3D Space-Time Correspondence using Normal-assisted Random Walk

Linyi Jin Changyuan Qiu Zhuowen Shen Yuliang Zhu
University of Michigan
{jinlinyi,peterqiu,mickshen,yuliangz}@umich.edu

1. Introduction

Humans have a remarkable ability to understand dynamic scenes from videos including estimating the 3D structure as well as finding correspondence at different time stamps. For example, when opening a fridge, one can track the door of the fridge and estimate the state and shape of the fridge. Many prior works [29, 20, 15] infer the 3D structure of a static scene from videos. Other lines of works [28, 35, 28, 11] solve the "what went where" [37] problem of dynamic scenes. However, estimating the 3D structure from videos of *dynamic* scenes is still a challenging problem. The classical method Structure from Motion (SfM) is limited to only static scenes [29, 30, 21]. In addition, video datasets with annotations for temporal visual correspondences are scarcely available and hard to create, making supervision a bottleneck for object tracking [14, 38].

In this work, we propose to estimate the 3D structure of dynamic scenes by combining a 3D detection system with a self-supervised visual temporal correspondence learning system. We reconstruct plane structures from video frames and track the reconstructed planes in dynamic scenes. The core methodology of this work is highly inspired by [11] from Jabri *et al*. We incorporate 3D planar surfaces information obtained from PlaneRCNN [17] of each frame into the framework of [11], which offers important geometric constraint for patch affinity prediction in dynamic scenes.

2. Related Works

Structure from Motion is a well-studied field that reconstructs 3D structures based on 2D images. However, prior works [31, 5, 36, 32, 29, 30, 21] are limited to static scenes and cannot handle the occlusions caused by dynamic objects. Visual SLAM systems [36, 25, 26] capture consistent room-scale and surface-based maps using RGB-D images or videos. They find point or plane correspondences across frames and use geometry to predict the camera location of each frame and a static 3D pointcloud. Aside from using sensor depthmaps, they also assume certain parts of the environment remain unchanged. Our approach, on the other hand, aims to use RGB frames only and work on dynamic scenes.

Recently, many works use deep learning methods to reconstruct 3D structure from images. There have been works to reconstruct normals [6, 34], voxels [3, 7], and depth [6, 24] from 2D images. However, single view 3D cannot guarantee consistency across video frames and cannot handle dynamic scenes. There are also efforts to estimate consistent depth from video frames using temporal correspondences [13, 22, 33, 41] or geometric correspondences [20]. Our approach differs from these works by estimating planes instead of depth, which is a higher level representation.

Our approach builds most heavily on works aiming to produce a planar reconstruction [18, 39, 17, 40, 2, 12]. In particular, we build on PlaneRCNN [17], which detects planes along with their geometric properties and instance masks from a single RGB image. Despite using surrounding frames to improve the detection quality, it only addresses the detection based on a single frame, and does not touch on the application when the temporal correspondence of the targeted image sequence needs to be learned.

[35, 11] aims to learn useful visual representations for visual correspondence from raw videos without any supervision. Recently, [10] learns pixel-wise correspondence in a weakly-supervised approach by using epipolar constraints. Our work builds on [11] which enforces strong affinity between image patches by learning embedding vectors to guide a random walk across a palindrome of frames. However, the original work suffers from very strict formulation to prevent the network from finding shortcuts during learning. Our approach adds plane normal aside from the embedding vectors to construct the affinity between nodes of adjacent frames. Our hope is that adding extra constraints can help the system to find robust planar correspondence across frames.

3. Method

3.1. Normal-assisted Random Walk

Following [11], we represent each video as a directed graphs with nodes being patches of each frame and edges connect nodes in neighboring frames. Let $\mathbf{I}$ be a set of

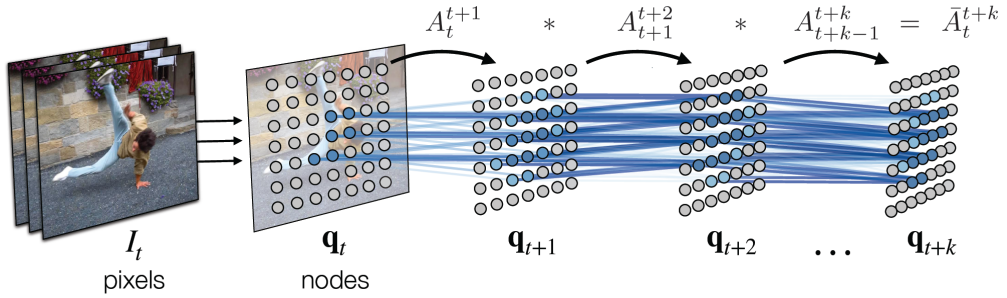


Figure 1: Figure credit [11]. **Correspondence as a Random Walk.** We build a space-time graph by extracting nodes from each frame and allowing directed edges between nodes in neighboring frames. The transition probabilities of a random walk along this graph are determined by the product of 2D pairwise similarity in a learned representation and 3D distance of normal vectors detected by a PlaneRCNN network.

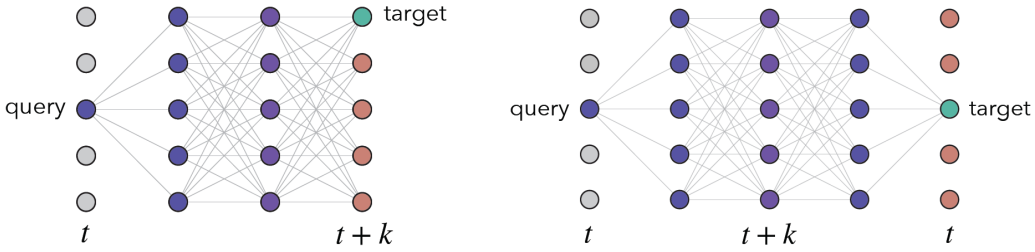


Figure 2: Figure credit [11]. **Guiding the walk with palindrome self-supervision.** (a) Specifying a target multiple steps in the future provides implicit supervision for latent correspondences along each path (*left*). (b) We can construct targets for free by choosing palindromes as sequences for learning (*right*).

frames from a video and $\mathbf{q}_t$ be the set of N nodes extract from frame I_t. We train an encoder ϕ which maps nodes to l_2-normalized $d-$dimensional vectors, which we use to compute a pairwise 2D similarity function between nodes $d_\phi(q_1, q_2) = \langle \phi(q_1), \phi(q_2) \rangle$ and an 2D embedding matrix for $\mathbf{q}_t$ denoted $Q_t \in \mathbb{R}^{N \times d}$. We convert pairwise similarities into non-negative affinities by applying a softmax (with temperature $\tau = 0.07$) over edges departing from each node. For timesteps t and $t+1$, the stochastic matrix of 2D affinities is

$$A_t^{t+1}(i,j) = \texttt{softmax}(Q_t, Q_{t+1}^{\mathrm{T}})_{ij} = \frac{\exp(d_\phi(\mathbf{q}_t^i, \mathbf{q}_{t+1}^j)/\tau)}{\sum_{l=1}^{N} d_\phi(\mathbf{q}_t^i, \mathbf{q}_{t+1}^l)/\tau)} \quad (1)$$

Our modification applies here that we add 3D normal vector distance (normal vector prediction process are detailed in Section 3.2) aside from the 2D embedding affinities to the stochastic matrix. Let $\mathbf{n}_t$ be the set of N normal vector extracted from frame $\mathbf{I}_t$, we construct our modified stochastic matrix by multiplying the raw 2D affinity probability $A_t^{t+1}(i,j)$ with the 3D normal eculidean distance $||\mathbf{n}_t^i - \mathbf{n}_{t+1}^j||^2$, and we further apply a softmax to obtain a probability. For timesteps t and $t+1$, our modified stochastic matrix is

$$A_t^{t+1\prime}(i,j) = \frac{\exp(A_t^{t+1}(i,j) \cdot ||\mathbf{n}_t^i - \mathbf{n}_{t+1}^j||^2)}{\sum_{l=1}^{N} \exp(A_t^{t+1}(i,j) \cdot ||\mathbf{n}_t^i - \mathbf{n}_{t+1}^l||^2)} \quad (2)$$

Note that this describes only the local affinity between the patches of two video frames, $\mathbf{q}_t$ and $\mathbf{q}_{t+1}$, and we relate all nodes in the video as a Markov chain following [11]. Given the spatio-temporal connectivity of the graph, a step of a random walker on this graph can be viewed as performing tracking by *contrasting* similarity of neighboring nodes. Let X_t be the state of the walker at time t, with transition probabilities $A_t^{t+1\prime}(i,j) = P(X_{t+1} = j | X_t = i)$, where $P(X_t = i)$ is the probability of being at node i at time t. And we can formulate long-range correspondence as walking multiple steps along the graph (Figure 1):

$$A_t^{t+k} = P(X_{t+k}|X_t) = \prod_{i=0}^{k-1} A_{t+i}^{t+i+1} \quad (3)$$

Guiding the walk with palindrome self-supervision. Our aim is to train the encoder ϕ to encourage the random walker to follow paths of corresponding patches as it steps through time. Following [11], we train our model with a cross-entrophy loss on the *palindrome* self-supervised training examples. To formalize, given a sequence of frames $(I_t, \ldots, I_{t+k})$, we form training examples by simply concatenating the sequence with a temporally reversed version of itself: $(I_t, \ldots, I_{t+k}, \ldots, I_t)$. Treating each query node's position as its own target (as shown in Figure 2), we obtain the following cycle-consistency cross-entropy objective which maximize the likelihood that a walker beginning at a query node at t ends at the same target node at time $t+2k$:

$$\mathcal{L}_{cyc}^{k} = \mathcal{L}_{CE}(A_t^{t+k} A_{t+k}^{t}, I) = -\sum_{i=1}^{N} \log P(X_{t+2k} = i | X_t = i) \tag{4}$$

As the model computes a soft attention distribution at every time step, we can backpropagate loss across – and thus learn from – the many alternate paths of similarity that link query and target nodes.

Pixel to nodes. We sample 64×64 patches on a 640×480 image with stride of 32 to allow overlapping. Following [11] and [19], we reuse the convolutional feature map between patches obtained from training for testing instead of processing the patches independently and thus features could be extracted with only a single feed-forward pass.

Encoder ϕ. We use ResNet-18 [9] for the encoder which outputs a 128-dimensional vector for each patch. We apply a linear projection and l_2 normalization after the last average pooling layer and modifies the last 2 `res3` and `res4` layer to have stride of 1 instead of 2 following [11].

3.2. Normal Prediction for Patches

Plane detection. We follow PlaneRCNN [17] to detect planes in each frame. Instead of using the whole pipeline for prediction, we only use the plane detection network in [17] and ignore the segmentation refinement network and warping loss module for simplicity. The plane detection network uses ResNet50-FPN [16] as the backbone. A region proposal network is used to extract box proposals. The plane masks and normals are inferred from features from RoIAlign [8]. The PlaneRCNN is supervised by ground truth label on ScanNet generated by fitting planes on the original house mesh provided by [18].

Patch normal assignment. For each patch, we calculate the GIoU [27] between that patch and each bounding box of the plane masks and assign the normal vector of the mask with the highest GIoU.

3.3. Training Details

Our Normal-assisted Random Walk network is trained on the Scannet training set, using one Tesla V100 GPU on Colab with learning rate 0.0001, batch size 4 and Adam optimizer with default parameters. The model is trained for 25 epochs.

Our PlaneRCNN network is implemented in Detectron2 and is trained on the Scannet training set, using 2 GPUs with learning rate 0.001, batch size 16, and SGD optimizer with momentum 0.9. The model is trained for 80k iterations.

4. Experiments

4.1. Dataset

We use ScanNet [4] to train and evaluate our model for tracking 3D planes. ScanNet [4] is an indoor RGB-D video dataset with surface reconstructions and instance segmentation annotations. The instance indices are consistent throughout the video frames which enables the evaluation for 3D object tracking. We downsample the image resolution from 1296×968 to 640×480 using bilinear interpolation. The instance masks are resized with the same ratio except that the nearest-neighbour interpolation is used to avoid introducing new instance indices. We use 10 scenes for training and 2 scenes for testing. Each scene contains 1000 to 5000 frames. See Figure 4 and 5 for illustrations.

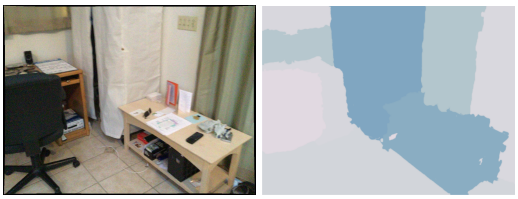
4.2. Evaluation Metric

We follow [11] and two complementary criteria are employed to measure how our mask prediction M fits the ground truth mask G.

Region Similarity J. To measure the region-based segmentation similarity, the commonly used metric IoU (intersection-over-union, also known as the Jaccard index) is used. J can be calculated as $J = \left| \frac{M \cap G}{M \cup G} \right|$.

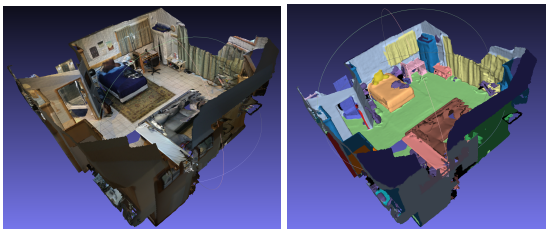
Contour Accuracy F. Precision P_c and recall R_c can be calculated based on the contours extracted from M and G. Then the F-measure F can be calculated as $F = \frac{2 P_c R_c}{P_c + R_c}$ to achieve a good trade-off between the two.

We follow [11] to report mean (m) and recall (r) of the region similarity (J) and boundary alignment (F) [23].



(a) Scene Example (b) Instance Segmentation

Figure 4: Image Mask Pair [4]



(a) Surface Reconstruction (b) Labeled Mesh

Figure 5: Mesh Model [4]

Baselines. We use two baseline models to compare with our proposed method. All models share the same network architecture. The first baseline is the pretrained model provided by [11], which only uses the self-supervised palindrome loss and is trained on the Kinetics400 [1]. The second baseline is trained on a subset of the ScanNet scenes from scratch but without considering the surface normal constraints in the affinity matrix and the random-walk loss. Our model in contrast, is trained on the same subset of ScanNet from scratch and takes the normal difference into consideration as described in section 3.

Method	Resolution	Train Data	$J\&F_m$	J_m	J_r	F_m	F_r
Pretrained [11]	256 × 256	Kinetics	72.74	73.92	74.76	71.56	69.56
Ours w/o normal	640×480	ScanNet	73.08	74.34	**74.87**	71.83	70.07
Ours w/ normal	640×480	ScanNet	**73.14**	**74.35**	74.81	**71.93**	**70.17**

Table 1: Video object segmentation results on ScanNet

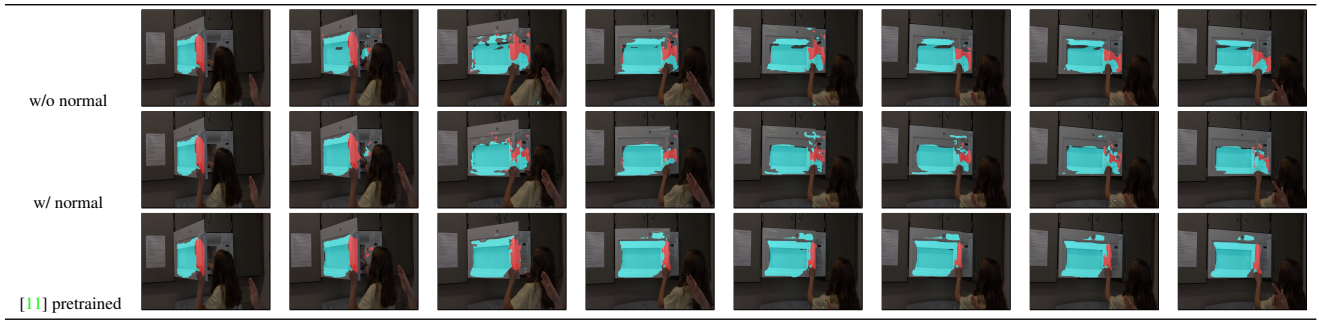
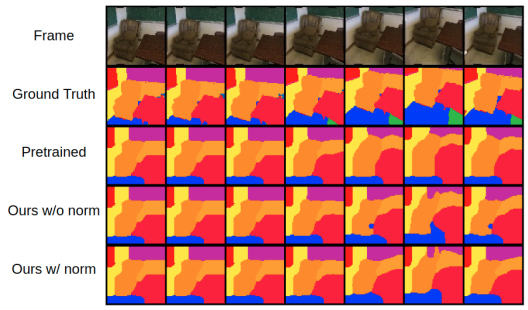
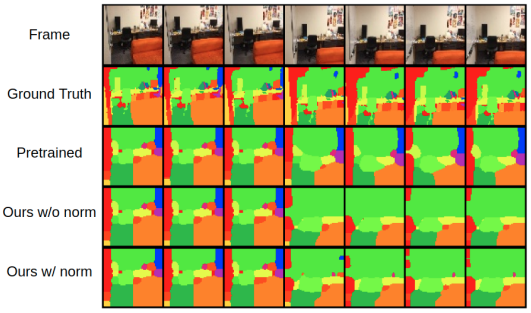


Figure 3: Qualitative results on videos with dynamic scene of the label propagation task given an initial plane detection from PlaneRCNN model.



(a) Test scene A



(b) Test scene B

Figure 6: Test Scenes

4.3. Qualitative Results

We present some inference results from the two baseline models as well as our model, see Figure 6. The inference results are the instance segmentation masks of the scene, where the instance indices are consistent throughout the frames. We select two new scenes from the ScanNet dataset that are different from the training data in order to test the generalization ability of our model.

As shown in Figure 6 a, our model with normal constraints can have denoising and smoothing effects on the plane detection (e.g. 5th column) compared with the model without normal constraints. However, it can also be subject to noises. As can be observed in Figure 6 b, a small object with regional salient normal can induce noisy plane detection compared to the pretrained model (e.g. 7th column).

We also evaluate the methods on a dynamic scene, and the results are shown in Figure 3. All the three methods cannot predict the planes for the door and the handle perfectly, underlining the improvement space for our model on dynamic scenes.

4.4. Quantitative Results

Our quantitative results are listed in table 1. Unsurprisingly, our model trained on the ScanNet [4] outperforms the pretrained model [11] on the test scenes because of the data distribution similarity. Our model with normal constraints slightly improves the model without normal constraints with limited training. This approach is promising if more training resources are available.

5. Conclusions

While existing 3D objection detection methods are capable of handling difficult cases such as dynamic scenes, occlusion, and deformation, the prediction between frames are usually independent and thus is incapable of evaluating the temporal correspondence. We propose a new framework that builds on the video random-walk model [11], and integrates 3D information constraints into the affinity matrix and the loss function. Under our formulation, we observe slight improvement with the surface normal constraint with limited training. However, this approach can be subject to small and irregular planes and lead to noisy predictions. In the future work, this approach can be more thoroughly evaluated with more training, and other types of constraints can be applied such as plane texture.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 3

[2] Weifeng Chen, Shengyi Qian, David Fan, Noriyuki Kojima, Max Hamilton, and Jia Deng. Oasis: A large-scale dataset for single image 3d in the wild. In *CVPR*, 2020. 1

[3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 1

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 3, 4

[5] Ashwin P Dani, Nicholas R Fischer, and Warren E Dixon. Single camera structure and motion. In *TACON*, 2011. 1

[6] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 1

[7] R. Girdhar, D.F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 1

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[10] Zhaoyang Huang, Xiaokun Pan, Runsen Xu, Yan Xu, Ka chun Cheung, Guofeng Zhang, and Hongsheng Li. Life: Lighting invariant flow estimation. *arXiv*, 2021. 1

[11] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 1, 2, 3, 4

[12] Ziyu Jiang, Buyu Liu, Samuel Schulter, Zhangyang Wang, and Manmohan Chandraker. Peek-a-boo: Occlusion reasoning in indoor scenes with plane representations. In *CVPR*, 2020. 1

[13] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. In *TPAMI*, 2014. 1

[14] Xueting Li, Sifei Liu, Shalini de Mello, Xiaolong Wang, Jan Kautz, and Ming Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*, 2019. 1

[15] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *CVPR*, 2019. 1

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[17] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *CVPR*, 2019. 1, 3

[18] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *CVPR*, 2018. 1, 3

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 3

[20] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. In *SIGGRAPH*, 2020. 1

[21] Kemal E Ozden, Konrad Schindler, and Luc Van Gool. Multibody structure-from-motion in practice. In *TPAMI*, 2010. 1

[22] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. In *RA-L*, 2020. 1

[23] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 3

[24] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. In *TPAMI*, 2020. 1

[25] Carolina Raposo and Joao P Barreto. πMatch: Monocular vslam and piecewise planar reconstruction using fast plane correspondences. In *ECCV*, 2016. 1

[26] Carolina Raposo, Miguel Lourenço, Michel Antunes, and João Pedro Barreto. Plane-based odometry using an rgb-d camera. In *BMVC*, 2013. 1

[27] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. June 2019. 3

[28] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1

[29] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[30] Gideon Schwarz et al. Estimating the dimension of a model. In *Annals of statistics*, 1978. 1

[31] Sudipta N Sinha, Drew Steedly, and Richard Szeliski. A multi-stage linear approach to structure from motion. In *ECCV*, 2010. 1

[32] Mohamed Tamaazousti, Vincent Gay-Bellile, Sylvie Naudet Collette, Steve Bourgeois, and Michel Dhome. Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. In *CVPR*, 2011. 1

[33] Rui Wang, Stephen M Pizer, and Jan-Michael Frahm. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *CVPR*, 2019. 1

[34] Xiaolong Wang, David F. Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015. 1

[35] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 1

[36] Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. In *RSS*, 2015. 1

[37] Josh Wills, Sameer Agarwal, and Serge J. Belongie. What went where. In *CVPR*, 2013. 1

[38] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 1

[39] Fengting Yang and Zihan Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *ECCV*, 2018. 1

[40] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *CVPR*, 2019. 1

[41] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *ICCV*, 2019. 1