Planar Object Tracking via Weighted Optical Flow

Jonáš Šerých, Jiří Matas

CMP Visual Recognition Group, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague

{serycjon,matas}@fel.cvut.cz

Abstract

We propose WOFT – a novel method for planar object tracking that estimates a full 8 degrees-of-freedom pose, i.e. the homography w.r.t. a reference view. The method uses a novel module that leverages dense optical flow and assigns a weight to each optical flow correspondence, estimating a homography by weighted least squares in a fully differentiable manner. The trained module assigns zero weights to incorrect correspondences (outliers) in most cases, making the method robust and eliminating the need of the typically used non-differentiable robust estimators like RANSAC. The proposed weighted optical flow tracker (WOFT) achieves state-of-the-art performance on two benchmarks, POT-210 [23] and POIC [7], tracking consistently well across a wide range of scenarios.

1. Introduction

In this paper, we address the rigid planar object tracking problem, which is a specific subtopic of visual object tracking. Given an object identified in the first frame, a tracker should output the tracked object pose or absence of the object in the frame, on every subsequent frame of a video sequence. In a general model-free setting, the tracker has no prior knowledge about the target class except for target-specific information coming from the first frame initialization. In standard tracking benchmarks, such as OTB [46], VOT [18, 20], LASOT [11], TRACKINGNET [30], or YT-BB [36], the object pose is represented by axis-aligned or rotated bounding boxes. Tracking with segmentation mask representation has gained popularity in recent years, with benchmarks such as VOT2020 [19], DAVIS [33, 35], and YOUTUBE-VOS [47].

In planar rigid object tracking, the object pose is related to its initial pose by an 8 degrees-of-freedom (DoF) homography when using a perspective camera, and the target is fully specified by the initialization mask. Planar trackers can output precise 8-DoF object poses, enabling applications not possible with bounding-box or segmentation mask level trackers, in areas such as film post-production, visual servoing [50, 5], SLAM [41], or markerless augmented reality [44, 34, 39]. Man-made objects are often either completely planar or consist of planar surfaces, allowing for planar object tracking in a wide range of scenarios.

Current state-of-the-art methods struggle on seemingly toy-like sequences in standard planar object tracking datasets, POT-210 [23] and POIC [7]. The target planarity poses challenges, e.g., strong perspective distortion, significant illumination changes caused by specular highlights, and motion blur caused by a shaking hand-held camera.

In this work, we introduce a novel model-free planar object tracker. The proposed method estimates dense correspondences between the template (initial image) and the current image with a deep optical flow network. A novel homography estimation module then assigns a weight to each optical flow correspondence, and a homography is estimated as a solution to a weighted least squares problem. The network assigns low weights to incorrect flow vectors and thus it is not necessary to use robust outlier detection algorithms like RANSAC.

Using dense OF correspondences has several advantages. First, OF estimation is well researched and high-quality methods are available off-the-shelf. Second, the dense per-pixel correspondences help on low-textured objects, where sparse key-point correspondences fail. Finally, having dense correspondences enables us to compute a homography correspondence support set and detect a tracking
failure if the support is small.

The proposed homography estimation procedure is fully differentiable, allowing us to train both the weight estimator and the optical flow network using homography supervision. The main contributions of this work are the following.

- We propose a novel fully differentiable homography estimation neural network module.
- We propose a novel planar target tracker employing the weighted flow homography estimation (code public).
- The proposed tracker sets a new state-of-the-art on the POT-210 [23], POT-280 [22], and POIC [23] datasets, performing well across all challenge types. On POT-210, the tracker error is half of the error of the best competing method.
- We analyze the ground truth on the POT-210 dataset and publish a precise re-annotation of its subset. The inaccuracy of the original annotation accounted for half of the errors of the proposed tracker.

2. Related Work

General visual object tracking methods have been improving consistently, with deep-learning-based trackers dominating classical methods [21, 19]. In contrast, planar object trackers have only recently started using deep learning.

The homography trackers can be roughly divided into three main categories [23]: keypoint methods, direct methods, and deep methods. Traditional keypoint-based tracking consists of three steps: (i) keypoint detection and description using, e.g., SIFT [27] or SURF [3], (ii) keypoint matching by nearest neighbor search in the descriptor space, and (iii) robust homography estimation with RANSAC [12]. The SOL [14] tracker uses SVM to learn keypoint descriptors and PROSAC [9] ordering. In GRACKER [45], the keypoints are not matched independently based only on descriptor similarity, but instead a graph-matching approach is used. The OBD [29] tracker uses ORB keypoints for target detection and optical flow tracking. In the POT-280 [22] benchmark, the authors compare several deep-learning based homography trackers. The best ones use the SIFT keypoint detector, a deep learning descriptor such as GIFT [25], MATCHNET [13], SOSNET [43], or LISRD [32], followed by RANSAC.

Direct methods formulate the tracking task as image registration. Given the current frame, they attempt to find a homography warping that optimizes the alignment of the current frame with the object in the initial frame. In the classical Lucas-Kanade [28] and the Inverse Compositional [2] methods, the warp quality is measured directly on the image intensities by a sum of squared differences. The ESM [4] tracker avoids the costly computation of Hessian in Lucas-Kanade by using an efficient second-order minimization (ESM) technique. GO-ESM [3] improved robustness to illumination changes by adding a gradient orientation feature on top of the image intensities and generalizing the ESM tracker to multidimensional features. The GOP-ESM [7] tracker extends GO-ESM with a feature pyramid and a coarse-to-fine iterative approach. Chen et al. [6] proposed to use the ESM algorithm as a differentiable layer in a siamese neural network architecture. The ESM layer iteratively aligns the template and the query frame feature maps obtained from a ResNet-18 [15] backbone pre-trained on IMAGENET. The whole architecture is then fine-tuned on image pairs synthesized from the MS-COCO dataset [24]. Direct methods perform well on the POIC [7] dataset, but typically fail on motion blur, partial occlusions and partially out-of-view targets, e.g., in the POT-210 [23] dataset.

Deep learning homography estimation is typically done by regression of four control points. The HomographyNet [10] and UDH [31] feed a concatenated pair of homography-related images through a CNN and formulate the homography estimation as direct regression of four control points. Rocco et al. [37] proposed to regress the four homography control points from a correlation cost-volume containing each-to-each similarities between Siamese VGG-16 [40] feature maps. The four-point regression is also used by the recently proposed HDN [49] method, which decomposes the homography into a similarity transform and a homography residual. These control-point regression methods struggle with occlusions and often assume that the whole images are related by a homography, and do not distinguish between the target and the background motion. The PFNET [48] uses a custom convolutional architecture to estimate a dense flow field, which is then used in RANSAC, making the method not differentiable and end-to-end training not possible.

3. Method

We propose a weighted flow homography module (WFH) that assigns a flow weight $w_i \in [0, 1]$ to each OF correspondence and estimates a homography using a weighted least squares formulation (Sec. 3.1). The WFH is differentiable, making end-to-end training of both the WFH and the OF network possible. In Sec. 3.2, we propose a weighted optical flow tracker (WOFT) built around the WFH homography estimator.

3.1. Weighted Flow Homography Module

The idea of the WFH module is to compute a flow weight $w_i \in [0, 1]$ for each optical flow vector and to predict a homography by solving a weighted least squares (LSq) problem. The standard least squares homography fitting is sen-
sitive to grossly incorrect correspondences (outliers). This is usually addressed by RANSAC, which uses repeated hypothesis sampling to find a homography and its outlier-free correspondence support set. The WFH instead eliminates outliers by setting their flow weights close to zero, allowing for a robust, single iteration, and fully differentiable weighted least squares fitting.

We process a pair of images with an optical flow estimation network, such as RAFT [42] to get OF correspondences \((p_i, p'_i)\), where \(p_i = (x_i, y_i)\) is a position in one image and \(p'_i = (x'_i, y'_i)\) the corresponding position in the second image. We then pass a suitable inner representation of the OF network to a weight-estimation CNN that predicts the flow weight \(w_i\) for each OF vector. Finally, we estimate homography by solving a system of equations by weighted least squares. First, we introduce plain least squares homography by solving a system of equations by weighted least squares. We solve the least squares problem

\[
\min_{\mathbf{h}} \sum_{j=1}^{2N} w_j \| \mathbf{A}_j \mathbf{h} - \mathbf{b}_j \|_2^2
\]

with the corresponding estimated flow weight \(w_i\) and find

\[
\min_{\mathbf{h}} \sum_{j=1}^{2N} w_j \| \mathbf{A}_j \mathbf{h} - \mathbf{b}_j \|_2^2 = \min_{\mathbf{h}} \sum_{j=1}^{2N} \left( \sqrt{w_i A_i} \right) \| \mathbf{h} - \left( \sqrt{w_i b_i} \right) \|_2^2
\]

The weighted problem (3) is transformed into non-weighted (Eq. (2)) by multiplying each row of \(\mathbf{A}\) and each element of \(\mathbf{b}\) by the square root of the corresponding weight \(\sqrt{w_i}\).

**Training WFH.** We train the WFH weight estimation CNN using a loss function on the predicted homography. We warp points forward by the ground truth homography \(\mathbf{H}_{GT}\) then backward by the inverse of the estimated \(\mathbf{H}\) and finally compute L1 loss on the projection errors as:

\[
L(\mathbf{H}) = \frac{1}{N} \sum_{i=1}^{N} \| p_i - \mathbf{H}^{-1} \mathbf{H}_{GT} p_i \|_2
\]

Both the optical flow network and the flow weight estimation CNN are trained using a single loss function, and we do not use additional direct supervision of the flow weight predictor. The learned flow weights resemble a keypoint detector output (corners, well-textured patches), but with information from both images, therefore giving low weights on occlusions or significant appearance changes as shown in figure 2.

**Weight Estimation CNN** The proposed WFH module operates on the correlation cost-volume pyramid of the RAFT [42] optical flow estimator, but the idea is applicable to other OF networks (Sec. 4.2). RAFT computes a correlation volume \(\mathbf{C}^1 \in \mathbb{R}^{H/8 \times W/8 \times H/8 \times W/8}\) that captures the similarity between all pairs of feature vectors extracted from the two input images. Next, they construct a 4-layer correlation pyramid \(\{\mathbf{C}^1, \mathbf{C}^2, \mathbf{C}^3, \mathbf{C}^4\}\). Finally, \(9 \times 9\) patches
Figure 3. The WOFT tracker pre-wars the current frame using the last good pose (1). It then estimates a homography between the template and the pre-warped frame (2) and the reliability of the estimated homography is assessed (3). When the estimate is not reliable (‘lost’ state) a homography based on a local flow (4) is returned instead.

centered on current flow vector estimates are sampled from this pyramid and processed by a neural network that produces a flow vector update. This is repeated several times to produce the final optical flow field.

In WFH we sample the correlation pyramid once more on the final OF positions, resulting in a $9 \times 9 \times 4$ feature map for each flow vector in the spatial resolution of $H/8 \times W/8$. To capture the global context, we then append an additional channel containing the mean correlation volume response $M(i, j) = \frac{H}{8} \sum_{m=1}^{W/8} C_{i,j,k,l}$ for the given position $(i, j) \in \{1, \ldots, H/8 \} \times \{1, \ldots, W/8 \}$ in the first input image feature map. We process the resulting features $f_{i,j} \in \mathbb{R}^{9 \times 9 \times 5}$ with a three-layer convolutional network (kernel size 3, 128 output channels, ReLU) followed by a $1 \times 1$ convolution (single output channel) and global average pooling. Finally, we up-sample the results with the RAFT up-sampling module and apply a sigmoid activation to get a $H \times W$ score map with weights between 0 and 1.

### 3.2. Homography tracker

We propose a planar object tracker based on the weighted flow homography module, WFH. Our weighted optical flow tracker, denoted WOFT, consists of four main parts as shown in Fig. 3.

First, we apply a pre-warping technique to reduce large pose differences, which are not handled well by OF methods. The current video frame $I_t$ is pre-warped (1) by the homography from the last reliable frame $I_G$, with $G = 0$ initially. The pre-warp $\tilde{I}_t = \mathcal{W}(H_0^{-1}G, I_t)$ (1) reduces the pose difference between the template and the current images, resulting in a motion similar to the typical $I_{(t-1)} \rightarrow I_t$ optical flow scenario. The possible appearance difference between the template and a temporarily distant frame (caused mainly by illumination changes and motion blur) is implicitly handled by the optical flow feature encoder.

Second, we compute the global optical flow (2) between the template frame $I_0$ and the pre-warped current frame $\tilde{I}_t$ and the corresponding flow weights. We mask the flow correspondences, only leaving the ones starting inside the template mask and ending inside the current image. To speed up the homography estimation, we randomly subsample the correspondences, only keeping 500. We then estimate homography $H_{0 \rightarrow t}$ using weighted least squares as described in Sec. 3.1. Computing the homography between the template and the pre-warped current frame prevents error accumulation and target drift (Sec. 4.2).

We pass the weighted optical flow together with the computed homography to a state logic block (3) that decides whether the tracking was successful or not. The lost/not-lost decision is made based on the support set size of the estimated homography. In particular, with optical flow correspondences $(p_i, p_i')$ we warp each position $p_i = (x_i, y_i)$ using the homography $H_{0 \rightarrow t}$ and compute the Euclidean distance to the position $p_i' = (x_i', y_i')$. The i-th correspondence is an inlier when $\|\mathcal{W}(H_{0 \rightarrow t}, p_i) - p_i'\| \leq 5$ pixels – a standard threshold on planar tracking benchmarks [23, 24]. We declare the tracker lost when it has a small support set, i.e. less than 20% inliers.

When the tracker is not lost, we return $H_{0 \rightarrow t} = H_{0 \rightarrow G}^{-1}H_{0 \rightarrow t}$ and update the last good frame index used for pre-warping $G = t$. When the tracker is lost, we make a second attempt to estimate the pose using a local optical flow $I_{(t-1)} \rightarrow I_t$. The local flow tends to drift, but it helps to keep track of the target pose in the short term. The temporarily close input images are close in appearance (similar illumination, similar motion blur, etc.). We estimate $H_{(t-1) \rightarrow t}$ (4) by weighted least squares as described above and output $H_{0 \rightarrow t} = H_{(t-1) \rightarrow t}H_{0 \rightarrow (t-1)}$. Moreover, when the tracker is lost for more than 10 frames, we reset the pre-warping last good frame index $G = 0$. The target pose can change significantly over the 10 frames, making the pre-warp information outdated. Moreover, a bad pre-warp homography can ruin any chance of recovering, e.g. an outdated strong perspective change pre-warp distorts the current target area beyond being recognizable, and the identity homography with $G = 0$ is the safest choice.

### 3.3. Implementation details

For optical flow, we use the author-provided RAFT checkpoint trained on Sintel. We then train the weight estimation CNN for 10 epochs on a synthetic dataset with 50000 image pairs. We generate the training set by repeatedly sampling a random MS COCO [24] image and warp-
for 2 epochs, starting from the learning rate $1e^{-5}$. Finally, we fine-tune the whole network, including RAFT and flow masking, and subsampling cost an additional 7ms. Image pre-warping (done on CPU), optical flow computation (275ms). The majority of time is spent on the optical flow computation (275ms). The weight computation (2ms), the weight up-sampling (1ms), and the least squares homography estimation (5ms) take negligible time. Image pre-warping (done on CPU), optical flow masking, and subsampling cost an additional 7ms.

A faster variant WOFT$_{ls}$ downscales the input images to $H/s \times W/s$ and rescales the output homographies to the original resolution.

4. Experiments

We evaluate the proposed tracker on two standard planar object tracking datasets, POT-210 and POIC and show that it consistently achieves high accuracy and robustness.


POIC [7]: the Planar Objects with Illumination Changes dataset consists of 20 sequences of varying length giving a total of 22971 frames. The dataset contains sequences with translation, in- and out-of-plane rotations, and scale changes, but mainly focuses on strong specular highlights and other significant illumination changes, making it complementary to POT-210.

Evaluation protocol: On both POT-210 and POIC, a tracker is initialized on the first frame and left to track till the end of the sequence. The alignment error $e_{AL}$ is computed for each annotated frame. Given four reference points $x_i \in X$ in the first frame, the alignment error is defined as root-mean-square error between their projection into the current frame by the ground truth homography $H^*$ and by the tracker homography $H$,

$$e_{AL}(H; H^*, X) = \sqrt{\frac{1}{4} \sum_{i=1}^{4} (W(H^*, x_i) - W(H, x_i))^2}$$

with $W(H, x)$ representing the projection of vector $x$ by a homography $H$. Tracker precision is measured as a fraction of frames with $e_{AL} \leq 5$ px (P@5 score). Additionally, we measure $e_{AL} \leq 15$ px (P@15 score), corresponding to the fraction of frames with target not tracked perfectly, but not completely lost either – we call this robustness regime.

4.1. Ground truth quality

During the analysis of WOFT performance on POT-210, we found that in many cases the ground truth (GT) annotations are less accurate than the official 5px error threshold. We have performed reannotation of a subset of the POT-210 dataset to measure the original GT quality and provide more accurate estimates of tracker performance, see Fig. 4. Our annotation tool shows the template, the object on the current frame warped with the current annotation, and, most importantly, an alignment visualization. We convert both the template frame and the current frame to grayscale and overlay the warped frame over the template, putting the template into the green channel and the current frame into the red and blue channels. This allows for very precise alignment over the whole extent of the target, unlike the annotation interface used for the original annotation (Fig. 4 in [23]). We have fully manually reannotated frames 82, 172, 252, 332, and 412 from each sequence, without seeing the WOFT estimated poses and the new GT will be made publicly available. More examples of the reannotation overlay are in the supplementary materials. The alignment error of the original GT evaluated
on our re-annotation is 3.63 on average, and worse than the official 5px threshold in 15% cases.

### 4.2. Ablation study

In Table 1, we show the impact of various design choices of WOFT on POT-210 performance (both on the original and the more accurate re-annotated ground truth). First, we show the importance of computing the optical flow between the template and the pre-warped current frame. In rows 1, 2 we only use the local flow (from $I_{t-1}$ to $I_t$). The tracker drifts and quickly loses the target, resulting in overall poor performance. A big performance improvement is achieved by using global flow (from $I_0$ to $I_t$) and always using the previous frame for pre-warping (rows 3, 4). Another boost in performance is achieved with the controlled pre-warping (rows 5 - 9), where the local flow is used when the global flow fails and the pre-warp homography is reset when the target is ‘lost’ for more than 10 frames.

Using the weighted least squares homography estimation consistently improves the performance – compare row 2 to row 1 (P@5 +1.3), row 4 to row 3 (P@5 +10.7), and row 6 to row 5 (P@5 +8.3). In row 7, we used the same settings as in WOFT (row 6), but without the RAFT fine-tuning, resulting in a drop in P@5 (−7.4). We have also experimented (row 8) with estimating homography by weighted iterative reweighted least squares (IRLSq) instead of ordinary weighted least squares. We have set the IRLSq to optimize the Huber loss (also called smooth L1 loss) which is more robust to outliers than least squares. This did not change the performance (w.r.t. row 6), indicating that our estimated weights already take care of outliers and the robust estimator is not necessary. Next, we compare RANSAC (row 9) with the proposed WOFT (row 6). The weighted least squares approach achieves better results (P@5 +0.9) in a single differentiable pass.

Rows 10-12 show WOFT with LITEFLOWNET2 [17] flow (details in supplementary) instead of RAFT. Again, the weighted LSq estimator (row 12) works better than plain LSq (row 10) or RANSAC (row 11).

### 4.3. Weights Evaluation

Figure 5 shows how the learned weights correlate with the optical flow quality. Low-textured areas and ambiguous features are often assigned a low weight (Fig. 3) even when the corresponding optical flow is correct. Importantly, the incorrect flow vectors are assigned low weights.

### 4.4. POT-210 and POT-280 evaluation

We compare WOFT method against the best performing methods on the POT-210 [23] dataset. Namely key-point methods: SIFT [27], OBD [29], and GRACKER [45], deep control point regression HDN [49], the deep learning based methods evaluated in [22]: SOSNET [43], SUPERGLUE [38], LISRD [32], the direct methods: GOP-ESM [7], and Siam-ESM [6] (deep + direct).

The proposed WOFT achieves state-of-the-art on the POT-210 dataset. The Alignment Error $\epsilon_{AL}$ results are depicted on Fig. 6 and in Tab. 2. Evaluated over all 210 sequences (all plot) The WOFT tracker performs better than all the other methods, both in terms of accuracy (P@5), and robustness (P@15). More than half of the 5px threshold errors of WOFT are explained by imprecise GT. The WOFT$_{13}$ variant operating on $H/3 \times W/3$ images runs close to real-time and achieves state-of-the-art accuracy (details in supplementary). WOFT also achieves top results on POT-280 [22] (76.9 P@5, 93.2 P@15), see supplementary.
4.5. POIC evaluation

We compare (Fig. 7) the WOFT tracker performance with the top methods evaluated on the POIC [7] dataset. Apart from the methods evaluated on POT-210, this includes SOL [14] and BIT-PLANES [11]. WOFT achieves state-of-the-art results with 96.1 P@5 and 98.0 P@15. More results are in supplementary materials. See Fig. 8 for WOFT output examples on both POT-210 and POIC.

5. Discussion and Limitations

The WOFT tracker handles partial occlusions, a moderate amount of motion blur, and the illumination changes and lack of texture present in the POIC dataset. In comparison, other methods performing well on POIC (SiamESM [6], GOP-ESM [7]) have low performance on POT-210 and POT-210 unconstrained
vice versa (LISRD [32], SIFT [27]). WOFT does not feature a re-detection scheme and estimates only the residual transformation after the pre-warp step. This causes issues when the tracker gets lost for more than 10 frames on the scale subset. After resetting the pre-warp source frame to $G = 0$ (pre-warp with an identity homography), the scale component of the residual transformation is sometimes bigger than what the flow network can handle (see Fig. 8).

We tested the proposed WFH homography method on the RAFT OF network, which is accurate (Fig. 5), but slow (275ms per frame). However, the OF estimation is an active area of research and we expect new accurate and fast methods to be published in the future. The core idea of WFH – flow weights computed from an OF cost-volume and a differentiable homography estimation with weighted LSq – is applicable to other OF methods. The ablation study results with RAFT replaced by LITEFLOWNET2 support this claim. We also proposed a simple WOFT_{1,3} variant that operates fast (19.2 FPS) and still achieves state-of-the-art.

6. Conclusions

We proposed a novel formulation of deep homography estimation by weighted least squares. The weighted flow homography (WFH) module is differentiable and can be trained end-to-end together with an optical flow network that provides dense correspondences. A novel planar object tracker, called WOFT, that uses WFH was evaluated on two complementary planar object tracking benchmarks and sets a new state-of-the-art on POIC, POT-210, and POT-280. On POT-210 it outperforms all other published methods by a large margin. Inaccuracy of the POT-210 ground truth accounted for half of the WOFT errors. We publish the WOFT code, trained model and an improved GT annotation of a POT-210 subset.

Acknowledgements. This work was supported by Toyota Motor Europe, by CTU student grant SGS20/171/OHK3/3T/13, and by the Research Center for Informatics project CZ.02.1.01/0.0/0.0/16_019/0000765 funded by OP VVV.

[https://cmp.felk.cvut.cz/~serycjon/WOFT]
References


