Self-supervised Interest Point Detection and Description for Fisheye and Perspective Images

Marcela Mera-Trujillo[*] Shivang Patel[*] Yu Gu Gianfranco Doretto
West Virginia University
Morgantown, WV 26506
{mameratrujillo, sap00008, yugu, gidoretto}@mix.wvu.edu

Abstract

Keypoint detection and matching is a fundamental task in many computer vision problems, from shape reconstruction, to structure from motion, to AR/VR applications and robotics. It is a well-studied problem with remarkable successes such as SIFT, and more recent deep learning approaches. While great robustness is exhibited by these techniques with respect to noise, illumination variation, and rigid motion transformations, less attention has been placed on image distortion sensitivity. In this work, we focus on the case when this is caused by the geometry of the cameras used for image acquisition, and consider the keypoint detection and matching problem between the hybrid scenario of a fisheye and a projective image. We build on a state-of-the-art approach and derive a self-supervised procedure that enables training an interest point detector and descriptor network. We also collected two new datasets for additional training and testing in this unexplored scenario, and we demonstrate that current approaches are suboptimal because they are designed to work in traditional projective conditions, while the proposed approach turns out to be the most effective.

1. Introduction

Many computer vision tasks require the automated detection of corresponding keypoints in multiple images, ranging from Structure-from-Motion [29,30] and 3D reconstruction [20, 30] to SLAM [10, 21], AR/VR [34] applications, and many others. The typical approach is to use an algorithm that we call interest point detector and descriptor. It addresses the problem of detecting keypoints and assigns a descriptor to them for matching across images. This is a very well studied area, and the popular solutions are based on classic algorithms like SIFT [18], or the most recent deep learning based approaches like SuperPoint [7].

Despite the recent advances in this field, the robustness against image distortions has received less attention compared to other factors such as noise, illumination variation, and rigid motion transformations in terms of the common metrics of repeatability and matching. This issue is crucial when dealing with keypoint matching between images captured by cameras with different geometries, as in the case of a rover equipped with a fisheye camera doing visual odometry over terrain previously mapped with a perspective camera. The hybrid camera scenario just described is challenging because keypoints may easily undergo image distortions that can hinder their detection or matching accuracy.

In this work, we set out to address the challenge of keypoint detection and matching in the hybrid camera scenario. We specifically focus on the case of fisheye images and perspective images. We build on a popular approach like [7], and we show that by deriving the hybrid homography that relates fisheye and projective images, it is possible to design a self-supervised training procedure that is effective for this particular case. We also design a new set of losses and show that recent contrastive losses used for self-supervised learning are effective for descriptor matching. In order to train and test our approach, we also developed two datasets, which we plan to release to the public. One is made of synthetic images generated from a video game, and the other was collected with real cameras. We tested the approach also on previously available datasets, but the data variety of current benchmarks in this domain is limited to do a robust training. The results demonstrate that the proposed approach is showing good promise by exhibiting the state-of-the-art metrics in the hybrid camera scenario while highlighting the difficulties of the current approaches that focus on the traditional perspective case.

2. Related Work

Existing feature extraction and descriptor methods can be categorized into two groups: traditional approaches and learning-based approaches.

[*] Denotes equal contribution.

Traditional approaches. Over the years, several approaches have been developed to efficiently solve the point feature extraction problem. One of the oldest and most basic feature detection approaches is the Harris Corner Detector [11], which distinguishes between edges and corners. Another method, SIFT [18], is scale-invariant and aims to solve issues related to intensity, viewpoint changes, and image rotation in feature matching. To address the slowness of SIFT, the SURF [3] algorithm was developed, which is faster and more efficient than SIFT, while still being robust and exhibiting similar matching performance. Another alternative to SIFT and SURF is FAST [26], which is a faster corner detection algorithm used for real-time applications. Additionally, BRIEF [5] can work with any other feature detector since it does not provide any method to find the features; it converts any other feature descriptors in floating-point numbers to binary strings. ORB [27] and AKAZE [23] are efficient alternatives to SIFT or SURF that use the FAST keypoint detector and BRIEF descriptor, providing more efficient performance.

While the majority of point feature detection and descriptor models work well in perspective images, they are less effective in fisheye images [32]. Some works have addressed this problem by either modifying the previous methods to work on omnidirectional images [1], or by transforming the omnidirectional image to a perspective image and then applying traditional methods [19]. However, our paper takes a different approach by using direct fisheye and perspective images as input, rather than altering the nature of the data to address the feature extraction problem.

Learning-based approaches. Feature extraction and descriptor techniques typically involve three steps: detecting keypoints, estimating orientation, and extracting robust descriptors. Initially, works such as TILDE [33], [14], [36], and [31] successfully tackled each of these problems individually. Later, LIFT [35] proposed a deep network architecture that unified the three steps by detecting keypoints, estimating orientation, and extracting feature descriptors in a single process. This pattern of unified approaches can be seen in subsequent methods. For example, SuperPoint [7] is a self-supervised framework that computes both keypoints and descriptors, but it performs poorly on rotation. In contrast, LF-Net [22] uses a two-branch network setup to learn sparse keypoints and descriptors that are both scale invariant and oriented. Similarly, D2-net [9] and R2D2 [25] focus on reliable dense feature descriptors and feature detectors. More recently, RoRD [24] proposed a framework that extends D2-net by learning rotation-robust local descriptors through data augmentation and orthographic viewpoint projection. Moreover, FisheyeSuperPoint [13] fine-tunes SuperPoint to improve keypoint detection on fisheye images.

Despite the significant progress in learning based frameworks, no other previous works addresses the learning based feature extraction and descriptor in a hybrid scenario. To the best of our knowledge, [2, 17] are the methods closest to ours that in some way address the hybrid scenario, although they do so by taking a traditional approach.

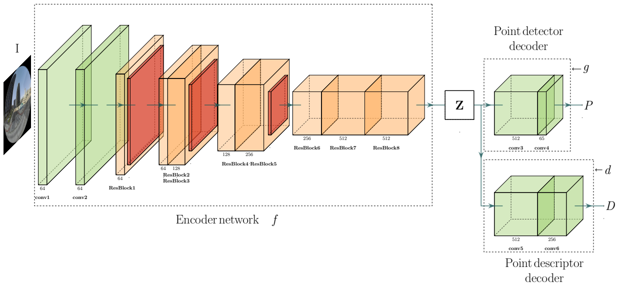

Figure 1. **Hybrid Interest Point Architecture.** Architecture of the hybrid interest point descriptor and detector.

3. Hybrid Interest Point Model

Our goal is to design an interest point detector and descriptor that is effective in a hybrid camera system scenario, meaning that pairs of corresponding keypoints to be selected and matched, can belong to images acquired by cameras with different geometries. We specifically focus on the hybrid case of fisheye and perspective camera models.

3.1. Architecture

The architecture and methods build on the SuperPoint approach [7], with some notable differences. It is summarized in Figure 1. It comprises of three major components. Given an input image I, of size $h \times w$, a shared *encoder* network f produces a feature $\mathbf{Z} = f(I)$. $\mathbf{Z}$ is the input of a *point detector decoder* network g that generates a pixel-wise map $\mathcal{P} = g(\mathbf{Z})$, where $\mathcal{P}_i$ indicates the probability of an interest point being located at pixel i. $\mathbf{Z}$ is also the input of a *point descriptor decoder* network d that generates a pixel-wise map $\mathcal{D} = d(\mathbf{Z})$, where $\mathcal{D}_i$ is the descriptor of pixel i, which has length 256, and ℓ_2-norm one.

Compared to [7] the encoder network f is a ResNet-18 [12] with some modifications. We retain only the 18 convolutional layers, and we insert three 2×2 non-overlapping max-pooling layers, after the 5-th, 10-th, and 14-th convolutional layers. Therefore, $\mathbf{Z}$ has dimensions $h_c \times w_c \times 512$, where $h_c = h/8$, and $w_c = w/8$. In this way, the decoders g and d are the same as in [7], only that their first convolutional layers accept an input feature with 512 channels, rather than 256.

The architecture is fully convolutional, and the intent is to process the whole input image, which could be either a fisheye or a perspective image, with a single feed-forward computation to generate interest point detections and descriptors. The detections are produced by the pipeline $g \circ f$, and the descriptors by the pipeline $d \circ f$, so that the computations made by f are shared.

3.2. Loss Functions

The three networks f, g, and d are trained jointly, with a self-supervised procedure. The point detection pipeline $g \circ f$ is trained in a supervised manner, where for an image I we use a set of pixel-wise automatically generated pseudo-labels $\mathcal{Y}$ indicating interest point positions. In § 3.3 we explain how $\mathcal{Y}$ is obtained.

The detection loss function used is designed around the architecture described in § 3.1. In particular, the image I is virtually divided in $h_c \times w_c$ disjoint cells of size 8×8 pixels, and the output $\mathcal{P}_{i,j}$ of a cell in position (i,j) is produced by a softmax layer, which has 65 outputs to handle the case when there are no interest points present. So, if $\mathcal{Y}_{i,j}$ indicates the labels of cell (i,j), then the detection loss function for image I is

$$\mathcal{L}_{g \circ f}(I, \mathcal{Y}) = \frac{1}{h_c w_c} \sum_{i,j=(1,1)}^{(h_c,w_c)} \ell(\mathcal{P}_{i,j}, \mathcal{Y}_{i,j}) \ , \tag{1}$$

where $\ell(\cdot)$ is the cross-entropy loss. Note that if there is more than one interest point in cell (i,j), only one is randomly picked.

For a hybrid interest point detector it is necessary to detect points in fisheye as well as perspective images. Typically, fisheye images have a significantly larger field of view than perspective images, and it is reasonable to assume that they will contain more image structures, leading to more interest points. Therefore, for a given fisheye image I we assume that there will be K perspective images $I'_1, \dots, I'_K$ with a field of view that significantly overlaps with the one of image I. In § 4 we describe our approach to randomly generate the set $\{I'_k\}$ automatically from I, with field of view that fully overlaps with the one of I. Also, from $\{I'_k\}$ we can generate the pseudo-labels $\{\mathcal{Y}'_k\}$. In this way, the complete *detection loss* for image I becomes

$$\mathcal{L}_{det}(I, \{I'_k\}, \mathcal{Y}, \{\mathcal{Y}'_k\}) = \mathcal{L}_{g \circ f}(I, \mathcal{Y}) + \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{g \circ f}(I'_k, \mathcal{Y}'_k) \ , \tag{2}$$

To learn instead that corresponding points in a fisheye image I and a perspective image I'_k should have the same descriptor, we set up a contrastive prediction task [6]. Specifically, given the architecture of the descriptor decoder d, similarly to the detector decoder g, an image is divided in the same set of 8×8 cells, and $d \circ f$ predicts the descriptor for each cell. The pixel-wise descriptors are obtained via bicubic interpolation (see [7] for details). Therefore, let $\mathbf{d} \in \mathcal{D}$ be a cell descriptor of image I. Let us also assume that $\mathcal{H}_k$ is the image domain transformation that has mapped I onto I_k (which will be defined in § 4). Then, the centroid position of the cell of $\mathbf{d}$ will be potentially mapped, according to $\mathcal{H}_k$, onto a cell of I'_k. Let us indicate the descriptor of that cell with $\mathbf{d}'_{k,\mathbf{d}}$. We expect $\mathbf{d}$ and $\mathbf{d}'_{k,\mathbf{d}}$ to be as close as possible since they describe the same region in the two images. So, $(\mathbf{d}, \mathbf{d}'_{k,\mathbf{d}})$ will form a positive pair. The other pairs $(\mathbf{d}, \mathbf{d}')$, where $\mathbf{d}' \in \mathcal{D}'_k$ does not represent the cell of $\mathbf{d}'_{k,\mathbf{d}}$, and are expected to be made of different descriptors, and thus are negative pairs. Therefore, the descriptor loss for the pair of images I, and I'_k, is given by

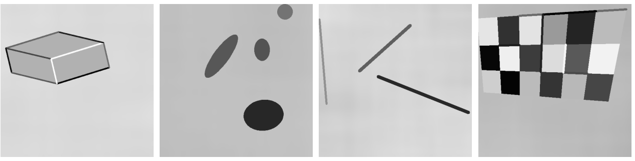

Figure 2. **Synthetic images with geometric primitives.** Image samples of size 320 × 320 created at runtime to pre-train the interest point detector pipeline.

$$\mathcal{L}_{d \circ f}(I, I'_k, \mathcal{H}_k) = \frac{1}{|\mathcal{D}_k|} \sum_{\mathbf{d} \in \mathcal{D}_k} - \log \frac{\exp\left(\mathbf{d}^\top \mathbf{d}'_{k,\mathbf{d}} / \tau\right)}{\sum_{\mathbf{d}' \in \mathcal{D}'_k} \exp\left(\mathbf{d}^\top \mathbf{d}' / \tau\right)} \ , \tag{3}$$

where $\mathcal{D}_k \subseteq \mathcal{D}$ is the subset of descriptors with centroid location that maps onto a location inside image I'_k. The complete *descriptor loss* function for image I is

$$\mathcal{L}_{des}(I, \{I'_k\}, \{\mathcal{H}_k\}) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{d \circ f}(I, I'_k, \mathcal{H}_k) \ . \tag{4}$$

Finally, the *total loss* for the joint training on a per fisheye image basis is

$$\mathcal{L} = \mathcal{L}_{det} + \gamma \mathcal{L}_{des} \ , \tag{5}$$

where γ strikes a balance between the detection and descriptor losses.

3.3. Training Procedure

The first training step is to arrive at a mechanism for generating pseudo-labels $\mathcal{Y}$ from the image I. Similar to [7], we train the detector pipeline $g \circ f$ from scratch with synthetically generated images like triangles, lines, polygons and so on, which have 2D geometric primitives that allow to identify interest points unambiguously, like corresponding to T-junctions, Y-junctions, L-junctions, or center of elipsoidal blobs. We use OpenCV and the Kornia image library to generate 2.4M synthetic images with relative interest point annotations that we use for pre-training the detector with loss (1) in a self-supervised manner. The images are randomly generated and are seen only once by the detector during training using batches of size 16. The training continues for 150K iterations, processing the same number of batches. Figure 2 shows samples of such images.

The next training step is to use another self-supervised process called Homographic Adaptation, described in [7].

The goal is to make the detector pipeline work well in more general scenarios, and also with real images. We use Homographic Adaptation where we generate random homographies as explained in § 4.2, and we leverage the perspective training images of the KITTI-360 dataset [16], together with synthetic perspective training images that we generate from the game Grand Theft Auto V (GTAV), as explained in § 5, for an enhanced exposure to image variety. We used 12K images from KITTI-360, 30K images from GTAV, all resized to 320 × 320, and trained for more than 100K iterations until convergence, forming batches of 16 images, and using loss (1). For interest point superset generation we sampled 100 random homographies per image. We repeat Homographic Adaptation twice.

After completing the two steps above, given a fisheye image I, and corresponding perspective images $\{I'_k\}$, we can use the current detector pipeline $g \circ f$ to compute the *pseudo-labels* $\mathcal{Y}$ and $\{\mathcal{Y}'_k\}$. At this point training of all the networks f, g, and d can continue with the joint total loss (5). This is based on using fisheye images from the KITTI-360 dataset, and the synthetic dataset GTAV-Hybrid that we introduce in § 5. More details are included in § 6.

4. Generation of Perspective Images

Given a fisheye image I we show how to produce a perspective image I'. This will allow to randomly sample the set $\{I'_k\}$, as well as the set of transformations $\{\mathcal{H}_k\}$.

4.1. Hybrid Homography Model

We derive a model that is a hybrid homography between a fisheye and a perspective image. With reference to Figure 3, we consider a system with an omnidirectional camera with fisheye lenses, centered in O_1, and a perspective camera centered in O_2. The cameras are observing a 3D scene made by a planar surface. Let P be a point on such surface, represented in homogeneous coordinates with respect to a world reference frame with the origin on the surface and the xy-plane parallel to the surface. Let p_1 be a 3D vector pointing towards P from O_1, then the following relationship holds [28]

$$\lambda_1 p_1 = \lambda_1 \begin{bmatrix} u_1 \\ v_1 \\ \varphi(u_1, v_1) \end{bmatrix} = \Pi_1 P \ , \qquad (6)$$

where λ_1 is an appropriate scalar, (u_1, v_1) represent the sensor coordinates, $\varphi(u_1, v_1)$ is a polinomial function in $\rho_1 = \sqrt{u_1^2 + v_1^2}$, and $\Pi_1 = [R_1, T_1]$ is a projection matrix, where the extrinsic parameters (R_1, T_1) map P onto the coordinate system of the camera. Finally, since $P = [X, Y, 0, 1]^\top$ because it is on the planar surface, we also have that

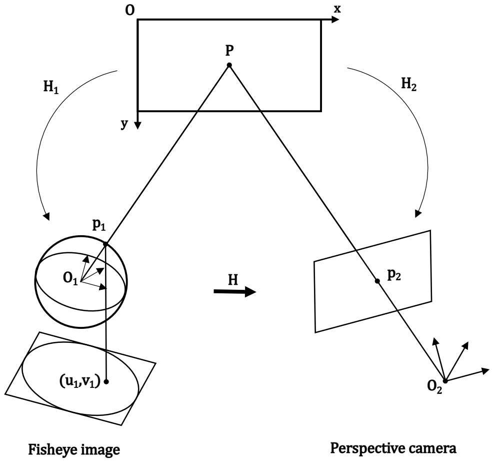

Figure 3. **Hybrid fisheye and perspective imaging system.** O_1 is the camera center of a fisheye camera and O_2 is the camera center of a perspective (conventional) camera.

$$\lambda_1 p_1 = [R_1, T_1] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = H_1 \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \ , \qquad (7)$$

where H_1 is a 3×3 matrix obtained from $[R_1, T_1]$ by removing its third column.

Similarly, let p_2 be a 3D vector pointing towards P from O_2, then the following relationship holds

$$\lambda_2 p_2 = \lambda_2 \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \Pi_2 P \ , \qquad (8)$$

where λ_2 is a suitable parameter. If $\Pi_2 = [R_2, T_2]$ is the projection matrix, where (R_2, T_2) map P onto the coordinate system of the perspective camera, it follows that

$$\lambda_2 p_2 = [R_2, T_2] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = H_2 \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \ , \qquad (9)$$

and H_2 is $[R_2, T_2]$ without its third column.

From the discussion above it follows that

$$\lambda \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = H \begin{bmatrix} u_1 \\ v_1 \\ \varphi(u_1, v_1) \end{bmatrix} \ , \qquad (10)$$

where λ is a suitable scalar, $H \doteq H_2 H_1^{-1}$ is a homography that encodes the geometric relationship between the planar scene and the camera system. The mapping $\mathcal{H} : (u_1, v_1) \mapsto (u_2, v_2)$ defined by (10) is what we refer to as the *hybrid homography* of the system.

To derive a mapping between pixel coordinates, recall that (u_1, v_1) is related to the fisheye image coordinates (x_1, y_1) via an affine transformation (A, t), and that p_2 is related to the perspective image coordinates (x_2, y_2) via the intrinsic parameters matrix $\mathtt{K}$. This gives the relationship

$$\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \mathtt{K} H \begin{bmatrix} A[x_1, y_1]^\top + t \\ \varphi(A[x_1, y_1]^\top + t) \end{bmatrix} . \tag{11}$$

4.2. Random Generation of Homographies

The random generation of hybrid homographies can occur by sampling the matrix H. We can consider the intrinsics of the cameras to remain constant, as it often happens during the collection of a dataset. One way to sample H would be to randomly sample the quadruple (R_1, T_1, R_2, T_2). However, this approach is difficult to control, and we take a more direct approach.

First, we observe that when the cameras share the origin, i.e., $O_2 = O_1$, and their reference systems are aligned, then $R_2 = R_1, T_2 = T_1$, and H is the identity. We then perturb this configuration with a series of transformations. One is an in-plane 2D rotation H_R (12), then a 2D scaling H_s (13), then a 2D skew H_k (13), then a shear H_h (14) of the xy-plane into the plane passing through $(0, 0, 0)$, $(1, 0, h_x)$, and $(0, 1, h_y)$, and finally a 2D translation H_T (14). The transformations are summarized as follows

$$H_R = \begin{bmatrix} \cos a & \sin a & 0 \\ -\sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{12}$$

$$H_s = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} , \quad H_k = \begin{bmatrix} 1 & k_x & 0 \\ k_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{13}$$

$$H_h = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h_x & h_y & 1 \end{bmatrix} , \quad H_T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} . \tag{14}$$

H is then randomly sampled as $H = H_R H_s H_k H_h H_T$. This allows to generate the homographies for the Homographic Adaptation, and also the set of hybrid homography transformations $\{\mathcal{H}_k\}$.

4.3. Perspective Image Synthesis

As described in § 4.2 we can sample a hybrid homography $\mathcal{H}$ as in (11), and from a fisheye image I we generate a perspective image I' by using inverse warping with bilinear interpolation. Note that since the projection center of the perspective view is close if not overlapping with the projection center of the fisheye camera, and since the scene objects are normally at a much greater distance than the distance between the projection centers, model (11) is fully respectful of the 3D nature of the scene, regardless of its planarity. By setting an appropriate sampling range of parameters, it is easy to draw a set of homographies $\{\mathcal{H}_k\}$ that give rise to a set of projective images $\{I'_k\}$, with field of view completely overlapping with the field of view of the fisheye image I.

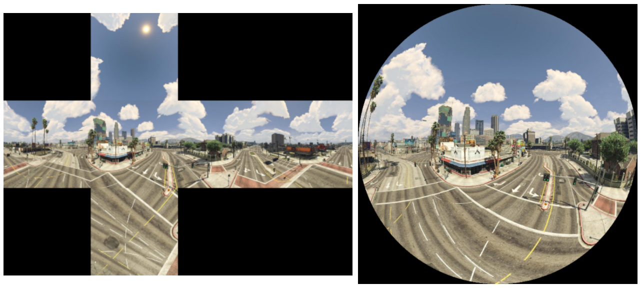

Figure 4. **Cube maps and fisheye images.** Cube map generated from GTAV with G2D (left), and corresponding conversion to a fisheye image (right).

5. GTAV-Hybrid Synthetic Dataset

Due to the lack of publicly available datasets with annotations that correlate fisheye and perspective images, we design a process for generating synthetic fisheye images for which the calibration parameters are known. The scenarios from which we capture images is given by Grand Theft Auto V (GTAV), a popular role playing game in an expansive virtual city. We use a software package called G2D [8] to capture computer generated images of GTAV. Specifically, we set up a path in the simulator and then tour the path capturing images with the perspective virtual camera with six degrees of freedom (6DoF) at specific timestamps.

Since G2D implements only a perspective virtual camera, but we are interested in capturing fisheye images, we take a *cube mapping* approach. This means that at every location we capture 6 views with a virtual camera with a field of view of 90 degrees, and which points in the 6 principal directions also separated by 90 degrees. Subsequently, we follow [4] to convert a cube map into a fisheye image. See Figure 4 for an example of a GTAV cube map and a fisheye image generated from it.

From fisheye images, we use the procedure outlined in § 4.3 to sample perspective images. Figure 5 shows examples of fisheye images from *GTAV-Hybrid*, our new dataset which will be made publicly available, and perspective images generated from them.

6. Experiments

6.1. Datasets

GTAV-Hybrid. We use GTAV-Hybrid for training and testing. The dataset includes a total number of 41280 fisheye images, where only 30000 fisheye images were used

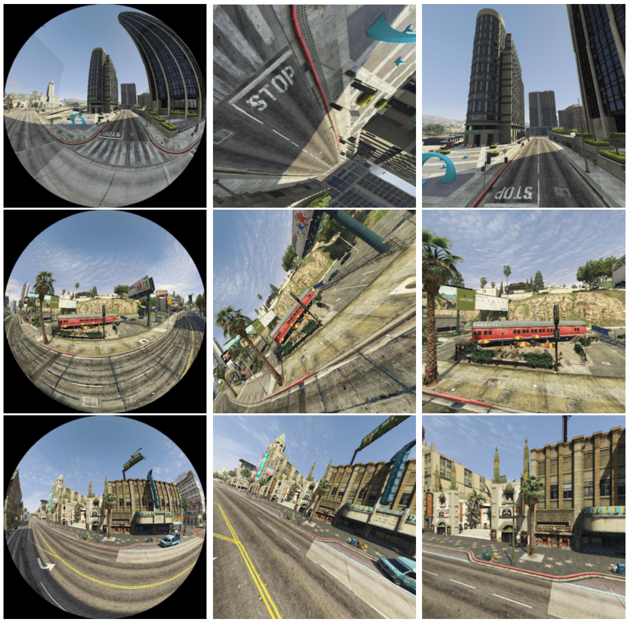

Figure 5. **Paired fisheye and perspective images.** Examples of synthetic fisheye images (left), and paired perspective images (right) obtained by applying different hybrid homography transformations (11).

for training, with $K = 5$ paired homography generated perspective images. We keep 600 distinct images from the remaining images for testing.

KITTI-360. KITTI-360 [16] comprises of a vast collection of images captured in the suburbs of Karlsruhe, Germany. Due to the sequential nature of the images, not all of them are useful for training purposes, since many of them overlap with subsequent images. To address this issue, we used a skip sampling approach to select a representative subset of 16,000 images. Out of these, 12,000 images were used for training, while the remaining 4,000 images were further reduced to a set of 600 distinct images for testing.

Evansdale. This is a new dataset that we have collected around our campus with a Kodak PIXPRO SP360 fisheye camera, and a Canon EOS 80D DSLR. It has 30 pairs of fisheye and perspective images. The intrinsic and extrinsic camera parameters have been extracted. This dataset will also be made publicly available.

We used GTAV-Hybrid and the KITTI-360 datasets for training and the Evansdale dataset for testing.

6.2. Implementation Details

The network f uses ResNet-18, which includes batch normalization, max pooling, and downsampling layers. The first layer of f has a kernel size of 3 instead of the original 7, followed by batch normalization and another convolutional layer with a kernel size of 3. We opted to use Leaky ReLU activations to maintain training stability throughout the network. The channel widths of the original network (64-128-256-512) were retained. To achieve a feature size of $\frac{h}{8} \times \frac{h}{8}$, where h and w represent the height and width of the image, we used downsampling layers.

	Mean Matching Score		Average number of matches	Repeatability Score	
	$\epsilon = 3$	$\epsilon = 5$		$\epsilon = 3$	$\epsilon = 5$
SIFT [18]	0.437	0.475	107	0.287	0.539
ORB [27]	0.446	0.503	92	**0.488**	**0.670**
AKAZE [23]	0.334	0.393	91.4	0.356	0.584
BRISK [15]	0.185	0.223	90	0.195	0.294
SuperPoint [7]	0.235	0.244	24	0.395	0.564
D2-Net [9]	0.026	0.052	18.5	0.207	0.466
RoRD [24]	0.101	0.199	45.42	0.275	0.526
Ours	**0.480**	**0.520**	109.28	0.449	0.573

Table 1. Results on the GTAV-Hybrid test dataset.

	Mean Matching Score		Average number of matches	Repeatability Score	
	$\epsilon = 3$	$\epsilon = 5$		$\epsilon = 3$	$\epsilon = 5$
SIFT [18]	0.374	0.409	104.53	0.339	0.542
ORB [27]	0.415	0.461	84.70	0.514	**0.633**
AKAZE [23]	0.325	0.380	81.65	0.455	0.597
BRISK [15]	0.273	0.313	83.32	0.320	0.461
SuperPoint [7]	0.244	0.254	30.13	**0.522**	0.541
D2-Net [9]	0.033	0.074	16.45	0.257	0.482
RoRD [24]	0.113	0.227	58.0	0.339	0.544
Ours	**0.478**	**0.518**	172.92	0.455	0.574

Table 2. Results on the KITTI-360 test dataset.

Training. During the training of the interest point detector network $g \circ f$ we used a batch size of 16 and a learning rate of 10^{-3}. Following that, we utilized the trained detector network to create pseudo ground truth labels for the GTAV-Hybrid and KITTI-360. We resized each image to 320 × 320 and converted it to grayscale. Subsequently, we generated 100 random homographies to generate detection labels and trained the network with these images for 5,000 iterations. We repeat this Homography Adaptation step twice to achieve robust detection.

To self-supervise the hybrid point descriptor and detector model training, we required two sets of images: $K = 5$ perspective images and one fisheye image. We created unique perspective images on-the-fly, given a fisheye image, throughout the training process. We keep the loss balancing term $\gamma = 0.001$ and a temperature $\tau = 0.15$ in (3). We used a batch size of 16 and a learning rate of 10^{-3}, and we train for more than $100,000$ iterations until convergence.

Our method is implemented in PyTorch in a distributed setting using four Nvidia A6000 GPUs with an Intel Xenon CPU. We employed standard data augmentation techniques such as random Gaussian blur and random brightness changes in all of our training.

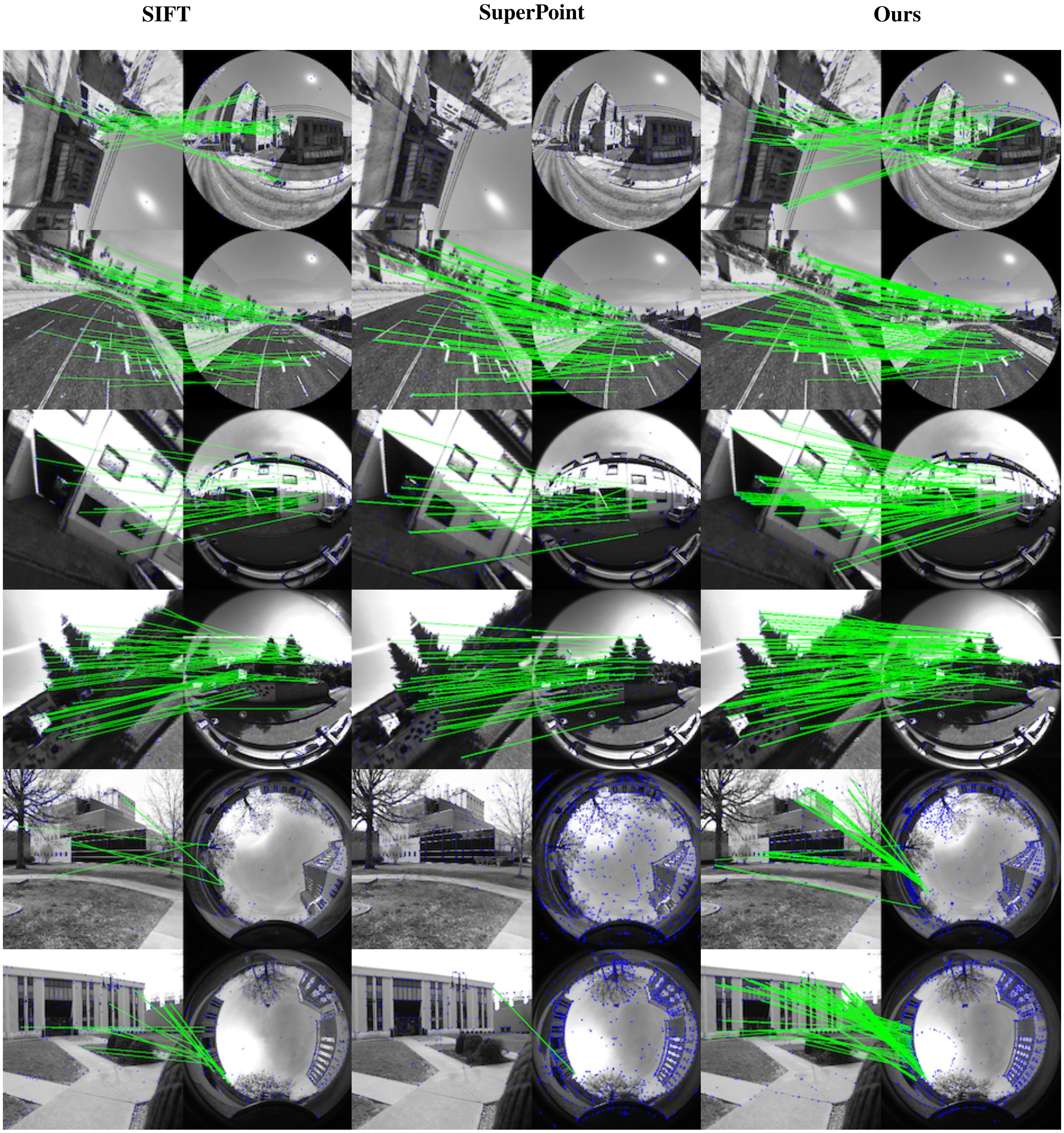

Figure 6. **Qualitative Results.** First two rows are from the GTAV-Hybrid dataset, the second two from the KITTI-360 dataset, and the last two from the Evansdale dataset. The green lines indicate accurate matches. Our approach appears to perform better in cases of distortion, and is comparable to other methods in cases of minimal distortion. It's worth mentioning that SuperPoint, due to its lack of rotation invariance, encounters difficulties generating matches when confronted with rotational homography in rows 1. Similarly, it struggles to produce matches in the Evansdale dataset in rows 5 and 6 for the same reason.

6.3. Results

We evaluate our hybrid interest point detector on three datasets: GTAV-Hybrid, KITTI-360 [16], and Evansdale. We compare our approach to state-of-the-art techniques including the OpenCV implementations of classical detectors such as SIFT [18], ORB [27], BRISK [15], and AKAZE [23], as well as recent learning-based methods such as SuperPoint [7], D2-Net [9], and RoRD [24], for

	Mean Matching Score		Average number of matches	Repeatability Score	
	$\epsilon = 5$	$\epsilon = 10$		$\epsilon = 5$	$\epsilon = 10$
SIFT [18]	0.043	0.199	77.16	0.195	0.332
ORB [27]	0.0962	0.186	94.5	0.260	0.398
AKAZE [23]	0.152	0.235	97.66	0.294	0.492
BRISK [15]	0.001	0.006	38	0.120	0.188
SuperPoint [7]	0.034	0.045	5.13	0.336	**0.521**
D2-Net [9]	0.0	0.0	0.0	0.187	0.458
RoRD [24]	0.044	0.148	1.06	0.193	0.454
Ours	**0.223**	**0.340**	33.43	**0.343**	0.497

Table 3. Results on the Evansdale dataset.

which we used pre-trained models obtained from the authors' GitHub repository. Note, however, that these pre-trained models were designed to perform well on perspective images, and they may not generalize well to the hybrid scenario. They are used here as baselines, given also the lack of learning-based approaches for the hybrid scenario, and to motivate the development of our approach.

We used a nearest neighbor matching strategy for all the detected descriptors within an image pair. The Matching Score is based on a *correct distance* ϵ of 3 px, 5 px, except for the Evansdale dataset, where we used a *correct distance* of 5 px and 10 px due to a higher reprojection error. We resized the perspective and fisheye images to 320×320 during testing, keeping a maximum of the top 300 keypoints for perspective images and the top 1000 keypoints for fisheye images to balance the information gap due to the smaller field of view of the perspective images. We use two evaluation metrics: Mean Matching Score (MMS) for descriptor evaluation, and Repeatability for detector evaluation, calculated by dividing the number of correct matches by the total matches suggested by the pipeline. We follow [7] to compute the Repeatability.

Tables 1, 2, and 3 summarize the performance of our method. It can be seen that the proposed rigorous training with random homography images, including extreme random homographies, has allowed our method to consistently achieve the highest MMS.

Traditional detection methods, such as [15, 18, 23, 27], underperform on fisheye image feature extraction and descriptor due to the strong distortion and non-linear projection inherent in fisheye cameras, which violates the assumption of local linearity that these methods rely on. Thus, they produce comparable results on both GTAV-Hybrid and KITTI-360, where distortion is less. However, they struggle when distortion increases, as it can be seen in the Evansdale dataset. These methods are rotation invariant but not distortion invariant.

ORB has been shown to have high repeatability compared to other methods, likely due to the clustering of points that allows it to produce more repeatable matches. However, the proposed method demonstrates competitive results on Evansdale, where it achieves the best repeatability score. This score depends heavily on the quality of the learned point detector, and although we trained our network to detect a diverse range of points, the number of points detected may not be sufficient for some datasets. Therefore, increasing the number of training passes could further improve performance in this area.

Learning-based approaches, such as [7,9,24], are trained to work with perspective images that are not affected by distortion. They yield a good number of matches as shown in Tables 1, 2, 3. SuperPoint is not rotation invariant and does not perform well when faced with extreme rotation, whereas D2-Net is not trained on homographies and it cannot deal with extreme homography or viewpoint changes. RoRD, using the D2-Net approach, developed a rotation invariant version that performs better than D2-Net on these test datasets.

The proposed method, on the other hand, despite not matching the overall repeatability of ORB or SuperPoint, it generates a robust descriptor for the hybrid scenario, which is effective with both synthetic and real image datasets. This is because it is designed to be distortion and rotation invariant, making it more effective for datasets with high levels of distortion, such as Evansdale. Therefore, we infer that our method effectively overcomes the limitations of traditional and learning-based approaches that we have discussed.

7. Conclusions

We introduced a novel approach for learning an interest point detector and descriptor that is specifically designed for hybrid camera systems. It leverages a hybrid homography model and a self-supervised learning approach, along with synthetic data generation to address the lack of real-world data in this domain. We constructed two datasets, one synthetic and one real, for evaluating the methods. Furthermore, we proposed a modified contrastive loss to enhance the learning process for descriptors. Our experiments demonstrate the effectiveness of our approach on a real world hybrid dataset, further validating its potential. We believe that this work can pave the way for developing more effective keypoint detector and descriptor models for hybrid camera systems.

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration under Grant No. WV-80NSSC17M0053 issued through the NASA EPSCoR Program, and also by the National Science Foundation under Grant No. 1657179.

References

[1] Zafer Arican and Pascal Frossard. Omnisift: Scale invariant features in omnidirectional images. In *2010 IEEE International Conference on Image Processing*, pages 3505–3508. IEEE, 2010. 2

[2] Yalin Bastanlar, Alptekin Temizel, Yasemin Yardimci, and Peter Sturm. Multi-view structure-from-motion for hybrid camera scenarios. *Image and Vision Computing*, 30(8):557–572, 2012. 2

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Lecture notes in computer science*, 3951:404–417, 2006. 2

[4] Bruno Berenguel-Baeta, Jesus Bermudez-Cameo, and Jose J Guerrero. Omniscv: An omnidirectional synthetic image generator for computer vision. *Sensors*, 20(7):2066, 2020. 5

[5] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 778–792. Springer, 2010. 2

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3

[7] 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, 6, 7, 8

[8] Anh-Dzung Doan, Abdul Mohsi Jawaid, Thanh-Toan Do, and Tat-Jun Chin. G2d: from gta to data. *arXiv preprint arXiv:1806.07381*, 2018. 5

[9] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019. 2, 6, 7, 8

[10] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pages 834–849. Springer, 2014. 1

[11] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 2

[12] 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. 2

[13] 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. 2

[14] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 100–117. Springer, 2016. 2

[15] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, 2011. 6, 7, 8

[16] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv preprint arXiv:2109.13410*, 2021. 4, 6, 7

[17] Huei-Yung Lin and Ming-Liang Wang. Hopis: Hybrid omnidirectional and perspective imaging system for mobile robots. *Sensors*, 14(9):16508–16531, 2014. 2

[18] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 1, 2, 6, 7, 8

[19] Chuiwen Ma, Liang Shi, Hanlu Huang, and Mengyuan Yan. 3d reconstruction from full-view fisheye camera. *arXiv preprint arXiv:1506.06273*, 2015. 2

[20] Roger Mohr, Long Quan, and Françoise Veillon. Relative 3d reconstruction using multiple uncalibrated images. *The International Journal of Robotics Research*, 14(6):619–632, 1995. 1

[21] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1

[22] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *Advances in neural information processing systems*, 31, 2018. 2

[23] Adrien Bartoli Pablo Alcantarilla (Georgia Institute of Technolog), Jesus Nuevo (TrueVision Solutions AU). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013. 2, 6, 7, 8

[24] Udit Singh Parihar, Aniket Gujarathi, Kinal Mehta, Satyajit Tourani, Sourav Garg, Michael Milford, and K Madhava Krishna. Rord: Rotation-robust descriptors and orthographic views for local feature matching. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1593–1600. IEEE, 2021. 2, 6, 7, 8

[25] 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. 2

[26] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 430–443. Springer, 2006. 2

[27] 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. 2, 6, 7, 8

[28] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701. IEEE, 2006. 4

[29] 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

[30] Johannes L Schonberger, Filip Radenovic, Ondrej Chum, and Jan-Michael Frahm. From single image query to detailed 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5126–5134, 2015. 1

[31] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE international conference on computer vision*, pages 118–126, 2015. 2

[32] Guofeng Tong, Xue Chen, and Ning Ye. A spherical model based keypoint descriptor and matching algorithm for omnidirectional images. *Advances in Mechanical Engineering*, 6:154376, 2014. 2

[33] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5279–5288, 2015. 2

[34] Miao Wang, Xu-Quan Lyu, Yi-Jun Li, and Fang-Lue Zhang. Vr content creation and exploration with deep learning: A survey. *Computational Visual Media*, 6:3–28, 2020. 1

[35] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 467–483. Springer, 2016. 2

[36] Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent Lepetit. Learning to assign orientations to feature points. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 107–116, 2016. 2