Learning the Depths of Moving People by Watching Frozen People

Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, William T. Freeman

Abstract—We present a method for predicting dense depth in scenarios where both a monocular camera and people in the scene are freely moving (right of Figure 1). Existing methods for recovering depth for dynamic, non-rigid objects from monocular video impose strong assumptions on the objects' motion and may only recover sparse depth. In this paper, we take a data-driven approach and learn human depth priors from a new source of data: thousands of Internet videos of people imitating mannequins, i.e., freezing in diverse, natural poses, while a hand-held camera tours the scene (left of Figure 1). Because people are stationary, geometric constraints hold, thus training data can be generated using multi-view stereo reconstruction. At inference time, our method uses motion parallax cues from the static areas of the scenes to guide the depth prediction. We evaluate our method on real-world sequences of complex human actions captured by a moving hand-held camera, show improvement over state-of-the-art monocular depth prediction methods, and demonstrate various 3D effects produced using our predicted depth.

Index Terms—Depth Prediction, Mannequin Challenge, Dynamic Scene Reconstruction

1 Introduction

A hand-held camera capturing video of a dynamic scene is a common scenario. Recovering dense geometry in this case is a challenging task: moving objects violate the epipolar constraint commonly used in 3D vision (Figure 2), and are often treated as noise or outliers in existing structure-from-motion (SfM) and multi-view stereo (MVS) methods. Human depth perception, however, is not easily fooled by object motion—rather, we maintain a feasible interpretation of the objects' geometry and depth ordering even if both the observer and the objects are moving, and even when the scene is observed with just one eye [15]. In this work, we take a step towards achieving this ability computationally.

We focus on the task of predicting accurate, dense depth from ordinary videos where both the camera and *people* in the scene are *naturally moving*. We focus on humans for two reasons: i) in many application areas, such as augmented reality, humans constitute the salient objects in the scene, and ii) human motion is articulated and difficult to model. By taking a data-driven approach, we avoid the need to explicitly impose assumptions on the shape or deformation of people, but instead learn these priors from data.

Where do we get data to train such a method? Generating high-quality synthetic data where both the camera and the people in the scene are naturally moving is very challenging. One approach would be to record real scenes with an RGBD sensor (e.g., a Microsoft Kinect), but such data is typically limited to indoor environments and requires significant manual work to capture and process. In addition, if such a dataset is captured in the lab, a model trained on it may have difficulty generalizing to real scenes. It is also difficult to gather a diverse collection of people with diverse poses at scale.

Instead, we derive data from a surprising source: YouTube videos in which people imitate mannequins, i.e., freeze in elaborate, natural poses, while a hand-held camera tours the scene (Figure 3). These videos comprise our new *MannequinChallenge (MC)* dataset, which we have released for the research community [25]. Because the entire scene in such videos is stationary—including the people—we can accurately estimate camera poses and depth using modern SfM and MVS algorithms, and then use this derived 3D data as supervision for training a model to predict depth for moving scenes.

In particular, we design and train a deep neural network that takes an input RGB image, a mask indicating human regions, and an initial depth defined for the static environment (i.e., the non-human regions), and outputs a dense depth map over the *entire* image—both the environment and the people. Note that the initial environmental depth is computed using motion parallax between two video frames, providing the network with information not available from a single frame. Once trained, our model can handle natural videos with arbitrary camera and human motion.

We demonstrate our method on a variety of real-world Internet videos shot with a hand-held camera and depicting complex human actions such as walking, running, and dancing. Our model predicts depth with higher accuracy than state-of-the-art monocular depth prediction and motion stereo methods. We further show how our predicted depth maps can be used to produce various 3D effects such as synthetic depth-of-field, depth-aware inpainting, and insertion of virtual objects into 3D scenes with correct handling of occlusion.

In summary, our contributions are: i) a new source of data for depth prediction consisting of a large number of Internet videos in which the camera moves around people "frozen" in natural poses, along with a methodology for generating accurate depth maps and camera poses; and ii) a deep-network-based model that makes use of motion parallax cues from video sequences, and that is designed and trained to predict dense depth maps in the challenging case of simultaneous camera motion and complex human motion.

2 Related Work

Learning-based depth prediction. Numerous algorithms, based on both supervised and unsupervised learning methods, have

- *Z. Li and N. Snavely are with the Department of Computer Science, Cornell Tech, Cornell University. T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu and W. Freeman are with Google Research.*
Project website with dataset and code: https://mannequin-depth.github.io

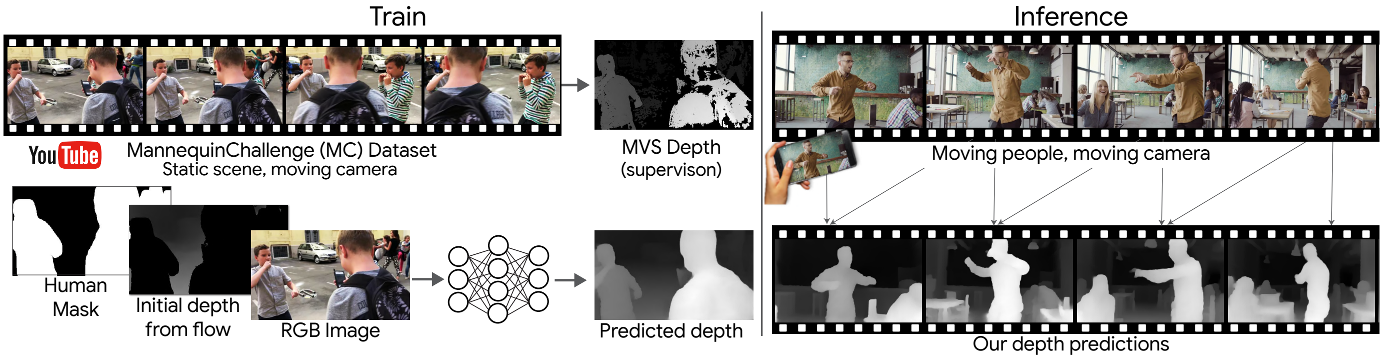


Fig. 1: Our model predicts dense depth when both an ordinary camera and people in the scene are freely moving (right). We train our model on our new *MannequinChallenge* dataset—a collection of Internet videos of people imitating mannequins, i.e., freezing in diverse, natural poses, while a camera tours the scene (left). Because people are *stationary*, geometric constraints hold; this allows us to use multi-view stereo to estimate depth which serves as supervision during training. In all figures, we use inverse depth maps for visualization purposes, and refer to them as depth maps.

recently been proposed for predicting dense depth from a single RGB image [5], [8], [9], [10], [23], [26], [28], [41], [50], [56], [60], [63]. However, because these methods use a single RGB image, they ignore useful motion parallax cues present in video sequences. Some recent learning-based methods also consider multiple images for depth estimation, either assuming known camera poses [16], [58] or simultaneously predicting camera poses along with depth [48], [62]. However, these methods assume that the captured scenes are completely static. They are not designed to estimate depth for dynamic objects, which is the focus of our work.

Depth estimation for dynamic scenes. Depth information captured from RGBD sensors or stereo cameras has been widely used for 3D modeling of dynamic scenes [1], [2], [7], [18], [20], [32], [38], [51], [59], [66]. However, only a few methods attempt to estimate depth from a monocular camera. Several methods have sought to reconstruct *sparse* geometry for dynamic scenes using either a single monocular camera [34], [44], [61], or multiple unsynchronized cameras [49]. Russell *et al.* [39] and Ranftl *et al.* [36] suggest motion/object segmentation–based algorithms to decompose a dynamic scene into piecewise rigid parts before inferring depth ordering. However, these methods impose strong assumptions about object motion that can be violated by articulated human motion. More recently, Rematas *et al.* [37] predict depth for moving soccer players using synthetic training data from FIFA video games. However, their method is limited to soccer players, and cannot handle general people in the wild.

RGBD datasets for learning depth. There are a number of RGBD datasets of indoor scenes, captured using depth sensors [4], [6], [43], [55] or rendered from synthetic data [45]. However, none of these datasets provide depth supervision for moving people in natural environments. In particular, several action recognition methods use depth sensors to capture human actions [29], [33], [42], [65], but most of these use a static camera and provide only a limited number of indoor scenes. REFRESH [27] is a recent semi-synthetic scene flow dataset created by overlaying animated people on NYUv2 images. Here, too, the data is limited to interior scenes and consists of synthetic humans placed in unrealistic configurations with respect to their surroundings. The resulting trained models thus have limited ability to generalize to real scenarios.

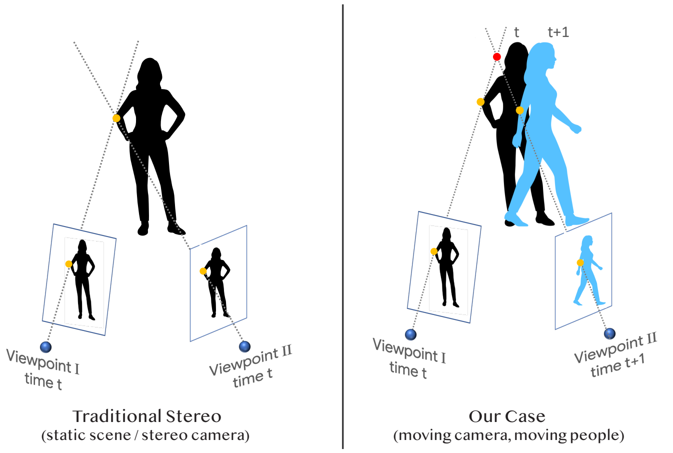


Fig. 2: **Traditional stereo vs. our setup.** Left: a person is observed at the same time instant from two different views. The 3D position of points can be computed using triangulation. Right: when both the camera and the objects in the scene are moving, triangulation is no longer possible since the epipolar constraint does not apply.

Human shape and pose prediction. Recovery of a posed 3D human mesh from a single RGB image has attracted significant attention [3], [11], [21], [24], [30], [35]. Recent methods achieve impressive results on natural images spanning a variety of poses, some of which can also model fine details such as hair and clothing [12], [57], [57]. However, such approaches do not model geometric relations between the people and the static parts of the scenes. Finally, many of these methods rely on correctly detecting human keypoints, requiring most of the body to be visible in each video frame.

3 The MannequinChallenge Dataset

The *Mannequin Challenge* [52] is a popular video trend in which people freeze in place—often in interesting poses—while the camera operator moves around the scene filming them. Thousands of such videos have been created and uploaded to YouTube since late 2016. These videos comprise our new *MannequinChallenge (MC) Dataset* [25], which spans a wide range of scenes with people of different ages, naturally posing in different group configurations

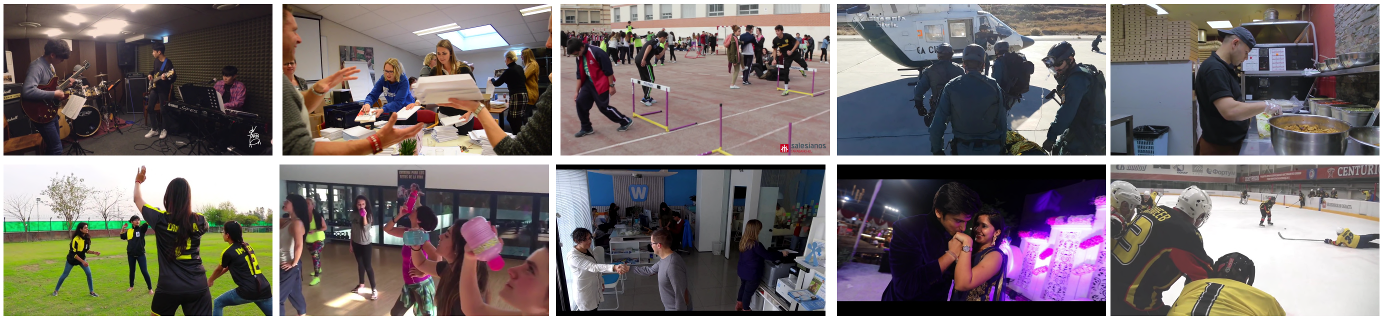


Fig. 3: **Sample images from Mannequin Challenge videos.** Each image is a frame from a video sequence in which the camera is moving but the *humans are all static*. The videos span a variety of natural scenes, poses, and configurations of people.

(see Figure 3). To the extent that people succeed in staying still during the videos, we can assume the scenes are static and obtain accurate camera poses and depth information by processing them with SfM and MVS algorithms. However, recovering accurate geometry from such raw Internet videos is challenging, and requires careful filtering of noisy video clips and individual frames in each clip. After processing, we obtain around 2,000 candidate videos from which we derive 4,690 sequences comprised of a total of more than 170K valid image-depth pairs.

We now describe in detail how we process the raw videos and derive our training data.

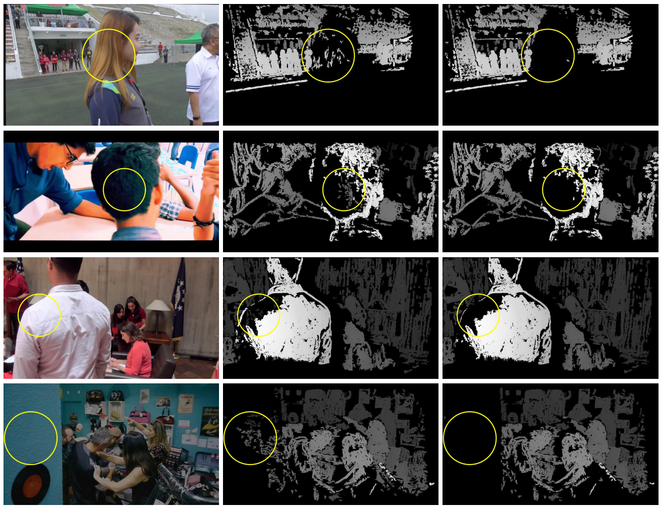
Estimating camera poses. Following a similar approach to Zhou *et al.* [64], we use ORB-SLAM2 [31] to identify trackable sequences in each video and to estimate an initial camera pose for each frame. At this stage, we process a lower-resolution version of the video for efficiency, and set the field of view to 60 degrees (a typical value for modern cell-phone cameras). We then reprocess each sequence at a higher resolution using a visual SfM system [40], which refines the initial camera poses and intrinsic parameters. This method extracts and matches features across frames in the videos, then performs a global bundle adjustment optimization. Finally, sequences with non-smooth camera motion are removed using the technique of Zhou *et al.* [64], as we observe that such sequences often have erroneous camera poses.

Computing dense depth with MVS. Once the camera poses for each clip are estimated, we then reconstruct each scene's dense geometry. In particular, we recover per-frame dense depth maps using COLMAP, a state-of-the-art MVS system [41].

Because our data consists of challenging Internet videos that exhibit camera motion blur, shadows, reflections, etc., the raw depth maps estimated by MVS are often too noisy for use in training a model. We address this issue with a careful depth cleaning procedure. We first filter outlier depths using the depth refinement method proposed by Li and Snavely [26]. We further remove erroneous depth values by considering the consistency between the MVS depth and the depth obtained from motion parallax between pairs of frames. Specifically, for each frame, we compute a normalized error $\Delta(\mathbf{p})$ for every valid pixel $\mathbf{p}$:

$$\Delta(\mathbf{p}) = \frac{|D_{\mathsf{MVS}}(\mathbf{p}) - D_{\mathsf{pp}}(\mathbf{p})|}{D_{\mathsf{MVS}}(\mathbf{p}) + D_{\mathsf{pp}}(\mathbf{p})} \tag{1}$$

where D_{MVS} is the depth map obtained by MVS and D_{pp} is the depth map computed from two-frame motion parallax (see Section 4.1). Depth values for which $\Delta(\mathbf{p}) > \delta$ are removed, where we empirically set $\delta = 0.2$.



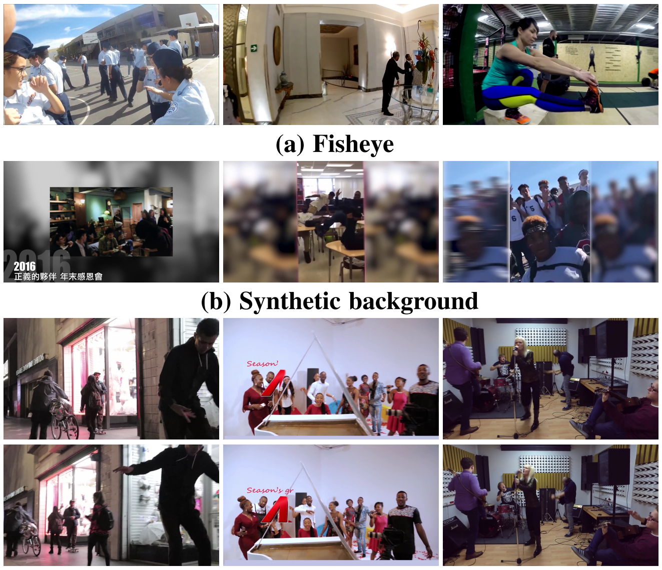
(a) Image (b) Raw D_{MVS} (c) Cleaned D_{MVS}

Fig. 4: **Effect of depth cleaning.** (a-b) Raw MVS depth maps, D_{MVS}, may contain errors and outliers, especially in untextured regions (see regions circled in yellow). (c) Our depth cleaning method effectively filters out such erroneous depth values.

Figure 4 shows examples of MVS depth maps before and after our proposed depth cleaning method. The regions circled in yellow illustrate that our depth cleaning method can effectively remove incorrect depth regions. Because these depth maps serve as supervision during training, this filtering has a significant impact on our model's performance, as shown in our experiments (Sec. 5.2). Figure 7 shows additional examples of our processed sequences with corresponding estimated MVS depths after cleaning.

Filtering clips. Several factors can make a video clip unsuitable for training. For example, people may "unfreeze" (start moving) at some point in the video, or the video may contain synthetic graphical elements in the background. Dynamic objects and synthetic backgrounds do not obey multi-view geometric constraints and hence are treated as outliers and filtered out by MVS, potentially leaving few valid pixels. Therefore, we remove frames where $< 20\%$ of pixels have valid MVS depth after our two-pass cleaning stage.

Further, we remove frames where the estimated radial distortion coefficient $|k_1| > 0.1$ (indicative of a fisheye camera) or where the estimated focal length is ≤ 0.6 or ≥ 1.2 (indicating that the camera parameters are likely inaccurate). We keep sequences that are at least 30 frames long, have an aspect ratio of 16:9, and have a



(a) Fisheye

(b) Synthetic background

(c) Moving objects

Fig. 5: **Sample frames from clips removed during filtering.** (a) Videos captured with fisheye cameras; (b) videos with synthetic backgrounds; (c) sequences with truly moving objects (pairs of frames shown in each column).

width of ≥ 1600 pixels. Finally, we visually inspect the trajectories and point clouds of the remaining sequences and remove obviously incorrect reconstructions.

Figure 5 shows examples of images filtered out from the raw Mannequin Challenge video clips by our data creation pipeline. These examples include images captured by fisheye cameras, as well as images with large regions of synthetic background or moving objects.

After processing, we obtain 4,690 sequences with a total of more then 170K valid image-depth pairs. We split our MC dataset into training, validation and testing sets with a 80:3:17 split over clips.

4 Depth Prediction Model

We train our depth prediction model on our MannequinChallenge dataset in a supervised manner, i.e., by regressing to the depth generated by the SfM and MVS pipeline. A key question is how to structure the input to the network to allow training on frozen people but inference on moving people.

One possible approach is to regress to depth from a single RGB image (RGB-to-depth), but this approach disregards geometric information about the static regions of the scene that is available by considering more than a single frame. To benefit from such information, we design a two-frame model that uses depth estimated from motion parallax for the static, non-human regions of the scene (Figure 6).

The full input to our network (Figure 7) includes 1) a reference image I^r, 2) a binary mask M indicating human regions, 3) an initial depth map D_{pp} estimated from motion parallax and with human regions removed, 4) a confidence map C, and 5) an optional human keypoint map K. We assume known, accurate camera poses from SfM during both training and inference. In an online inference-time setting, accurate camera poses can also be obtained using visual-inertial odometry. Given these inputs, the network predicts a full depth map for the entire scene. To match the MVS depth values, the network must inpaint the depth in human regions, refine the depth in non-human regions from the estimated D_{pp}, and finally make the depth of the entire scene consistent.

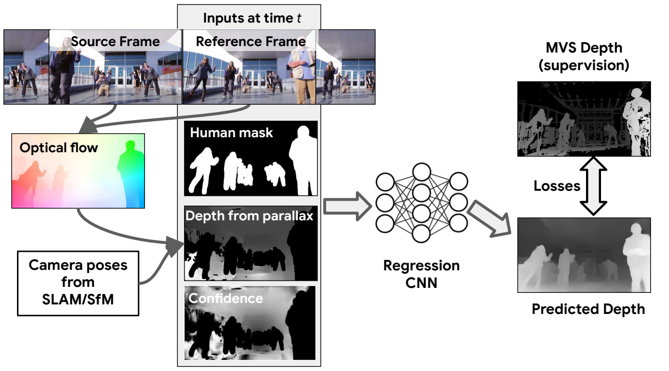


Fig. 6: **System overview**. Our model takes as input the RGB frame, a human segmentation mask, masked depth from motion parallax (via optical flow and SfM pose), and associated confidence map. We ask the network to use these inputs to predict depths that match the ground truth MVS depth.

Our network architecture is a variant of the hourglass network proposed by Chen *et al.* [5]. Specifically, the network has a standard encoder-decoder U-Net structure, with matching input and output resolution, consisting of approximately 5M parameters. In addition, an Inception module variant [47] is used in each convolutional layer of the network. We replace nearest-neighbor upsampling layers with bilinear upsampling layers, which we found to produce sharper depth maps while slightly improving overall accuracy. We refer readers to the supplementary material and to Chen *et al.* [5] for full details of our network architecture. The following sections describe our model inputs and training losses in detail.

4.1 Depth from motion parallax

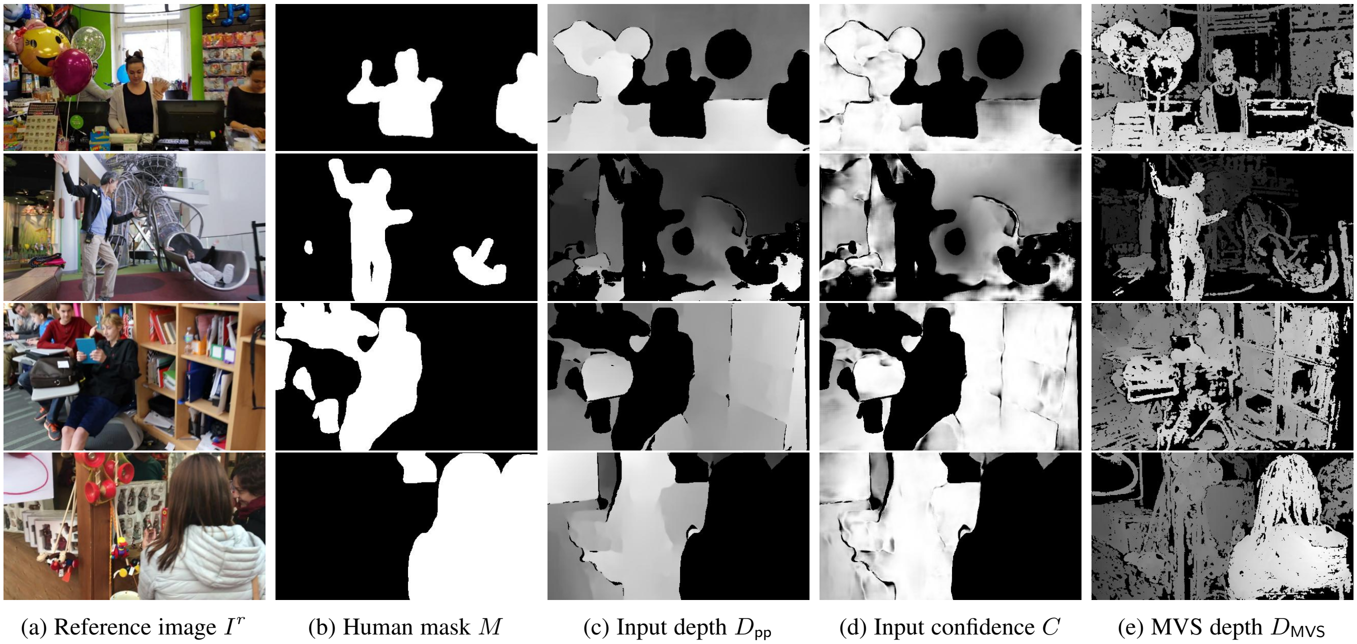
Motion parallax between two video frames provides an initial depth estimate for the static regions of the scene. We assume humans are dynamic while the rest of the scene is static. Specifically, for each reference frame, I^r, we select another frame in the video I^s, and estimate an optical flow field from I^r to I^s using FlowNet2.0 [17]. Given the estimated flow field and the relative camera poses between the two views, we then compute an initial depth map using the Plane-Plus-Parallax (P+P) representation [19], [53].

Note that P+P is typically used to estimate the relative structure of a scene with respect to a reference plane, either a plane in the scene or a virtual reference plane. In our case, we use it as means to cancel out relative camera rotation, as described below.

Formally, suppose we have a relative camera pose relating I^s and I^r consisting of a 3D rotation $\mathbf{R} \in SO(3)$ and 3D translation $\mathbf{t} \in \mathbb{R}^3$, with shared intrinsics matrix $\mathbf{K}$. Given an arbitrary planar surface, Π, the geometric relation between a 2D image point $\mathbf{p} \in I^r$ and its corresponding point $\mathbf{p}' \in I^s$ (expressed in homogeneous coordinates) can be represented as a combination of a *planar* component and *residual parallax* component:

$$\mathbf{p} = \mathbf{p}_w + \boldsymbol{\mu}, \tag{2}$$

where $\mathbf{p}_w$ is the 2D image point in I^r that results from warping $\mathbf{p}' \in I^s$ by a homography $\mathbf{A}$, which aligns the plane Π between the two views, and $\boldsymbol{\mu}$ is the remaining 2D parallax motion. We



(a) Reference image I^r (b) Human mask M (c) Input depth D_{pp} (d) Input confidence C (e) MVS depth D_{MVS}

Fig. 7: **System inputs and training data.** The input to our network consists of: (a) an RGB image, (b) a human mask, (c) a masked depth map computed from motion parallax w.r.t. a selected source image, and (d) a masked confidence map. Low confidence regions (dark circles) in the first two rows indicate the vicinity of the camera epipole, where depth from parallax is unreliable and removed. The network is trained to regress to MVS depth (e).

refer readers to the supplementary material for a detailed definition of $\mathbf{p}_w$ and $\boldsymbol{\mu}$.

One can show that when setting the reference plane Π to the plane at infinity, the expression in Eq. 2 can be written as:

$$\mathbf{p} = \mathbf{p}_w + \frac{t_z(\mathbf{p}_w - \mathbf{Kt})}{D_{\mathsf{pp}}(\mathbf{p})}, \quad (3)$$

where $D_{\mathsf{pp}}(\mathbf{p})$ is the depth value at $\mathbf{p}$ in the coordinate system of the reference view I^r, and t_z is the third component of the translation vector $\mathbf{t}$. In addition, the homography $\mathbf{A}$ in this case is computed as $\mathbf{A} = \mathbf{KRK}^{-1}$.

From Eq. 3, we can estimate the depth $D_{\mathsf{pp}}(\mathbf{p})$ as:

$$D_{\mathsf{pp}}(\mathbf{p}) = \frac{||t_z\mathbf{p}_w - \mathbf{Kt}||_2}{||\mathbf{p} - \mathbf{p}_w||_2}, \quad (4)$$

We found this computation to be more efficient and robust for dense depth estimation compared to standard triangulation methods, which are usually applied to sparse correspondences. See the supplementary material for a detailed derivation of Eq. 4.

In some cases, such as forward/backward relative camera motion, $||\mathbf{p} - \mathbf{p}_w||_2$ will be close to zero in some image regions (i.e., near the camera epipole), resulting in ill-defined depth values. We detect and remove these image regions as described in Sec. 4.2.

Keyframe selection. Depth from motion parallax can be ill-posed if the 2D displacement between two views is small or well-approximated by a homography (e.g., in the case of pure camera rotation). To avoid such cases, we use a heuristic to select a reference frame I^r and a corresponding source keyframe I^s. We want the two views to have significant overlap, while having sufficient baseline (i.e., distance between camera centers). In particular, for each I^r, we find the index s of I^s as

$$s = \arg\max_j d_{rj} o_{rj} \quad (5)$$

where d_{rj} is the L_2 distance between the camera centers of I^r and neighboring frame I^j. The term o_{rj} is the fraction of co-visible SfM features in I^r and I^j:

$$o_{rj} = \frac{2|V^r \bigcap V^j|}{|V^r| + |V^j|}, \quad (6)$$

where V^j is the set of features visible in I^j. We discard pairs of frames for which $o_{rj} < \tau_o$, *i.e.*, the fraction of co-visible features should be larger than a threshold τ_o (we set $\tau_o = 0.6$), and limit the maximum frame interval to 10. We found these view selection criteria to work well in our experiments.

4.2 Depth confidence

Our data consists of challenging Internet video clips with camera motion blur, shadows, low lighting, and reflections. In such cases, optical flow is often noisy [54], leading to uncertainty in the input depth map D_{pp}. We thus estimate, and feed to the network, a confidence map C. This map allows the network to rely more on the input depth in high-confidence regions, and potentially to improve its prediction in low-confidence regions. The confidence value at each pixel $\mathbf{p}$ in the non-human regions is defined as:

$$C(\mathbf{p}) = C_{\mathsf{lr}}(\mathbf{p})C_{\mathsf{ep}}(\mathbf{p})C_{\mathsf{pa}}(\mathbf{p}). \quad (7)$$

where the individual terms are defined as follows.

Flow consistency. The term C_{lr} measures "left-right" consistency between the forward and backward flow fields. Specifically, we denote forward flow from I^r to I^s as $\mathbf{f}_{\mathsf{fwd}}$, and backward flow from I^s to I^r as $\mathbf{f}_{\mathsf{bwd}}$. C_{lr} is then defined as:

$$C_{\mathsf{lr}}(\mathbf{p}) = \max\left(0, 1 - r(\mathbf{p})^2/\bar{\sigma}^2\right) \quad (8)$$

where $r(\mathbf{p}) = ||\mathbf{f}_{\mathsf{fwd}}(\mathbf{p}) + \mathbf{f}_{\mathsf{bwd}}(\mathbf{p}')||_2$ is the forward-backward optical flow warping error, and $\bar{\sigma}$ is a tolerance parameter. For perfectly consistent forward and backward flows $C_{lr}=1$, while $C_{\mathsf{lr}}=0$ when the error is greater than $\bar{\sigma}$ pixels (we set $\bar{\sigma}$ = 1px in our experiments).

Geometric consistency. The term C_{ep} measures how well the flow field complies with the epipolar constraint between the views [13]. C_{ep} gives low confidence to pixels where the flow field and the epipolar constraint disagree:

$$C_{\mathsf{ep}}(\mathbf{p}) = \max\left(0, 1-(\gamma(\mathbf{p})/\bar{\gamma})^2\right) \tag{9}$$

where $\bar{\gamma}$ controls the epipolar distance tolerance (we set $\bar{\gamma}$ = 2px in our experiments), and the geometric epipolar distance $\gamma(\mathbf{p})$ is defined as:

$$\gamma(\mathbf{p}) = \frac{|\mathbf{p}'^T \mathbf{F}\mathbf{p}|}{\sqrt{(\mathbf{F}\mathbf{p})_x^2 + (\mathbf{F}\mathbf{p})_y^2}} \tag{10}$$

where $\mathbf{F} = \mathbf{K}^{-T}[\mathbf{t}]_\times \mathbf{R}\mathbf{K}^{-1}$ is the fundamental matrix relating the two views, and $(\mathbf{F}\mathbf{p})_x$ and $(\mathbf{F}\mathbf{p})_y$ are the first and second elements of $\mathbf{F}\mathbf{p}$, respectively.

Parallax confidence. The term C_{pa} assigns low confidence to pixels for which the parallax between the views is small [41]:

$$C_{\mathsf{pa}}(\mathbf{p}) = 1 - \left(\frac{\min(\bar{\beta}, \beta(\mathbf{p})) - \bar{\beta}}{\bar{\beta}}\right)^2 \tag{11}$$

where

$$\beta(\mathbf{p}) = \cos^{-1}\left(\frac{\mathbf{v}(\mathbf{p})\mathbf{v}(\mathbf{p}')}{\|\mathbf{v}(\mathbf{p})\|_2 \|\mathbf{v}(\mathbf{p}')\|_2}\right) \tag{12}$$

is the angle between the camera rays meeting at pixel $\mathbf{p}$, and $\mathbf{v}(\mathbf{p}) = \mathbf{K}^{-1}\mathbf{p}$ and $\mathbf{v}(\mathbf{p}') = \mathbf{K}^{-1}\mathbf{p}'$ are viewpoint direction vectors at $\mathbf{p}$ in I^r and $\mathbf{p}'$ in I^s respectively. $\bar{\beta}$ is the angle tolerance (we use $\bar{\beta} = 1°$ in our experiments).

Figure 7(d) shows examples of computed confidence maps. Note that human regions as well as regions for which the confidence $C(\mathbf{p}) < 0.25$ are masked out.

4.3 Keypoints

We optionally use human keypoints as an additional input to the network, providing the network with explicit information about the poses of the people featured. In particular, we apply the Mask-RCNN [14] human keypoint detection algorithm to each frames. This algorithm detects, for each person, a set of keypoints on salient points such as joint locations. We encode these detections as an image for use as a network input by simply setting the image pixel value at each keypoint location to the corresponding keypoint index (normalized to lie between 0 and 1), and the rest of the pixels to zero. Figure 8 shows examples of human keypoints predicted by Mask-RCNN. We show that adding keypoints as an input can boost depth prediction performance for people, as shown in Tables 1 and 2.

4.4 Losses

We train our network to regress to depth maps computed by our proposed data pipeline. Because the estimated depth values from SfM and MVS have an arbitrary scale, we use a scale-invariant depth regression loss. That is, our loss is computed on log-space depth values. Our loss function consists primarily of three terms:

$$\mathcal{L}_{\mathsf{si}} = \mathcal{L}_{\mathsf{MSE}} + \alpha_1 \mathcal{L}_{\mathsf{grad}} + \alpha_2(\mathcal{L}_{\mathsf{sm}^1} + \mathcal{L}_{\mathsf{sm}^2}). \tag{13}$$

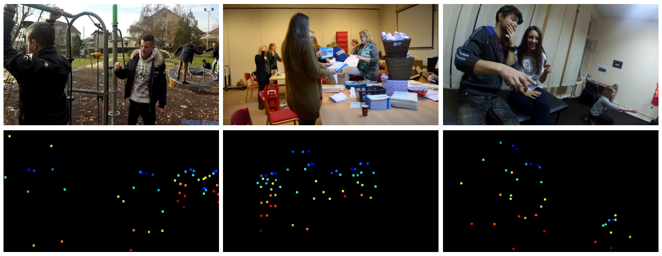


Fig. 8: **Examples of keypoint images**. The top row shows examples of input images and the bottom row shows corresponding detected human keypoint images, where different colors indicating different joints. We perform morphological dilation to the keypoint maps to make each keypoint location more visible.

We compute our losses with respect to the reference image I^r. To simplify notations, we remove the superscript r in the loss equations.

Scale-invariant MSE. $\mathcal{L}_{\mathsf{MSE}}$ denotes the scale-invariant mean square error (MSE) adopted from [8]. This loss term computes the squared, log-space difference in depth between two pixels in the prediction and the same two pixels in the ground truth, averaged over all pairs of valid pixels. That is, it penalizes differences in the depth ratio between any two pixels in the prediction and the ground truth. Further, this loss can be computed in linear time in terms of the number of pixels, as derived in the supplementary material:

$$\mathcal{L}_{\mathsf{MSE}} = \frac{1}{2N^2} \sum_{\mathbf{p}\in I} \sum_{\mathbf{q}\in I} (R(\mathbf{p}) - R(\mathbf{q}))^2 \tag{14}$$

$$= \frac{1}{N} \sum_{\mathbf{p}\in I} R(\mathbf{p})^2 - \frac{1}{N^2} \left(\sum_{\mathbf{p}\in I} R(\mathbf{p})\right)^2 \tag{15}$$

where $R(\mathbf{p}) = \log \hat{D}(\mathbf{p}) - \log D_{\mathsf{gt}}(\mathbf{p})$, and $\hat{D}$ and and D_{gt} denote the predicted and ground truth depth, respectively.

Multi-scale gradient consistency term. To improve depth predictions, we use a multi-scale gradient consistency term to encourage smoother gradient changes and sharper depth discontinuities in the predicted depth images [26]:

$$\mathcal{L}_{\mathsf{grad}} = \sum_{s=0}^{S-1} \frac{1}{N_s} \sum_{\mathbf{p}\in I_s} \left(|\nabla_x R_s(\mathbf{p})| + |\nabla_y R_s(\mathbf{p})|\right) \tag{16}$$

where the subscript s on R_s and I_s indicates that images are computed at scale s, and N_s denotes the number of valid pixel at scale s.

Multi-scale edge-aware smoothness terms. To encourage smooth interpolation of depth in texture-less regions where MVS fails to recover depth, we add smoothness terms at multiple scales based on first- and second-order image derivatives [50], and smoothness weight is modulated by the distance to neighborhood pixels:

$$\mathcal{L}_{\mathsf{sm}^1} = \sum_{s=0}^{S-1} \frac{1}{N_s 2^s} \sum_{\mathbf{p}\in I_s} \exp(-|\nabla I_s(\mathbf{p})|)|\nabla \log \hat{D}_s(\mathbf{p})| \tag{17}$$

$$\mathcal{L}_{\mathsf{sm}^2} = \sum_{s=0}^{S-1} \frac{1}{N_s 2^s} \sum_{\mathbf{p}\in I_s} \exp(-|\nabla^2 I_s(\mathbf{p})|)|\nabla^2 \log \hat{D}_s(\mathbf{p})| \tag{18}$$

For the $\mathcal{L}_{\mathsf{grad}}$, $\mathcal{L}_{\mathsf{sm}^1}$ and $\mathcal{L}_{\mathsf{sm}^2}$ terms, we create $S = 5$-scale image pyramids for both the predicted and ground truth depth images, using nearest-neighbor down-sampling, since we find, compared with bilinear interpolation, nearest-neighbor down-sampling leads to much sharper depth prediction.

5 Results

We test our method quantitatively and qualitatively and compare it with several state-of-the-art single-view and motion-based depth prediction algorithms. We show additional qualitative results on challenging Internet videos with complex human motion and natural camera motion, and demonstrate how our predicted depth maps can be used for several visual effects.

Implementation details. We use FlowNet2.0 [17] to estimate optical flow since it handles large displacements well and preserves sharp motion discontinuities. We use Mask-RCNN [14] to generate human masks and human keypoints. The predicted masks sometimes have errors and miss small parts of people, so we apply a morphological dilation operation to the binary human masks to ensure that the masks are conservative and include all the human regions. When keypoints are used, we normalize their values to between 0 and 1 before feeding them to the network.

Our network predicts log depth at both the training and inference stages. During training, we randomly normalize the input log-depth before feeding it to the network by subtracting a value sampled from between the 40th and 60th percentile of valid input $\log D_{\mathsf{pp}}$. During inference, we normalize input log-depth by subtracting the median of $\log D_{\mathsf{pp}}$. Additionally, during training, we randomly zero out the initial input depth and confidence (with probability 0.1) to address the potential situation where input depth is unavailable (e.g., camera is nearly static or estimated optical flow is completely incorrect) during inference. When using human keypoints as input, we also use the depth from motion parallax D_{pp} with high confidence ($C_{lr} > 0, C_{ep} > 0$ and $C_{pa} > 0.5$) at these locations as ground truth if MVS depth D_{MVS} is not available.

In our experiments, we set hyperparameters in our loss terms $\alpha_1 = 0.5, \alpha_2 = 0.05$ based on the validation set. We train our networks for 20 epochs from scratch using the Adam [22] optimizer with initial learning rate of 0.0004. We halve the learning rate every 8 epochs. During training, we downsample all the images to a resolution of 532×299, use a mini-batch size of 16, and perform data augmentation though random flips and central crops so that input image resolution to the networks is 512×288.

Error metrics. We measure error using scale-invariant RMSE (si-RMSE), equivalent to $\sqrt{\mathcal{L}_{\mathsf{MSE}}}$, described in Section 4.4. We evaluate si-RMSE on five different regions: 1) **si-full** measures the error between all pairs of pixels, giving the overall accuracy across the entire image; 2) **si-env** measures pairs of pixels in non-human regions $\mathcal{E}$, providing depth accuracy of the environment; and 3) **si-hum** measures pairs where at least one pixel lies in the human region $\mathcal{H}$, providing depth accuracy for people. **si-hum** can further be divided into two error measures: 4) **si-intra** measures si-RMSE within $\mathcal{H}$, or human accuracy independent of the environment; and 5) **si-inter** measures si-RMSE between pixels in $\mathcal{H}$ and in $\mathcal{E}$, or human accuracy w.r.t. the environment. We include derivations in the supplementary material.

5.1 Evaluation on the MC test set

We evaluated our method on our MC test set, which consists of more than 29K images taken from 756 video clips. Processed MVS depth values D_{MVS} obtained by our pipeline (see Section 3) are considered as ground truth.

	Network inputs	**si-full**	**si-env**	**si-hum**	**si-intra**	**si-inter**
I.	I	0.333	0.338	0.317	0.264	0.384
II.	$IFCM$	0.330	0.349	0.312	0.260	0.381
III.	$ID_{\mathsf{pp}}M$	0.255	0.229	0.264	0.243	0.285
IV.	$ID_{\mathsf{pp}}CM$	0.232	**0.188**	0.237	0.221	0.268
V.	$ID_{\mathsf{pp}}CMK$	**0.227**	**0.189**	**0.230**	**0.212**	**0.263**
Unmasked D_{pp} (oracle)		0.202	0.206	0.200	0.192	0.213

TABLE 1: **Quantitative comparisons on the MC test set.** Different input configurations of our model: (I) single image; (II) optical flow masked in the human region (F), confidence and human mask; (III) masked input depth, human mask; and (IV) additional confidence; in (V), we also input human keypoints. The last row indicates the error for the depth estimated from motion parallax between two frames in all image regions (human and non-human); this serves as an oracle and can only be measured if the entire scene is static. Lower is better for all metrics.

To quantify the importance of each component of the model's input, we compare the performance of several models, each trained on our MC dataset with a different input configuration. The two main configurations are: (i) a single-view model (input is RGB image) and (ii) our full two-frame model, where the input includes a reference image, an initial masked depth map D_{pp}, a confidence map C, and a human mask M. We also perform ablation studies by replacing the input depth with optical flow F, removing C from the input, and adding the human keypoint map K.

Quantitative evaluations are shown in Table 1. By comparing rows (I), (III) and (IV), it is clear that adding the initial depth of the environment as well as the confidence map significantly improves the performance for both human and non-human regions. Adding human keypoint locations to the network input further improves performance.

Note that if we input an optical flow field to the network instead of depth (II), the performance is only on par with the single-view method. The mapping from 2D optical flow to depth depends on the relative camera poses, which are not provided to the network. This result indicates that the network is unable to implicitly learn relative poses and extract depth information.

Finally, we report the errors for full (unmasked) depth maps computed from motion parallax between two frames (last row of Table. 1). Note that these depth maps can be only computed if the entire scene, including people, is static (thus, this baseline serves as an oracle and cannot be used at test time). As can be seen from the second column (si-env), our model leads to 20% improvement compared to this baseline for non-human regions, which suggests that our model refines the initial input depth (D_{pp}), rather than just copying it. In human regions, where our model has no input depth information, our performance is only 15% below that of depth from motion parallax (si-hum).

Figure 9 shows qualitative comparisons between our single-view model (I) and our full model ($ID_{\mathsf{pp}}CMK$). Our full model results are more accurate in both human regions (first column) and non-human regions (second column). In addition, the depth relationships between people and their surroundings are improved in all examples.

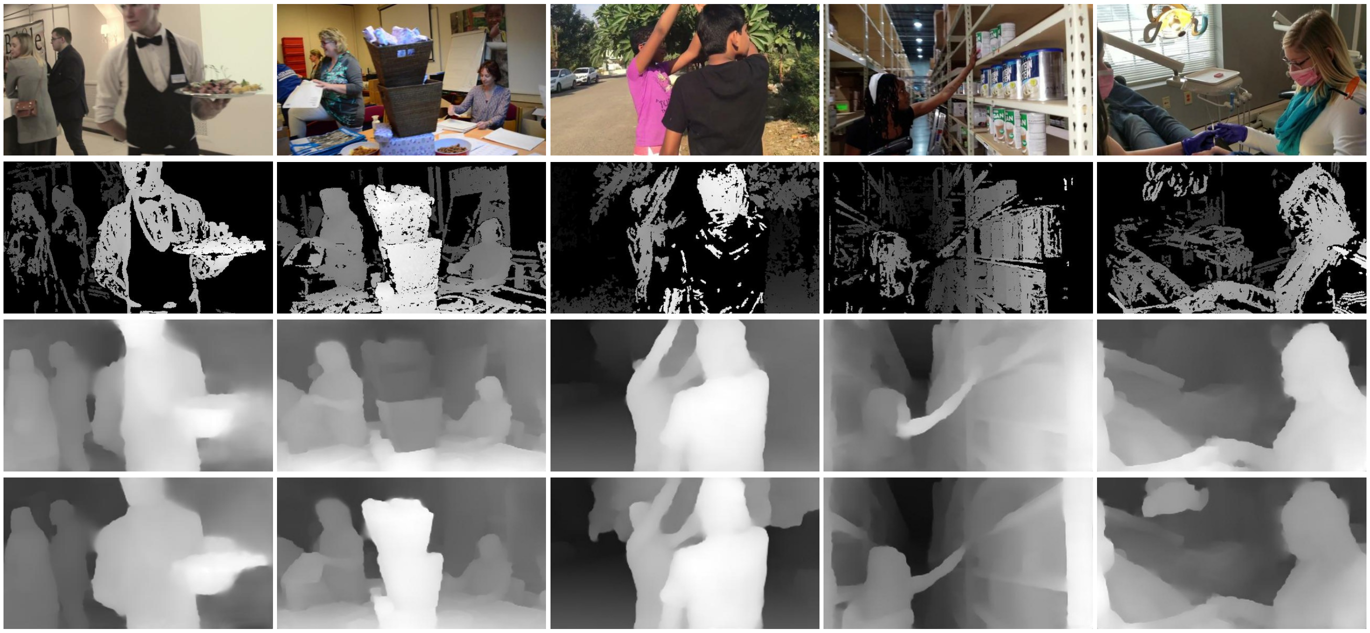


Fig. 9: **Qualitative results on the MC test set.** From top to bottom: reference images and their corresponding MVS depth (pseudo ground truth); our depth predictions using: our single view model (third row) and our two-frame model (forth row). The additional network inputs give improved performance in both human and non-human regions.

Methods	Dataset	two-view?	**si-full**	**si-env**	**si-hum**	**si-intra**	**si-inter**	**RMSE**	**Rel**
Russell *et al.* [39]	-	Yes	2.146	2.021	2.207	2.206	2.093	2.520	0.772
DeMoN [48]	RGBD+MVS	Yes	0.338	0.302	0.360	0.293	0.384	0.866	0.220
Chen *et al.* [5]	NYU+DIW	No	0.441	0.398	0.458	0.408	0.470	1.004	0.262
Laina *et al.* [23]	NYU	No	0.358	0.356	0.349	0.270	0.377	0.947	0.223
Xu *et al.* [56]	NYU	No	0.427	0.419	0.411	0.302	0.451	1.085	0.274
Fu *et al.* [9]	NYU	No	0.351	0.357	0.334	0.257	0.360	0.925	0.194
I	MC	No	0.318	0.334	0.294	0.227	0.319	0.840	0.204
$IFCM$	MC	Yes	0.316	0.330	0.302	0.228	0.323	0.843	0.206
$ID_{\text{pp}}M$	MC	Yes	0.246	0.225	0.260	0.233	0.273	0.635	0.136
$ID_{\text{pp}}CM$ (raw depth)	MC	Yes	0.272	0.238	0.293	0.258	0.282	0.688	0.147
$ID_{\text{pp}}CM$	MC	Yes	0.232	0.203	0.252	0.224	0.262	0.570	0.129
$ID_{\text{pp}}CMK$	MC	Yes	**0.221**	**0.195**	**0.238**	**0.215**	**0.247**	**0.541**	**0.125**

TABLE 2: **Results on the TUM RGBD dataset.** Different si-RMSE metrics as well as standard RMSE and relative error (Rel) are reported. We evaluate our models (light gray background) under different input configurations, as described in Table 1. *Raw depth* indicates the model is trained using raw MVS depth predictions as supervision, without our depth cleaning method. A dataset denoted as '-' indicates that the method is not learning-based. Lower is better for all error metrics.

5.2 Evaluation on the TUM RGBD dataset

We also evaluate on a subset of the TUM RGBD dataset [46], which contains indoor scenes featuring people performing complex actions, captured from different camera poses. Sample images from this dataset are shown in Figure 10(a-b).

To run our model, we first estimate camera poses using ORB-SLAM2, because we found that estimates from ORB-SLAM2 were better synchronized with the RGB images compared to the ground truth poses provided with the TUM dataset. In some cases, due to low image quality and motion blur, the estimated camera poses can be incorrect. We manually filter such failures by inspecting the camera trajectory and point cloud. In total, we obtain 11 valid image sequences with 1,815 images in total for evaluation. We downsample these images to 512×384 resolution in order to preserve their original aspect ratio (our model is fully convolutional and thus can be applied to different image resolutions at test time).

We compare our depth predictions (using our MC trained models) with several state-of-the-art monocular depth prediction methods trained on the indoor NYUv2 [9], [23], [56] and Depth in the Wild (DIW) datasets [5], as well as with a recent two-frame stereo model DeMoN [48], which assumes a static scene. We also compare with Video-Popup [39], which deals with dynamic scenes. We use the same image pairs that were used for computing D_{pp} as inputs to DeMoN and Video-Popup.

Quantitative comparisons are shown in Table 2, where we report five different scale-invariant error measures as well as the standard RMSE metric and relative error; these last two are computed by applying a single scaling factor that best aligns the predicted and ground-truth depths in the least-squares sense. Our single-view model already outperforms the other single-view models, demonstrating the benefit of the MC dataset for training. Note that VideoPopup [39] failed to produce meaningful results

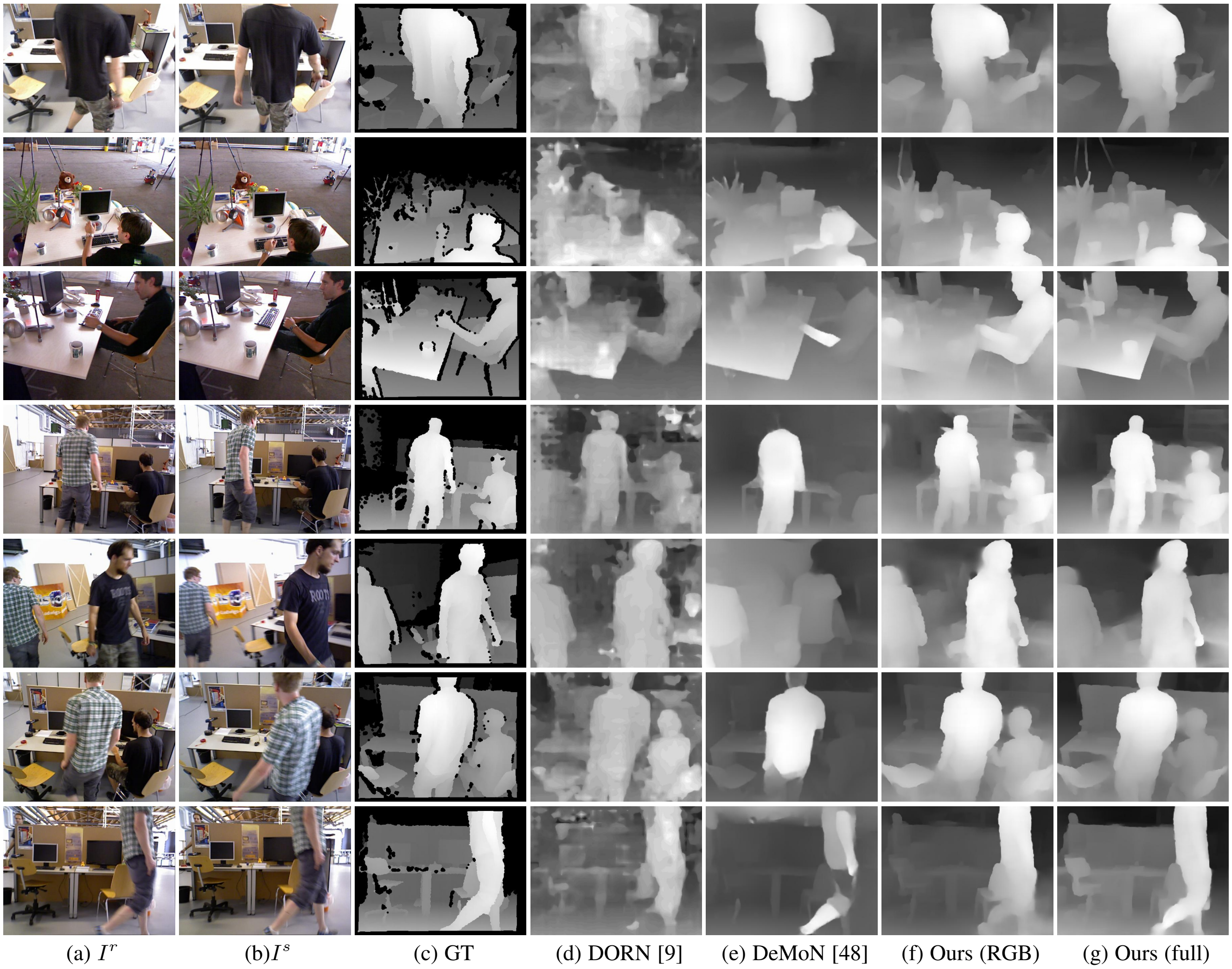


Fig. 10: **Qualitative comparisons on the TUM RGBD dataset.** (a) Reference images, (b) source images (used to compute our initial depth input), (c) ground truth sensor depth, (d) results of the single-view depth prediction method DORN [9], (e) result of the two-frame motion stereo method DeMoN [48], (f-g) depth predictions from our single view and two-frame models, respectively.

due to the challenging camera and object motion present in the data. Our full model, by making use of the initial (masked) depth map, significantly improves performance for all error measures. Consistent with our MC test set results, when we use optical flow as input (instead of the initial depth map) the performance is only slightly better than the single-view network. Finally, we show the importance of our proposed depth cleaning methods that we apply to the training data (see Eq. 1). The same model trained using the raw MVS depth estimates as supervision ("raw depth") leads to a drop of about 15% in performance.

Figure 10 shows a qualitative comparison between these different methods. Our models' depth predictions (Figure 10(f-g)) strongly resemble the ground truth and show a high level of detail, as well as sharp depth discontinuities. This result is a notable improvement over competing methods, which often produce significant errors in both the human regions (e.g., legs in the second row of Figure 10), and the non-human regions (e.g., table and ceiling in the last two rows).

5.3 Internet videos of dynamic scenes

We tested our method on challenging Internet videos (downloaded from YouTube and Shutterstock) that involve simultaneous natural camera motion and human motion. Our SLAM/SfM pipeline was used to generate sequences ranging from 5 to 15 seconds with smooth and accurate camera trajectories, after which we apply our method to obtain the required network input buffers.

We qualitatively compare our full model ($ID_{\mathsf{pp}}CMK$) with several recent learning based depth prediction models: DORN [9], Chen *et al.* [5], and DeMoN [48]. For fair comparisons, we use DORN with a model trained on NYUv2 for indoor videos and a model trained on KITTI for outdoor videos; For Chen *et al.* [5], we use the models trained on both NYUv2 and DIW. For all of our predictions, we use a single model trained from scratch on our MC dataset.

As illustrated in Figure 11, our depth predictions are significantly better than the baseline methods. In particular, DORN [9] has very limited generalization to Internet videos, and Chen *et al.* [5], which is mainly trained on Internet photos, is not able to capture accurate depth. DeMoN often produces incorrect depth, especially in human regions, as it designed for static scenes. Our

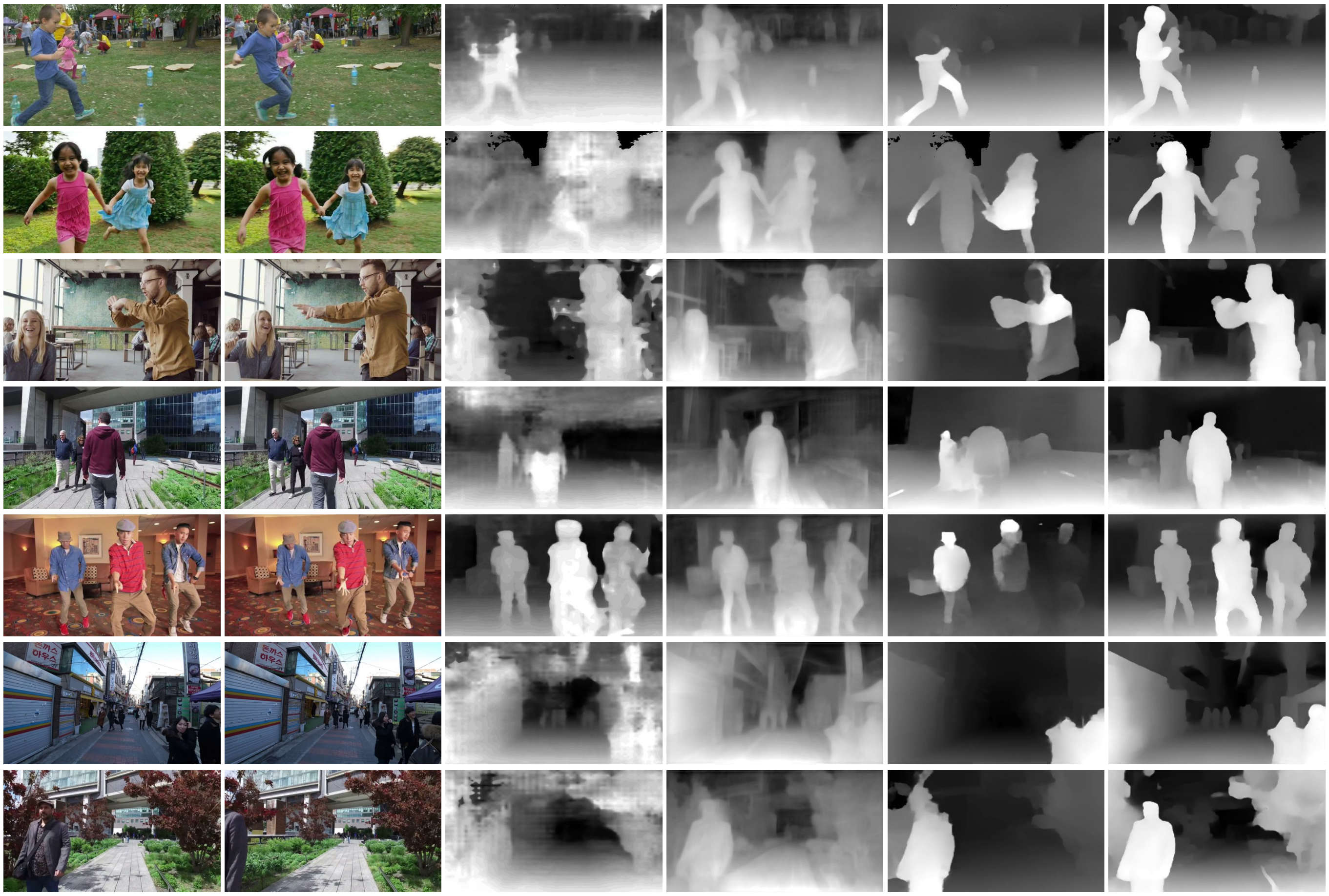


Fig. 11: **Comparisons on Internet video clips with moving cameras and people.** From left to right: (a) reference image, (b) source image, (c) results of DORN [9], (d) results of Chen *et al.* [5], (e) results of DeMoN [48], (f) results of our full method.

predicted depth maps capture accurate depth ordering both between people and other objects in the scene (e.g., between the people and buildings in the fourth row of Figure 11), and within human regions (such as the arms and legs of the people in the first three rows of Figure 11).

Depth-based visual effects. Our depth predictions can be used to apply a range of depth-based visual effects to video. Figure 12 shows depth-based defocus, insertion of synthetic 3D graphics, as well as stereo pairs displayed as anaglyph images. In Figure 13, we show an example of image inpainting by removing nearby humans using our predicted depths.

The depth estimates are sufficiently stable over time to allow inpainting from frames elsewhere in the video. To use a frame for inpainting, we construct a triangle heightfield from the depth map, texture the heightfield with the video frame, and render the heightfield from the target frame using the relative camera transformation. Figure 12 (d, f) shows the results of inpainting two street scenes. Humans near the camera are removed using the human mask M, and holes are filled with colors from up to 200 frames later in the video. Some artifacts are visible in areas that the human mask misses, such as shadows on the ground.

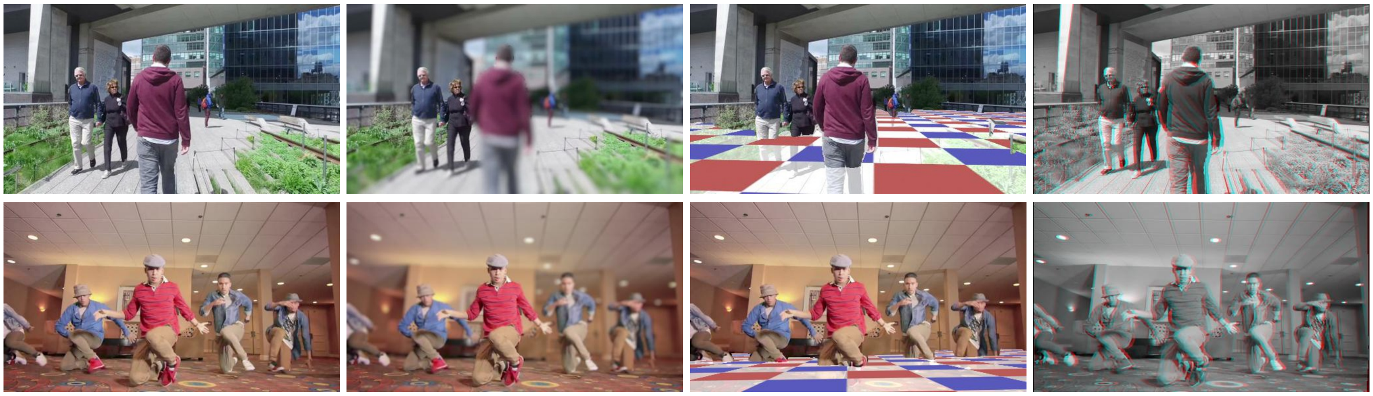
6 Discussion and Conclusion

We demonstrated the power of a learning-based approach for predicting dense depth for dynamic scenes where a monocular camera and people are freely moving. We make a new source of data available for training: a large corpus of Mannequin Challenge videos from YouTube, in which the camera moves around and people are “frozen” in natural poses. We showed how to obtain reliable depth supervision from such noisy data, and demonstrated that by using motion parallax cues available in a video sequence, our models can significantly improve over prior state-of-the-art methods.

Our approach has a number of limitations. First, we assume known and accurate camera poses, which can be difficult to compute accurately if moving objects cover most of the scene or camera motion is close to a pure rotation. Second, our model can fail to generalize to non-standard human poses, as shown in the first three rows of Fig. 14. Third, the depths predicted by our model may be inaccurate for non-human moving regions such as animals, cars, and shadows, as shown in the last three rows of Fig. 14. Finally, our approach also uses just two views, rather than operating on an entire video sequence. This can lead to temporally inconsistent depth estimates and reconstructions across a video. Despite these limitations, we hope that our work can guide and enable further progress in dense reconstruction of dynamic scenes.

References

[1] T. Basha, S. Avidan, A. Hornung, and W. Matusik. Structure and motion from scene registration. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2012.



(a) Input image (b) Defocus (c) Object insertion (d) Anaglyph

Fig. 12: **Depth-based visual effects.** Using our predicted depth maps, we can apply depth-aware visual effects on (a) input images; we show (b) defocus, (c) object insertion, and (d) Anaglyph effects.

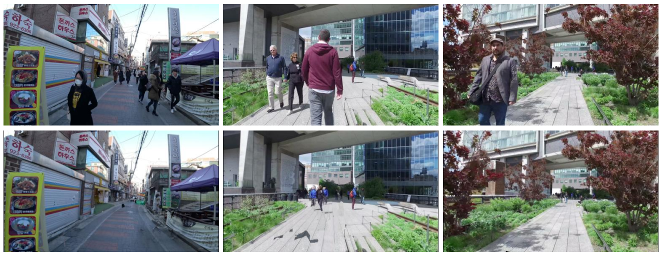


Fig. 13: **Depth-based image inpainting.** We use depth prediction and camera poses to warp the pixels in nearby frames for image inpainting and people removal. Top row shows original images and bottom row shows inpainted images.

[2] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. *Int. J. of Computer Vision*, 2013.

[3] F. Bogo, A. Kanazawa, C. Lassner, P. V. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 561–578, 2016.

[4] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *Int. Conf. on 3D Vision (3DV)*, pages 667–676, 2017.

[5] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Neural Information Processing Systems (NeurIPS)*, pages 730–738, 2016.

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Niessner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017.

[7] M. Dou, S. Khamis, Y. Degtyarev, P. L. Davidson, S. R. Fanello, A. Kowdle, S. Orts, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4D: real-time performance capture of challenging scenes. *ACM Trans. Graphics*, 35:114:1–114:13, 2016.

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Neural Information Processing Systems (NeurIPS)*, pages 2366–2374, 2014.

[9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] R. A. Güler, N. Neverova, and I. Kokkinos. DensePose: Dense Human Pose Estimation In The Wild. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] M. Habermann, W. Xu, M. Zollhoefer, G. Pons-Moll, and C. Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions on Graphics (TOG)*, 38(2):14, 2019.

[13] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[15] I. P. Howard. *Seeing in depth, Vol. 1: Basic mechanisms.* University of Toronto Press, 2002.

[16] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang. DeepMVS: Learning multi-view stereopsis. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] M. Innmann, M. Zollhöfer, M. Niessner, C. Theobalt, and M. Stamminger. VolumeDeform: Real-time volumetric non-rigid reconstruction. In *Proc. European Conf. on Computer Vision (ECCV)*, 2016.

[19] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3d scene analysis. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 17–30, 1996.

[20] H. Jiang, H. Liu, P. Tan, G. Zhang, and H. Bao. 3d reconstruction of dynamic scenes with multiple handheld cameras. In *Proc. European Conf. on Computer Vision (ECCV)*, 2012.

[21] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Int. Conf. on 3D Vision (3DV)*, pages 239–248, 2016.

[24] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] Z. Li, T. Dekel, F. Cole, R. Tucker, and N. Snavely. MannequinChallenge Dataset. https://google.github.io/mannequinchallenge/, 2019.

[26] Z. Li and N. Snavely. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] Z. Lv, K. Kim, A. Troccoli, D. Sun, J. M. Rehg, and J. Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. *Proc. European Conf. on Computer Vision (ECCV)*, 2018.

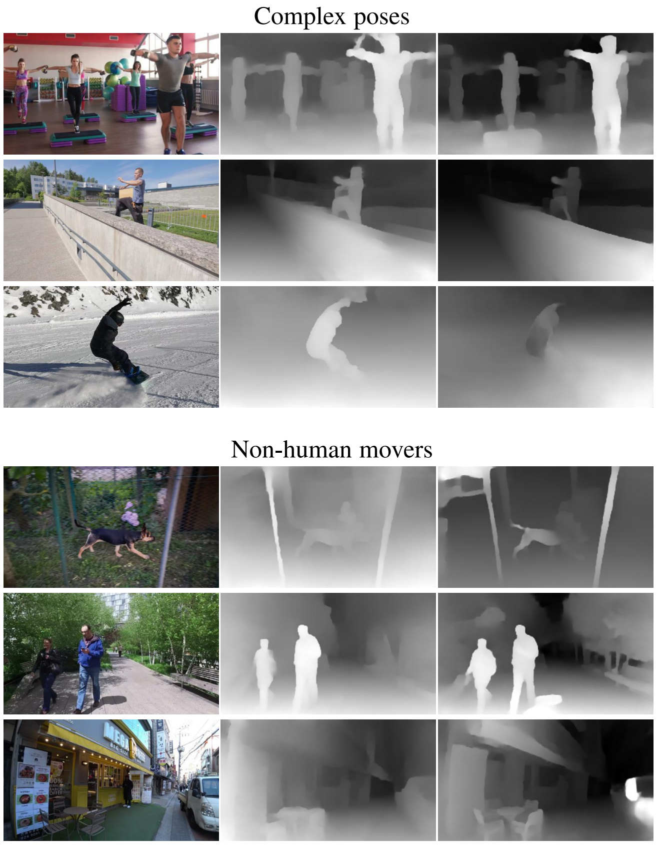
[28] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] O. Mees, A. Eitel, and W. Burgard. Choosing Smartly: Adaptive Multimodal Fusion for Object Detection in Changing Environments. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[30] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Trans. Graphics*, 36:44:1–44:14, 2017.

[31] R. Mur-Artal and J. D. Tardós. Orb-Slam2: An open-source slam system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[32] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruc-



Complex poses

Non-human movers

(a) Input image (b) Ours (RGB) (c) Ours (full)

Fig. 14: **Failure cases.** From left to right: (a) input RGB image (b) depth predicted from our single-view method (c) depth predicted from our proposed full method. Our proposed full method can fail for reasons including (1) a failure to generalize to complex human poses, (first three rows), or due to non-human movers such as animals, cars, and shadows (last three rows). In some of these cases, our single-view method can outperform our full two-view method, because added complexities can sometimes arise in the presence of multiple views.

tion and tracking of non-rigid scenes in real-time. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] B. Ni, G. Wang, and P. Moulin. RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *Proc. ICCV Workshops*, 2011.

[34] H. S. Park, T. Shiratori, I. A. Matthews, and Y. Sheikh. 3D Reconstruction of a Moving Point from a Series of 2D Projections. In *Proc. European Conf. on Computer Vision (ECCV)*, 2010.

[35] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 7025–7034, 2017.

[36] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] K. Rematas, I. Kemelmacher-Shlizerman, B. Curless, and S. Seitz. Soccer on your tabletop. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[38] C. Richardt, H. Kim, L. Valgaerts, and C. Theobalt. Dense wide-baseline scene flow from two handheld video cameras. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 276–285. IEEE, 2016.

[39] C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3d reconstruction of dynamic scenes. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 583–598, 2014.

[40] J. L. Sch"onberger and J.-M. Frahm. Structure-from-motion revisited. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[41] J. L. Sch"onberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 501–518, 2016.

[42] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from Simulated and Unsupervised Images through Adversarial Training. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. European Conf. on Computer Vision (ECCV)*, 2012.

[44] T. Simon, J. Valmadre, I. A. Matthews, and Y. Sheikh. Kronecker-Markov Prior for Dynamic 3D Reconstruction. *Trans. Pattern Analysis and Machine Intelligence*, 39:2201–2214, 2017.

[45] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[46] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[48] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. DeMoN: Depth and motion network for learning monocular stereo. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[49] M. Vo, S. G. Narasimhan, and Y. Sheikh. Spatiotemporal Bundle Adjustment for Dynamic 3D Reconstruction. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[50] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[51] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3d motion understanding. *Int. J. of Computer Vision*, 95(1):29–51, 2011.

[52] Wikipedia. Mannequin Challenge. https://en.wikipedia.org/wiki/Mannequin_Challenge, 2018.

[53] J. Wulff, L. Sevilla-Lara, and M. J. Black. Optical flow in mostly rigid scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[54] K. Xian, C. Shen, Z. Cao, H. Lu, Y. Xiao, R. Li, and Z. Luo. Monocular relative depth perception with web stereo data supervision. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[55] J. Xiao, A. Owens, and A. Torralba. Sun3D: A database of big spaces reconstructed using sfm and object labels. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 1625–1632, 2013.

[56] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Monocular depth estimation using multi-scale continuous crfs as sequential deep networks. *Trans. Pattern Analysis and Machine Intelligence*, 2018.

[57] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt. MonoPerfCap: Human performance capture from monocular video. *ACM Transactions on Graphics (ToG)*, 37(2):27, 2018.

[58] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. MVSNet: Depth Inference for Unstructured Multi-view Stereo. *Proc. European Conf. on Computer Vision (ECCV)*, pages 767–783, 2018.

[59] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2014.

[60] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[61] E. Zheng, D. Ji, E. Dunn, and J.-M. Frahm. Sparse Dynamic 3D Reconstruction from Unsynchronized Videos. *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 4435–4443, 2015.

[62] H. Zhou, B. Ummenhofer, and T. Brox. DeepTAM: Deep Tracking and Mapping. In *Proc. European Conf. on Computer Vision (ECCV)*, 2018.

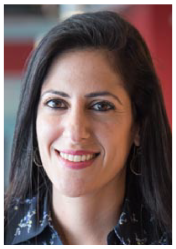
[63] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[64] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo Magnification: Learning view synthesis using multiplane images. *ACM Trans. Graphics (SIGGRAPH)*, 2018.

[65] Y. Zhu, W. Chen, and G. Guo. Evaluating spatiotemporal interest point features for depth-based action recognition. *Image and Vision Computing*, 32(8):453–464, 2014.

[66] M. Zollhöfer, M. Niessner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, et al. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Trans. Graphics*, 33(4):156, 2014.

Zhengqi Li is a CS Ph.D. Candidate at Cornell Tech, Cornell University. Prior to that he received Bachelor of Computer Engineering with High Distinction at University of Minnesota, Twin Cities. His research interests include 3D computer vision, computational photography and inverse graphics. He is a recipient of the CVPR Best Paper Hornorable Mention Award in 2019 and Adobe Research Fellowship Award in 2020.



Tali Dekel is a Senior Research Scientist at Google, Cambridge, developing algorithms at the intersection of computer vision and computer graphics. Before Google, she was a Postdoctoral Associate at the Computer Science and Artificial Intelligence Lab (CSAIL) at MIT, working with Prof. William T. Freeman. Tali completed her Ph.D studies at the school of electrical engineering, Tel-Aviv University, Israel, under the supervision of Prof. Shai Avidan, and Prof. Yael Moses. Her research interests include computational photography, image synthesize, geometry and 3D reconstruction. She is a recipient of the Rothschild Postdoctoral Fellowship, and The National Postdoctoral Award for Advancing Women in Science.

Forrester Cole is a software engineer at Google Research, where he works on combining machine learning and computer graphics techniques. He received an A.B. from Harvard in 2002, and a Ph.D. from Princeton in 2009, both in computer science. He has previously worked at MIT and Pixar Animation Studios, and contributed to major film and game productions.

Richard Tucker is a Software Engineer at Google Research in New York. His research interests include machine learning for 3D perception and view synthesis. He received a PhD in computer science from the University of Cambridge.

Noah Snavely received the B.Sc. degree in computer science from the University of Arizona in 2003, and the Ph.D. degree in computer science and engineering from the University of Washington in 2008. He is a researcher at Google Research, and an associate professor of computer science at Cornell University. He works in computer graphics and computer vision, with a particular interest in using vast amounts of imagery from the Internet to reconstruct and visualize our world in 3D. He is the recipient of a Microsoft New Faculty Fellowship, a PECASE, and a SIGGRAPH Significant New Researcher Award. He is a member of the IEEE.

Ce Liu is a staff research scientist at Google Research, conducting research in the area of computer vision, computer graphics and machine learning. He received his B.E. and M.E. from Tsinghua University in 1999 and 2002, respectively, and received a Ph.D. from MIT Department of Electrical Engineering and Computer Science in 2019. He worked at Microsoft Research Asia from 2002 to 2003, and Microsoft Research New England from 2009 to 2014. He received the best student paper award at NIPS 2006 and CVPR 2009, and the best paper award honorable mention at CVPR 2019. He is a recipient of TPAMI Young Research Award in 2016. He has been serving as area chairs for CVPR/ICCV/ECCV/NeurIPS, and will serve as a co-Program Chair for CVPR 2020.

Willam T. Freeman is a staff research scientist at Google, and the Thomas and Gerd Perkins Professor of Electrical Engineering and Computer Science at MIT, a member of the Computer Science and Artificial Intelligence Laboratory (CSAIL). He was the Associate Department Head from 2011 - 2014.

His current research interests include machine learning applied to computer vision, Bayesian models of visual perception, and computational photography. He received outstanding paper awards at computer vision or machine learning conferences in 1997, 2006, 2009 and 2012, and test-of-time awards for papers from 1990, 1995 and 2005. Previous research topics include steerable filters and pyramids, orientation histograms, the generic viewpoint assumption, color constancy, computer vision for computer games, and belief propagation in networks with loops.