descriptors in the panorama without additional processing.

Labelling point correspondences in panoramas is a critical issue in learning-based feature point detection and description methods. Some researchers [12,40] address this issue by estimating the homography using a pair of corresponding planar regions extracted from the panoramas. Then, they use traditional feature point detection methods to detect feature points in one panorama and use the estimated homography [16, 22, 31, 44, 45] to calculate corresponding feature points in another panorama. However, homography estimation requires a set of correct matching point pairs, thus depending on matching accuracy. Our work avoids the above problems, and we design a method to simulate camera motion and imaging, which can correctly generate corresponding points without estimating homography.

3. Self-Supervised Training Pipeline

We propose a unified network for feature point detection and description in panoramas. Our detector is first pre-trained on a synthetic dataset with known ground truth. Then the whole detection and description heads are trained using the pre-trained model to generate pseudo-ground truth feature points on real panoramas. In the following sections, we explain the details of our training pipeline in Figure 2 and detail its parts.

3.1. Synthetic Pre-Training

There is currently no annotation feature point dataset for panoramas. If the feature points are labelled manually, not only the accuracy of the labelling cannot be guaranteed, and the labelling requires a lot of manpower and time. Inspired by [9], we used OpenCV [3] to create a large-scale synthetic panorama dataset. Various graphics are included in the datasets, such as checkerboards, line segments, polygons, cubes, etc., and the endpoints of these graphics are used as marked feature points. The placement of these graphics is random. Unlabeled ellipses and pure noise [5, 27] are also added to avoid ambiguity. The difference from [9] is that we first draw graphics on six plane images and then use the projection relationship from cube to equirectangular [14, 39] to project these images onto the panorama and transform the coordinates of the feature points labelled on the planar images to the panorama. The generated synthetic panorama has the same distortion effect as an actual panorama, and the closer to the poles, the greater the degree of graphic distortion [55].

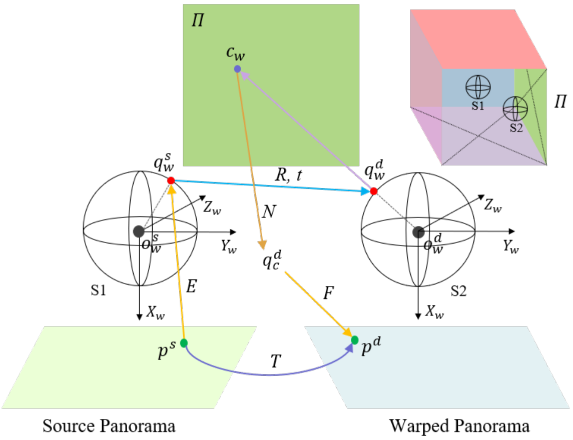
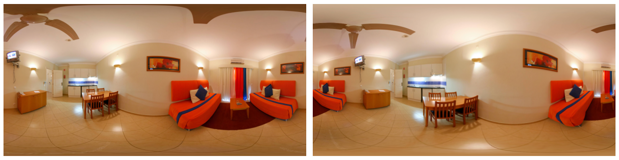


Figure 3. **PanoMotion Transformation.** Point-to-point correspondence transformation relationship between panoramas from different viewpoints.



(a) Source Panorama (b) Warped Panorama

Figure 4. **Panorama Generation by PanoMotion.** Input the source panorama, and given random (R, t), we generate the warped panorama with PanoMotion.

3.2. PanoMotion

Current public panorama datasets, such as Matterport3D [4] and SUN3D [53], do not provide point correspondences in panoramas taken from different viewpoints. We design a method for simulating camera motion and imaging called PanoMotion to solve this problem. The PanoMotion process only needs to input a source panorama and specify the viewpoint's camera pose (R, t) to generate a warped panorama. At the same time, a point-to-point correspondence between the source panorama and the warped panorama is also generated.

As shown in the upper right corner of Figure 3, we use the unit spheres S1 and S2 as the first and second viewpoints, respectively. The unit sphere S1 is located at the centre of the cube, which is also the centre of the world coordinate system. Each point on the source panorama can establish a point-to-point connection through the equirectangular-to-cube transformation [42, 52]. Unlike the actual scene, each face of the cube has the same depth. The following describes the process of generating the warped panorama and coordinate transformation.

Figure 3 shows the pixel point p^s on the source panorama. According to the mapping relationship E from the equirectangular to the unit sphere S1, the projection point

$$q_w^s = E(p^s) \tag{1}$$

corresponding to p^s on the unit sphere S1 is calculated. Where q_w^s is a point on the unit sphere S1.

The pose of the unit sphere S2 relative to S1 is (R, t), so the position of q_w^s on S2 is

$$q_w^d = R \cdot q_w^s + t, \tag{2}$$

and the centre coordinate of S2 is

$$o_w^d = R \cdot o_w^s + t, \tag{3}$$

where o_w^s is (0, 0, 0), o_w^d is equal to t. The intersection point of the vector $\overrightarrow{o_w^d q_w^d}$ and the cube Π is expressed as

$$c_w = \overrightarrow{o_w^d q_w^d} \cap \Pi, \tag{4}$$

then normalize the c_w coordinates to the camera coordinate system

$$q_c^d = Norm(c_w), \tag{5}$$

calculate the point coordinate q_c^d in the camera coordinate system, transform F according to the projection from the unit sphere to the equirectangular, and convert q_c^d to p^d,

$$p^d = F(q_c^d). \tag{6}$$

Therefore, the position transformation of each point from the source to the warped panorama can be expressed as:

$$p^d = F(Norm(R \cdot E(p^s) \cap \Pi). \tag{7}$$

To simplify the above formula, the coordinate transform is denoted by T, and we get:

$$p^d = T \cdot p^s. \tag{8}$$

We map the pixels of the source panorama into an array of size $(H, W, 3)$ through T to generate the warped panorama (Figure 4 shows an example). T is only determined by the given (R, t). All pixel values of the generated warped panorama are from the source panorama and sampled using bilinear interpolation [23]. PanoMotion generates N different warped panoramas and corresponding coordinate relationships to make the labelled feature point.

3.3. PanoMotion Adaptation

To enable feature points to perceive a scene from different viewpoints and scales, we employ a technique known as PanoMotion Adaptation. This technique involves multiple warping of the source panorama, extraction of feature points using a Base Detector for each warped panorama, and transfer of the N sets of feature points through their corresponding inverse-warp aggregation to the source panorama (refer to Figure 2). Due to the possibility of detecting the exact feature point multiple times in different warped panoramas, the feature points on the image after conversion to the source panorama may cluster near the same point. To address this, we use non-maximum value suppression (NMS) [2] to retain only a single feature point within a specific range of 4 pixels.

3.4. Joint Training

PanoMotion warps the source panorama, and PanoPoint predicts the feature points. The predicted feature points and the ground truth are used to calculate the location loss (see Section 4.3). If the distance between the predicted feature points and the ground truth is less than 4 pixels, then the prediction of this point is considered correct. For the training of the description head, we directly use the feature point location of the ground truth of the source panorama, then calculate the corresponding position of the warped panorama, calculate the Euclidean distance between the descriptors of the corresponding points of the two panoramas, and continuously optimize the model to make the distance minimize.

4. PanoPoint Architecture

The PanoPoint network structure includes the backbone, detection head and description head (see Figure 5). The backbone outputs to extract high-dimensional features of the source panorama, the detection head outputs the location of feature points, and the description head outputs associated descriptors of feature points. Considering the inference time, we did not increase the model's convolutional layer or pooling layer for the sizeable polar distortion of the panorama. It can quickly and directly predict the location and descriptors of feature points, which is very practical for tasks that require high efficiencies like SFM and VSLAM.

4.1. Backbone

To preserve the original information of Panorama, we use the backbone ResNet [15] and a convolutional layer with a step size of 2 [46]. At the same time, to make the training easier to converge, the activation layer is composed of ELU [7] instead of RELU [13]. The backbone comprises of BL1 and BL2 (see Figure 6). BL1 has a convolution with a step size of 1, a BN, and an activation layer. The input and output of BL1 are skip-connect; BL2 is composed of

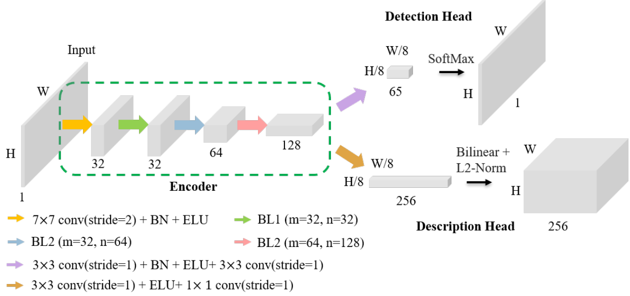
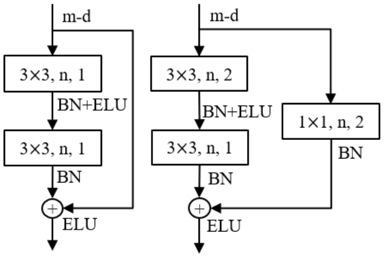


Figure 5. **PanoPoint Network Structure.** The proposed PanoPoint architecture utilizes a shared-encoder backbone with two output heads for feature point probabilities and descriptions.



(a) BL1

(b) BL2

Figure 6. **BL1 and BL2.** The main block of the backbone, input m-dimensional, convolution parameters (*convolution kernel size, number of channels, step size*)

a convolution with a step size of 2, a BN layer, and an activation layer, the input of BL2 is also connected with the output of BL2 through 1×1 convolution and a BN layer. The backbone downsamples the input image three times in total, the input tensor size is $W \times H$, and the output size tensor is $W/8 \times H/8$.

4.2. Detection and Description Heads

The detector head outputs the probability of whether each point on the panorama of size $W \times H$ is a feature point. The detector is output through two convolution layers to output a 65-dimensional vector, including an 8×8 pixel grid region and an extra "no feature point" dustbin [9]. The probability distribution of $W \times H$ is obtained after SoftMax [19] and Reshape functions.

The descriptor head uses a 256-dimensional vector to describe the feature. The feature shape is $\frac{w}{8} \times \frac{h}{8}$. After bilinear interpolation [43] and L2 normalization [8] is extended to $W \times H$, each pixel corresponds to one descriptor.

4.3. Loss Functions

The final loss is the sum of two sub-task losses: L_l for the feature point detector and L_d for learning descriptors. Each loss term is weighted by a factor ω to balance the final loss:

$$L_{total} = \omega_1 L_l + \omega_2 L_d. \tag{9}$$

The location loss function. Similar to [9], and the output is a coarse $\frac{h}{8} \times \frac{w}{8} \times 65$ feature map $\mathbf{v}_{pre}$, while each 65-dimensional vector corresponds to an 8×8 patch and an additional "no feature point" bin. We define the ground truth $v_{gt} \in \{1, \ldots, 65\}^{\frac{h}{8} \times \frac{w}{8}}$, denoting the index of the ground truth location in each patch, and the value of 65 means "no feature point". The location loss is a cross-entropy loss between $\mathbf{v}_{pre}$ and $\mathbf{v}_{gt}$:

$$L_l = \frac{64}{h \times w} \sum_{i,j=1}^{\frac{h}{8} \times \frac{w}{8}} - \log \left(\frac{\exp\left(v_{ijv_{ij}^{gt}}^{pre}\right)}{\sum_{k=1}^{65} \exp\left(v_{ijk}^{pre}\right)} \right) \tag{10}$$

The descriptor loss function. Following [9], We use a hinge loss with positive margin m_p and negative margin m_n. The hinge loss minimizes the descriptor distance for matching points and maximizes for non-matching points. The descriptor loss is defined as:

$$\begin{aligned} L_d = \sum_k (c_k \cdot max(0, m_p - {f_k^s}^T f_k^d) \\ + (1 - c_k) \cdot max(0, {f_k^s}^T f_k^d - m_n)) \end{aligned} \tag{11}$$

where ${f_k^s}^T$ and f_k^d are representation of p_k^s and p_k^d (see Section 3.2), respectively. If the Euclidean distance between Tp_k^s and p_k^d is less than a threshold γ, c_k is 1. Otherwise is 0.

5. Experimental Details

In this section, we introduce the details of training Base Detector and PanoPoint. The difference between Base Detector and the PanoPoint model is that there is no description head, which only predicts the feature points of the image.

In Section 3.1, we described the creation of the synthetic training dataset. The synthetic panorama created is the grayscale of 960 × 480. We have created nine graphics, including Lines, Polygon, Multiple Polygons, Star, Stripes, Ellipses, Checkerboards, and Cubes. Moreover, We add Gaussian Noise without the label, which helps improve the robustness of the Base Detector. Each class of graphics is randomly distributed on the image. We created 50,000 panoramas for each graphics class as a training dataset and 500 panoramas for each graphics class as an evaluation dataset.

We also use the RICOH THETA panoramic camera to shoot 7501 panoramas as a training dataset. These images include indoor and outdoor scenes. The images are 960×480 and converted to grayscale. We use PanoMotion Adaptation ($N = 20$) to generate feature points for the pseudo-ground truth. We generate distorted panoramas in real-time during the model training by adjusting R and t parameters. The given camera rotation param R, mentioned in Section 3.2, comprises rotating angles for Pitch-, Yaw- and Roll-axis. The value range is $[-pi, pi]$. The given camera translation param t comprises translating distance along the X-, Y-, and Z-axis. Their value range of each angle is $[-r, r]$, and we set $r = 6$ because the distance from S1 (see Section 3.2) to each side of the cube is 10. In addition, the value of R and t obey the uniform distribution.

We use ADAM [33] as the optimizer with its default parameters and an initial learning rate of 0.0001. We adopt the descriptor loss weights from [9] with a positive margin $m_p = 1$, a negative margin $m_n = 0.2$ and a balancing factor $d = 250$. The selected weight terms were $\omega_1 = 1$ and $\omega_2 = 5$.

6. Experiments

6.1. Metrics

The Repeatability Score (RS) measures the quality of feature points and is the ratio of the number of point correspondences found and the total number of feature points between a pair of images [6, 28]. Alternatively, to eliminate the influence caused by distortion, the evaluation is carried out on the sphere. The feature points detected on the source image are converted to the warped panorama by T (see Section 3.2) to get projection points. Calculate the angle between the obtained projection points and the feature points detected on the warped panorama. It is considered repeatable if feature points are within the predetermined threshold value of $\epsilon = 1.5$.

The Localization Error (LE) is the average angle between point correspondences [6]. Localization error measures the accuracy of feature points.

The Mean Average Precision (mAP) is the ratio of correct point correspondences to the total number of point correspondences [29, 34]. The number of point correspondences refers to the number of nearest neighbour corresponding points. Like repeatability, the distance between the correct match points must be less than the predetermined threshold value of ϵ.

The Frames Per Second (FPS) is the derivative of the run-time from inputting two images to getting correct point correspondences, which measures the performance of feature point detection and matching speed. The larger the FPS, the faster the running speed.

6.2. Detector Repeatability and Accuracy

We measure repeatability on the LayoutNet dataset [59]. The LayoutNet dataset contains panoramas collected in large indoor environments and open spaces like corridors. We select 1061 RGB panoramas as the evaluation dataset, and all images are resized to 960 × 480 resolution. Based on these images, we perform PanoMotion (see Section 3.2) and illumination, creating three data types, Viewpoint change, Illumination change and Both. Viewpoint change refers to the use of PanoMotion to simulate viewpoint change. The value of the parameter R of PanoMotion is in $[-pi, pi]$, and the value of t is in $[-6, 6]$. Their values are evenly distributed. Illumination change refers to only adjusting the brightness and contrast. We use the ColorJitter function of Pytorch [33] to change the illumination, where the parameters are $brightness = 0.8$ and $contrast = 0.5$. The Both data type means that both viewpoint and illumination are changed. Each data type includes the source panorama, the warped panorama and their point correspondences. Repeatability is computed at 960×480 resolution with 1000 points detected in each image. Feature points should have high repeatability and accuracy at the same time. We compute the repeatability and localization error on the sphere and use a correct distance of $\epsilon = 1.5°$, which is 4 pixels in the image coordinate system. We compare our method with SuperPoint (SP) [9], BRISK [25], ORB [37], D2-Net [10], and R2D2 [36] as shown in Table 1, the RS and LE are top two on all evaluation data types.

Method	Viewpoint		Illumination		Both	
	RS↑	LE↓	RS↑	LE↓	RS↑	LE↓
BRISK	0.388	0.619	0.354	0.523	0.301	0.586
ORB	**0.786**	**0.436**	0.822	**0.351**	0.670	**0.454**
SP	0.742	0.509	0.820	0.424	**0.683**	0.537
D2-Net	0.400	0.938	0.648	0.603	0.381	0.940
R2D2	0.704	0.545	**0.917**	**0.305**	0.681	0.554
Ours	**0.769**	**0.503**	**0.855**	0.401	**0.726**	**0.524**

Table 1. **Detector Repeatability and Accuracy.** PanoPoint has the highest repeatability at both viewpoint and illumination changes.

Method	Matcher	LayoutNet	CVPG	FPS
BRISK	NN	0.278	0.237	**34**
ORB	NN	0.484	0.477	**46**
SP	NN	**0.534**	0.589	18
D2-Net	NN	0.256	0.230	2
R2D2	NN	0.434	0.433	1
SP	SuperGlue	0.367	**0.616**	15
LoFTR	LoFTR	0.187	0.251	5
Ours	NN	**0.659**	**0.621**	**34**

Table 2. **Point Correspondences Precision and Runtime.** PanoPoint not only has the highest mAP but also has competitive execution speed.

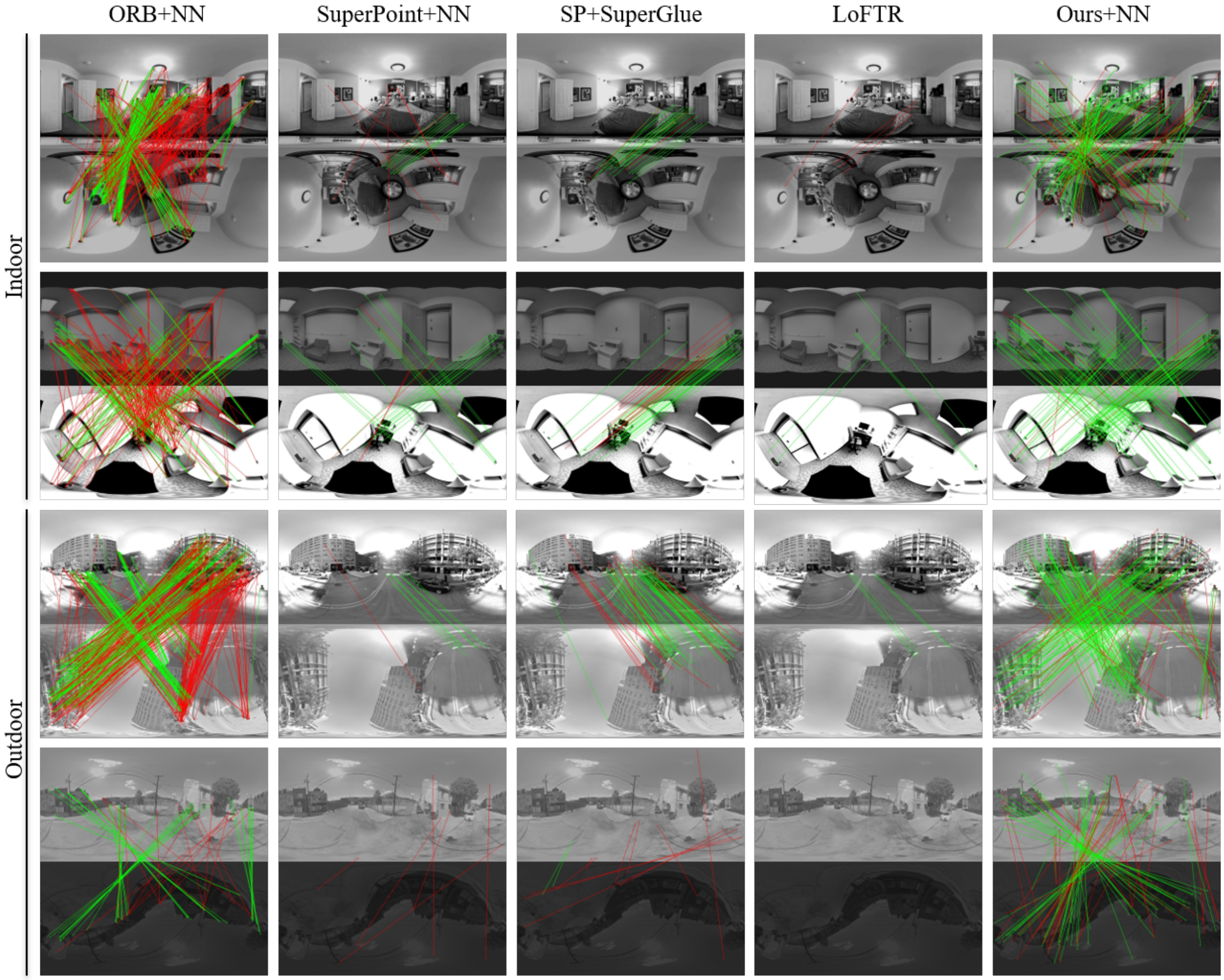


Figure 7. **Qualitative Results.** In indoor and outdoor environments, we compare PanoPoint to ORB, SuperPoint, and LoFTR with two outlier rejectors, handcrafted and learned. PanoPoint achieved more correct point correspondences (green line) and fewer false point correspondences (red line), successfully handling significant viewpoint and illumination changes.

6.3. Point Correspondence Precision

We measure the mAP of point correspondences on the LayoutNet dataset [59] and the CVPG dataset [32]. The CVPG dataset contains 600 outdoor panoramas with semantic pixel-level annotations. We resize the image to 960×480 resolution and then perform PanoMotion and illumination change. We evaluate the Both data type mentioned in the previous section on these two datasets and use the Nearest Neighbor (NN) matcher [30] to generate point correspondences with a correct distance of $\epsilon = 1.5°$. In addition to the methods mentioned in the previous section, we add two transformer-based matchers [47, 51]: the detect-based method SuperGlue and the detect-free method LoFTR. PanoPoint outperforms all published methods on indoor and outdoor datasets as shown in Table 2. Panopoint's descriptors are more robust to significant viewpoint and illumination changes. Even outlier rejectors like SuperGlue and LoFTR do not produce more correct point correspondences(see Figure 7).

6.4. Runtime

We measured the run time using the timing tool with Nvidia's CUDA deep learning library [38]. We start to calculate the time from inputting two 960×480 resolution panoramas, pass feature point detection and end the timing when point correspondences are generated. Except BRISK and ORB are evaluated on CPU (Intel i7-8700K), other methods are executed on GPU (GeForce GTX 1080Ti). The running time includes post-processing steps such as interpolation and point selection. PanoPoint is slightly slower than

ORB but much faster than the other deep learning methods.

7. Conclusion

This paper presents PanoPoint, a new learning framework to feature point detection and description for panoramas that can compute pixel-level feature point locations and associated descriptors with a single forward network. We propose a new representation approach PanoMotion, for training to simulate point-to-point correspondences between different viewpoints and generate warped panoramas. Based on the approach, we create the world's first dense point-to-point annotated panorama dataset by PanoMotion Adaptation. Our experiments show that PanoPoint achieves state-of-the-art performances on repeatability, localization accuracy, point correspondence precision and real-time metrics by using the dataset, especially for panoramas with significant viewpoint and illumination changes. Future work will investigate whether PanoMotion Adaptation can improve the performance of other applications, such as object detection and semantic segmentation. Finally, our PanoPoint network helps establish common-view relations in panoramic vision, such as SLAM and SFM. Combined with a deep back-end, PanoPoint is an essential milestone for the end-to-end panoramic slam.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 1, 2

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 4

[3] Gary Bradski, Adrian Kaehler, et al. Opencv. *Dr. Dobb's journal of software tools*, 3(2), 2000. 3

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 3

[5] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563, 2021. 3

[6] Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. *arXiv preprint arXiv:1907.04011*, 2019. 1, 2, 6

[7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. 4

[8] Zhuang Dai, Weinan Chen, Xinghong Huang, Birong Li, Lei Zhu, Li He, Yisheng Guan, and Hong Zhang. Cnn descriptor improvement based on l2-normalization and feature pooling for patch classification. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 144–149. IEEE, 2018. 5

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 1, 2, 3, 5, 6

[10] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019. 1, 2, 6

[11] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. Tangent images for mitigating spherical distortion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12426–12434, 2020. 3

[12] Robert Frohlich, Levente Tamás, and Zoltan Kato. Homography estimation between omnidirectional cameras without point correspondences. In *Handling Uncertainty and Networked Structure in Robot Control*, pages 129–151. Springer, 2015. 3

[13] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011. 4

[14] Seo Woo Han and Doug Young Suh. Piinet: A 360-degree panoramic image inpainting network using a cube map. *arXiv preprint arXiv:2010.16003*, 4, 2020. 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[16] Van-Dung Hoang, Diem-Phuc Tran, Nguyen Gia Nhu, Van-Huy Pham, et al. Deep feature extraction for panoramic image stitching. In *Asian Conference on Intelligent Information and Database Systems*, pages 141–151. Springer, 2020. 3

[17] Haijing Jia, Hong Yi, Hirochika Fujiki, Hengzhi Zhang, Wei Wang, and Makoto Odamaki. 3d room layout recovery generalizing across manhattan and non-manhattan worlds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5192–5201, 2022. 1

[18] Khushbu Joshi and Manish I Patel. Recent advances in local feature detector and descriptor: a literature survey. *International Journal of Multimedia Information Retrieval*, pages 1–17, 2020. 2

[19] Armand Joulin, Moustapha Cissé, David Grangier, Hervé Jégou, et al. Efficient softmax approximation for gpus. In *International conference on machine learning*, pages 1302–1310. PMLR, 2017. 5

[20] Timothy Jung and M Cluaudia tom Dieck. Augmented reality and virtual reality. *Empowering Human, Place and Business. Cham: Springer International Publishing*, 2018. 1

[21] Niklas Karlsson, Enrico Di Bernardo, Jim Ostrowski, Luis Goncalves, Paolo Pirjanian, and Mario E Munich. The vslam algorithm for robust localization and mapping. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 24–29. IEEE, 2005. 1

[22] Zoltan Kato, Gabor Nagy, Martin Humenberger, and Gabriela Csurka. Detecting low-rank regions in omnidirectional images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3682–3692, 2021. 3

[23] Anna Konrad, Ciarán Eising, Ganesh Sistu, John McDonald, Rudi Villing, and Senthil Yogamani. Fisheyesuperpoint: Keypoint detection and description network for fisheye images. *arXiv preprint arXiv:2103.00191*, 2021. 4

[24] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9181–9189, 2019. 1

[25] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011. 6

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 2

[27] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018. 3

[28] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I 7*, pages 128–142. Springer, 2002. 6

[29] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005. 6

[30] Marius Muja and David Lowe. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 5, 2009. 7

[31] Sooyol Ok and Suk-Hwan Lee. 360-degree panoramic video generation using virtual and actual multiple cameras. *Nonlinear Theory and Its Applications, IEICE*, 14(1):2–17, 2023. 3

[32] Semih Orhan and Yalin Bastanlar. Semantic segmentation of outdoor panoramic images. *Signal, Image and Video Processing*, 16(3):643–650, 2022. 7

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 6

[34] Rémi Pautrat, Juan-Ting Lin, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Sold2: Self-supervised occlusion-aware line description and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11368–11378, 2021. 6

[35] Luis Pérez, Íñigo Rodríguez, Nuria Rodríguez, Rubén Usamentiaga, and Daniel F García. Robot guidance using machine vision techniques in industrial environments: A comparative review. *Sensors*, 16(3):335, 2016. 1

[36] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019. 1, 2, 6

[37] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 1, 2, 6

[38] Jason Sanders and Edward Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010. 7

[39] Frode Eika Sandnes and Yo-Ping Huang. Translating the viewing position in single equirectangular panoramic images. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 000389–000394. IEEE, 2016. 3

[40] Mark Sastuba. Structureless camera motion estimation of unordered omnidirectional images. 3

[41] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[42] Zhijie Shen, Chunyu Lin, Lang Nie, Kang Liao, and Yao Zhao. Distortion-tolerant monocular depth estimation on omnidirectional images using dual-cubemap. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 4

[43] PR Smith. Bilinear interpolation of digital images. *Ultramicroscopy*, 6(2):201–204, 1981. 5

[44] Dae-Young Song, Geonsoo Lee, HeeKyung Lee, Gi-Mun Um, and Donghyeon Cho. Weakly-supervised stitching network for real-world panoramic image generation. In *European Conference on Computer Vision*, pages 54–71. Springer, 2022. 3

[45] Dae-Young Song, Gi-Mun Um, Hee Kyung Lee, and Donghyeon Cho. End-to-end image stitching network via multi-homography estimation. *IEEE Signal Processing Letters*, 28:763–767, 2021. 3

[46] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 4

[47] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 7

[48] Jiexiong Tang, Hanme Kim, Vitor Guizilini, Sudeep Pillai, and Rares Ambrus. Neural outlier rejection for self-supervised keypoint learning. *arXiv preprint arXiv:1912.10615*, 2019. 1, 2

[49] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2d: Keypoint extraction with describe to detect approach. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1

[50] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 661–669, 2017. 1, 2

[51] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2019. 7

[52] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 462–471, 2020. 4

[53] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013. 3

[54] Yasushi Yagi. Omnidirectional sensing and its applications. *IEICE transactions on information and systems*, 82(3):568–579, 1999. 1

[55] Wenyan Yang, Yanlin Qian, Joni-Kristian Kämäräinen, Francesco Cricri, and Lixin Fan. Object detection in equirectangular panorama. In *2018 24th international conference on pattern recognition (icpr)*, pages 2190–2195. IEEE, 2018. 3

[56] Yupeng Zhang, Hengzhi Zhang, Daojing Li, Liyan Liu, Hong Yi, Wei Wang, Hiroshi Suitoh, and Makoto Odamaki. Toward real-world panoramic image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 628–629, 2020. 1

[57] Qiang Zhao, Wei Feng, Liang Wan, and Jiawan Zhang. Sphorb: A fast and robust binary feature on the sphere. *International journal of computer vision*, 113:143–159, 2015. 3

[58] Yining Zhao, Chao Wen, Zhou Xue, and Yue Gao. 3d room layout estimation from a cubemap of panorama image via deep manhattan hough transform. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pages 637–654. Springer, 2022. 3

[59] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2051–2059, 2018. 6, 7