Perceiving 3D Humans and Objects in Motion

Zhe Cao

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-25
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-25.html

May 1, 2022

Perceiving 3D Humans and Objects in Motion

by

Zhe Cao

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jitendra Malik, Chair
Professor Angjoo Kanazawa
Professor Ken Goldberg

Summer 2021

Perceiving 3D Humans and Objects in Motion

Abstract

Perceiving 3D Humans and Objects in Motion

by

Zhe Cao
Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley
Professor Jitendra Malik, Chair

We exist in a 3D world, where we accomplish everyday tasks by perceiving and interacting with other people and objects in dynamic scenes. Could we develop a perception system to understand such rich interactions? This is crucial for future intelligent systems to collaborate with humans and to create immersive AR/VR experiences. While great progress has been achieved in the individual perception tasks of 3D humans, objects, and scenes, the connections between these components have not been explored much. In this thesis, we attempt to build the connections between these three components to understand their rich interactions.

We start by bridging the scene and object component in Chapter 2, where we present an end-to-end learning system to perceive 3D scene and independent object motions. We next show how 3D scenes influence human motion in Chapter 3, where we design a framework to predict future 3D human motion considering the scene context. In Chapter 4, we study the interaction between human hands and objects, where we introduce an optimization-based method to reconstruct the interaction in the wild. Finally, we conclude with several interesting future directions.

Professor Jitendra Malik
Dissertation Committee Chair

To my parents for their love and support.

Contents

Acknowledgments

This dissertation is a product of the affection and guidance of a great many excellent people including but not limited to my advisor, collaborators, and colleagues. None of this work would have been possible without their contributions and continual support. First and foremost, I would like to thank my advisor, Jitendra Malik, for teaching me how to think about the big picture, how to choose a meaningful problem, and how to pursue impactful research. I really appreciate him giving me the freedom to pursue my own interests while giving numerous helpful suggestions throughout the process. I am grateful for all the valuable history lessons I have learned from him throughout the years. Thanks to Alyosha Efros for showing me the passion for teaching the undergraduate class, it was a lot of fun to design the new assignment together. I also want to thank my qualification and thesis committee members including Angjoo Kanazawa, Ken Goldberg, Bruno Olshausen, Anca Dragon, and Trevor Darrell for their valuable feedback on my research.

I am very fortunate to be surrounded by many talented and empathetic people during my time at Berkeley. Thanks to my co-authors - Hang Gao who has been my great friend and taught me the importance of considering the details, Ilija Radosavovic who taught me the value of being organized and always gave me helpful suggestions over plenty of discussions, Abhishek Kar and Christian Häne for deepening my knowledge in 3D vision research. Thanks to all amazing peers and colleagues in Malik and Efros groups: Karttikeya Mangalam, Shubham Tulsiani, Shubham Goel, Yu Sun, Ashish Kumar, Allan Jabri, Haozhi Qi, Jasmine Collins, Sasha Sax, Weicheng Kuo, Ke Li, Georgios Pavlakos, Pulkit Agrawal, Saurabh Gupta, Anastasios Angelopoulos, Panna Felsen, Jason Zhang, Jeffrey O. Zhang, Tinghui Zhou, Jun-Yan Zhu, Tim Brooks, Bill Peebles, Shiry Ginosar, Deepak Pathak, Taesung Park, David Fouhey, Andrew Owen, Xiaolong Wang, Lerrel Pinto, and others, for all the discussions in the lab. My ideas and opinions have been greatly shaped by the discussions and I have learned immensely from them. Thanks to my many other Berkeley friends for sharing the journey with me, including Zhuang Liu, Shizhan Zhu, Dequan Wang, Tete Xiao, Yang Gao, Huazhe Xu, Xin Wang, Guanhua Wang, and others. I am thankful to Angie Abbatecola for always being there to help and support the group.

Before coming to Berkeley, I have the great fortune to work with Yaser Sheikh when I was a master student at CMU. I am thankful for his valuable guidance and unequivocal support during the beginning of my research journey. I am grateful for the wonderful experience that I had with other group members: Natasha Kholgade Banerjee, Varun Ramakrishna, Tomas Simon, Shih-En Wei, Gines Hidalgo, Minh Vo, Aayush Bansal, and Hanbyul Joo. My life in CMU has been made so enjoyable with the support from my amazing group of friends, including Mengtian Li, Zehua Huang, Haoqi Fan, Wenhao Luo, Wei-Chiu Ma, Chen-Hsuan Lin, Tianyu Gu, Zhiding Yu, Rui Zhu, Xiaofang Wang, Yu Zhang, Yiying Li, and Chao Liu.

I also enjoyed my two-time internships at Facebook. I am thankful for the

general support and helpful feedback from the FRL Sausalito team including Min Vo, Carsten Stoll, Christoph Lassner, Tony Tung, Ronald Mallet, and others. Thanks to Georgia Gkioxari for hosting me in the Menlo Park office, and Kaiming He, Piotr Dollar for their helpful suggestions. I would also like to thank all my fellow interns for making both internships an incredible experience, including Yi Wu, Hang Zhao, Zeng Huang, Tiancheng Zhi, and many others.

Thanks to my group of friends at distance - Qingqing Cao, Yanghua Peng for motivating me and keeping me going when my research hits roadblocks. I want to thank Ophelia for always believing in me and helping me become a better human being over these years. Finally, I would like to thank my family who are the most important people in my life: my father, mother, and brother. Thank you for giving me the freedom to pursue my dreams, teaching me the immense value of helping others, and giving me boundless love.

Chapter 1

Introduction

We humans exist in a 3D world, where we accomplish everyday tasks by perceiving and interacting with other moving agents in the 3D dynamic scene. Consider the outdoor scenario in Figure 1.1, we humans have the remarkable capability to infer the 3D structure of the environmisent, such as the floor plan, the building, and how far each of these elements in the scene is from us. With temporal frames, we can perceive the nearby moving object and people in terms of their moving speed and direction in this environment. Moreover, we can perceive the rich human-object interactions such as riding a bicycle or pushing a cart. All this perceived 3D information enables us to predict the environment state in the near future and to plan our next actions.

Could we equip future intelligent agents with similar capabilities to perceive and meaningfully interact with the 3D world? There are three main components to consider in such a perception system: perceiving 3D scenes, humans, and objects. In recent years, we have seen large progress in each of the perception tasks [36, 57, 172]. However, these problems are not isolated in many cases. For example, the 3D scene layout will constrain the possible human and object motion inside the environment. By jointly considering multiple components, we can impose additional constraints to obtain more natural and feasible results. This thesis attempts to bridge those three perception tasks, as shown in the triangle in Figure 1.1, and demonstrate the benefits of building the connection in different applications.

We begin in Chapter 2 by briding the scene and object components, we present a system for learning motion maps of independently moving objects from stereo videos. The only annotations used in our system are 2D object bounding boxes which introduce the notion of objects in our system. Unlike prior learning-based approaches which have focused on predicting dense optical flow fields and/or depth maps for images, we propose to predict instance specific 3D scene flow maps and instance masks from which we derive a factored 3D motion map for each object instance. Our network takes the 3D geometry of the problem into account which allows it to correlate the input images and distinguish moving objects from static ones. We present experiments evaluating the accuracy of our 3D flow vectors, as well as depth maps

Figure 1.1: *Left:* an example outdoor scenario where people interacting with the 3D scene and objects. *Right:* the overall structure of the thesis.

and projected 2D optical flow where our jointly learned system outperforms earlier approaches trained for each task independently.

In Chapter 3, we consider the scene and human components jointly, where we present a framework to predict the future human motion considering the scene context. Human movement is goal-directed and influenced by the spatial layout of the objects in the scene. To plan future human motion, it is crucial to perceive the environment – imagine how hard it is to navigate a new room with lights off. Existing works on predicting human motion do not pay attention to the scene context and thus struggle in long-term prediction. We instead introduce a novel three-stage framework that exploits scene context to tackle this task. Given a single scene image and 2D pose histories, our method first samples multiple human motion goals, then plans 3D human paths towards each goal, and finally predicts 3D human pose sequences following each path. For stable training and rigorous evaluation, we contribute a diverse synthetic dataset with clean annotations. We show our method shows consistent quantitative and qualitative improvements over existing methods.

In Chapter 4, we study the problem of understanding hand-object interactions from 2D images in the wild. This task requires reconstructing both the hand and the object in 3D, which is challenging because of the mutual occlusion between the hand and the object. We present a novel reconstruction technique that reconstructs 3D poses of both the hand and the object with the help of 2D image cues and 3D contact priors. Moreover, we contribute a dataset MOW (Manipulating Objects in the Wild) of 500 examples of hand-object interaction images that have been "3Dfied" with the help of the RHO technique together with human intervention. Our dataset contains 121 distinct object categories, with a much greater diversity of manipulation actions, than in previous 3D hand-object datasets.

We conclude the thesis with a discussion on the limitations of current systems and promising future directions of perceiving 3D humans and objects in motion.

Chapter 2

Perceiving 3D Scene and Object Motion

Consider the crowded road scene in Figure 2.1, what information do we as humans use to navigate effectively in this environment? We need to have an understanding of the structure of the environment, i.e. how far other elements in the scene (cars, bikes, people, trees) are from us. Moreover, we also require knowledge of the speed and direction in which other agents in the environment are moving relative to us. Such a representation, in conjunction with our ego-motion, enables us to produce a hypothesis of the environment state in the near future and ultimately allows us to plan our next actions.

In order to gather this information, humans use stereo-motion, i.e. a stream of images captured with our two eyes as we move through the environment. In this chapter, we develop a computational system that aims to produce such a factored scene representation of 3D structure and motion from a binocular video stream. Specifically, we propose to predict the 3D object motion of each moving object (represented by 3D scene flow) in addition to a detailed depth map of the scene from a stereo image sequence. This task and its variants have been tackled in supervised settings which require labels such as dense depth maps and motion annotations that are prohibitively expensive to collect or alternatively obtained from synthetic datasets [22, 25, 50, 60, 88]. We present a system that learns to predict these quantities using only unlabelled stereo videos, thus making it applicable at scale. In addition to producing pixel-wise depth and scene flow maps, our network is aware of the notion of independent objects. This allows us to produce a rich factored 3D representation of the environment where we can measure velocities of independent objects in addition to their 3D positions in the scene. The only labels used by our system are those introduced by off-the-shelf object detectors which are very cheap to acquire at scale.

Prior work in this domain has focused on certain sub-problems such as learning depth or optical flow prediction without explicit labels [165, 37, 30]. In Section 2.4,

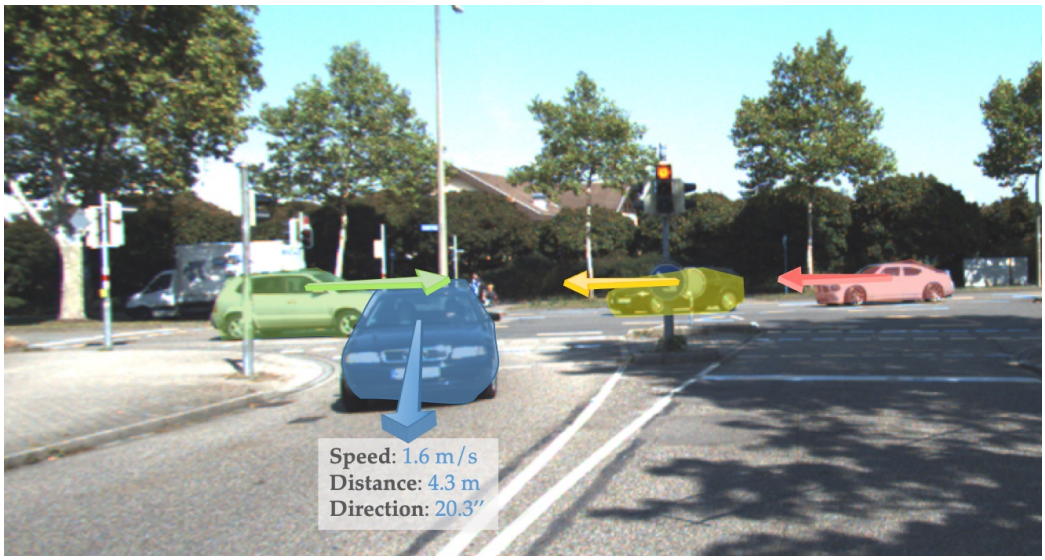


Figure 2.1: Object motion predicted by our system. Trained with raw stereo motion sequences in a self-supervised manner, our model learns to predict object motion together with the scene depth using sequence of stereo images and object proposals as input. The speed and moving direction of each moving object is derived from our scene flow prediction.

we demonstrate that by jointly learning the full problem of depth and scene flow prediction, we outperform these methods for each of these sub-problems as well. The key contributions of our work are as follows: (1) formulating a learning objective which works with the limited amount of supervision that can be gathered in a real world scenario (object bounding box annotations), (2) factoring the scene representation into independently moving objects for predicting dense depth and 3D scene flow and (3) designing a network architecture that encodes the underlying 3D structure of the problem by operating on plane sweep volumes.

The sections in this chapter are organized as follows. Section 2.1 discusses prior work on inferring scene structure and motion. Section 2.2 presents our technical approach for inferring scene flow from stereo motion - loss functions, object-centric prediction and priors. In Section 2.3, we describe our network architecture designed for geometric matching and 3D reasoning in plane sweep volumes. Section 2.4 details our experiments on the KITTI dataset [91] with extensive evaluation of our depth and scene flow prediction.

2.1 Background

In this chapter, we recover scene geometry and object motion jointly while traditionally these problems have been solved independently. The geometry of a scene is reconstructed by first recovering the relative camera pose between two or more im-

ages taken from different viewpoints using Structure-from-Motion (SfM) techniques [82, 40]. Subsequently, with dense matching and triangulation a dense 3D model of the scene is recovered [105]. The underlying assumption within the aforementioned methods is that the scene is static, i.e. does not contain moving objects. The case for independently moving objects has been studied in a purely geometric setting [19]. The key difficulties are degenerate configurations and outliers in point correspondences [98]. Therefore additional priors are used - a common example is objects moving on a ground plane [167]. Similarly, estimating the shape of non-rigid objects is ambiguous and hence using additional constraints such as maximizing the rigidity of the shape [140] or representing the non rigid shape as linear combination of base shapes [11] have been proposed. When reconstructing videos captured in unconstrained environments additional difficulties such as incomplete feature tracks and bleeding into the background have to be handled [27]. Our proposed approach is trained on real world data which makes it robust to appearance variations and suitable priors are directly learned from data.

Vedula *et al.*. [143] introduced the problem of 3D scene flow estimation, where for each point a 3D motion vector between time t and $t+1$ is computed. Different variants are considered depending on the amount of 3D structure that is given as input. A common variant is to consider a stream of binocular image pairs of a moving camera as input [49, 156, 147, 91, 131], and give a depth and 3D scene flow as output. This is often referred to as the stereo scene flow estimation problem. Similarly RGBD scene flow considers a stream of RGBD (color and depth) images as input [52].

Recently learning-based approaches, especially convolutional neural networks have been applied for single view depth prediction [70, 22], optical flow [25], stereo matching and scene flow [88]. These learning systems are trained using ground truth geometry and/or flow data. In practice such data is only available for synthetic data in a large scale. A natural way to complement the limited amount of ground truth data is using weaker supervision. For the aforementioned problems, loss functions which are purely based on images and rely on photometric consistency as learning objective have been proposed [30, 172, 37, 137, 144]. They essentially utilize a classical non-learned system [28] within the loss function. A few recent works [165, 177, 162, 83, 110] use such a self-supervised approach to predict optical flow and depth. To our knowledge our work is the first network that learns to directly predict object specific 3D scene flow without relying on pixel-wise flow or depth annotations.

Another key difference of our work from prior works that predict depth and optical flow is that they predict depth based on a single image. This limits their performance as demonstrated in our results. Geometric reasoning can be included into the network architecture as demonstrated in [60, 58, 54, 163]. We extend these ideas to full 3D scene flow estimation while also operating at the level of object instances allowing us to produce rich factored geometry and motion representation of the scene.

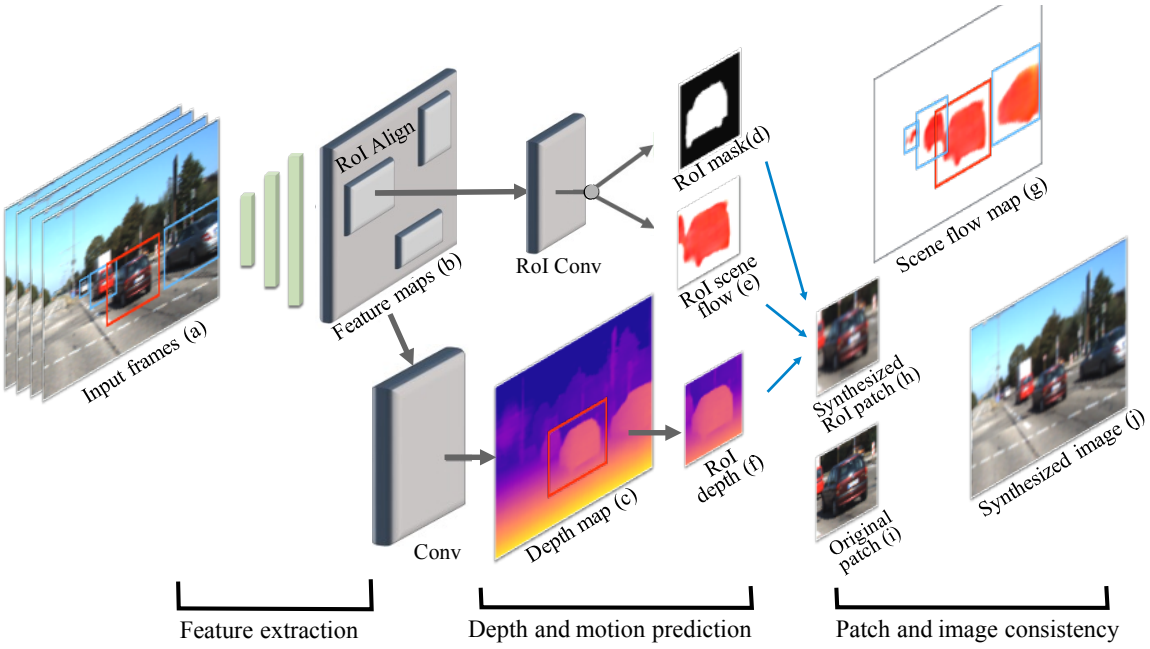


Figure 2.2: Our pipeline for learning depth and object motion. Using a stereo motion sequence as input, our system predicts a depth map (c), instance mask (d) and 3D scene flow (e) for each independent moving object in a single forward pass. Using the instance mask and scene flow, we compose a full scene flow map (g). For each region of interest (RoI), we synthesize a patch (h) based on the RoI camera intrinsics, RoI depth (f), 3D scene flow (e) and instance mask (d) as explained in Section 2.2.2. We use the synthesized patch (h) and original patch (i) from the input image to enforce consistency losses to supervise the RoI prediction. We use stereo reprojection to supervise the depth prediction. Finally, we use the full map scene flow and depth to synthesize a image (j) for computing the consistency loss.

2.2 Scene Flow from Stereo Motion

Figure 2.2 illustrates our system. A stream of calibrated binocular stereo image pairs $\mathcal{I} = \{I_1^l, I_1^r, \ldots I_n^l, I_n^r\}$ captured from times 1 to n is given as input. The most common case we are investigating is $n = 2$, i.e. two binocular frames at time t and t+1. The intrinsic camera calibration K is assumed to be known. The camera poses of the left camera at each time instant are denoted by $\mathcal{T} = \{T_1, \ldots, T_n\}$ and are precomputed using visual SLAM [32]. For any time instant t, we also have a set of j 2D bounding box detections $\mathcal{B} = \{B^1, \ldots, B^j\}$ on the left image I_t^l predicted by an off-the-shelf object detector. The task is to compute the following quantities for the reference frame - a dense depth map D, a set of dense 3D flow fields $\mathcal{F} = \{F^1, \ldots, F^j\}$ that describe the motion between t and $t + 1$ and a set of instance masks $\mathcal{M} = \{M^1, \ldots, M^j\}$ for each moving object. From these instance-level predictions, we can compose the full scene flow map F by assigning a 3D scene flow vector to each image pixel in the full image.

We design our system as a convolutional neural network (CNN) which learns

to predict all quantities jointly and train the network in a self-supervised manner. The supervision comes from the consistency between synthesized images and input images at different time instants and from different camera viewpoints. The basic principle is that given the predictions of the scene flow F and depth D in a frame I_{ref}, we can use the precomputed ego-motion to warp another image I into the reference view. This process generates a synthesized image which we call $\hat{I}$. We then define our learning objective as the similarity between the captured images I_{ref} and the synthesized images $\hat{I}$. The above principle is then applied to each region of interest (RoI) independently followed by an assembly procedure for full image scene flow. This allows us to produce a factored representation of the environment into static and dynamic objects with high-quality estimates of instance masks, depth and motion.

2.2.1 Disentangling Camera and Object Motion

The motion in a dynamic scene captured by a moving camera can be decomposed into two elements - the motion of static background resulting from the camera motion and the motion of independently moving objects in the scene. A common way to represent the scene motion is 2D optical flow. However, this representation confounds the camera and object motion. We model the motion of the static background using the 3D structure represented as a depth map and the camera motion. Dynamic objects are modelled with full 3D scene flow. To this end, we utilize 2D object detections in the form of bounding boxes and reason about the 3D motion of each object independently.

2.2.2 Supervising Scene Flow by View Synthesis

The key supervision for the scene flow prediction comes from the photometric consistency of multiple views of the same scene. The process is illustrated in Figure 2.3. Our network predicts a depth map D and a scene flow map F for the reference view I_{ref}. Using a different image I we can use the predictions to warp I into the reference view and generate a synthesized image $\hat{I}$. We then minimize the photometric difference between I_{ref} and $\hat{I}$ given as

$$\mathcal{L}_{\text{photo}} = \alpha \frac{1 - \text{SSIM}(I_{\text{ref}}, \hat{I})}{2} + (1-\alpha)\|I_{\text{ref}} - \hat{I}\|_1 \tag{2.1}$$

where SSIM denotes the structural similarity index [155] and α denotes a weighting parameter.

We denote the homogeneous coordinates of pixel p as $h(p)$. A pixel p from the reference frame is transformed to a pixel $\hat{p}$ within a frame I

$$h(\hat{p}) = KT_{\text{rel}}(D(p)K^{-1}h(p) + F(p)) \tag{2.2}$$

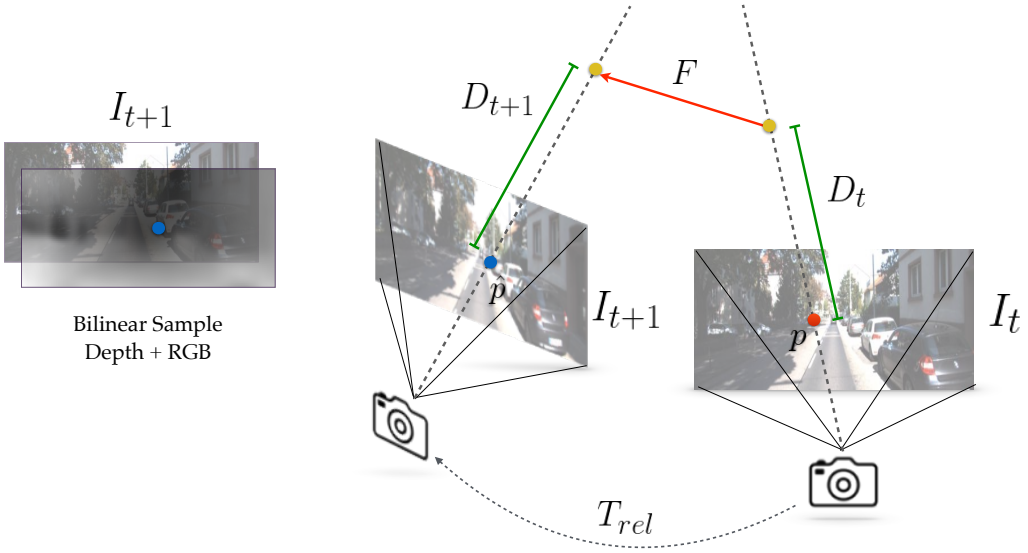


Figure 2.3: Illustration of our image reprojection process. A pixel p from image I_t is unprojected using its predicted depth and subsequently transformed to the frame of I_{t+1} using the predicted flow F and the camera transform T_{rel}. The photometric consistency loss is derived from the photometric difference between I_t and $\hat{I}_{t+1\to t}$ where $\hat{I}_{t+1\to t}$ is created by warping I_{t+1} into I_t. The geometric consistency loss is computed by comparing the difference between depth maps warped in the above manner and having them consistent with the z-dimension of the predicted flow F. Note that using only photometric consistency would not resolve the ambiguity in the z direction of the flow.

with T_{rel} the relative transformation from reference frame to I. This allows us to do a reverse warp using bilinear interpolation, keeping the formulation differentiable.

Using the photometric consistency alone is insufficient for supervising the 3D flow prediction. The reason is that along a viewing ray multiple photo consistent solutions are possible, as shown in Figure 2.3. Therefore we use an additional geometric loss leveraging depth consistency which further constrains the flow. The idea is that the flow in z-direction, sometimes also called disparity difference has to agree with the depth maps predicted for the two time instants t and $t+1$. In order to utilize this loss function a depth map for both time instants needs to be predicted and the warping is applied to the depth map.

Analogous to the photometric consistency, the geometric consistency is defined by comparing the predicted depth values of the warped image and reference image,

$$\mathcal{L}_{\text{geo}} = \left\| D_{\text{ref}} - \hat{D} + F_z \right\|_1 \tag{2.3}$$

where D_{ref} refers to the predicted depth at time t and $\hat{D}$ is the predicted depth at time $t+1$ warped back to time t, F_z is the z-dimension of the predicted sceneflow.

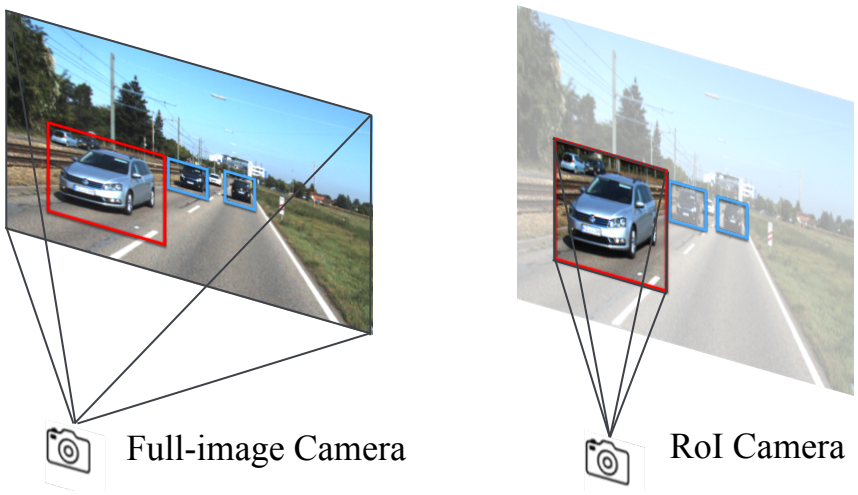


Figure 2.4: Illustration of image rescale and crop process and the change in the camera intrinsics.

2.2.3 Object-centric Scene Flow Prediction

Image based consistency losses are typically applied by warping the whole image and then computing the consistency over the whole image - examples for optical flow prediction can be found in [165, 177]. For 3D scene flow this is not an ideal choice due to the sparsity of non-zero flow vectors. Compared to the static background, moving objects constitute only a small fraction of the image pixels. This unbalanced moving/static pixel distribution makes naively learning full image flow hard and ends up in zero flow predictions even on moving objects. To make the network focus on predicting the correct flow on moving objects and provide a more balanced supervision, we therefore use object bounding box detections obtained from a state-of-the-art 2D object detection system [81]. It is important to note that the object detection does not actually tell us if the object is moving or not. This information is learned by our network using our view synthesis based loss functions.

Formally each flow prediction happens in a region of interest (RoI) within the original image, with size and location $B = \left[x, y, w, h\right]$. In our system the per-object flow map is predicted at a fixed size $w_r \times h_r$ using a RCNN based architecture as detailed in Section 2.3. For our view synthesis based loss functions we need to transform the image intrinsics $K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$ into RoI specific versions. The change only affects the intrinsic camera parameters and hence we need to compute a new intrinsic matrix K^j for each RoI j. The transformation ends up to be a displacement of the principal point and scaling of the focal length - $K^j = \begin{bmatrix} f_x w_r/w & 0 & (c_x - x)w_r/w \\ 0 & f_y h_r/h & (c_y - y)h_r/h \\ 0 & 0 & 1 \end{bmatrix}$.

Note that we do not need bounding box associations between different viewpoints or time instants. We only compute detections for frame I_t^l and use a slightly expanded area as our RoI in frames that we warp to our reference frame for computing consistency losses in Eq. 2.1 and 2.3.

2.2.4 RoI Assembly for Full Frame Scene Flow

We assemble a complete scene flow from the object specific maps F^j. However, overlapping RoIs and certain RoIs may even contain multiple moving objects. Therefore we predict an object mask M^j for each RoI j in addition to F^j. The full 3D scene flow map F is computed as:

$$F = \sum_j M^j \odot F^j \tag{2.4}$$

We then use the full image flow map F with Eq. 2.1 and Eq. 2.3 for full image photometric and geometric losses. Note that the assembly procedure is fully differentiable and we are able to train instance masks $\mathcal{M} = \{M^1, \ldots, M^j\}$ without any explicit mask supervision. We later use these instance masks (with flow) to identify moving objects (cf.Figure 2.6).

2.2.5 Full Learning Objective

We first state our full image synthesis based loss and then explain further priors we impose in our training loss. Our image synthesis loss function is based on four images I_t^l, I_t^r, I_{t+1}^l and I_{t+1}^r and can be split into three parts

$$\mathcal{L}^{\text{tot}} = \mathcal{L}^{lr} + \mathcal{L}^{\text{RoI}} + \mathcal{L}^{\text{t}} \tag{2.5}$$

Where $\mathcal{L}^{lr}$ is the loss for left-right consistency, $\mathcal{L}^{\text{RoI}}$ is the RoI based loss function and $\mathcal{L}^{\text{t}}$ is the full image based loss function on flow and depth over time. To state how the three parts are defined we introduce the notation $s \to t$ to indicate the warping from source s to target t.

$$\begin{aligned} \mathcal{L}^{lr} &= \mathcal{L}_{\text{photo}}(I_t^l, \hat{I}_t^{r\to l}) + \mathcal{L}_{\text{photo}}(I_{t+1}^l, \hat{I}_{t+1}^{r\to l}) \\ \mathcal{L}^{\text{RoI}} &= \sum_j \mathcal{L}_{\text{photo}}(I_t^{l,j}, \hat{I}_{t+1\to t}^{l,j}) + \mathcal{L}_{\text{geo}}(D_t^{l,j}, \hat{D}_{t+1\to t}^{l,j}, F_t^{lj}) \\ \mathcal{L}^{\text{t}} &= \mathcal{L}_{\text{photo}}(I_t^l, \hat{I}_{t+1\to t}^l) + \mathcal{L}_{\text{geo}}(D_t^l, \hat{D}_{t+1\to t}^l, F_t^l) \end{aligned} \tag{2.6}$$

Beside the loss detailed above, we use additional priors such as smoothness for depth and flow while respecting discontinuties at boundaries [37]. Optionally, we use the classical stereo system ELAS [31] to compute an incomplete disparity map and use it for weak supervision with an L_1 loss.

2.3 Network Architecture

Figure 2.5 illustrates our network for scene flow, mask and depth prediction. We first talk about the 3D grid representation used to integrate the information from all images and then describe each component of the network.

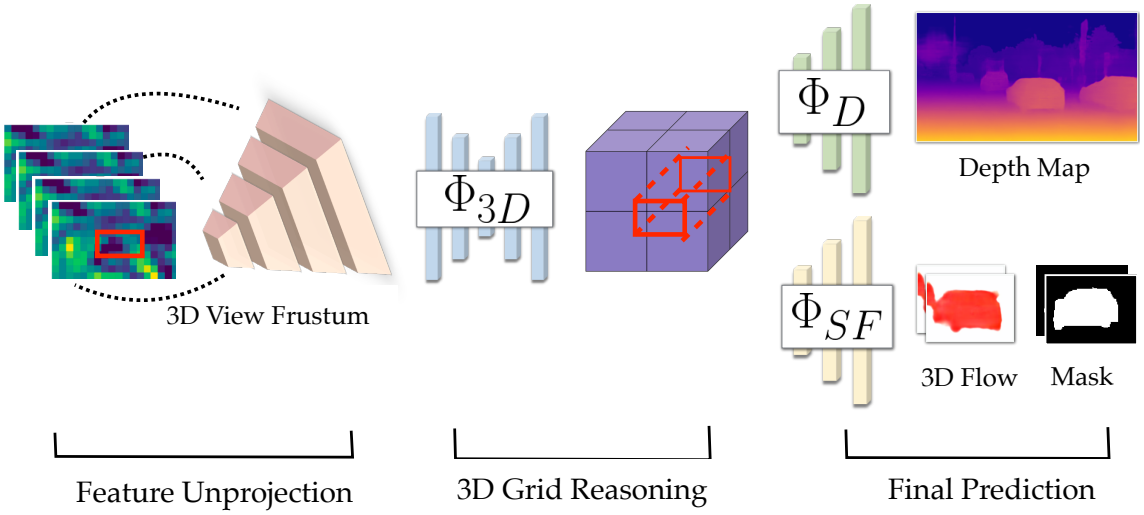
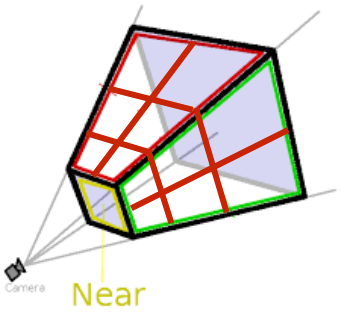


Figure 2.5: Network architecture. Our system predicts depth and instance-level 3D scene flow in a single forward pass. With extracted image features, we unproject features into a discretized view frustum grid, and then use a 3D CNN Φ_{3D} and finally perform prediction using depth Φ_D and scene flow Φ_{SF} decoders.

2.3.1 3D Grid Representation

In order to enable the network to reason about the scene geometry in 3D, we unproject the 2D features into a 3D grid [58]. A common discretization is to split a 3D cuboid volume of interest into equally sized voxels. This representation is used for 3D object shape reconstruction [138, 58]. However, it is not suitable for outdoor scenes with a large depth range, where we want to be more certain about foreground objects' geometry and motion, and allow increasing uncertainty with increasing depth in the 3D world. This lends to using the well known frustum shaped grid called matching cost volume or plane sweep volume in classical (multi-view) stereo. In learning based stereo it has recently been used in [163]. The grid is discretized in image space plus an additional inverse depth ("nearness") coordinate, as shown in above image.



2.3.2 Network Components

Image Encoder. In the first stage the images are processed using a 2D CNN Φ_I, which outputs for each image a 2D feature map with c feature channels. The weights for this CNN are shared for all input frames - typically stereo frames at two time instants $\{I_t^l, I_t^r\}$ and $\{I_{t+1}^l, I_{t+1}^r\}$.

Unprojection. Using the 3D grid defined in Section 2.3.1, we lift the 2D information into the 3D space. We use the two left camera images as references images $\{I_t^l, I_{t+1}^l\}$ and generate these 3D grids in both their camera coordinates. Each grid is populated with image features from all 4 images by projecting the grid cell centers into the respective images using the corresponding projection matrices [58]. We use the left images as reference frames as we predict disparity maps and scene flow from I_t^l to I_{t+1}^l.

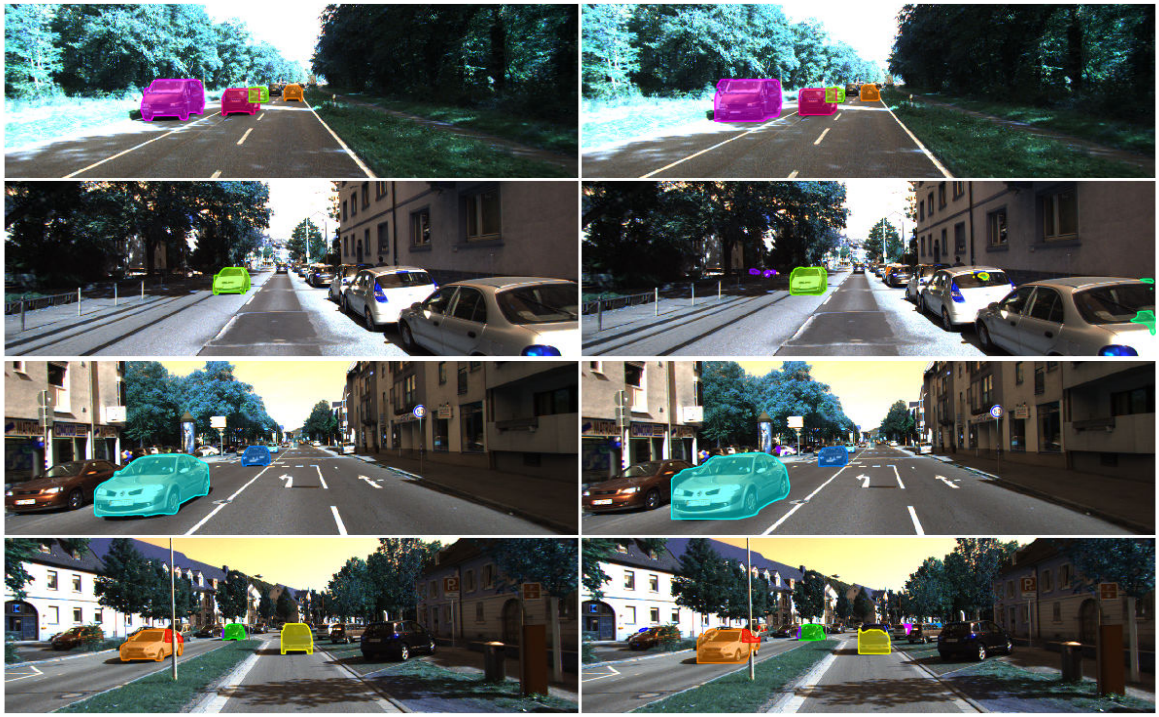
Grid Pooling. The grids from the previous stage contain image features from all 4 frames. In order to combine the information from multiple frames we use two strategies. We use element-wise max pooling for features from left and right pairs and concatenate the features for different time instants in each grid cell. The motivation is that for stereo frames, there is no object motion and hence the feature should align well after unprojection. Thus a simple strategy of max pooling works well. Whereas for frames at different time instants, we expect motion in the scene and thus there would be misalignment where objects move. The output from this stage are two grids G_t^l and G_{t+1}^l.

3D Grid Reasoning. The next module Φ_{3D} processes the above two grids independently and generates output grids of the same resolution $\tilde{G}_t^l$ and $\tilde{G}_{t+1}^l$. This module is implemented as a 3D encoder-decoder CNN module with skip connections following the U-Net architecture [115].

Output Modules. The final output is based on two CNN modules - one producing full frame depth for each reference image and one producing scene flow for each RoI in frame I_t. For each image I_i^l, with $i \in \{t, t+1\}$ we first collapse $\tilde{G}_i^l$ (a 4D tensor) into a 3D tensor C_i^l by concatenating features in the depth dimension. As the grid is aligned with the reference image's camera, this corresponds to accumulating the features from various disparity planes at every pixel into a single feature. This tensor is further processed using ϕ_D to produce the full frame disparity map. The 3D flow prediction follows an RCNN [35] based architecture where given RoIs, we crop out corresponding regions C_t^l using an RoI align layer [45] and pass them to ϕ_{SF} which predict the scene flow and instance mask for each RoI. We also use skip connections from the image encoder in ϕ_D and ϕ_{SF} to produce sharper predictions. The full frame scene flow map is created from the RoIs by pasting back as described in Section 2.2.4. The final outputs from our system are disparity maps D_t^l and D_{t+1}^l and a forward scene flow map F_t^l.

2.4 Experiments

We evaluate our instance-level 3d object motion and mask prediction on the KITTI 2015 sceneflow dataset [91]. This is the only available dataset that contains real images together with ground-truth scene flow annotations. Following existing



(a) Ground-truth (b) Prediction

Figure 2.6: Qualitative results on our instance-level moving object mask prediction. Instances are color-encoded.

work [89, 165, 177, 37], we adopt the official 200 training images as test set. The official testing set is adopted for the final finetuning process. This is possible as we do not require the ground truth for training. All the related images in the 28 scenes covered by test data are excluded for training. Figure 2.6 and Figure 2.7 show some qualitative results.

Training details Our system is implemented using TensorFlow [2]. All models are optimized end-to-end using Adam [61] with a learning rate of 1×10^{-4}, decay rate of 0.5 and decay steps of 100000. During training, we randomly crop the input images in the horizontal direction to obtain patches with the size of 384×640 as input to the network. We set the output size of each RoI as 128×128, we set the number of channels in the 3D grid to 64. The batch size is set as 1 to deal with flexible RoI number for training patch. For the image encoder, we finetune the first 4 convolutional layers from Inception ResNet V2 [127] pretrained on ImageNet. The rest of network is trained from scratch. We first train the depth prediction for 80K iterations on the KITTI raw dataset and then jointly train the depth and scene flow prediction for another 100k iterations. We finetune the model on the official testing set for another 120k iterations and use official 200 training images for comparison with other methods. The whole training process takes about 30 hours using a single NVIDIA Titan-X GPU.

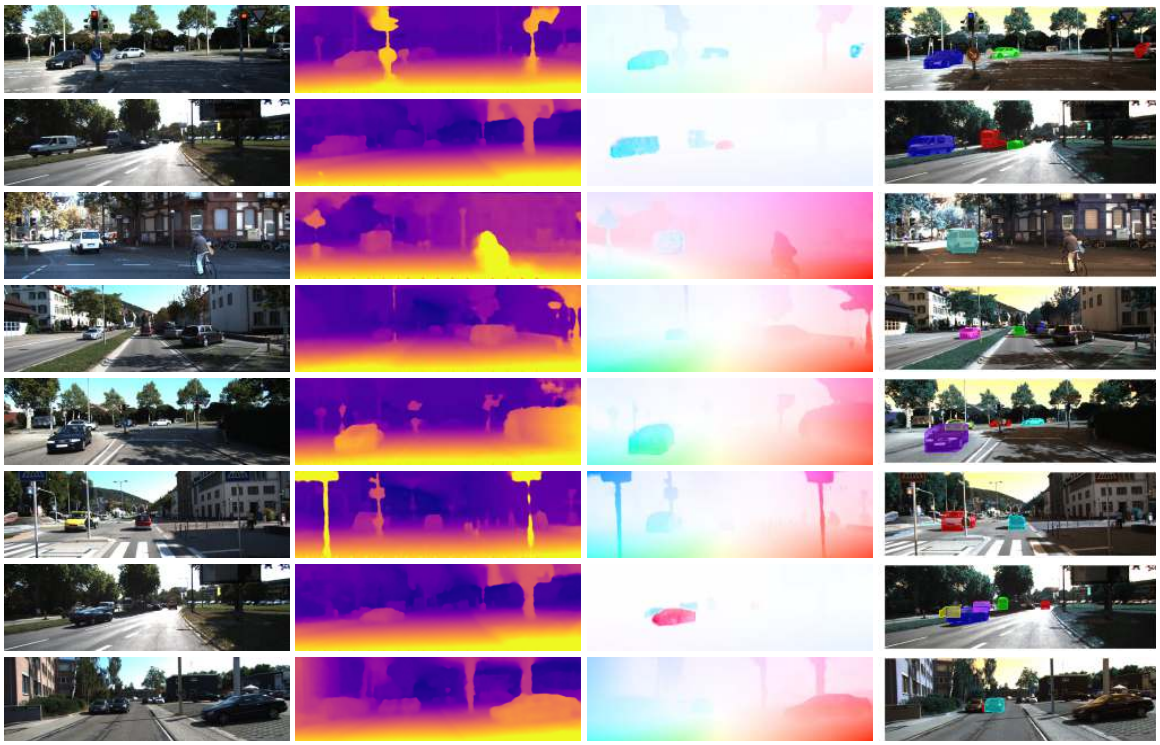


Figure 2.7: Qualitative results of our method. From left to right, reference image, depth, optical flow and instance-level moving object mask.

Method	AMAD$^{\downarrow}$	AMAE$^{\downarrow}$	AE$\leq$15$^{\circ\uparrow}$	AE$\leq$30$^{\circ\uparrow}$	SMAD$^{\downarrow}$	SMAE$^{\downarrow}$	SE$\leq$0.15$^{\uparrow}$	SE$\leq$0.3$^{\uparrow}$
GeoNet [165] + Godard [37]	6.98°	28.82°	62.93	77.16	0.256	0.503	0.351	0.554
UnflowC [89] + Godard [37]	5.96°	26.94°	64.87	77.58	0.240	0.471	36.21	58.62
Ours (no RoI consistency loss)	6.03°	29.34°	67.59	75.94	0.207	0.358	37.46	58.93
Our 3D scene flow	**5.19°**	**22.92°**	**74.78**	**78.87**	**0.193**	**0.334**	**40.95**	**62.72**

Table 2.1: Comparison of instance-level object motion in terms of motion direction(A) and speed (S). MAE denotes the mean average error, MAD denotes the median absolute deviation. The lower the better. We also report the percentage of the angle/speed error below different thresholds, where AE denotes the absolute angular error, SE denotes the absolute speed error. The higher the better.

2.4.1 Moving Object Speed and Direction Evaluation

Our method predicts 3D sceneflow for each independently moving object. For each test image pair, ground-truth annotation of the disparity image at time t, the disparity image at time $t+1$ warped into the first image's coordinate frame and the 2D optical flow from time t to time $t+1$ are provided. Using these GT annotations together with the estimated camera egomotion obtained from Libviso2 [32], we compute the 3D scene flow in the format of (x, y, z) for each image. To provide an instance-

Method	Image IoU	Instance IoU
Zhou et al. [172]	0.380	-
Bounding box detections [81]	0.365	0.655
Our mask prediction	**0.624**	**0.842**

Table 2.2: Moving object mask evaluation. We report IoU number in both the full image and the moving instance bounding box.

level analysis, we use the bbox detections [81], and find the dominant 3d flow for each object. As a result, we represent the motion direction and speed for each instance using a single 3d flow vector in the ground truth and all algorithms. We evaluate with the following metrics: the mean average error of the euclidean length of the 3d flow (speed), the mean average error of the angle of the 3d flow (motion direction) from the moving object pixels. For robustness to outliers we report the percentage of the mean average error below different thresholds. For comparison with other self-supervised flow and depth learning methods we need to reconstruct scene flow from depth and optical flow prediction. Geonet provides depthmaps with unknown scale factor and unflow does not estimate depth, we therefore use the depth results from Godard *et al.* [37]. As shown in Table 2.1, the average instance-level motion direction error of our method is less than 23°, about 15% smaller than the result obtained from the best self-supervised optical flow combined with the best self-supervised depth algorithm. In our prediction, about 75% of moving instances have an angular error below 15°.

2.4.2 Moving Object Instance Mask Evaluation

Our method can produce instance-level moving object segmentation from object bounding boxes and stereo videos. This is achieved without any instance mask ground truth supervision. We evaluate our predictions on the KITTI sceneflow 2015 training split. The dataset provides an "Object map" which contains the foreground moving cars in each image. We use this motion mask as ground truth in our segmentation evaluation. Figure 2.6 shows some qualitative result of our moving object mask prediction. As shown in Table 2.2, we evaluate our mask prediction using the Intersection Over Union (IoU) metric. Specifically, We compute the mean image-level IoU which considers both moving object and static background and the mean instance-level IoU for only moving objects. Our method achieves highest IoU for mask prediction. As a baseline comparison, we use mask generated from SSD [81] 2D bounding box detections. Those masks contain both moving and static cars, thus it can only achieve an mean IoU of 0.34 for the full image mask. Even with the GT object movement information, it does not have tight object boundary and thus can only achieve a mean IoU of 0.655. This illustrates how our method effectively learns to determine which object is moving and identify an accurate instance segmentation

Method	Dataset	Non-occluded	All Regions
EpicFlow [111]	-	4.45	9.57
FlowNetS [25]	C+ST	8.12	14.19
FlowNet2 [50]	C+T	4.93	10.06
GeoNet [165]	K	8.05	10.81
DF-Net [177]	K+SY	-	8.98
UnFlowC [89]	K+SY	-	8.80
Ranjan *et al.* [110]	K	-	7.76
Ours	K	4.97	5.39
Ours (refined)	K	**4.19**	**5.13**

Table 2.3: Results on KITTI 2015 flow training set over non-occluded regions and overall regions. We use the average end-point error (EPE) metric to do the comparison. The classical method EpicFlow takes 16s per frame at runtime; The FlowNetS and FlowNet2 are learned with GT flow supervision. SY denotes SYNTHIA dataset [116], ST denotes Sintel dataset, C denotes FlyingChairs dataset, T denotes FlyingThings3D dataset. Numbers from other methods are directly taken from the paper.

for moving cars. We improve the result on both image-level and instance-level IoU. We also compare with Zhou *et al.* [172] which generates the foreground mask for all moving objects and occlusion region in the image. Their methods do not provide instance-level information, hence we cannot obtain the instance-level IoU numbers.

2.4.3 Optical Flow Evaluation

An additional evaluation is to project our 3D flow predictions back to 2D to obtain the optical flow. As shown in Table 2.3, our method achieves the lowest EPE in both non-occluded regions and overall regions compared to other self-supervised methods. As a baseline comparison, we train a model without RoI consistency loss, which shows a decrease in performance. Optionally, we add an optical flow refinement sub-network, to further improve our optical flow result. The subnetwork is a unet which takes the warped image and the raw optical flow, together with original image frames as input. This enables the network to further improve the optical flow prediction in a similar way as the architecture proposed in [109].

2.4.4 Depth Evaluation

To evaluate our depth prediction we use the KITTI 2015 stereo training set of 200 disparity images as test data and compare to other self-supervised learning and

Method	Binocular	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Godard et al. [37]	no	0.124	1.388	6.125	0.217	0.841	0.936	0.975
libelas [31]	yes	0.1862	2.192	6.307	3.528	0.8197	0.8355	0.8414
Godard et al. [37]	yes	0.068	0.835	4.392	0.146	0.942	**0.978**	**0.989**
Ours	yes	**0.064**	**0.699**	**3.896**	**0.144**	**0.945**	0.975	0.987

Table 2.4: Results on the KITTI 2015 stereo training set of 200 disparity images. All learning-based methods are trained on KITTI raw dataset excluding the testing image sequences. The top half shows method which uses monocular image as input, the bottom half shows methods which use binocular images as input.

Method	**D1**			**D2**			**FL**			**ALL**		
	bg	fg	bg+fg	bg	fg	bg+fg	bg	fg	bg+fg	bg	fg	**bg+fg**
EPC [162]	23.62	27.38	26.81	18.75	70.89	60.97	25.34	28.00	25.74			
EPC++ [83] (mono)	30.67	34.38	32.73	18.36	84.64	65.63	17.57	27.30	19.78	>30.67	>84.64	>65.63
EPC++ [83] (stereo)	22.76	26.63	23.84	16.37	70.39	60.32	17.58	**26.89**	**19.64**	>22.76	>70.39	>60.32
Godard *et al.* [37]	9.43	18.74	10.86	-	-	-	-	-	-	-	-	-
GeoNet [165]	-	-	-	-	-	-	43.54	48.24	44.26	-	-	-
Godard [37] + GeoNet flow	9.43	18.74	10.86	9.10	25.95	25.42	43.54	48.24	44.26	48.22	55.75	49.38
Ours	**6.27**	**15.95**	**7.76**	**8.46**	**23.60**	**10.92**	**14.36**	51.25	20.16	**16.58**	**53.20**	**22.64**

Table 2.5: Results on KITTI 2015 scene flow training split. All number shows the percentage of correctly predicted pixels. D1 denotes the disparity image at time t, D2 denotes the disparity image at time $t + 1$ warped into the first frame, FL denotes the 2D optical flow between the two time instances, fg denotes the foreground, and bg denotes the background.

classical algorithms in Table. 2.4. We compare to algorithms that take binocular stereo as input at test time. Our method achieves a higher accuracy as we input two consecutive binocular frames and our network also manages to match over time.

2.4.5 Scene Flow Evaluation

We compare other unsupervised method in the sceneflow subset by directly using their released results or running their released code. For this benchmark, a pixel is considered to be correctly estimated if the disparity or flow end-point error is ≤ 3 pixels or $\leq 5\%$. For scene flow this criterion needs to be fulfilled for two disparity maps and the flow map. As shown in Table 2.5, our method has an overall better accuracy than earlier self-supervised methods. Compared to classical approaches which optimize at test time our accuracy is still lower. However, test time optimization is in general prohibitively slow for real-time systems.

2.5 Discussion

We presented a system to predict depth and object scene flow. Our network is trained using raw stereo sequences with off-the-shelf object detectors using image consistency as key learning objective. Our formulation is general and can be applied in any setting where a dynamic scene is imaged by multiple cameras - e.g. a multi-view capture system [56]. In future work, we would like to extend our system to integrate longer range temporal information. An emergent notion of objects to remove the dependence on pretrained object detectors is a further research direction. We also intend to explore general scenarios such as casual video captures using dual camera consumer devices and leverage large scale training for a truly general purpose depth and scene flow prediction system.

Chapter 3

Predicting Long-term Human Motion

Figure 3.1 shows the image of a typical indoor scene. Overlaid on this image is the pose trajectory of a person, depicted here by renderings of her body skeleton over time instants, where Frames 1-3 are in the past, Frame 4 is the present, and Frames 5-12 are in the future. In this paper, we study the following problem: *Given the scene image and the person's past pose and location history in 2D, predict her future poses and locations.*

Human movement is goal-directed and influenced by the spatial layout of the objects in the scene. For example, the person may be heading towards the window, and will find a path through the space avoiding collisions with various objects that might be in the way. Or perhaps a person approaches a chair with the intention to sit on it, and will adopt an appropriate path and pose sequence to achieve such a goal efficiently. We seek to understand such goal-directed, spatially contextualized human behavior, which we have formalized as a pose sequence and location prediction task.

With the advent of deep learning, there has been remarkable progress on the task of predicting human pose sequences [26, 87, 157, 169]. However, these frameworks do not pay attention to scene context. As a representative example, Zhang et al. [169] detect the human bounding boxes across multiple time instances and derive their predictive signal from the evolving appearance of the human figure, but do not make use of the background image. Given this limitation, the predictions tend to be short-term (around 1 second), and local in space, e.g., walking in the same spot without global movement. If we want to make predictions that encompass bigger spatiotemporal neighborhoods, we need to make predictions conditioned on the scene context.

We make the following philosophical choices: (1) To understand long term behavior, we must reason in terms of goals. In the setting of moving through space, the goals could be represented by the destination points in the image. We allow multi-

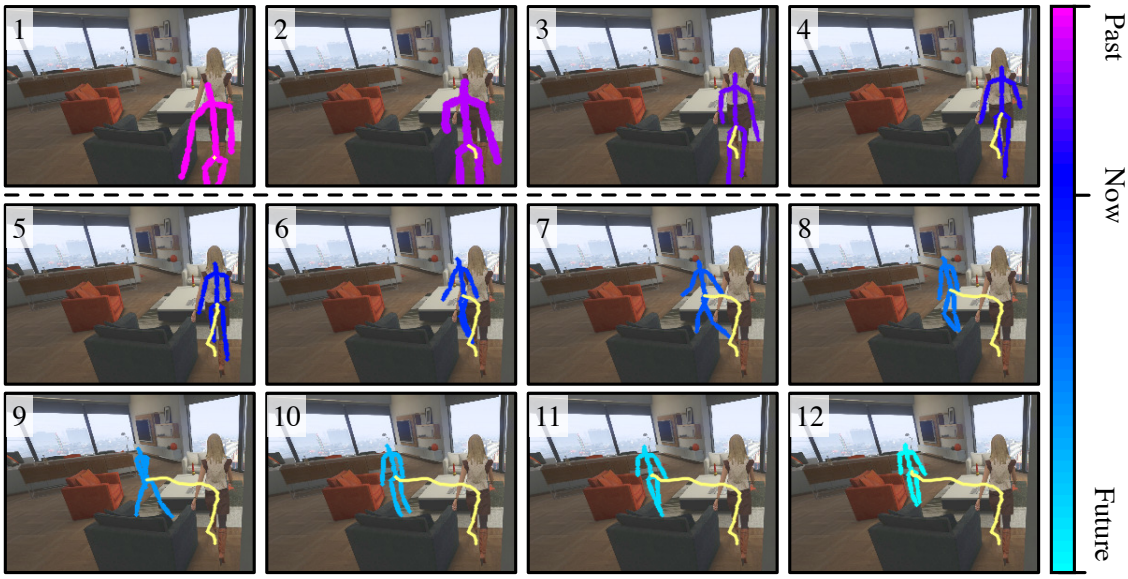


Figure 3.1: Long-term 3D human motion prediction. Given a single scene image and 2D pose histories (the 1st row), we aim to predict long-term 3D human motion (projected on the image, shown in the 2-3rd rows) influenced by scene. The human path is visualized as a yellow line.

modality by generating multiple hypotheses of human movement "goals", represented by 2D destinations in the image space. (2) Instead of taking a 3D path planning approach as in the classical robotics literature [7, 71], we approach the construction of likely human motions as a learning problem by constructing a convolutional network to implicitly learn the scene constraints from lots of human-scene interaction videos. We represent the scene using 2D images.

Specifically, we propose a learning framework that factorizes this task into three sequential stages as shown in Figure Figure 3.2. Our model sequentially predicts the motion goals, plans the 3D paths following each goal and finally generates the 3D poses. In Section 3.4, we demonstrate our model not only outperforms existing methods quantitatively but also generates more visually plausible 3D future motion.

To train such a learning system, we contribute a large-scale synthetic dataset focusing on human-scene interaction. Existing real datasets on 3D human motion have either contrived environment [51, 153], relatively noisy 3D annotations [119], or limited motion range due to the depth sensor [41, 119]. This motivates us to collect a diverse synthetic dataset with clean 3D annotations. We turn the Grand Theft Auto (GTA) gaming engine into an automatic data pipeline with control over different actors, scenes, cameras, lighting conditions, and motions. We collect over one million HD resolution RGB-D frames with 3D annotations which we discuss in detail in Section 3.3. Pre-training on our dataset stabilizes training and improves prediction performance on real dataset [41].

In summary, our key contributions are the following: (1) We formulate a new

(a) predicted goals (b) planned paths (c) final poses

Figure 3.2: Overall pipeline. Given a single scene image and 2D pose histories, our method first samples (a) multiple possible future 2D destinations. We then predict the (b) 3D human path towards each destination. Finally, our model generates (c) 3D human pose sequences following paths, visualized with the ground-truth scene point cloud.

task of long-term 3D human motion prediction with scene context in terms of 3D poses and 3D locations. (2) We develop a novel three-stage computational framework that utilizes scene context for goal-oriented motion prediction, which outperforms existing methods both quantitatively and qualitatively. (3) We contribute a new synthetic dataset with diverse recordings of human-scene interaction and clean annotations.

3.1 Background

Predicting future human motion under real-world social context and scene constraints is a long-standing problem [5, 38, 46, 64, 117]. Due to its complexity, most of the current approaches can be classified into global trajectory prediction and local pose prediction. We connect these two components in a single framework for long-term scene-aware future human motion prediction.

Global trajectory prediction: Early approaches in trajectory prediction model the effect of social-scene interactions using physical forces [46], continuum dynamics [135], Hidden Markov model [64], or game theory [84]. Many of these approaches achieve competitive results even on modern pedestrian datasets [74, 103]. With the resurgence of neural nets, data-driven prediction paradigm that captures multi-modal interaction between the scene and its agents becomes more dominant [5, 6, 15, 38, 85, 117, 130, 166]. Similar to our method, they model the influence of the scene implicitly. However, unlike our formulation that considers images from diverse camera viewpoints, they make the key assumption of the bird-eye view image or known 3D information [5, 38, 64, 117].

Local pose prediction: Similar to trajectory prediction, there has been plenty of interest in predicting future pose from image sequences both in the form of image generation [145, 171], 2D pose [16, 149], and 3D pose [18, 33, 158, 169]. These methods exploit the local image context around the human to guide the future pose

generation but do not pay attention to the background image or the global scene context. Approaches that focus on predicting 3D pose from 3D pose history also exist and are heavily used in motion tracking [23, 146]. The goal is to learn 3D pose prior conditioning on the past motion using techniques such as Graphical Models [10], linear dynamical systems [102], trajectory basis [3, 4], or Gaussian Process latent variable models [132, 141, 150, 151], and more recently neural networks such as recurrent nets [26, 53, 78, 87, 101], temporal convolution nets [47, 48, 75], or graph convolution net in frequency space [157]. However, since these methods completely ignore the image context, the predicted human motion may not be consistent with the scene, i.e, waling through the wall. In contrast, we propose to utilize the scene context for future human motion prediction. This is similar in spirit to iMapper [93]. However, this approach relies on computationally expensive offline optimization to jointly reason about the scene and the human motion. Currently, there is no learning-based method that holistically models the scene context and human pose for more than a single time instance [17, 73, 77, 152, 154].

3D Human Motion Dataset Training high capacity neural models requires large-scale and diverse training data. Existing human motion capture datasets either contain no environment [1], contrive environment [51, 153], or in the outdoor setting without 3D annotation [148]. Human motion datasets with 3D scenes are often much smaller and have relatively noisy 3D human poses [41, 119] due to the limitations of the depth sensor. To circumvent such problems, researchers exploit the interface between the game engine and the graphics rendering system to collect large-scale synthetic datasets [24, 65]. Our effort on synthetic training data generation is a consolidation of such work to the new task of future human motion prediction with scene context.

3.2 Approach

In this chapter, we focus on long-term 3D human motion prediction that is goal-directed and is under the influence of scene context. We approach this problem by constructing a learning framework that factorizes long-term human motions into modeling their potential goal, planing 3D path and pose sequence, as shown in Figure Figure 3.3. Concretely, given a N-step 2D human pose history $\mathbf{X}_{1:N}$ and an 2D image[1] of the scene $\mathbf{I}$ (the Nth video frame in our case), we want to predict the next T-step 3D human poses together with their locations, denoted by a sequence $\mathbf{Y}_{N+1:N+T}$. We assume a known human skeleton consists of J keypoints, such that $\mathbf{X} \in \mathbb{R}^{J\times 2}, \mathbf{Y} \in \mathbb{R}^{J\times 3}$. We also assume a known camera model parameterized by its intrinsic matrix $\mathbf{K} \in \mathbb{R}^3$. To denote a specific keypoint position, we use the super-

[1]We choose to represent the scene by RGB images rather than RGBD scans because they are more readily available in many practical applications.

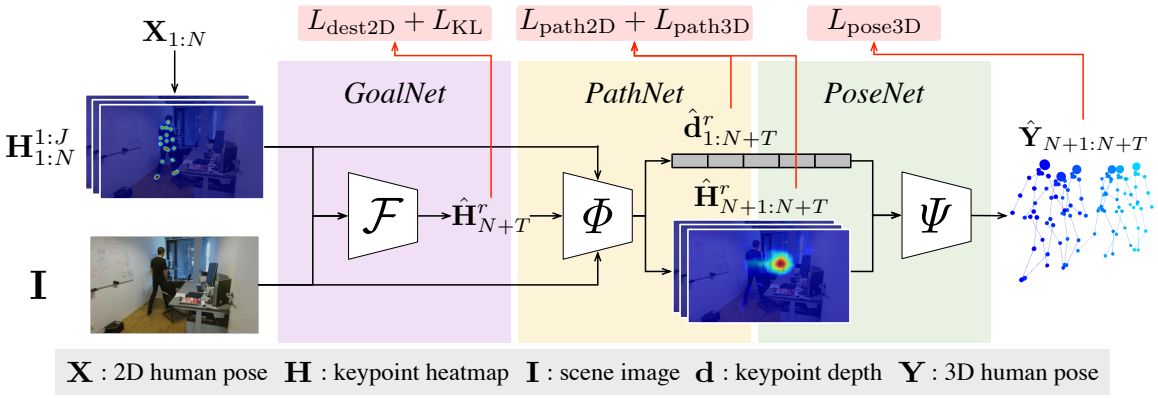


Figure 3.3: Network architecture. Our pipeline contains three stages: *GoalNet* predicts 2D motion destinations of the human based on the reference image and 2D pose heatmaps (Section 3.2.1); *PathNet* plans the 3D global path of the human with the input of 2D heatmaps, 2D destination, and the image (Section 3.2.2); *PoseNet* predicts 3D global human motion, i.e., the 3D human pose sequences, following the predicted path (Section 3.2.3).

script of its index in the skeleton, e.g., $\mathbf{X}^r$ refers to the 2D location of the human center (torso) indexed by $r \in [1, J]$.

We motivate and elaborate our modular design for each stage in the rest of the section. Specifically, *GoalNet* learns to predict multiple possible human motion goals, represented as 2D destinations in the image space, based on a 2D pose history and the scene image. Next, *PathNet* learns to plan a 3D path towards each goal – the 3D location sequence of the human center (torso) – in conjunction with the scene context. Finally, *PoseNet* predicts 3D human poses at each time step following the predicted 3D path. In this way, the resulting 3D human motion has global movement and is more plausible considering the surrounding scene.

Thanks to this modular design, our model can have either deterministic or stochastic predictions. When deploying GoalNet, our model can sample multiple destinations, which results in stochastic prediction of future human motion. If not deploying GoalNet, our model generates single-mode prediction instead. We discuss them in more detail in the rest of the section and evaluate both predictions in our experiments.

3.2.1 *GoalNet*: Predicting 2D Path Destination

To understand long-term human motion, we must reason in terms of goals. Instead of employing autoregressive models to generate poses step-by-step, we seek to first directly predict the destination of the motion in the image space. We allow our model to express uncertainty of human motion by learning a distribution of possible motion destinations, instead of a single hypothesis. This gives rise to our GoalNet denoted as $\mathcal{F}$ for sampling plausible 2D path destination.

GoalNet learns a distribution of possible 2D destinations $\{\hat{\mathbf{X}}^r_{N+T}\}$ at the end of the time horizon conditioned on the 2D pose history $\mathbf{X}_{1:N}$ and the scene image $\mathbf{I}$. We parametrize each human keypoint $\mathbf{X}^j$ by a heatmap channel $\mathbf{H}^j$ which preserves spatial correlation with the image context.

We employ GoalNet as a conditional variational auto-encoder [62]. The model encodes the inputs into a latent $\mathbf{z}$-space, from which we sample a random $\mathbf{z}$ vector for decoding and predicting the target destination positions. Formally, we have

$$\mathbf{z} \sim \mathcal{Q}(\mathbf{z}|\mathbf{H}^{1:J}_{1:N}, \mathbf{I}) \equiv \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}), \text{ where } \boldsymbol{\mu}, \boldsymbol{\sigma} = \mathcal{F}_{\text{enc}}(\mathbf{H}^{1:J}_{1:N}, \mathbf{I}). \tag{3.1}$$

In this way, we estimate a variational posterior $\mathcal{Q}$ by assuming a Gaussian information bottleneck using the decoder. Next, given a sampled $\mathbf{z}$ latent vector, we learn to predict our target destination heatmap with our GoalNet decoder,

$$\hat{\mathbf{H}}^r_{N+T} = \mathcal{F}_{\text{dec}}(\mathbf{z}, \mathbf{I}), \tag{3.2}$$

where we additionally condition the decoding process on the scene image. We use soft-argmax [125] to extract the 2D human motion destination $\hat{\mathbf{X}}^r_{N+T}$ from this heatmap $\hat{\mathbf{H}}^r_{N+T}$. We choose to use soft-argmax operation because it is differentiable and can produce sub-pixel locations. By constructing GoalNet, we have

$$\hat{\mathbf{H}}^r_{N+T} = \mathcal{F}(\mathbf{I}, \mathbf{H}^{1:J}_{1:N}). \tag{3.3}$$

We train GoalNet by minimizing two objectives: (1) the destination prediction error and (2) the KL-divergence between the estimated variational posterior $\mathcal{Q}$ and a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$:

$$\begin{aligned} L_{\text{dest2D}} &= \|\mathbf{X}^r_{N+T} - \hat{\mathbf{X}}^r_{N+T}\|_1, \\ L_{\text{KL}} &= \text{KL}\big[\mathcal{Q}\big(\mathbf{z}|\mathbf{H}^{1:J}_{1:N}, \mathbf{I}\big)||\mathcal{N}(0, 1)\big], \end{aligned} \tag{3.4}$$

where we weigh equally between them. During testing, our GoalNet is able to sample a set of latent variables $\{\mathbf{z}\}$ from $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and map them to multiple plausible 2D destinations $\{\hat{\mathbf{H}}^r_{N+T}\}$.

3.2.2 *PathNet*: Planning 3D Path towards Destination

With predicted destinations in the image space, our method further predicts 3D paths (human center locations per timestep) towards each destination. The destination determines where to move while the scene context determines how to move. We design a network that exploits both the 2D destination and the image for future 3D path planning. A key design choice we make here is that, instead of directly regressing 3D global coordinate values of human paths, we represent the 3D path as a combination of 2D path heatmaps and the depth values of the human center over

time. This 3D path representation facilitates training as validated in our experiments (Section 3.4.3).

As shown in Figure 3.3, our PathNet Φ takes the scene image $\mathbf{I}$, the 2D pose history $\mathbf{H}^{1:J}_{1:N}$, and the 2D destination assignment $\hat{\mathbf{H}}^r_{N+T}$ as inputs, and predicts global 3D path represented as $(\hat{\mathbf{H}}^r_{N+1:N+T}, \hat{\mathbf{d}}^r_{1:N+T})$, where $\hat{d}^r_t \in \mathbb{R}$ denotes the depth of human center at time t:

$$\hat{\mathbf{H}}^r_{N+1:N+T}, \hat{\mathbf{d}}^r_{1:N+T} = \Phi(\mathbf{I}, \mathbf{X}^{1:J}_{1:N}, \mathbf{X}^r_{N+T}). \tag{3.5}$$

We use soft-argmax to extract the resulting 2D path $\hat{\mathbf{X}}^r_{N+1:N+T}$ from predicted heatmaps $\hat{\mathbf{H}}^r_{N+1:N+T}$. Finally, we obtain the 3D path $\hat{\mathbf{Y}}^r_{1:N+T}$ by back-projecting the 2D path into the 3D camera coordinate frame using the human center depth $\hat{\mathbf{d}}^r_{1:N+T}$ and camera intrinsics $\mathbf{K}$.

We use Hourglass54 [72, 96] as the backbone of PathNet to encode both the input image and 2D pose heatmaps. The network has two branches where the first branch predicts 2D path heatmaps and the second branch predicts the depth of the human torso.

We train our PathNet using two supervisions. We supervise our path predictions with ground-truth 2D heatmaps:

$$L_{\text{path2D}} = \|\mathbf{X}^r_{N+1:N+T} - \hat{\mathbf{X}}^r_{N+1:N+T}\|_1. \tag{3.6}$$

We also supervise path predictions with 3D path coordinates, while encouraging smooth predictions by penalizing large positional changes between consecutive frames:

$$L_{\text{path3D}} = \|\mathbf{Y}^r_{1:N+T} - \hat{\mathbf{Y}}^r_{1:N+T}\|_1 + \|\hat{\mathbf{Y}}^r_{1:N+T-1} - \hat{\mathbf{Y}}^r_{2:N+T}\|_1. \tag{3.7}$$

These losses are summed together with equal weight as the final training loss. During training, we use the ground-truth destination to train our PathNet, while during testing, we can use predictions from the GoalNet.

The GoalNet and PathNet we describe so far enable sampling multiple 3D paths during inference. We thus refer to it as the stochastic mode of the model. The modular design of GoalNet and PathNet is flexible. By removing GoalNet and input $\mathbf{X}^r_{N+T}$ from Equation 3.5, we can directly use PathNet to produce deterministic 3D path predictions. We study these two modes, deterministic and stochastic mode, in our experiments.

3.2.3 *PoseNet*: Generating 3D Pose following Path

With the predicted 3D path $\hat{\mathbf{Y}}^r_{1:N+T}$ and 2D pose history $\mathbf{X}_{1:N}$, we use the transformer network [142] as our PoseNet Ψ to predict 3D poses following such path. Instead of predicting the 3D poses from scratch, we first lift 2D pose history into 3D to obtain a noisy 3D human pose sequence $\bar{\mathbf{Y}}_{1:N+T}$ as input, and further use Ψ to

refine them to obtain the final prediction. Our initial estimation consists of two steps. We first obtain a noisy 3D poses $\bar{\mathbf{Y}}_{1:N}$ by back-projecting 2D pose history $\mathbf{X}_{1:N}$ into 3D using the human torso depth $\hat{\mathbf{d}}^r_{1:N}$ and camera intrinsics $\mathbf{K}$. We next replicate the present 3D pose $\bar{\mathbf{Y}}_N$ to each of the predicted future 3D path location for an initial estimation of future 3D poses $\bar{\mathbf{Y}}_{N+1:N+T}$. We then concatenate both estimations together to form $\bar{\mathbf{Y}}_{1:N+T}$ as input to our PoseNet:

$$\hat{\mathbf{Y}}_{N+1:N+T} = \Psi(\bar{\mathbf{Y}}_{1:N+T}). \tag{3.8}$$

The training objective for PoseNet is to minimize the distance between the 3D pose prediction and the ground-truth defined as:

$$L_{\text{pose3D}} = \|\mathbf{Y}_{N+1:N+T} - \hat{\mathbf{Y}}_{N+1:N+T}\|_1. \tag{3.9}$$

During training, ground-truth 3D path $\mathbf{Y}^r_{1:N+T}$ is used for estimating coarse 3D pose input. During testing, we use the predicted 3D path $\hat{\mathbf{Y}}^r_{1:N+T}$ from PathNet.

3.3 GTA Indoor Motion Dataset

We introduce the GTA Indoor Motion dataset (GTA-IM) that emphasizes human-scene interactions. Our motivation for this dataset is that existing real datasets on human-scene interaction [41, 119] have relatively noisy 3D human pose annotations and limited long-range human motion limited by depth sensors. On the other hand, existing synthetic human datasets [24, 65] focus on the task of human pose estimation or parts segmentation and sample data in wide-open outdoor scenes with limited interactable objects.

To overcome the above issues, we spend extensive efforts in collecting a synthetic dataset by developing an interface with the game engine for controlling characters, cameras, and action tasks in a fully automatic manner. For each character, we randomize the goal destination inside the 3D scene, the specific task to do, the walking style, and the movement speed. We control the lighting condition by changing different weather conditions and daytime. We also diversify the camera location and viewing angle over a sphere around the actor such that it points towards the actor. We use in-game ray tracing API and synchronized human segmentation map to track actors. The collected actions include climbing the stairs, lying down, sitting, opening the door, and etc. – a set of basic activities within indoor scenes. For example, the character has 22 walking styles including 10 female and 12 male walking styles. All of these factors enable us to collect a diverse and realistic dataset with accurate annotations for our challenging task.

In total, we collect one million RGBD frames of 1920×1080 resolution with the ground-truth 3D human pose (98 joints), human segmentation, and camera pose. Some examples are shown in Figure 3.4. The dataset contains 50 human characters

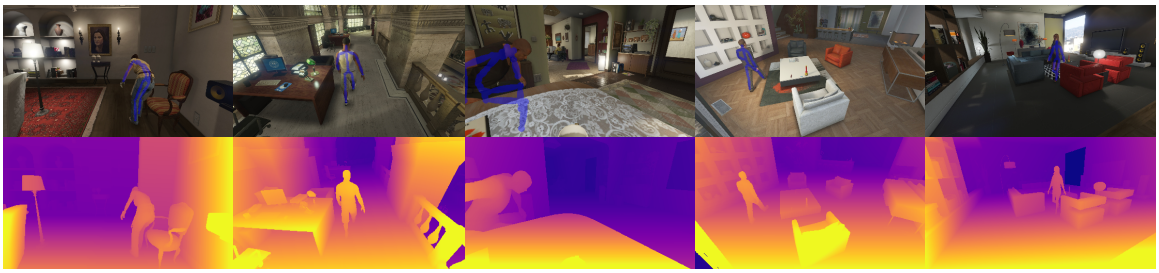


Figure 3.4: Sample RGBD images from GTA-IM dataset. Our dataset contains realistic RGB images (visualized with the 2D pose), accurate depth maps, and clean 3D human pose annotations.

acting inside 10 different large indoor scenes. Each scene has several floors, including living rooms, bedrooms, kitchens, balconies, and etc., enabling diverse interaction activities.

3.4 Evaluation

We perform extensive quantitative and qualitative evaluations of our future 3D human path and motion predictions. The rest of this section is organized as follows: We first describe the datasets we use in Section 3.4.1. We then elaborate on our quantitative evaluation metrics and strong baselines in Section 3.4.2. Further, we show our quantitative and qualitative improvement over previous methods in Section 3.4.3. Finally, we evaluate our long-term predictions and show qualitative results of destination samples and final 3D pose results in Section 3.4.4. We discussed some failure cases in Section 3.4.5.

3.4.1 Datasets

GTA-IM: We train and test our model on our collected dataset as described in Section 3.3. We split 8 scenes for training and 2 scenes for evaluation. We choose 21 out of 98 human joints provided from the dataset. We convert both the 3D path and the 3D pose into the camera coordinate frame for both training and evaluation.

PROX: Proximal Relationships with Object eXclusion (PROX) is a new dataset captured using the Kinect-One sensor by Hassan et al. [41]. It comprises of 12 different 3D scenes and RGB sequences of 20 subjects moving in and interacting with the scenes. We split the dataset with 52 training sequences and 8 sequences for testing. Also, we extract 18 joints from the SMPL-X model [99] from the provided human pose parameters.

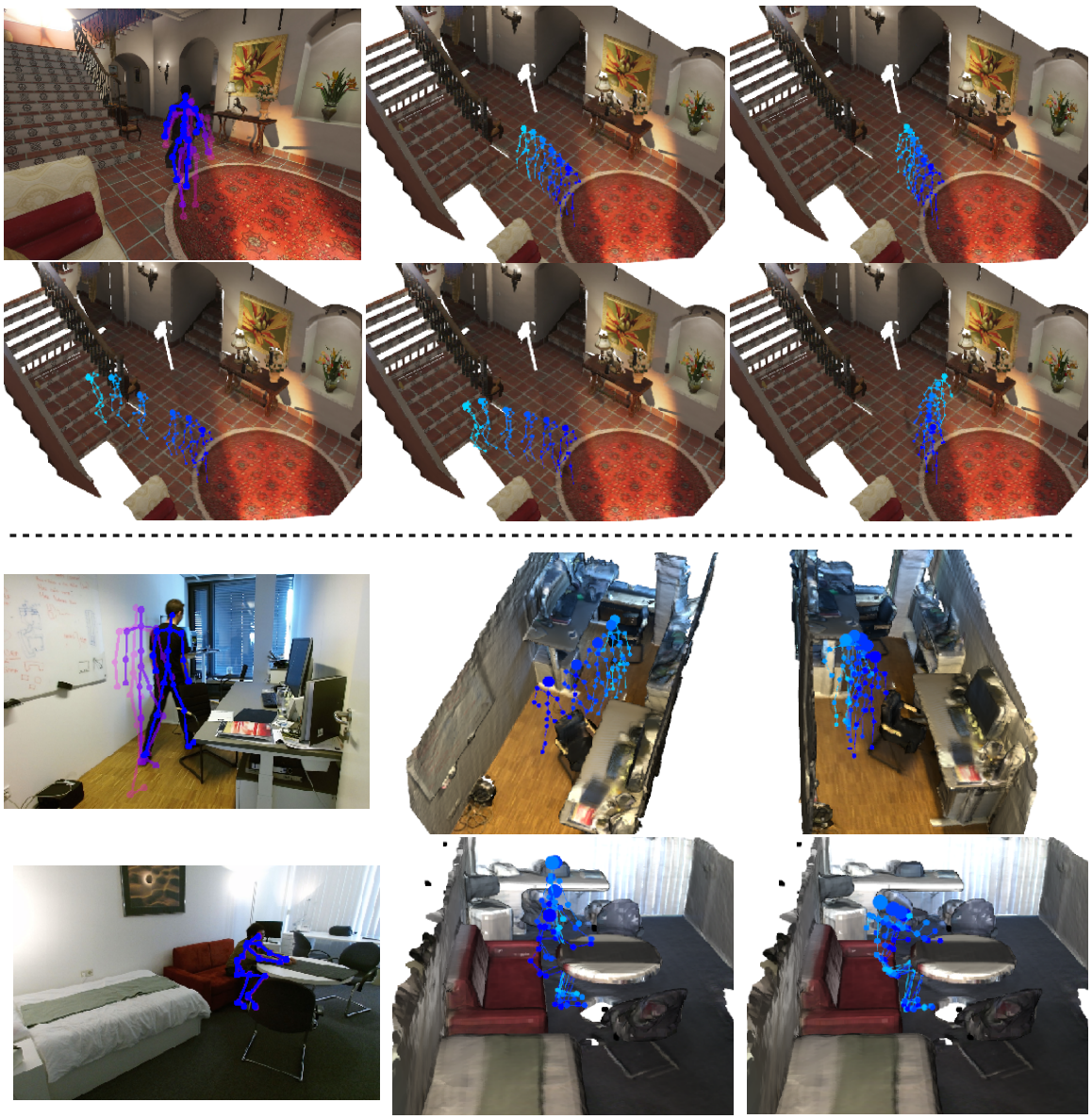


Figure 3.5: Qualitative results on long-term stochastic prediction. In each example, we first show the input image with 2D pose histories and then our stochastic predictions. In the first example (1st and 2nd row), we show five different future human movement predictions by sampling different human "goals", e.g., turning left to climb upstairs, or going straight through the hallway. For the following examples at each row, we only show two stochastic predictions per example due to space limitation. Our method can generate diverse human motion, e.g., turning left/right, walking straight, taking a u-turn, standing up from sitting, and laying back on the sofa.

3.4.2 Evaluation Metric and Baselines

Metrics: We use the Mean Per Joint Position Error (MPJPE) [51] as a metric for quantitatively evaluating both the 3D path and 3D pose prediction. We report the performance at different time steps (seconds) in millimeters (mm).

Baselines: To the best of our knowledge, there exists no prior work that predicts 3D human pose with global movement using 2D pose sequence as input. Thus, we propose three strong baselines for comparison with our method. For the first baseline, we combine the recent 2D-to-3D human pose estimation method [100] and 3D human pose prediction method [157]. For the second baseline, we use Transformer [142], the state-of-the-art sequence-to-sequence model, to perform 3D prediction directly from 2D inputs treating the entire problem as a single-stage sequence to sequence task. For the third baseline, we compare with is constructed by first predicting the future 2D human pose using [142] from inputs and then lifting the predicted pose into 3D using [100]. Note that none of these baselines consider scene context or deal with uncertainty in their future predictions. We train all models on both datasets for two-second-long prediction based on 1-second-long history and report their best performance for comparison.

3.4.3 Comparison with Baselines

In this section, we perform quantitative evaluations of our method in the two datasets. We also show some qualitative comparisons in Figure 3.6. We evaluate the two modes of our model: the stochastic mode that can generate multiple future predictions by sampling different 2D destinations from the GoalNet; and the deterministic mode that can generate one identical prediction without deploying GoalNet.

GTA-IM: The quantitative evaluation of 3D path and 3D pose prediction in GTA-IM dataset is shown in Table 3.1. Our deterministic model with image input can outperform the other methods by a margin, i.e., with an average error of 173 mm vs. 193 mm. When using sampling during inference, the method can generate multiple hypotheses of the future 3D pose sequence. We evaluate different numbers of samples and select the predictions among all samples that best matches ground truth to report the error. We find using four samples during inference can match the performance of our deterministic model (173 mm error), while using ten samples, we further cut down the error to 165 mm. These results validate that our stochastic model can help deal with the uncertainty of future human motion and outperform the deterministic baselines with few samples.

As an ablation, we directly regress 3D coordinates ("Ours w/ xyz." in the Table 3.1) and observe an overall error that is 18 mm higher than the error of our deterministic model (191 mm vs. 173 mm). This validates that representing the 3D path as the depth and 2D heatmap of the human center is better due to its strong

	3D path error (mm)				3D pose error (mm)				
Time step (second)	0.5	1	1.5	2	0.5	1	1.5	2	**All ↓**
TR [142]	277	352	457	603	291	374	489	641	406
TR [142] + VP [100]	157	240	358	494	174	267	388	526	211
VP [100] + LTD [157]	124	194	276	367	121	180	249	330	193
Ours (deterministic)	**104**	**163**	**219**	**297**	**91**	**158**	**237**	**328**	**173**
Ours (samples=4)	114	162	227	310	94	161	236	323	173
Ours (samples=10)	110	154	213	289	90	154	224	306	165
Ours w/ xyz. output	122	179	252	336	101	177	262	359	191
Ours w/o image	128	177	242	320	99	179	271	367	196
Ours w/ masked image	120	168	235	314	96	170	265	358	189
Ours w/ RGBD input	100	138	193	262	93	160	235	322	172
Ours w/ GT destination	104	125	146	170	85	133	178	234	137

Table 3.1: Evaluation results in GTA-IM dataset. We compare other baselines in terms of 3D path and pose error. The last column (All) is the mean average error of the entire prediction over all time steps. VP denotes Pavllo et al. [100], TR denotes transformer [142] and LTD denotes Wei et al. [157]. GT stands for ground-truth, xyz. stands for directly regressing 3D coordinates of the path. We report the error of our stochastic predictions with varying number of samples.

correlation to the image appearance. We also ablates different types of input to our model. Without image input, the average error is 23 mm higher. With only masked images input, i.e., replacing pixels outside human crop by ImageNet mean pixel values, the error is 16 mm highe. This validates that using full image to encode scene context is more helpful than only observing cropped human image, especially for long-term prediction. Using both color and depth image as input ("Ours w/ RGBD input"), the average error is 172 mm which is similar to the model with RGB input. This indicates that our model implicitly learns to reason about depth information from 2D input. If we use ground-truth 2D destinations instead of predicted ones, and the overall error decreases down to 137 mm. It implies that the uncertainty of the future destination is the major source of difficulty in this problem.

PROX: The evaluation results in Table 3.2 demonstrate that our method outperforms the previous state of the art in terms of mean MPJPE of all time steps, 270 mm vs. 282 mm. Overall, we share the same conclusion as the comparisons in GTA-IM dataset. When using sampling during inference, three samples during inference can beat the performance of our deterministic model (264 mm vs. 270 mm), while using ten samples, the error decreases to 249 mm. Note that these improvements are more prominent than what we observe on GTA-IM benchmark. This is because the uncertainty of future motion in the real dataset is larger. Therefore, stochastic

	3D path error (mm)				3D pose error (mm)				
Time step (second)	0.5	1	1.5	2	0.5	1	1.5	2	**All ↓**
TR [142]	487	583	682	783	512	603	698	801	615
TR [142] + VP [100]	262	358	461	548	297	398	502	590	326
VP [100] + LTD [157]	194	263	332	394	216	274	335	**394**	282
Ours w/o GTA-IM pretrain	192	258	336	419	192	273	352	426	280
Ours (deterministic)	**189**	**245**	**317**	**389**	**190**	**264**	**335**	406	**270**
Ours (samples=3)	192	245	311	398	187	258	328	397	264
Ours (samples=6)	185	229	285	368	184	249	312	377	254
Ours (samples=10)	181	222	273	354	182	244	304	367	249
Ours w/ gt destination	193	223	234	237	195	235	276	321	237

Table 3.2: Evaluation results in PROX dataset. We compare other baselines in terms of 3D future path and pose prediction. VP denotes Pavllo et al. [100], TR denotes transformer [142] and LTD denotes Wei et al. [157]. GT stands for ground-truth. We rank all methods using mean average error of the entire prediction (last column).

predictions have more advantage.

Moreover, we find that pre-training in GTA-IM dataset can achieve better performance (270 mm vs. 280 mm). Our method exploits the image context and tends to overfit in PROX dataset because it is less diverse in terms of camera poses and background appearance (both are constant per video sequence). Pre-training in our synthetic dataset with diverse appearance and clean annotations can help prevent overfitting and boost the final performance.

Qualitative comparison: In Figure 3.6, we show qualitative comparison with the baseline of VP [100] and LTD [157]. This baseline is quantitatively competitive as shown in Table 3.1 and 3.2. However, without considering scene context, their predicted results may not be feasible inside the 3D scene, e.g., the person cannot go through a table or sofa. In contrast, our model implicitly learns the scene constraints from the image and can generate more plausible 3D human motion in practice.

3.4.4 Evaluation on Longer-term Predictions

To demonstrate our method can predict future human motion for more than 2 seconds, we train another model to produce the 3-second-long future prediction. In Figure Figure 3.7, we show the self-comparisons between our stochastic predictions and deterministic predictions. Our stochastic models can beat their deterministic counterpart using 5 samples. With increasing number of samples, the testing error decreases accordingly. The error gap between deterministic results and stochastic results becomes larger at the later stage of the prediction, i.e., more than 100 mm

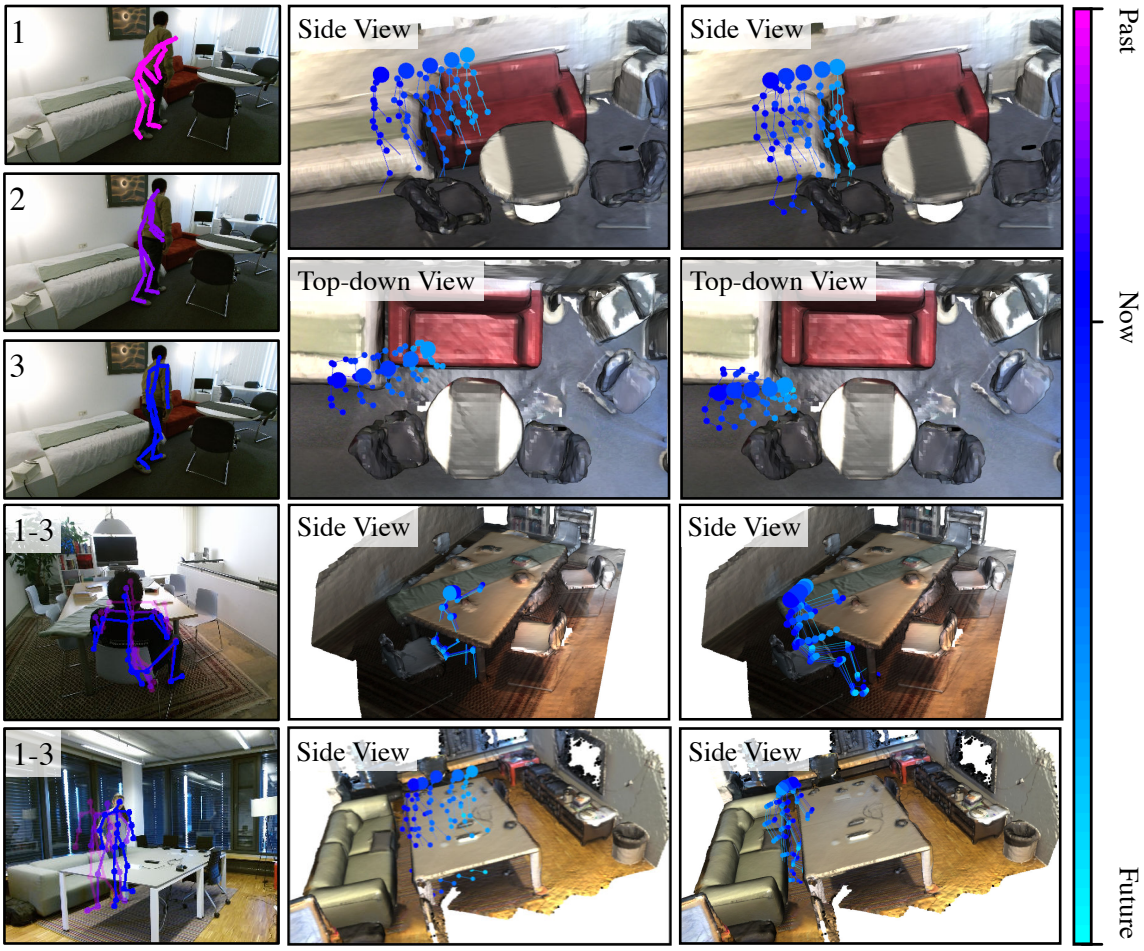
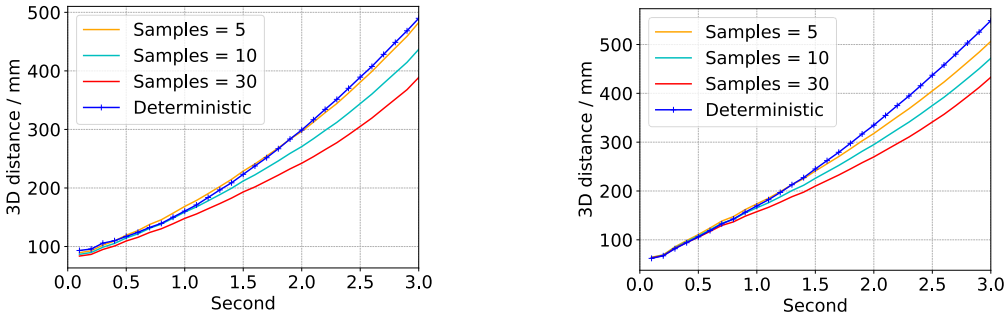


Figure 3.6: Qualitative comparison. We visualize the input (a), the results of VP[100] and LTD [157] (b) and our results (c) in the ground-truth 3D mesh. The color of pose is changed over timesteps according to the color bar. The first example includes both top-down and side view. From the visualization, we can observe some collisions between the baseline results and the 3D scene, while our predicted motion are more plausible by taking the scene context into consideration.

difference at 3 second time step. This indicates the advantage of the stochastic model in long-term future motion prediction.

We show qualitative results of our stochastic predictions on movement destinations in Figure 3.8, and long-term future motion in Figure 3.5. Our method can generate diverse human movement destination, and realistic 3D future human motion by considering the environment, e.g., turning left/right, walking straight, taking a U-turn, climbing stairs, standing up from sitting, and laying back on the sofa.



(a) predicted 3D paths (b) predicted 3D poses

Figure 3.7: Comparison between our stochastic predictions and deterministic predictions. We show error curves of predicted (a) 3D paths and (b) 3D poses with varying numbers of samples over varying timesteps on GTA-IM dataset. In all plots, we find that our stochastic model can achieve better results with a small number of samples, especially in the long-term prediction (within 2-3 second time span).

Figure 3.8: Destination sampling results. In each image, the blue dots denote the path history, the green dots are ground-truth future destination, red dots are sample destinations from the GoalNet which we draw 30 times from the standard Gaussian. Our method can generate diverse plausible motion destination samples which leads to different activities.

3.4.5 Failure cases

Our model implicitly learns scene constraints in a data-driven manner from large amounts of training data, and can produce consistent 3D human paths without serious collision comparing to previous methods which do not take scene context into consideration as shown in Figure 3.6. However, without assuming we have access to the pre-reconstructed 3D mesh and using expensive offline optimization as [41], the resulting 3D poses may not strictly meet all physical constraints of the 3D scene geometry. Some failure cases are shown in Figure 3.9. In the red circled area, we observe small intersections between the human feet and the 3D scene mesh, e.g., the ground floor or the bed. This issue could be relieved by integrating multi-view/temporal images as input to the learning system to recover the 3D scene better. The resulting 3D scene could be further used to enforce explicit scene geometry constraints for refining the 3D poses. We leave this to the future work.

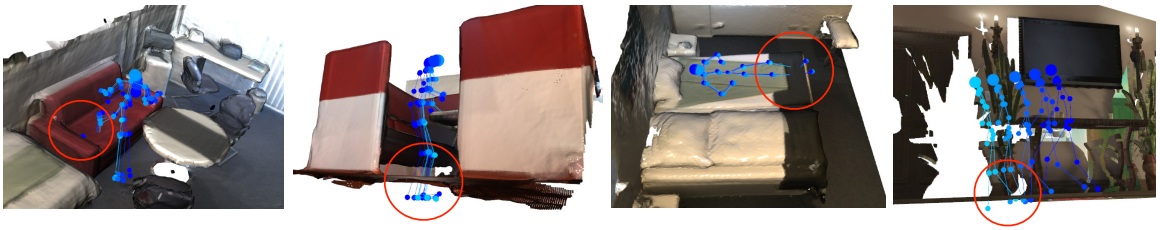


Figure 3.9: Visualization of failure cases. In each red circle area, we observe the intersection between the human feet and the 3D mesh, e.g., the bed.

3.5 Discussion

In this chapter, we study the challenging task of long-term 3D human motion prediction with only 2D input. This research problem is very relevant to many real-world applications where understanding and predicting feasible human motion considering the surrounding space is critical, e.g., a home service robot collaborating with the moving people, AR glass providing navigation guide to visually impaired people, and autonomous vehicle planning the action considering the safety of pedestrians. We present an initial attempt in attacking the problem by contributing a new dataset with diverse human-scene interactions and clean 3D annotations. We also propose the first method that can predict long-term stochastic 3D future human motion from 2D inputs, while taking the scene context into consideration. There are still many aspects in this problem that can be explored in the future, such as how to effectively evaluate the naturalness and feasibility of the stochastic human motion predictions, and how to incorporate information of dynamic objects and multiple moving people inside the scene.

Chapter 4

Perceiving Hand-Object Interaction

Our hands are the primary way we interact with objects in the world. In turn, we designed our world with hands in mind. Therefore, understanding hand-object interactions is an important ingredient for building agents that perceive and act in the *real world*. For example, it can allow them to learn object affordances [34], infer human intents [90], and learn manipulation skills from humans [104, 107, 86].

What does it mean to understand hand-object interactions? We argue that fully capturing the richness of hand-object interactions requires *3D understanding*.

Figure 4.1: Reconstructions in the wild. For each row, we show the input image (top), the reconstructed hand and object in two different viewpoints (bottom). Our method can achieve compelling results for a variety of object categories, grasp types, and interaction scenarios.

Figure 4.2: Images from existing hand-object dataset. The reality gap between the existing in-the-lab datasets with 3D annotations (left) and in-the-wild images (right) is large.

In general, recovering 3D from a single RGB image is an under-constrained problem. In the case of hand-object interactions, the problem is even more challenging due to heavy occlusions that occur during object manipulation, a wide range of small daily objects that are not even present in labeled recognition datasets, and fine-grained interactions with complex contacts that are difficult to model.

Overall, our community has made substantial progress toward this goal. However, due to the difficulty in obtaining 3D annotations in the wild, the data collection efforts have focused mainly on in-the-lab setting [39, 176, 29, 9, 129]. As shown in Figure 4.2, there is a large reality gap between the existing in-the-lab settings and the richness of environments and interactions found in images in the wild. Indeed, as shown in Table 4.1, existing datasets have a limited number of participants and objects.

In this chapter, we make two main contributions: (1) we develop a new technique for reconstructing 3D hands and objects from single images *in-the-wild*, called RHO for Reconstructing Hands and Objects and (2) we use this technique in conjunction with human intervention to create a new 3D dataset of humans Manipulating Objects in-the-Wild, that we call *MOW*.

Specifically, *RHOI* is a new optimization-based method for reconstructing hand-object interactions in the wild. The core idea is to leverage 2D image cues and 3D contact priors to provide constraints. RHOI consists of four steps: hand pose estimation using 2D hand keypoints, object pose estimation using 2D object mask and depth via differentiable rendering, joint optimization for hand-object configuration in 3D, and pose refinement using 3D contact priors.

A key feature of our method is the ability to deal with a wide variety of objects in the wild—an order of magnitude more than any previous work in hand-object reconstruction or general object reconstruction areas. This required several innovations. First, a new insight that segmentation masks for unknown object categories can be

	HO3D [39]	CP [9]	GRAB [129]	Ours
setting	lab	lab	lab	wild
data type	video	video	mocap	image
particip.	10	50	10	450
objects	10	25	51	121

Table 4.1: Existing 3D hand-object datasets. Our dataset contains *in-the-wild* images, as shown in Figure 4.2 right, and a large number of different participants and objects.

obtained using available recognition models. Second, a scalable data-driven way to enforce contact priors using a large collection of mocap data recorded in the lab.

We compare RHOI to existing approaches quantitatively in the lab settings where ground truth annotations are available and qualitatively on in-the-wild images. We find that RHOI performs better or on par with the state-of-the-art method on in-the-lab datasets. Moreover, we show that the existing approaches struggle on challenging in-the-wild images reinforcing the need for the dataset we collect.

We employ our method as part of a semi-automatic data annotation process. Specifically, we use human intervention for two reasons. First, to find and prepare the appropriate 3D model for the object being manipulated in the image. Second, to ensure high quality annotations by verifying and adjusting the results of our method in an iterative fashion. Using this procedure, we annotate 500 images from the EPIC Kitchens [21] and the 100 Days of Hands dataset [120]. These depict a rich diversity of manipulation actions, which we augment with newly collected 3D object models from *121 object categories*, 3D object poses, and 3D hand poses.

Our collected dataset in the wild, MOW, can be useful in many ways. It enables us to study and *understand* human manipulation actions using in-the-wild data. Indeed, the analysis presented in our work already leads to interesting findings that have not been shown before outside the lab settings. For example, we discover a low-dimensional embedding whose first dimension corresponds to the closure of the grasp (Figure 4.8).

In summary, our key contributions are: (1) We present a novel optimization-based procedure, RHOI, that is able to reconstruct hand-object interactions in the wild across diverse object categories; (2) We show quantitative and qualitative improvements over existing methods, especially on in-the-wild setting; (3) We contribute a new 3D dataset, MOW, of 500 images in the wild, spanning 121 object categories with annotation of instance category, 3D object models, 3D hand pose, and object pose annotation.

4.1 Background

3D hand pose estimation. Many previous works on hand pose estimation directly predict 3D joint locations from either depth [121, 124, 128, 139, 164, 168, 94] or RGB [112, 95, 12, 161, 175] images. Some recent works predict 3D hand joint rotations and shape parameters of parametric hand models such as MANO [113]. Fitting-based approaches [99, 159, 66] fit such parametric models to 2D keypoint detections to optimize 3D hand parameters. Learning-based approaches [173, 114] utilize deep networks to directly predict the hand parameter from RGB image input. Recently, [67, 66] proposes to use mesh convolution to directly predict 3D hand mesh reconstruction. We use a learning-based method [114] to obtain the initial hand pose estimation and further improve the result by imposing constraints on 2D hand keypoints and 3D hand-object contact priors.

3D object pose estimation. There are many existing works on estimating 3D object pose from a single image. Some approaches [136, 68, 36, 69] utilize neural network to predict the object shape, translation, and global orientation in the camera coordinate. These methods are trained with limited object categories and have difficulty generalizing to new objects. On the other hand, some approaches [79, 92, 160, 170, 126, 118] assume known 3D object model and focus on 6DOF object pose prediction. In this chapter, we take a fitting-based approach similar to [126, 170]. Our main novelty is the usage of a depth loss term which improves the results by imposing object shape constraints.

3D hand and object pose estimation. Early approaches in modeling hand and object require the input of multi-view image [97] or RGB-D sequence [139]. Recently, [44] proposes a deep model trained on synthetic data to reconstruct hand and object meshes from a monocular RGB image. [133] designs a neural network to jointly predict 3D hand pose and 3D object bounding boxes with a focus on egocentric scenarios. [43] proposes to leverage photometric consistency from temporal frames as additional signal for training the model with sparse set of annotated data. All these approaches were trained and tested on in-the-lab or synthetic datasets. In this chapter, we propose an approach without 3D supervision and we are the first to achieve good hand-object results in the wild from a single image.

3D hand-object datasets. Early datasets in hand grasping scenario requires manual annotations [123] or depth tracking [139] to obtain the ground truth, resulting in limited dataset size. To avoid the manual efforts, [29] uses motion capture system with magnetic sensors to collect annotations. [44] uses simulation to collect a synthetic hand-object dataset. [176, 39] introduces large-scale dataset with 3D annotation optimized from multi-view setups. Some recent datasets [8, 9, 129] also provide annotation for hand-object contact area in addition to 3D hand pose and object pose. The contact area is collected from either thermal sensor [8, 9] or marker-based MoCap system. All these datasets are of great efforts in modeling 3D hand-object interaction,

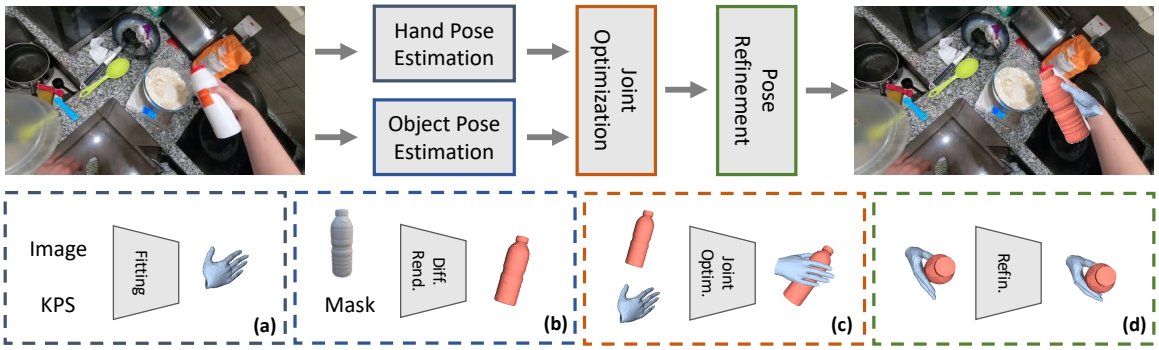


Figure 4.3: Method. In this chapter, we present an optimization-based method, called *RHOI*, that leverages 2D image cues and 3D contact priors for reconstructing hand-object interactions in the wild. It consists of four steps: (a) hand pose estimation by 2D keypoints fitting, (b) object pose estimation via differentiable rendering, (c) joint optimization for hand-object configuration, and (d) pose refinement using 3D contact priors learnt from mocap data.

however, they can only be collected in the lab setting due to the specific camera setup. As a result, limited number of participants and objects are present in them (as in Table 4.1). In this chapter, we contribute a dataset with in-the-wild images and diverse object categories. 3D annotations are obtained by running our optimization-based method *and* human intervention to achieve high quality.

Optimizing 3D interactions. Our method is in line with recent optimization-based approaches for modeling 3D interactions between human and scene [42], human and objects [170], and among multiple persons [55]. To obtain good 3D reconstructions, these methods require extra 3D input. For example, [42] requires the input of full 3D reconstructed scene mesh to impose geometry constraints. [170] requires manually labeling human-object mesh vertices for fine-grained interaction pairs and is only applied to 8 object categories in COCO [80]. In this chapter, we focus on modeling hand-object interactions. Our key advantage is the ability to deal with diverse objects in the wild without extra input. We propose to model contact priors using a scalable data-driven approach that levarages the available 3D mocap data. Together with a new method to obtain object masks, our approach is shown to be able to reconstruct hand interactions with 121 different object categories.

4.2 Method

We first describe our method for reconstructing hand-object interactions in the wild, called *RHOI*. As shown in Figure 4.3, it involves 4 steps: estimating the hand pose, the object pose, their 3D configuration jointly, and finally refining the pose using 3D contact priors. Intermediate results from each step are shown in Figure 4.4. We describe each step next. We note that while RHOI can be applied to multiple hands and objects, we assume a single pair for brevity. We will then evaluate RHOI,

then discuss how we curate our new dataset MOW and its analysis in the following sections.

4.2.1 Hand Pose Estimation

The first step of RHOI involves hand pose estimation (Figure 4.3a). Given an input image, we aim to reconstruct the full 3D hand mesh. We use a learning-based method to obtain the initial result and further improve the estimation by fitting it to 2D hand keypoints.

In particular, we represent the hand using a parametric model defined by MANO [113]: $\boldsymbol{V}_h = H(\boldsymbol{\theta}, \boldsymbol{\beta})$, where $\boldsymbol{\theta} \in \mathbb{R}^{3\times15}$ and $\boldsymbol{\beta} \in \mathbb{R}^{10}$ are the pose and shape parameters, respectively. Taking a single RGB image as input, we use FrankMocap [114] to estimate the weak-perspective camera model $\Pi_h = (t_x, t_y, s_h)$, and initial 3D hand parameters $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. We further optimize the hand pose by fitting to 2D hand keypoints obtained from [14, 122].

The hand pose optimization objective is to minimize the difference between 2D keypoints detection and 2D projection of 3D hand keypoints:

$$\boldsymbol{\theta}^*, \boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\theta},\boldsymbol{\beta}} L_{joints} + L_{reg}, \tag{4.1}$$

consisting of a 2D keypoints distance term L_{joints} and a regularization term L_{reg} for hand shape $\boldsymbol{\beta}$.

We convert the weak-perspective to perspective camera by assuming a fixed focal length f. The depth of the hand is approximated by the focal length divided by the camera scale s_h. We obtain the final hand vertices by:

$$\boldsymbol{V}_h^* = H(\boldsymbol{\theta}^*, \boldsymbol{\beta}^*) + [t_x, t_y, f/s_h], \tag{4.2}$$

4.2.2 Object Pose Estimation

In the next step of our method, Figure 4.3b, we recover the object pose using an *analysis-by-synthesis* approach. Given an input image and 3D model, we want to optimize the object scale $s \in \mathbb{R}$, 3D rotation $\boldsymbol{R} \in SO(3)$, and translation $\boldsymbol{T} \in \mathbb{R}^3$. We use a differentiable renderer [59] to render 3D model into 2D mask and depth maps. By comparing the rendered mask/depth with the targets, we compute the gradients to update the object parameters.

Object mask estimation. How can we obtain good objects masks for diverse objects in images in the wild? Modern 2D recognition models trained on large labeled datasets can provide reasonable predictions on real-world data [106]. However, in our case, we require instance masks for a range of object categories that are not even present in the available labeled datasets (*e.g.*, spatula, pliers, mic, *etc.*). Thus, we cannot expect the available models to recognize the objects correctly in our setting.

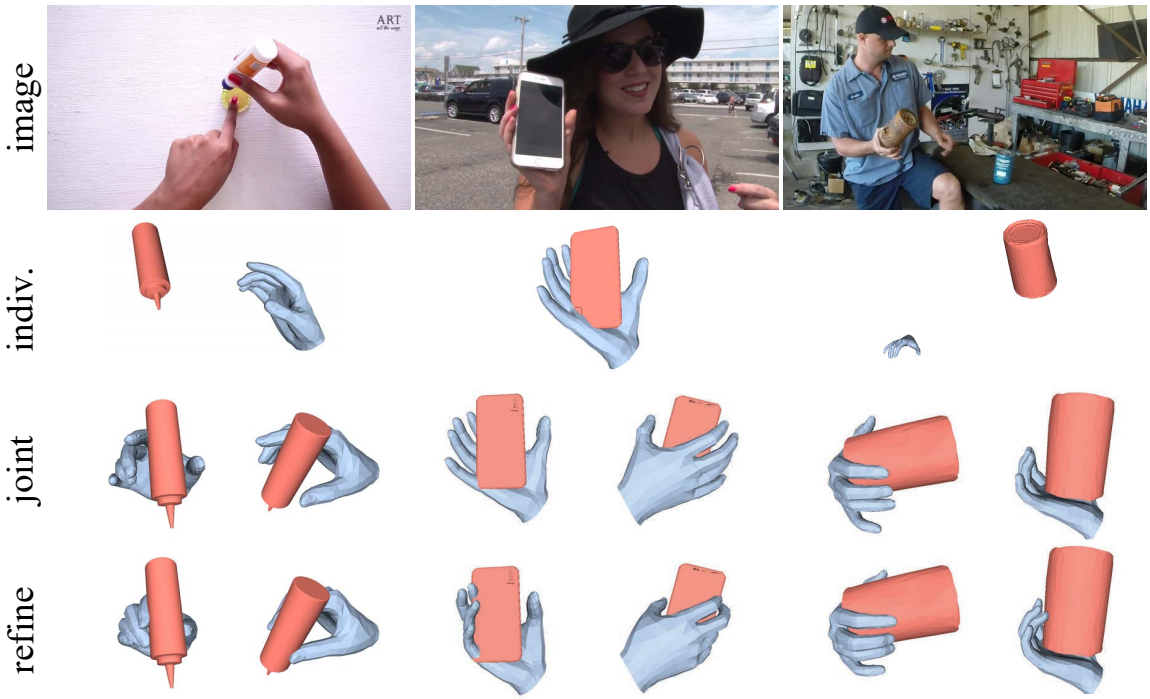


Figure 4.4: Intermediate results. *Top row:* input images. *2nd row:* results from individually optimizing hand and object. *3rd row:* results from joint optimization (two viewpoints per example). *Bottom row:* results after refinement.

Our key insight is that even if the predicted categories are incorrect, the instance masks are still quite reasonable for a variety of objects. For example, the models do not know what a spatula is called but are still able to segment it.

With this observation, we use available recognition models to estimate instance mask ignoring the category information. Specifically, we use PointRend model [63] trained on COCO [80]. For all object instances predictions in the image, we decide the instance that the hand is interacting with by running a hand detector [120]. The instance with highest IoU with the detected hand bounding box is selected. This automatic way allows us obtain instance masks for more than 100 daily object categories as shown in Section 4.4.3.

Mask loss. Given the estimated object mask, we optimize the object pose via differentiable rendering. In particular, we define the object mask loss as the L1 difference between the rendered and the estimated object masks:

$$L_{mask} = \|\mathrm{NR}_m(s, \boldsymbol{R}, \boldsymbol{T}) - \boldsymbol{M}\|, \tag{4.3}$$

where $\mathrm{NR}(\cdot)$ denotes the differentiable rendering operation which renders the 3D mesh into the 2D mask.

Depth loss. While the 2D mask loss is sufficient in some cases, it does not capture geometry information and can be ambiguous—multiple object poses can lead to sim-

ilar 2D masks. To overcome this, we employ a new loss term which fits 3D model to the depth map $\boldsymbol{D}$ estimated using [108]:

$$L_{depth} = \|\mathrm{NR}_d(s, \boldsymbol{R}, \boldsymbol{T}) - \boldsymbol{D}\|, \tag{4.4}$$

Object pose objective. Combining the mask and depth losses, we obtain the object pose estimation objective:

$$s^*, \boldsymbol{R}^*, \boldsymbol{T}^* = \underset{s,\boldsymbol{R},\boldsymbol{T}}{\arg\min}\ L_{mask} + L_{depth}, \tag{4.5}$$

We perform the optimization in the image region centered on the object. We start with a number of randomly initialized poses and select the one that leads to the lowest loss.

4.2.3 Joint Optimization

In this section, we describe how to jointly optimize the 3D hand and object results from previous sections (Figure 4.3c). Naively putting them together may result in implausible hand-object reconstructions (Figure 4.4, row 2), *i.e.*, the hand and object are far away from each other in 3D or having interpenetration. This issue is caused by the depth and scale ambiguity given only 2D input: a large object distant from the camera can have the same rendering result in 2D as a smaller object closer to the camera. To help resolve the ambiguity, we impose additional constraints based on hand-object distance and collision.

Interaction loss. The reconstructed hand and object could be distant in 3D space. However, when the hand is interacting with objects, their distance should be small. To push the interaction pair closer in 3D, we define an interaction loss based on their chamfer distance:

$$L_{dist} = \frac{1}{|\boldsymbol{V}_o|} \sum_{x \in \boldsymbol{V}_o} \min_{y \in \boldsymbol{V}_h} \|x - y\|_2 + \frac{1}{|\boldsymbol{V}_h|} \sum_{y \in \boldsymbol{V}_h} \min_{x \in \boldsymbol{V}_o} \|x - y\|_2. \tag{4.6}$$

For each vertex in the mesh, chamfer distance function finds the nearest point in the other point set, and sums up the distances. We find this loss term to be helpful in correcting the object scale by moving it closer to the hand.

Collision loss. Using the interaction loss alone may result in implausible artifacts, *e.g.*, hand colliding with the object. To resolve the issue, we add an interpenetration loss term to penalize the object vertices that are inside the hand mesh. We use the Signed Distance Field (SDF) from the hand mesh to check if any object vertex is inside the hand. We first calculate a tight box around the hand and voxelize it as a 3D grid for storing the SDF value. We use a modified SDF function ϕ for the hand mesh:

$$\phi(\boldsymbol{c}) = -\min(\mathrm{SDF}(c_x, c_y, c_z), 0). \tag{4.7}$$

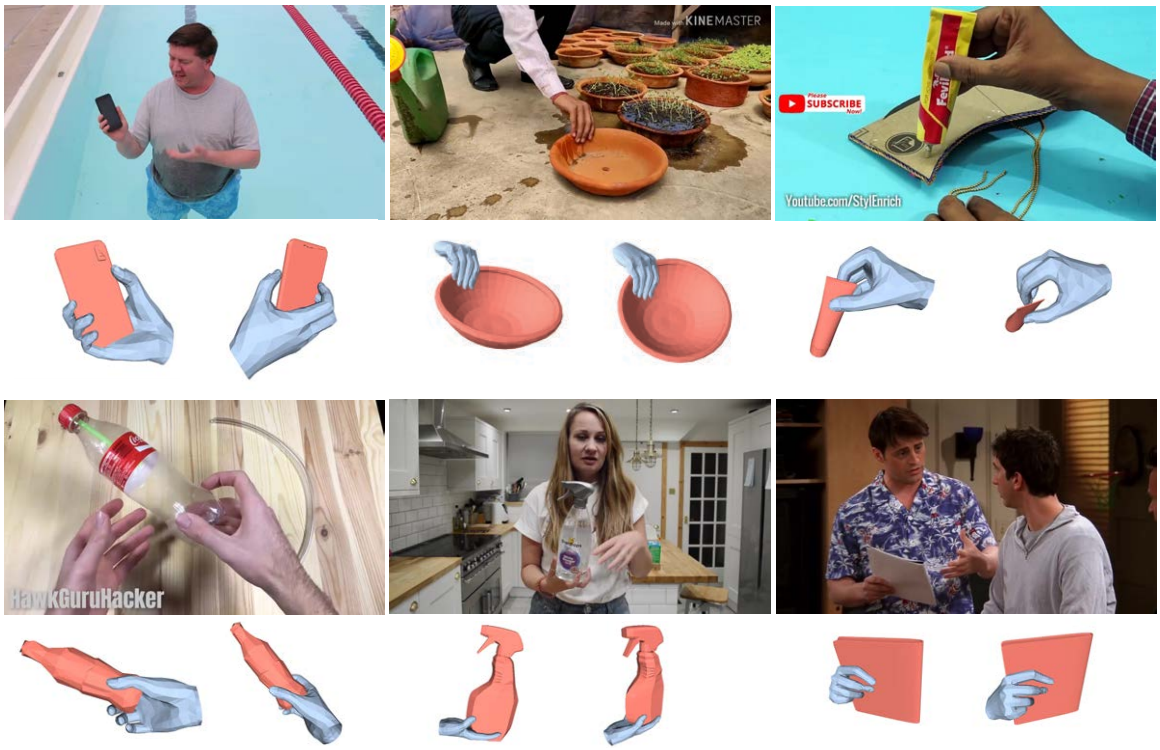


Figure 4.5: Qualitative results. Our method produces reconstructions of reasonably high-quality across a range of viewpoints, activities, and objects (see also the supplement).

For each voxel cell $\boldsymbol{c} = (c_x, c_y, c_z)$ in the 3D grid, if the cell is inside the hand mesh, ϕ takes positive values, proportional to the distance from the hand surface, while ϕ is 0 outside of the hand mesh. Then, the interpenetration loss can be calculated as:

$$L_{collision} = \sum_{\boldsymbol{v} \in \boldsymbol{V}_o^*} \phi(\boldsymbol{v}), \tag{4.8}$$

where $\phi(\boldsymbol{v})$ samples the SDF value of each object vertex $\boldsymbol{v}$ from the 3D hand grid in a differentiable way.

Joint objective. By incorporating the loss terms from object pose estimation, we obtain the overall objective for jointly optimizing the hand and the object:

$$L = \lambda_1 L_{mask} + \lambda_2 L_{depth} + \lambda_3 L_{dist} + \lambda_4 L_{collision}. \tag{4.9}$$

4.2.4 Pose Refinement

A physically plausible hand-object reconstruction should not only be collision-free, but also have enough hand-object contact area to support the action. However,

the interaction loss described in Section 4.2.3 does not take into account the fine-grained hand-object contact. To further refine the 3D reconstruction, we impose constraints on the hand-object contact as the final step of RHOI (Figure 4.3d).

Addressing this issue would be easy if we had per-vertex contact area annotation for both hand and object as we could enforce the contact region to be closer. However, obtaining such annotations for large collection of in-the-wild images is challenging. As a more scalable solution, we learn 3D contact priors from a large-scale hand mocap dataset [129]. The priors include the region of an object that the person is likely to contact. For example, human is more likely to hold the mug by its handle.

Given the hand mesh and object mesh obtained from the joint optimization, we want to update the hand pose so that it has more reasonable contact with the object. We train a small network to perform hand pose refinement.

The input to the network are the initial hand parameters (θ, γ) and the distance field $\boldsymbol{F}$ from the hand vertices $\boldsymbol{V}_h^*$ to the object vertices $\boldsymbol{V}_o^*$. For each hand vertex $\boldsymbol{v}_h$, we compute the distance to its nearest object vertex:

$$\boldsymbol{F}(\boldsymbol{v}_h) = \min_{\boldsymbol{v}_o \in \boldsymbol{V}_o^*} \|\boldsymbol{v}_h - \boldsymbol{v}_o\|_2^2 \tag{4.10}$$

Then, the network refines the hand parameters $(\boldsymbol{\theta}, \boldsymbol{\beta})$ in an iterative fashion. After each iteration, the distance field between hand and object is updated so that it can be used as input to the next step. The training data is obtained by perturbing the ground-truth hand pose parameters and translation to simulate noisy input estimates. As shown in Figure 4.4, we can observe that the results after refinement (4th row) can reconstruct more realistic interaction between hand and object than the previous step (3rd row).

4.3 Method Evaluation

In this section, we compare our method to existing methings in two settings: quantitatively in the lab and qualitatively in the wild. We further present ablation studies of different aspect of our method.

4.3.1 Quantitative Comparison in the Lab

In Table 4.2, we perform quantitative evaluation of our method in the HO3D dataset [39] and FPH dataset [29]. HO3D [39] dataset contains 3D annotations for both the hand and object of 68 video sequences, 10 subjects, and 10 objects. FPHA [29] dataset utilizes a MoCap system to capture hand-object interaction. 3D object pose annotations are available for 4 objects and subset of videos. We follow the same testing split as [43] for comparison.

Method for comparison We compare against the state-of-the-art (SOTA) approach [43] with the same input of monocular RGB image and the known 3D object

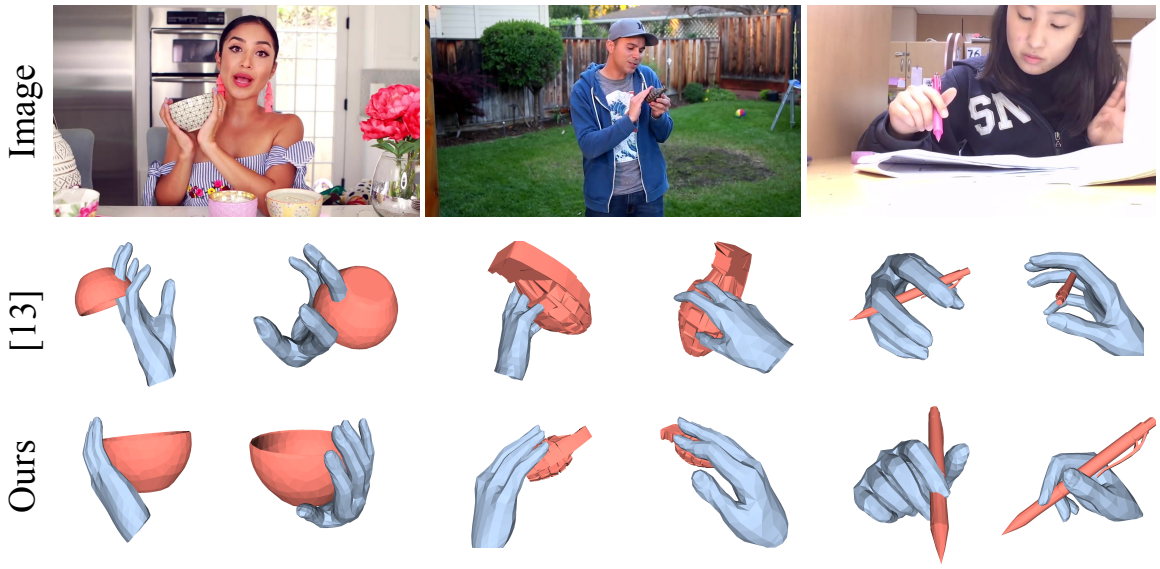


Figure 4.6: Qualitative comparison in the wild. Compared to existing method [43], our approach produces better hand-object reconstruction across diverse object categories.

model. [43] uses a feed-forward neural network to predict 3D hand pose and object pose where its single-frame model with full 3D supervision shows SOTA performance.

Evaluation metric. We report the mean average error (MAE) over 21 hand joints. The error measures the Euclidean distance between predictions and ground truth. Following [39], we calculate the error after aligning hand root position and global scale with the ground-truth.

For evaluating object pose, we calculate the Chamfer distance between ground-truth object vertices and predicted object vertices (obtained by rotating the input CAD model with the predicted object pose).

Results. Table 4.2 shows our method achieves better accuracy than [43] in 3D hand and object error. In HO3D dataset (left table), our predictions have smaller hand joint error of 9.7 mm *vs.* 14.7 mm and smaller object Chamfer distance of 19.9 *vs.* 26.8. In FPHA dataset (right table), our method achieves smaller hand joints error (14.2 mm *vs.* 18.0 mm). Our object error is slightly larger than [43]. The main reason is that [43] uses the action split of FPHA, i.e., same objects with different action labels are used for training and testing. In comparison, our method are tested directly without 3D supervision in those datasets.

4.3.2 Qualitative Comparison in the Wild

In Figure 4.6, we show side-by-side qualitative comparisons with [43] using in-the-wild images from [120], which clearly shows the advantage of our method. Though [43] achieves good performance in the lab, it struggles on in-the-wild images.

Metrics	[43]	Ours
Hand MAE ↓	14.7	**9.7**
Obj CF dist ↓	26.8	**19.9**

Metrics	[43]	Ours
Hand MAE ↓	18.0	**14.2**
Obj MAE ↓	**22.3**	23.9

Table 4.2: Quantitative comparison in the lab. Our method achieves results better or on par with the state of the art on popular in-the-lab datasets: HO3D (left) and FPHA (right).

	HO Distance ↓	Collision Score ↓
Individual results	414.8	0
+ Interaction loss	71.5	39.8
+ Depth loss	75.2	17.6
+ Penetration loss	76.4	7.7
+ Refinement	75.8	6.5

Table 4.3: Ablations on loss terms and pose refinement. From top to bottom, we add each component one by one (cumulative) and evaluate the prediction in terms of the distance between hand and object, and the collision score.

This was primarily due to the lack of labeled in-the-wild training data with diverse object categories. As a result, the model trained on limited object categories in the lab has difficulty in generalizing to new unseen objects.

In Figure 4.9, we show additional qualitative results of our method, called RHOI, on images from the 100 Days of Hands and the Epic Kitchens datasets. For each example, we show the 3D reconstructions from two different viewpoints. Overall, RHOI produces high quality reconstructions across a variety of scenarios and objects, e.g., holding a pen, grab a spoon/knife, touch a mobile phone, etc.

4.3.3 Ablation Studies

We now present the ablation studies (see also the supplement). We evaluate the influence of the joint optimization loss terms and the refinement stage on the overall results. We report the distance between the estimated hand and object centers and the collision score computed based on SDF. The more the object intersects with the hand the larger the collision score is. A good reconstruction should have small collision and small hand-object distance.

In Table 4.3, we observe that the individually reconstructed hand and object are far from each other, resulting in large distance and no collision. By adding the interaction loss, the distance decreases quickly to 71.5 mm but also results in a large collision, i.e, 39.8. Adding the depth and collision losses reduces the collision score to 7.7 while keeping a similar hand-object distance, i.e., 76.4 mm. The refinement stage

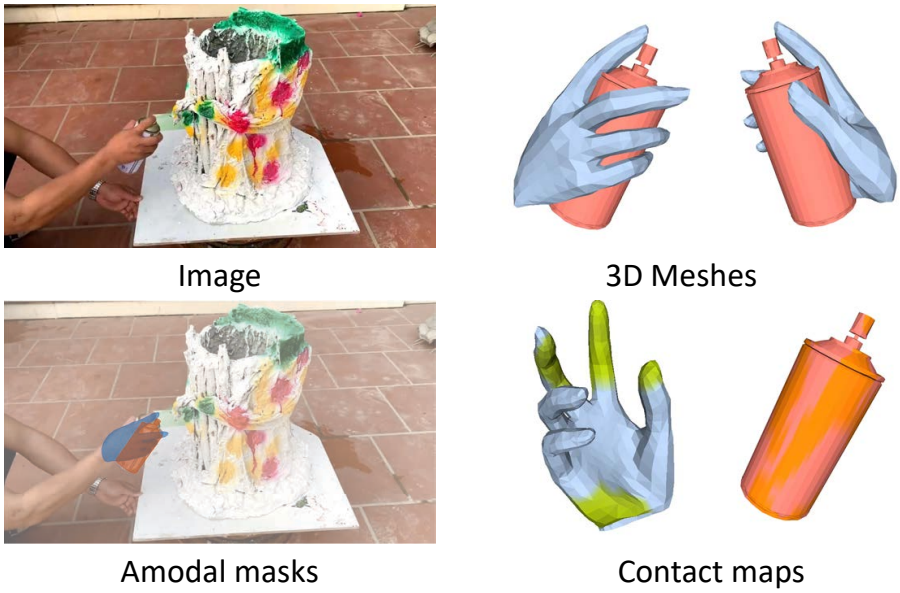


Figure 4.7: Example annotations. We use the techniques proposed in this chapter to annotate in-the-wild images and obtain 3D meshes, amodal masks, and contact maps.

makes small adjustment to the final result and can slightly reduce both the collision score (6.5 *vs.* 7.7) and hand-object distance (75.8 mm *vs.* 76.4 mm). These findings are consistent with visualization in Figure 4.4.

4.4 Dataset

We describe our dataset collection procedure and present the analysis that highlights the variety our data.

Image collections. As a source of in the wild data we use static frames from the EPIC Kitchens [21, 20] and the 100 Days of Hands [120] datasets, noting that we do not exploit any temporal information. These datasets contain a range of interesting hand-object interaction scenarios with varied objects, people, and viewpoints (both first- and third-person). To determine candidate images for reconstruction, we use a hand and object detector [120] and select images that contain a high bounding box overlap between an object and a hand.

4.4.1 Dataset Construction

Our annotation procedure consists of three steps: selecting a 3D object model, performing reconstruction using the method proposed in §4.2, and verification of the results.

Step 1: Model selection. The first step of our annotations requires the annotator

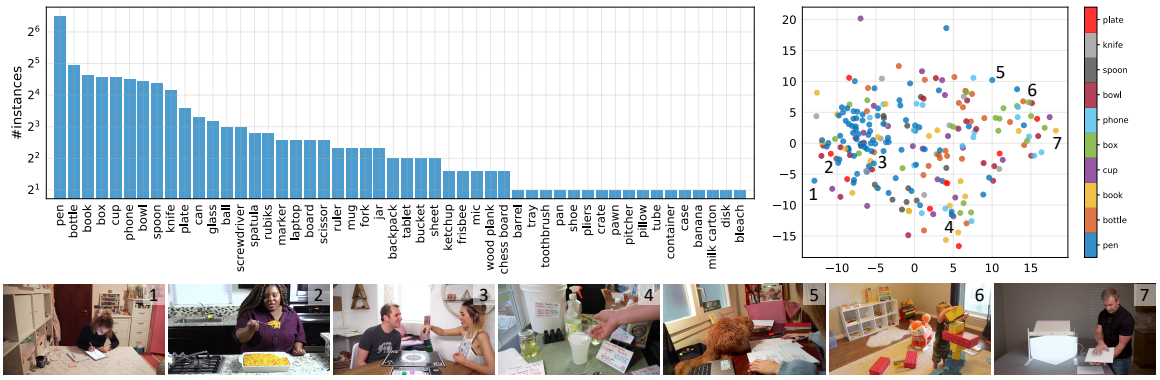


Figure 4.8: Variety of objects and grasps. We present analysis that shows the variety of objects and grasp types in our data. *Top left:* Our data contains *121 object categories* and a total of 500 instances. The object distribution has a long tail. *Top right:* We embed 3D hand poses into 2D space using Isomap [134]. Each point is a hand-object interaction and is color-coded by object category. We notice that there is a cluster of pens but no other clear clusters. This suggests that our data contains a variety of grasp types for each object category rather than only iconic grasps. Indeed, we see examples of different object categories with similar grasp types (pen and spoon) and same object category with different grasp types (pen). *Bottom:* We observe that the first embedding dimension (x axis) corresponds to the closure of the grasp. We show examples for increasing value of x. From left to right, we see that the grasps gradually transitions from fully closed to fully open.

to choose an appropriate 3D object model for the object being manipulated by the hand. We maintain a collection of available object models. If the required object is already present in the collection, the annotator directly selects the model. If not, the annotator finds an appropriate model online and adds it to the collection. Two primary sources of 3D object models we use are the YCB dataset [13] and the Free3D online platform.

Step 2: Reconstruction. Next, we perform the hand-object reconstruction using our method, called RHOI, proposed in §4.2. This step is semi-automatic and relies on the annotator to select the appropriate loss weights to obtain a good reconstruction. In practice, most annotators find that our default loss settings lead to a reasonable starting point.

Step 3: Verification. In practice, we find that RHOI results in good reconstructions in many cases. However, there are still cases where the results are imperfect across different viewpoints due to ambiguity. Thus, to ensure good annotation quality, we perform an additional step and verify the reconstructions obtained in step 2. Specifically, we ask the annotator to inspect the result from step 2 and take one of three possible actions: accept it if good, return it to step 2 if promising, and remove it from consideration if unlikely to improve. We iterate back and forth with step 2 until we converge to a set of reconstructions of reasonable quality.

Summary. To summarize, the output of our annotation procedure is that for each image we have: 3D object model, 3D object pose, and 3D hand pose. Moreover,

	Object IoU	Hand IoU	Quality	Match?
Large	0.84	0.67	-	-
Medium	0.78	0.69	-	-
Small	0.64	0.63	-	-
All	0.77	0.68	4.16	92%

Table 4.4: Dataset evaluation. *Left:* Amodal masks derived from our 3D annotations have a high overlap with ground truth amodal masks labeled by humans. *Right:* Users, asked to rate the quality of our 3D annotations from 1 to 5, find that they are of good quality on average and include a 3D object model that matches the true object in most cases.

we can easily derive additional annotations, such as amodal masks or contact maps. Example annotations are shown in Figure 4.7.

4.4.2 Dataset Evaluation

Annotating data in 3D is hard. Evaluating the quality of annotations is harder. To judge the quality of the collected annotations, we use two types of evaluation. The evaluation is performed on a sample set of 100 images.

Amodal mask accuracy. To evaluate our annotations, we require a signal that is predictive of reconstruction quality and can be labeled reliably by humans. Amodal instance masks, that include both visible and occluded parts of the object [76], are a good fit. Given only the visible portions of the image, there are many plausible configurations for the hidden object parts, especially for articulated objects like hands. Nevertheless, humans are capable of predicting occluded regions with high consistency [174].

We ask human annotators to label amodal masks for hands and objects, which serves as ground truth. We then compare amodal masks derived from our reconstructions to the ground truth. In Table 4.4, we report the mean intersection (IoU) scores for the hands and the objects. Similar to [80], we show results for different object sizes. We observe that our amodal masks have a high overlap with the ground truth. As expected, the overlaps are higher for larger objects.

User study. We also perform a user study. Given the input image and the annotated hand-object reconstruction, we ask the users to assign a quality score to each example on a scale of 1 to 5. The users are instructed to assign 1 when the reconstruction is poor (*e.g.* heavy collision or hand being far from the object) and 5 when there are no clear imperfections visible. The users can rotate the result in 3D visualization to view from different angles. We also ask the users to say if the object in the image matches the 3D model.

In Table 4.4, we report the results. The average reconstruction quality we obtain

is 4.16. This suggests that most of our annotations are of good quality. Moreover, we find that the 3D object model matches the true object in 92% of the cases. Among the 8%, most are due to imprecise mesh topology, *e.g.* a cylinder fitted to a mug with a handle.

4.4.3 Dataset Analysis

We annotated 500 images using the proposed procedure. We now present the analysis of the collected data.

Object variety. Our dataset contains *121 object categories* covering a wide variety of daily objects. In Figure 4.8, top-left, we show the object distribution for the 50 most frequent objects. There are some categories with many examples and a long tail of object categories with few examples.

Grasp variety. A unique feature of our data is that it provides a variety of hand-object interactions *in-the-wild*. This allow us to study and learn about human grasps using real-world data. In Figure 4.8, top-right, we embed 3D hand poses into 2D space using Isomap [134]. Each point corresponds to an interaction and is color-coded by object category.

We observe that there is a cluster of pens on the left but no other clear clusters. This suggests that our data contains a variety of grasp types for each object category, rather than only iconic grasps. Indeed, looking closer we notice that there are many examples of similar grasps for different object categories (*e.g.* pen and spoon) as well as different grasp types for the same object category (*e.g.* pen).

Grasp structure. We further discover an interesting pattern in the data. In particular, we find that the first dimension of the hand pose embedding (x axis) corresponds to the closeness of the hand. In Figure 4.8, bottom, we show example images for increasing value of x. We see that the grasps gradually transition from fully closed to fully open.

4.5 Discussion

In this chapter, we propose an optimization-based method that leverages 2D image cues and 3D contact priors for reconstructing hand-object interactions in the wild. Using the proposed method for semi-automatic labeling, we construct a new 3D hand-object interaction dataset in the wild. We hope that our effort attracts the community's attention to this challenging setting and facilitate our future progress.

Figure 4.9: Additional qualitative results. Our method, RHOI, produces strong results for a range of interactions and objects.

Chapter 5

Conclusion

In this thesis, we have presented a number of advances towards perceiving 3D humans and objects in motion. In Chapter 2, where we present an end-to-end learning system to perceive 3D scenes and independent object motions. We next show how 3D scenes influence human motion in Chapter 3, where we design a framework to predict future 3D human motion considering the scene context. In Chapter 4, we study the interaction between human hands and objects, where we introduce an optimization-based method to reconstruct the interaction in the wild.

While these are encouraging steps towards the goal of understanding rich interactions between humans, objects, and scenes, a number of challenges still remain. We will conclude the thesis by discussing some interesting future directions below.

Spatial-temporal scene graph: The 3D world is compositional, actionable, and evolving over time. One way to understand the dynamics in the 3D world is building a spatial-temporal scene graph to model the relationship between components. In this representation, we decompose the 3D scene into different modules, the scene layout/structure, a set of objects and humans represented in terms of their shape and pose, and motion. Moreover, the state of each object/person is influenced by other moving agents. A spatial-temporal scene graph can be built to understand the pattern of the scene dynamics. In this thesis, we have presented attempts in building pairwise connections between humans, scenes, and objects components. It would be an interesting direction to develop a joint framework that perceives all components and their rich interactions in this scene graph representation.

Model 3D human full-body contact: We humans interact with the world and our bodies often have contact with other people, objects, or even self-contact such as holding our arms. These contact scenarios result in occlusion and make it hard for 3D reconstruction. Some recent optimization-based approaches [42, 55] proposed to enforce geometry constraints to model the contact, however, they either require

the extra input of 3D reconstructed scene mesh or require manually labeling human-object mesh vertices. In Chapter 4, we propose to make advantage of 3D hand-object contact priors learned from available 3D MoCap data. This enables us to deal with different scenarios in the wild without extra input. It is natural to extend the idea of learning contact priors to the case of full human body for more fine-grained 3D reconstruction of humans during interactions.

Learn object functionality through interactions: Understanding objects is more than estimating their instance segmentation, reconstructing their 3D shape and texture. There are much richer properties about the object including their structure, whether they are rigid, deformable, or articulated, and more importantly, their functionality (affordance as defined by Gibson [34]). These properties are extremely useful for robotic application such as manipulation. Children learn object functionality by interacting with them and observe the consequence of their action. Could we develop a pipeline for learning the object functionality using lots of videos of humans interacting with the scenes and objects? Could we further develop an active learning system where an actual robot is exploring and interacting with the world? These are all interesting but unsolved problems that are worth exploring in the future.

Bibliography

[1] `CMUMotionCaptureDatabase.http://mocap.cs.cmu.edu`. 22

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, 2016. 13

[3] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In *NIPS*, 2009. 22

[4] Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yaser Sheikh. Bilinear spatiotemporal basis models. *SIGGRAPH*, 2012. 22

[5] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016. 21

[6] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. Socially-aware large-scale crowd forecasting. In *CVPR*, 2014. 21

[7] Christos Alexopoulos and Paul M Griffin. Path planning for a mobile robot. *IEEE Transactions on systems, man, and cybernetics*, 1992. 20

[8] Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. ContactDB: Analyzing and predicting grasp contact via thermal imaging. In *CVPR*, 2019. 38

[9] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. ContactPose: A dataset of grasps with object contact and hand pose. In *ECCV*, 2020. 36, 37, 38

[10] Matthew Brand and Aaron Hertzmann. Style machines. *SIGGRAPH*, 2000. 22

[11] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, 2000. 5

[12] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, 2018. 38

[13] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *ICRA*, 2015. 48

[14] Zhe Cao, Gines Hidalgo Martinez, Tomas Simon, Shih-En. Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *TPAMI*, 2019. 40

[15] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, 2019. 21

[16] Yu-Wei Chao, Jimei Yang, Brian Price, Scott Cohen, and Jia Deng. Forecasting human dynamics from static images. In *CVPR*, 2017. 21

[17] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. *ICCV*, 2019. 22

[18] Hsu-kuang Chiu, Ehsan Adeli, Borui Wang, De-An Huang, and Juan Carlos Niebles. Action-agnostic human pose forecasting. In *WACV*, 2019. 21

[19] João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *IJCV*, 1998. 5

[20] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv:2006.13256*, 2020. 47

[21] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 37, 47

[22] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 3, 5

[23] Ahmed Elhayek, Carsten Stoll, Nils Hasler, Kwang In Kim, H-P Seidel, and Christian Theobalt. Spatio-temporal motion tracking with unsynchronized cameras. In *CVPR*, 2012. 22

[24] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *ECCV*, 2018. 22, 26

[25] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox". Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 3, 5, 16

[26] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *ICCV*, 2015. 19, 22

[27] Katerina Fragkiadaki, Marta Salas, Pablo Arbelaez, and Jitendra Malik. Grouping-based low-rank trajectory completion and 3d reconstruction. In *NIPS*, 2014. 5

[28] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 2010. 5

[29] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*, 2018. 36, 38, 44

[30] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 3, 5

[31] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010. 10, 17

[32] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011. 6, 14

[33] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *3DV*, 2017. 21

[34] James J Gibson. The theory of affordances. *Hilldale, USA*, 1977. 35, 53

[35] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 12

[36] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *ICCV*, 2019. 1, 38

[37] C Godard, O Mac Aodha, and GJ Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 3, 5, 10, 13, 14, 15, 17

[38] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 21

[39] Shreyas Hampali, Mahdi Rad Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 36, 37, 38, 44, 45

[40] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[41] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *ICCV*, 2019. 20, 22, 26, 27, 33

[42] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *ICCV*, 2019. 39, 52

[43] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 2020. 38, 44, 45, 46

[44] Yana Hasson, Gül Varol, Dimitris Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 38

[45] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 12

[46] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 1995. 21

[47] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno-Noguer. Human motion prediction via spatio-temporal inpainting. In *CVPR*, 2019. 22

[48] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asian Technical Briefs*, 2015. 22

[49] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *ICCV*, 2007. 5

[50] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 3, 16

[51] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2013. 20, 22, 29

[52] Mariano Jaimez, Mohamed Souiai, Javier Gonzalez-Jimenez, and Daniel Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In *ICRA*, 2015. 5

[53] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *CVPR*, 2016. 22

[54] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *ICCV*, 2017. 5

[55] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *CVPR*, 2020. 39, 52

[56] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *ICCV*, 2015. 18

[57] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1

[58] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, 2017. 5, 11, 12

[59] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018. 40

[60] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 3, 5

[61] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 13

[62] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014. 24

[63] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 41

[64] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*, 2012. 21

[65] Philipp Krähenbühl. Free supervision from video games. In *CVPR*, 2018. 22, 26

[66] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. 38

[67] Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael Bronstein, and Stefanos Zafeiriou. Single image 3d hand reconstruction with mesh convolutions. *BMVC*, 2019. 38

[68] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018. 38

[69] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2cad: 3d shape prediction by learning to segment and retrieve. *ECCV*, 2020. 38

[70] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *CVPR*, 2014. 5

[71] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006. 20

[72] Hei Law, Yun Teng, Olga Russakovsky, and Jia Deng. Cornernet-lite: Efficient keypoint based object detection. *arXiv preprint arXiv:1904.08900*, 2019. 25

[73] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *NIPS*, 2018. 22

[74] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *CGF*, 2007. 21

[75] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *CVPR*, 2018. 22

[76] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*, 2016. 49

[77] Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *CVPR*, 2019. 22

[78] Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *ICLR*, 2018. 22

[79] Joseph J Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *ICCV*, 2013. 38

[80] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 39, 41, 49

[81] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 9, 15

[82] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981. 5

[83] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv preprint arXiv:1810.06125*, 2018. 5, 17

[84] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *CVPR*, 2017. 21

[85] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *CVPR*, 2019. 21

[86] Priyanka Mandikal and Kristen Grauman. Dexterous robotic grasping with object-centric visual affordances. *arXiv:2009.01439*, 2020. 35

[87] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, 2017. 19, 22

[88] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 3, 5

[89] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. *AAAI*, 2018. 13, 14, 16

[90] Andrew N Meltzoff. Understanding the intentions of others: re-enactment of intended acts by 18-month-old children. *Developmental psychology*, 1995. 35

[91] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 4, 5, 12

[92] Frank Michel, Alexander Kirillov, Eric Brachmann, Alexander Krull, Stefan Gumhold, Bogdan Savchynskyy, and Carsten Rother. Global hypothesis generation for 6d object pose estimation. In *CVPR*, 2017. 38

[93] Aron Monszpart, Paul Guerrero, Duygu Ceylan, Ersin Yumer, and Niloy J Mitra. imapper: Interaction-guided joint scene and human motion mapping from monocular videos. *SIGGRAPH*, 2019. 22

[94] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *CVPR*, 2018. 38

[95] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*, 2018. 38

[96] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 25

[97] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*, 2011. 38

[98] Kemal Egemen Ozden, Kurt Cornelis, Luc Van Eycken, and Luc Van Gool. Reconstructing 3d trajectories of independently moving objects using generic constraints. *CVIU*, 2004. 5

[99] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 27, 38

[100] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, 2019. 29, 30, 31, 32

[101] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. In *BMVC*, 2018. 22

[102] Vladimir Pavlovic, James M Rehg, and John MacCormick. Learning switching linear models of human motion. In *NIPS*, 2001. 22

[103] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *CVPR*, 2009. 21

[104] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Trans. Graph.*, 2018. 35

[105] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *IJCV*, 2004. 5

[106] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *CVPR*, 2018. 40

[107] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. *arXiv:2004.04650*, 2020. 35

[108] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 42

[109] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 16

[110] Anurag Ranjan, Varun Jampani, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Adversarial collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. *arXiv preprint arXiv:1805.09806*, 2018. 5, 16

[111] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 16

[112] Javier Romero, Hedvig Kjellström, and Danica Kragic. Hands in action: real-time 3d reconstruction of hands in interaction with objects. In *ICRA*, 2010. 38

[113] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *TOG*, 2017. 38, 40

[114] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv:2008.08324*, 2020. 38, 40

[115] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 12

[116] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 16

[117] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *CVPR*, 2019. 21

[118] Mihir Sahasrabudhe, Zhixin Shu, Edward Bartrum, Riza Alp Guler, Dimitris Samaras, and Iasonas Kokkinos. Lifting autoencoders: Unsupervised learning of a fully-disentangled 3d morphable model using deep non-rigid structure from motion. In *ICCV*, 2019. 38

[119] Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. PiGraphs: Learning Interaction Snapshots from Observations. *TOG*, 2016. 20, 22, 26

[120] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 37, 41, 45, 47

[121] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *CHI*, 2015. 38

[122] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 40

[123] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *ECCV*, 2016. 38

[124] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV*, 2013. 38

[125] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018. 24

[126] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018. 38

[127] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 13

[128] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, 2015. 38

[129] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *ECCV*, 2020. 36, 37, 38, 44

[130] Lei Tai, Jingwei Zhang, Ming Liu, and Wolfram Burgard. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In *ICRA*, 2018. 21

[131] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Fast multi-frame stereo scene flow with motion segmentation. In *CVPR*, 2017. 5

[132] Meng Keat Christopher Tay and Christian Laugier. Modelling smooth paths using gaussian processes. In *Field and Service Robotics*. Springer, 2008. 22

[133] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+o: Unified egocentric recognition of 3d hand-object poses and interactions. In *CVPR*, 2019. 38

[134] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000. 48, 50

[135] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. In *TOG*, 2006. 21

[136] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018. 38

[137] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 5

[138] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 11

[139] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 2016. 38

[140] Shimon Ullman. Maximizing rigidity: The incremental recovery of 3-d structure from rigid and nonrigid motion. *Perception*, 1984. 5

[141] Raquel Urtasun, David J Fleet, Andreas Geiger, Jovan Popović, Trevor J Darrell, and Neil D Lawrence. Topologically-constrained latent variable models. In *ICML*, 2008. 22

[142] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 25, 29, 30, 31

[143] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *ICCV*. IEEE, 1999. 5

[144] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. Technical report, arXiv:1704.07804, 2017. 5

[145] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017. 21

[146] Minh Vo, Srinivasa G Narasimhan, and Yaser Sheikh. Spatiotemporal bundle adjustment for dynamic 3d reconstruction. In *CVPR*, 2016. 22

[147] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *ICCV*, 2013. 5

[148] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018. 22

[149] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *CVPR*, 2017. 21

[150] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *TPAMI*, 2007. 22

[151] Jack M Wang, David J Fleet, and Aaron Hertzmann. Multifactor gaussian process models for style-content separation. In *ICML*, 2007. 22

[152] Xiaolong Wang, Rohit Girdhar, and Abhinav Gupta. Binge watching: Scaling affordance learning from sitcoms. In *CVPR*, 2017. 22

[153] Zhe Wang, Liyan Chen, Shaurya Rathore, Daeyun Shin, and Charless Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. *arXiv preprint arXiv:1905.07718*, 2019. 20, 22

[154] Zhe Wang, Daeyun Shin, and Charless C Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. *arXiv preprint arXiv:2004.03143*, 2020. 22

[155] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 7

[156] Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers. Efficient dense scene flow from sparse or dense stereo data. In *ECCV*, 2008. 5

[157] Mao Wei, Liu Miaomiao, Salzemann Mathieu, and Li Hongdong. Learning trajectory dependencies for human motion prediction. In *ICCV*, 2019. 19, 22, 29, 30, 31, 32

[158] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *CVPR*, 2019. 21

[159] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*, 2019. 38

[160] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *RSS*, 2018. 38

[161] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *CVPR*, 2019. 38

[162] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding. *arXiv preprint arXiv:1806.10556*, 2018. 5, 17

[163] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 5, 11

[164] Qi Ye, Shanxin Yuan, and Tae-Kyun Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *ECCV*, 2016. 38

[165] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 3, 5, 9, 13, 14, 16, 17

[166] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *IROS*, 2018. 21

[167] Chang Yuan and Gerard Medioni. 3d reconstruction of background and objects moving on ground plane viewed from a moving camera. In *CVPR*, 2006. 5

[168] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *CVPR*, 2018. 38

[169] Jason Y. Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *ICCV*, 2019. 19, 21

[170] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild. In *ECCV*, 2020. 38, 39

[171] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*, 2018. 21

[172] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 1, 5, 15, 16

[173] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *CVPR*, 2020. 38

[174] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 49

[175] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017. 38

[176] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 2019. 36, 38

[177] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018. 5, 9, 13, 16