Understanding 3D Object Articulation in Internet Videos

Shengyi Qian   Linyi Jin   Chris Rockwell   Siyi Chen   David F. Fouhey

University of Michigan

{syqian,jinlinyi,cnris,siyich,fouhey}@umich.edu

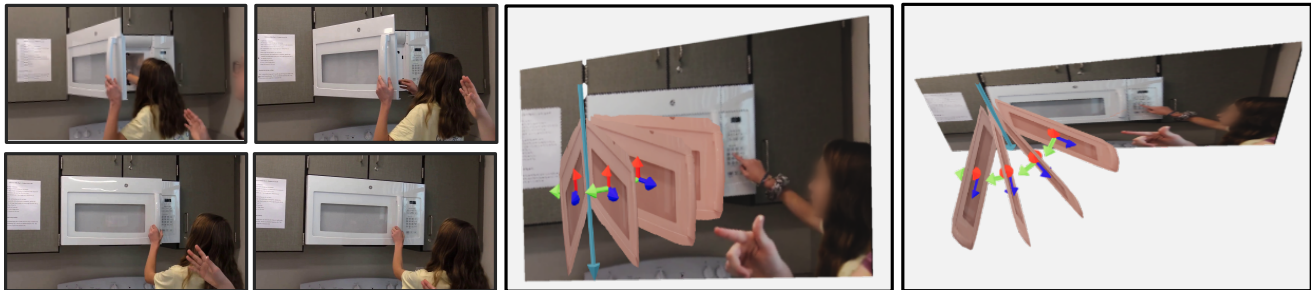https://jasonqsy.github.io/Articulation3D

Figure 1. Given an ordinary video, our system produces a 3D planar representation of the observed articulation. The 3D renderings illustrate how the microwave (in **Pink**) can be articulated in 3D space. We also show the predicted rotation axis using a **Blue** arrow.

Abstract

We propose to investigate detecting and characterizing the 3D planar articulation of objects from ordinary RGB videos. While seemingly easy for humans, this problem poses many challenges for computers. Our approach is based on a top-down detection system that finds planes that can be articulated. This approach is followed by optimizing for a 3D plane that explains a sequence of detected articulations. We show that this system can be trained on a combination of videos and 3D scan datasets. When tested on a dataset of challenging Internet videos and the Charades dataset, our approach obtains strong performance.

1. Introduction

How would you make sense of Figure 1? Behind the set of RGB pixels that make up the video is a real 3D transformation consisting of a 3D planar door rotating about an axis. The goal of this paper is to give the same ability to computers. We focus on planar articulation taking the form of a rotation or translation along an axis. This special case of articulation is ubiquitous in human scenes and understanding it lets a system understand objects ranging from refrigerators and drawers to closets and cabinets. While we often learn about these shapes and articulations with physical embodiment [50], we have no difficulty understanding them from video cues alone, for instance while watching a movie or seeing another person perform an action. We formalize this ability for computers as recognizing and characterizing a class-agnostic planar articulation via a 3D planar segment, articulation type (rotation or translation), 3D articulation axis, and articulation angle.

This problem is beyond the current state of the art in scene understanding since it requires reconciling single image 3D understanding with *dynamic 3D* understanding. While there has been substantial work on 3D reconstruction from a single image [4, 9, 12, 61], including work dedicated to planes [33], these works focus on reconstructing static scenes. On the other hand, while there has been work understanding articulation, these works often require the placement of tags for tracking [36, 41], a complete 3D model or depth sensor [20, 30, 39], or successful 3D human reconstruction [65]. Moreover, making progress is challenging because of data. Unsupervised approaches based on motion analysis [42, 51] require something to track, which breaks in realistic data since many human-made articulated objects are untextured (e.g., refrigerators) or transparent (e.g., ovens). While supervised approaches [30, 38, 39] can perhaps bypass tracking features, they seemingly require access to large amounts of RGBD data of interactions. For now, this data does not exist, and training on synthetic data can fall short when tested on real data (as our experiments empirically demonstrate).

We overcome these challenges with a learning-based approach that combines both detection and 3D optimization

and is trained with supervision from multiple sources (Section 4). The foundation of our approach is a top-down detection approach that recognizes articulation axes and types and 3D planes; this approach's outputs are processed with an optimization method that seeks to explain the per-frame results in terms of a single coherent 3D articulation.

Via this model, we show that one can build an understanding of 3D object dynamics via a mix of 2D supervision on Internet videos of objects undergoing articulation as well as 3D supervision on existing 3D datasets that do not depict articulations. To provide 2D supervision, we introduce (Section 3) a new set of 9447 Creative Commons Internet videos. These videos depict articulation with a variety of objects as well as negative samples and come with sparse frame annotations of articulation boxes, axes, and surface normals that can be used for training and evaluating planar articulation models.

Our experiments (Section 5) evaluate how well our approach can recognize and characterize articulation. We evaluate on our new dataset of videos as well as the Charades [48] dataset. We compare with a variety of alternate approaches, including bottom-up signals like optical flow [53] and changes in surface normals [3], training on synthetic data [64], as well as systems that analyze human-object interaction [65]. Our approach outperforms these approaches on our data, often even when the baselines are given access to ground-truth location of articulation.

Our primary contributions include: (1) The new task of detecting 3D object articulation on unconstrained ordinary RGB videos without requiring RGBD video at training time; (2) A dataset of Internet videos, with sparse frame annotations of articulation boxes, axes, and surface normals that can be used for training and evaluating planar articulation models; (3) A top-down detection network and optimization to tackle this problem, which has strong performance on the Internet video dataset and Charades.

2. Related Work

Our paper proposes to extract 3D models of articulation from ordinary RGB videos. This problem lies at the intersection of 3D vision, learning from videos, and touches on robotics applications. We note that there are specialized approaches for understanding *general* articulation (e.g., non-rigid structure from motion [55]) as well as for understanding specialized motion models (e.g., for a full human 3D mesh models [69] or quadrupeds [28]) or for understanding more general transformations [19, 59]. Our work focuses on understanding the articulation of general objects whose articulated pieces can be represented by a 3D plane rotating or translating.

Due to the ubiquitous nature of articulated objects, the task of understanding them has long been an interest across all of artificial intelligence. In vision, the understanding of

the motion of rigid objects undergoing transformations was one of the early successes of computer vision [24, 54, 58]. Unfortunately, these early works rely on reliable motion tracks, which is made difficult by the textureless or reflective nature of many indoor planes (e.g., refrigerator doors). Our top-down detector gives 3D planes that can help provide correspondence between frames where correspondence is challenging.

More recent work in robotics has used the value of 3D and integrated it into their modeling approaches [5, 8, 37, 42, 51]; however their approaches often use an RGBD sensor, unlike our use of ordinary RGB sensors. This dependence on RGBD has been carried forward to the most recent work that uses deep learning frameworks [1, 20, 30, 35, 62, 64]. In fact, some methods require full 3D models [39], which is typically unavailable in real world 3D scans. [38] by Mo et al. can be run on 2D images as long as the point cloud encoder is replaced with a RGB encoder, but its 2D images contain a single object without any background, instead of challenging Internet videos. While there has been increasing amounts of work aimed at virtual articulated objects [52, 64], simultaneously achieving scale and quality is challenging. For instance ReplicaCAD [52] has only 92 objects. In contrast, our approach works at test time on standard RGB videos by bringing its own 3D via a learned detector [33] trained on RGBD data [6].

While our outputs are 3D planar regions, our approach is deeply connected to the task of understanding human-object interactions. In these works [2, 13, 47], the goal is to recognize the relationship between humans and the objects they interact with. The interactions that we study are caused by these humans, and so we use an approach that can predict human-object interactions [47] to help identify the data we train our systems on. The most related work in this area is [65], which aims to jointly understand dynamic 3D human-object interactions in 3D. This work, however assumes that the object CAD model is known once the articulated object is detected, which we do not need. Our method also works with articulation videos that are more varied in viewpoint and perspective. A more thorough understanding of the joint relationship between articulated objects and human-object interaction, akin to early work [10, 27], is beyond the scope of this work, but of future value.

We solve the problem of describing 3D articulation by producing 3D planar models. This uses advances in 3D from a single image. In particular, we build on PlaneR-CNN [33], which is part of a growing body of works aimed at extracting planes from single images [34, 66, 67]. These planes have advantages for the articulation reasoning since they offer a compact representation to track and describe. While we use plane recognition, the plane is just one component of our output (along with rotation axes) and we analyze our output in a video with temporal optimization.

3. Dataset

One critical component of our approach is accurate 2D annotations of articulation occurring in RGB data. We show that these 2D annotations can be combined with existing RGBD data and the right method to build systems that understand 3D articulation on video data. We next describe how we collect a dataset of articulations. Our goals are to have a large collection of annotations of object box, articulation type, and axis. Rather than directly look for examples of people articulating objects, we follow the data-first approach of [7,11,47,68], namely to gather data containing many related activities and then analyze and annotate it.

Data Collection. Our pipeline generates a set of candidate clips to be annotated from a collection of candidate videos via an automatic pipeline that aims to eliminate frames that are easy to see as not depicting articulation. We begin with candidate videos that are found by variants of searches for a set of 10 objects among Creative Commons videos on YouTube. Within these videos, we find stationary continuous shots in these videos with homographies [15] fit on ORB [46] features. Many of these clips cannot depict interaction since they do not contain any people or do not contain the objects of interest. We filter by responses by a hand detector [47] trained on 100K+ frames of Internet data, as well as object detectors trained on COCO [32] and LVIS [14]. These filtering steps maximize the use of annotator time by eliminating clear negatives, and generate a large number of candidate clips.

With a collection of candidate clips of interest, we then turn to manual annotation. For a given clip, we hire an annotation company to annotate frames sparsely (every 10 frames) within the clip. They annotate: (**box**) a box around the articulated plane and its type, if it exists; and (**axis**) the projection of the articulation axis, framed as a line segmentation annotation problem. This results in a set of 19411 frames, containing 19411 boxes around articulating planes with 13508 rotation axes and 2755 translation axes, as well as 39411 negative frames. The number of articulation axes is not equal to the number of boxes, since some articulation axes are outside the image. We provide training, validation, and test splits based on uploader, leading to 7845/601/1001 videos in the train/val/test split. A more complete description of our annotation pipeline appears in the supplement.

We collect two additional annotations. For the test set, we also annotate the surface normal of the plane following [3], so we can evaluate how well our model can learn 3D properties. To show generalization, we also collect the same annotations except surface normals on the Charades [48] dataset.

Data Availability and Ethics. Our data consists of videos that users uploaded publicly and chose to share as Creative Commons data. These do not involve interaction with hu-mans or private data. We filtered obviously offensive content, videos depicting children, and cartoons. Examples appear throughout the paper; screenshots of annotation instructions and details appear in the supplement.

4. Approach

The goal of our approach is to detect and characterize planar articulation in an unseen RGB video clip. These articulations are an important special case that are ubiquitous in human scenes. As shown in Figure 2, we propose a *3D Articulation Detection Network* (3DADN) to solve the task. As output, the 3DADN produces the type of motion (rotation or translation), a bounding box around where the motion is located, the 2D location of the rotation or translation axis, and the 3D location of the articulated plane. The 3DADN's output is followed by post-processing to find a consistent explanation over the whole video.

4.1. 3D Articulation Detection Network

The 3DADN processes each frame independently. Its output consists of: a segment mask $\mathcal{M}_i$; plane parameters $\boldsymbol{\pi}_i = [\mathbf{n}_i, o_i]$ giving the plane equation $\boldsymbol{\pi}_i^T[x, y, z, -1] = 0$ (where $\mathbf{n}_i$ is the the plane normal with $||\mathbf{n}_i||_2 = 1$ and o_i is the plane offset); a projected rotation or translation axis $\mathbf{a}_i = [\theta, p]$ which is the projection of the 3D articulation axis; and articulation type.

We use a top-down approach to detect this representation, which we train on RGB videos that depict articulation *without* 3D information as well as RGBD images *with* 3D information that do not depict articulation. Our backbone is a Faster R-CNN [44] style network that first detects bounding boxes for the articulating objects and classifies them into two classes (rotation and translation). These boxes provide ROI-pooled features that are passed into detection heads that predict our outputs ($\mathcal{M}_i, \boldsymbol{\pi}_i, \mathbf{a}_i$). Our heads and losses for $\mathcal{M}_i$ follow the common practice of Mask R-CNN [16]. We describe $\mathbf{a}_i$ and $\boldsymbol{\pi}_i$ below.

Parameterizing Rotation and Translation Axis. We model the projected articulation axis as a 2D line in the image. This projected axis is the projection of the 3D articulation axis (e.g., the hinge of a door). We describe the projected axis with the normal form of the line, $x \cos(\theta) + y \sin(\theta) = p$ where $p \geq 0$ is the distance from the box to the center and θ is the inclination of the normal of the axis in pixel coordinates. Since a translation corresponds to a direction/family of lines as opposed to a line, we define $p = 0$ for translation arbitrarily.

The articulation head contains two independent branches for predicting the rotation and translation axes. We handle the circularity of the prediction of θ by lifting predictions and ground-truth for the angle to the 2D unit circle; since the line is 180-degree-ambiguous (i.e., $\theta + \pi$ is the same as
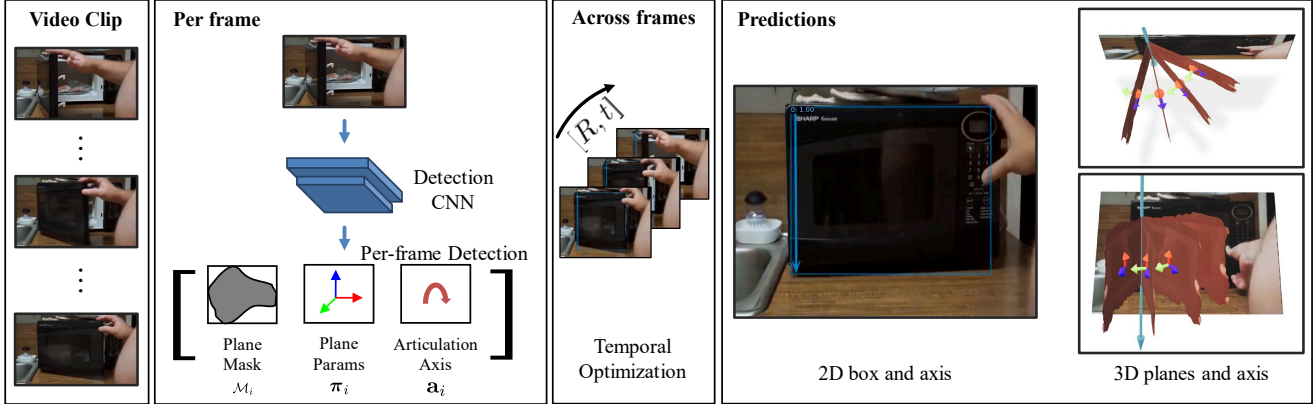
Figure 2. Overview of our approach. (a) Given an ordinary video clip, we first apply our 3D Articulation Detection Network (3DADN) to detect 3D planes can be articulated for each frame. (b) We then apply temporal optimization to fit the articulation model. Final results are demonstrated in both 2D image and 3D rendering.

θ), we predict a 2D vector $[\sin(2\theta), \cos(2\theta)]$. The resulting network thus predicts a 3D vector containing θ and p, which we supervise with a L1 loss.

Parameterizing Plane Parameters. Following a line of work on predicting planes in images, we use a 3D plane [34] to represent the 3D locations of the articulated objects, since many common articulated objects like doors, refrigerators, and microwaves can be modeled as planes and because past literature [21, 22, 33] suggests that R-CNN-style networks are adept at predicting plane representations.

A 3D plane is represented by plane parameters $\pi_i = [\mathbf{n}_i, o_i]$ giving the plane equation $\pi_i^T[x, y, z, -1] = 0$ With camera intrinsics, planes can be recovered in 3D and with a mask, this plane can be converted to a plane segment. Following [22, 33], we extend R-CNN by adding a plane head which directly regresses the normal of the plane. A depth head is used to predict depth of the image. The depth is only used to calculate the offset value of the plane. We supervise with L2 loss for the plane normal regression and L1 loss for the depth regression.

Training. There is no dataset that is non-synthetic and large enough to train a 3DADN directly: the 3DADN needs both realistic interactions and 3D information. However, we can train the 3DADN in parts. In the first stage, we train the backbone, RPN, and axis heads directly on our Internet video training set, which has boxes and axes. We then freeze the backbone, RPN, and axis heads and fine-tune the mask and plane head on a modified version of ScanNet [6].

In particular, we found that humans often occlude the objects they articulate and models that had not seen humans in training produced worse qualitative results. We therefore composited humans from SURREAL [57] into the scenes. We randomly sample 98,235 ScanNet images, select a synthetic human and render it on ScanNet backgrounds. In training, we do not change the ground truth, pretending the ground truth plane is partially occluded by humans and training our model to identify them.

Meanwhile, we found that the order of training the heads was crucial. Planes in ScanNet [6] are defined geometrically, and so unopened doors often merge with walls; similarly, ScanNet [6] does not contain transitional moments during which planes are articulating. Thus, RPNs trained on ScanNet [6] perform poorly on articulation videos. Instead, it is important to train the RPN on our Internet videos, freeze the backbone, and only rely on ScanNet to train plane parameters and masks, which are unavailable in Internet videos. During inference we keep the ScanNet camera since our data does not have camera intrinsics.

Implementation Details. Full architectural details of our approach are in the supplemental. Our model is implemented using Detectron2 [63]. The backbone uses ResNet50-FPN [31] pretrained on COCO [32].

4.2. Temporal Optimization

After the 3DADN provides per-frame estimates of articulations, we perform temporal optimization to find a single explanation for the detections across frames. We are given a sequence of detections indexed by a time of the form $[\mathcal{M}_i^{(t)}, \pi_i^{(t)}, \mathbf{a}_i^{(t)}]$. We aim to find a single consistent explanation for these detections.

Tracking. Optimizing requires a sequence of planes to optimize over. We match box i with the box in the next frame according to pairwise intersection over union (IoU). Box i at t matches box $j = \arg\max_{j'} \text{IoU}(\mathcal{M}_i^{(t)}, \mathcal{M}_{j'}^{(t+1)})$ at time $t + 1$; we then track greedily to get a sequence. We subsequently drop the subscripts for clarity.

Articulation Model Fitting. Given a sequence of detections, we find a consistent explanation via a RANSAC-like

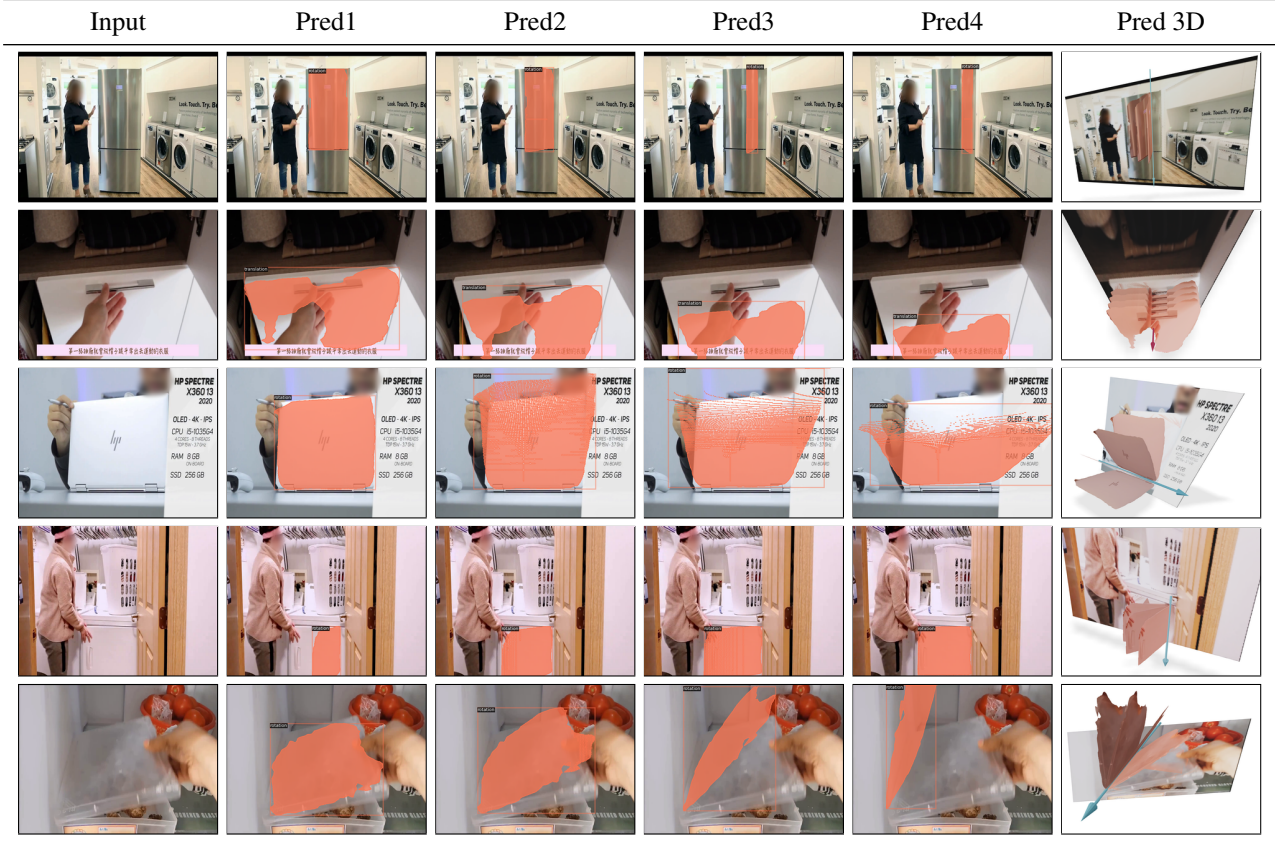| Input	Pred1	Pred2	Pred3	Pred4	Pred 3D

Figure 3. Predictions on Internet videos. For each example, we show the input (left), detected 2D planes and how they will be articulated using the predicted articulation axes and surface normals (middle). We also show 3D renderings to illustrate how these common objects are articulated in the 3D space (right). The predicted rotation axis is shown as the **Blue** arrow, and translation axis is the **Pink** arrow.

approach. We begin with a hypothesis of a plane segment $\boldsymbol{\pi}$ and articulation axis $\mathbf{a}$, which we obtain by selecting output on a reference frame. Along with an assumed camera intrinsics $\mathbf{K}$, the plane parameters let us lift the plane segment and axis to 3D, producing 3D plane segment $\boldsymbol{\Pi}$ and 3D axis $\mathbf{A}$. Then, for each frame t, we solve for an articulation degree $\alpha^{(t)}$ maximizing the reprojection agreement with the predicted mask at time t. Let us define the reprojection score as

$$r(\alpha, t) = \text{IoU}\left(\mathcal{M}^{(t)}, \mathbf{K}\left[\mathbf{R}_\alpha, \mathbf{t}_\alpha\right]\boldsymbol{\Pi}\right), \quad (1)$$

where $\mathbf{R}_{\mathbf{A},\alpha}$ and $\mathbf{t}_{\mathbf{A},\alpha}$ are α steps over the rotation and translation for axis $\mathbf{A}$. We then solve for $\alpha^{(t)}$ by solving $\arg\max_\alpha r(\alpha)$, which gives a per-frame angle using grid search. We detect articulation by calculating how well the rotation degree $\alpha^{(t)}$ can be explained as a linear function of t (i.e., that there is constant motion). Since many scenes are not constant motion, we have loose thresholds: we consider $R^2 \geq 0.4$ and slope $k > 0.1$ to be an articulation. We exclude hypotheses where all $r(\alpha^{(t)}, t) < 0.5$.

5. Experiments

We have introduced a method that can infer 3D articulation in Section 4. In the experiments, we aim to answer the following questions: (1) how well can one detect 3D articulating objects from ordinary videos; (2) how well do alternate approaches to the problem do?

5.1. Experimental Setup

We first describe the setup of our experiments. Our method aims to look at an ordinary RGB video and infer information about an articulated plane on a object in 3D, including: whether the object is articulating, its extent, and the projection of its rotation or translation axis. We therefore evaluate our approach on two challenging datasets, using metrics that capture various aspects of a 3D plane articulating in 3D.

Datasets: We validate our approach on both Internet videos (described in Section 3) and the Charades dataset [49]. We use Charades for cross-dataset evaluations. We focus on Charades videos that are opening objects (doors, refrigera-

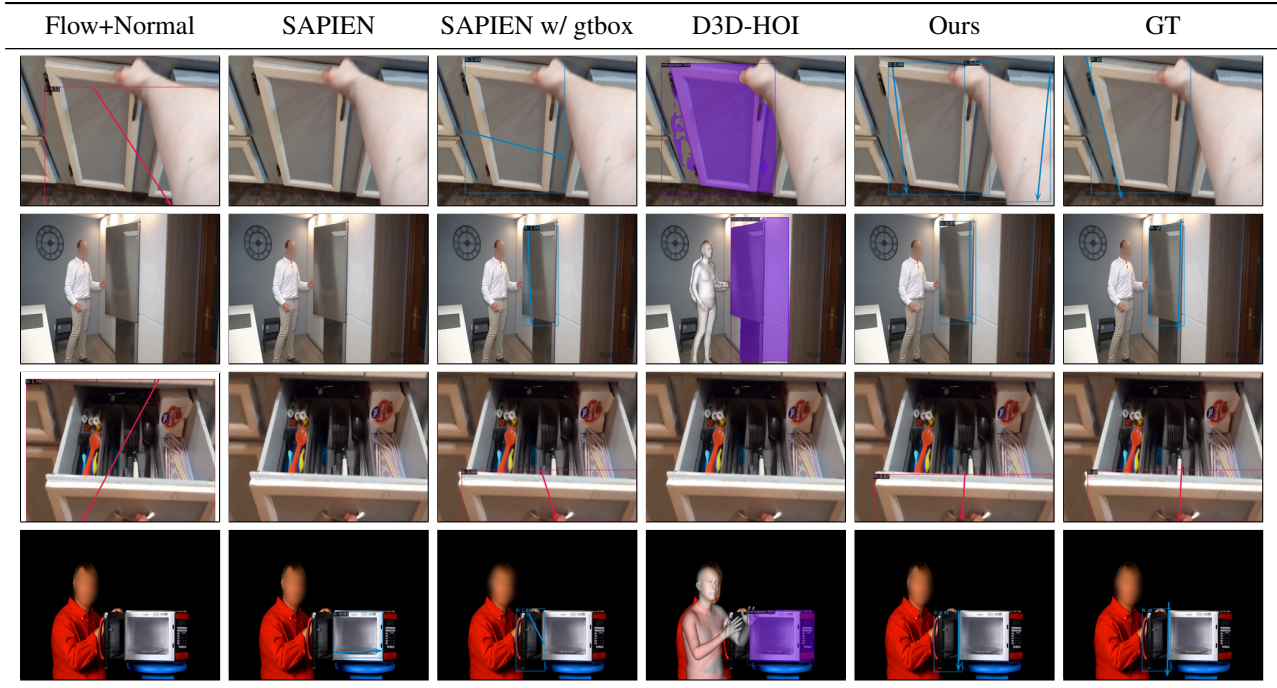| Flow+Normal	SAPIEN	SAPIEN w/ gtbox	D3D-HOI	Ours	GT

Figure 4. We compare our approach with four baselines. See detailed discussions in the text. We show translation in **Pink** and rotation in **Blue**, except D3D-HOI which uses a different detector.

tors, etc.), and annotate 2491 frames across 479 videos; we also randomly sample 479 negative videos containing 4401 negative frames. Our Charades annotation process is similar to Internet videos, with the exceptions that: we annotate only rotations as Charades contains few translation articulations; and we do not annotate surface normals.

Evaluation Criteria: Evaluation of our approach is nontrivial, since our assumed input (RGB videos) precludes measuring outputs quantitatively in 3D. We therefore evaluate our approach on a series of subsets of the problem. We stress from the start though that these metrics are what can be measured (due to the use of RGB inputs), as opposed to the full rich output.

Articulation Recognition: We first independently evaluate the ability to detect whether someone is articulating this object at a point in time. We frame this as a binary prediction problem. This is surprisingly difficult in real scenes because objects are typically partially occluded by humans when humans articulate them, and because humans often touch a articulated objects (e.g., cleaning the surface) without opening it. We use AUROC to measure the performance.

Articulation Description: We next evaluate the ability of a system to detect the articulated object, corresponding articulation type (rotation/translation), axes, and surface normals. We follow other approaches [22, 29, 40, 43, 56] that reconstruct the scene factored into components and treat it as a 3D detection problem, evaluated using average precision (AP). We define error metrics as follows: (*Bounding*

box) IoU, thresholded as 0.5. We find the normal COCO AP, which measures IoU up to 0.95, to be too strict because the precise boundaries of articulating parts are often occluded by people and hard to annotate. (*Axes*) EA-score from the semantic line detection literature [70]. This metric handles a number of edge cases; we use 0.5 as the threshold as recommended by [70]. (*Surface normal*) mean angle error, thresholded at $30°$, following [9, 60]. A prediction is a true positive only if all errors are lower than our thresholds. We calculate the precision-recall curve based on that and report AP for varying combinations of metrics.

Baselines: Prior approaches for articulation detection have focused on robots, synthetic datasets, and real-world RGBD scans. These are different from our setting for two reasons. First, videos of people articulating objects show a noisy background with a person interacting with and occluding the object, as opposed to an isolated articulated object in a simulator. Second, RGB videos do not have depth, which is often a requirement of existing articulation models. For example [30] requires depth, and while they show results on real-world depth scans, their RGBD scans only contain a static object without humans. We propose to compare with the following methods.

3DADN + SAPIEN [64] Data: To test whether we can solve the problem just by training on synthetic data, we create a synthetic data-based method where we train our 3DADN system on synthetic data. We render a synthetic dataset us-

Table 1. We report AUROC for Articulation Recognition, as well as AP for Articulation Description. To separate out difficulties in detecting articulation and characterizing its parameters, we assist Flow+Normal and 3DADN+SAPIEN with ground truth bounding box and denote it as gtbox. 3DADN+SAPIEN cannot detect most objects without the help of gtbox.

Methods	gtbox	Recog. AUROC	Rotation bbox	Rotation bbox+axis	Rotation bbox+axis+normal	Translation bbox	Translation bbox+axis	Translation bbox+axis+normal
Flow [53] + Normal [3]	✗	68.5	7.7	0.3	0.0	0.3	0.0	0.0
Flow [53] + Normal [3]	✔	-	-	3.0	0.3	-	1.4	0.7
D3D-HOI [65] Upper Bound	✗	62.7	28.8	19.7	n/a	4.70	4.7	n/a
3DADN + SAPIEN [64]	✔	-	-	16.8	1.40	-	15.1	0.40
Ours	✗	**76.6**	**61.3**	**30.4**	**17.2**	**34.0**	**27.1**	**17.9**

ing SAPIEN [64] by randomly sampling and driving 3D objects. We filtered 1053 objects of 18 categories with movable planes from PartNet-Mobility Dataset [64], such as doors and laptops. We render frames with the objects articulated, with location parameters picked to give plausible scenes, and extract the information needed to train 3DADN. Without a background, the detection problem becomes trivial, so to mimic real 3D scenes, we blend the renderings with random ScanNet [6] images as the background and render synthetic humans from SURREAL [57]. For fair comparison, we use the same ScanNet+SURREAL images used to train our system's plane parameter head. When evaluated on SAPIEN data, this approach performs well and obtains an AP of (bbox) 60.3, (bbox+rot) 64.1, (bbox+rot+normal) 41.0.

Bottom-up Optical Flow [53] and Surface Normal Change [3] (Flow+Normal): To test whether the data can be solved by the use of fairly simple cues, we construct a baseline that uses Optical Flow [53] (since articulating objects tend to cause movement) and Surface Normals [3] (since rotating planes change their orientation). Both flow and normals provide a $H \times W$ map that can be analyzed. We also use the output of a human segmentation system [18] that was trained on multiple datasets and mask normal and flow magnitude maps wherever it improves performance. Given these maps, we recognize the presence of articulation via logistic regression on a feature vector consisting of the fraction of pixels above multiple thresholds; we recognize bounding boxes via thresholding and finding the tightest enclosing box; we estimate rotation axis as perpendicular to the mean flow change in the bounding box (flow tends to increase away from hinges); we find translation axis using mean flow direction in the box; we find articulation normal using mean predicted normals in the box. Throughout, we use the optimal option of surface normals and flow; this hybrid system performs substantially better than either flow or normals alone.

Baselines with + GT Box: To separate out difficulties in detecting articulation and characterizing its parameters, we also experiment with giving baselines ground-truth bounding box information about the articulating object. This gives

an upper-bound on performance.

D3D-HOI [65] Upper Bound: We compare with D3D-HOI since it accepts RGB video as input and detects how humans articulate objects. A direct comparison with D3D-HOI is challenging since it only works when EFT [23] reconstructs 3D human poses and Pointrend [25] detects the objects that are assumed to articulate and correct CAD models are chosen for the object. However, EFT does not work well on the dataset due to truncated or multiple humans on Internet videos [26, 45]. We therefore report upper-bounds on the performance. We assume it predicts the ground truth bounding box, when EFT mask and pseudo ground truth 2D human segmentation mask [17] has IoU > 0.5 and PointRend [25] produces a mask on articulated objects with confidence > 0.7.

Ours: This is our proposed method. It includes both the per-frame approach described in Section 4.1 and the optimization approach of Section 4.2. We note that this approach also produces outputs that are not being quantitatively measured, such as a 3D plane articulating in 3D. These are qualitatively shown in Figures 1 and 3.

5.2. Results

We first show qualitative results in Figure 3. On challenging Internet videos, our approach usually detects and recovers the 3D articulated plane regardless of categories.

In Figure 4, we compare our approach with four baselines visually. Flow can occasionally locate articulation (third row), but in most cases, flow is not localized to only the object articulating (e.g. camera movement, top row). Training purely on SAPIEN [64] data has difficulty detecting articulated objects in Internet videos, even if we show all detected objects with confidence score > 0.1. It learns some information of articulation axes when we assist it with ground truth bounding boxes. D3D-HOI [65] relies on both EFT [23] to detect humans and PointRend [25] to detect objects. However, EFT has diffculty predicting 3D humans on Internet videos.

Quantitative Results. We evaluate the approach quantitatively on the three tasks in Table 1. Our approach substantially outperforms the alternate methods. While

Pred	GT	Pred	GT	Pred	GT

Figure 5. Qualitative results on Charades dataset. Without fine-tuning on Charades data, our model obtains strong performance on detecting and characterizing 3D articulation.

Table 2. Evaluation on Charades dataset [49]. We only report rotation AP since Charades does not have sufficient translation motion.

Methods	gtbox	Recog. AUROC	Rotation bbox	Rotation bbox+axis
Flow [53] + Normal [3]	✗	53.7	3.1	0.2
Flow [53] + Normal [3]	✔	-	-	4.2
D3D-HOI Upper Bound	✗	55.9	**14.9**	**13.7**
3DADN + SAPIEN [64]	✔	-	-	1.54
Ours	✗	**58.4**	12.0	12.8

statistically-combined bottom-up cues [3,53] do better than chance at predicting the presence of an articulation, they are substantially worse than the proposed approach and fail to obtain sensible bounding boxes. Even when given the ground-truth box, this method fails to obtain good axes. Due to the frequency of truncated humans in Internet videos [26,45], D3D-HOI [65]'s performance upper-bound is substantially lower than our method's performance. The detection system when trained on synthetic data from [64] fails on our system; when given a good bounding box, synthetic training data obtains reasonable, but inferior numbers and poor accuracy in predicting normals.

Ablations – Optimization. Our optimization produces modest gains in recognition accuracy and axis localization in 2D: It improves recognition AUROC from 74.0 to 76.6, rotation AP from 16.6 to 17.2 and translation AP from 14.3 to 17.9. This small gain is understandable because the evaluation is per-frame and the optimization mainly seeks to make the predictions more consistent. If we quantify the consistency in the results before and after optimization, we find that the EAScore [70] between tracked predicted frames increases from 0.69 (before optimization) to 0.96 (after optimization).

5.3. Generalization Results

We next test our trained models *without fine-tuning* on Charades [48]. We show results in Figure 5. Our approach typically generates reasonable estimations. We find that the video quality and resolution of Charades is lower relative to our videos, with many dark or blurry videos.

We also show quantitative evaluations in Table 2. Here, our performance is slightly diminished. However, we sub-

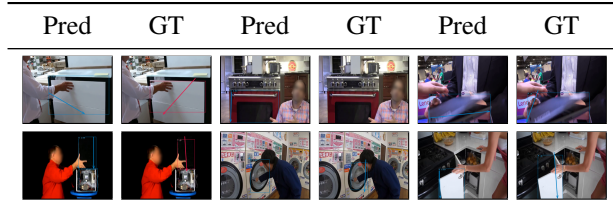| Pred	GT	Pred	GT	Pred	GT

Figure 6. Typical failure modes. (1) Ambiguity of articulation type; (2) Axis outside of the frame or ambiguity of articulation axis location due to symmetry; (3) Object has complex motions (a person moving an object while articulating it; the rotation axis is outside of the articulating surface).

stantially outperform the baselines. We are only marginally outperformed by D3D-HOI upper bound, which assumes perfect performance so long as the data can be obtained.

5.4. Limitations and Failure Modes

We finally discuss our limitations and typical failure modes in Figure 6. We find some examples are particularly challenging: (1) Column 1: some images may contain hard examples where the axis types are hard to figure out. (2) Column 2: the axis is outside of the image frame or its location is ambiguous due to symmetry or occlusion. (3) Column 3: the object has complex dynamics or dual axes; for example, a person moving a laptop while opening it or the cabinet has multiple joints.

6. Conclusion

We have demonstrated our approach's ability to detect and characterize 3D planar articulation of objects from ordinary videos. Future work includes combining 3D shape reconstruction with the articulation detection pipeline.

Our approach can have positive impacts by helping build smart robots that are able to understand and manipulate articulated objects. On the other hand, our approach may be useful for surveillance activities. Moreover, our network is trained on Internet videos and deep networks may amplify biases in the data.

References

[1] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020. 2

[2] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. Hico: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015. 2

[3] Weifeng Chen, Shengyi Qian, David Fan, Noriyuki Kojima, Max Hamilton, and Jia Deng. Oasis: A large-scale dataset for single image 3d in the wild. In *CVPR*, 2020. 2, 3, 7, 8

[4] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 1

[5] Cristina Garcia Cifuentes, Jan Issac, Manuel Wüthrich, Stefan Schaal, and Jeannette Bohg. Probabilistic articulated real-time tracking for robot manipulation. *IEEE Robotics and Automation Letters*, 2(2):577–584, 2016. 2

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 4, 7

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 3

[8] Karthik Desingh, Shiyang Lu, Anthony Opipari, and Odest Chadwicke Jenkins. Factored pose estimation of articulated objects using efficient nonparametric belief propagation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7221–7227. IEEE, 2019. 2

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

[10] David F. Fouhey, Vincent Delaitre, Abhinav Gupta, Alexei A. Efros, Ivan Laptev, and Josef Sivic. People watching: Human actions as a cue for single-view geometry. In *ECCV*, 2012. 2

[11] David F. Fouhey, Weicheng Kuo, Alexei A. Efros, and Jitendra Malik. From lifestyle VLOGs to everyday interactions. In *CVPR*, 2018. 3

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

[13] Georgia Gkioxari, Ross Girshick, Piotr Dollar, and Kaiming He. Detecting and recognizing human-object interactions. In *CVPR*, 2018. 2

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 3

[15] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 3

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3

[17] Vladimir Iglovikov. Binary segmentation of people. https://github.com/ternaus/people_segmentation, 2020. 7

[18] Vladimir Iglovikov and Alexey Shvets. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018. 7

[19] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 2

[20] Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. *arXiv preprint arXiv:2008.10518*, 2020. 1, 2

[21] Ziyu Jiang, Buyu Liu, Samuel Schulter, Zhangyang Wang, and Manmohan Chandraker. Peek-a-boo: Occlusion reasoning in indoor scenes with plane representations. In *CVPR*, 2020. 4

[22] Linyi Jin, Shengyi Qian, Andrew Owens, and David F. Fouhey. Planar surface reconstruction from sparse views. In *ICCV*, 2021. 4, 6

[23] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. *arXiv preprint arXiv:2004.03686*, 2020. 7

[24] Ken-ichi Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, 2001. 2

[25] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 7

[26] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *ICCV*, 2021. 7, 8

[27] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013. 2

[28] Nilesh Kulkarni, Abhinav Gupta, David Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, 2020. 2

[29] Nilesh Kulkarni, Ishan Misra, Shubham Tulsiani, and Abhinav Gupta. 3D-RelNet: Joint object and relational network for 3D prediction. In *ICCV*, 2019. 6

[30] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *CVPR*, 2020. 1, 2, 6

[31] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[32] 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. 3, 4

[33] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. PlaneRCNN: 3D plane detection and reconstruction from a single image. In *CVPR*, 2019. 1, 2, 4

[34] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *CVPR*, 2018. 2, 4

[35] Liu Liu, Han Xue, Wenqiang Xu, Haoyuan Fu, and Cewu Lu. Towards real-world category-level articulation pose estimation. *IEEE Transactions on Image Processing*, 2022. 2

[36] Yizhou Liu, Fusheng Zha, Lining Sun, Jingxuan Li, Mantian Li, and Xin Wang. Learning articulated constraints from a one-shot demonstration for robot manipulation planning. *IEEE Access*, 7:172584–172596, 2019. 1

[37] Frank Michel, Alexander Krull, Eric Brachmann, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *BMVC*, 2015. 2

[38] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *ICCV*, 2021. 1, 2

[39] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. In *ICCV*, 2021. 1, 2

[40] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020. 6

[41] Claudia Pérez-D'Arpino and Julie A Shah. C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *ICRA*, 2017. 1

[42] Sudeep Pillai, Matthew R Walter, and Seth Teller. Learning articulated motions from visual demonstration. In *RSS*, 2014. 1, 2

[43] Shengyi Qian, Linyi Jin, and David F. Fouhey. Associative3d: Volumetric reconstruction from sparse views. In *ECCV*, 2020. 6

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[45] Chris Rockwell and David F Fouhey. Full-body awareness from partial observations. In *ECCV*, 2020. 7, 8

[46] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 3

[47] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 2, 3

[48] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2, 3, 8

[49] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 5, 8

[50] Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29, 2005. 1

[51] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011. 1, 2

[52] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *arXiv preprint arXiv:2106.14405*, 2021. 2

[53] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 7, 8

[54] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2):137–154, 1992. 2

[55] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):878–892, 2008. 2

[56] Shubham Tulsiani, Saurabh Gupta, David F Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2D image of a 3D scene. In *CVPR*, 2018. 6

[57] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 4, 7

[58] John YA Wang and Edward H Adelson. Representing moving images with layers. *IEEE transactions on image processing*, 3(5):625–638, 1994. 2

[59] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions ˜ transformations. In *CVPR*, 2016. 2

[60] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015. 6

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

[62] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *CVPR*, 2019. 2

[63] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4

[64] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *CVPR*, 2020. 2, 6, 7, 8

[65] Xiang Xu, Hanbyul Joo, Greg Mori, and Manolis Savva. D3d-hoi: Dynamic 3d human-object interactions from videos. *arXiv preprint arXiv:2108.08420*, 2021. 1, 2, 7, 8

[66] Fengting Yang and Zihan Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *ECCV*, 2018. 2

[67] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *CVPR*, 2019. 2

[68] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, 2019. 3

[69] Jason Y. Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *ICCV*, 2019. 2

[70] Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. Deep hough transform for semantic line detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 6, 8