MFTIQ: Multi-Flow Tracker with Independent Matching Quality Estimation

Jonas Serych, Michal Neoral, Jiri Matas
CMP Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
{serycjon,neoramic,matas}@fel.cvut.cz

Abstract

In this work, we present MFTIQ, a novel dense long-term tracking model that advances the Multi-Flow Tracker (MFT) framework to address challenges in point-level visual tracking in video sequences. MFTIQ builds upon the flow-chaining concepts of MFT, integrating an Independent Quality (IQ) module that separates correspondence quality estimation from optical flow computations. This decoupling significantly enhances the accuracy and flexibility of the tracking process, allowing MFTIQ to maintain reliable trajectory predictions even in scenarios of prolonged occlusions and complex dynamics. Designed to be "plug-and-play", MFTIQ can be employed with any off-the-shelf optical flow method without the need for fine-tuning or architectural modifications. Experimental validations on the TAP-Vid Davis dataset show that MFTIQ with RoMa [16] optical flow not only surpasses MFT but also performs comparably to state-of-the-art trackers while having substantially faster processing speed. Code and models available at https://github.com/serycjon/MFTIQ.

1. Introduction

Point-level visual tracking is a hot research topic [10, 12, 27, 31]. Instead of the classical task of tracking objects by bounding boxes [2, 8, 26] or segmentation masks [28, 43], the goal is to track arbitrary points lying on surfaces in the scene. The resulting point correspondences are useful for various downstream applications, like SLAM [18, 38] or motion prediction [56]. While most current methods [10–12, 27, 57] focus on *sparse* point-tracking, applications like 3D reconstruction, video editing, or augmented reality, benefit from *dense* correspondences, *i.e*., correspondences estimated for every pixel of the initial video frame.

Traditionally, long-range dense tracking may be achieved by sequential chaining of optical flow (OF). However this approach has major drawbacks. The estimated trajectory drift over time due to error accumulation, and tracking stops to be reliable in presence of occlusion since the sequential chaining has no mechanism to recover. Recently, Multi-Flow Tracker (MFT) [39] revisited flow chaining [6, 7] for not only consecutive frames, but also for temporally distant frame pairs. MFT produces long, dense trajectories by selecting the most reliable chain of optical flows for each tracked point. The flow chain reliability is determined by accumulating uncertainties and occlusion state computed for each optical flow in the flow chain. However, this uncertainty accumulation can lead to error accumulation and drift. Moreover, MFT is tightly coupled with the RAFT optical flow, but it has been shown [25] that other OF methods

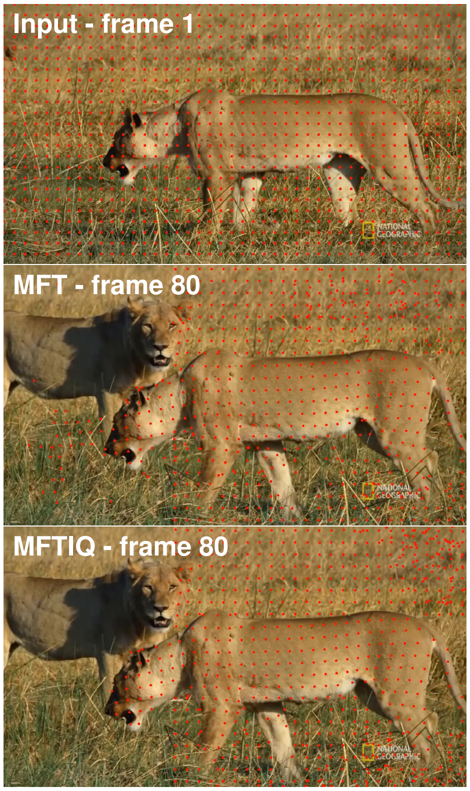

Figure 1. **Dense long-term tracking – MFT and MFTIQ comparison**. Visualisation of query positions *(red)* tracked from frame 1 to frame 80. MFTIQ generates a lower number of false re-detections than MFT, especially on the grass in bottom-left, which was out-of-view on frame 1. Best viewed zoomed-in and in color.

perform better.

We propose MFTIQ, a dense, long-term tracker. Like other dense point-trackers [5,25,29,39,53] it is based on optical flow computation. We design the method to be trained once and then work with an arbitrary optical flow method in a "plug-and-play" fashion and without any fine-tuning or architecture changes. This allows the user to choose a suitable speed/performance trade-off by using the appropriate flow. The MFTIQ generalizes to multiple optical flow methods not seen during training, as we have experimentally evaluated. We expect future faster and/or higher quality flows to improve the proposed tracker performance for free, *i.e.* without any re-training needed.

The proposed method replaces the flow-chain-selection method proposed in MFT, with an improved Independent Quality and occlusion estimation module (IQ). Unlike the MFT method, which estimates occlusion and correspondence quality (uncertainty) jointly with optical flow estimation, MFTIQ decouples these estimations from the optical flow computation. This separation also allows the occlusion and correspondence quality to be estimated directly for the optical flow chain between the template and the current frame, without need for error-prone uncertainty accumulation. Figure 1 shows an example where the MFTIQ strategy produces significantly less false re-detections than MFT.

MFTIQ achieves results comparable to state-of-the-art trackers when using RoMa [16] as the optical flow estimator and consistently outperforms MFT across most tested optical flow methods. It is important to note that MFTIQ was not trained with RoMa, highlighting the "plug-and-play" functionality of the proposed method. Moreover, for dense tracking, MFTIQ is significantly faster than state-of-the-art trackers, even with the slowest optical flow methods tested. MFTIQ is also causal, *i.e.*, it only uses the current frame and the previous ones, which is not the case for most point-trackers.

In this work, we introduce MFTIQ, a novel dense, long-term tracking method improving on the MFT [39] flow-chaining idea. **Our contributions are as follows**: 1) We have developed the Independent Quality (IQ) module, which decouples occlusion and correspondence quality estimation from optical flow computation. This separation enhances tracking accuracy and flexibility. 2) MFTIQ features "plug-and-play" functionality, allowing integration with any off-the-shelf optical flow method without re-training or fine-tuning. This flexibility enables users to tailor tracker performance to specific needs. 3) We conducted experimental evaluations using multiple optical flow methods, demonstrating that MFTIQ matches the performance of state-of-the-art trackers while being significantly faster in dense tracking scenarios.

2. Related Work

Optical flow is a fundamental problem [22] in computer vision in which the pixel-level displacement between pair of frames is to be densely estimated. Most of the current methods are based on learning [13, 14, 16, 24, 40, 47, 48, 50, 63]. FlowNet [14] introduced correlation cost-volume (CCV) to learning based optical flow estimation to provide similarity measurement between neighboring features from consecutive frames. Later, RAFT [51] employs 4D CCV for all pairs of pixel on lower resolution and iteratively estimates optical flow. FlowFormer [23, 48] updates RAFT with transformer blocks. RoMa [16] is a dense wide-baseline stereo matcher that can be, however, used as optical flow estimator. Both, FlowFormer and RoMa bring higher accuracy for the cost of slower processing time and bigger memory requirements. This is addressed with NeuFlow [63] or SEA-RAFT [54], which focus more on efficient, higher speed computation.

While these methods are state-of-the-art in field of optical flow estimation, their possibility for employment in *long-term* tracking are rather limited. There are multi-frame optical flow approaches, such as VideoFlow [47] or MemFlow [13], but they are still focused on estimation of optical flow between adjacent frames rather than long-term optical flow and mentioned limitation of optical flow chaining remains.

Sparse long-term point-tracking focus on tracking small number of query points on the object surface throughout the video. Particle-video [45] tracks only visible query points and fails to track continuously through occlusions, instead starting new tracks. Revisiting this, PIPs [20] takes frames from fixed-size temporal window (8 frames) and estimate sparse point tracking with iterative updates. They propose a strategy for linking the eight-frame tracks over longer period of time, however their method cannot recover from longer occlusions. TAP-Net [10] computes CCV for each query point with each frame of a video and from it estimate occlusion and position by their two branch network. TAPIR [12] combines per-frame global-matching prediction of TAP-Net with refining process inspired by PIPs. Current state-of-the-art – BootsTAP [11] is the TAPIR tracker fine-tuned in a self-supervised fashion on large amount of in-the-wild videos. The long training using 256 A100 GPUs on the 15M YouTube video clips is however too costly to reproduce for most researchers.

CoTracker [27] processes query points with a sliding-window transformer that enables multiple tracks to influence each other. However, the best performance is achieved by tracking single query point at time, supplemented with auxiliary grid of queries. SpacialTracker [57] builds upon CoTracker. Instead of tracking points in 2D, it lifts them to 3D using an off-the-shelf monocular depth estimation method. All the mentioned methods can track densely by tracking the points one-by-one or in batches, but the resulting speed is low.

Dense long-term tracking approaches track all points from reference frame to current frame simultaneously. OMNIMOTION [53] employs NERF [35] for modeling of a dynamic scene, enabling it to produce dense tracking outputs. However, it relies on optical flow estimated between all pairs of frames followed by computationally demanding test-time training for each video sequence, making it impractical for general use.

DOT [30] estimates dense-point tracking by a two-stage process. First, sparse point-tracks are estimated by COTRACKER [27]. Then they are densified and serve as an initialization for RAFT [51] optical flow, which provides the final dense predictions. FLOWTRACK [5] chains optical flow and corrects it by error compensation module utilizing optical flow forward-backward cycle consistency.

Recently, MFT [39] extends optical flow chaining by not only tracking between consecutive frames but also between frames that are temporally distant. The chaining approach over various intervals between frames was addressed before [6,7], however MFT proposed effective strategy where a long-dense trajectory is computed by evaluating the quality of track estimates among various combinations of chained flows, enabling it to maintain accurate tracking over longer sequences than typical flow-based methods. MFT-ROMA [25] integrates wide-baseline dense matchers DKM [15] and ROMA [16] into MFT, further increasing its performance.

3. Method

We propose a dense long-term tracker based on chaining of optical flows computed from neighboring, but also from more distant frames. We will first explain the flow-chaining technique that was used before [6, 7] and recently revisited in MFT [39]. Then we describe the proposed MFTIQ tracker and how it differs from MFT.

Task and notation. The task is to track all points from an initial frame to the rest of the video. In particular, given a sequence of video frames $(I_t)_{t=1}^{T}$ with $H \times W$ resolution, the MFTIQ tracker computes *long-term* flow fields $(\mathbf{F}_{1\to t})_{t=1}^{T}$ between the initial and the current frame. For a 2D position $\mathbf{p}_A = (x, y)$ in image I_A, a flow field $\mathbf{F}_{A\to B}$ bilinearly sampled at $\mathbf{p}_A$ gives a position in the image I_B as $\mathbf{p}_B = \mathbf{p}_A + \mathbf{F}_{A\to B}[\mathbf{p}_A]$, where $\cdot[\cdot]$ is the bilinear sampling. MFTIQ also outputs binary segmentation maps $(\mathbf{O}_t)_{t=1}^{T}$ indicating that a point $\mathbf{p}_1$ is occluded or out-of-view in frame I_t when $\mathbf{O}_t[\mathbf{p}_1] > 0.5$.

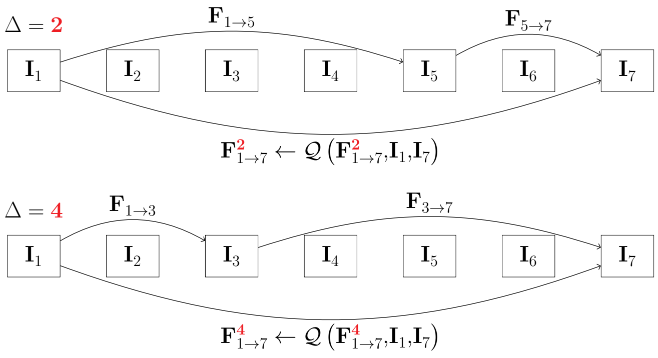

Figure 2. **Example of MFTIQ optical flow chaining strategy** for estimating the flow between I_1 and the current frame $I_{t=7}$. MFTIQ constructs a flow chain $\mathbf{F}^{\Delta}_{1\to7}$ by going through an *intermediate frame* $I_{7-\Delta}$. This is done for multiple values of Δ, here shown for $\Delta = 2$ and $\Delta = 4$. The most reliable flow chain $\mathbf{F}^{\Delta^\star}_{1\to7}$ is selected independently in each pixel based on flow quality $\mathcal{Q}\left(\mathbf{F}^{\Delta}_{1\to7}, I_1, I_7\right)$ assigned to each chain $\mathbf{F}^{\Delta}_{1\to7}$ by a neural network, which takes the flow chain, the template frame, and the current frame as inputs. Note that the flow from the template into the intermediate frame is itself a previously computed flow chain, while the flow from the intermediate frame into the current frame is output of an OF method.

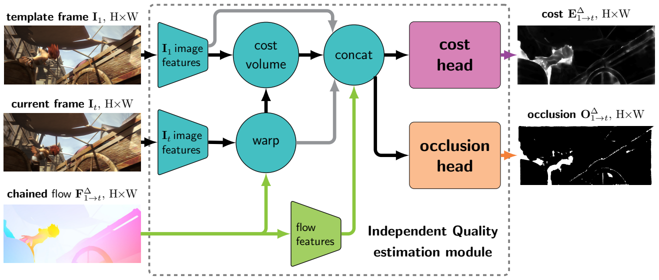
template frame $\mathbf{I}_1$, H×W
$\mathbf{I}_1$ image features
cost volume
concat
cost head
cost $\mathbf{E}^{\Delta}_{1\to t}$, H×W
current frame $\mathbf{I}_t$, H×W
$\mathbf{I}_t$ image features
warp
occlusion head
occlusion $\mathbf{O}^{\Delta}_{1\to t}$, H×W
chained flow $\mathbf{F}^{\Delta}_{1\to t}$, H×W
flow features
Independent Quality estimation module

Figure 3. **Overview of the Independent Quality (IQ) estimation network.** First, image features are extracted from the template frame I_1 and the current frame I_t. Then, the current frame features are warped using the positions given by the chained flow $\mathbf{F}^{\Delta}_{1\to t}$. The now-aligned feature maps are compared with a local (displacement up to ±3) correlation cost-volume. Finally a concatenation of the features extracted from both images and the flow are concatenated with the cost-volume and processed by two small CNNs to output the occlusion map and the cost map which together represent the quality of the input flow chain.

Both the flows $\mathbf{F}_{1\to t}$, and the occlusion masks $\mathbf{O}_t$ have the full $H \times W$ resolution. To simplify the explanation, we refer to the first frame I_1 of the video as *template frame*, I_t as *current frame*.

3.1. Optical Flow Chaining

On each frame the proposed MFTIQ constructs flow fields $\mathbf{F}^{\Delta}_{1\to t}$ as a *chain* of flow field $\mathbf{F}_{1\to(t-\Delta)}$ computed previously in an intermediate frame number $t-\Delta$ and a flow field from the intermediate to the current frame $\mathbf{F}_{(t-\Delta)\to t}$. The chaining operation samples the second flow field at the positions given by the first one such that

$$\mathbf{F}^{\Delta}_{1\to t}\left[\mathbf{p}_1\right] = \mathbf{F}_{(t-\Delta)\to t}\left[\mathbf{p}_1 + \mathbf{F}_{1\to(t-\Delta)}\left[\mathbf{p}_1\right]\right] \qquad (1)$$

Note that this can form an arbitrarily long chain of flows, as the $\mathbf{F}_{1\to(t-\Delta)}$ was itself formed as a chain of flows.

Like in MFT, the proposed MFTIQ computes a small number of such flows for varying logarithmically spaced time deltas $\Delta \in \mathcal{D} = \{1, 2, 4, 8, 16, 32\} \cup \{t-1\}$, where the $\{t-1\}$ stands for direct match, in which the optical flow is computed in single step between the template and the current frame. Flows for $\Delta \geq t$ are not computed. This results in a set of candidate optical flow chains (each represented by a single flow field), from which the most reliable one is selected according to its *quality*.

In the original MFT [39], the quality was measured by flow uncertainty and occlusion. In particular, the RAFT [51] optical flow network was extended by two additional CNN heads, estimating the occlusion state and positional uncertainty in each pixel. These two quantities were aggregated during the chaining of the flow fields to produce the overall positional uncertainty $\mathbf{U}^{\Delta}_{t}$ and occlusion state $\mathbf{O}^{\Delta}_{t}$ of the whole flow chain. Finally these were used to select the most reliable flow chain $\mathbf{F}^{\Delta^{\star}}_{1\to t}$ *per-pixel* as

$$\Delta^{\star}\left[\mathbf{p}_1\right] = \arg\min_{\Delta\in\mathcal{D}} \mathbf{U}^{\Delta}_{t}\left[\mathbf{p}_1\right] + \infty \cdot \mathbf{O}^{\Delta}_{t}\left[\mathbf{p}_1\right], \qquad (2)$$

where multiplying the occlusion by infinity ensures that a flow chain that was occluded at any time is only selected if there is no unoccluded, *i.e.*, better chain.

The MFT approach has three major drawbacks. First, the uncertainty estimation and the chaining of the uncertainty scores need to be well calibrated not to be overly optimistic or pessimistic. This is not the case in MFT which is slightly pessimistic, leading to a strong preference of flow chains with small number of links, *i.e.*, a preference of $\Delta^{\star}$ being large. In our experience, this often happens even though there are more accurate longer chains (more links with smaller $\Delta^{\star}$s) available. On the other hand, when the uncertainty of a single incorrect chain link is optimistically low, the tracker drifts and tracks a different point from that moment on. Finally, it is not straightforward to use different optical flow methods due to the direct integration of occlusion and uncertainty into the flow network architecture.

The proposed MFTIQ, addresses these issues with its *Independent Quality* (IQ) module, which replaces the problematic chaining of uncertainties with a direct estimation of the quality of the chained optical flow. The best delta is again selected per-pixel similarly to Eq. (2) as

$$\Delta^{\star}\left[\mathbf{p}_1\right] = \arg\min_{\Delta\in\mathcal{D}} \mathbf{E}^{\Delta}_{t}\left[\mathbf{p}_1\right] + M \cdot \mathbf{O}^{\Delta}_{t}\left[\mathbf{p}_1\right], \qquad (3)$$

where M is a large constant used instead of the ∞ in Eq. (3). This still ensures that unoccluded chains are always preferred, but also preserves the ordering by $\mathbf{E}^{\Delta}_{t}\left[\mathbf{p}_1\right]$ when all the candidate flow chains contain an occlusion. The cost map $\mathbf{E}$ functions analogously to the MFT flow chain uncertainty $\mathbf{U}$, but is trained with a different cost function. Cost $\mathbf{E}$ is analogous to MFT uncertainty $\mathbf{U}$ in that the lower values means higher positional accuracy.

Most importantly, we propose to estimate the cost $\mathbf{E}$ and the occlusion map $\mathbf{O}$ directly as a function of the chained flow $\mathbf{F}^{\Delta}_{1\to t}$ and the two images it relates to, I_1 and I_t.

$$\{\mathbf{E}^{\Delta}_{t}, \mathbf{O}^{\Delta}_{t}\} = \mathcal{Q}\left(\mathbf{F}^{\Delta}_{1\to t}, I_1, I_t\right). \qquad (4)$$

The independent quality estimation function $\mathcal{Q}$ is implemented as a neural network. It takes the chained optical flow, the template frame, and the current frame as inputs and estimates the cost map $\mathbf{E} \in \mathbb{R}^{H\times W}_{0+}$ and the occlusion map $\mathbf{O} \in \{0,1\}^{H\times W}$. An example diagram of the MFTIQ flow chaining and selection is shown in Fig. 2.

3.2. Flow Quality Estimation

In this section, we detail the architecture and the training of the proposed quality estimation network $\mathcal{Q}$, shown in overview in Fig. 3. First, we extract image features to produce a $\frac{H}{4} \times \frac{W}{4}$ feature map. In particular, both I_1 and I_t are processed by the DINOV2 [41] network. We bilinearly upscale the resulting coarse $\frac{H}{14} \times \frac{W}{14}$ feature map into the target $\frac{H}{4} \times \frac{W}{4}$ resolution. To add more spatially fine-grained information, we also compute the IMAGENET1K-pre-trained RESNET50 [21] CNN features and features from a shallow CNN. See Sec. A in supplementary for more details. We resize all the resulting feature maps into the $\frac{1}{4}$ resolution and compress them with a convolutional layer to have 32 channels each.

Warping + Cost Volume The next stage in our process is the formation of a local Correlation Cost Volume (CCV), which serves to measure the similarity between the corresponding (as predicted by the optical flow chain) features, while also considering adjacent pixel information. To perform this, the feature maps from the current frame I_t, are warped to the template frame I_1 using the chained optical flow $\mathbf{F}_{1\to t}$, which is scaled to match the featuremap resolution. Then a local CCV (maximum displacement of 3px on the featuremap resolution) is independently computed for each input feature map, like in FLOWNET [14].

Finally, we concatenate the feature similarities computed by the cost-volumes with the image features and features computed from the chained optical flow, more details in supplementary Sec. B. The resulting $\frac{H}{4} \times \frac{W}{4}$ featuremap with 422 channels is then used to estimate the flow-chain quality.

Flow-Chain Quality Estimation We use two three-layer CNN heads, each followed by a bilinear upsampling to the full image resolution, to estimate the cost and occlusion maps. The occlusion estimation CNN classifies each pixel as either occluded or non-occluded and is trained using standard binary cross-entropy loss, denoted as $\mathcal{L}^{\text{occl}}$.

The cost is constructed from $M = 5$ binary classifiers again trained by binary cross-entropy loss $\mathcal{L}^{\text{match}\,\theta}$. Pixels that have the flow end-point-error (EPE, euclidean distance from the ground-truth) over θpx or are occluded belong to the positive class, while the visible and precisely matched (EPE under θpx) belong to the negative class. The binary classifiers differ in the EPE threshold $\theta \in \{1, 2, 3, 4, 5\}$, ranging from 1 to 5px.

During inference, the final cost map is constructed as a weighted average of the soft (Sigmoid activation) classification maps $\mathbf{E}_\theta$,

$$\mathbf{E} = \sum_{\theta=1}^{M} 2^{\theta-1}\mathbf{E}_\theta. \tag{5}$$

The $\mathbf{E}$ should be low for well matched points and high for poorly matched or occluded points.

The overall training loss, $\mathcal{L}$, is computed as follows:

$$\mathcal{L} = \frac{1}{H \times W} \sum_{i=1}^{H \times W} \mathbf{V}_i \left(\mathcal{L}_i^{\text{occl}} + \frac{1}{M} \sum_{\theta=1}^{M} \mathcal{L}_i^{\text{match}\,\theta} \right), \tag{6}$$

where $\mathbf{V}_i$ is a binary ground-truth validity flag of pixel i.

3.3. Implementation Details

We trained the independent quality network using a synthetic dataset from the Kubric rendering tool [19]. The dataset includes 200 sequences with a variable number of static and dynamic objects rendered at a 1024×1024 resolution, each 240 frames long. The sequence length is much longer than the typically used 24 or 48 frames. We had to ensure that the objects do not become static after falling to the ground as in the default Kubric scenario, otherwise the long sequences would not bring much. To do this and keep the objects non-intersecting, we left the default Kubric physical engine to simulate the scene for 48 frames, after which we disabled it and replayed the simulated motions back and forth for the rest of the video. The camera motion is generated independently, with the panning from TAPIR and a random camera shake to introduce motion blur and make the camera movement more realistic. Due to the independent non-looping motion of the camera, the resulting video is not repetitive and information-rich for the whole duration.

The training involved sampling random image pairs with temporal separations, *i.e.*, the flow Δ, ranging from 2 to 150 frames. We generated a pre-sampled set of 20,000 training pairs with dense[1] ground truth optical flow, occlusion, and validity masks $\mathbf{V}$. The input OFs were uniformly drawn from the ground truth flow, RAFT [51], and ground-truth-initialized FLOWFORMER++ [48], computed directly between the two input images. This generates plausible long-term input OFs without tracking the whole sub-sequence in each training step.

Both the optical flow and the input images were augmented and resized to 368 × 768 pixels.

The training was conducted on a single RTX A5000 GPU for approximately one day using a batch size of 8 for 200,000 iterations, with an initial learning rate of 2.5×10^{-3} and OneCycleLR [49] learning rate policy.

We set $\Delta \in \{1, 2, 4, 8, 16, 32, t-1\}$, *i.e.*, the same as in MFT. See Sec. D in supplementary for experimental evaluation of different Δ-set configurations.

Inference-time caching To speed up the proposed MFTIQ tracker, we cache and re-use intermediate results where possible. Namely, the image features are needed multiple times per frame and especially the DINOV2 network is slow, so we cache them in GPU memory. We also cache the optical flows, which is useful when tracking from multiple query frames, like in the strided TAP-VID. If the application allows it, both the image features and the optical flows can be precomputed to get fast tracking. Timing details are reported in supplementary Sec. C.

4. Experiments

Since there is no *dense* long-term tracking benchmark, we evaluate the proposed MFTIQ tracker on standard *sparse* point-tracking datasets TAP-VID [10] and ROBOTAP [52]. We also evaluate on the POT-210 [34] dataset for planar object tracking, which contains challenging scenarios different from TAP-VID and was not leveraged for point-tracking evaluation before.

4.1. Point-Tracking Benchmark

The point-tracking is typically evaluated using three metrics introduced in TAP-VID [10]. The $<\delta^x_{avg}$measures the percentage of cases where the euclidean distance between the predicted and the ground-truth position is smaller than a threshold, averaged over five thresholds of 1, 2, 4, 8, and 16 pixels. This evaluation is done on coordinates re-scaled to 256 × 256 resolution. The quality of occlusion prediction – occlusion accuracy (OA) – is measured by standard binary classification accuracy. Finally, the average Jaccard (AJ) metric combines the position and occlusion accuracy into an unified score. Please refer to [10] for details.

[1] The original version of the Kubric tool supports only sparse ground truth generation for point-tracking tasks.

				OF runtime [ms] ↓	
method	AJ ↑	$<\delta^x_{avg}$↑	OA ↑	512x512	720x1080
MFT [39]	56.28	71.03	86.96	47	142
MFTIQ with					
RAFT [51]	60.54	74.22	84.42	47	142
GMFlow [58]	55.28	69.83	83.55	24	137
NeuFlow [63]	55.73	70.26	80.87	10	18
GMFlow-R [58]	59.57	73.38	86.49	69	335
NeuFlowV2 [62]	56.92	70.97	81.59	7	8
RAPIDFlow [36]	59.56	73.14	84.37	32	55
LLA-Flow [59]	61.78	75.18	85.44	117	475
MemFlow [13]	62.30	75.97	85.95	121	610
FFormer++ [48]	62.72	76.22	86.34	142	782
RPKNet [37]	62.78	76.61	86.39	126	174
SEA-RAFT [55]	63.51	77.18	86.22	34	105
RoMa [16]	**65.67**	**79.82**	**87.75**	714	729

Table 1. TAP-Vid DAVIS [10] (strided) evaluation with single MFTIQ model using various OF methods. The first two rows compare the original MFT with the proposed MFTIQ both using the RAFT [51] OF. The rest of the table shows MFTIQ results when used with different OF methods. Runtime of a single OF computation shown on right.

There are two evaluation modes, *first* and *strided*. In the *first* mode, trackers are initialized on the first frame where the particular point is visible and left to track until the end of the video. In the *strided* mode, every fifth frame is taken as an initialization frame. Trackers are initialized on all annotated points visible in the particular frame and left to track in both directions until the start and until the end of the video. The resulting tracks are shorter (half the video length on average), making the task simpler. Also, in the *first* mode, the query points are often on the object boundary or just after de-occlusion, further complicating the tracking.

The TAP-Vid DAVIS [10] dataset contains 30 videos from [44], mostly containing people and animals, with one or a few salient objects moving against a background. The TAP-Vid Kinetics [10] has 1189 videos from Kinetics-700 [3, 4] human action recognition dataset. The RoboTAP [52] dataset contains 265 videos of robotic arms picking up and dropping objects in a lab scenario. All the datasets have point tracks semi-automatically annotated for around 20 points in each video, including the visibility state.

Plug-n-Play Optical Flow After training the MFTIQ flow quality estimation with RAFT [51] and ground-truth initialized FlowFormer++ [48], we fixed the model and evaluated it with various different OF methods. The Tab. 1 shows that the RAFT-based MFTIQ already outperforms the original MFT. More importantly, we get even better results when using other off-the-shelf optical flows and dense matchers. The best performance is achieved with the wide-baseline matcher RoMa [16] thanks to its ability to match densely both between consecutive and between more distant frames. Table 1 also lists the runtime of the respective OF methods, measured on a RTX A5000 GPU. While the best-performing RoMa is also the slowest on smaller images, it scales better than the second-best FlowFormer++ to larger images. Depending on the intended application, one could also use a fast optical flow method, such as NeuFlow [63], for a cost of reduced tracking quality. For the rest of the experiments we use the RoMa-based MFTIQ.

Main point-tracking results The overall results of the proposed RoMa-based MFTIQ tracker are shown in Tab. 2. MFTIQ achieves the best (DAVIS) and the second-best (RoboTAP, Kinetics) position accuracy $<\delta^x_{avg}$. This is thanks to the quality of the used RoMa dense matcher. Note that RoMa was used with the original MFT in MFT-RoMa [25], however due to better flow quality estimation the proposed MFTIQ performs much better on all metrics. Also we have designed MFTIQ to be independent on the OF method, so we expect it to get better with future even-higher-quality optical flows and dense matchers without re-training.

The occlusion accuracy (OA) of MFTIQ is comparatively lower, also affecting the overall AJ score. While it is an improvement over MFT, achieving state-of-the-art occlusion accuracy is yet an open challenge.

While MFTIQ does not achieve performance as good as the recent sparse point-trackers, it tracks densely and outperforms the original MFT. Note that the point-trackers in 2 are not *causal*, *i.e*., the trackers can "see" into the future which is helpful to resolve occlusions. Both MFT and MFTIQ only use the previous frames. For dense tracking the inference time is significantly faster than methods with similar accuracy, as measured by the points-per-second metric in Tab. 2, more timing details in supplementary Sec. C.

MFTIQ vs MFT Chain Selection We further evaluate the MFTIQ chain selection and how it compares to the original MFT on TAP-Vid DAVIS. The Fig. 4 shows that the uncertainty score chaining of MFT leads to a significant preference of selecting short chains with big Δs. In particular, the optical flow matching directly between the template and the current frame ($\Delta = t - 1$) without chaining is selected with probability increasing with the current frame number. However the probability of this selection being accurate decreases rapidly during the video. On the other hand MFTIQ selects the short chains with big deltas conservatively, keeping the result accuracy high. In other words, given the same optical flows, MFTIQ selects chains leading to better accuracy.

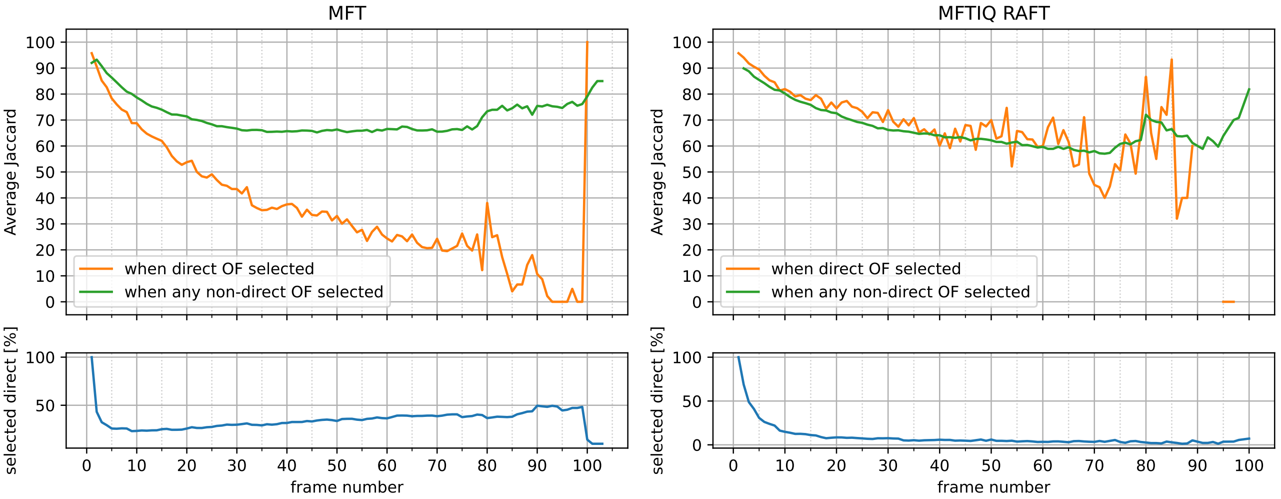

Figure 4. **Comparison of flow candidate selection in MFT *(left)* and MFTIQ *(right)*.** MFT often selects *(bottom left)* the *direct* optical flow, *i.e*. the flow chain with $\Delta = t - 1$ with probability increasing during the video. The probability of the selected *direct* flow to be accurate as measured by Average Jaccard (AJ) is, however, decreasing with time *(orange)*. In contrast, the proposed MFTIQ *(right)* chooses the direct optical flow more conservatively *(bottom)* and mostly when it has high accuracy *(orange)*. Both methods are evaluated on TAP-VID DAVIS strided using RAFT OF. Non-direct OF accuracy *(green)* represents the average over all cases, when some $\Delta \neq t-1$ was selected.

		DAVIS strided			DAVIS first			ROBOTAP first			KINETICS first		
method	PPS↑	AJ↑	$<\delta^x_{avg}$↑	OA↑	AJ↑	$<\delta^x_{avg}$↑	OA↑	AJ↑	$<\delta^x_{avg}$↑	OA↑	AJ↑	$<\delta^x_{avg}$↑	OA↑
TAP-NET [10]	† 555	38.4	53.1	82.3	33.0	48.6	78.8	45.1	62.1	82.9	38.5	54.4	80.6
COTRACKER [27]	‡ 0.8	64.8	79.1	88.7	60.6	75.4	**89.3**	54.0	65.5	78.8	48.7	64.3	**86.5**
TAPIR [12]	† 200	61.3	72.3	87.6	56.2	70.7	86.5	59.6	73.4	**87.0**	49.6	64.2	85.0
BOOTSTAP [11]	–	**66.4**	78.5	**90.7**	**61.4**	74.0	88.4	**64.9**	**80.1**	86.3	**54.7**	**68.5**	86.3
DOT [30]	2473	65.9	79.2	90.2	60.1	74.5	89.0	-	-	-	48.4	63.8	85.2
FLOWTRACK [5]	∗ 499	63.2	76.3	89.2	-	-	-	-	-	-	-	-	-
MFT [39]	10671	56.3	71.0	87.0	51.1	67.1	84.0	–	–	–	39.6	60.4	72.7
MFT-ROMA [25]	–	58.0	77.2	80.5	52.1	72.7	77.1	–	–	–	–	–	–
MFTIQ (ours)	709	65.7	**79.8**	87.8	59.9	**75.5**	84.5	60.0	77.5	85.2	48.7	65.9	85.2

Table 2. MFTIQ ROMA evaluation on TAP-VID [10] and ROBOTAP [52] benchmarks. On the KINETICS dataset, MFTIQ was evaluated only on the first 465 sequences due to time constraints. Results of the other trackers were taken from their papers and from [11] in case of ROBOTAP. The ROMA-based correspondences chained by MFTIQ provide a very good position precision ($<\delta^x_{avg}$) - best on DAVIS, second on ROBOTAP and KINETICS. The occlusion accuracy (OA) is lower, also affecting the AJ score. The speed is compared with points-per-second (PPS). Timing computed on RTX A5000. Timing with † obtained from [12] (TESLA V100), with ‡ from [31] (TESLA A100), and with ∗ from [5] (RTX 3090) and recalculated to PPS.

4.2. Planar Object Tracking Dataset

In addition to the point-tracking benchmark, we have evaluated the proposed MFTIQ on the POT-210 [34] planar object tracking dataset. The POT-210 contains 210 videos capturing rigid flat objects. The target is specified by coordinates of four control points forming a rectangle on the first frame of each video. The tracker is to output the positions of these control points on each frame of the video. There are 30 objects in total in POT-210, each captured in seven scenarios: *motion blur*, *occlusion*, *out-of-view*, *perspective distortion*, in-plane *rotation*, *scale change*, and *unconstrained* combining all of the previous challenging factors. From these only the partial occlusion factor is present in TAP-VID point-tracking benchmark.

Since there is no occlusion ground-truth available on POT-210, we evaluate only the $<\delta^x_{avg}$TAP-VID metric. Also we discard points outside the initialization region on the first frame as we only have ground-truth for the pla-

challenge	RoMa $<\delta^x_{avg}\uparrow$	MFTIQ $<\delta^x_{avg}\uparrow$
blur	88.4	$86.9_{(-1.5)}$
occlusion	99.2	$96.9_{(-2.3)}$
out-of-view	90.7	$89.2_{(-1.5)}$
perspective	96.8	$94.8_{(-2.0)}$
rotation	72.8	$96.5_{(+23.7)}$
scale	92.2	$98.0_{(+5.8)}$
unconstrained	93.5	$93.3_{(-0.2)}$
all	90.5	$93.7_{(+3.2)}$

Table 3. MFTIQ RoMa performance on POT-210 using a point-tracking metric, compared to plain RoMa. While the plain RoMa performs slightly better on some of the challenging scenarios, MFTIQ is significantly better on *rotations* and *scale change* due to the flow chaining, making it better on average – *all*.

nar object. We scale the output coordinates to 256×256 resolution as usual [10] and evaluate with the standard $1, 2, 4, 8, 16$ point error thresholds. The results in Tab. 3 indicate overall good performance, with RoMa-based MFTIQ being particularly good on *rotation* and *scale change* scenarios compared to the plain RoMa.

MFTIQ Homography Tracking. On top of the point-tracking evaluation, we also propose and evaluate a simple MFTIQ-based planar homography tracker. We initialize MFTIQ on the initial frame and let it tracking all the initial frame pixels to get dense correspondences between the first and the current frame. On each frame we mask out the background correspondences, *i.e.* outside the initial rectangle on the first frame. Finally we use the correspondences to robustly (with RANSAC [1, 17]) estimate a planar homography $\mathbf{H} \in \mathbb{R}^{3\times3}$ mapping from the initial to the current frame and transfer the control points from the initial frame into the current frame with $\mathbf{H}^*$ to get their current position.

This MFTIQ RoMa homography tracker out-performs the state-of-the-art on the POT-210 benchmark as shown in Tab. 4. The MFTIQ planar tracker performs particularly well on the *blur* subset of POT-210, which contains many frames on which trackers fail due to big motion blur. MFTIQ is able to recover from such failures by "jumping" over the problematic frames using the optical flows with bigger frame delta. Note that the resulting planar tracker is only practical for real-time online application when the optical flows and IQ module features are precomputed.

method	BL	OCCL	OOV	PERS	ROT	SC	UNC	all
LISRD [33, 42]	54.1	93.8	83.7	65.0	86.3	30.0	67.1	68.3
HDN [60]	48.8	78.2	66.1	54.4	91.4	94.8	60.7	70.9
CGN [32]	41.6	88.1	82.8	76.5	96.1	90.3	72.4	78.5
WOFT [46]	60.4	**98.6**	96.3	95.4	99.3	94.0	88.2	90.4
HVC-Net [61]	60.5	**98.6**	**97.2**	92.7	99.3	100.0	**90.1**	91.4
MFTIQ (ours)	**72.0**	**98.6**	95.0	**96.6**	**99.5**	**100.0**	89.1	**93.1**

Table 4. MFTIQ evaluation on planar tracking POT-210 [34] benchmark. Percentage of frames with alignment error under 5px threshold evaluated on the improved ground-truth from [46]. The RoMa-based MFTIQ followed by a RANSAC homography estimation on the resulting correspondences sets a new state-of-the-art performance. It achieves the most significant performance gain $+11.5\%$ on the *BLur* sequences.

5. Conclusion

In this work, we propose MFTIQ, a novel method for dense long-term tracking of points in video sequences. By leveraging flow-chaining of the Multi-Flow Tracker (MFT) and enhancing it with our Independent Quality (IQ) module, MFTIQ significantly improves tracking accuracy and flexibility compared to existing methods. Our approach effectively decouples the estimation of correspondence quality from the optical flow computations, enabling MFTIQ to handle complex occlusions and maintain accurate trajectories over extended period of time.

The "plug-and-play" nature of MFTIQ, which allows for seamless integration with any off-the-shelf optical flow method, further exemplifies its practical utility. This flexibility enables users to choose the most suitable optical flow method based on their specific performance or computational efficiency needs without the requirement for additional fine-tuning or architectural adjustments.

Our experimental results demonstrate that MFTIQ not only matches but often surpasses current state-of-the-art point-tracking methods in terms of accuracy and speed. Particularly, it shows significant improvements over the baseline MFT method. The capability of MFTIQ to operate efficiently with various optical flow methods underscores its robustness and adaptability. Looking forward, we anticipate that future advancements in optical flow technology will further enhance the performance of MFTIQ. The architecture's compatibility with evolving flow estimation techniques promises continual improvements in tracking precision and computational efficiency. We publish[2] the MFTIQ code and models.

Acknowledgements. This work was supported by Toyota Motor Europe, and by the Grant Agency of the Czech Technical University in Prague, grant No.SGS23/173/OHK3/3T/13.

References

[1] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1304–1312, 2020. 8

[2] https://github.com/serycjon/MFTIQ

[2] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2544–2550. IEEE, 2010. 1

[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 6

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6

[5] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. FlowTrack: Revisiting optical flow for long-range dense tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19268–19277, 2024. 2, 3, 7

[6] Pierre-Henri Conze, Philippe Robert, Tomas Crivelli, and Luce Morin. Multi-reference combinatorial strategy towards longer long-term dense motion estimation. *Computer Vision and Image Understanding*, 150:66–80, 2016. 1, 3

[7] Tomas Crivelli, Pierre-Henri Conze, Philippe Robert, and Patrick Pérez. From optical flow to dense long term correspondences. In *2012 19th IEEE International Conference on Image Processing*, pages 61–64. IEEE, 2012. 1, 3

[8] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4660–4669, 2019. 1

[9] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. 12

[10] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens Continente, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. TAP-Vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 2022. 1, 2, 5, 6, 7, 8, 12, 13

[11] Carl Doersch, Yi Yang, Dilara Gokay, Pauline Luc, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ross Goroshin, João Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped training for tracking-any-point, 2024. 1, 2, 7

[12] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10061–10072, October 2023. 1, 2, 7

[13] Qiaole Dong and Yanwei Fu. MemFlow: Optical flow estimation and prediction with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19068–19078, 2024. 2, 6, 12

[14] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, Dec. 2015. 2, 4

[15] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023. 3

[16] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Revisiting robust losses for dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024. 1, 2, 3, 6, 12

[17] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 8

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361, June 2012. 1

[19] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3761, 2022. 5

[20] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 59–75. Springer, 2022. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 12

[22] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 2

[23] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 668–685, 2022. 2

[24] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[25] Tomáš Jelínek, Jonáš Šerých, and Jiří Matas. Dense matchers for dense tracking. In *Proceedings of the 27th Computer Vision Winter Workshop (CVWW 2024)*, 2024. 1, 2, 3, 6, 7

[26] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2011. 1

[27] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 1, 2, 3, 7

[28] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, Ondrej Drbohlav, et al. The eighth visual object tracking vot2020 challenge results. In *European Conference on Computer Vision*, pages 547–601. Springer, 2020. 1

[29] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: connecting the dots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2024. 2

[30] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: Connecting the dots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2024. 3, 7

[31] Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. TAPTR: Tracking any point with transformers as detection. *arXiv preprint arXiv:2403.13042*, 2024. 1, 7

[32] Kunpeng Li, He Liu, and Tao Wang. Centroid-based graph matching networks for planar object tracking. *Machine Vision and Applications*, 34(2):31, 2023. 8

[33] Pengpeng Liang, Haoxuanye Ji, Yifan Wu, Yumei Chai, Liming Wang, Chunyuan Liao, and Haibin Ling. Planar object tracking benchmark in the wild. *Neurocomputing*, 454:254–267, 2021. 8

[34] Pengpeng Liang, Yifan Wu, Hu Lu, Liming Wang, Chunyuan Liao, and Haibin Ling. Planar object tracking in the wild: A benchmark. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 651–658. IEEE, 2018. 5, 7, 8

[35] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[36] Henrique Morimitsu, Xiaobin Zhu, Roberto M Cesar, Xiangyang Ji, and Xu-Cheng Yin. RAPIDFlow: Recurrent Adaptable Pyramids with Iterative Decoding for Efficient Optical Flow Estimation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2946–2952. IEEE, 2024. 6, 12

[37] Henrique Morimitsu, Xiaobin Zhu, Xiangyang Ji, and Xu-Cheng Yin. Recurrent partial kernel network for efficient optical flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4278–4286, 2024. 6

[38] Gokul B Nair, Swapnil Daga, Rahul Sajnani, Anirudha Ramesh, Junaid Ahmed Ansari, and K Madhava Krishna. Multi-object monocular slam for dynamic environments. *arXiv preprint arXiv:2002.03528*, 2020. 1

[39] Michal Neoral, Jonáš Šerých, and Jiří Matas. MFT: Long-term tracking of every pixel. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6837–6847, 2024. 1, 2, 3, 4, 6, 7, 13

[40] Michal Neoral, Jan Šochman, and Jiří Matas. Continual occlusion and optical flow estimation. In *Asian Conference on Computer Vision*, pages 159–174. Springer, 2018. 2

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4

[42] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *European Conference on Computer Vision*, pages 707–724. Springer, 2020. 8

[43] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 1

[44] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675v2*, 2017. 6

[45] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80:72–91, 2008. 2

[46] Jonáš Šerých and Jiří Matas. Planar object tracking via weighted optical flow. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1593–1602, 2023. 8

[47] Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. *arXiv preprint arXiv:2303.08340*, 2023. 2

[48] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1599–1610, 2023. 2, 5, 6, 12

[49] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 5

[50] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Janf Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. *CVPR*, 2018. 2

[51] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020. 2, 3, 4, 5, 6, 12

[52] Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. RoboTAP: Tracking arbitrary points for few-shot visual imitation. *arXiv preprint arXiv:2308.15975*, 2023. 5, 6, 7

[53] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah

Snavely. Tracking everything everywhere all at once. *arXiv:2306.05422*, 2023. 2, 3

[54] Yihan Wang, Lahav Lipson, and Jia Deng. SEA-RAFT: Simple, efficient, accurate raft for optical flow. *arXiv preprint arXiv:2405.14793*, 2024. 2

[55] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. *arXiv preprint arXiv:2405.14793*, 2024. 6, 12

[56] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5539–5548, 2020. 1

[57] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. *arXiv preprint arXiv:2404.04319*, 2024. 1, 2

[58] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022. 6, 12

[59] Jiawei Xu, Zongqing Lu, and Qingmin Liao. Lla-flow: A lightweight local aggregation on cost volume for optical flow estimation. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3220–3224. IEEE, 2023. 6

[60] Xinrui Zhan, Yueran Liu, Jianke Zhu, and Yang Li. Homography decomposition networks for planar object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3234–3242, 2022. 8

[61] Haoxian Zhang and Yonggen Ling. HVC-Net: Unifying homography, visibility, and confidence learning for planar object tracking. In *Computer Vision and Pattern Regognition (CVPR)*, 2022. 8

[62] Zhiyong Zhang, Aniket Gupta, Huaizu Jiang, and Hanumant Singh. Neuflow v2: High-efficiency optical flow estimation on edge devices. *arXiv preprint arXiv:2408.10161*, 2024. 6, 12

[63] Zhiyong Zhang, Huaizu Jiang, and Hanumant Singh. NeuFlow: Real-time, high-accuracy optical flow estimation on robots using edge devices. *arXiv preprint arXiv:2403.10425v1*, 2024. 2, 6, 12

Supplementary Materials

A. Image Feature Extraction

For the DinoV2 features we use the author-provided `ViT-S/14-reg` network checkpoint. The ResNet50 [21] network, pre-trained on the ImageNet1K [9] dataset, is used to extract features from its first three blocks: the input block, residual block 1, and residual block 2. Each output feature is up-sampled to $\frac{H}{4} \times \frac{W}{4}$ and compressed to 32 channels using a convolutional layer.

The custom image features CNN is trained from scratch, and it is inspired by NeuFlow's feature CNN [63]. Initially, an image pyramid is created by subsampling the input image at different scales (1/1, 1/2, 1/4). For each level of the image pyramid, a convolutional layer is applied with specific kernel sizes, strides, and padding to ensure the output resolution is $\frac{H}{4} \times \frac{W}{4}$ (k4:s4:p0 | k8:s2:p3 | k7:s1:p3). The outputs from each pyramid level are concatenated and compressed to 32 channels using an additional convolutional layer.

The features from all the feature providers (DinoV2, ResNet, custom CNN) are aggregated and compressed through a convolutional operation (from 5×32 channels down to 32 channels) to produce an additional *fused feature* for the cost-volume.

The impact of feature extractors on performance is demonstrated in Tab. 6. Excluding DinoV2 features causes a decrease in AJ from 65.7 to 64.6. Further removing both DinoV2 and ResNet features, leaving only the custom shallow CNN features, results in a more pronounced drop to AJ 61.5. Since the overall runtime is dominated by optical flow computation, it remains nearly unchanged (≈ -0.01 FPS) without the Dino and the ResNet backbones. Thus, we keep all three feature extractors.

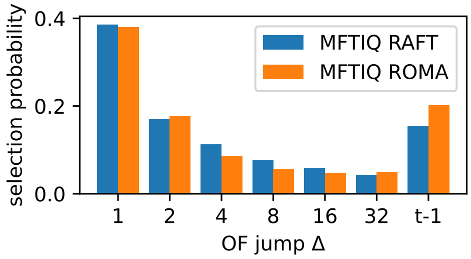

Figure 5. Probability of selecting OF with a given Δ on TAP-Vid DAVIS [10], evaluated on frames more than 32 frames distant from template. Statistics are similar for RAFT and RoMa, but long jumps $\Delta = t - 1$ are selected more often with RoMa.

B. Feature Concatenation and Flow Features

The final featuremap contains 6 (Dino, 3× ResNet, custom CNN, *fused*) cost-volumes, each flattened to 49 channels from the ± 3 range 7×7 cost-volume response maps, resulting in a total of 294 channels. In addition to that it contains 2×32 channels of the *fused features* from the template and the current frame (warped by the flow). Finally it has 64 channels of flow features derived from the input $\mathbf{F}_{1 \to t}$ flow chain by a small CNN, for a grand total of 422 channels.

C. Timing

As mentioned in Section 3.3, we implement caching for optical flow estimates and image features to improve efficiency. Table 5 reports the overall tracking timing for results shown in Tab. 1 and Tab. 2 in the paper both with standard caching during computation and with the caches pre-computed offline. With optical flow and image features computed in advance, MFTIQ runs at 3.7 FPS on 720×1080 and at over 10 FPS on 512×512 video resolution.

	FPS ↑		PPS ↑		FPS pre-computed ↑		PPS pre-computed ↑	
MFTIQ with	512×512	720×1080	512×512	720×1080	512×512	720×1080	512×512	720×1080
RAFT [51]	2.66	0.90	8234	8897	10.95	3.76	26944	33921
NeuFlowV2 [62]	5.67	2.03	16446	19348	10.56	3.59	27589	32175
RAPIDFlow [36]	3.06	1.35	9603	13058	10.65	3.49	28396	31960
GMFlow [58]	3.63	0.76	11365	7638	10.31	3.47	27304	32075
SEA-RAFT [55]	2.93	0.93	9285	9195	10.24	3.40	27296	31591
MemFlow [13]	1.16	0.29	3836	2985	10.95	3.71	27412	32907
FFormer++ [48]	1.04	0.24	3457	2437	10.47	3.76	27183	33303
RoMa [16]	0.21	0.19	709	1948	10.10	3.67	24986	32703

Table 5. **Runtime evaluation** of the whole MFTIQ tracker with various OF methods with *(right)* and without *(left)* OF and features pre-computed. All results shows processing speed in frames-per-second (FPS) and points-per-second (PPS) for two different resolutions of images. PPS were evaluated for a sequence of 80 images. In the case of pre-computed optical flow and image feature cache, speed is the same regardless of the OF method used up to a measurement noise.

method	AJ ↑	$<\delta^x_{avg}\uparrow$	OA ↑
(1) Full MFTIQ (RoMa)	65.67	79.82	87.75
(2) -Dino	64.61	79.59	87.80
(3) -Dino -ResNet	61.54	78.58	85.02

Table 6. Influence of IQ feature extractors in the MFTIQ model. The table shows the performance variations when different backbones are omitted, with the remainder of the network held constant. All models followed identical training and evaluation protocols. The evaluation was conducted using the TAP-Vid DAVIS [10] (strided) dataset.

				runtime [FPS] ↓	
Δ-set hyper-parameter	AJ ↑	$<\delta^x_{avg}\uparrow$	OA ↑	512x512	720x1080
$\Delta \in \{1, 2, 4, 8, 16, 32, t-1\}$	65.67	79.82	87.75	0.21	0.19
$\Delta \in \{1, 4, 16, t-1\}$	65.50	79.57	87.42	0.35	0.32
$\Delta \in \{1, 8, 32, t-1\}$	59.03	72.79	82.34	0.35	0.32
$\Delta \in \{t-1\}$	57.46	70.08	78.73	1.31	1.14
$\Delta \in \{1\}$	54.67	70.99	73.35	1.31	1.14

Table 7. Ablation of different sets of Δ used for optical flow chaining. The default set of Δs *(first row)* (same as in MFT) performs the best. The base-4 *(second row)* set achieves a better speed / performance trade-off. MFTIQ RoMa evaluated on TAP-Vid DAVIS [10] (strided). Performance measured by average Jaccard (AJ), position accuracy ($<\delta^x_{avg}$), and occlusion accuracy (OA). Speed of tracking densely measured by average frames per second (FPS).

D. Delta Set Ablation

Tab. 7 shows the effect of using different sets of Δs. Our default base-2 configuration, $\Delta \in \{1, 2, 4, 8, 16, 32, t-1\}$, follows the MFT setup [39]. However, we found that using a base-4 set, $\Delta \in \{1, 4, 16, t-1\}$, achieves a 1.6× speedup with only a minimal performance decrease on the TAP-Vid DAVIS dataset [10]. Both direct matching between the template and the current frame ($\Delta \in \{t-1\}$) and consecutive frame chaining ($\Delta \in \{1\}$) result in a significant performance decrease across all evaluated metrics.

We have also evaluated (Fig. 5) the frequency of selection for each Δ in MFTIQ RAFT and MFTIQ RoMa in the default Δ-set. The results show similar statistics between the two OFs, though the direct jump ($\Delta = t-1$) is selected more frequently in RoMa. This is expected since the RoMa was trained on wide-baseline matching data, making it more reliable with more distant pairs of frames. Only frames beyond timestep 32 are evaluated to avoid biasing the results with smaller Δs at the beginning of the sequence, where longer Δs are not yet available for matching.