

Detect by Track: Making Detector-Free Matcher Trackable

Yusuke Sekikawa¹ , Hideki Shirai, Ruka Eto, Yuzhe Hao , Kengo Mitsui, and
Nakamasa Inoue² 

DENSO IT Laboratory, Inc., Japan¹, Institute of Science Tokyo, Japan²

Abstract. Detector-free matchers (e.g., EDM, JamMa) offer a strong accuracy–latency trade-off for pairwise matching. However, their *implicit* keypoint selection—where keypoints emerge as a by-product of matching—often leads to *multi-view inconsistency* and fragmented tracks. This limitation hinders applications that require connected tracks, such as object tracking, SfM, and SLAM. Prior attempts to mitigate this issue rely on coarsely quantized matches combined with heavy post-processing for multiple images; thus, they are typically limited to batch-based SfM pipelines. We take a different approach and directly address multi-view inconsistency through a *Detect-by-Tracking (DeT)* mechanism. Given an image pair I_A, I_B , and a sub-pixel query q_A , *DeT* steers the local similarity matrix of a detector-free matcher so that the row-wise softmax represents the correspondence probability of an *arbitrary* sub-pixel query q_A . This produces connected tracks across views *without* heavy post-processing, while preserving the core strength of detector-free matching: implicit keypoint selection driven by matching. Moreover, *DeT* can be incorporated into off-the-shelf matchers *without* additional training. Experiments demonstrate that *DeT* maintains the base detector-free matcher’s matching accuracy while producing connected tracks.

Keywords: Detector free · Tracking

1 Introduction

Motivation. Local feature tracks—repeated 2D observations of the same 3D point across images—are the backbone of object tracking (OT), Visual Odometry (VO), Structure from Motion (SfM), and visual Simultaneous Localization and Mapping (SLAM). Detector-based pipelines detect keypoints (e.g. SuperPoint [3], XFeat [23]) and then match descriptors using attention-based matchers (e.g. SuperGlue [26], LightGlue [18]). Such detection-based pipelines naturally produce connected tracks and are OT/VO/SfM/SLAM-friendly. However, they often lag behind modern detector-free matchers in the accuracy-latency trade-off [21], due in part to keypoint selection that is not fully aligned with the matching objective and to attention-based matching, whose computational costs scale quadratically with the number of points.

Why not just go detector-free? Recent detector-free matchers (e.g., LoFTR [27], EDM [16], JamMa [21]) often outperform detector-based methods in both two-view accuracy and runtime. Their dense cross-image interactions in *image* space and matching-oriented implicit keypoint detection (*detection by pair matching*) yield strong accuracy,

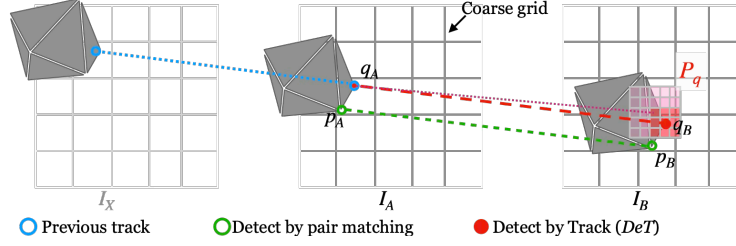


Fig. 1: Concept of Detect by Tracking. Existing detector-free matchers suffer from *multi-view inconsistency*. The keypoint locations from *detect-by-pair-matching* in I_A vary ($p_A \neq q_A$) depending on its pair (I_X or I_B), because the keypoint is selected implicitly as a consequence of matching. This yields fragmented tracks. The proposed *Detect by Tracking (DeT)* detects the keypoints q_B , starting from the sub-pixel previous track q_A using connection probability map P_q which is computed by steering the similarity matrix of the detector-free matcher (Sec. 3).

and the coarse-to-fine pipeline scales favorably with the number of correspondences. However, implicit keypoint selection also has a downside: keypoints are *not* detected explicitly but are *determined as a consequence of matching between paired images* instead, leading to keypoint locations varying with their counterparts. This *multi-view inconsistency* breaks track formation and renders standard OT/SfM/SLAM pipelines unusable (Figure 1).

Existing remedies. Pioneering work [11, 15] integrates detector-free matchers into SfM through coarse quantization, followed by rich refinement. For example, Detector-Free SfM [11] first enforces coarse consistency on a quantized grid and then applies transformer-based optimization to refine poses and points. Although accurate, such pipelines are computationally intensive and introduce additional post-processing stages. As a result, these methods are difficult to apply to incremental SLAM, VO, and OT, where one needs to update the camera (and 3D point) incrementally.

Our idea: Detect by Tracking. We seek an approach that *directly outputs sub-pixel-accurate connected tracks* from a detector-free matcher while preserving its advantages: dense cross-image interactions in image space and matching-oriented implicit keypoint selection. To this end, we shift from *detect by pair matching* to *detect-by-tracking (DeT)*. We realize *DeT* by constructing a connection probability map P_q from the previous sub-pixel track q_A (Figure 1).

Given an image pair (I_A, I_B) , detector-free matchers compute a local similarity matrix S from feature crops $F_A^{\bar{p}_A}, F_B^{\bar{p}_B}$ centered at coarse matches $\bar{p}_A, \bar{p}_B$. Keypoints are then obtained by finding the maximum of the dual-softmax of S (Figure 2). In other words, keypoints are detected as a consequence of *pair matching*. We exploit the fact that S is *translation equivariant*, and its row-wise softmax represents the local matching probabilities *conditioned on the actual correspondence lying within the crop around $\bar{p}_B$* . Our idea is to *steer* the anchor (F_A) and target (F_B) crops so that the resulting matrix yields the probability map P_q from the previous track’s sub-pixel keypoint location q_A to the pixels in I_B , which can then be used to detect the corresponding pixel q_B . In other

words, keypoints q_B are detected as a consequence of tracking. Iterating across views yields *connected sub-pixel tracks* (Figure 4).

Through this novel steering mechanism, *DeT* ensures track continuity while preserving the key advantage of detector-free methods: the keypoint is still selected as a consequence of matching. Moreover, *DeT* can be plugged into existing matchers, such as EDM [16] and JamMa [21], via an inference-time modification, without requiring retraining (Figure 4). Experiments demonstrate that this zero-shot conversion maintains the strong matching capability of the original detector-free matcher while producing connected tracks.

2 Related Work

We briefly review local correspondence methods, emphasizing multi-view consistency.

2.1 Detector-based methods

Classical pipelines follow a *detect–describe–match* paradigm: salient keypoints are first extracted, and then corresponding features are computed using a handcrafted or learned algorithm. SIFT [19] is a seminal method that uses a scale-space pyramid to detect distinctive keypoints, followed by the extraction of rotation- and scale-invariant descriptors. Learned detectors/descriptors such as SuperPoint [3], R2D2 [24], ALIKE/ALIKED [36, 37], and XFeat [23] are commonly paired with transformer-based matchers like SuperGlue [26] and LightGlue [18]. Recent work also explores direct multi-view matching [25, 35]; for example, CoMatcher [35] performs collaborative matching across image groups to produce reliable tracks. End-to-end multi-view approaches further couple matching to pose optimization [25]. While these detector-based approaches naturally support tracking via explicit keypoint anchors, they rely on detector repeatability and can struggle in low-texture or repetitive regions. Moreover, the runtime of attention-based matchers scales quadratically with the number of detected keypoints, making it difficult to reconstruct (semi-)dense structures directly.

2.2 Detector-free methods

Detector-free methods bypass explicit keypoint detection and instead predict dense or semi-dense correspondences. LoFTR [27] introduced a coarse-to-fine transformer architecture that is robust in challenging regions where detector-based methods fail. Subsequent work further improved the accuracy-efficiency tradeoff via hierarchical attention and sparsification (e.g., MatchFormer [30], ASpanFormer [1], Efficient-LoFTR [31]) and through linear-complexity scanning mechanisms based on state-space models (e.g., JamMa [21]). From a different perspective, RCM [20] introduces a conflict-free matching mechanism to mitigate the *problem of matching conflicts* in dense matching through a novel dynamic view-switching strategy. EDM [16] introduced *correlation injection* into the backbone to enhance cross-image feature extraction further, demonstrating an improved latency-accuracy tradeoff over Efficient-LoFTR [31].

These detector-free mechanisms offer several advantages over sparse detector-based pipelines, including an efficient coarse-to-fine pipeline and strong, matching-oriented implicit keypoint selection that utilizes rich features extracted through dense cross-image interactions in image space (Figure 2). Despite strong performance, deploying detector-free matchers in SfM/SLAM or tracking poses challenges due to *multi-view inconsistency*: partner-dependent keypoint selection fragments tracks and undermines multi-view consistency (Figure 1), which is critical for consistent tracking and 3D reconstruction.

This *multi-view inconsistency* issue also arises in dense matchers (e.g., RoMA [7], DKM [6]). A dense matcher generates dense sub-pixel warps $I_X \rightarrow I_A$; however, the $I_A \rightarrow I_B$ warp cannot be initialized from the previous track, leading to *disconnected* tracks at sub-pixel accuracy (similar to the semi-dense case; see Figure 1). We refer the reader to the discussion of the *Kaggle IMC2024 competition* [34] for a further detailed analysis of the *multi-view inconsistency* issues specific to dense matchers.

2.3 Using detector-free methods for SfM

Previous attempts to leverage detector-free methods for SfM [11, 15] rely on heavy post-processing. Detector-Free SfM [11] uses coarse discretization to avoid fragmented tracks, followed by transformer-based refinement. More recent work, Dense-SfM [15], uses Gaussian-splatting to improve reconstruction precision further. Although these pipelines achieve impressive results, particularly in texture-poor scenes, their coarse quantization at the initial stage necessitates costly refinement, limiting real-time applicability. Furthermore, these methods are specifically designed for SfM with batched inputs that process multiple images at once, making them difficult to apply to applications such as incremental SLAM, VO, or object tracking that require connected tracks during online estimation. Our goal differs: we aim to obtain connected tracks directly from the incremental application of pairwise matching. Specifically, we seek a method that avoids a task-specific post-processing stage and applies to a wide range of applications beyond batch SfM.

3 Method

In this section, we derive *DeT*, which makes detector-free matchers trackable. We first present an overview of detector-free matchers and formalize the tracking problem we address (Sec. 3.1). We then explain the core mechanism of *DeT* for creating connected tracks without explicit keypoint detection (Sec. 3.2). Finally, we discuss the computational cost of enabling the trackable mechanism introduced by *DeT* (Sec. 3.4).

3.1 Preliminaries

General view of a detector-free matcher. Given an image pair (I_A, I_B) , modern detector-free matchers, such as LoFTR [27], Efficient LoFTR [31], EDM [16], and JamMa [21], adopt a coarse-to-fine (C2F) pipeline (Figure 2). A strong backbone $\mathcal{M}_B$ —typically involving cross-image interaction via attention or the State Space Model (SSM)—extracts

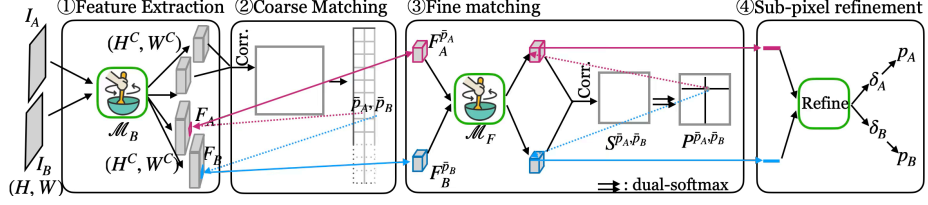


Fig. 2: General detector-free matcher. Recent efficient detector free matcher adopt coarse-to-fine matching mechanism. Fine-level matching is evaluated only in local windows centered at coarse matches $\bar{p}_A, \bar{p}_B$, enabling memory- and compute-efficient matching. $\mathcal{M}_{\{B, F\}}$ denotes mixing function (e.g., MLP-Mixer [28] or Attention [29]). Refer to Sec. 3.1 for detailed explanation.

coarse features $C_A, C_B \in \mathbb{R}^{H^C \times W^C}$ and fine features $F_A, F_B \in \mathbb{R}^{H^F \times W^F}$. Coarse matching produces locations $\bar{p}_A, \bar{p}_B$ in a stride Δ_c (8 px by default in the case of JamMA). Around these locations, fine-level crops $F_A^{\bar{p}_A}, F_B^{\bar{p}_B} \in \mathbb{R}^{\kappa \times \kappa}$ are extracted from F_A, F_B (fine stride $\Delta_f=2$ px). A local similarity matrix is then computed using a lightweight *mixing* function $\mathcal{M}_F$ followed by correlation,

$$S^{\bar{p}_A, \bar{p}_B} = \text{Corr}(\mathcal{M}_F(F_A^{\bar{p}_A}), \mathcal{M}_F(F_B^{\bar{p}_B})) \in \mathbb{R}^{\kappa^2 \times \kappa^2}, \quad (1)$$

The similarity matrix is converted into a probability matrix using dual softmax: $P^{\bar{p}_A, \bar{p}_B} = \text{softmax}_{\text{row}}(S^{\bar{p}_A, \bar{p}_B}) \odot \text{softmax}_{\text{col}}(S^{\bar{p}_A, \bar{p}_B})$. The fine-level correspondence is obtained by $\arg \max P^{\bar{p}_A, \bar{p}_B}$, after which a sub-pixel refiner predicts a sub-pixel offset, giving the two-view match (p_A, p_B) . This C2F design underpins the efficiency of detector-free matchers.

Multi-view inconsistency. Because keypoints are determined as a consequence of pairwise matching, the keypoint location detected in I_A depends on its partner image. Thus, a point in I_A paired with I_X may not map to the exact location when I_A is paired with I_B , causing track fragmentation ($q_A \neq p_A$ in Figure 1). Refer to [11, 12, 34] for more detailed discussions of this issue.

Notation. The rows and columns of $S^{\alpha, \beta} \in \mathbb{R}^{\kappa^2 \times \kappa^2}$ index the local coordinates of $\kappa \times \kappa$ in I_A (anchor) and I_B (target) around α and β , respectively. Let $S_d^{\alpha, \beta} \in \mathbb{R}^{\kappa \times \kappa}$ denote the row of $S^{\alpha, \beta}$ for the anchor offset $d=(d_x, d_y)$, and let $P_d^{\alpha, \beta} = \text{softmax}(S_d^{\alpha, \beta})$ denote the corresponding conditional distribution (Sec. 3.2 explains this) over the target pixels.

Problem setup. Given images $\{I_t\}_{t=1}^T$, we seek multi-view-consistent tracks $\{q_{j,t}\}$ for 3D points j representing the same 3D object, where t denotes the frame index. Without loss of generality, consider (I_A, I_B) and a sub-pixel track location q_A in I_A ; our goal is to compute its continuation q_B in I_B to obtain the connected tracks (Figure 4).

3.2 DeT: Detect by Tracking

We aim to output *connected, sub-pixel-accurate tracks* directly from a detector-free matcher while preserving its strengths: dense image-space interaction and matching-oriented implicit keypoint selection. Our key observation is that $P_d^{\alpha, \beta}$ can be interpreted

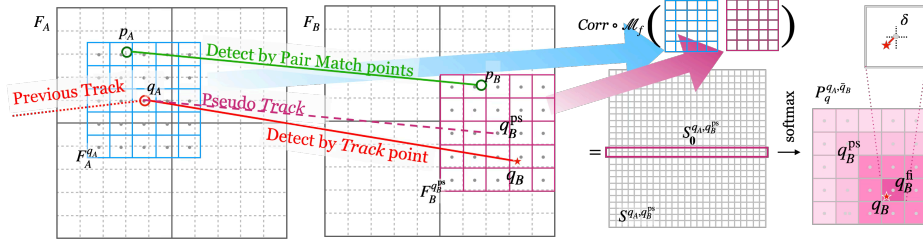


Fig. 3: Detect by Track Algorithm. i) *Steer target crop*: Find the nearest detect-by-pair-matching keypoint p_A and its pair p_B to compute the pseudo-track q_B^{ps} , then, steer the target crop to obtain $F_B^{q_B^{ps}}$ (size of $\kappa \times \kappa$). ii) *Steer anchor crop*: For each sub-pixel track anchor q_A (from previous image) extract $F_A^{q_A}$ (size of $\kappa \times \kappa$) from F_A using bilinear interpolation. iii) *Find fine correspondence*: Compute $S^{q_A, q_B^{ps}}$ as $\text{Corr} \circ \mathcal{M}_f \cdot (F_A^{q_A}, F_B^{q_B^{ps}})$, and extracts P_q as $\text{softmax}(S_{[0,0]})$, then select the fine-grid destination $\hat{q}_B$ as $\arg \max P_q$ (at resolution of fine stride $\Delta_f = 2$ px). iv) *Target fine refinement*: Apply sub-pixel regressor to get refined q_B .

as the probability of matching from the anchor pixel $\alpha + d$ to the local coordinates in I_B , conditioned on the ground-truth (GT) correspondence located within the crop around β . DeT constructs a matching probability map P_q for the previous sub-pixel anchor q_A by steering both the target crop (F_B) and the anchor crop (F_A).

Computing matching probability P_q for query point Recall the pairwise keypoint detection mechanism of the detector-free matcher (Equation (1), Figure 2). The *detect by pair matching* keypoints p_A and p_B are computed as the maximizers of $P \in \mathbb{R}^{\kappa^2 \times \kappa^2}$. In other words, not only the target keypoint (I_B), but also the anchor keypoint (I_A) are detected as a consequence of matching; therefore, the previous track q_A does not map to the detected keypoint p_A (Figure 3).

Notice that softmax of $S_d^{\bar{p}_A, \bar{p}_B}$, i.e., $P_d^{\bar{p}_A, \bar{p}_B}$, can be interpreted as the probability of a matching from the pixel $\bar{p}_A + d$ to the local pixels around $\bar{p}_B$, conditioned on the GT match q_B^{GT} inside the $\kappa \times \kappa$ around $\bar{p}_B$. Thus, we may be prompted to use the similarity matrix $S_d^{\bar{p}_A, \bar{p}_B}$ of the detector free matcher to find the corresponding matching point for the previous track q_A , which we aim to connect to I_B . However, two problems arise that prevent this. ① *Target point coverage*: The GT match may be out of range in the $\kappa \times \kappa$ region centered around $\bar{p}_B$. ② *Sub-pixel anchoring*: The probability map $P_d^{\bar{p}_A, \bar{p}_B}$ is defined only on the discrete (fine stride of Δ_f , typically 2 px) anchor pixel $p_A + d$, while we seek the probability map for the sub-pixel query q_A . To mitigate these two issues and realize the *detector-free tracker*, we formulate the algorithm for computing P_q , the probability of matching from the sub-pixel location q_A , by steering the target (F_B) and anchor (F_A) crops as follows:

① *Target point coverage*. To ensure that the GT match q_B^{GT} falls inside the receptive window of S (Figure 3-(i)), we use a pseudo-track to steer the target crop as follows:

$$q_B^{ps} = \Delta_f \cdot \text{round} \left(\frac{q_A + (p_B - p_A)}{\Delta_f} \right), \quad (2)$$

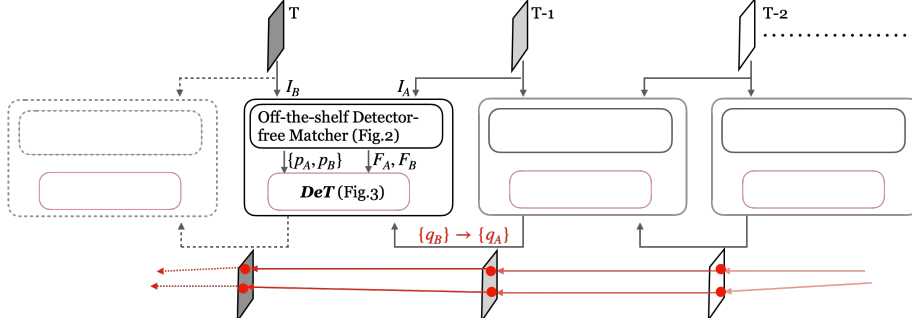


Fig. 4: Tracking pipeline. *DeT* can be integrated into off-the-shelf detector free matcher (e.g., JamMA [21]). Recursively using previous output q_B as new input q_A yields connected tracks.

where p_A is the closest key point for matching the detect key to q_A (i.e., the output of the conventional detector-free matcher). If p_A and q_A lie at similar depths and p_B is the true match of p_A , then, under translational motion, $q_B^{\text{GT}} \approx q_A + (p_B - p_A)$. We crop $F_B^{q_B^{\text{ps}}}$ centered on q_B^{ps} , which (with high probability) covers q_B^{GT} .

ii) *Sub-pixel anchoring.* While $\text{softmax}(S_d^{\alpha, \beta})$ provides a conditional distribution over I_B , the index d corresponds to anchor pixels only on the discrete grid of Δ_f resolution. To enable anchoring from an arbitrary sub-pixel location q_A , we exploit the translational equivariance of the matching probability map: for integer shifts d and the identity $\mathcal{M}_F$ (i.e, $S_d^{\alpha, \beta} = \text{Corr}(F_A^\alpha, F_B^\beta)$), then, $\text{softmax}(S_d^{\alpha, \beta}) = \text{softmax}(S_0^{\alpha-d, \beta})$. As shown in Figure 5, this relation remains empirically/approximately valid even for non-integer shifts and when $\mathcal{M}_F$ consists of a mixing function (e.g., JamMa mixes the cropped fine features with an MLP-Mixer before correlation). Hence, the probability map for the sub-pixel query q_A can be written as:

$$P_q = \text{softmax}(S_{q_A - p_A}^{p_A, q_B^{\text{ps}}}), \quad (3)$$

$$\approx \text{softmax}(S_0^{q_A, q_B^{\text{ps}}}), \quad (4)$$

where $S^{q_A, q_B^{\text{ps}}} = \text{Corr}(\mathcal{M}_F(F_A^{q_A}, F_B^{q_B^{\text{ps}}}))$ with $F_A^{q_A}$ is obtained by bilinear sampling around the sub-pixel location q_A . We want to emphasize the difference between the two equations. Equation (3) holds only for discrete q_A with a stride of Δ_f ; (4) extends this to arbitrary sub-pixel query locations q_A (Figure 3-(i)). This change in variable is the key mechanism enabling *DeT* to produce the connected track from the previous *sub-pixel* tracks.

Sub-pixel track update. We obtain a fine-grid estimate $q_B^{\text{fi}} = \arg \max P_q$ and refine it using the standard regression head to get the sub-pixel offset δ . The updated track is $q_B = q_B^{\text{fi}} + \delta$. Iterating across views yields connected sub-pixel tracks (Figure 4).

Newborn tracks (optional). To compensate for tracks lost due to FOV exits, occlusions, or missed detections, we optionally seed new tracks from high-confidence detect-by-pair-matching keypoints (maxima of P). In the demo provided in the supplement, we

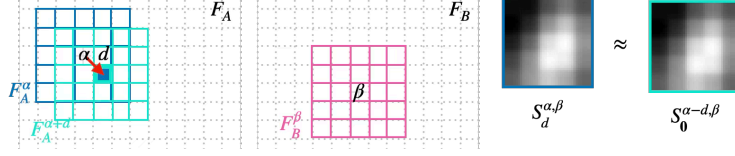


Fig. 5: Empirical translational equivariance of similarity matrix. Even when $\mathcal{M}_F$ consists of mixing function (e.g., MLP-Mixer [28] or Attention [29]) before correlation, the row-wise softmax of S , $S_0^{\alpha-d, \beta}$, remains approximately translationally equivariant with respect to shifts of the anchor crop $d = (d_x, d_y)$ of F_A . Thus shifting F_A by d approximately matches the probability vector obtained by reading the corresponding row for the local 2D coordinate d of S softmax($S_d^{\alpha, \beta}$).

present the results using this option. Note that it is also possible to use external keypoint detectors such as SIFT [19] or SuperPoint [3] instead.

3.3 Necessary condition of *DeT* to work

DeT first associates the previous track q_A with a *detect-by-pair-matching* (e.g., by vanilla JamMA) correspondence (p_A, p_B) using nearest-neighbor (NN) search. Then it predicts a pseudo-track $q_B^{\text{ps}} = \lfloor q_A + (p_B - p_A) \rfloor$. This *assumes* that the tracked point pair (q_A, q_B) undergoes a motion similar to (p_A, p_B) , i.e., $p_B - p_A \approx q_B - q_A$. This pseudo-track provides a coarse estimate of q_B and determines the center of the crop on the fine feature map P_q . Therefore, the *necessary condition* for *DeT* to succeed is: “*GT track q_B^{GT} lies within the κ^2 region centered on the pseudo-track q_B^{ps}* ”, i.e., $\|q_B^{\text{ps}} - q_B^{\text{GT}}\|_\infty < \kappa/2$.

This condition can be violated under the presence of severe 1) depth discontinuities, 2) motion that changes the depth, 3) non-rigid deformation, 4) occlusion, or 5) keypoint-localization errors (e.g., of JamMA). If this assumption fails, the cropped feature no longer contains the correct target location and *DeT* cannot recover the track. Conversely, if q_A and p_A have similar depth, camera movement keeps the distance to the corresponding 3D points, and match (p_A, p_B) is accurate, then $q_B^{\text{GT}} \approx q_A + (p_B - p_A)$. In the ideal case, the equality holds even when q_A and p_A are far apart in the image.

Because violations arise from multiple interacting factors, it is difficult to derive a complete analytical characterization. Therefore, we conducted an empirical analysis of $\|q_A - p_A\|_2$, the distance between the previous track q_A and the associated keypoint by NN. The results are shown in Figure 6. The naive NN-based pseudo-track algorithm works well on average when p_A is close to q_A , and tends to violate the condition when the distance is large. In the IMC dataset, the average distance $\|p_A - q_A\|_2$ is 4.47 px; P_q contains the GT track with high probability, only 6.4% below the upper bound.

3.4 Computational cost to make matcher trackable

The additional computational cost required to make a detector-free matcher trackable, i.e. the cost of *DeT* consists of: (i) extraction of steered crops of features $(F_A^{q_A}, F_B^{q_B^{\text{ps}}})$; (ii) evaluation $\mathcal{M}_F$; (iii) computation $\arg \max$ of the $P_q^{q_A, q_B^{\text{ps}}}$; (iv) sub-pixel regression for q_B . The increases in FLOPs from these additional computations are very small in relation to backbone feature extraction and coarse matching (attention/SSM). In practice, this increases the total FLOPs by only a few percent relative to the base matcher ($\approx 2\%$ in the case of *DeT-JamMA* Figure 7).

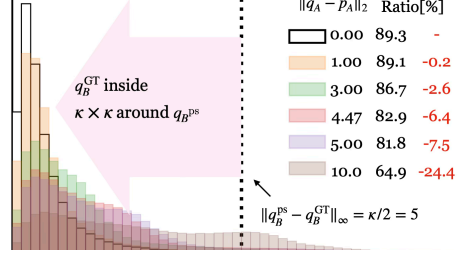


Fig. 6: Pseudo-track analysis. Histogram of $\|q_B^{\text{ps}} - q_B^{\text{gt}}\|_\infty$ for different $\|p_A - q_A\|_2$. The *Ratio* represents the percentage of q_B^{ps} to satisfy the condition. $\|q_A - p_A\|_2 = 0$ is the case where q_A coincides with p_A , which serves as an upper bound of the pseudo-track estimation for a given matcher. This histogram uses JamMA.

4 Experiments

We evaluate the performance of the *DeT* by integrating it with state-of-the-art, detector-free matchers, EDM [16] and JamMA [21] (Sec. 4.1). We assess performance in terms of low-level tracking capability and high-level application accuracy. First, we benchmark the tracking performance on real-world phototourism datasets (Sec. 4.3) and compare *DeT* with the state-of-the-art (SoTA) trackable methods. We then evaluate the accuracy in higher-level tasks: pose estimation (Sec. 4.4) and incremental SLAM (Sec. 4.5). We also conduct ablation studies of the *DeT* mechanism, comparing it with detector-free matchers by adopting the nearest-neighbor association, and analyze the negative effects of incorporating *DeT* in *two-view* evaluation (Sec. 4.6).

4.1 Implementation details.

We implement *DeT-EDM* and *DeT-JamMA* in PyTorch [22] by incorporating the proposed *DeT* mechanism into the official implementation of EDM [16] and JamMA [21]. For both variants, we do not modify their core mechanisms and make the detector-free matcher trackable solely by manipulating its fine-matching similarity matrix. Therefore, we use the official pretrained weights without any fine-tuning.

DeT-JamMA JamMA [21] is one of the state-of-the-art detector-free matchers. The feature extractor consists of two ConvNeXt V2 stages [32] followed by four Mamba blocks [9]. We set $\Delta_c=8$ px and $\Delta_f=2$ px for coarse and fine matching, respectively. The radius for associating p_A with q_A (Figure 3) is set to $5\sqrt{2}$ px.

DeT-EDM EDM [16] is another state-of-the-art detector-free matcher. Compared to JamMa [21], EDM [16] shows improved latency on certain benchmarks. The main difference with JamMA is its backbone, which uses *correlation injection*. Aside from differences in the backbone, it adopts a smaller fine-matching window, does not include a mixing function for fine features, and employs a different sub-pixel refinement algorithm. By accounting for the minor difference in their fine-matching mechanism, alg makes EDM trackable in a similar manner. We bilinearly sample F_q^A around q_A from F_f^A ,

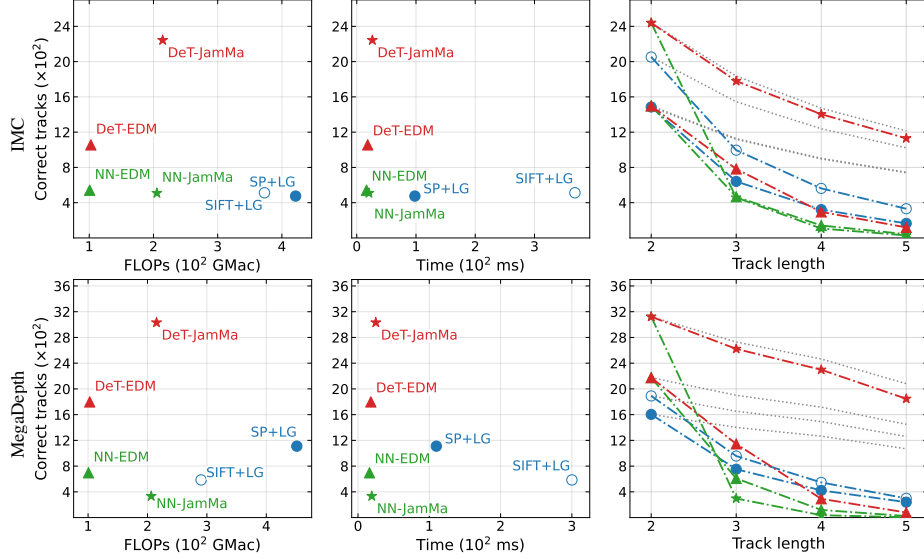


Fig. 7: Tracking benchmark. Left and middle: Reports the average number of correct (epipolar error $< 10^{-3}$) tracks from start (frame 1) to end (frame 5). Right: Report evolution of correct tracks ($\|q_B^{GT} - q_B\|^2 < 3$ px for each consecutive pair) for frame 1 to 5. *DeT-EDM* and *DeT-JamMA* keep significantly more correct tracks than prior methods. Without *DeT* (*NN-EDM*, *NN-JamMA*), the number decreases quickly due to fragmented tracks.

sample F_q^B around $\hat{q}_B$ from F_f^B , and feed them into the axis-based regression head to obtain q_B . The radius for associating p_A with q_A (Figure 3) is set to $5\sqrt{2}$ px.

NN-JamMA* and *NN-EDM In the tracking (Sec. 4.3), multi-view pose estimation (Sec. 4.4), and incremental SLAM (Sec. 4.5) experiments, we compare *DeT* variants with EDM and JamMA by incorporating nearest-neighbor (NN) association as an ablation of the *DeT* mechanism. As detailed in Sec. 3, detector-free matchers cannot obtain connected tracks due to the multi-view inconsistency issue. Therefore, we adopt an NN association scheme that connects the closest tracks if their distance falls within a predefined threshold, using their midpoints as the keypoint locations. Throughout the experiments, the connection threshold for *NN-EDM* and *NN-JamMA* is set to $5\sqrt{2}$ px.

4.2 Dataset.

Throughout the experiments, we use the 3D matching benchmark datasets MegaDepth [17] and IMC [33]. The MegaDepth dataset, a de facto standard for training and evaluating keypoint matching, is a phototourism dataset that captures the same landmarks at different times and poses. IMC [33] is another phototourism dataset primarily designed for evaluation. IMC is accompanied by sequence data (lists of image IDs), making it well-suited for evaluating tracking performance in the wild. Since the MegaDepth dataset lacks such sequence data, we construct frame sequences from the evaluation splits (MegaDepth1500) using the view overlap statistic. Specifically, starting

Table 1: Multi-view pose estimation. Evaluation on the *IMC* and *MegaDepth*. We report pose AUC@ 5° computed from essential matrix estimation using frames 1–5.

Method		<i>IMC</i>			<i>MegaDepth</i>	
		<i>Reichstag</i>	<i>Sacre C.</i>	<i>St-Peters</i>	<i>0015</i>	<i>0022</i>
<i>Detector-based</i>	SIFT [19]+LG [18]	69.0	81.8	60.7	46.3	48.2
	SP [3]+LG [18]	75.0	72.6	47.0	58.0	37.8
<i>Detector-free</i>	<i>NN-JamMA</i>	35.0	45.0	11.0	24.4	18.3
	<i>NN-EDM</i>	19.0	36.0	5.0	12.6	15.9
<i>Detect by Track</i>	<i>DeT-JamMA</i>	84.0	77.0	61.0	72.4	46.3
	<i>DeT-EDM</i>	29.0	52.0	12.0	17.3	19.5

from each image in the dataset, we select the next image as the one with the highest overlap in the range of 0.3–0.7 to create a sequence with view overlap.

4.3 Multi-view tracking

Metrics To benchmark low-level tracking capability, we evaluate the number of correct tracks along with computational complexity. Specifically, for a sequence of length $T=5$, we connect the tracks sequentially (Figure 4) and evaluate the number of correct tracks between the first and last images. We use the ground-truth (GT) essential matrix to compute the epipolar error and count tracks with an epipolar error smaller than 10^{-3} as correct tracks. We also evaluate sequential statistics. For each consecutive pair, we evaluate the number of correct tracks using reprojection error and count tracks with a reprojection error of less than 3 px as correct. We report the number of *GT* tracks for reference. Starting from the initial track of each algorithm, it counts the connected tracks using the GT pose and depth. Therefore, the GT count decreases only due to occlusion and non-overlapping views, serving as the upper bound for each algorithm.

Results Figure 7 reports the results. As expected, the number of correct tracks of *DeT-JamMA* is significantly higher than that of *NN-JamMA*, which uses fragmented tracks. *DeT* detects more than five times as many correct tracks as the default SuperPoint [3] + LightGlue [18] pipeline, maintaining a higher number of correct tracks for longer sequences in challenging tracking scenarios (Figure 7). We observe a similar trend for *DeT-EDM*. We note that for variants using a detector-free matcher, a large portion of tracks could be disconnected due to the *conflict* problem [20], where the number of matches decreases when the camera zooms out because the detector-free matcher extracts only one match within the Δ_C coarse patch. In other words, even when all previous tracks are connected, the count would be significantly reduced (see Sec. 5).

4.4 Multi-view Pose

Metrics We evaluate multi-view camera pose accuracy using the detected tracks from the sequences of five images. The essential matrix is estimated using LO-RANSAC [2] using the connected tracks between frame-1 and frame-5. The camera pose is recovered by decomposing the estimated essential matrix. The estimated pose is compared with the GT. We report AUC@5°. Across all sequences, we report mAcc@5° (the ratio of poses with rotational error $\leq 5^\circ$) and AUC@5° computed over the $[0^\circ, 5^\circ]$ range.

Table 2: Incremental SLAM. Evaluation on the *IMC* and *MegaDepth* benchmarks. Evaluation on the *IMC* and *MegaDepth* benchmarks. We report pose AUC@ 5° obtained from the incremental SLAM pipeline using frames 1–5.

Method		<i>IMC</i>			<i>MegaDepth</i>	
		<i>Reichstag</i>	<i>Sacre C.</i>	<i>St-Peters</i>	<i>0015</i>	<i>0022</i>
<i>Detector-based</i>	SIFT [19]+LG [18]	64.9	82.3	56.3	61.8	34.7
	SP [3]+LG [18]	46.0	70.9	41.1	43.4	24.6
<i>Detector-free</i>	<i>NN-JamMA</i>	59.8	83.0	61.9	58.0	41.1
	<i>NN-EDM</i>	55.1	79.7	57.4	51.9	34.3
<i>Detect by Track</i>	<i>DeT-JamMA</i>	68.7	88.0	69.6	65.5	44.9
	<i>DeT-EDM</i>	55.4	79.5	59.4	55.2	35.7

Results Tab. 1 compares pose accuracy. *DeT-JamMA* outperforms both the NN variant and the sparse baseline while requiring less than half of the FLOPs needed to compute the tracks for pose estimation. Note that although the *DeT-EDM* lags behind the sparse baseline, it requires only a small fraction of the computations compared with the baseline (less than 1/5) to obtain the correspondences (Figure 7).

4.5 Incremental SLAM

Metrics. We evaluate incremental SLAM without bundle adjustment (BA) using 5-frame sequences constructed from adjacent image pairs. We bootstrap the reconstruction from the first image pair by estimating the essential matrix using RANSAC [8], recovering the relative pose, and triangulating the initial 3D points. For subsequent frames, we estimate the camera pose by minimizing the reprojection error of registered 3D points and triangulating newly observed correspondences. This process is repeated incrementally until the final frame. For each 5-frame sequence, we obtain four relative poses with respect to the first image and compare them with the ground-truth extrinsics. We do not optimize the 3D point cloud and camera poses using BA; instead, we report the initial poses from the incremental 3D reconstruction. Following the LightGlue evaluation protocol, we report the camera pose AUC using the same threshold as in Sec. 4.4.

Results. Tab. 2 reports the pose accuracy of the incremental SLAM. As in the previous experiments, *DeT-JamMA* achieves higher SLAM pose accuracy. In addition to improved pose accuracy, *DeT-JamMA* registers more than four times as many 3D points, which benefits high-density 3D reconstruction.

4.6 Ablations

Detect-by-tracking vs. Detect-by-pair-matching. *NN-EDM* and *NN-JamMA* in Figure 7 and Tabs. 1 and 2 represent ablations obtained by removing the *DeT* mechanism for tracking. Although EDM [16] and JamMa [21] achieve excellent two-view accuracy, their disconnected tracks degrade tracking and multi-view pose estimation. Figure 8 visualizes tracks *detected by tracking* (*DeT*) versus those *detected by pair matching*; both are pairwise consistent. However, keypoints produced by detector-free matchers become fragmented (even with nearest-neighbor pairs) due to multi-view inconsistency, whereas the *DeT* variant produces connected tracks with sub-pixel accuracy. Figure 9

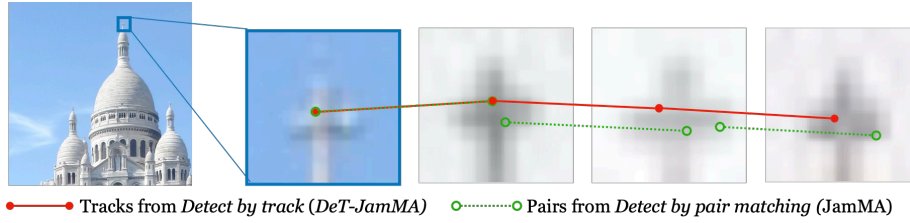


Fig. 8: *DeT*'s track analysis. *Detect by pair matching* (JamMA [21]) vs *detect by tracking* (*DeT-JamMA*). Both are pairwise consistent, but *Detect by pair matching* results in the fragmented track, while *DeT* yields multi-view-consistent tracks. Refer to ?? for more results.

Table 3: Ablation of *DeT* in two-view settings. Extended two-view relative pose estimation error evaluated on the MegaDepth Dataset [17]. The AUC of relative pose estimation error at three thresholds of models are presented. Trackable: Indicates whether the method is trackable (without multi-view inconsistency). Pair/Tracks reports the numbers of detected pairs or tracks.

Method	Trackable	Pair/Tracks $\uparrow$		Pose AUC $\uparrow$		
		Detected	Correct	@ 5°	@ 10°	@ 20°
<i>Detector-free</i>	EDM [16]	2395	93.0	64.5	77.7	86.8
	JamMa [21]	4021	88.9	64.1	77.4	86.5
<i>Detect-by-track</i>	<i>DeT-EDM</i> ✓	2393	93.0	63.6	77.0	86.3
	<i>DeT-JamMA</i> ✓	2427	90.9	61.4	75.1	84.9

visualizes tracks in challenging IMC [33] scenes. Without the tracking mechanism of *DeT*, the track quickly drifts over time; this behavior can also be observed quantitatively in Figure 7. *DeT* maintains more tracks under strong view and illumination changes, where sparse detectors tend to produce short or fragmented tracks.

Negative effect of *DeT*. In this ablation, we evaluate the potential negative effect of making the matcher trackable using *DeT*. To produce the connected tracks, *DeT* initializes matching from the previous tracks in image I_A , whereas a detector-free matcher can start from arbitrary locations in I_A . This restriction may negatively affect the quality and number of detected points because *DeT* must rely on an anchor point and can only connect tracks when detect-by-matching pairs exist near the previous tracks (within $5\sqrt{2}$ in our experiments) to form pseudo-tracks (Sec. 3.2). To evaluate this effect, we adopt a triplet protocol. For the *DeT* variant, we synthesize an initial image I'_A by applying a homography transformation to I_A , and generate an initial track by matching (I'_A, I_A) (Notice that for the first pair, the *DeT* variant works exactly the same as the detector-free matchers). We then connect tracks to I_B and evaluate the correspondences between (I_A, I_B). Under this setting, the results of the base detector-free matcher (EDM/JamMA) can be regarded as an upper bound for the *DeT* version. We report the pose AUC at 5°, 10°, and 20°, where poses are estimated using LO-RANSAC [2]. Similar to the other experiments, we follow the standard protocol of LightGlue [18] for the remaining settings. Table 3 summarizes the results. The number of tracks produced by *DeT-EDM* and *DeT-JamMA* is smaller than that of the original detector-free matchers. However, this behavior is expected because *DeT* relies on tracks from the previous image pair. Adopting more advanced algorithms for constructing pseudo-tracks or incorporating fine-tuning would reduce this gap (Sec. 5).

5 Limitations and Future works

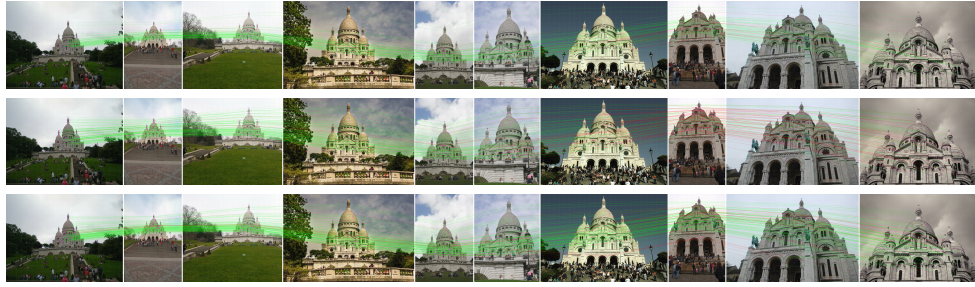
Translational motion assumption. As discussed in Sec. 3.3, our current NN-based algorithm assumes that p_A and q_A lie at a similar depth when computing the pseudo-track q_B^{ps} using Equation (2). Small violations of this assumption are acceptable as long as the distance between the pseudo-track and the true track is smaller than the size of the fine matching window Figure 6. A simple way to further improve this ratio is to increase κ , but this increases computational cost. Another possible improvement is to select p_A based on both spatial proximity and fine-feature similarity, thereby favoring points from the same object at similar depths and increasing the likelihood of satisfying the condition. Or one could adopt a small neural network to estimate the A more thorough analysis of this issue and the exploration of more advanced methods for estimating the pseudo-track q_B^{ps} — rather than relying on the naive translational motion assumption — remain directions for future work.

Adopting RCM. As discussed in Sec. 4.3, the number of connected tracks is limited by the *conflict* problem [20] in the zoom-in sequence, which is an inherent limitation of detector-free matchers. Incorporating the *Raises the Ceiling of Matching* (RCM) mechanism [20] into the base detector-free matcher (EDM and JAmMA) could significantly increase the number of tracks when combined with *DeT* in these zoom-in scenarios. We leave this evaluation for future work.

Finetuning with *DeT* loss. Our approach uses an off-the-shelf, pre-trained, detector-free model to produce connected tracks. Even without fine-tuning, our method achieves a high precision (Sec. 4). Fine-tuning with a tracking loss may further improve performance, but it requires modifications to the data loader to support multi-frame input. We leave this direction for future work.

***DeT*+Dense matchers.** Dense matchers such as RoMA [7] and EDM [16] also suffer from the multi-view inconsistency issue [34]. Dense matchers output sub-pixel pairwise warps for $I_A \rightarrow I_B$, but a subsequent $I_B \rightarrow I_C$ warp is not initialized from the previous sub-pixel track; hence, tracks remain disconnected at sub-pixel accuracy. They can be converted to produce connected tracks with sub-pixel accuracy by integrating *DeT*. In this work, we focus on combining *DeT* with *semi-dense* matchers, as they offer more scalable solutions for real-time applications. For applications where latency is less critical, such as an offline/high-accuracy/dense reconstruction scenario, exploring the *DeT* for dense matchers would be a good solution, optionally combined with post-processing refinement (Sec. 2). We left this exploration for future work.

Loop consistency and multi-view tracks. This study mainly addresses *online SLAM* and *incremental SfM*, where *chain tracks* suffice, such as in autonomous driving scenarios that discard older frames without loop closure. Consequently, this work focuses on the *chain track* settings. However, some applications benefit from a closed loop. For the sequence of $I_A \rightarrow I_B \rightarrow I_C \rightarrow I_A$, the closed loop in $I_C \rightarrow I_A$ can be established with a minor modification of the *DeT* procedure. In the $I_C \rightarrow I_A$ track, among those detected in the $I_A \rightarrow I_B$ track, we *select* the keypoint with the highest probability using P_q obtained from the $I_C \rightarrow I_A$ match rather than *identifying the peak*.



(a) *Sacre Coeur* sequence. SP+LG: 20/44, NN-JamMA: 21/88, DeT-JamMA: 94/127.



(b) *Reichstag* sequence. SP+LG: 20/29, NN-JamMA: 15/90, DeT-JamMA: 77/117.

Fig. 9: Qualitative tracks on IMC. We visualize detected tracks on the IMC [33] dataset. Green lines indicate correct tracks, while magenta lines indicate incorrect tracks. A track is regarded as correct if its epipolar error is less than 10^{-4} with respect to the initial image. SP [3] +LG [18] (top), NN-JamMA (middle), DeT-JamMA (bottom). Best viewed under magnification. We visualize only tracks that remain connected throughout the entire sequence (frames 1–10). Refer to ?? for more results, including the ones adopting newborn tracks (Sec. 3.2). For each sub-figure, we report the number of correct/detected tracks at the final (10th) frame.

Relation to Track Any Point. Recent point-tracking models operate on short windows of consecutive frames and maintain trajectories through occlusions: CoTracker [13, 14], TAPIR/TAP-Vid [4, 5], and PIPs/PointOdyssey [10, 38]. These methods assume sequential video access and do not directly expose the coarse-to-fine (C2F) local correlation maps used by detector-free *pairwise* matchers in SfM/SLAM. By contrast, DeT builds connected tracks solely from the local similarity maps, making it compatible with two-view pipelines and arbitrary view orderings. A unified framework is an interesting direction.

6 Conclusion

In this work, we propose an approach that *directly outputs sub-pixel-accurate connected tracks* from a detector-free matcher while preserving its advantages: dense cross-image interactions in image space and matching-oriented implicit keypoint selection. To this end, we advocate the shift from *detection by pair matching* to *detection by tracking* (DeT). We experimentally verified that the proposed approach significantly outperforms existing trackable methods by a large margin. We hope the concept of *detect by track* inspires further research toward more efficient and robust multi-view trackers based on detector-free matching.

References

1. Chen, H., Luo, Z., Zhou, L., Tian, Y., Zhen, M., Fang, T., Mckinnon, D., Tsin, Y., Quan, L.: Aspanformer: Detector-free image matching with adaptive span transformer. In: European conference on computer vision. pp. 20–36. Springer (2022)
2. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: DAGM 2003 (2003)
3. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
4. Doersch, C., Gupta, A., Lenc, K., Recasens, A., Smaila, L., Aytar, Y., Carreira, J., Zisserman, A., Yang, Y.: Tap-vid: A benchmark for tracking any point in a video. In: NeurIPS (2022)
5. Doersch, C., Yang, Y., Večerký, M., Gökay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: Tapir: Tracking any point with per-frame initialization and temporal refinement. In: ICCV (2023)
6. Edstedt, J., Athanasiadis, I., Wadenbäck, M., Felsberg, M.: DKM: Dense kernelized feature matching for geometry estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (2023)
7. Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., Felsberg, M.: RoMa: Robust Dense Feature Matching. In: IEEE Conference on Computer Vision and Pattern Recognition (2024)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6) (Jun 1981). <https://doi.org/10.1145/358669.358692>
9. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. In: First Conference on Language Modeling (2024), <https://openreview.net/forum?id=tEYskw1VY2>
10. Harley, A.W., Fang, Z., Fragkiadaki, K.: Particle video revisited: Tracking through occlusions using point trajectories. In: ECCV (2022)
11. He, X., Sun, J., Wang, Y., Peng, S., Huang, Q., Bao, H., Zhou, X.: Detector-free structure-from-motion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
12. He, X., Sun, J., Wang, Y., Huang, D., Bao, H., Zhou, X.: Onepose++: Keypoint-free one-shot object pose estimation without cad models. *Advances in Neural Information Processing Systems* **35**, 35103–35115 (2022)
13. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. arXiv:2307.07635 (2023)
14. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In: CVPR (2025)
15. Lee, J., Yoo, S.: Dense-sfm: Structure from motion with dense consistent matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6404–6414 (June 2025)
16. Li, X., Rao, T., Pan, C.: Edm: Efficient deep feature matching. In: ICCV (2025)
17. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
18. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: Lightglue: Local feature matching at light speed. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 17627–17638 (2023)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision* **60**(2), 91–110 (2004)

20. Lu, X., Du, S.: Raising the ceiling: Conflict-free local feature matching with dynamic view switching. In: ECCV (2024)
21. Lu, X., Du, S.: Jamma: Ultra-lightweight local feature matching with joint mamba. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14934–14943 (2025)
22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library (2019), <https://arxiv.org/abs/1912.01703>
23. Potje, G., Cadar, F., Araujo, A., Martins, R., Nascimento, E.R.: Xfeat: Accelerated features for lightweight image matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2682–2691 (2024)
24. Revaud, J., De Souza, C., Humenberger, M., Weinzaepfel, P.: R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems* **32** (2019)
25. Roessle, B., Nießner, M.: End2end multi-view feature matching with differentiable pose optimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 477–487 (2023)
26. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
27. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8922–8931 (2021)
28. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., Dosovitskiy, A.: Mlp-mixer: an all-mlp architecture for vision. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. NIPS '21, Curran Associates Inc., Red Hook, NY, USA (2021)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
30. Wang, Q., Zhang, J., Yang, K., Peng, K., Stiefel, R.: Matchformer: Interleaving attention in transformers for feature matching. In: Proceedings of the Asian conference on computer vision. pp. 2746–2762 (2022)
31. Wang, Y., He, X., Peng, S., Tan, D., Zhou, X.: Efficient loftr: Semi-dense local feature matching with sparse-like speed. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21666–21675 (2024)
32. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: Convnext v2: Co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16133–16142 (June 2023)
33. Yi, K.M., Trulls, E., Mishkin, D., Balntas, V., Morelli, L., Bellavia, F., Matas, J., Schönberger, J., Lepetit, V.: Image matching challenge (2021 edition) workshop on local features & beyond. In: Seventh Workshop on Image Matching Local Features & Beyond, in conjunction with CVPR 2021 (2021), <https://image-matching-workshop.github.io/>
34. Yu, H., Xingyi, H., gogogo, Xiaowei, Z., Tan, D.: Detector-free sfm and transparent scene trick. In: kaggle competitions (2024), <https://www.kaggle.com/competitions/image-matching-challenge-2024>

35. Zhang, J., Xia, Z., Dong, M., Shen, S., Yue, L., Zheng, X.: Comatcher: Multi-view collaborative feature matching. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21970–21980 (2025)
36. Zhao, X., Wu, X., Chen, W., Chen, P.C., Xu, Q., Li, Z.: Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement* **72**, 1–16 (2023)
37. Zhao, X., Wu, X., Miao, J., Chen, W., Chen, P.C., Li, Z.: Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE Transactions on Multimedia* **25**, 3101–3112 (2022)
38. Zheng, Y., Li, G., Harley, A.W., Shen, B., Wetzstein, G., Guibas, L.J.: Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In: ICCV (2023)