HVC-Net: Unifying Homography, Visibility, and Confidence Learning for Planar Object Tracking

Haoxian Zhang*[1][0000−0001−7078−868X], Yonggen Ling*†[2][0000−0001−8294−6286]

[1] Tencent AI Lab, China
leohxzhang@tencent.com
[2] Tencent Robotics X, China
rolandling@tencent.com

Abstract. Robust and accurate planar tracking over a whole video sequence is vitally important for many vision applications. The key to planar object tracking is to find object correspondences, modeled by homography, between the reference image and the tracked image. Existing methods tend to obtain wrong correspondences with changing appearance variations, camera-object relative motions and occlusions. To alleviate this problem, we present a unified convolutional neural network (CNN) model that jointly considers homography, visibility, and confidence. First, we introduce correlation blocks that explicitly account for the local appearance changes and camera-object relative motions as the base of our model. Second, we jointly learn the homography and visibility that links camera-object relative motions with occlusions. Third, we propose a confidence module that actively monitors the estimation quality from the pixel correlation distributions obtained in correlation blocks. All these modules are plugged into a Lucas-Kanade (LK) tracking pipeline to obtain both accurate and robust planar object tracking. Our approach outperforms the state-of-the-art methods on public POT and TMT datasets. Its superior performance is also verified on a real-world application, synthesizing high-quality in-video advertisements.

Keywords: Planar Object Tracking, Homography, Visibility, Confidence

1 Introduction

Planar object tracking is a classic computer vision task with a wide range of applications. Given the initial corners of a planar object in the reference frame, the primary goal of planar tracking is to estimate the movements of these corners, modeled by a geometric transformation called a homography, in consecutive frames. Though lots of advances have been made in past decades, obtaining accurate and robust results remains challenging. These difficulties are mainly caused by three factors: appearance variation, camera-object relative motion and occlusion. The appearance variation is a camera-related issue. It is usually known as image blur, sensor noise, non-linear response of brightness. The camera-object

* Equal Contribution listed alphabetically. † Corresponding author.

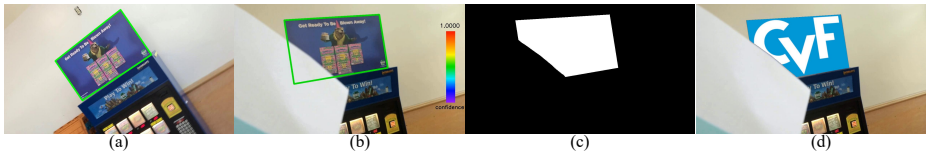

Fig. 1. One of our synthetic frames in the lottery sequence of [23]. (a) A planar object to be tracked in the reference image, denoted by a green quadrilateral. (b) The estimated homography with very high tracking confidence in one video frame. (c) Corresponding visibility mask of the tracked object. (d) The synthetic frame after placing the CVF logo using results from (b) and (c). More results are shown in Fig. 7 and the supplemental.

relative motion leads to geometry transformations of an object on the image. Typical effects on the image plane are scale changes, rotations, translations, and perspective distortions. Occlusion is referred as the fact that the tracked object is occluded by another object. The situation becomes worse if the 'another object' looks very similar to the tracked object. These factors pose strong challenges for traditional keypoint-based methods that estimate the homography using hand-crafted features [35, 12, 6], since the extracted features are prone to be different under the influence of these factors. Learned features like D2-Net [14], LF-Net [45], and R2D2 [30] are proposed to decrease this influence. Direct methods [4, 7], usually with the LK pipeline [4], estimate the homography iteratively. [4, 7] assume the intensity consistency and compute the homography increment for each iteration. [9, 27, 24, 46, 47] extend direct methods with the learned 'feature consistency' assumption for increasing the robustness. We argue that efforts are still needed on better feature representation. Moreover, these methods have not discussed occlusions that are widely existed in real-world video sequences. The last to mention is the CNN-based method [11] that directly regresses the homography in one step with CNN.It is not robust to these three factors, neither.

In this work, we propose a novel CNN model for handling mentioned difficulties. The base of our model is correlation blocks (Sect. 3.3). It firstly extracts features in the intensity domain for handling appearance variations. Cost volumes, representing distributions of pixel correlations, are then constructed in the pixel displacement domain to account for the camera-object relative motion. We find that estimating the homography with these two cascaded steps is much better than methods with one step [11, 27, 9, 24]. Moreover, in contrast to methods that learn homography alone [11, 27, 9, 24], we learn it jointly with another task called visibility, which is defined as a binary mask that indicates which part of the reference image is visible on the tracked image (Fig. 2). A reference image pixel is regarded as visible if and only if it satisfies the homography constraint of the tracked planar object (geometry-induced) and it is not occluded by other objects on the tracked image (disocclusion-induced). Joint learning homography and visibility not only improves the correlation block representations, but also links camera-object relative motions with occlusions (Sect. 3.5). Lastly, as estimations with the LK pipeline are sensitive to initializations, we further improve

Fig. 2. (from left to right) Reference image, current image, motion-induced visibility, disocclusion-induced visibility, combined visibility used in our model.

the estimation robustness by monitoring the tracking quality and rebooting estimations. This is done by introducing a confidence module that evaluates the planar tracking quality from pixel correlation distributions obtained in correlation blocks (Sec. 3.7). By equipping all these presented modules with a LK pipeline, our model obtains both accurate and robust homography estimations. We achieve significantly higher homography precision than state-of-the-art homography estimation methods (Sect. 4). Besides, as a by-product, our model provides visibility masks that other works have not mentioned. With these masks, we are able to easily place planar advertisements in videos (Fig. 1).

2 Related Work

2.1 Homography Estimation

Existing planar tracking methods for estimating the underlying homography can be roughly classified into three categories: keypoint-based methods [35, 12, 6, 28, 3, 13], direct methods [4, 7, 10, 31, 5, 26], and CNN-based methods [11, 27, 9, 24]. Keypoint-based methods firstly detect and describe keypoints (using ORB [35], SIFT [12], SURF [6] and etc.) both in the reference planar region and subsequent consecutive frames. These keypoints are then matched by minimizing the distances in the descriptor space. Homography, the planar surface in the projection space is related, is then calculated with the obtained matches. To remove potential outlier matches, RANSAC [16] is usually performed. Different from keypoint-based methods, direct methods [4, 7] assume that the planar template does not move fast in consecutive images. The homography is directly optimized by minimizing the photometric error between the planar template and its projection in the incoming video frames. Recently, CNN-based methods have been proposed. Homography is regressed from input images in one forward step [11, 27, 46, 47]. [9, 24, 20, 48] adopt the Lucas-Kanade framework [4] and compute homography with multiple iterations.

2.2 Object Segmentation

The visibility of planar object tracking is less discussed in the past. The closest work is segmentation. There are three main approaches for object segmentation according to the level of supervision required. Supervised methods require iterative human interactions for adding segmentation prior as well as refining segmentation outputs [2, 15]. They obtain high-quality segmentations at the cost of

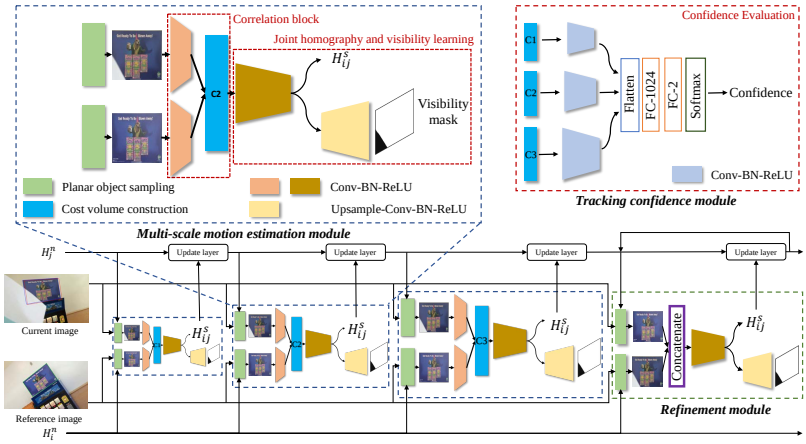

Fig. 3. The framework of our model. It follows the LK scheme. There are three modules: the multi-scale motion estimation module, the refinement module, and the tracking confidence module. The base of this model is correlation blocks that extract features in the intensity domain for handling appearance variations and construct cost volumes in the pixel displacement domain for handling motion-related issues in a cascaded way (Sect. 3.3). Pyramid blocks are build (Sect. 3.4), where homography and visibility are jointly learned (Sect. 3.5). The refinement module for further improvements is optional (Sect. 3.6). Tracking estimation confidence is also evaluated (Sect. 3.7).

extensive expert efforts. To relax this mass manual supervision, semi-supervised methods propagate sparse human labeling in the reference frame to the remaining frames, and then formulate the segmentation problem as an optimization problem with energy defined over graphs [1, 29, 42]. The last to mention is the unsupervised methods that do not require any manual annotation or utilize prior information on the segmented objects. Early unsupervised methods focus on over-segmentation [17] or motion segmentation [8]. They are extended to foreground-background separation in recent years [44, 41].

2.3 Patch Similarity

The most related work to confidence prediction is to compute the similarity between two patches [36, 18, 37]. The confidence score is learned by training the network with reflective loss in [37]. The similarity is trained via a classification pipeline in [36]. Patched representation as well as robust feature comparison is jointly learned in [18].

3 Our Approach

3.1 The LK-based CNN Framework

Our model framework is shown in Fig. 3. We follows the LK scheme [4] to compute homography, denoted as $\mathbf{H}_{ij} \in \mathbb{R}^{3\times3}$. For each 3D object point o_k, its

projection on image frame i and j is denoted as $\mathbf{p}_i^k$ and $\mathbf{p}_j^k$ respectively. According to the derivation from [19], we have $\mathbf{p}_i^k = \mathbf{H}_{ij}\mathbf{p}_j^k$. Supposing we have an initial homography $\mathbf{H}_{ij}$, the LK scheme consists of two iterated steps:
1) solving for homography increment $\delta\mathbf{H}_{ij}$,
2) updating homography $\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij} * \delta\mathbf{H}_{ij}$.
For the first step, the classic LK method [4] assume that intensities are consistent across images. We improve this step with three aspects. Firstly, as the intensity consistency assumption is prone to be broken in real-world cases with appearance variations and occlusions, we extend it with the 'feature consistency' assumption and improve the effectiveness of feature representation (Sect. 3.3). Secondly, homography increments are computed with difference scales (Sect. 3.4). Thirdly, based on the 'feature consistency' assumption, we compute homography increments with joint homography and visibility learning (Sect. 3.5). The improved first LK step is implemented as the multi-scale motion estimation module in our model. We also have an optional step without correlation block, i.e. the refinement module (Sect. 3.6). As computed homography increments are sensitive to homography initializations, we present a tracking confidence module to evaluate the estimation quality and re-initializes the homography computations (Sect. 3.7). We follow the same second step as the LK pipeline, where we update homography through update layers. Lastly, we notice that the concerned planar object tracking problem is to solve for homography between object projections on two images while existing LK-based methods consider homography between two images. We thus propose a sampling trick to turn the concerned problem into a classic LK-based homography problem that is more suitable for CNN models (Sect. 3.2).

3.2 Homography Surrogate & Sampling

The projection shape of a 3D plane on video images deforms as the camera moves relatively to the tracked object. Processing the full-resolution video images with CNNs will waste a lot of memory as well as computations on useless image regions outside the projection shape. What's worse, information on outside regions will distort the estimations and make CNN predictions more challenging. To this end, we propose a planar object sampling layer for CNNs for handling planar objects in arbitrarily deformed shapes or sizes. As shown in Fig. 4, the key idea is NOT to predict the original homography in the original image space. Instead, we predict a surrogate homography in the normalized space. We sample the planar object in the reference image into a $W \times H$ template: $\mathbf{p}_i^n = \mathbf{H}_i^n\mathbf{p}_i$, where $\mathbf{H}_i^n$ can be easily computed using SVD [19] once the reference planar object with four-corner representation is given. We denote the homography used to sample the planar object in the current image into a $W \times H$ template as $\mathbf{H}_j^n$, and the homography between two normalized images i and j is $\mathbf{H}_{ij}^s$. We have:

$$\mathbf{H}_j^n = (\mathbf{H}_{ij}^s)^{-1}\mathbf{H}_i^n\mathbf{H}_{ij} = (\mathbf{H}_{ij}^s)^{-1}\mathbf{H}_{ij}^* \tag{1}$$

where $\mathbf{H}_{ij}^* = \mathbf{H}_i^n\mathbf{H}_{ij}$. We define $\mathbf{H}_j^n$ as a surrogate for $\mathbf{H}_{ij}$, and $\mathbf{H}_{ij}^s$ as a surrogate for $\delta\mathbf{H}_{ij}$. $\mathbf{H}_{ij}^s$ will be an identity matrix if and only if $\mathbf{H}_j^n$ is equal to ground

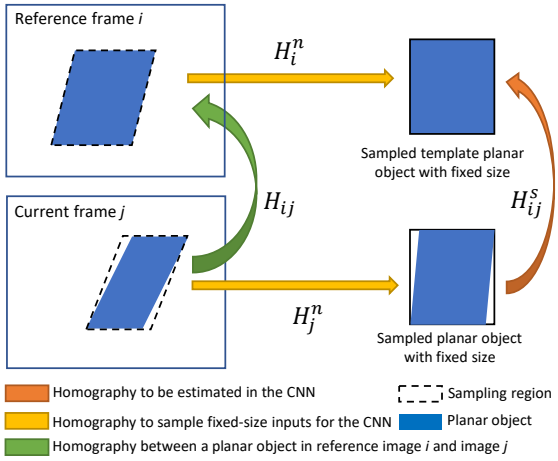

Fig. 4. The planar object sampling. We sample the planar object in the reference frame and in the current frame to fixed-size images with $\mathbf{H}_i^n$ and $\mathbf{H}_j^n$, and use our mode to predict the increment $\mathbf{H}_{ij}^s$. $\mathbf{H}_{ij}^s$ will be the identity matrix if and only if the sampled planar objects on both sampled images are aligned perfectly. $\mathbf{H}_j^n$ and $\mathbf{H}_{ij}^s$ are used as surrogates for $\mathbf{H}_{ij}$ and $\delta\mathbf{H}_{ij}$ respectively.

truth $\mathbf{H}_{ij}^*$. If the final $\mathbf{H}_j^n$ is obtained, $\mathbf{H}_{ij}$ is computed as $\mathbf{H}_{ij} = (\mathbf{H}_i^n)^{-1}\mathbf{H}_j^n$. By using surrogates, we maintain a fixed-size input to CNNs.

3.3 Correlation Block

Different from previous works [11, 24] that regress homography on images, we decompose the homography regression into two cascaded steps:
1) The first step is to extract features representing image local appearances. These features are designed to be robust for image blur, illumination variations, occlusions, scale changes, perspective distortions, etc, through data argumentation covering various image conditions. Since the template size is small, we use the U-Net structure [32] for simplicity. Other feature extraction structures, such as ResNet, EfficientNet and MultiResUNet, can also be used.
2) The second step is to construct cost volumes with extracted features, whose elements are pixel correlations between sampled images. These pixel correlations are designed to encode the relative geometry transformation between objects and cameras. Each element in this cost volume is computed as the correlation [40] between a pixel x_i in reference feature map $\mathbf{f}_r$ and a pixel x_j in the tracked feature map $\mathbf{f}_t$: $c(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{f}_r(\mathbf{x}_i)^T\mathbf{f}_t(\mathbf{x}_j)$, where T is the transpose operator. Given a maximum displacement d_m, for each location $\mathbf{x}_i$ we compute correlations $c(\mathbf{x}_i, \mathbf{x}_j)$ for $\mathbf{x}_j$ s.t. $|\mathbf{x}_j - \mathbf{x}_i| <= d_m$. Correlations at each location $\mathbf{x}_i$ are reorganized in the channel dimension. Thus, the size of the 3D cost volume is $H \times W \times (2d_m + 1)^2$. d_m is set to be 4 at each pyramid here by balancing the complexity and movement range.

3.4 Pyramids

Inspired by the classic pyramid methods in image processing, we build correlation blocks in different scales. We sample objects with different template resolutions (1/16x, 1/4x, 1x). Homography increments are computed sequentially from the smallest resolution to the highest resolution.

3.5 Joint Learning of Homography and Visibility

Homography is obtained by information that is visible on both reference and tracked images. Hence, we learn homography jointly with visibility, in order to extract a more reliable feature representation. This leads to three loss functions during training: L_d, L_m, and L_v. For benefit of CNNs, we adopt representation in [11], where homography is represented by four corner displacements $\{d_1, d_2, d_3, d_4\}$. L_d is a homograph loss. It is defined as the $l1$ norm between the ground truth 4-point displacement d_k^* and the predicted 4-point displacement d_k at each scale level:

$$L_d = \frac{1}{4}\sum_{k=1}^{4} \|d_k^* - d_k\|_1 \tag{2}$$

L_m is a visibility loss. Pixel visibility prediction of the sampled tracked image is regarded as a 2-class classification problem. We denote the ground truth label and the predicted label for a pixel's visibility as m_k^* and m_k. Cross-entropy is adopted for the visibility loss L_m at each scale level:

$$L_m = -\frac{1}{N^k}\sum_{k=1}^{N^k}(m_k^* \log(m_k) + (1 - m_k^*)\log(1 - m_k)) \tag{3}$$

where N^k is the total number of pixels at each scale level. To further improve the feature representations used to construct cost volumes, we add a visible alignment loss L_v that minimizes the visible feature distance between extracted reference feature map $\mathbf{f}_r$ and tracked feature map $\mathbf{f}_t$. It is defined as followed,

$$L_v = \frac{1}{N^k}\sum_{\mathbf{x}_k} m_k^* \left\|\mathbf{f}_t'(\mathbf{x}_k) - \mathbf{f}_t(\mathbf{x}_k)\right\|_1 \tag{4}$$

where $\mathbf{x}_k$ is the pixel location on the sampled tracked image, $\mathbf{f}_t' = Warp(\mathbf{f}_r, \mathbf{H}_{tr})$ is a wrapped feature map from $\mathbf{f}_r$ to $\mathbf{f}_t$ using the homography $\mathbf{H}_{tr}$. The total loss is the combination of these three losses:

$$L_{all} = \lambda_d L_d + \lambda_m L_m + \lambda_v L_v \tag{5}$$

where λ_d, λ_m and λ_v are balancing parameters. In our experiments, they are all empirically set to be 1.0.

With the visibility loss, we explicitly connect homography with occlusion. This is in contrast to competing methods [9, 27, 24, 46, 47] that handle occlusions implicitly with the learned feature capability. Moreover, with the visible alignment loss, we ale able to connect homography, visibility and features in the correlation block.

Notice that, the supervised visibility mask varies in each scale level. It is generated at each training iteration.

3.6 Homography and Visibility Refinement

This module is similar to that of Sect. 3.5 expect that the correlation block is removed and the visible alignment loss is ignored. It is designed to capture tiny modifications to the homography and visibility. The VGG structure [39] is used for simplicity. Three iterations are usually conducted for convergence. Note that, this module is optional.

3.7 Estimation Confidence Evaluation

This section discusses the homography initialization in the LK pipeline (Sect. 3.1). The initial homography of the first scale level is equal to the homography obtained at the previous video frame $j-1$. For the following scale levels, their initializations are equal to homography obtained at previous scale levels. For the refinement module, its first homography initial value is equal to the homography output from the multi-scale motion estimation module. In the following refinement step, its initial homography is equal to the homography in last iteration.

With this homography initialization mechanism, we see the significance of the homography obtained at the previous video frame $j-1$, as it is the base of estimation in the current video frame j. However, though we have tried our best to improve the homography estimation robustness and accuracy, our trained model inevitably fails under extreme conditions, such as large appearance variations, rapid camera-object relative motions, and severe occlusions. That is, the homography obtained at the previous video frame $j-1$ may be unreliable. To check this, we add a tracking confidence module to evaluate the estimation confidence. This confidence is regarded as a regression whose output ranges between 0 and 1. 0 indicates the estimation is unreliable while 1 indicates it is reliable. In contrast to previous works [36, 18, 37] that regress confidences from images, we regress them from cost volumes of correlation blocks. These multi-scale cost volumes, representing distributions of pixel correlations, encode the 'uncertainty' of the estimation. For an object pixel in the reference image, its corresponding pixel on the tracked image is ambiguous if the pixel correlation distribution is flat, or obvious if the pixel correlation distribution is concentrated on one specific location. We train this tracking confidence module after the multi-scale motion estimation module and the optional refinement module is trained using an independent dataset.

We consider the estimation as unreliable if the homography loss L_d between the ground truth and predicted homography is larger than 5 while reliable otherwise. We denote the ground truth label and the predicted label as p^* and p. Cross-entropy loss is used for confidence loss:

$$L_c = -(p^* \log(p) + (1-p^*) \log(1-p)) \tag{6}$$

In implementations, each cost volume of each pyramid layer is convoluted to a $\frac{H}{8} \times \frac{W}{8} \times 15$ feature map by several convolutional layers respectively. These feature maps are then followed by two fully connected (FC) layers, whose dropout ratio is set to 0.5, with 1024 and 2 channels. The final layer is a soft-max layer that output the confidence. 3×3 kernels are used in convolutional layers.

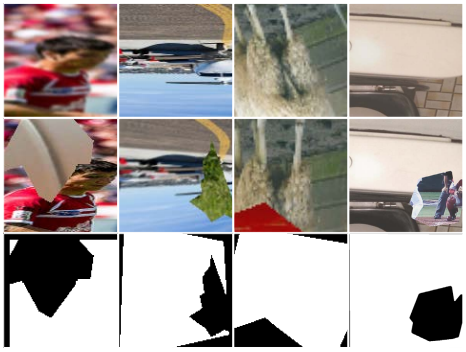

Fig. 5. Samples of the generated dataset. First two rows are image pairs with variations of brightness, contrast, saturation, image blur, and occlusions. The last row shows ground truth visibility masks.

After the tracking confidence module is trained, we monitor the tracking confidence on the fly. If the homography obtained at the previous video frame $j-1$ is classified as unreliable, we use the homography estimated in more previous times (e.g. 2 to 60 frames before) for homography initialization and re-run our model pipeline. This process is repeated until this tracking is reliable.

4 Experiments

Similar to [9, 11], we use the MS-COCO dataset [25] to generate the training data. All images are resized to 240×240. We randomly select an image, assign a 120×120 window to its center. We then randomly perturb the four corners of this window to generate a random homography. The corner displacement is uniformly distributed between [-32, 32] in both horizontal and vertical directions. Pixels within the perturbed window are wrapped to a sample image whose size is $W \times H$. To increase the robustness of our network, we augment our samples with more conditions that we meet in real-world applications. We add variances of brightness, contrast, saturation and image blur to the sample images [38]. Moreover, we simulate real-world object occlusions by randomly placing arbitrary polygons, whose textures are cropped natural images from [25], into our training samples [38]. 280000 image pairs with ground truth homography are generated in total (Fig. 5). Among them, 200000 samples are used for training the motion estimation network and refinement network, 40000 samples are used for validation, and the rest 40000 samples are tested for ablation study (Sect. 4.2). GT visibility masks are generated at each training iteration.

4.1 Training & Quantitative Evaluation

In all experiments, we set $W = H = 120$. Adam [22] optimization with $\beta_1 = 0.9$, $\beta_2 = 0.999$ is used, and the batch size is set to 32. Batch normalization [21] is

adopted for accelerating convergence. The learning rate is initialized to be 10^{-4}. It is then decreased by a factor of 10 every 5 epochs. After the model is trained, its processing rate is about 10hz on a commodity GPU card GeForce GTX 1080.

In this paper, two quantitative metrics, alignment error (AE) [34] and homography discrepancy (HD) [23], are used to evaluate the quality of predicted homography accuracy.

4.2 Ablation Study

In this section, we perform ablation studies to analyze the contribution of each component in our proposed model. All methods are trained on the training dataset as well as tested on the dataset from Sect. 4 introduction.

Homography Precision We firstly analyze component contributions to the homography precision. We train our model with increasing components proposed in this paper: the correlation block in Sect. 3.3 (D), pyramids in Sect. 3.4 (P), joint learning of homography and visibility in Sect. 3.5 (M), the refinement module in Sect. 3.6 (R): Ours-D, Ours-DP, Ours-DPR, Ours-DPM, Ours-DPMR. If our model is trained without any proposed components (Ours w/o DPMR), it is equivalent to DeepHomography [11]. Tab. 1 shows the results:

Table 1. Ablation study and comparison on our test set.

Method	AE [34]	HD [23]
Ours w/o DPMR	6.678	14.983
Ours-P	5.280	10.627
Ours-PR	2.970	5.104
Ours-PM	4.051	7.984
Ours-PMR	2.426	4.262
Ours-D	4.173	9.147
Ours-DP	1.145	2.216
Ours-DPR	**0.876**	1.739
Ours-DPM	1.097	2.107
Ours-DPMR	**0.876**	**1.695**

– From line 2 and line 7, we see that the model with correlation blocks (Ours-D) performs significantly better than that without them (Ours w/o DPMR).
– Pyramids (P) do help both approaches (Ours-D and Ours w/o DPMR). This improvement is more significant for the model Ours-D as the cost volume is constructed on limited displacements.
– The refinement module is able to capture tiny displacement between images. It further increases the accuracy for all models (Ours-DP vs Ours-DPR, Ours-DPM vs Ours-DPMR, Ours-P vs Ours-PR, Ours-PM vs Ours-PMR).
– By jointly training homography and visibility, our model generalizes better on each original task (Ours-DP vs Ours-DPM, Ours-P vs Ours-PM, and Ours-PR vs Ours-PMR).

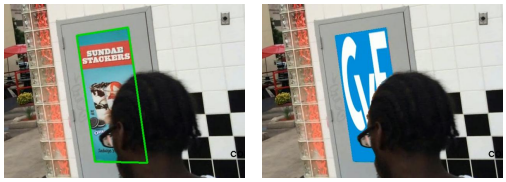

Fig. 6. A challenging case with large and irregular occlusions.

Table 2. Visibility loss of models w/ or w/o the correlation block (D), w/ or w/o joint homography and visibility learning (M vs V).

Method	Ours-PVR	Ours-PMR	Ours-DPVR	Ours-DPMR
Visibility loss	0.347	0.346	0.335	**0.328**

Visibility Accuracy Apart from the improvement to homography precision, we wonder whether learning of homography and visibility jointly (M) leads to higher visibility accuracy than learning these two tasks independently (V). We also test if the correlation block helps visibility accuracy. We train four models on the generated training dataset: Ours-PMR, Ours-DPMR, Ours-PVR and Ours-DPVR. We then compute the visibility loss (Sect. 3.5) on the test set. Results are shown in Tab. 2. We find that the correlation block and joint learning not only help the homography predictions but also improve the visibility estimations. We see strong connections between homography and visibility. Visibility, a by-product of our work, can be used for in-video advertising. We show one synthesized frame (Fig. 6) using our obtained visibility during experiments on the POT dataset [23]. We meet large and irregular occlusions that are challenging to our model. Fortunately, our model is able to overcome this difficulty.

Confidence Effectiveness One way to evaluate the confidence effectiveness is to compute the classification statistics using the predicted confidence (0.5 is used as the threshold). We follow data generations in Sect. 4 introduction to generate an additional large dataset covering challenging conditions. This dataset, on which tracking is much harder than that of in Sect. 4 introduction, contains 50000 samples. The percents of training, validation and testing are 80%, 10% and 10% respectively. Our trained models (Ours-DPR and Ours-DPMR) are then run on this dataset. If the computed L_d is smaller than 5, the tracking result is labeled to be reliable. Otherwise, it is labeled to be unreliable. Obtained labels are adopted for training the confidence network and testing the confidence performance. PatchCon [36] that directly regresses this confidence from wrapped images is the baseline/competing method. Both OursCon and PatchCon are trained to evaluate pre-trained Ours-DPMR and Ours-DPR.

True-positive rate (TPR), false-positive rate (FPR), false-negative rate (FNR) and true-negative rate (TNR) are shown in Tab. 3. Comparing OursCon and PatchCon [36] that both evaluate Ours-DPMR, we see that tracking confidence

Table 3. Classification statistics using the estimated confidence.

Method + Pre-trained Base	TPR	FPR	FNR	TNR
PatchCon [36]+Ours-DPMR	93.1%	14.5%	6.9%	85.5%
OursCon+Ours-DPMR	**96.6%**	**8.7%**	**3.4%**	**91.3%**
OursCon+Ours-DPR	96.5%	10.2%	3.5%	89.8%

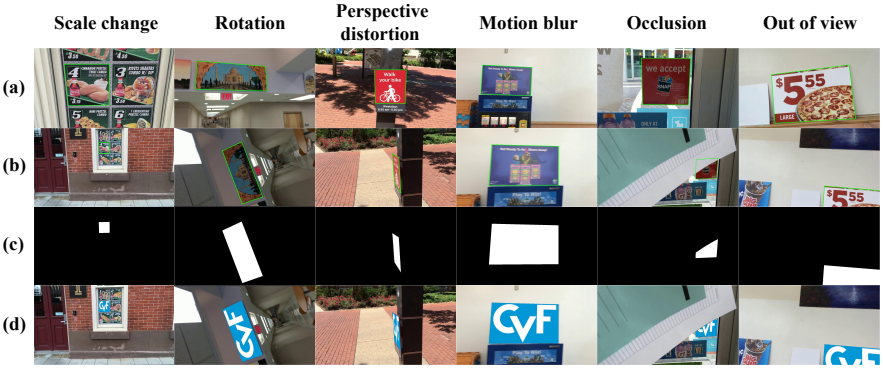

Fig. 7. Results obtained by our model in different conditions. (a) A planar object in the reference frame. (b) The tracked planar object in the current frame. (c) Predicted visibility mask corresponding to (b). (d) The synthetic frame after placing the CVF logo on (b). More results can be found in the supplementary material.

predicted from correlation blocks is more accurate. Moreover, from OursCon +Ours-DPR and OursCon+Ours-DPMR, we see that joint learning of visibility mask and homography does improve the effectiveness of our correlation block and model generalization, leading to performance gains of confidence prediction.

4.3 Comparisons on Other Datasets

Two public datasets, POT [23] and TMT [34], are used to evaluate the homography accuracy. State-of-the-art methods, including SIFT [12], SURF [6], L1 [5, 26], IVT [33], ESM [7], Gracker [43], DeepHomography [11], IC-STN [24], Ctx-Unsupervise [47], PFN [46], MHN [20] and DLKFM [48] are compared. Our models are all with our tracking confidence module (OursCon), except the one named Ours-DPMR w/o OursCon. The competing confidence prediction method, PatchCon [36], is also included for comparison (Ours-DPMR-PatchCon). The model with all our modules achieves the best performance.

POT is a planar object tracking benchmark containing 210 videos of 30 planar objects in natural environments. It contains scenes with various challenging conditions, including scale change, rotation, perspective distortion, motion blur, occlusion, out-of-view, and a combination of these factors. For better presentation, comparisons are shown with precision plots and success plots. Precision plot counts the percentage of frames whose AE is within the threshold t_p. Success plot counts the percentage of frames whose HD is within a threshold t_s. Results are

Table 4. Success rate of different approaches on the TMT dataset with AE < 5 [34]. Larger is better. Best and second best are colored. (*) Models of Ours-DPMR, Ours-DPMR-PatchCon and Ours-DPMR w/o OursCon perform the same. We omit rested notations for short.

Method	Cereal	Book1	Book2	Book3	Juice	Mug1	Mug2	Mug3	Bus	Highlight	Letter	Newspaper
SIFT [12]	0.92	0.74	1.00	0.84	0.89	0.91	0.43	0.55	0.19	0.97	0.18	0.16
SURF [6]	0.91	0.64	1.00	0.74	0.50	0.07	0.14	0.06	0.19	0.94	0.08	0.01
L1 [26]	0.24	0.10	0.79	0.42	0.16	0.10	0.30	0.54	0.57	0.67	0.19	0.61
IVT [33]	0.99	0.48	0.30	0.72	0.98	0.91	0.72	0.68	0.94	0.95	0.25	0.92
ESM [7]	1.00	1.00	1.00	0.34	1.00	1.00	0.89	1.00	1.00	0.76	1.00	1.00
Gracker [43]	0.91	1.00	1.00	0.88	1.00	1.00	0.83	0.75	0.97	1.00	0.78	1.00
DeepHomography	0.92	1.00	1.00	0.82	0.99	0.93	0.65	0.80	0.50	0.99	1.00	0.95
IC-STN [24]	0.92	1.00	1.00	0.82	1.00	1.00	0.77	0.79	0.99	0.98	1.00	0.95
PFN [46]	0.74	0.28	0.92	0.38	0.39	0.89	0.40	0.88	0.24	0.78	0.29	0.53
Ctx-Unsupervise	0.54	0.38	1.00	0.38	0.29	0.28	0.23	0.39	0.16	1.00	0.17	0.14
MHN [20]	0.62	0.18	0.92	0.40	0.63	0.99	0.50	0.41	0.50	0.76	0.22	0.14
DLKFM [48]	0.58	0.18	0.92	0.41	0.63	0.99	0.50	0.41	0.50	0.76	0.21	0.14
Ours-D	0.85	0.65	0.84	0.67	0.37	1.00	0.78	0.72	0.71	0.92	0.50	0.32
Ours-DP	0.93	1.00	1.00	0.86	1.00	1.00	0.84	0.81	0.97	1.00	1.00	0.93
Ours-DPR	0.93	1.00	1.00	0.88	1.00	1.00	0.83	0.80	0.95	1.00	1.00	0.98
Ours-DPM	0.93	1.00	1.00	0.88	1.00	1.00	0.72	0.83	0.99	1.00	1.00	0.93
Ours-DPMR (*)	0.93	1.00	1.00	0.88	1.00	1.00	0.89	0.85	0.99	1.00	1.00	1.00

shown in Fig. 8 and the supplementary material. Our proposed method shows superior performance in all scenes. Especially for scenes with motion blur, perspective distortion, scale change or combinations of these factors, our approach works much better because it is hard for non-learning algorithms to model the underlying variation or tuning related parameters manually.

TMT consists of sequences for manipulation tasks. There are 100 annotated and tagged sequences in total. Similar to POT, sequences in this dataset also have a large condition variation. We use the same evaluation metric as in [34]. That is, the success rate that counts the percentage of frames whose AE < 5. Comparison results are summarized in Table. 4. Overall, our model achieves a better or similar performance in all sequences compared to other methods.

We visualize some qualitative results obtained by our model during experiments and place a product (i.e. the CVF logo) on the tracked planar object in Fig. 1 and Fig. 7. More results can be found in the supplementary material.

5 Discussions & Limitations & Conclusions

The main limitation of our work is that the predicted visibility mask is not perfect. With the own constraints of LK-based methods, our approach is sometimes disturbed by the factor of similar occluded objects. In conclusion, we proposed a novel model for planar object tracking. Homography, visibility and confidence are jointly learned based on a correlation block. We achieved a superior planar tracking performance compared to state-of-the-art methods on the public dataset, provided visibility masks that other works had not discussed, calculated more reliable confidence than competing approaches. To better take multi-frame constraints and similar occlusions into consideration is our future work.

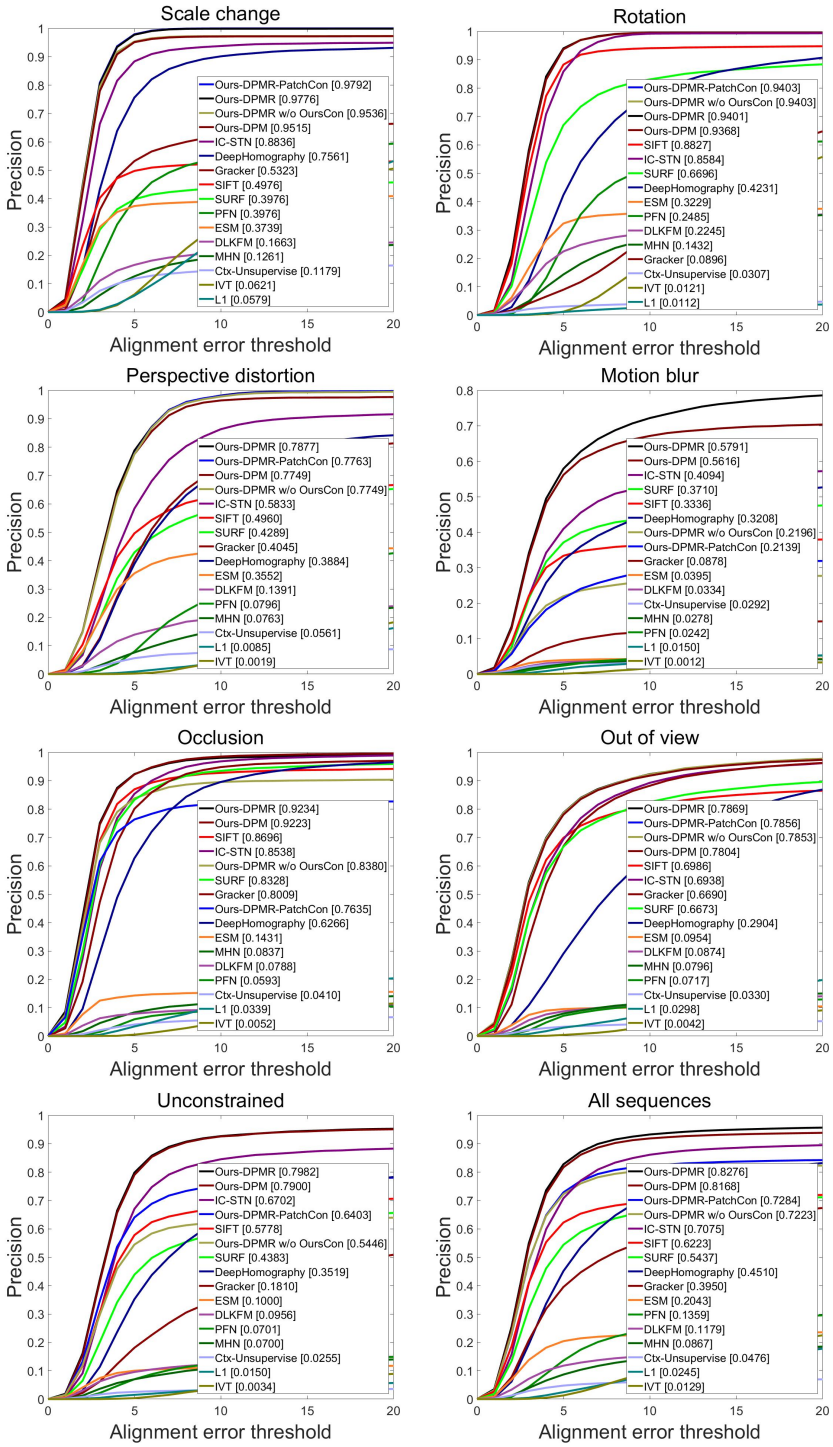

Fig. 8. The comparison of different approaches shown in precision plots on the POT dataset [23]. Curves with larger areas are better. The AE at threshold = 5 [34] is illustrated within brackets. Zoom-in is recommended. Video comparisons are in the supplementary material.

References

1. Badrinarayanan, V., Galasso, F., Cipolla, R.: Label propagation in video sequences. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2010)
2. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. ACM Trans. Graph. (2009)
3. Baker, S., Matthews, I.: Equivalence and efficiency of image alignment algorithms. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2001)
4. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. International Journal of Computer Vision **56**(3), 221–255 (2004)
5. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2012)
6. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: European Conference on Computer Vision. (2016)
7. Benhimane, S., Malis, E.: Real-time image-based tracking of planes using efficient second-order minimization. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (2004)
8. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: European Conference on Computer Vision. (2010)
9. Chang, C.H., Chou, C.N., Chang, E.Y.: CLKN: Cascaded lucas-kanade networks for image alignment. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2017)
10. Chen, L., Zhou, F., Shen, Y., Tian, X., Ling, H., Chen, Y.: Illumination insensitive efficient second-order minimization for planar object tracking. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2017)
11. DeTone, D., Malisiewicz, T., Rabinovich, A.: Deep image homography estimation. In: ArXiv preprint arXiv:1606.03798 (2016)
12. D.G. Lowe: Object recognition from local scale-invariant features. In: Proc. of the IEEE Intl. Conf. Comput. Vis (1999)
13. Dick, T., Quintero, C.P., Jägersand, M., Shademan, A.: Realtime registration-based tracking via approximate nearest neighbour search. In: Proc. of Robot.: Sci. and Syst. (2013)
14. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2019)
15. Fan, Q., Zhong, F., Lischinski, D., Cohen-Or, D., Chen, B.: Jumpcut: Non-successive mask transfer and interpolation for video cutout. ACM Trans. Graph. (2015)
16. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM (Jun 1981)
17. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2010)
18. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2015)
19. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision (2nd). Cambridge University Press (2003)

20. Hoang Le, Feng Liu, S.Z.A.A.: Deep homography estimation for dynamic scenes. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2020)
21. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Liang, P., Wu, Y., Lu, H., Wang, L., Liao, C., Ling, H.: Planar object tracking in the wild: A benchmark. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2017)
24. Lin, C.H., Lucey, S.: Inverse compositional spatial transformer networks. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2017)
25. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. (2014)
26. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: Proc. of the IEEE Intl. Conf. Comput. Vis (2009)
27. Nguyen, T., Chen, S.W., Shivakumar, S.S., Taylor, C.J., Kumar, V.: Unsupervised deep homography: A fast and robust homography estimation model. IEEE Robotics and Automation Letters (2018)
28. Ozuysal, M., Calonder, M., Lepetit, V., Fua, P.: Fast keypoint recognition using random ferns. IEEE transactions on pattern analysis and machine intelligence **32**(3), 448–461 (2009)
29. Ramakanth, S.A., Babu, R.V.: Seamseg: Video object segmentation using patch seams. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2014)
30. Revaud, J., Weinzaepfel, P., de Souza, C.R., Humenberger, M.: R2D2: repeatable and reliable detector and descriptor. In: Neural Information Processing Systems (2019)
31. Richa, R., Sznitman, R., Taylor, R., Hager, G.: Visual tracking using the sum of conditional variance. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (2011)
32. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (2015)
33. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Intl. J. Comput. Vis. **77**(1-3), 125–141 (2008)
34. Roy, A., Zhang, X., Wolleb, N., Perez, Quenterio, C., Jagersand, M.: Tracking benchmark and evaluation for manipulation tasks. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2015)
35. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: Proc. of the IEEE Intl. Conf. Comput. Vis (2011)
36. Seki, A., Pollefeys, M.: Patch based confidence prediction for dense disparity map. In: Proceedings of the British Machine Vision Conference (2016)
37. Shaked, A., Wolf, L.: Improved stereo matching with constant highway networks and reflective loss. In: arXiv preprint arxiv:1701.00165 (2016)
38. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning (2019)
39. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ArXiv preprint arXiv:1606.03798 (2014)

40. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2018)
41. Taylor, B., Karasev, V., Soattoc, S.: Causal video object segmentation from persistence of occlusions. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2015)
42. Vijayanarasimhan, S., Grauman, K.: Active frame selection for label propagation in videos. In: European Conference on Computer Vision. (2012)
43. Wang, T., Ling, H.: Gracker: A graph-based planar object tracker. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
44. Wang, W., Shen, J., Porikli, F.: Saliency-aware geodesic video object segmentation. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2015)
45. Yuki Ono, Eduard Trulls, P.F., Yi, K.M.: LF-Net: Learning Local Features from Images. In: Neural Information Processing Systems (2018)
46. Zeng, R., Denman, S., Sridharan, S., Fookes, C.: Rethinking planar homography estimation using perspective fields (2018)
47. Zhang, J., Wang, C., Liu, S., Jia, L., Ye, N., Wang, J., Zhou, J., Sun, J.: Content-aware unsupervised deep homography estimation. In: European Conference on Computer Vision. pp. 653–669. Springer (2020)
48. Zhao, Y., Huang, X., Zhang, Z.: Deep lucas-kanade homography for multimodal image alignment. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2021)