Prior Work

Tomáš Jelínek

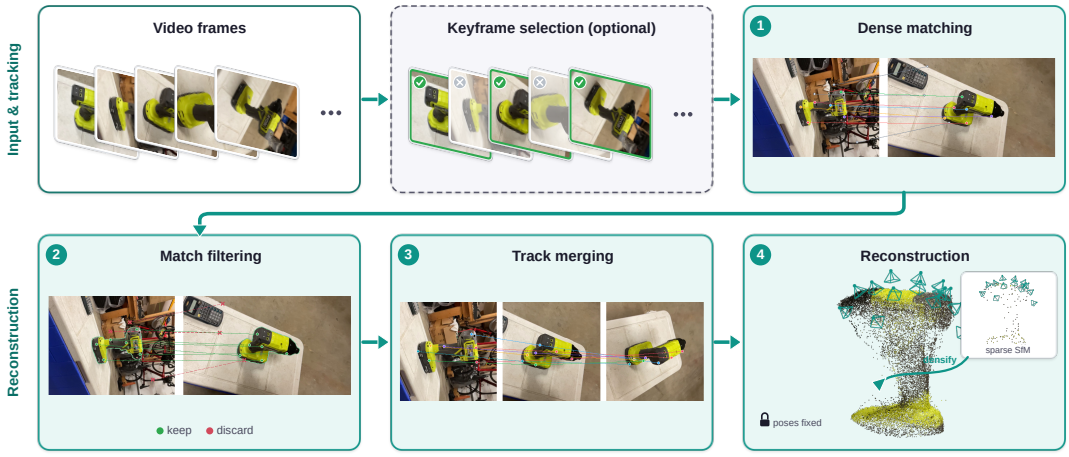

Figure 1: **Model-free object reconstruction pipeline.** The four numbered stages ①–④ are described in section 1.1. Figure from a manuscript in preparation.

1. Prior Work

1.1. 3D Object Reconstruction from Videos

Manuscript in preparation, available upon request. Technical description: tjelinek.github.io/reconstruction.
Given a video stream $(\mathcal{I}_1, \cdots, \mathcal{I}_n)$ capturing a rigid object, a first-frame segmentation mask $\mathcal{S}_1$, and known camera intrinsics $(\mathcal{K}_1, \cdots, \mathcal{K}_n)$, the goal is to recover the object's 3D point cloud $\mathcal{M}$ and the camera poses in the object's own frame.

When the object is rigid with respect to a textured background this is easy: reconstruct the whole scene, by structure-from-motion or a feed-forward network such as VGGT [1], and keep the points inside the mask. We care about the harder regime of a hand-held object turned over in front of the camera: there, the feed-forward reconstructors collapse, and masking the background with a uniform color results in broken reconstruction, presumably because such a regime is out-of-distribution for these methods. While point trackers allow for long tracks, which is beneficial for SfM, we found that they too fail in dynamic scenes.

We design our solution as an SfM pipeline and envision a pipeline that uses the same matching mechanism for correspondences used for object reconstruction as well as for correspondences used to solve a PnP problem when estimating the pose of the object in a novel scene.

Built on pairwise correspondences rather than a feed-forward pass, our pipeline (fig. 1) handles this moving-object regime and works whether the input is left intact or masked. It runs in four stages:

① **Dense matching.** SAM2 [2] propagates $\mathcal{S}_1$ to every frame; the dense matcher UFM [3] then produces dense matches with per-match certainties on the *unmasked* frames. We add a segmentation input channel and fine-tune it on objects held out from the test set, so the matcher is forced to produce correct correspondences between the source and target masks, treating that region as a single rigid body while the rest of the scene may move. Keyframes are picked online by the share of confident matches.

② **Match filtering.** A match is kept only when both endpoints fall inside the propagated masks, so the reconstructed points belong to the object.

③ **Track merging.** The surviving matches are chained across frames into multi-view feature tracks. This step prevents duplicate reconstruction and substantially increases robustness.

④ **Reconstruction.** Incremental SfM (COLMAP [4]) with intrinsics held fixed recovers camera poses and a sparse cloud. To obtain a dense point cloud $\mathcal{M}$, we densify the model with frame-wise

correspondences.

While the problem looks simple and its solution is, one could argue, a relatively simple SfM pipeline, it is in fact hard: according to the BOP benchmark, model-free object pose estimation (where you get a video capturing the object with some reference camera poses) has not been sufficiently solved [5], unlike the model-based problem, where you have a CAD model at your disposal. The main challenges for the reconstruction are a) low-quality matches, b) correlated matching outliers (see section 1.3), and c) heavy occlusions, where even synthetically created occlusions after matching heavily downgrade the reconstruction performance.

This is a project I was working on in spring and summer 2025. I am now working on finishing it and want to publish it as soon as possible. If I were to do this project again, I would move closer to end-to-end solutions such as NeMO [6], with the encoder producing a reconstruction (alongside a latent representation) and the decoder detecting the object, computing a matching, and directly regressing the object's pose consistent with the matching.

1.2. Dense Matchers for Dense Tracking

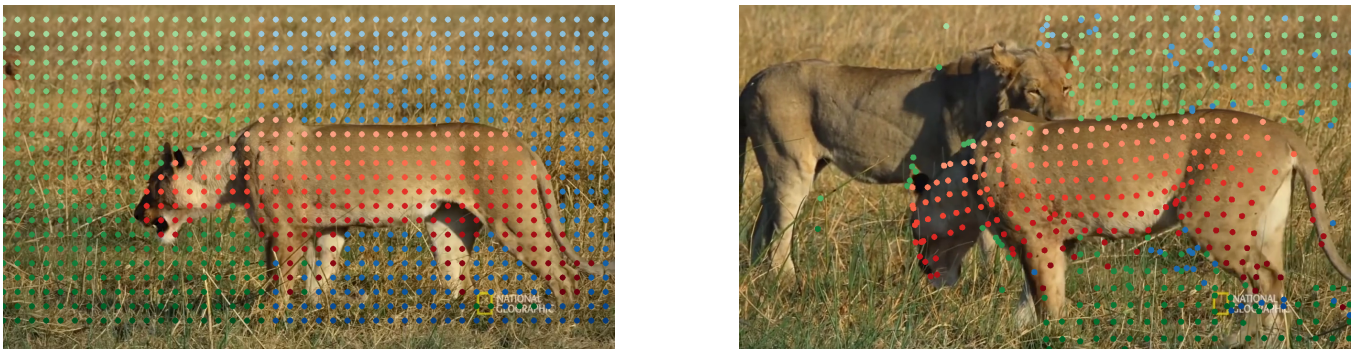

Figure 2: **Long-term point tracking with a dense-matcher ensemble.** Left: reference frame #0 of a TAP-Vid DAVIS sequence. Right: the same points tracked to frame #140 by our selective RoMa position-prediction ensemble (RAFT+RoMa). Green: visible in both frames (correct); blue: occluded in frame #140, so blue on the right is a false match; red: the moving lioness. Panels (a) and (h) of Fig. 2 in the paper.

Technical description: tjelinek.github.io/dense-tracking. Dense Matchers for Dense Tracking [7] builds on MFT [8], which tracks a point by a point-wise combination of optical flow computed over logarithmically spaced distances relative to the current frame. MFT's flow backbone is RAFT [9], augmented with uncertainty and occlusion heads.

However, RAFT is trained for small-displacement flow between nearby frames, whereas the long log-spaced baselines that give MFT its drift resistance induce exactly the large-displacement, large-appearance-change regime for which wide-baseline dense matchers are trained. We therefore combine RAFT flow with a dense matcher (RoMa [10]) inside MFT's chaining scheme, where we use RoMa matches that are also classified by RoMa as confident, and RAFT matches for the rest.

Concretely, writing $\mathcal{F}^{(i,j)}$ for the flow from frame i to j and $\mathbf{p}_i = \mathbf{p}_1 + \mathcal{F}^{(1,i)}_{\mathrm{MFT}}(\mathbf{p}_1)$ for the tracked position of a reference point $\mathbf{p}_1$, MFT builds the long-term flow by chaining through an intermediate frame i_M,

$$\mathcal{F}^{(1,j)}_{\mathrm{MFT}}(\mathbf{p}_1) = \mathcal{F}^{(1,i_M)}_{\mathrm{MFT}}(\mathbf{p}_1) + \mathcal{F}^{(i_M,j)}(\mathbf{p}_{i_M}), \tag{1}$$

where $i_M = j - \Delta_k$ (log-spaced $\Delta_k = 2^{k-1}$) is the intermediate frame whose chain accumulates the lowest variance σ among chains with no occluded point; the variance and the occlusion score o accumulate along a chain as $\sigma^{(1,i_M,j)} = \sigma^{(1,i_M)}_{\mathrm{MFT}} + \sigma^{(i_M,j)}$ and $o^{(1,i_M,j)} = \max\left(o^{(1,i_M)}_{\mathrm{MFT}}, o^{(i_M,j)}\right)$. For a matcher to work in this scheme, it has to compute variance and an occlusion score. RoMa provides

neither, only a per-match certainty $\rho \in [0, 1]$, so we derive both,

$$o = 1 - \rho, \qquad \sigma = \begin{cases} 0 & \text{if } \rho \geq \theta_\rho, \\ 1000 & \text{otherwise,} \end{cases} \tag{2}$$

treating a match as confident with zero-variance when its certainty is above θ_ρ. In the latter case, we assign it a value well above highest observed variance. A point is deemed occluded when o exceeds an occlusion threshold, set to $\theta_o^{\text{RoMa}} = 0.95$ for RoMa against $\theta_o^{\text{RAFT}} = 0.02$ for RAFT.

On TAP-Vid DAVIS [11] (30 sequences at 512×512 resolution), fusing RoMa into MFT raises the Average Jaccard table 1 (AJ, rewarding both position and occlusion accuracy) from 47.4 to 51.6 over the MFT-RAFT baseline. The gain is concentrated in position accuracy: at $\delta_{\text{avg}} = 73.4$ this strictly causal ensemble rivals the non-causal, attention-refined sparse trackers TAPIR [12] (70.0) and approaches CoTracker [13] (75.9), while trading away occlusion accuracy.

	Position / Occlusion	AJ	δ_{avg}	OA
(1)	RAFT / RAFT	47.4	67.1	77.7
(2)	RoMa / RoMa	48.8	72.7	71.7
(3)	RoMa / RAFT	50.2	72.7	77.7
(4)	RAFT+RoMa / RAFT	**51.6**	**73.4**	**77.7**
	TAPIR [12]	56.2	70.0	86.5
	CoTracker [13]	61.0	75.9	89.4

Table 1: **Two-tracker combinations on TAP-Vid DAVIS** (512 px, *first* mode). **AJ**: Average Jaccard, jointly scoring position and visibility; δ_{avg}: average position accuracy, the fraction of visible points predicted within a set of pixel thresholds; **OA**: occlusion accuracy of the visible/occluded call. Rows (1)-(4) show MFT's flow and occlusion prediction source. Unlike us, TAPIR and CoTracker are non-causal sparse trackers. Table source: [7].

The choice of matcher matters: substituting a different dense matcher, DKM [14], instead degraded tracking sharply, so the gain comes from the matcher's wide-baseline robustness, not from ensembling a second model per se.

The paper was completed end-to-end in 2.5 weeks (with an existing MFT infrastructure).

1.3. Differentiable Object Reconstruction and Tracking

Technical description: tjelinek.github.io/differentiable-reconstruction. Building on the model-free tracker of Rozumnyi et al. [15], we are given a video stream $\{I_1, \ldots, I_n\}$ of a rigid object and an initial segmentation mask S_1, and seek globally consistent 6DoF poses $\mathbf{p}_1, \ldots, \mathbf{p}_n$ together with a 3D model represented as a mesh M. Under the rigidity assumption M is constant across frames; it is recovered by deforming per-vertex offsets from a prototype (a sphere whose radius matches the projected object in I_1) through a differentiable renderer.

The poses and mesh are fit over a sparse keyframe set $K_n \subseteq \{1, \ldots, n-1\}$ using the objective of [15]. Let $\hat{I}_i$ and $\hat{S}_i$ be the appearance and silhouette rendered from the model M at pose $\mathbf{p}_i = (T_i, Q_i)$ (translation and unit quaternion), $F(I_i)$ the deep features of frame I_i, S_i its input segmentation, and $\mu_n = (|K_n| + 1)^{-1}$. The appearance term uses a Cauchy robust norm $\|\cdot\|_\gamma$, the silhouette term an IoU and a distance-transform (DT) penalty:

$$L_F = \mu_n \sum_{i \in K_n \cup \{n\}} \left\| \hat{S}_i \cdot (\hat{I}_i - F(I_i)) \right\|_\gamma, \tag{3}$$

$$L_S = \mu_n \sum_{i \in K_n \cup \{n\}} (1 - \text{IoU}(S_i, \hat{S}_i)) + \left\| \text{DT}(S_i) \cdot \hat{S}_i \right\|. \tag{4}$$

Motion is regularized toward the most recent keyframe $k = K_n[-1]$,

$$L_M = \max\Big(0, \tfrac{\|T_k - T_n\|_2}{n-k} - \nu_T\Big) + \max\Big(0, \tfrac{\angle(Q_k, Q_n)}{n-k} - \nu_Q\Big), \tag{5}$$

and a texture total-variation term L_T smooths the feature map. The shape is regularized by a Laplacian term L_L [16] that pulls every vertex toward the centroid of its single-edge neighbourhood $N(v)$,

$$L_L = \sum_{v \in M} \left\| v - \tfrac{1}{|N(v)|} \sum_{u \in N(v)} u \right\|_2^2. \tag{6}$$

These terms combine into a total objective $L = \alpha_F L_F + \alpha_S L_S + \alpha_M L_M + \alpha_L L_L + \alpha_T L_T$. A frame is promoted to a keyframe only when its view differs sufficiently from the last: relative translation above half the object size, or relative rotation above 45°.

The Laplacian term regularizes vertex spacing: without it, the optimization develops shapes with large spikes; with it, the regularizer's optimum is a flat surface with uniformly spaced vertices.

For many objects, especially those without distinctive texture, the mesh collapses into a flat disk upon observing the first few frames. The easy fixes like hyper-parameter tuning or adding random frames as keyframes did not help: the degenerate solution blends the texture onto the flat disk to minimize the appearance loss.

To solve it, our idea was to pre-warp the object pose using 2D –2D correspondences from optical flow, minimizing the rendered flow error

$$E(\Delta \mathbf{p}_{ij}) = \sum_{x} \left\| \pi(\Delta \mathbf{p}_{ij}\, X(x)) - x - W^{i \to j}(x) \right\|_2, \tag{7}$$

between the flow induced by the relative pose $\Delta \mathbf{p}_{ij}$ of frames i and j and the observed flow $W^{i \to j}$, over $\Delta \mathbf{p}_{ij} \in SE(3)$ with a Levenberg–Marquardt solver for fast convergence, where $X(x)$ is the mesh surface point at x in frame i.

Interestingly, assume that the object rotates around the vertical axis (relative to the image plane) to the right. When observing frame i, the pose estimates the true translation in the direction of the rotation with error $\epsilon_i > 0$. Then, when observing frame $i+1$, the translation estimate that minimizes the flow error has an error w.r.t. the ground-truth rotation $\epsilon_{i+1} > \epsilon_i$.

The decisive obstacle was occlusion: neither optical flow (RAFT [9], MFT [8]) nor dense matchers (RoMa [10]) reliably predict occlusion. Also, the occluded matches were spatially correlated rather than independent, which violates the core assumption of RANSAC, so even RANSAC-filtered correspondences stayed contaminated. At the same time, circular or transitive correspondence checks slowed down an already slow pipeline. We hence discontinued the project.

1.4. Open-Set 2D Detection: Template Condensation

Technical description: tjelinek.github.io/template-condensation. In pose estimation, object detection is an indispensable part of the pipeline: the goal is to (a) detect the objects of interest and (b) estimate their object-to-camera transformation.

The community standard is CNOS [17], which assumes being provided with CAD models of the objects of interest. Subsequently, they render the object from 42 views uniformly spaced around the object forming a template set T_i for object i. Then, they compute a descriptor set $D_i = \{d(t) \mid t \in T_i\}$ where $d(\cdot)$ computes DINOv2 [18] class token.

At inference time, they use an object-agnostic class detector such as SAM [19] to compute all objects proposals in the input image. Let p_j be the j-th proposal, with descriptor $d_j = d(p_j)$, and let $c_j(t)$ denote the cosine similarity between $d(t)$ and d_j. Let $c_{ij}^{(1)} \geq c_{ij}^{(2)} \geq \cdots \geq c_{ij}^{(|T_i|)}$ be the similarities $\{\, c_j(t) \mid t \in T_i \,\}$ sorted in non-increasing order. The score of object i averages its k largest similarities to the proposal,

$$s_{ij} = \frac{1}{k} \sum_{l=1}^{k} c_{ij}^{(l)}, \tag{8}$$

where CNOS use $k = 5$. The proposal is then assigned the best-scoring object across all objects,

$$o_j = \arg\max_i \; s_{ij}, \qquad s_j = \max_i \; s_{ij}. \tag{9}$$

A proposal is accepted as a detection of object o_j when $s_j \geq \theta$ for a global threshold θ.

Template bank	HANDAL	HOPE	LM-O	T-LESS	Mean
Uncondensed	0.41	0.57	0.34	0.50	0.46
Condensed (ours)	0.37	0.51	0.34	0.47	0.42

Table 2: **Condensation preserves detection quality.** BOP detection mAP with the full onboarding template bank versus the condensed bank, on four BOP datasets.

Matching cost grows linearly with the template bank: every proposal is compared against all $\sum_i |T_i|$ templates. Built from an onboarding video rather than a few CAD renders, the bank holds hundreds of highly redundant views per object. Moreover, the k most similar templates will arguably be from very similar viewpoints, decreasing the robustness of eq. (9).

We reduce the template bank using its DINO class descriptors with Hart's symmetric nearest neighbor condensation algorithm [20]. The algorithm iteratively processes objects partitioned into classes. At the beginning, it sets a small condensed set S_c for objects of class c. Then it matches each template $t_c \notin S_c$ against a store combining S_c with *all* templates of the other objects. It then adds t into S_c only when its nearest neighbor $t^\star$ belongs to a different object ($y(t^\star) \neq y(t)$) or is insufficiently similar ($\langle d(t) \,|\, d(t^\star) \rangle < \tau$). This process is repeated until S_c does not change. An example of a condensed set is displayed in fig. 3.

Matching jointly against every other object enforces inter-object separability, while the coverage threshold τ retains enough exemplars to span each object's appearance manifold, not only its decision boundary. The motivation behind introducing τ is to make the method more robust at inference time, where we not only seek to distinguish the objects from one another but also from detections that were not observed at condensation time.

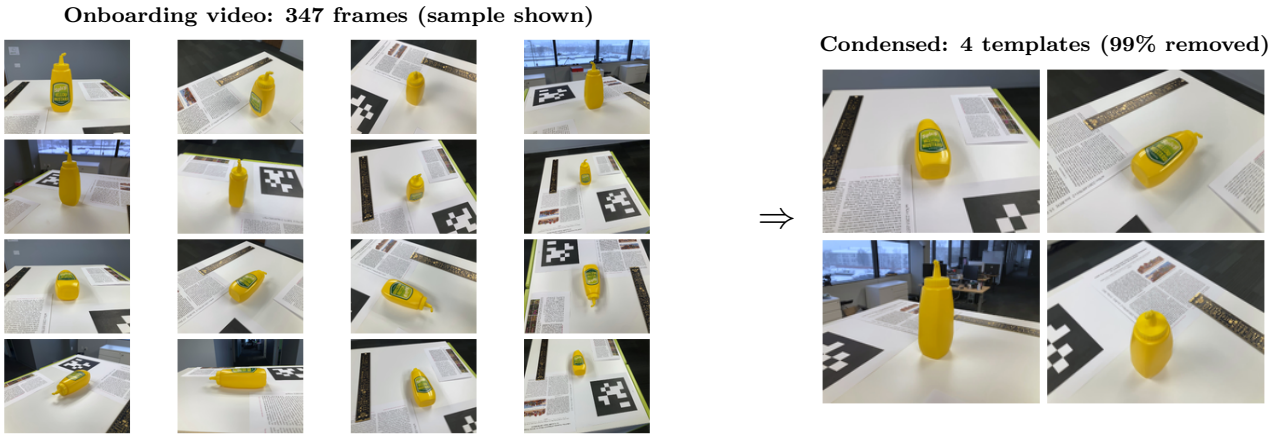

Figure 3: **Condensation keeps diverse representatives, not near-duplicates.** From a 347-frame static HOPE onboarding sequence, the symmetric condensed nearest-neighbour rule retains just 4 templates (right) that still span the object's distinct appearances while discarding redundant views.

Condensing the onboarding bank this way removes roughly 90% of the templates, a median of only a few to a few dozen surviving per object (fig. 3), while retaining about 92% of the full bank's BOP detection mAP (mean average precision; table 2). The condensed bank needs no CAD model, yet stays competitive with CNOS with the FastSAM configuration that uses a CAD model for template rendering.

Limitations. The method leans on a single global threshold θ, assuming every object sits the same distance from its false positives: a pink elephant on snow and a needle in a haystack are given the same θ, though their optimal margins differ. It also assumes access to the test-time distribution of negatives for fine-tuning.

What I would try next. True positives are classified into their classes well; specificity is the hard part. Per-object θ_c heuristics (Lowe's ratio, inter-class similarity quantiles, outlier detection) did

not help, which suggests the class-token comparison itself, fast and data-efficient as it is, is simply too weak. Two directions look more promising: metric learning for separability, at the cost of needing the large test-time detection set; and geometric verification to reject the obvious false positives, where today an indistinct background pattern can outscore a genuine detection. Robust TP/FP classification without access to the test-time negatives is genuinely hard, yet central to open-set detection.

References

[1] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "VGGT: Visual geometry grounded transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[2] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "SAM 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[3] Y. Zhang, N. Keetha, C. Lyu, B. Jhamb, Y. Chen, Y. Qiu, J. Karhade, S. Jha, Y. Hu, D. Ramanan, S. Scherer, and W. Wang, "UFM: A simple path towards unified dense correspondence with flow," *arXiv preprint arXiv:2506.09278*, 2025.

[4] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] V. N. Nguyen, S. Tyree, A. Guo, M. Fourmy, A. Gouda, T. Lee, S. Moon, H. Son, L. Ranftl, J. Tremblay, *et al.*, "BOP challenge 2024 on model-based and model-free 6D object pose estimation," *arXiv preprint arXiv:2504.02812*, 2025.

[6] S. Jung, L. Klüpfel, R. Triebel, and M. Durner, "Finding NeMO: A geometry-aware representation of template views for few-shot perception," in *International Conference on 3D Vision (3DV)*, 2026.

[7] T. Jelínek, J. Šerỳch, and J. Matas, "Dense matchers for dense tracking," in *Proceedings of the 27th Computer Vision Winter Workshop (CVWW)*, 2024.

[8] M. Neoral, J. Šerých, and J. Matas, "MFT: Long-term tracking of every pixel," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6823–6833, 2024.

[9] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 402–419, 2020.

[10] J. Edstedt, Q. Sun, G. Bökman, M. Wadenbäck, and M. Felsberg, "RoMa: Robust dense feature matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19790–19800, 2024.

[11] C. Doersch, A. Gupta, L. Markeeva, A. Recasens, L. Smaira, Y. Aytar, J. Carreira, A. Zisserman, and Y. Yang, "TAP-Vid: A benchmark for tracking any point in a video," in *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2022.

[12] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman, "TAPIR: Tracking any point with per-frame initialization and temporal refinement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[13] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht, "CoTracker: It is better to track together," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[14] J. Edstedt, I. Athanasiadis, M. Wadenbäck, and M. Felsberg, "DKM: Dense kernelized feature matching for geometry estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[15] D. Rozumnyi, J. Matas, M. Pollefeys, V. Ferrari, and M. R. Oswald, "Tracking by 3D model estimation of unknown objects in videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14040–14050, 2023.

[16] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 317–324, 1999.

[17] V. N. Nguyen, T. Groueix, G. Ponimatkin, V. Lepetit, and T. Hodan, "CNOS: A strong baseline for CAD-based novel object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2126–2132, 2023.

[18] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research (TMLR)*, 2024.

[19] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003, 2023.

[20] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.