Homography Decomposition Networks for Planar Object Tracking

Xinrui Zhan[1], **Yueran Liu**[1], **Jianke Zhu**[1,2*], **Yang Li**[3*]

[1] Zhejiang University
[2] Alibaba-Zhejiang University Joint Institute of Frontier Technologies
[3] East China Normal University
{xrzhan, liuyueran97, jkzhu}@zju.edu.cn, yli@cs.ecnu.edu.cn

Abstract

Planar object tracking plays an important role in AI applications, such as robotics, visual servoing, and visual SLAM. Although the previous planar trackers work well in most scenarios, it is still a challenging task due to the rapid motion and large transformation between two consecutive frames. The essential reason behind this problem is that the condition number of such a non-linear system changes unstably when the searching range of the homography parameter space becomes larger. To this end, we propose a novel Homography Decomposition Networks (HDN) approach that drastically reduces and stabilizes the condition number by decomposing the homography transformation into two groups. Specifically, a similarity transformation estimator is designed to predict the first group robustly by a deep convolution equivariant network. By taking advantage of the scale and rotation estimation with high confidence, a residual transformation is estimated by a simple regression model. Furthermore, the proposed end-to-end network is trained in a semi-supervised fashion. Extensive experiments show that our proposed approach outperforms the state-of-the-art planar tracking methods at a large margin on the challenging POT, UCSB and POIC datasets. Codes and models are available at https://github.com/zhanxinrui/HDN.

Introduction

Planar object tracking is a fundamental problem in many AI applications, which aims at estimating the transformation of a planar object in videos. A reliable planar tracker is usually employed as a reasonable surrogate for 3D structure tracking methods, such as augmented reality, visual servoing, and robotics. Despite the encouraging progress (Chen et al. 2019; Wang and Ling 2018; Pautrat et al. 2020; Liu et al. 2019) has been made in past decades, tracking planar object robustly is still a challenging problem due to the appearance changes and large displacements between the consecutive video frames.

Matching the salient keypoints between two images is typically employed to recover the homography, which is widely used in planar object tracking. Since the keypoints-based method cannot make full use of the whole image, jitters commonly occur during tracking. Moreover, the conventional feature detector SIFT (LoweDavid 2004) often fails in the case of large perspective transformation. The deep learning-based methods such as LISRD (Pautrat et al. 2020) and GIFT (Liu et al. 2019) introduce the invariant descriptors to tackle this problem, which still suffer from the issue of insufficient keypoints to rebuild the homography. On the other hand, the appearance-based approaches are able to take full advantage of information from the whole image. However, it is easy to get stuck in the local optima. Although the gradient-based method such as GOP-ESM (Chen et al. 2019) can deal with the illumination changes, they are not robust to motion blurs. In addition, large displacements may lead to the fractional sampled object patches, which are harmful to the minimization-based approaches.

Figure 1: A comparison of our approach with state-of-the-art trackers. Our method HDN obtains the robust tracking results comparing to the keypoint-based method (SIFT, LISRD) and direct approach (GOP-ESM).

In contrast to the conventional approaches, the deep learning-based method becomes the promising direction for planar object tracking. (DeTone, Malisiewicz, and Rabinovich 2016; Nguyen et al. 2018) directly regress the corners' offsets of the rectangular region, however, it cannot accommodate the large transformation. Specifically, directly estimating the homography parameterized by corner offsets with eight coefficients, the condition number of the system becomes extremely large (up to $5e^7$), which makes the system very unstable. Therefore, any perturbation may lead to tracking failure, especially with large displacement. It can be found that the condition number becomes lower with fewer parameters to be estimated. To predict four transformation parameters of an object, (Black and Jepson 2004; Li et al.

*corresponding authors

2019) employ the rigid motion along with scaling model for the region-based trackers, which obtain the very robust tracking result. However, they do not estimate the homography transformation with eight parameters for the planar object tracking.

In this paper, we propose a novel Homography Decomposition Networks approach to planar object tracking in video sequences, which decomposes the homography transformation into two groups, including a similarity group and a residual group. By estimating the similarity group firstly, the condition number of the entire system reduces substantially. Inspired by group convolution theory (Henriques and Vedaldi 2017), we employ a rotation-scale invariant convolution operator to predict similarity robustly. Intuitively, this gives a very robust and good initial guess of where the object is. Then, the second stage predicts the residual transformation through the semi-supervised regression, where the residual transformation is the residual group with the extra error from the first stage. To the best of our knowledge, this is the first work to decompose homography into two stages in deep learning.

The contribution of our work can be summarized as below: 1) a novel deep planar object tracker by decomposing the homography matrix into two groups; 2) a deep similarity equivalent estimator robustly recovers the similarity transformation; 3) an end-to-end differentiable semi-supervised model with negative samples loss bridges the gap from homography estimation; 4) experiments on challenging POT, UCSB, and POIC datasets, show that our method performs better than the state-of-the-art approaches.

Related Works

Planar Object Tracking

The conventional planar trackers can be roughly categorized into two groups.

One category is the keypoint-based methods, which often adopt the salient feature point detection technique such as SIFT (LoweDavid 2004) and match the object across images using the keypoints. Then, the homography is estimated from those inlier correspondences between the reference frame and target image. Gracker (Wang and Ling 2018) establishes the correspondences in a geometric graph matching manner, which adopts the match-filtering and optimization strategy in order to bring robustness for more scenes and transformation. SuperGlue (Sarlin et al. 2020) introduces a deep attentional GNN with a keypoint matching layer. LISRD (Pautrat et al. 2020) presents a novel CNN-based dense descriptor, which is invariant to a group of transformations. GIFT (Liu et al. 2019) uses joint learning to select the right invariance to match the descriptors.

Another group is the region-based approaches. ESM (Benhimane and Malis 2004) reduces the hessian matrix computation in optimization while still retaining the high convergence rates. However, SSD-based ESM is not robust to illumination changes. To tackle this issue, GOP-ESM (Chen et al. 2019) proposes an illumination insensitive ESM method using the gradient pyramid. Although having achieved encouraging results, these trackers usually suffer from appearance changes such as occlusions and motion blurs. UDH (Nguyen et al. 2018) and (Tsai and Feng 2019) approximate the corner offsets to compose the homography matrix based on the global region feature. These deep learning-based methods can only deal with the small local transformation, which are not robust to the drastic changes. Our proposed approach overcomes this issue through homography decomposition networks. Specifically, the first similarity component accommodates the large motion including scale and in-plane rotation, and the second stage predicts the small residual transformation.

Visual Object Tracking

The planar tracker can be viewed as a special case of general object tracking methods. Recently, deep learning-based approaches dominate the leaderboard of mainstreamed benchmarks. SiamFC (Bertinetto et al. 2016) first uses Siamese Network to generate two feature maps, whose correlation is further employed to locate the center of the target. For the scale changes, multi-scale pooling is adopted, which incurs the computational burden. Anchor-free-based trackers (Chen et al. 2020; Zhang et al. 2020b; Guo et al. 2020) acquire the probability to be the center of the target of each point from the feature map. Another crucial problem is that the conventional CNN is not equivalent to transformation except translation, which makes it hard to predict the common transformation such as rotation. To deal with this problem, RE-SiamNets (Gupta, Arya, and Gavves 2021) use the rotation equivalence of steerable filters. Despite the easiness to insert to other trackers, its sampling scheme is time-consuming. In this paper, we exert the correlation and the anchor-free-based method used in visual tracking to deal with the robust similarity estimation problem in HDN.

Method

The objective of the planar object tracking is to recover the underlying homography transformation from the template image T to the i_{th} input frame $I_i \in \mathbb{R}^{n\times n}$, where the size of I_i is $n \times n$. Let $\mathbf{p} = (u, v, 1)^T$ be the homogenous coordinates vector of a pixel, and $I(\mathbf{p})$ denotes the pixel value of $\mathbf{p}$ in image I.

Let $\mathbf{H}_i$ be the estimated transformation matrix (from T to I_i). $\mathbf{P} = [\mathbf{p}_{lt}, \mathbf{p}_{rt}, \mathbf{p}_{rb}, \mathbf{p}_{lb}]$ is the four corner points quadrilateral coordinates of an object, which is a (3×4) matrix. Specially, $\mathbf{P}_i$ represents the coordinates of the target object in I_i, and $\mathbf{P}_T$ for the object coordinates in T. Therefore, the prediction of $\mathbf{P}_i$ can be derived as follows:

$$\hat{\mathbf{P}}_i = \mathbf{H}_i \cdot \mathbf{P}_T = \prod_{k=1}^{i} \hat{\mathbf{H}}_k \cdot \mathbf{P}_T \tag{1}$$

where $\hat{\mathbf{H}}_k$ is the the estimated transformation from template image T to the resampled image $\mathcal{W}(I_k, \mathbf{H}_{k-1}^{-1})$ for $k > 1$. For $k = 1$, $\hat{\mathbf{H}}_1$ denotes the transformation from T to I_1. $\mathcal{W}(I_i, \mathbf{H})$ is a warped image for I_i with respect to $\mathbf{H}$.

Since a planar object's motions within the video sequences are continuous, it is unnecessary to predict the homography from the template T to I_i at every frame for a

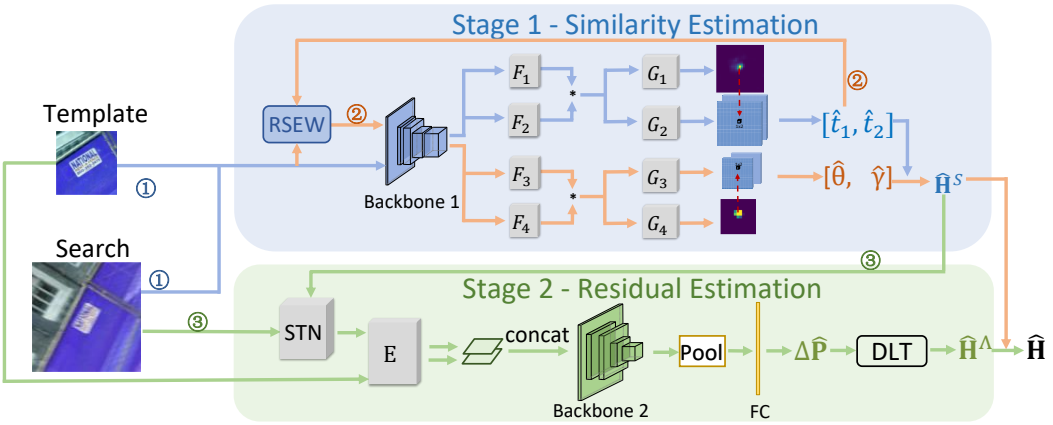

Figure 2: The tracking pipeline of HDN. The data flows in the networks obey the order number ($1 \rightarrow 2 \rightarrow 3$). In Similarity Estimation, F denotes the neck for feature cut and convolutional layer, and G denotes the convolutional layer for generating the output map. Residual Estimation Network is composed of the shared coarse feature extraction E, backbone, pooling layer (Pool), an FC (Fully Connected Layer) and Direct Linear Transformation (DLT) (Harltey and Zisserman 2003a), which converts the prediction to homography. STN (Jaderberg et al. 2015) is used to make the whole networks differentiable.

robust tracker. Therefore, a compositional matrix $\prod_{k=1}^{i} \hat{\mathbf{H}}_k$ is introduced to denote the cumulative transformation in the previous frames, and we only need to predict $\hat{\mathbf{H}}_i$ at each frame. Our deep approach adopts a forward method that only warps the input image I_i to avoid the object out of view when warping the template. Moreover, this reduces the extra computational cost for extracting the deep features of the template in the backbone at each frame and stabilize the template during tracking. In our proposed HDN approach, we only compute template's deep feature only once as the template image is kept constant.

Decomposition

The conventional approach (DeTone, Malisiewicz, and Rabinovich 2016) directly regresses the four corners' displacements of a rectangular planar object to predict the homography. Since the homography transformation does not directly relate to the corners' movements, it is determined by the planar object's pose in 3D space. Given a vector of eight transformation parameters $\mathbf{x} = [t_1, t_2, \gamma, \theta, k_1, k_2, \nu_1, \nu_2]^T$, the homography transformation $\mathbf{H}$ can be constructed efficiently as detailed in (Harltey and Zisserman 2003b). To further analyse the numerical properties of the proposed approach, let $\Delta\mathbf{p}(\mathbf{x}) = \mathbf{H}(\mathbf{x}) \cdot \mathbf{p} - \mathbf{p}$ be the difference between pixel coordinates before and after the homography transformation parameterized by $\mathbf{x}$.

Fig. 3 shows the condition number distribution of $\Delta\mathbf{p}(\mathbf{x})$ for a planar transformation. We sample $\mathbf{x}$ in the uniform distribution[1] and plot the condition number distribution in different decomposition settings. Fig. 3(a) demonstrates that the condition number of directly estimating eight parameters of $\mathbf{H}(\mathbf{x})$ is up to $5e^7$, which makes the system extremely unstable and unreliable for a planar object tracking task. We argue that this gives the insight of directly estimating homography parameters is a very hard problem for neural networks. On the other hand, Fig. 3(b) and Fig. 3(c) reduce the parameter space by the given translation $[t_1, t_2]$ and similarity $[t_1, t_2, \gamma, \theta]$ parameters, respectively. With these known parameters, the sensitivity drops in a remarkable magnitude, which improves the robustness of the system significantly. In addition, Fig. 3(d) shows the change of condition number as all parameters increased by a fixed ratio when large displacement occurs. Intuitively, directly estimating eight parameters in a large displacement setting becomes an ill-conditioned problem, however, simply decomposing the transformation contributes more robustness to the system.

Following these observations, we formulate the planar object tracking into a two-step estimation process and decompose the homography into two groups:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \gamma\, cos\theta & -\gamma\, sin\theta & t_1 \\ \gamma\, sin\theta & \gamma\, cos\theta & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 & 0 \\ 0 & 1/k_1 & 0 \\ \nu_1 & \nu_2 & 1 \end{bmatrix} = \mathbf{H}^S \cdot \mathbf{H}^\Lambda \quad (2)$$

where $\mathbf{H}^S$ is the similarity transformation, and $\mathbf{H}^\Lambda$ denotes the residual group. After decomposition, $\hat{\mathbf{H}}_k$ in Eq. (1) can be rewritten as $\hat{\mathbf{H}}_k = \hat{\mathbf{H}}_k^S \cdot \hat{\mathbf{H}}_k^\Lambda$, where $\hat{\mathbf{H}}_k^S$ is the estimated similarity transformation from T to $\mathcal{W}(I_k, \mathbf{H}_{k-1}^{-1})$. $\hat{\mathbf{H}}_k^\Lambda$ is the estimated residual transformation from T to $\mathcal{W}(I_k, (\hat{\mathbf{H}}_k^S)^{-1} \cdot \mathbf{H}_{k-1}^{-1})$. In this way, our framework is more stable for estimating by the neural networks.

Homography Decomposition Networks

To robustly track the planar object, we introduce novel two-stage end-to-end decomposition networks. Fig. 2 illustrates the pipeline of our proposed HDN which consists of Similarity Component and Residual Component. They estimate two transformations in order, which are connected by STN (Jaderberg et al. 2015) to make the whole network differentiable. The loss function for our proposed network can be derived as below:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_s(\hat{\mathbf{H}}^S) + \mathcal{L}_\Lambda(\hat{\mathbf{H}}) \quad (3)$$

[1] All the settings of $\mathbf{x}$ range is detailed in experimental setting

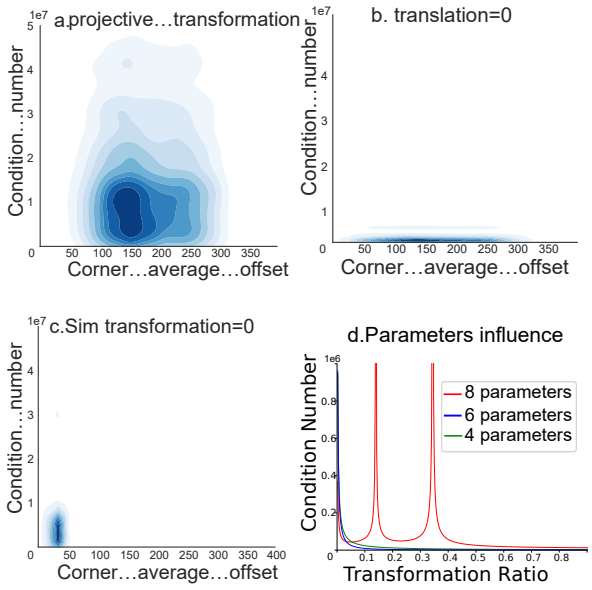

Figure 3: (a,b,c) represent the condition number of randomly sampled parameters permute with corner delta, and the value of the right bar represents the probability density of the sampled input point. (d) represents the condition number permute with transformation degree.

where $\mathcal{L}_s$ denotes the similarity component loss, $\mathcal{L}_\Lambda$ is the residual component loss and λ_1 is the weight parameter. Our training process only involves two frames due to the formulation of our approach. For simplicity, we remove all the subscript i of $\mathbf{H}_i$ and I_i in the following sections.

Similarity Component As depicted in Fig. 2, our similarity component gives reliable initial parameters and does not require to predict a bounding box for the sake of tracking the perspective change. To this end, a classification map and an offset map are generated in the two heads of HDN similar to (Chen et al. 2020) for translation estimation, except that we add more specific object labels adapted to the rotated objects according to the ground-truth θ. For the classification map labeling, we simply assign a target possibility for every position in the map and adopts a regression loss to achieve a close probability to the real value. A hamming window is used to construct the label map $\mathbf{M}_c$ as exhibited in Fig. 4.

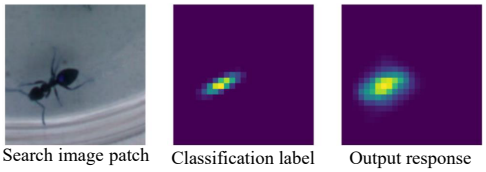

Figure 4: The classification map of similarity component.

The indices of the highest probability in classification map $\hat{\mathbf{M}}_c$ give the target's center coarsely, since a pixel in the classification map corresponds to eight pixels in the input image due to the total stride ζ of the network being eight. To achieve an accurate estimation, an offset map $\hat{\mathbf{M}}_o$ is thereby utilized to predict the difference between the grid coordinates of the offset map and the real target center location in the search image I. Note that, the grid coordinates are aligned to the size of image I, and the origin of the coordinates is the center of the object in search image. We choose the max response location $(u_m, v_m) = \arg_{u,v}\max(\hat{\mathbf{M}}_c(u,v))$ in classification map to index the offset map and its grid location $\zeta \cdot (u_m, v_m)$. Therefore, the translation is calculated as $\hat{\mathbf{t}} = (\hat{t}_1, \hat{t}_2) = \hat{\mathbf{M}}_o(u_m, v_m) + \zeta \cdot (u_m, v_m)$.

As in (Cohen and Welling 2015; Lenc and Vedaldi 2015), the conventional convolution does not have the equivariance to large scale and rotation changes. We thereby adopt the warped convolution (Henriques and Vedaldi 2017) to extend group convolution to image operator, which obtains an architecture equivariant to arbitrary two-parameter spatial transformations with a proper warping function.

Define a Lie group $\mathcal{G}$ and transformation g belonging to $\mathcal{G}$. To convert the transformation g into real space, an $\exp$ warp function is adopted to map scale and rotation operator in the real domain to the element in the Lie group (Sola, Deray, and Atchuthan 2018). The warped element owns the equivalence to scale and rotation with convolution. The warp function about two parameters scale γ' and rotation θ is defined as:

$$g_{\gamma',\theta}(\tau) = \begin{bmatrix} s^{\gamma'}\|\tau\|\cos(\arctan_2(\tau_2,\tau_1)+\theta) \\ s^{\gamma'}\|\tau\|\sin(\arctan_2(\tau_2,\tau_1)+\theta) \end{bmatrix} \quad (4)$$

where s controls the degree of scaling, $\tau = (\tau_1, \tau_2)$ is the pivot, and $\arctan_2$ denotes standard 4-quadrant inverse tangent function. Using Eq. (4), predicting the scale and rotation could be formulated as a translation estimation problem (Henriques and Vedaldi 2017; Li et al. 2019).

In this work, Rotation-Scale Equivariant Warping (RSEW) is proposed to implement the equivariant convolution about scale and rotation. Different from the previous approach PTN (Esteves et al. 2018), our method employs the correlation operator to find the region of interest, where the warped image is further used to estimate the scale and rotation changes by the equivariance with translation. Let image center be the origin, i.e. the object center of I, and left-top point be the resampled image origin. γ' is defined as $\frac{2\mu_1}{n}$, and θ is $\frac{4\pi\mu_2}{n}$. $\mu = (\mu_1, \mu_2)$ is set as the resampled image coordinates. s in Eq. (4) is $\frac{n}{4}$, where n is the edge length of squared image I. Moreover, we set the pivot $\tau = (1, 0)$. The converted grid sampling point (u, v) in I can be represented as follows:

$$\begin{cases} u = \hat{t}_1 + (\frac{n}{4})^{\frac{2\mu_1}{n}}\cos(\frac{4\pi\mu_2}{n}) \\ v = \hat{t}_2 + (\frac{n}{4})^{\frac{2\mu_1}{n}}\sin(\frac{4\pi\mu_2}{n}) \end{cases} \quad (5)$$

As illustrated in Fig. 2, estimating classification and offset maps for the scale and rotation is similar to the translation prediction. Once the scale and rotation are obtained, the estimated $\hat{\gamma}$ and $\hat{\theta}$ can be recovered as $(\hat{\gamma}, \hat{\theta}) = ((\frac{n}{4})^{\frac{2\hat{\mu}_1}{n}}, \frac{4\pi\hat{\mu}_2}{n})$,

where $\hat{\mu}$ is the estimation for scale and rotation in the warped image. Composing them and translation estimated before forms similarity parameters $\hat{\mathbf{x}}_S$.

Classification loss for translation is based on the weighted label map, which adopts the combined regression loss for the positive samples and negative samples as follows:

$$\begin{aligned}\mathcal{L}_{cls} =& \frac{1}{K}\sum_{i\in\Omega_-} -\log(1-\hat{\mathbf{M}}_c(i)) \\ &+ \frac{1}{Q}\sum_{j\in\Omega_+} |\hat{\mathbf{M}}_c(j) - \mathbf{M}_c(j)|\end{aligned} \tag{6}$$

where Ω_+ denotes the set of top K positive samples in $\hat{\mathbf{M}}_c$. The positive samples are selected when the probability is higher than the given threshold τ before choosing the top K samples. To deal with the imbalance between negative and positive samples, $\mathcal{L}_{cls}$ is designed to pick the top K negative samples in order to focus on the hard negatives, and Ω_- represents the chosen negative samples. Q and K are the total quantity of positive and negative samples, respectively.

For the offset loss $\mathcal{L}_{reg}$, we only calculate the loss of pixels when its positive class probability is higher than the threshold. We use $\mathcal{L}_{reg} = \frac{1}{U}\sum_i \mathcal{R}(\hat{\mathbf{M}}_o(i) - \mathbf{M}_o(i))$, where $\mathcal{R}$ is the robust loss function (i.e. smooth l_1) defined in (Girshick 2015), $\mathbf{M}_o$ is the label of offset map, and U denotes the total number of samples in the map. The rotation-scale label maps are set similar to SiamBAN (Chen et al. 2020). The classification loss $\mathcal{L}_{cls2}$ for the scale and rotation estimation uses the cross-entropy loss, and the regression employs the same smooth l_1 loss as the translation estimation. Thus, the loss of similarity component L_s can be derived as below:

$$\mathcal{L}_s = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \mathcal{L}_{cls2} + \mathcal{L}_{reg2} \tag{7}$$

Residual Component Residual transformation estimation is subsequently accomplished by regressing the corners offsets $\Delta\mathbf{P}$ between the tracked target quadrilateral $(\hat{\mathbf{H}}^S)^{-1}\mathbf{P_1}$ and $\mathbf{P}_T$ as introduced in Fig. 2, where $\mathbf{P}_1$ is the object corners coordinates in the search patch. The prediction $\Delta\hat{\mathbf{P}}$ is converted to the homography $\hat{\mathbf{H}}^\Lambda$ by DLT (Harltey and Zisserman 2003a). Inspired by (Nguyen et al. 2018; Zhang et al. 2020a), a fused semi-supervised network is adopted to estimate the homography between two images, which makes the whole network differentiable through STN (Jaderberg et al. 2015). The reason of using the semi-supervised setting is that many tracking datasets are not labeled with homography and the simple augmentation for the supervised setting cannot cover the appearance changes in real-world scenarios. The loss for residual homography estimation employs a triplet loss $\mathcal{L}^*_{\Lambda+} = \mathcal{L}_{Triplet}$ defined in (Schroff, Kalenichenko, and Philbin 2015), where the embedding matrices are the anchor embedding feature $E(I)$, the positive embedding $E(\mathcal{W}(T, \hat{\mathbf{H}}^S \cdot \hat{\mathbf{H}}^\Lambda))$, and the negative embedding $E(T)$. E is the coarse feature extractor as shown in Fig. 2. Concretely, $\mathcal{L}^*_{\Lambda+}$ can be written as follows:

$$\begin{aligned}\mathcal{L}^*_{\Lambda+} = \frac{1}{m^2}\max\{&\|E(I) - E(\mathcal{W}(T, \hat{\mathbf{H}}^S \cdot \hat{\mathbf{H}}^\Lambda))\|_2 \\ &- \|E(T) - E(I)\|_2 + \alpha, \mathbf{0}\}\end{aligned} \tag{8}$$

where m denotes the edge length of the embedding, and α is set to one by default. Pairwise distance with l_2-norm is adopted in calculating the distance of maps.

To adapt the synthetic training data and acquire higher estimation accuracy, we add an additional supervised loss with l_1-norm for the object corners' offsets as $\mathcal{L}_{\Lambda+} = \frac{1}{4}\sum_i^4 ||\Delta\mathbf{P}(i) - \Delta\hat{\mathbf{P}}(i)||_1$.

The negative samples of two different objects are added to further improve the robustness of our model, and we employ the strategy to set corners' translation to zero. We define the homography estimation of a negative sample loss as $\mathcal{L}_{\Lambda-} = \frac{1}{4}\sum_i^4 \mathcal{R}(\Delta\hat{\mathbf{P}}(i))$. Intuitively, this strategy guides the tracker staying on its previous status when it loses the target.

From the above all, the total loss $\mathcal{L}_\Lambda$ of residual component can be obtained as below:

$$\mathcal{L}_\Lambda = \lambda_2\mathcal{L}_{\Lambda-} + \lambda_3\mathcal{L}^*_{\Lambda+} + \lambda_4\mathcal{L}_{\Lambda+} \tag{9}$$

where λ_2, λ_3 and λ_4 are the weights to balance the loss. Our proposed method with robust similarity prediction and the constrained negative loss bridges the gap between homography estimation and planar tracking.

Training Details

Existing datasets lack the transformation parameters including rotation and scale, etc. Therefore, we augment the possible transformations according to Eq. (2). The algorithm randomly chooses $\mathbf{x}$ from a certain range to perform the transformation on COCO14 (Lin et al. 2014). The range is smaller than the whole domain because our proposed compositional method accumulates the transformation between inter-frames. To solve the unrealistic problem of synthetic datasets for residual component training, we sample the images of tracking dataset GOT10k (Huang, Zhao, and Huang 2019) with a small interval threshold as training data and adopt an unsupervised residual loss $\mathcal{L}^*_{\Lambda+}$. We further discuss the effect of the supervision method on the ablation section.

Experiment

Experiment Setup

We conducted experiments on a PC with an intel E5-2678-v3 processor (2.5GHz), 32GB RAM and Nvidia GTX 2080Ti GPU. Our proposed method is implemented in Pytorch. The size of input template T for our networks is 127×127, while search image I has the size of 255×255 to deal with the large homography changes. All the hyperparameters are set empirically, and we do not use any re-initialization and failure detection scheme. For the hyperparameters of HDN in training and testing, we set $\lambda_1 = 100$, $\lambda_2 = 1$, $\lambda_3 = 1$, $\lambda_4 = 0.25$, $K = 100$. $\gamma \in [1/1.38, 1.38]$, $\theta \in [-0.7, 0.7]$, $t \in [-32, 32]$, $k_1 \in [-0.1, 0.1]$, $k_2 \in [-0.015, 0.015]$, $\nu \in [-0.0015, 0.0015]$. The probability threshold τ in $\mathcal{L}_{cls}$ is set to 0.7.

State-of-the-art Comparison

We compare our proposed method with the state-of-the-art trackers on four tracking datasets. These methods can be divided into three categories, including keypoint-based

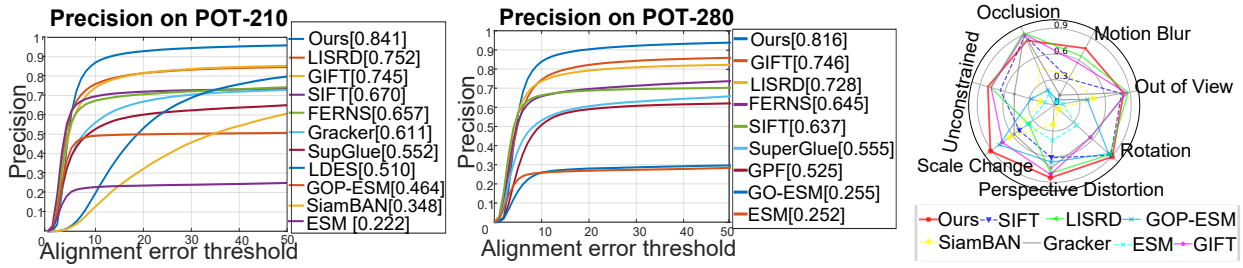

Figure 5: Comparisons on POT-210 and POT-280 with various challenging factors. The left subfigure and middle subfigure are the Precision on POT-210 (Liang, Wu, and Ling 2018) and POT-280 (Liang et al. 2021) with different thresholds, respectively. Legends show the average precision (avg Prec). The third radar figure compares the trackers' avg Prec on 7 challenging factors.

tracker	avg Prec/HSR	Prec (e≤ 5)	Prec (e≤ 10)	Prec (e≤ 20)	avg CP	avg SR
SIFT	0.670/0.570	**0.622**	0.695	0.719	0.698	0.701
SURF	0.657/0.536	0.543	0.663	0.711	0.689	0.705
LISRD	0.752/0.619	0.617	0.759	0.815	0.788	0.805
GIFT	0.745/0.580	0.551	0.741	0.815	0.785	0.801
SuperGlue	0.552/0.509	0.389	0.526	0.601	0.592	0.659
Gracker	0.611/0.527	0.392	0.560	0.668	0.684	0.716
ESM	0.222/0.209	0.204	0.227	0.235	0.242	0.261
FERNS	0.657/0.601	0.565	0.673	0.706	0.686	0.724
SCV	0.250/0.233	0.228	0.257	0.265	0.274	0.287
GOP-ESM	0.464/0.442	0.430	0.492	0.499	0.484	0.497
SiamBAN	0.348/0.335	0.022	0.127	0.321	0.650	0.693
LDES	0.510/0.518	0.028	0.196	0.531	0.617	0.737
Ocean	0.255/0.261	0.011	0.089	0.241	0.464	0.521
TransT	0.464/0.381	0.090	0.276	0.486	0.764	0.774
Ours	**0.841/0.710**	0.611	**0.870**	**0.928**	**0.886**	**0.904**

Table 1: Tracking results on POT-210.

trackers (SIFT (LoweDavid 2004), SURF (Bay, Tuytelaars, and Gool 2006), FERNS (Özuysal, Fua, and Lepetit 2007), Gracker (Wang and Ling 2018), SuperGlue (Sarlin et al. 2020), LISRD (Pautrat et al. 2020), GIFT (Liu et al. 2019)), region-based method (ESM (Benhimane and Malis 2004), GOP-ESM (Chen et al. 2019)) and generic visual tracking method (GPF (Kwon et al. 2014), Ocean (Zhang et al. 2020b), TransT (Chen et al. 2021) and SiamBAN (Chen et al. 2020)). We evaluate all the trackers on the challenging POT-210 dataset and choose top-ranked trackers in three categories for the POT-280, UCSB, and POIC. All the results are either based on the publications or from the benchmark (Liang, Wu, and Ling 2018; Liang et al. 2021).

POT. POT-210 (Liang, Wu, and Ling 2018) contains 210 videos of 30 planar objects sampled in the natural environment. It includes seven challenging scenarios: scale, rotation, occlusion, etc. POT-280 (Liang et al. 2021) adds another 70 sequences to POT-210, and adopt two evaluation metrics: Precision (Prec) and Homography Success Rate (HSR). Precision is defined as the percentage of frames whose alignment error is smaller than the given threshold. The alignment error is computed by the average of the four points L_2 distance between the predicted polygon and the ground truth label. The Homography Success Rate (HSR) describes the percentage of frames whose homography discrepancy score is less than a threshold. Fig. 5 reports the result, which indicates that our HDN outperforms the other trackers in most of the metrics on both benchmarks. We give more results in the appendix. Using the relative improvement ratio, it achieves 12% and 9.4% improvement compared to the second-best methods in avg Prec on POT-210 and POT-280, respectively. Moreover, our presented HDN method performs better than the best non-keypoint-based tracker LDES (Li et al. 2019) by 64.7% on POT-210. Our decomposition networks can robustly estimate the homography in all challenging scenarios. For some hard cases (see the right subfigure in Fig. 5), our method performs better than other trackers except for the occlusion and out-of-view scenes due to the correlation mechanism. The reason is that the center of its response map is estimated from the non-occluded region if an object is heavily occluded. LISRD (Pautrat et al. 2020) and GIFT (Liu et al. 2019) have difficulties in dealing with scale variations or rotation changes. Shrinking object reduces the total number of keypoints, which makes the result of homography estimation unreliable. In terms of perspective change, our method performs better than the keypoint-based methods. This is because they estimate from the template to the current frame to avoid failure while incurring the jitters at the same time.

To evaluate the quality of trackers from other aspects and reveal the underlying reasons, we include two extra metrics. Centroid Precision (CP) is the precision of the object's polygon centroid, and Success Rate (SR) is defined as the successfully tracked ratio with the overlap (Intersection over Union) greater than the given ratio. Table 1 lists more trackers and detailed results on POT-210. HDN exhibits a great improvement on almost all the metrics, where its average Prec and HSR are much higher than the second-best method. Average Centroid Precision manifests our higher accuracy of translation estimation and average SR exhibits the better overlap score. When the error threshold is small, our method is not as good as other conventional trackers. Unlike other trackers, we only apply HDN once, ESM-based method iteratively optimizes the estimation with more accurate results in easy cases. Besides, the input image size of HDN is smaller than most of the conventional methods.

UCSB, POIC. We further conduct experiments on an-

	Trackers	SIFT	Gracker	TransT	GOP-ESM	Ours
	avg Prec	0.527	0.819	0.367	0.873	**0.874**
P	Prec(e$\leq$ 5)	0.400	0.671	0.047	**0.868**	0.749
O	Prec(e$\leq$ 10)	0.520	0.829	0.185	0.893	**0.894**
I	Prec(e$\leq$ 20)	0.573	0.878	0.366	0.901	**0.948**
C	avg CP	0.547	0.870	0.722	0.889	**0.916**
	avg SR	0.570	0.871	0.709	0.882	**0.923**
	avg Prec	\	0.837	0.422	0.528	**0.871**
U	Prec(e$\leq$ 5)	\	0.648	0.000	0.487	**0.660**
C	Prec(e$\leq$ 10)	\	0.831	0.001	0.519	**0.916**
S	Prec(e$\leq$ 20)	\	0.903	0.141	0.552	**0.964**
B	avg CP	\	0.885	0.908	0.575	**0.912**
	avg SR	\	0.859	0.565	0.579	**0.887**

Table 2: Results on UCSB and POIC.

other two challenging datasets UCSB (Gauglitz, Höllerer, and Turk 2011) and POIC (Chen et al. 2019). Table 2 reports the results compared to other trackers. Note that the keypoint-based methods such as LISRD and GIFT are not designed for planar tracking. There are no published results on UCSB and POIC. Our approach performs the best on all the metrics on both datasets, except when the error is smaller than 5 pixels in POIC. The reason behind this is the same as on POT. The performance of SIFT is inferior to the region-based methods, which may due to the low texture quality, motion blurs, and illumination changes.

Tracking Robustness To further evaluate the robustness of the tracker, we plot the trajectory robustness which is the ratio of trajectories with different lengths. It is defined as the length having no failure within the threshold IoU > 0.2. We set the trajectory length threshold to 10 for comparison in the legend of Fig. 6. All the keypoint-based methods generate large trajectory fragments, which means they are easy to lose their target. Although SIFT achieves a high average Prec in POT, it has the lowest robustness with the highest fragments ratio 75.5% in the case of trajectories length less than 10. This cannot be ignored because the frequent lost of objects brings the unsatisfied experiences, especially in augmented reality applications. HDN and LDES have fewer short trajectories and more trajectories cover the whole sequence than LISRD and GOP-ESM. It reveals that predicting translation first is crucial for improving the robustness of planar tracking.

Ablation Study

We study the impact of individual components in HDN, and conduct the ablation on POT-210, as reported in Table 3. To evaluate the contribution of different stages, we separate stages in HDN, and No.(1,2,8) give their performance. With only similarity component (Sim), the average Prec of HDN is higher than SIFT. On the other hand, only using the residual component (Res) leads to a rapid drop of the average Prec, which is only 31.9%. It coincides with the observation in Fig. 3 and shows the effectiveness of our proposed homography decomposition networks. HDN is efficient in practice, which runs at 10.6 fps. To be unsupervised or not, that is a question in learning. However, we find they are

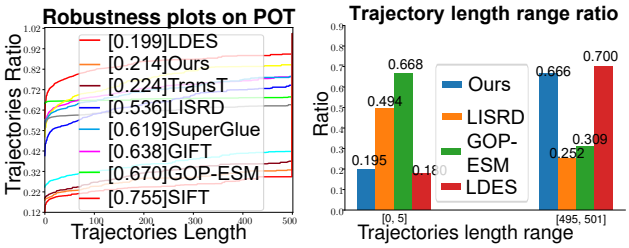

Figure 6: Tracking robustness evaluation on POT. The right subfigure is trajectories range ratio with length range [0,5] and [495, 501].

No.	Comp.	Sup.	$\mathcal{L}_{cls}$ &$\mathcal{L}_{loc}$	Rot-label	Neg-loss	Prec	Speed (fps)
1	Similarity	both	✓	✓	✓	0.689	13.7
2	Residual	both	✓	✓	✓	0.319	21.2
3	Sim+Res	Sup	✓	✓	✓	0.837	\
4	Sim+Res	Unsup	✓	✓	✓	0.417	\
5	Sim+Res	both		✓	✓	0.732	\
6	Sim+Res	both	✓		✓	0.836	\
7	Sim+Res	both	✓	✓		0.733	\
8	Sim+Res	both	✓	✓	✓	0.841	10.6

Table 3: Ablation on different components, Residual component supervision method, other designs and speed testing.

not conflicted. No.(3,4,8) show that using both the supervised and unsupervised loss in residual component obtains the higher average Prec than the single loss.

We further discuss the effect of other aspects. A rotated classification map label (No.6) brings a little improvement. This is because the rotated label is useful for the occluded object, where the offset estimation is inaccurate. Our proposed $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ is crucial in our HDN as a result of the relief of positive-negative samples imbalance and hard negative sampling problem. We combine them in Table 3, as they are related. The average Prec descends to 73.2% without them. Negative loss is crucial in the residual component, and the average Prec drops to 73.2% without it. Large appearance changes typically occur during tracking, and a compositional method may fail due to the large estimation error in the single frame.

Conclusion

In this paper, we have proposed novel homography decomposition networks that drastically reduce and stabilize the condition number by decomposing the homography transformation into two groups. Specifically, a similarity transformation estimator was designed to predict the first group robustly by a deep convolution equivariant network. By taking advantage of the scale and rotation estimation with high confidence, a residual transformation was estimated by a simple regression model. Extensive experiments show that our proposed approach outperforms the state-of-the-art planar tracking methods at a large margin on four datasets.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grants (61831015 and 62102152) and sponsored by CAAI-Huawei MindSpore Open Fund.

References

Bay, H.; Tuytelaars, T.; and Gool, L. 2006. SURF: Speeded Up Robust Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 404–417.

Benhimane, S.; and Malis, E. 2004. Real-time image-based tracking of planes using efficient second-order minimization. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1: 943–948 vol.1.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 850–865.

Black, M. J.; and Jepson, A. 2004. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, 26: 63–84.

Chen, L.; Ling, H.; Shen, Y.; Zhou, F.; Wang, P.; Tian, X.; and wu Chen, Y. 2019. Robust visual tracking for planar objects using gradient orientation pyramid. *Journal of Electronic Imaging*, 28: 013007 – 013007.

Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; and Lu, H. 2021. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8126–8135.

Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; and Ji, R. 2020. Siamese Box Adaptive Network for Visual Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6668–6677.

Cohen, S. T.; and Welling, M. 2015. Transformation Properties of Learned Visual Representations. *International Conference on Learning Representations (ICLR)*.

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2016. Deep Image Homography Estimation. arXiv:1606.03798.

Esteves, C.; Allen-Blanchette, C.; Zhou, X.; and Daniilidis, K. 2018. Polar Transformer Networks. arXiv:1709.01889.

Gauglitz, S.; Höllerer, T.; and Turk, M. 2011. Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking. *International Journal of Computer Vision*, 94: 335–360.

Girshick, R. B. 2015. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 1440–1448.

Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; and Chen, S. 2020. SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6269–6277.

Gupta, D. K.; Arya, D.; and Gavves, E. 2021. Rotation Equivariant Siamese Networks for Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12362–12371.

Harltey, A.; and Zisserman, A. 2003a. *Multiple view geometry in computer vision*, chapter 4, 88. Cambridge University Press, 2 edition.

Harltey, A.; and Zisserman, A. 2003b. *Multiple view geometry in computer vision*, chapter 2, 42. Cambridge University Press, 2 edition.

Henriques, J. F.; and Vedaldi, A. 2017. Warped Convolutions: Efficient Invariance to Spatial Transformations. In *Conference on International Conference on Machine Learning (ICML)*, 1461–1469.

Huang, L.; Zhao, X.; and Huang, K. 2019. GOT-10k: A large high-diversity benchmark for Ggeneric object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and Kavukcuoglu, K. 2015. Spatial Transformer Networks. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 6992–7003.

Kwon, J.; Lee, H. S.; Park, F.; and Lee, K. M. 2014. A Geometric Particle Filter for Template-Based Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36: 625–643.

Lenc, K.; and Vedaldi, A. 2015. Understanding image representations by measuring their equivariance and equivalence. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 991–999.

Li, Y.; Zhu, J.; Hoi, S. C.; Song, W.; Wang, Z.; and Liu, H. 2019. Robust Estimation of Similarity Transformation for Visual Object Tracking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33: 8666–8673.

Liang, P.; Ji, H.; Wu, Y.; Chai, Y.; Wang, L.; Liao, C.; and Ling, H. 2021. Planar object tracking benchmark in the wild. *Neurocomputing*, 454: 254–267.

Liang, P.; Wu, Y.; and Ling, H. 2018. Planar Object Tracking in the Wild: A Benchmark. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 651–658.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 740–755.

Liu, Y.; Shen, Z.; Lin, Z.; Peng, S.; Bao, H.; and Zhou, X. 2019. GIFT: Learning Transformation-Invariant Dense Visual Descriptors via Group CNNs. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*.

LoweDavid, G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*.

Nguyen, T.; Chen, S. W.; Shivakumar, S. S.; Taylor, C. J.; and Kumar, V. 2018. Unsupervised Deep Homography: A Fast and Robust Homography Estimation Model. *IEEE Robotics and Automation Letters*, 3: 2346–2353.

Özuysal, M.; Fua, P.; and Lepetit, V. 2007. Fast Keypoint Recognition in Ten Lines of Code. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.

Pautrat, R.; Larsson, V.; Oswald, M. R.; and Pollefeys, M. 2020. Online invariance selection for local feature descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 707–724.

Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4938–4947.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823.

Sola, J.; Deray, J.; and Atchuthan, D. 2018. A micro Lie theory for state estimation in robotics. arXiv:1812.01537.

Tsai, C.-Y.; and Feng, Y.-C. 2019. Planar Tracking based on Deep Learning. *2019 8th International Conference on Innovation, Communication and Engineering (ICICE)*, 21–24.

Wang, T.; and Ling, H. 2018. Gracker: A Graph-Based Planar Object Tracker. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40: 1494–1501.

Zhang, J.; Wang, C.; Liu, S.; Jia, L.; Wang, J.; and Zhou, J. 2020a. Content-Aware Unsupervised Deep Homography Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 653–669.

Zhang, Z.; Peng, H.; Fu, J.; Li, B.; and Hu, W. 2020b. Ocean: Object-aware anchor-free tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 771–787.