Local Homography Estimation on User-specified Textureless Regions

Zheng Chen
Tsinghua University
Beijing, China
chenz20@mails.tsinghua.edu.cn

Xiaonan Fang
Tsinghua University
Beijing, China
fangxn18@mails.tsinghua.edu.cn

Songhai Zhang
Tsinghua University
Beijing, China
shz@tsinghua.edu.cn

Abstract

This paper presents a novel deep neural network for designated point tracking (DPT) in a monocular RGB video. More concretely, the aim is to track four designated points correlated by a local homography on a textureless planar region in the scene. DPT can be applied to augmented reality and video editing, especially in the field of video advertising. Existing methods predict the location of four designated points without appropriately considering the point correlation. To solve this problem, our network predicts the motion of the four designated points correlated by a local homography within the heatmap prediction framework. Our network refines the heatmaps of designated points through two stages. On the first stage, we introduce a context-aware and location-aware network to learn a local homography for the designated plane in a supervised way. On the second stage, we further introduce an iterative heatmap refinement module to improve the tracking accuracy. We propose the DPT dataset focusing on textureless planar regions, named ScanDPT, for training and evaluation. We show that our algorithm outperforms the state-of-the-art approaches on ScanDPT.

1. Introduction

Tracking is a widely studied topic for video understanding and editing. In general, tracking algorithm could be categorized into two types: object-level tracking and pixel-level tracking. Object tracking aims to locate the important object in each video frame, usually providing a bounding box, while pixel-level tracking, known as optical-flow estimation, aims to find pixel correspondence between current frame and the next one.

In this work, we investigate a different problem, called designated point tracking (DPT). This problem is depicted in Figure 1. In video editing, people often need to modify a rectangular region on a textureless plane, for example, putting a poster on the wall. Obviously, tracking the planar region is equivalent to tracking the four corner points. Given the initial state (four corner points on a target rectangle) in the first frame of a video sequence, the aim of DPT is to automatically obtain the states of the four target corner points in the subsequent frames. Designated point tracking is of great value in the field of augmented reality and video editing. We can map a prepared image to the designated region on each frame to obtain a new video. This problem has two major characteristics. Firstly, the four designated points are on the same plane of a complex scene. In other words, they are correlated by a local homography between frames. Secondly, different from planar object tracking, four designated points are usually on the background where few features inside the four designated points could be extracted for tracking. Compared to textured planar object, textureless background regions, e.g. walls and doors, are more common in general videos.

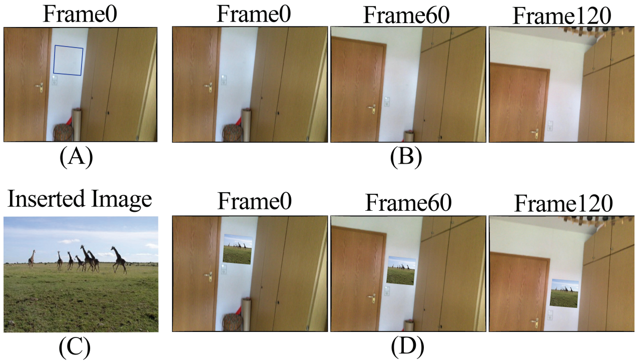

Figure 1. Task Explanation. Designated point tracking (DPT) takes as input four corner points of the first frame like (A) and a video sequence like (B). It outputs four target corner points in the subsequent frames. Finally a prepared image like (C) can be inserted into the video sequence with the assistance of the local homography between four estimated designated points in each frame and the image corner points of the prepared image. (D) is the composition result.

It is nontrivial to predict the motion of the four designated points correlated by a local homography on textureless background regions with existing methods. If we use optical flow methods [44, 7, 18, 29, 51, 55, 65, 57], the motion of four designated points are predicted individually without considering the correlation. Object tracking

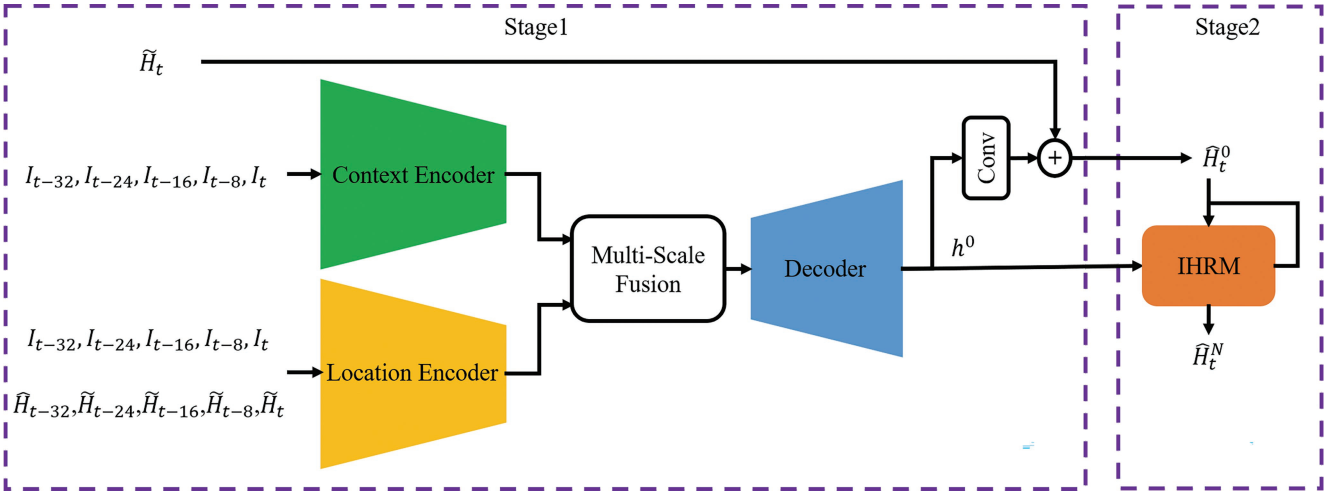
Stage1
Stage2
$\widetilde{H}_t$
$I_{t-32}, I_{t-24}, I_{t-16}, I_{t-8}, I_t$
Context Encoder
$I_{t-32}, I_{t-24}, I_{t-16}, I_{t-8}, I_t$
$\widetilde{H}_{t-32}, \widetilde{H}_{t-24}, \widetilde{H}_{t-16}, \widetilde{H}_{t-8}, \widetilde{H}_t$
Location Encoder
Multi-Scale Fusion
Decoder
Conv
+
h^0
$\widehat{H}_t^0$
IHRM
$\widehat{H}_t^N$

Figure 2. The overall network structure. The context encoder encodes contextual information with the input of a video sequence. The location encoder encodes the spatial information from heatmap proposals. The location features and contextual features are fused in a multi-scale way. Then the fused features are decoded into a latent hidden state h^0 and intermediate heatmaps $\hat{H}_t^0$. The predicted heatmaps and the latent hidden state are jointly refined to produce better heatmaps $\hat{H}_t^N$ in the proposed iterative heatmap refinement module (IHRM) after N iterations.

methods are inappropriate for tracking background points as they are generally designed to track foreground objects such as pedestrians, animals, bikes, vehicles and so on [4, 41, 52, 3, 27, 50, 35, 34, 25, 24, 11, 58]. Those objects being tracked have more discriminative features than the background rectangle defined by the designated points. Homography estimation methods consider the prior that four designated points are on the same plane visually [40, 17, 43, 54, 30, 19, 20, 6, 9, 8]. However, inside the region of four designated points, few features can be detected. Under these circumstances, template-based homography estimation methods are unable to get a reasonable local homography. With the input of full images, these methods extract and match features in two frames, and then a global homography is estimated by using almost all the matched features after removing outliers. When a local homography is needed for a designated plane, it is nontrivial to select the relevant features for the designated planar region using existing methods. If we directly apply a plane detection method like [39, 38, 31] to filter feature points with the plane mask, we may neglect important feature points on the edges of the plane. Besides, these plane detection methods are not robust to small planar region.Without enough appropriate corresponding features relevant to the designated plane, the estimated homography matrix deviates from the actual one.

We adopt 4-point parameterization [16] to represent homography. Different from HomographyNet [16], the four points in our methods are user-specified and can be placed on any plane in a complex scene while four points in HomographyNet are the first input image corner points by default. As the template has few features, we have to input the full image with the contextual information as the reference. HomographyNet tends to predict a global homography by predicting image corner points. In contrast, our network is dedicated to predict a local homography for a designated plane with the input of full image and four designated points.

In detail, we use state-of-the-art optical flow method RAFT [57] to calculate coarse-grained location individually for each point in the next frame. We adopt heatmaps to represent the locations of designated points and try to optimize them through a two-stage network. The first stage contains two encoder branches (location encoder and context encoder) and one decoder, and the second stage contains an iterative heatmap refinement module (IHRM). We improve the result of RAFT with a local homography constraint in a supervised way. On the first stage of our network, the designated plane in each frame is emphasized by explicitly feeding four coarse-grained points into our network for each frame. In addition, the location of each point is refined by aggregating the spatial information of other three points. And according to the setting of 4-point parameterization, our network learns a local homography for the designated plane by jointly supervising the locations of the four designated points. Besides, we leverage a context encoder to complement the features of surrounding edges besides the inner region of four designated points, which benefits a lot especially when features are rare inside the designated plane. Due to the dual-encoder structure and the local homography supervision, our network can get fine-grained predictions. On the second stage, inspired by the success of iterative methods in other fields [57, 32], we propose to utilize a iterative heatmap refinement module to refine the locations of four designated points and the latent hidden state repeatedly. In each iteration, the predictions are constrained by the local homography of designated points, which is the same as that on the first stage. After several iterations and intermediate supervision, we can get more accurate fine-grained predictions.

Existing planar object tracking benchmark datasets [37, 53, 22, 36, 9] focus on textured objects. Instead, we constructed a video dataset (ScanDPT) based on ScanNet [13] mainly for textureless background regions.

Our contributions can be summarized as follows. 1) We propose a context-aware and location-aware network which predicts a local homography for a textureless plane in a supervised way. 2) We propose an iterative heatmap refinement module to further improve the tracking accuracy. 3) We construct the first dataset mainly for textureless planar background regions named ScanDPT, on which our network makes significant improvements over the current state-of-the-art methods.

2. Related Work

We firstly review a potential approach, 3D reconstruction, for the DPT task. And then we review three relevant types of 2D approaches, including homography estimation, object tracking and optical flow estimation.

2.1. 3D reconstruction

Some methods like [47, 59, 14, 49] can well reconstruct 3D scenes for RGB-D cameras since both depths and camera parameters are provided. 3D objects or 2D objects can be inserted into these scenes and keep relatively static in the video. Some other methods utilize additional sensors like inertial measurement unit (IMU) to support the reconstruction [46]. Without any other sensor, some researchers use structure from motion (SFM) or visual simultaneous localization and mapping (vSLAM) methods to estimate camera poses, and then estimate dense depth maps with deep neural networks. And finally these depths are fused to generate surface mesh [61, 64]. However, these methods still need camera intrinsic matrices obtained by calibration to assist the calculation of re-projection errors. A group of people like [63] utilize other SFM methods such as OpenSfM [1] to estimate both camera intrinsic and extrinsic matrices. Nevertheless, the reconstructed scene is unreliable without enough different views. We assume that designated point tracking (DPT) is a task constrained in monocular RGB videos without given camera parameters. And for these methods in 3D reconstruction, most of cases encountered in DPT do not provide enough different views to reconstruct a proper 3D geometry and estimate accurate camera parameters.

2.2. General Object Tracking

General object tracking methods include statistical learning [4], subspace learning [52], template matching [41], discriminative correlation filters [27], particle filters [3] and deep neural networks [50, 35, 34, 25, 24, 11, 58]. These object tracking methods are inappropriate for the designated point tracking (DPT) task. They solely predict one coarse rectangular bounding box for one object, not for a single point. Besides, the aim of the DPT task is to track four designated points on the background while object tracking methods are designed to track a foreground object, such as pedestrians, animals, vehicles, etc. In practice, large tracking drifts often occur when object tracking methods are applied to background points.

2.3. Optical Flow Estimation

Traditional optical flow estimation method like [28] formulates dense pixel tracking as an energy minimization problem based on the prior of spatial smoothness and brightness constancy. To solve large displacements better, the coarse-to-fine strategy is widely used [7]. Some later works propose to match features using CNN [60]. FlowNet [18] is the first end-to-end optical flow estimation network and is trained on a synthetic dataset. Afterwards, many follow-up works improve the initial network [29, 51, 55, 65]. Recently, RAFT [57] has achieved state-of-the-art accuracy as it maintains and updates a single fixed flow field at high resolution with a recurrent and lightweight updating operator. As for the designated point tracking (DPT) task, optical flow methods estimate four designated points individually without considering the correlation of a local homography, so the trajectories of the four points could deviate, leading to the deformation of rectangular region defined by them. We use the optical flow prediction to initialize the heatmap for designated points and improve it through our network.

2.4. Homography Estimation

Homography estimation is of great importance in computer vision. A homography is a 3×3 matrix that relates two images of a planar scene. It consists of 8 degrees of freedom (DOF), with 2 parameters for scale, translation, rotation and perspective respectively [2]. Feature-based methods detect feature points, such as SIFT [40], SuperPoint [17] and ASLFeat [43]. They match the corresponding feature points with Nearest Neighbor (NN) search [45] or a learned matcher like SuperGlue [54]. Some methods like DFM [19] and COTR [30] refine the feature extraction and matching jointly. A robust homography estimation algorithm like RANSAC [21] and MAGSAC [5] is often used to reject outliers. Seminal Lucas-Kanade algorithm [42] updates homography and guides the shift of the image by calculating the sum of squared differences (SSD) between template and the target image. To improve the accuracy, robust enhanced correlation coefficient (ECC) is used to replace SSD [20]. More later works like ESM [6], GO-ESM [9] and GOP-ESM [8] improve the initial template-based methods. Some deep-learning methods predict homography in an end-to-end way. HomographyNet [16] is an end-to-end network to estimate the homography between image pairs using the 4-point parameterization. An unsupervised deep learning algorithm [48] is proposed to estimate homography by minimizing a pixel-wise photometric error. Besides, a content-aware robust network structure with a triplet loss improves the initial idea by predicting a mask highlighting the aligned inliers [62].

These homography estimation algorithms are not designed to predict the homography for a textureless plane. They assume that the inner region of the template has rich textures and thus they can use only the template as the reference. On the other hand, when inputting the full image to previous algorithms as the reference, they generally predict a global homography. Moreover, it is nontrivial and infeasible to select features relevant to the designated plane with plane detection algorithms since the plane mask predicted by these plane detection algorithms cannot include all edges relevant to the plane. These plane detection algorithms might fail to detect the designated plane when it occupies a very small region. Consequently, the estimated homography deviates from the actual one without enough corresponding features relevant to the designated plane. In contrast, our method is dedicated to predict the local homography for a textureless plane by explicitly feeding the four designated points and full images into our network and the output locations of the four points are jointly supervised during training.

3. Proposed Method

3.1. Input Preparation and Overview

It is nontrivial to directly estimate the locations of the designated points in the subsequent frames by deep neural networks. Empirically, it is uneasy for the network to search the optimal location in a large image domain without given location proposals. Therefore, we adopt the current state-of-the-art optical flow method RAFT [57] to estimate the target points' location proposals of subsequent frames. We generate heatmaps with 2D Gaussian filter centered on the locations of the proposal points, i.e., current coordinate (x, y) plus the motion vector $(u(x, y), v(x, y))$, and feed them into our network.

Optical flow includes the displacement of every pixel between two neighboring frames. Since the DPT task requires sub-pixel level accuracy, i.e., float value coordinates, we use bilinear interpolation to calculate the displacements $u(x, y)$ and $v(x, y)$ from the four neighboring pixels of (x, y). Besides, previous template-based methods which use the template as the reference are unable to handle textureless planar regions. Consequently, full images are fed into our network to leverage the contextual hints such as the outer edges of textureless planar regions. Besides, to learn a local homography for a designated plane the locations of designated points are embedded

The overall network structure is illustrated in Figure 2. The network includes two encoders, a heatmap decoder and an iterative heatmap refinement module. The input heatmap is refined through two stages: an context-aware and location-aware structure as well as an iterative heatmap refinement module (IHRM). We calculate the point coordinates from predicted heatmap using integral regression.

3.2. Stage 1: Context-aware and Location-aware Refinement

On the first stage, we adopt an dual-encoder network to improve the accuracy of heatmaps. This model receives a video frame sequence $\{I_{t-32}, I_{t-24} ..., I_t\}$, aflow-to-heatmap proposals $\{\tilde{H}_{t-24}, \tilde{H}_{t-16}, ..., \tilde{H}_t\}$ and estimated locations of the $(t-32)$-th frame $\hat{H}_{t-32}$. It predicts the refined heatmaps of current frame $\hat{H}^0_t$. These flow-to-heatmap proposals $\{\tilde{H}_{t-24}, \tilde{H}_{t-16}, ..., \tilde{H}_t\}$ are calculated from $\hat{H}_{t-32}$ frame by frame according to the state-of-the-art optical flow method RAFT.

The dual-encoder structure we proposed is composed of a context encoder and a location encoder. The context encoder is designed to extract contextual information from a large receptive field because features are usually rare inside the designated rectangular area. We utilize the backbone of ResNet-50 [26] as the context encoder. This encoder only processes the video frames. The feature maps of three different levels from the first convolution, layer 1 and layer 4, with 1/2, 1/4 and 1/8 resolution of the input image respectively, are selected for later processing. The highest level feature map from layer 4 is then processed by an ASPP block proposed in DeepLab [10]. These feature maps are denoted by $f^k_{in}, k = 1, 2, 3$ respectively.

The video frames $\{I_{t-32}, ..., I_t\}$ and prior heatmaps $\{\hat{H}_{t-32}, ..., \tilde{H}_t\}$ are embedded as feature priors by our location encoder. We emphasize the designated plane in each frame by explicitly inputting four coarse-grained points into our network for each frame. The location of each point is refined by aggregating the spatial information of other three points. Parameter maps w^k and b^k in three different levels are predicted for each corresponding feature layer of the context encoder. To spread further the prior location information, dilated convolution is utilized to expand the receptive field with $stride = 2$. We fuse contextual features and prior location features by applying element-wise linear transform to feature maps at each level:

$$f^k_{out} = f^k_{in} \cdot w^k + b^k, k = 1, 2, 3. \quad (1)$$

A low level of feature map denoted as f_{out}^0 is extracted using a single convolution layer from the video sequence and prior heatmaps. The feature maps f_{out}^0 maintain the same resolution as the input. Then the feature maps $f_{out}^{0,1,2,3}$ are decoded into the latent hidden state h^0. Skip connection is used in the decoder to exploit different feature layers. Higher level features are fused and upsampled first and then concatenated with lower level features. The predicted latent hidden state also has the same resolution as the input. Then the latent hidden state h^0 is used to regress the intermediate heatmaps $\hat{H}_t^0$ with a single convolution layer. The detailed structures of our location encoder and decoder are listed in the supplementary.

3.3. Stage 2: Iterative Heatmap Refinement Module

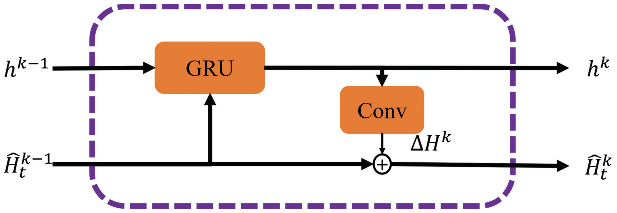

Figure 3. An updating block inside IHRM.

Inspired by the success of iterative refinement methods in other fields [57, 32], we propose the iterative heatmap refinement module as shown in Figure 3. We use intermediate heatmaps $\hat{H}_t^0$ to update the latent hidden state by explicitly enhancing the location information of the current frame. Then the updated latent hidden state can be used to predict more accurate heatmaps constrained by the intermediate supervision of the local homography.

A sequence of heatmaps $\{\hat{H}_t^1, ..., \hat{H}_t^N\}$ are estimated from intermediate heatmaps $\hat{H}_t^0$. In each iteration, the module predicts residuals ΔH_t^k which is added to the previous estimation: $\hat{H}_t^k = \Delta H_t^k + \hat{H}_t^{k-1}$. An update block inside IHRM takes heatmaps $\hat{H}_t^{k-1}$ and a latent hidden state h^{k-1} as input, and outputs four updated heatmaps $\hat{H}_t^k$ and an updated latent hidden state h^k.

The core part of IHRM is a gated recurrent unit (GRU) [12] with fully connected layers replaced by convolutions:

$$z^k = \sigma(Conv_{3\times3}([h^{k-1}, \hat{H}_t^{k-1}]), W_z), \tag{2}$$

$$r^k = \sigma(Conv_{3\times3}([h^{k-1}, \hat{H}_t^{k-1}]), W_r), \tag{3}$$

$$\tilde{h}^k = tanh(Conv_{3\times3}([r^k \odot h^{k-1}, \hat{H}_t^{k-1}]), W_h), \tag{4}$$

$$h^k = (1 - z^k) \odot h^{k-1} + z^k \odot \tilde{h}^k. \tag{5}$$

3.4. Supervision of Local Homography

The four designated points represented by heatmaps are jointly supervised. In other words, our network is supervised by a local homography for a designated plane according to the concept of 4-point parameterization. After our network outputs refined heatmaps, we need to acquire locations at sub-pixel level according to the heatmaps. Here we obtain the sub-pixel level locations using integral regression [56].

Firstly, we use softmax function to normalize the heatmap. For each pixel p, we define

$$\tilde{H}_i(p) = \frac{e^{H_i(p)}}{\sum_{q\in\Omega} e^{H_i(q)}} \tag{6}$$

where H_i is the i-th output heatmap and q is a pixel in the image domain Ω.

Then, we compute the weighted average to estimate the i-th sub-pixel level location P_i:

$$P_i = \sum_{p\in\Omega} p \cdot \tilde{H}_i(p), i = 0, 1, 2, 3 \tag{7}$$

We measure the difference between network prediction and ground-truth position with smooth L1 loss [23]. Besides, with to handle multiple outputs in IHRM, we adopt exponentially increasing weights. The total loss is defined as

$$L_{total} = \sum_{k=0}^{N} \gamma^{N-k} SmoothL1(P^k, P_{gt}) \tag{8}$$

where P^k are the predicted points of k-th iteration in IHRM and P_{gt} are their corresponding ground-truth coordinates. We set $N = 7$ and $\gamma = 0.8$ in our experiments.

3.5. ScanDPT Dataset

Training deep neural networks requires a large amount of data. To meet this requirement, we generated a large number of labeled training examples based on ScanNet [13], a large-scale RGB-D video dataset. We did not directly label the ground-truth designated points by hand on every frame, because it is error-prone and takes too much time. Instead, we labeled the four designated points in the 3D scenes reconstructed by BundleFusion [14]. For each video, we chose a textureless planar region in the corresponding 3D scene and then recorded four corner points of an approximate rectangle denoted as V on the plane:

$$V = \left(V1, V2, V3, V4\right). \tag{9}$$

Then we projected the points to the image plane and obtain the corresponding coordinates P:

$$P = M_{int} \times M_{ext} \times V \tag{10}$$

We represent the points V and P in the homogeneous coordinate system, while M_{int} and M_{ext} represent camera intrinsics and extrinsics respectively, which have been

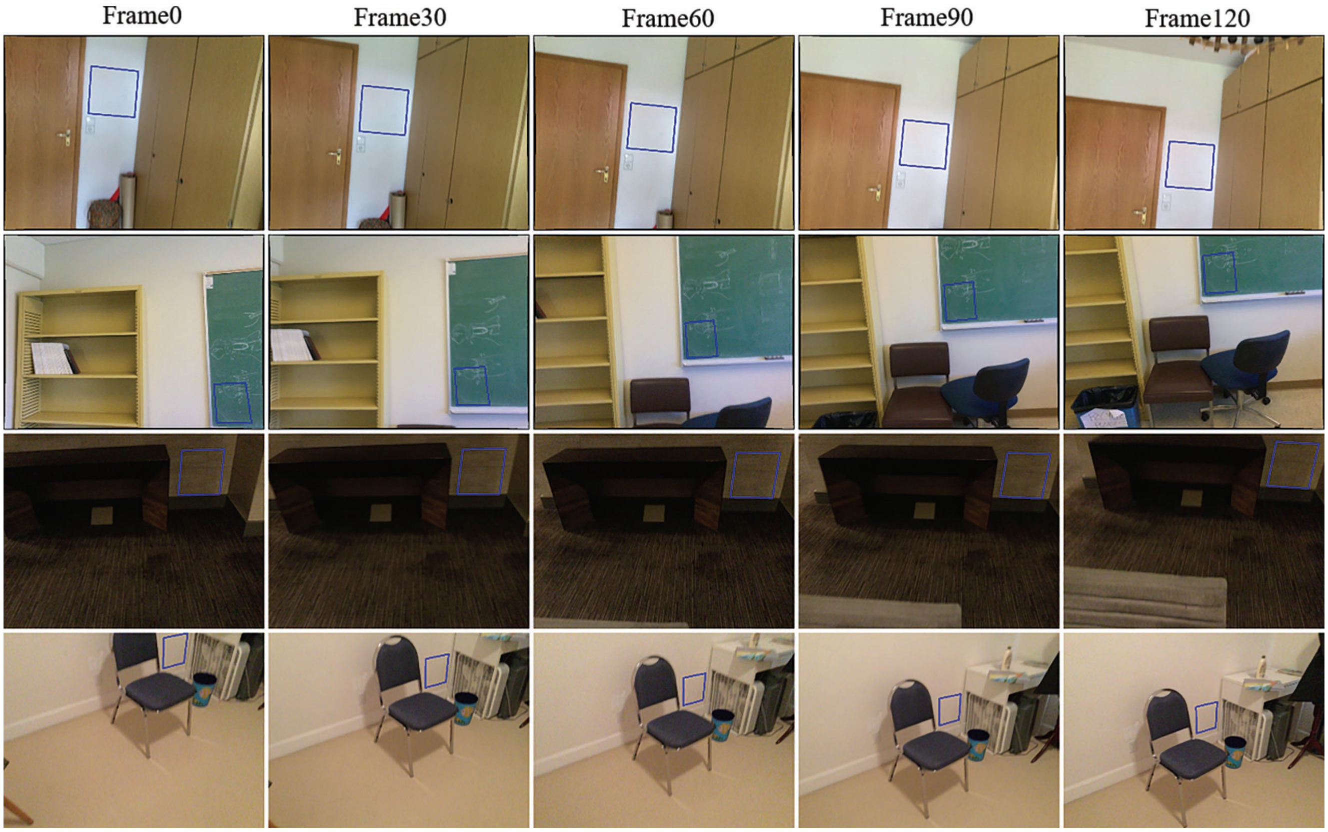

Figure 4. Example frames from ScanDPT dataset. The four corner points of blue quadrangles in each frame are the ground truth.

provided in ScanNet. We convert the coordinates into 2D form and obtain the ground-truth location of four designated points.

We played the labeled video and chose appropriate clips including the labeled areas by manually labeling the start time when the designated plane appears and the end time when the plane disappears.

The final dataset (ScanDPT) is composed of 225 videos, which are split into 161 training videos and 64 test videos. Some examples in our dataset are shown in Figure 4. More details of the dataset ScanDPT can be seen in the supplementary.

4. Experiments

4.1. Implementation Details

To train our network, we resized the video frame to a resolution of 320 × 240 for efficiency. The ResNet-50 backbone in the context encoder was pre-trained on ImageNet [15] and fine-tuned during the training process. We adopted ADAM [33] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size was set to 16. The initial learning rate was set to 2e-5, and multiplied by 0.5 every 5 epochs. We terminated the training process of our network at the 20-th epoch. The whole training process took about 24 hours on four TITAN RTX graphics cards.

We utilize the fMSE metric to evaluate various methods on the DPT task. This metric is used to evaluate the average performances of various algorithms in each video. It is defined as follows:

$$fMSE = \frac{1}{f} \sum_{j=0}^{f} \frac{\sum_{i=0}^{3} (x_i - x_i^{gt})^2 + (y_i - y_i^{gt})^2}{4} \quad (11)$$

where f denotes the number of frames in each video. Then, we calculate the mean value of fMSE for all videos.

4.2. Comparison With Existing Methods

We compared our method with other existing methods including optical-flow-based methods, homography-based methods and object-tracking-based methods on the ScanDPT dataset.

Optical-flow-based methods: We compared our algorithm with several optical flow methods, including FlowNet2.0, SPyNet, PWCNet, MaskFlownet and the state-of-the-art network RAFT. As dense optical flow methods calculate pixel-level displacements, bilinear interpolation was used to get the final sub-pixel level displacements. For

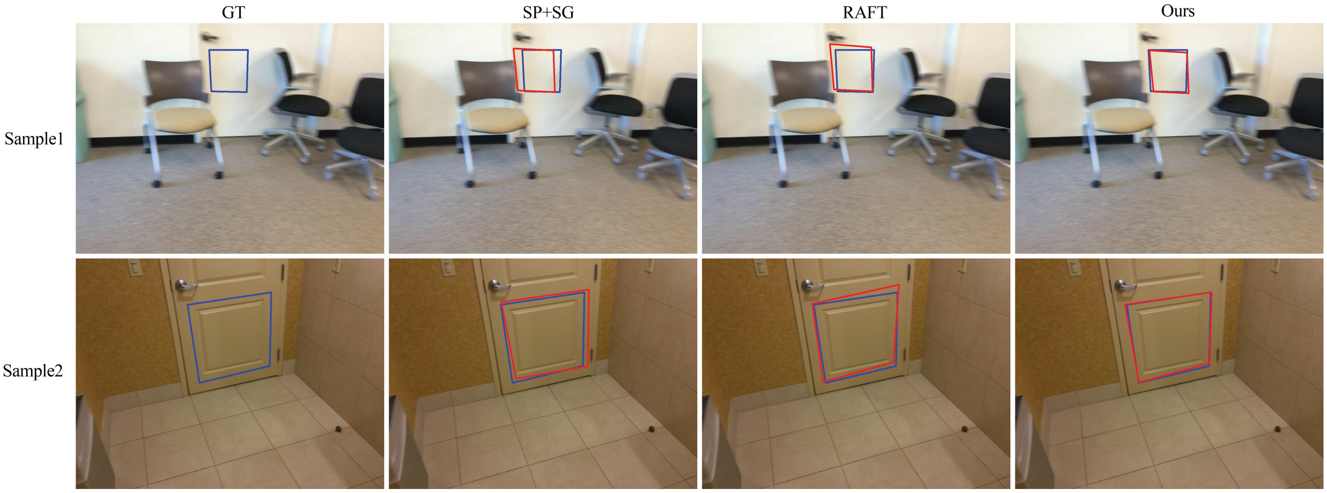

Figure 5. Comparison of our method, RAFT and SP+SG. The two test images are the 120th frame of the test videos. The corner points of the blue quadrangles are the ground truth and the corner points of the red ones are the predictions of corresponding approaches.

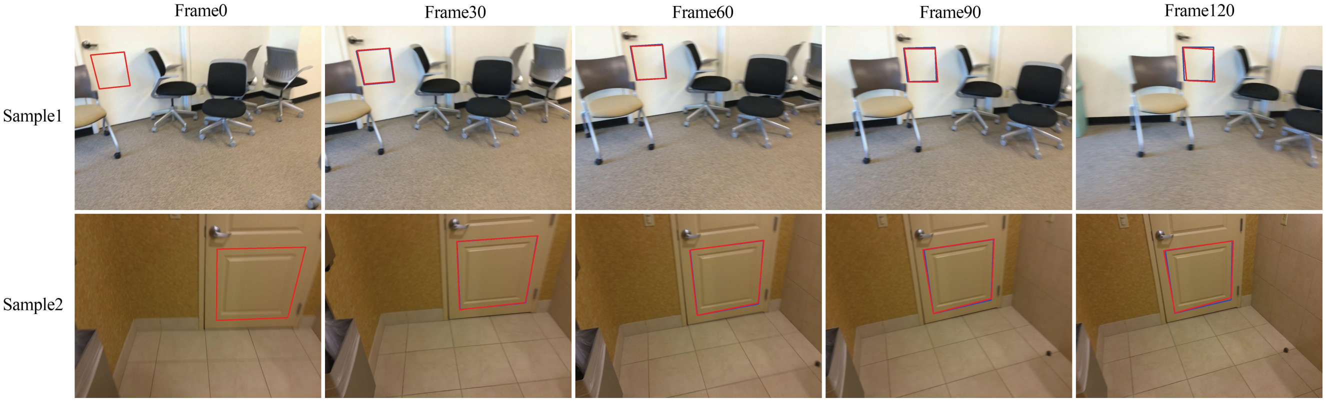

Figure 6. Sequential predictions of our method on two test videos. The corner points of the blue quadrangles are the ground truth and the corner points of red ones are the predicted by our network.

current frame, the locations of fours points were updated from the location of previous frame by $P_t = P_{t-1} + \Delta P$.

Homography-based methods: We used SIFT and ASLFeat as the feature descriptors and Nearest Neighbor (NN) search as the feature matcher. SuperPoint was combined with the learned matcher SuperGlue (denoted by SP+SG). Besides, DFM and COTR were chosen as two baselines which extract features and match features jointly in one framework. RANSAC algorithm was chosen to regress the homography. For the current frame, the locations of four points were updated from those of previous frame by $P_t = M \times P_{t-1}$. M is the 3×3 estimated homography matrix between the neighbor frames. P_t and P_{t-1} are represented in homogeneous coordinate system.

Object-tracking-based methods: We chose SiamRPN++, SiamCAR, SiamGAT, TransT and TrDiMP as object tracking baselines. With each point centered in a bounding box at the initial frame, object tracking methods tracked the location of each bounding box in subsequent frames. Then the center of the predicted bounding box in subsequent frames was considered as the final result. We experimented with bounding boxes sized 25×25, 20×20, 15×15, 10×10 and 5×5 respectively for each object tracking algorithm. The best results of different bounding box sizes for the five object tracking algorithms are reported in Table 1.

We demonstrate the performance of our method by comparing it with all of the methods above quantitatively. We report the errors for the overall average error fMSE in Table 1. We measure the mean error in the first $30, 60, 90, 120$ frames of all the testing videos respectively. Our method outperforms the others in fMSE. Our error is about 29% lower than the best other method RAFT in fMSE when testing in the first 120 frames of testing videos. SuperPoint+SuperGlue worked best in homography-based methods, but its fMSE is higher than ours by 127% in the first

frame intervals	[0, 30)	[0, 60)	[0, 90)	[0, 120)
SiamGAT	339.72	628.20	898.82	1164.05
TransT	184.67	420.77	687.44	904.48
TrDiMP	93.50	291.04	546.34	826.04
SiamCAR	135.88	324.24	530.27	720.12
SiamRPN++	106.48	279.49	476.79	665.05
DFM	47.50	163.81	327.10	508.25
COTR	18.82	60.10	113.17	171.20
SIFT+NN	18.30	55.41	103.38	164.96
ASLFeat+NN	25.38	62.63	100.90	161.00
SP+SG	4.09	10.87	17.02	22.61
SPyNet	13.56	40.31	78.01	121.36
MaskFlownet	6.66	18.54	31.23	42.47
PWCNet	4.51	13.13	24.40	35.76
FlowNet2.0	4.99	13.20	23.14	33.98
RAFT	2.45	6.11	10.12	14.09
Ours	2.05	4.51	7.26	9.98

Table 1. Comparison with existing methods. Each row (except the first column) reveals the tracking statistics on dataset ScanDPT in fMSE. The first row shows the frame intervals we used. SP+SG denotes SuperPoint+SuperGlue. NN denotes Nearest Neighbor search.

120 frames. All the five object tracking methods have very poor performances as large tracking drifts often occur in these object tracking methods especially when tracking the background points.

As shown in Figure 5, our method predicts the most accurate locations because it learns the local homography for the textureless planar region in a supervised way. Sequential predictions of the two examples can be found in Figure 6. More implementation details of other methods can be seen in the supplementary.

4.3. Ablation Study

On the ScanDPT dataset, we also conducted the ablation study of alternative input solutions, loss function and encoder based on our basic model excluding IHRM. And then we added IHRM to the basic model to validate its effectiveness.

Input solutions: Firstly we test our network with different frame sequences during training. We utilize two input variations:

(a): input multiple neighbor historical frames as $(I_{t-4}, I_{t-3}, I_{t-2}, I_{t-1}, I_t, \hat{H}_{t-4}, \tilde{H}_{t-3}, \tilde{H}_{t-2}, \tilde{H}_{t-1}, \tilde{H}_t)$;

(b): input two non-neighbor frames as $(I_{t-8}, I_t, \hat{H}_{t-8}, \tilde{H}_t)$;

Corresponding performances are reported in Table 2. The information from multiple non-neighbor frames helps the network perform better.

frame intervals	[0, 30)	[0, 60)	[0, 90)	[0, 120)
Input (a)	3.82	10.64	19.77	30.94
Input (b)	7.19	23.11	47.66	80.59
w/o integral	2.49	6.47	52.76	83.64
w/o context	3.04	6.94	11.23	15.58
Ours(basic)	2.18	4.98	8.08	11.12
Ours(IHRM)	2.05	4.51	7.26	9.98

Table 2. Ablation study of input solutions, loss function and encoder. Each row (except the first column) reveal the tracking statistics on dataset ScanDPT in fMSE. The first row shows the frame intervals we used. The next two rows of statistics show performance of the alternative input solutions. Then the results without the integral loss function and without the context encoder are presented in the fourth and fifth row respectively. The next-to-last row shows the performance of the proposed basic network without IHRM. The last row shows the performance of whole network including IHRM.

Loss function: To validate the effectiveness of the integral loss, we tested the proposed network by replacing the integral loss Equation 8 with the heatmap loss, which supervised output heatmaps directly without integral regression:

$$L_{total} = \sum_{k=0}^{N} \gamma^{N-k} SmoothL1(H^k, H_{gt}) \qquad (12)$$

We observe that the network supervised by the integral loss performs much better.

Encoder: We used two encoders to extract location features and contextual features respectively. The location encoder is indispensable as it encodes the locations of four points in the previous frame. We removed the context encoder and kept other parts unchanged. As a result, the tracking performance without context encoder is inferior to that of the full model we proposed as the context encoder provides discriminative features of the surrounding edges for the designated planar region.

IHRM: To evaluate the effectiveness of IHRM, we added IHRM to the basic model. As shown in Table 2, the network with IHRM reaches 9.98 in fMSE and has lower fMSE by about 10% than that without IHRM when testing in the first 120 frames.

5. Conclusion

In this work, we propose a new network to tackle the practical problem, designated point tracking, for textureless planar regions. Existing template-based methods can only track the textured objects since the features are rare inside the template of textureless planar regions. On the other hand, with the input of full image as reference, existing methods do not well handle the prior local homography correlation of four designated points. Our network learns the prior correlation of the four designated points even with the full images in a supervised way. Given the initial heatmap prediction from optical flow method RAFT, our model predicts intermediate heatmaps by an dual-encoder structure and further refines them using an recursive module. Moreover, to train and evaluate our network, we present the first dataset ScanDPT mainly for textureless planar regions for the DPT task. Comparative experiments and ablation studies demonstrate the effectiveness of our network design and show the superiority of our method over other methods.

References

[1] M. Adorjan. *OpenSfM: A Collaborative Structure-from-Motion System*. PhD thesis, 2016. 3

[2] A. M. Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001. 3

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002. 2, 3

[4] S. Avidan. Support vector tracking. *IEEE transactions on pattern analysis and machine intelligence*, 26(8):1064–1072, 2004. 2, 3

[5] D. Barath, J. Matas, and J. Noskova. Magsac: marginalizing sample consensus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10197–10205, 2019. 3

[6] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 1, pages 943–948. IEEE, 2004. 2, 4

[7] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2010. 1, 3

[8] L. Chen, H. Ling, Y. Shen, F. Zhou, P. Wang, X. Tian, and Y. Chen. Robust visual tracking for planar objects using gradient orientation pyramid. *Journal of Electronic Imaging*, 28(1):013007, 2019. 2, 4

[9] L. Chen, F. Zhou, Y. Shen, X. Tian, H. Ling, and Y. Chen. Illumination insensitive efficient second-order minimization for planar object tracking. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4429–4436. IEEE, 2017. 2, 3, 4

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 4

[11] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021. 2, 3

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. 5

[13] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 3, 5

[14] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *ACM Transactions on Graphics 2017 (TOG)*, 2017. 3, 5

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[16] D. DeTone, T. Malisiewicz, and A. Rabinovich. Deep image homography estimation, 2016. 2, 4

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2, 3

[18] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1, 3

[19] U. Efe, K. G. Ince, and A. Alatan. Dfm: A performance baseline for deep feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4284–4293, 2021. 2, 3

[20] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008. 2, 4

[21] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[22] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International journal of computer vision*, 94(3):335, 2011. 3

[23] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 5

[24] D. Guo, Y. Shao, Y. Cui, Z. Wang, L. Zhang, and C. Shen. Graph attention tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2021. 2, 3

[25] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6269–6277, 2020. 2, 3

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[27] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE*

transactions on pattern analysis and machine intelligence, 37(3):583–596, 2014. 2, 3

[28] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 3

[29] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 1, 3

[30] W. Jiang, E. Trulls, J. Hosang, A. Tagliasacchi, and K. M. Yi. Cotr: Correspondence transformer for matching across images, 2021. 2, 3

[31] L. Jin, S. Qian, A. Owens, and D. F. Fouhey. Planar surface reconstruction from sparse views, 2021. 2

[32] T. Ke, T. Do, K. Vuong, K. Sartipi, and S. I. Roumeliotis. Deep multi-view depth estimation with predicted uncertainty, 2021. 2, 5

[33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. 6

[34] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019. 2, 3

[35] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018. 2, 3

[36] P. Liang, H. Ji, Y. Wu, Y. Chai, L. Wang, C. Liao, and H. Ling. Planar object tracking benchmark in the wild. *Neurocomputing*, 454:254–267, 2021. 3

[37] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab. A dataset and evaluation methodology for template-based tracking algorithms. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 145–151. IEEE, 2009. 3

[38] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019. 2

[39] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018. 2

[40] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2, 3

[41] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. Vancouver, British Columbia, 1981. 2, 3

[42] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. Vancouver, British Columbia, 1981. 3

[43] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan. Aslfeat: Learning local features of accurate shape and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6589–6598, 2020. 2, 3

[44] E. Mémin and P. Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 7(5):703–719, 1998. 1

[45] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2227–2240, 2014. 3

[46] O. Muratov, Y. Slynko, V. Chernov, M. Lyubimtseva, A. Shamsuarov, and V. Bucha. 3dcapture: 3d reconstruction for a smartphone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 75–82, 2016. 3

[47] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. IEEE, 2011. 3

[48] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3):2346–2353, 2018. 4

[49] V. A. Prisacariu, O. Kähler, S. Golodetz, M. Sapienza, T. Cavallari, P. H. S. Torr, and D. W. Murray. Infinitam v3: A framework for large-scale 3d reconstruction with loop closure, 2017. 3

[50] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang. Hedged deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4303–4311, 2016. 2, 3

[51] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017. 1, 3

[52] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International journal of computer vision*, 77(1-3):125–141, 2008. 2, 3

[53] A. Roy, X. Zhang, N. Wolleb, C. P. Quintero, and M. Jägersand. Tracking benchmark and evaluation for manipulation tasks. In *2015 IEEE international Conference on Robotics and Automation (ICRA)*, pages 2448–2453. IEEE, 2015. 3

[54] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 3

[55] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 1, 3

[56] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018. 5

[57] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020. 1, 2, 3, 4, 5

[58] N. Wang, W. Zhou, J. Wang, and H. Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1571–1580, 2021. 2, 3

[59] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison. Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems, 2015. 3

[60] J. Xu, R. Ranftl, and V. Koltun. Accurate optical flow via direct cost volume processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1289–1297, 2017. 3

[61] X. Yang, L. Zhou, H. Jiang, Z. Tang, Y. Wang, H. Bao, and G. Zhang. Mobile3drecon: real-time monocular 3d reconstruction on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3446–3456, 2020. 3

[62] J. Zhang, C. Wang, S. Liu, L. Jia, N. Ye, J. Wang, J. Zhou, and J. Sun. Content-aware unsupervised deep homography estimation. In *European Conference on Computer Vision*, pages 653–669. Springer, 2020. 4

[63] S. Zhang, X. Li, Y. Liu, and H. Fu. Scale-aware insertion of virtual objects in monocular videos. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 36–44. IEEE, 2020. 3

[64] Z. Zhang, F. Cole, R. Tucker, W. T. Freeman, and T. Dekel. Consistent depth of moving objects in video. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021. 3

[65] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, Y. Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6278–6287, 2020. 1, 3