# GMM-IKRS: Gaussian Mixture Models for Interpretable Keypoint Refinement and Scoring

Emanuele Santellani[1,2], Martin Zach[1], Christian Sormann[3], Mattia Rossi[3], Andreas Kuhn[3], and Friedrich Fraundorfer[1]

[1] Graz University of Technology, Graz, Austria, name.surname@tugraz.at
[2] Pro2Future GmbH, Linz, Austria, name.surname@pro2future.at
[3] Stuttgart Laboratory 1, Sony Semiconductor Solutions Europe, Stuttgart, Germany, name.surname@sony.com

**Abstract.** The extraction of keypoints in images is at the basis of many computer vision applications, from localization to 3D reconstruction. Keypoints come with a score permitting to rank them according to their quality. While learned keypoints often exhibit better properties than handcrafted ones, their scores are not easily interpretable, making it virtually impossible to compare the quality of individual keypoints across methods. We propose a framework that can refine, and at the same time characterize with an interpretable score, the keypoints extracted by any method. Our approach leverages a modified robust Gaussian Mixture Model fit designed to both reject non-robust keypoints and refine the remaining ones. Our score comprises two components: one relates to the probability of extracting the same keypoint in an image captured from another viewpoint, the other relates to the localization accuracy of the keypoint. These two interpretable components permit a comparison of individual keypoints extracted across different methods. Through extensive experiments we demonstrate that, when applied to popular keypoint detectors, our framework consistently improves the repeatability of keypoints as well as their performance in homography and two/multiple-view pose recovery tasks.

**Keywords:** keypoint refinement · image matching · SfM

## 1 Introduction

Establishing point correspondences between images is a task that has been of critical importance in the computer vision field since its conception. Even in the deep learning era, reliable point-wise matches are still a fundamental requirement for many applications, including Structure-from-Motion (SfM) [1, 17, 32], Simultaneous Localization and Mapping (SLAM) [2, 21, 31], visual localization [30, 34] and object tracking [39]. The quality of the established correspondences is crucial for these algorithms when estimating core geometric relationships, such as camera poses and homographies. Therefore, there is a high demand for algorithms capable of identifying correspondences, even in challenging scenarios.

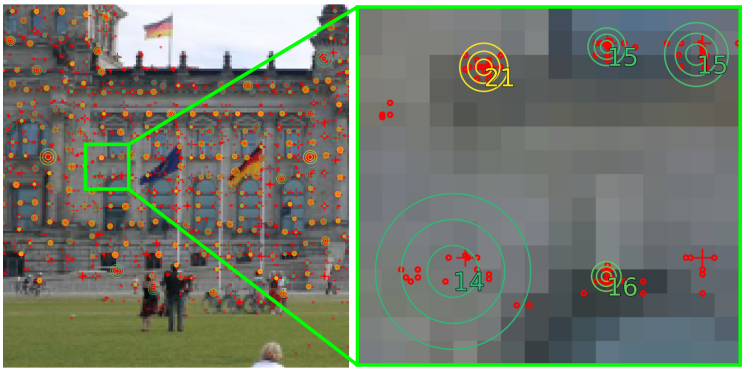



**Fig. 1:** Visualization of the input keypoints and their refined positions. The keypoints from the original image are represented as + in red, while the red circles represent back-projected keypoints detected in the warped images. The Gaussian fit at each keypoint cluster is represented as a set of concentric circles whose spread encodes the variance. The refined keypoints are the centers of the Gaussians, with *robustness* and *deviation* represented by the number next to the Gaussian and by its spread, respectively.

Early hand-crafted local feature extractors like the Scale Invariant Feature Transform (SIFT) [15], Speeded Up Robust Features (SURF) [5], and Oriented FAST and rotated BRIEF (ORB) [26], have been applied in a variety of different applications. These methods have established a three step paradigm: keypoint detection, descriptor extraction and pairwise matching. In recent years, deep learning has shown great promises for correspondence search, especially in highly challenging scenarios. This has led to the introduction of many novel paradigms for this task. In particular, inherent to their deep learning nature, these architectures oftentimes exhibit less separation between the aforementioned phases.

In addition, deep matching algorithms, such as SuperGlue [29], have emerged. These more advanced methods are capable of matching even less discriminative descriptors by employing a more global form of reasoning. Despite this advancement, deep matchers still rely on the input keypoints, which have now become a primary limiting factor for pose related applications.

Defining what makes a good keypoint is not an easy task, especially when taking into consideration the specific requirements of the downstream tasks. Nevertheless, regardless of how discriminative its surrounding area is, a good keypoint must at least be *repeatable*. In other words, it must be detected again in a different image depicting the same scene. Focusing on and expanding the concept of *repeatability*, in this work we introduce a general framework designed to enhance and evaluate keypoints. Our framework takes an existing keypoint detection method as input, refines its detections, potentially adding new stable ones, and subsequently assigns two distinct scores to each refined keypoint: the *robustness*, which relates to the probability of detecting the keypoint again when the image undergoes viewpoint changes, and the *deviation*, which quantifies its localization accuracy. The core of our idea is very simple and strongly relates to

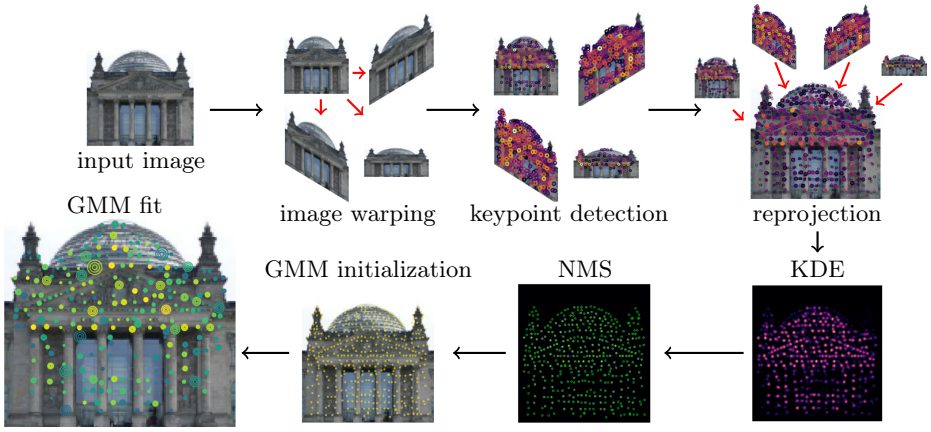



**Fig. 2:** Sketch of the proposed refinement and scoring framework. A set of image warping augmentations are applied to the input image. The chosen keypoint detector is applied to all the generated images, and detections in the warped images are projected back to the input image. The local maxima in the estimated density are used as initialization for a GMM fit. After convergence, each Gaussian component represents a refined keypoint characterized by the *robustness* and the *deviation* scores. This procedure adds additional robust keypoints not detected in the original image.

the fundamental definition of a good keypoint: As a first step, we generate multiple versions of the input image by applying some known affine transformations. We then run the given keypoint detector on each generated image, and use the inverse transformations to project the detected keypoints back into the original image. At this point, after a first coarse density estimation, we use our modified robust Gaussian mixture model (GMM) fit to group the keypoints into clusters. Our two scores are directly derived from the parameters of each estimated Gaussian.

The proposed refinement, owing to its compatibility with any keypoint detector and its linear complexity in the number of images, presents a valuable option for offline applications requiring robust and well localized keypoints, such as image-based 3D reconstruction. Additionally, thanks to the interpretability of the two scores, our method can provide more in-depth insights into the performance of a keypoint detector. Moreover, our two scores permit to define different keypoint rankings depending on the task at hand. As an example, more *robust* keypoints might be preferable in visual localization, whereas precise robotic applications might prefer low *deviation* ones.

In this work, we present the following contributions:

- A novel framework named GMM-IKRS (Gaussian Mixture Models for Interpretable Keypoint Refinement and Scoring) that can refine, and at the same time characterize the keypoints extracted by any detector with two interpretable scores.
- A modification of the expectation-maximization (EM) algorithm designed to make the GMM fitting robust to outliers.

– Insights into the properties of state-of-the-art learned and hand-crafted detectors through our extensive evaluation.

## 2 Related works

The literature on keypoint detectors and local feature extraction methods is large [5, 7, 10, 13, 16, 18, 19, 22, 25–27, 35, 38]. In this section, we focus on those methods that closely relate to our proposed framework and refer the reader to [2, 36] for a more extensive overview.

In a similar fashion to the first step of our pipeline, ASIFT [20] runs the SIFT [15] feature extractor (keypoint + descriptors) on multiple warped versions of the original image. Following this initial step, the descriptors extracted from each warped image need to be matched across all possible warped image pairs, which substantially increase the computational cost of the matching phase. In contrast, our framework only extract descriptors from the original image and leaves the matching phase unaltered. In addition, ASIFT directly relies on the set of keypoints extracted from the different warpings and does not apply any aggregation; instead, our clustering scheme evaluates and refines this initial set to obtain better localized keypoints and permits the estimation of our two scores.

The deep method SuperPoint [9] also employs a warping augmentation process, denoted homographic adaption, to generate heatmaps utilized in a second stage of the training. Specifically, random homography warpings are applied to the input image and subsequently processed by the network. An aggregated heatmap is then computed warping back all the network outputs. The initial step of our pipeline implements a similar warping scheme, however, our framework operates directly on the keypoints, rather than on the heatmaps. Furthermore, after re-projecting all the keypoints into the original image, our method applies a robust global clustering algorithm to produce a set of refined and well characterized keypoints.

After several successful works on joint keypoint and descriptor learning [10, 24, 27], promising deep matching algorithms like SuperGlue [29] and LightGlue [14] have led to a shift in focus within the research community. These deep matchers can cope with less reliable keypoint descriptors, sparkling renewed interest in methods focusing mainly on keypoint detection [4, 23, 28]. The recent work NeSS-ST [23] proposes a stability score against viewpoint transformation for keypoints detected by the Shi-Tomasi [33] detector. This method assesses the stability of a specified keypoint by firstly applying random homography warpings to a patch centered at the keypoint. It then picks the highest Shi-Tomasi [33] response for each patch and computes the sample covariance with respect to the original keypoint location. In contrast, our method does not require to define any local patch size to compute the scores, as our transformations are applied to the whole image. Moreover, our framework jointly assesses, in addition to the keypoint localization accuracy, its robustness conditioned on the detector in analysis. Lastly, as opposed to NeSS-ST [23], our framework does not incorporate any learned component and can be applied to arbitrary keypoint detectors.

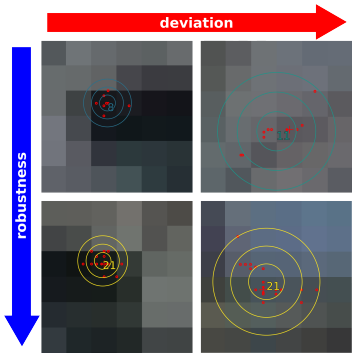



**Fig. 3:** Keypoint clusters and their two scores: *robustness* and *deviation*. *Robustness* measures the likelihood of detecting the keypoint again, while *deviation* measures its localization accuracy. A desirable keypoint has high *robustness* and low *deviation*, as in the bottom-left square.

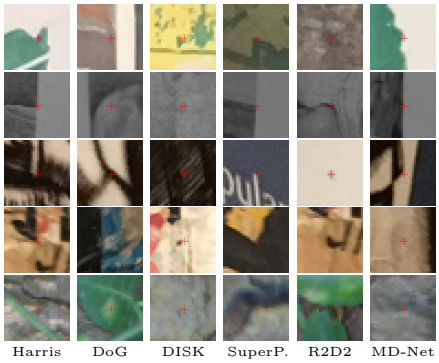



**Fig. 4:** Qualitative comparison between the best clusters found by different methods in the first 5 scenes of HPatches. For each scene (one per row), we select the lowest *deviation* cluster (best localization accuracy) among all the ones with *robustness* = 21 (which represents keypoints that have been detected in all warps).

## 3 Method

The purpose or our method is twofold: it refines the positions of a given keypoint set, additionally dropping low quality keypoints and adding new stable ones, while at the same time characterizing each resulting keypoint with two interpretable scores. These scores directly relate to two orthogonal keypoint properties: robustness to viewpoint changes and localization accuracy. As depicted in Figure 2, our pipeline consists of several stages, which we detail in the following subsections.

### 3.1 Multiple Image Warping

The goal of the first stage of our pipeline is to simulate real-world viewpoint changes. To this end, we generate multiple altered versions of the original image by applying a series of affine warpings, each sampled from a fixed set of possible transformations. In order to avoid uneven keypoint densities in the next step, we only use transformations that distort every part of the image uniformly.

### 3.2 Keypoint Detection & Reprojection

We run the chosen keypoint detector on all the warped images and pick the best $n$ keypoints from each, according to the scores given by the detector. Each set of detected keypoints is then re-projected into the input image reference frame through the inverse of the transformation originally used to generate the warped image. To compensate for very close keypoints, which may happen

as a consequence of warpings that increase the image area, we apply a non-maximum-suppression (NMS) to the reprojected keypoints from each warped image separately. Keypoints retained after NMS are then aggregated into a single set, regardless of the warped image they have been extracted from.

### 3.3 Density Estimation

The proposed approach characterizes keypoints based on robustness against viewpoint changes and localization accuracy, which we describe by metrics derived from a GMM. To ensure a proper fit, the parameters and initialization of the GMM, in particular the number of components and their respective means, have to be chosen in an informed manner. In this section, we briefly describe how we utilize classical kernel density estimation (KDE) to initialize the down-stream fitting task.

Let $(x_i)_{i=1}^N \subset \mathbb{R}^2$ denote the collection of $N$ re-projected keypoints extracted from the augmentation pipeline. From this collection, we construct a KDE $f_{\text{KDE}} : \mathbb{R}^2 \to \mathbb{R}_+$ as

$$f_{\text{KDE}}(x) = (Nh^2)^{-1} \sum_{i=1}^{N} \phi\left(\frac{x_i - x}{h}\right). \tag{1}$$

We utilize the symmetric Gaussian kernel $\phi : \mathbb{R}^2 \to \mathbb{R}_+$ :

$$\phi(x) = (2\pi h)^{-1} \exp\bigl(-\|x\|^2 / 2\bigr). \tag{2}$$

In the definitions above, $h \in \mathbb{R}_+$ is the KDE bandwidth in pixels, the choice of which determines the fidelity of $f_{\text{KDE}}$. We found that the choice of $h$ is not crucial (within a reasonable range) and barely influences the subsequent GMM fitting routine.

For the GMM fitting, we then utilize the set of maximizers $\arg\max f_{\text{KDE}}$ as the initialization for the component means. A classical algorithm to find $\arg\max f_{\text{KDE}}$ is the mean shift algorithm [8]. However, it is computationally demanding and there is no strict requirement for the resulting points to be exact maximizers, as they only serve as an initialization to the subsequent fitting algorithm. Thus, we resort to evaluating $f_{\text{KDE}}$ on the regular Cartesian grid $\Omega = \{1, \dots, W\} \times \{1, \dots, H\}$ of pixel centers for an image with dimensions $W \times H$ and find maxima on this grid in a local neighborhood. In particular, we find the points $\tilde{\mathcal{X}}_{\max} = \{x \in \Omega : f_{\text{KDE}}(x) > \zeta, f_{\text{KDE}}(y) < f_{\text{KDE}}(x) \, \forall y \in \omega_r(x)\}$, where $\zeta \in \mathbb{R}_+$ is a threshold to avoid low-density regions and $\omega_r(x) = \{y \in \Omega : \|y - x\|_\infty \leq r\}$ is a window of radius $r \in \mathbb{N}$ centered around $x$. To have an upper bound on the components in the subsequent GMM fitting routine, we take the $2N_{\text{kpts}}$ points that score best under $f$: $\mathcal{X}_{\max} = \{x_i \in \tilde{\mathcal{X}}_{\max} : i \in \mathcal{I}\}$, where $\mathcal{I}$ are the $2N_{\text{kpts}}$ indices into $\tilde{\mathcal{X}}_{\max}$ such that $f_{\text{KDE}}(x_i) > f_{\text{KDE}}(x_j)$, for all $i \in \mathcal{I}$ and all $j \notin \mathcal{I}$.

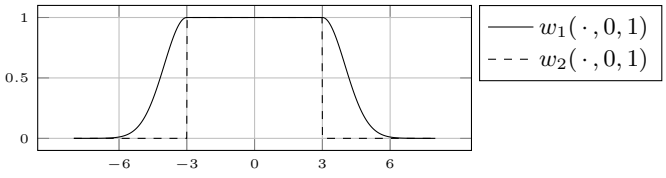



**Fig. 5:** Outlier weighting functions

### 3.4 Gaussian Mixture Model

To refine keypoint locations and assign scores, we leverage a GMM. We chose this model as its parameters inherently allow us to refine and score keypoints in an *interpretable* fashion. In particular, refined keypoint positions are given by the means of the components of the GMM, and for a specific component, *robustness* is measured via its weight and the *deviation* via its variance. As the negative influence of outliers needs to be considered in our setting, we propose a custom robust fitting algorithm, which we elaborate on in this section.

A $K$-component GMM

$$\sum_{k=1}^{K} \alpha_k G_{\mu_k, \Sigma_k} \tag{3}$$

is a convex combination of $K$ Gaussians

$$G_{\mu,\Sigma} : \mathbb{R}^d \to \mathbb{R}_+ : x \mapsto \frac{\exp\big(-\|x-\mu\|^2_{\Sigma^{-1}}\big)}{\sqrt{\det(2\pi\Sigma)}} \tag{4}$$

parameterized by their means $\mu \in \mathbb{R}^d$ and covariance matrices $\Sigma \in \mathbb{S}^d_+$. For proper normalization, the weight vector $\alpha = (\alpha_1, \ldots, \alpha_K)^\top$ must satisfy $\alpha \geq 0$, $\langle \alpha, \mathbb{1}_d \rangle = 1$. In our case, since we are fitting a GMM on the image domain $d = 2$, we restrict ourselves to isotropic covariance matrices of the form $\Sigma_k = \sigma_k^2 \mathrm{Id}_2$. We discuss the implications of this choice later and, with slight abuse of notation, denote with $G_{\mu,\sigma}$ a Gaussian with mean $\mu$ and covariance matrix $\sigma^2\mathrm{Id}$.

The most widely used estimator for the parameters $(\alpha_k, \mu_k, \sigma_k)_{k=1}^K$ maximizes the likelihood of the data $(x_i)_{i=1}^N$ (equivalently, minimizes the negative-log-likelihood):

$$\min_{(\alpha_k,\mu_k,\sigma_k)_{k=1}^K} \sum_{i=1}^{N} -\log\left(\sum_{k=1}^{K} \alpha_K G_{\mu_k,\sigma_k}(x_i)\right). \tag{5}$$

The EM algorithm is a popular algorithm to solve this problem iteratively: Given the estimates $\alpha_k^{(j)}, \mu_k^{(j)}, \sigma_k^{(j)}$ at the $j$-th iteration, the expectation step computes the *responsibilities* $\gamma_{k,i}^{(j)}$ of the $k$-th component w.r.t the $i$-th data point as

$$\gamma_{k,i}^{(j)} = \frac{\alpha_k^{(j)} w(x_i, \mu_k^{(j)}, \sigma_k^{(j)}) G_{\mu_k^{(j)},\sigma_k^{(j)}}(x_i)}{\sum_{k=1}^{K} \alpha_k^{(j)} w(x_i, \mu_k^{(j)}, \sigma_k^{(j)}) G_{\mu_k^{(j)},\sigma_k^{(j)}}(x_i)}. \tag{6}$$

Later, we will show our choice of the weighting function $w : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}_+$ to deal with outliers. In the standard EM algorithm, $w \equiv 1$.

With the current responsibilities $\gamma_{k,i}^{(j)}$, the maximization step amounts to updating

$$\begin{aligned} \alpha_k^{(j+1)} &= \frac{1}{N} \sum_{i=1}^{N} \gamma_{k,i}^{(j)}, \\ \mu_k^{(j+1)} &= \frac{\sum_{i=1}^{N} \gamma_{k,i}^{(j)} x_i}{\sum_{i=1}^{N} \gamma_{k,i}^{(j)}}, \\ (\tilde{\sigma}_k^{(j+1)})^2 &= \frac{\sum_{i=1}^{N} \gamma_{k,i}^{(j)} \left\| x_i - \mu_k^{(j+1)} \right\|^2}{\sum_{i=1}^{N} \gamma_{k,i}^{(j)}}. \end{aligned} \tag{7}$$

To avoid degenerate cases, we regularize the variances by adding a small positive scalar:

$$\sigma_k^{(j+1)} = \tilde{\sigma}_k^{(j+1)} + \epsilon, \tag{8}$$

where $\epsilon > 0$ is a tunable parameter, which should be set as small as possible while retaining stability.

In the fitting procedure, outliers present a major difficulty. They are represented by keypoints that have been detected in only few of the warped images, even just one, and should not disturb the clustering process. We tackle this by choosing $w = w_1$, where

$$w_1(x, \mu, \sigma) = \begin{cases} 1 \text{ if } \|x - \mu\| < 3\sigma, \\ \exp\left(-(\|x - \mu\| - 3\sigma^2)^2 / (2\sigma^2)\right) \text{ else.} \end{cases} \tag{9}$$

Thus, points that are far from a component mean are adaptively down-weighted based on the distance to the center as well as the variance of the component. To facilitate proper localization after the GMM has been fit using the adapted EM algorithm, we run additional iterations ignoring points that are outside the $3\sigma$ radius, i.e. we set $w = w_2$, where

$$w_2(x, \mu, \sigma) = \begin{cases} 1 & \text{if } \|x - \mu\| < 3\sigma, \\ 0 & \text{else.} \end{cases} \tag{10}$$

In addition, we perform a simple component selection scheme: recall that the component means are initialized to the KDE maxima, $\mathcal{X}_{\max}$. However, the KDE typically slightly overestimates the number of clusters; during the EM algorithm, components might "merge" to the same location. In this case, the scores derived from the weights of the GMM would systematically be too low as the weights are shared between two components. To avoid this, we drop concentric components based on the distance of their means: Let $k, l \in \mathbb{N}, k < l$ be two component indices at iteration $j$. If $\|\mu_k^{(j)} - \mu_l^{(j)}\| < \nu$, we discard the $k$-th component. Here, $\nu \in \mathbb{R}_+$ defines the minimal distance between component means.

Note that our method is not necessarily restricted to isotropic covariance matrices; we chose this as it makes the interpretation of the score straightforward.

**Table 1:** Comparison on HPatches v-set - keypoints budget 2048.

| | Rep @ ↑ | | | Rep MNN @ ↑ | | | MMA @ ↑ | | | MS @ ↑ | | | Hom. Acc. AUC @ ↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 px | 2 px | 3 px | 1 px | 2 px | 3 px | 1 px | 2 px | 3 px | 1 px | 2 px | 3 px | 1 px | 3 px | 5 px |
| Harris [11] | 0.215 | **0.434** | **0.553** | 0.215 | 0.398 | 0.458 | — | — | — | — | — | — | — | — | — |
| + GMM-IKRS | **0.229** | 0.426 | 0.532 | **0.229** | **0.421** | **0.504** | — | — | — | — | — | — | — | — | — |
| DoG [15] | 0.231 | 0.363 | 0.442 | 0.231 | 0.362 | 0.428 | — | — | — | — | — | — | — | — | — |
| + GMM-IKRS | **0.275** | **0.433** | **0.517** | **0.275** | **0.431** | **0.502** | — | — | — | — | — | — | — | — | — |
| DISK [37] | 0.211 | 0.388 | 0.464 | 0.211 | 0.387 | 0.460 | 0.352 | 0.621 | 0.709 | 0.206 | 0.352 | 0.398 | 0.092 | 0.299 | 0.420 |
| + GMM-IKRS | **0.257** | **0.425** | **0.499** | **0.257** | **0.425** | **0.494** | **0.409** | **0.647** | **0.727** | **0.248** | **0.380** | **0.422** | **0.110** | **0.326** | **0.444** |
| SuperPoint [9] | 0.197 | 0.392 | 0.502 | 0.197 | 0.383 | 0.466 | 0.242 | 0.483 | 0.595 | 0.159 | 0.314 | 0.383 | 0.121 | 0.360 | 0.488 |
| + GMM-IKRS | **0.247** | **0.446** | **0.545** | **0.247** | **0.445** | **0.538** | **0.312** | **0.531** | **0.627** | **0.211** | **0.353** | **0.413** | **0.144** | **0.382** | **0.507** |
| R2D2 [24] | 0.163 | 0.354 | 0.459 | 0.163 | 0.354 | 0.458 | 0.261 | 0.520 | 0.625 | 0.102 | 0.191 | 0.221 | 0.069 | 0.253 | 0.367 |
| + GMM-IKRS | **0.205** | **0.388** | **0.486** | **0.205** | **0.388** | **0.484** | **0.308** | **0.537** | **0.629** | **0.121** | **0.194** | **0.222** | **0.078** | **0.264** | **0.376** |
| MD-Net [27] | 0.193 | 0.380 | 0.466 | 0.193 | 0.378 | 0.462 | 0.329 | 0.614 | 0.717 | 0.167 | 0.296 | 0.339 | 0.140 | 0.348 | 0.454 |
| + GMM-IKRS | **0.237** | **0.406** | **0.486** | **0.237** | **0.406** | **0.483** | **0.388** | **0.639** | **0.729** | **0.199** | **0.308** | **0.345** | **0.143** | **0.356** | **0.470** |

An alternative would be to fit full covariance matrices and base the localization score on the largest singular value.

The resulting *robustness* score for the k-th refined keypoint is given by the number of points within a $3\sigma$ radius of its gaussian component as:

$$robustness_k = \sum_{i=1}^{N} w_2(x_i, \mu_k, \sigma_k), \tag{11}$$

while the *deviation* in pixels is defined as:

$$deviation_k = 6\sigma_k. \tag{12}$$

## 4 Experiments

### 4.1 Implementation Details

We apply the same set of augmentations to each image:

- isotropic scaling $\{1.5, 1.25, 0.75, 0.5\}$,
- anisotropic scaling along $x$, $y$ $\{1.5, 1.25, 0.75, 0.5\}$,
- anisotropic shear along $x$, $y$ $\{+0.2, -0.2, +0.6, -0.6\}$.

These result in a set of 21 images, including the original one. We found beneficial to add a slight Gaussian noise to all the warped images. This prevents weak noise patterns in the original image from triggering repeated detections.

In all the experiments we extract $N_{\text{kpts}} = 2048$ keypoints from each image. For the KDE, we choose the bandwidth $h = 0.5\,\text{px}$ and a window radius of $r = 3\,\text{px}$ to identify the maxima $\mathcal{X}_{\max}$. We initialize the GMM component means with the entries of $\mathcal{X}_{\max}$ and choose the initial variance such that the diameter at $3\sigma$ corresponds to $2\,\text{px}$ and set $\nu = 0.1\,\text{px}$. At each iteration, we also clamp the maximum variance for each component such that its diameter at $3\sigma$ does not exceed $10\,\text{px}$. In addition, we consider the rare case that clusters may contain more than one keypoint extracted from the same image. This may happen when

**Table 2:** Comparison on IMC [12] stereo, keypoints budget 2048.

| | Validation set | | | Test set | | |
|---|---|---|---|---|---|---|
| | Rep@3 px ↑ | N. inliers ↑ | mAA@10° ↑ | Rep@3 px ↑ | N. inliers ↑ | mAA@10° ↑ |
| DISK [37] | 0.493 | 438 | 0.707 | 0.452 | 393 | 0.503 |
| + GMM-IKRS | **0.507** | **469** | **0.722** | **0.468** | **426** | **0.524** |
| SuperPoint [9] | 0.350 | 154 | 0.270 | 0.359 | 176 | 0.282 |
| + GMM-IKRS | **0.380** | **182** | **0.319** | **0.391** | **210** | **0.318** |
| R2D2 [24] | 0.377 | 117 | 0.307 | 0.388 | 134 | 0.271 |
| + GMM-IKRS | **0.396** | **120** | **0.313** | **0.406** | **135** | **0.276** |
| MD-Net [27] | 0.357 | 144 | 0.470 | 0.396 | 199 | 0.401 |
| + GMM-IKRS | **0.370** | **152** | **0.484** | **0.411** | **211** | **0.412** |

**Table 3:** Comparison on IMC [12] multiview, keypoints budget 2048.

| | Validation set | | | | Test set | | | |
|---|---|---|---|---|---|---|---|---|
| | 3D pts. ↑ | Track L. ↑ | N. Inliers ↑ | mAA@10° ↑ | 3D pts. ↑ | Track L. ↑ | N. Inliers ↑ | mAA@10° ↑ |
| DISK [37] | 2195 | 6.032 | 450 | 0.842 | 2183 | 5.660 | 404 | 0.729 |
| + GMM-IKRS | **2217** | **6.088** | **482** | **0.850** | **2203** | **5.736** | **438** | **0.736** |
| SuperPoint [9] | 1332 | 4.330 | 163 | 0.568 | 1522 | 4.356 | 181 | 0.596 |
| + GMM-IKRS | **1467** | **4.488** | **193** | **0.601** | **1685** | **4.466** | **217** | **0.623** |
| R2D2 [24] | 1022 | 4.702 | 127 | 0.547 | 1287 | 4.462 | 139 | 0.548 |
| + GMM-IKRS | **1044** | **4.710** | **131** | **0.556** | **1309** | **4.473** | **141** | **0.560** |
| MD-Net [27] | 1108 | 5.002 | 154 | 0.727 | 1474 | 5.020 | 205 | 0.692 |
| + GMM-IKRS | **1133** | **5.053** | **162** | **0.740** | **1511** | **5.072** | **216** | **0.704** |

two badly localized clusters of keypoints partially intersect and the GMM uses a single component to fit both. In this case, we correct the cluster weight by removing the contributions from the duplicate keypoints.

For all the deep methods, we sample the descriptors from their dense descriptor volume at the integer locations of the refined keypoints. We run our pipeline with exactly the same parameters regardless of the keypoint detector used, which confirms the generality of our process.

### 4.2 HPatches

We test our framework on the popular HPatches [3] dataset. We run GMM-IKRS on top of several recent deep local feature extractors as well as hand-crafted classical keypoint detectors which have been widely used in the past. To test the performances of our method when dealing with viewpoint changes, we focus our experiments on the $v$ set of HPatches, which contains photos of planar scenes taken from different viewpoints. Each scene contains 1 reference image and 5 source images of different viewpoints. We run all the methods using the code provided by the authors and use OpenCV [6] RANSAC with 3.0 px threshold and 100k iterations to recover the homographies. As often done in the literature, we extract a maximum of 2048 keypoints from each image and match the descriptors using a simple mutual nearest neighbor (MNN) search, without any minimum score or ratio test. The results of our evaluations are reported in Table 1, where

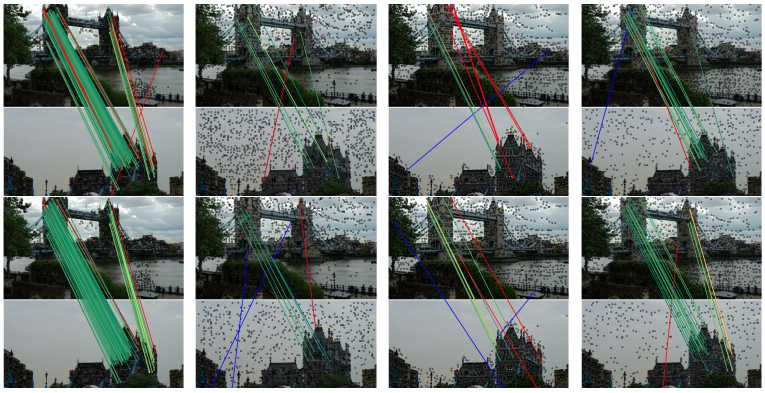
**Table 4:** Ablations on HPatches v-set.

| Method | Rep@3 px↑ | Rep MNN@3 px↑ | MMA@3 px ↑ | MS@3 px ↑ | s per img |
|---|---|---|---|---|---|
| DISK [37] | 0.464 | 0.460 | 0.709 | 0.398 | 0.03 |
| + 8 warps | 0.484 | 0.479 | 0.722 | 0.412 | 0.43 |
| + 14 warps | 0.493 | 0.488 | **0.732** | 0.416 | 0.69 |
| **+ 20 warps** | **0.499** | **0.494** | 0.727 | **0.422** | 0.98 |
| + 20 warps w/o outlier rejection | 0.483 | 0.478 | 0.724 | 0.395 | 0.98 |
| + 20 warps KDE only | 0.473 | 0.468 | 0.719 | 0.398 | 0.78 |
| + 20 warps round int | 0.494 | 0.489 | 0.726 | 0.417 | 0.98 |
| DISK [37] + 20 heatmap warps | 0.488 | 0.482 | 0.726 | 0.413 | 0.65 |

we compare the performance of each method with or without our refinement for various metrics at different pixel thresholds:

- **Repeatability**: Ratio between the number of keypoints that, from any of the two images, project close to a keypoint in the other and the total number of keypoints in the area of overlap.
- **Repeatability-Mutual-Nearest-Neighbor (MNN)**: This is a metric we propose to compensate for the overestimates of the standard repeatability metric in case of dense keypoint detections. A keypoint from image A is considered MNN-repeatable, if it is both repeatable and forms a mutual-nearest-neighbor pair in both images with the same keypoint from image B. The repeatability-MNN metric is then computed as the ratio between the sum of the number of mnn-repeatable keypoints from the two images and the total number of keypoints in the overlap area. This metric better relates to the pairwise descriptor matching task and upper-bounds to the standard repeatability in the case of well spaced keypoints.
- **Mean Matching Accuracy**: Ratio between the number of correct matches and the number of proposed matches.
- **Matching Score**: Ratio between the number of correct matches and the average number of detected keypoints in the overlap area.
- **Homography Accuracy AUC**: Area under the curve of the fraction of image pairs where the relative homography could be recovered with an average corner error lower than a specified threshold, evaluated at 0.1px steps.

The table shows a consistent improvement, with a single exception, over all the computed metrics at all pixel thresholds. Harris [11], due to its densely detected keypoints, tricks the repeatability measure obtaining higher scores at 2 and 3 pixels, but falls behind once evaluated with the more robust repeatability-MNN metric. The Homography Accuracy AUC, which is the most important score for many downstream tasks, is also improved when using our refined keypoints. The performance boost obtained using the refined keypoints does not only come from the sub-pixel accuracy of our method, as better discussed in Sec 5, but also from a more robust selection of keypoints. This is confirmed by the improved repeatabilities obtained for the already sub-pixel-accurate DoG [15] detector.



**(a)** DISK [37] **(b)** SuperPoint [9] **(c)** R2D2 [24] **(d)** MD-Net [27]

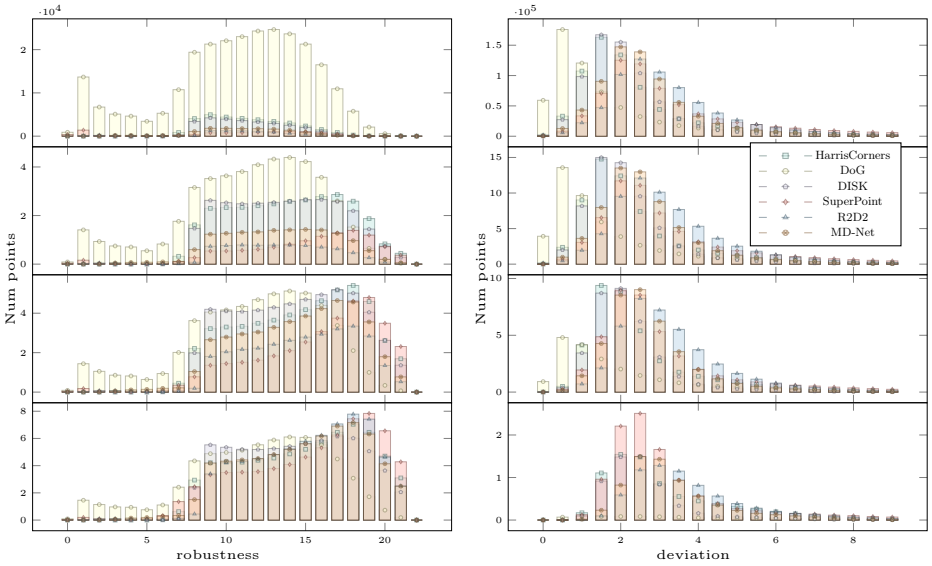
**Fig. 6:** Visual comparison between original methods (top) and their GMM-IKRS refined version (bottom) on a challenging image pair from Image Matching Challenge [12]. The RANSAC inliers are color coded depending on the reprojection error, from green (0px) to yellow (5px). Matches are shown in red when their reprojection error is larger than 5 px, and in blue when the depth is not available. For this image pair, all the methods except SuperPoint [9] obtain more and better localized matches.

### 4.3 Image matching benchmark

To evaluate the generalization of our framework to more challenging scenarios, we run the deep learning based methods, with and without our refinement framework, on the phototourism set of the 2021 Image Matching Challenge (IMC) [12]. Additionally to the stereo pose recovery task, the benchmark evaluates the local features performance in the more practical scenario of multi-view 3D reconstruction. Each of the scenes (3 validation set, 9 test set) contains 100 pictures of famous landmarks, captured by tourist with different cameras, from various angles and under diverse lighting conditions We run each method single-scale, using the matching and filtering parameters obtained from the online leaderboard, where available. The numerical results are shown in Table 2 and Table 3, where the methods are evaluated for:

- **mAA**: Mean Average Accuracy. This is the main benchmark metric, computed as the area under the curve of the fraction of relative poses recovered within a maximum error in degrees, evaluated at 1° steps.
- **N. inliers**: Average number of inlier matches after the robust fit.
- **3D pts.**: Average number of reconstructed 3D points.
- **Track L.**: Average 3D reconstruction track length.

For all methods, every metric improves, thus confirming the validity of our refinement approach for real tasks. SuperPoint [9], with a more than 10% improvement in stereo mAA and 7% in multiview mAA, is the method that most strongly benefits from our framework, followed by DISK [37] and MD-Net [27].




**(a)** Cluster *robustness* distribution at different *deviation* thresholds. From top to bottom, the maximum *deviation* is set to 1 px, 2 px, 3 px, ∞.

**(b)** Cluster *deviation* distribution at different *robustness*. From top to bottom, the minimum *robustness* is set to 5, 10, 15, 20 (max is 21).

**Fig. 7:** Statistical analysis of keypoints extracted from different detectors on the HPatches [3] v-set images. Better seen in color.

### 4.4 Statistical analysis

In addition to the metrics evaluated for the benchmark, our framework permits to shed more light on the different detector behaviours. In Figure 7, we report the distributions of *robustness* and *deviation* for all the keypoints extracted from HPatches. We recall that a *deviation* value of 1 px indicates a cluster whose diameter at $3\sigma$ measures 1 px. From the histograms we can notice how DoG [15], which obtains similar repeatabilities to other methods on the HPatches evaluations, is characterized by very different *robustness* and *deviation* distributions. Looking at the top-left chart, which shows the *robustness* distribution for well localized keypoints with *deviation* up to 1 px, it can be noticed how DoG dominates all the other methods; not only at higher *robustness* values, but also at lower ones. This leads to the conclusion that DoG is very good at finding well localized keypoints, but it does not excel at finding robust ones. When relaxing the requirements for localization accuracy to 3 px, shown in the 3rd row, methods like DISK, Harris and MD-Net are able to detect keypoints with higher *robustness* scores, that is points that are very likely to be detected again in another image. In the last row of the same chart, SuperPoint emerges as the method able to find the largest number of very robust keypoints. The distributions on the right chart describe instead how well localized keypoints with different degrees of robustness are. If looking for very robust keypoints, we should focus on the bottom chart, which shows the distributions of keypoints with *robustness* score of at least 20/21. In

this case, DoG results the worst performing method, whereas SuperPoint is the one able to find the largest number of keypoints with *deviation* lower than 4 px.

In Figure 4 we show for each method the best refined keypoint across 5 different HPatches scenes. We can notice a tendency of classical method to prefer hard contrast regions. SuperPoint, in accordance with its specific pretraining, shows a clear preference for corner-like structures, while DISK, R2D2 and MD-Net seem more oriented toward blobs. In particular, DISK appears able to find robust keypoints even in less contrasted regions (rows 2 and 5).

## 5 Ablation studies

To validate our design choices, we conduct further experiments using DISK [37], which was found to be the best performing method on IMC [12]. The results are shown in Table 4. On the upper section of the table, we compare our framework performance using different numbers of augmentations. In the +8 *warps* row, we use only the smallest shears and anisotropic scalings from the set of augmentations, while in the +14 *warps* row we reintroduce the isotropic scaling and stronger shears. When comparing results with different augmentation numbers, we can observe that the +20 *warps* row yields the best trade off in term of performances, keeping the computational time below one second. The mean keypoint refinement shift in this case is 0.34 px. In the *w/o outlier rejection* line we report the results obtained dropping our robustified GMM modification and, for the *warps KDE only* entry, we directly use the KDE local maxima as keypoints, skipping the GMM fit. For a more fair comparison against the discrete-pixel-accurate KDE result, we also report the results obtained rounding our GMM-IKRS keypoints coordinate to the closest integer. Comparing against these lines, we can see how our robust GMM-fit scores and refines the keypoint locations better and beyond a simple sub-pixel refinement. Finally, as a baseline comparison, we report as *DISK* +20 *heatmap warps* the results obtained aggregating the unwarped DISK heatmaps directly and then detecting the keypoints. This approach improves the original DISK results, but not as much as our GMM-IKRS pipeline.

## 6 Conclusion

In this work, we presented GMM-IKRS, a general framework capable of refining and evaluating the keypoints from any detector. The two scores assigned by our pipeline, *robustness* and *deviation*, characterize each keypoint in an interpretable manner and can be compared across different methods. This permits an in-depth analysis of qualities across different detectors, which would otherwise not be possible. The outcome of our experiments on the HPatches v-set confirmed the validity of our method, while the results of the Image Matching Challenge demonstrated its refinement capabilities for 3D reconstruction, even under challenging conditions. For the future, an interesting research direction could consist in the use of our method for the generation of sub-pixel keypoints to be used as ground truth to train, in a teacher-student fashion, a deep detector.

## Acknowledgement

This work has been supported by the FFG, Contract No. 881844: "Pro²Future is funded within the Austrian COMET Program Competence Centers for Excellent Technologies under the auspices of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs and of the Provinces of Upper Austria and Styria. COMET is managed by the Austrian Research Promotion Agency FFG."

## References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54**(10), 105–112 (2011)
2. Aulinas, J., Petillot, Y., Salvi, J., Lladó, X.: The slam problem: a survey. Artificial Intelligence Research and Development pp. 363–371 (2008)
3. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: CVPR. pp. 5173–5182 (2017)
4. Barroso-Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K.: Key. net: Keypoint detection by handcrafted and learned cnn filters. In: ICCV. pp. 5836–5844 (2019)
5. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV. pp. 404–417 (2006)
6. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
7. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision – ECCV 2010. pp. 778–792. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
8. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE TPAMI **24**(5), 603–619 (2002). https://doi.org/10.1109/34.1000236
9. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: CVPRW. pp. 224–236 (2018)
10. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint description and detection of local features. In: CVPR. pp. 8092–8101 (2019)
11. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. Alvey Vision Conference (1988)
12. Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image Matching across Wide Baselines: From Paper to Practice. IJCV (2020)
13. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: 2011 International Conference on Computer Vision. pp. 2548–2555 (2011). https://doi.org/10.1109/ICCV.2011.6126542
14. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: Lightglue: Local feature matching at light speed. arXiv preprint arXiv:2306.13643 (2023)
15. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV. vol. 2, pp. 1150–1157 vol.2 (1999). https://doi.org/10.1109/ICCV.1999.790410
16. Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., Li, S., Fang, T., Quan, L.: Aslfeat: Learning local features of accurate shape and localization. In: CVPR. pp. 6589–6598 (2020)

17. Mapillary: Opensfm, https://opensfm.org/
18. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing **22**(10), 761–767 (2004)
19. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. arXiv preprint arXiv:1705.10872 (2017)
20. Morel, J.M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM journal on imaging sciences **2**(2), 438–469 (2009)
21. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
22. Ono, Y., Trulls, E., Fua, P., Yi, K.M.: Lf-net: Learning local features from images. arXiv preprint arXiv:1805.09662 (2018)
23. Pakulev, K., Vakhitov, A., Ferrer, G.: Ness-st: Detecting good and stable keypoints with a neural stability score and the shi-tomasi detector. In: ICCV. pp. 9578–9588 (2023)
24. Revaud, J., Weinzaepfel, P., de Souza, C.R., Humenberger, M.: R2D2: repeatable and reliable detector and descriptor. In: NeurIPS (2019)
25. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European conference on computer vision. pp. 430–443. Springer (2006)
26. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: ICCV. pp. 2564–2571 (2011). https://doi.org/10.1109/ICCV.2011.6126544
27. Santellani, E., Sormann, C., Rossi, M., Kuhn, A., Fraundorfer, F.: Md-net: Multi-detector for local feature extraction. In: ICPR. pp. 1 – 6 (2022)
28. Santellani, E., Sormann, C., Rossi, M., Kuhn, A., Fraundorfer, F.: S-trek: Sequential translation and rotation equivariant keypoints for local feature extraction. In: ICCV. pp. 9728–9737 (October 2023)
29. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: CVPR. pp. 4938–4947 (2020)
30. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al.: Benchmarking 6dof outdoor visual localization in changing conditions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8601–8610 (2018)
31. Schenk, F., Fraundorfer, F.: Robust edge-based visual odometry using machine-learned edges. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1297–1304. IEEE (2017)
32. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
33. Shi, J., Tomasi: Good features to track. In: CVPR. pp. 593–600 (1994)
34. Svärm, L., Enqvist, O., Kahl, F., Oskarsson, M.: City-scale localization for cameras with known vertical direction. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(7), 1455–1461 (2017). https://doi.org/10.1109/TPAMI.2016.2598331
35. Tian, Y., Fan, B., Wu, F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: CVPR. pp. 661–669 (2017)
36. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Now Publishers Inc (2008)
37. Tyszkiewicz, M., Fua, P., Trulls, E.: Disk: Learning local features with policy gradient. NeurIPS **33** (2020)

38. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European conference on computer vision. pp. 467–483. Springer (2016)
39. Zhou, H., Yuan, Y., Shi, C.: Object tracking using sift features and mean shift. Computer Vision and Image Understanding **113**(3), 345–352 (2009). https://doi.org/https://doi.org/10.1016/j.cviu.2008.08.006, special Issue on Video Analysis