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
Google Research

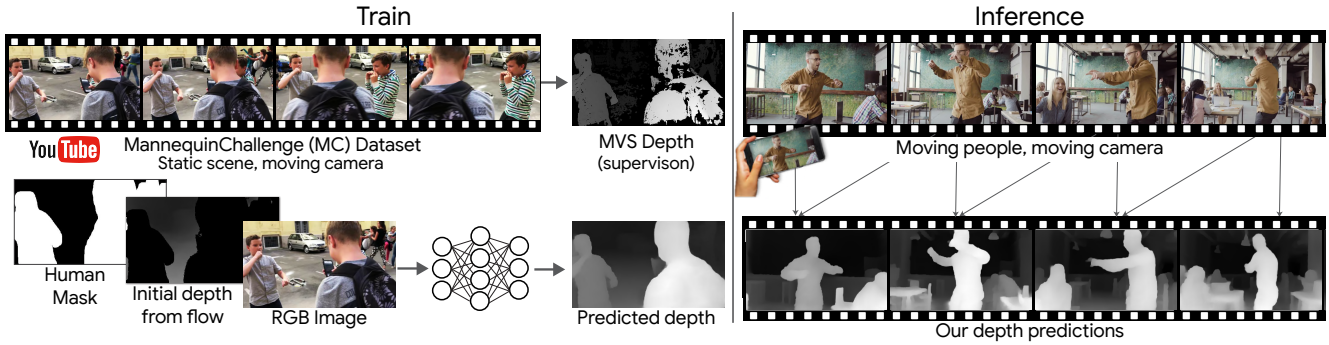


Figure 1. Our model predicts dense depth when both an ordinary camera and people in the scene are freely moving (right). We train our model on our new *MannequinChallenge* dataset—a collection of Internet videos of people imitating mannequins, i.e., freezing in diverse, natural poses, while a camera 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.[2]

Abstract

We present a method for predicting dense depth in scenarios where both a monocular camera and people in the scene are freely moving. 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. Because people are stationary, 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 demonstrate 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 show various 3D effects produced using our predicted depth.

1. Introduction

A hand-held camera viewing a dynamic scene is a common scenario in modern photography. Recovering dense geometry in this case is a challenging task: moving objects violate the epipolar constraint used in 3D vision, 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 objects and the observer are moving, and even when the scene is observed with just one eye [11]. 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 applications (e.g., 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 rather learn these priors from data.

Where do we get data to train such a method? Generating high-quality synthetic data in which both the camera and the people in the scene are naturally moving is very challenging. Depth sensors (e.g., Kinect) can provide useful data, but such data is typically limited to indoor environments and requires significant manual work in capture and process. Furthermore, it is difficult to gather people of different ages and genders 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 (Fig. 2). These videos comprise our new *MannequinChallenge (MC)* dataset, which we plan to release for the research community.

[2]In all figures, we use inverse depth maps for visualization purposes, and refer to them as depth maps.

Because the entire scene, including the people, is stationary, we estimate camera poses and depth using SfM and MVS, and use this derived 3D data as supervision for training.

In particular, we design and train a deep neural network that takes an input RGB image, a mask of human regions, and an initial depth of the environment (i.e., non-human regions), and outputs a dense depth map over the *entire* image, both the environment and the people (see Fig. 1). Note that the initial depth of the environment is computed using motion parallax between two frames of the video, 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 the applicability of our method on a variety of real-world Internet videos, shot with a hand-held camera, 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 depth maps can be used to produce various 3D effects such as synthetic depth-of-field, depth-aware inpainting, and inserting virtual objects into the 3D scene with correct 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 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, have recently been proposed for predicting dense depth from a single RGB image [46, 17, 7, 6, 3, 19, 33, 8, 52, 49, 21, 41]. Some recent learning based methods also consider multiple images, either assuming known camera poses [12, 47] or simultaneously predicting camera poses along with depth [39, 51]. However, none of them is designed to predict the depth of dynamic objects, which is the focus of our work.

Depth estimation for dynamic scenes. RGBD data has been widely used for 3D modeling of dynamic scenes [25, 55, 48, 5, 14], but only a few methods attempt to estimate depth from a monocular camera. Several methods have been proposed to reconstruct sparse geometry of a dynamic scene [27, 50, 36, 40]. Russell *et al*. [31] and Ranftl *et al*. [29] suggest motion/object segmentation based algorithms to decompose a dynamic scene into piecewise rigid parts. However, these methods impose strong assumptions of the object's motion that are violated by articulated human motion. Konstantinos *et al*. [30] predict depth of 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 data for learning depth. There are a number of RGBD datasets of indoor scenes, captured using depth sensors [35, 2, 4, 45] or synthetically rendered [37]. However, none of these datasets provide depth supervision for moving people in natural environments. Several action recognition methods use depth sensors to capture human actions [54, 34, 22, 26], but most use a static camera and provide only a limited number of indoor scenes. REFRESH [20] is a recent semi-synthetic scene flow dataset created by overlaying animated people on NYUv2 images. Here too, the data is limited to interiors and consists of synthetic humans placed in unrealistic configurations with their surrounding.

Human shape and pose prediction. Recovery of a posed 3D human mesh from a single RGB image has attracted significant attention [18, 9, 16, 1, 28, 23]. Recent methods achieve impressive results on natural images spanning a variety of poses. However, such methods only model the human body, disregarding hair, clothing, and the non-human parts of the scenes. Finally, many of these methods rely on correctly detecting human keypoints, requiring most of the body to be within the frame.

3. MannequinChallenge Dataset

The *Mannequin Challenge* [42] is a popular video trend in which people freeze in place—often in an interesting pose—while the camera operator moves around the scene filming them (e.g., Fig. 2). Thousands of such videos have been created and uploaded to YouTube since late 2016. 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. We found around 2,000 candidate videos for which this processing is possible. These videos comprise our new *MannequinChallenge (MC) Dataset*, which spans a wide range of scenes with people of different ages, naturally posing in different group configurations. We next describe in detail how we process the videos and derive our training data.

Estimating camera poses. Following a similar approach to Zhou *et al*. [53], we use ORB-SLAM2 [24] 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 (typical value for modern cell-phone cameras). We then reprocess each sequence at a higher resolution using a visual SfM system [32], which refines the initial camera poses and intrinsic parameters. This method extracts and matches features across frames, then performs a global bundle adjustment optimization. Finally, sequences with non-smooth camera motion are removed using the technique of Zhou *et al*. [53].

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

Figure 2. **Sample images from Mannequin Challenge videos**. Each image is a frame from a video sequence in which the camera is moving but *humans are all static*. The videos span a variety of natural scenes, poses, and configuration of people.

dense depth maps using COLMAP, a state-of-the-art MVS system [33].

Because our data consists of challenging Internet videos that involve camera motion blur, shadows, reflections, etc., the raw depth maps estimated by MVS are often too noisy for training purposes. We address this issue by a careful depth filtering mechanism. We first filter outlier depths using the depth refinement method of [19]. We further remove erroneous depth values by considering the consistency of the MVS depth and the depth obtained from motion parallax between two frames. Specifically, for each frame, we compute a normalized error $\Delta(\mathbf{p})$ for every valid pixel $\mathbf{p}$:

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

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

Fig. 3 shows sample frames from our processed sequences with corresponding estimated MVS depths after filtering. See the supplemental material for examples illustrating the effect of the proposed cleaning approach.

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 (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 width of ≥ 1600 pixels. Finally, we manually inspect the trajectories and point clouds of the remaining sequences and remove obviously incorrect reconstructions. Examples of removed images are shown in the supplemental material.

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 the Mannequin-Challenge dataset in a supervised manner, i.e., by regressing to the depth generated by the MVS pipeline. A key question is how to structure the input to the network to allow training on frozen people but inference on freely moving people. One option is to regress from a single RGB image to depth, but this approach disregards geometric information about the static regions of the scene that is available by considering more than a single view. To benefit from such information, we input to the network a depth map for the static, non-human regions, estimated from motion parallax w.r.t. another view of the scene.

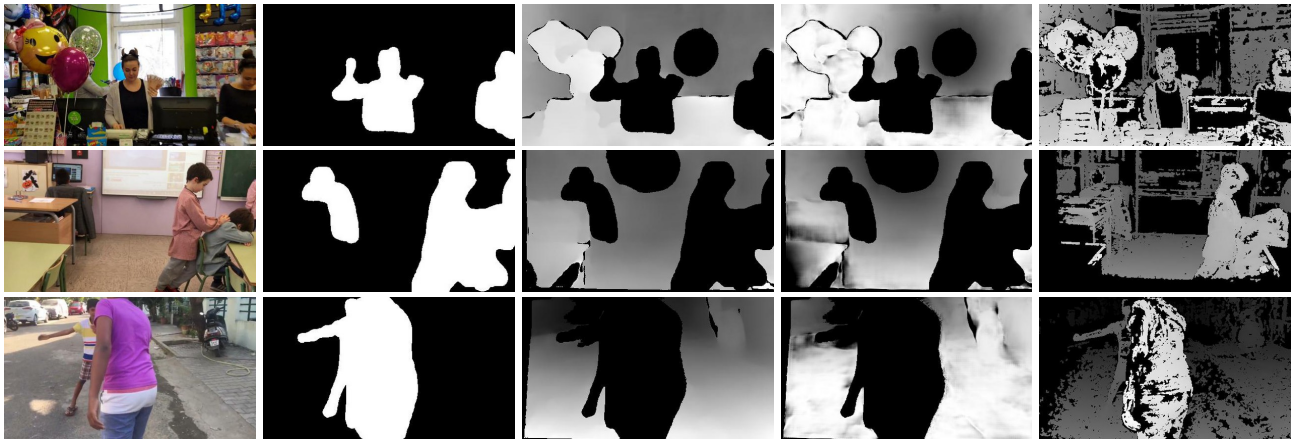
The full input to our network, illustrated in Fig. 3, includes a reference image I^r, a binary mask of human regions M, a depth map estimated from motion parallax (with human regions removed) D_{pp}, a confidence map C, and an optional human keypoint map K. We assume known, accurate camera poses from SfM during both training and inference. In an online inference setting, camera poses can be obtained by 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 entire scene consistent.

Our network architecture is a variant of the hourglass network of [3], with the nearest-neighbor upsampling layers replaced by bilinear upsampling layers.

The following sections describe our model inputs and training losses in detail. In the supplemental material we provide additional implementation details and full derivations.

4.1. Depth from motion parallax

Motion parallax between two frames in a video provides our initial depth estimate for the static regions of the scene (assuming humans are dynamic while the rest of the scene



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

Figure 3. **System inputs and training data.** The input to our network consists of: (a) RGB image, (b) human mask, (c) masked depth computed from motion parallax w.r.t. a selected source image, and (d) 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 is removed. The network is trained to regress to MVS depth (e).

is static). Given a reference image I^r and source image I^s pair, we estimate an optical flow field from I^r to I^s using FlowNet2.0 [13]. Using the relative camera poses between the two views, we compute an initial depth map D_{pp} from the estimated flow field, using the Plane-Plus-Parallax (P+P) representation [15, 43].

In some cases, such as forward/backward relative camera motion between the frames, the estimated depth may be ill-defined in some image regions (i.e., the epipole may be located within the image). We detect and filter out such depth values as described in Sec. 4.2.

Keyframe selection. Depth from motion parallax may 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 apply a baseline criterion when selecting 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. Formally, for each I^r, we find the index s of I^s as

$$s = \arg\max_j d^{rj} o^{rj} \tag{2}$$

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

$$o^{rj} = \frac{2|V^r \bigcap V^j|}{|V^r| + |V^j|}, \tag{3}$$

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. 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 [44], compounding uncertainty in the input depth map, D_{pp}. We thus estimate, and input to the network, a confidence map, C. This allows the network to rely more on the input depth in high-confidence regions, and potentially use it to improve its prediction in low-confidence regions. The confidence value at each pixel $\mathbf{p}$ in the non-human regions is defined as:

$$C(\mathbf{p}) = C_{lr}(\mathbf{p}) C_{ep}(\mathbf{p}) C_{pa}(\mathbf{p}). \tag{4}$$

The term C_{lr} measures "left-right" consistency between the forward and backward flow fields. That is, $C_{lr}(\mathbf{p}) = \max\left(0, 1 - r(\mathbf{p})^2\right)$, where $r(\mathbf{p})$ is the forward-backward warping error. For perfectly consistent forward and backward flows $C_{lr} = 1$, while $C_{lr} = 0$ when the error is greater than 1px.

The term C_{ep} measures how well the flow field complies with the epipolar constraint between the views [10]. Specifically, $C_{ep}(\mathbf{p}) = \max\left(0, 1 - (\gamma(\mathbf{p})/\bar{\gamma})^2\right)$, where $\gamma(\mathbf{p})$ is the distance between the warped pixel position of $\mathbf{p}$ based on its optical flow and its corresponding epipolar line; $\bar{\gamma}$ controls the epipolar distance tolerance (we set $\bar{\gamma} = 2$px in our experiments).

Finally, C_{pa} assigns low confidence to pixels for which the parallax between the views is small [33]. This is measured by the angle $\beta(\mathbf{p})$ between the camera rays meeting at the pixel $\mathbf{p}$. That is, $C_{pa}(\mathbf{p}) = 1 - \left(\frac{\min(\bar{\beta}, \beta(\mathbf{p})) - \bar{\beta}}{\bar{\beta}}\right)^2$, where $\bar{\beta}$ is the angle tolerance (we use $\bar{\beta} = 1^\circ$ in our experiments).

Fig. 3(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.

Figure 4. **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.

4.3. Losses

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

$$\mathcal{L}_{\text{si}} = \mathcal{L}_{\text{MSE}} + \alpha_1 \mathcal{L}_{\text{grad}} + \alpha_2 \mathcal{L}_{\text{sm}}. \quad (5)$$

Scale-invariant MSE. $\mathcal{L}_{\text{MSE}}$ denotes the scale-invariant mean square error (MSE) [6]. This 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. Intuitively, we look at all pairs of points, and penalize the difference in their *ratio* of depth values w.r.t. ground truth.

Multi-scale gradient term. We use a multi-scale gradient term, $\mathcal{L}_{\text{grad}}$, which is the L_1 difference between the predicted log depth derivatives (in x and y directions) and the ground truth log depth derivatives, at multiple scales [19]. This term allows the network to recover sharp depth discontinuities and smooth gradient changes in the predicted depth images.

Multi-scale, edge-aware smoothness terms. To encourage smooth interpolation of depth in texture-less regions where MVS fails to recover depth, we use a simple smoothness term, $\mathcal{L}_{\text{sm}}$, which penalizes L_1 norm of log depth derivatives based on the first- and second-order derivatives of images and is applied at multiple scales [41]. This term encourages piecewise smoothness in depth regions where there is no image intensity change.

	Net 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.	$ID_{\text{pp}}M$	0.255	0.229	0.264	0.243	0.285
IV.	$ID_{\text{pp}}CM$	0.232	**0.188**	0.237	0.221	0.268
V.	$ID_{\text{pp}}CMK$	**0.227**	**0.189**	**0.230**	**0.212**	**0.263**

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 additional confidence for IV.; in V, we also input human keypoints. Lower is better for all metrics.

5. Results

We tested 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.

Error metrics. We measure error using the scale-invariant RMSE (si-RMSE), equivalent to $\sqrt{\mathcal{L}_{\text{MSE}}}$, described in Sec. 4.3. We evaluate si-RMSE on 5 different regions: **si-full** measures the error between all pairs of pixels, giving the overall accuracy across the entire image; **si-env** measures pairs of pixels in non-human regions $\mathcal{E}$, providing depth accuracy of the environment; and **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: **si-intra** measures si-RMSE within $\mathcal{H}$, or human accuracy independent of the environment; **si-inter**

(a) I^r (b) I^s (c) GT (d) DORN [7] (e) DeMoN [39] (f) Ours (RGB) (g) Ours (full)

Figure 5. **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) single view depth prediction method DORN [7], (e) two-frame motion stereo DeMoN [39], (f-g) depth predictions from our single view and two-frame models, respectively.

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 supplemental 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 Sec. 3) are considered as ground truth.

To quantify the importance of our designed 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 a 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 environment as well as a 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 given to the network. This result indicates that the network is not able to implicitly learn the relative poses and extract the depth information.

Fig. 4 shows qualitative comparisons between our single-view model (I) and our full model ($ID_{\text{pp}}CMK$). Our full model results are more accurate in both human regions (e.g., first column) and non-human regions (e.g., second column). In addition, the depth relations between people and their surroundings are improved in all examples.

5.2. Evaluation on TUM RGBD dataset

We used a subset of the TUM RGBD dataset [38], which contains indoor scenes of people performing complex actions, captured from different camera poses. Sample images from this dataset are shown in Fig. 5(a-b).

To run our model, we first estimate camera poses using ORB-SLAM2 [3]. In some cases, due to severe low image quality, motion blur and rolling shutter effects, the estimated camera poses may 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 evaluations.

We compare our depth predictions (using our MC trained models) with several state-of-the-art monocular depth prediction methods trained on indoor NYUv2 [17, 46, 7] and Depth

[3] We found estimates from ORB-SLAM2 to be better synchronized with the RGB images than the ground truth poses provided by the TUM dataset.

Methods	Dataset	two-view?	**si-full**	**si-env**	**si-hum**	**si-intra**	**si-inter**	**RMSE**	**Rel**
Russell *et al*. [31]	-	Yes	2.146	2.021	2.207	2.206	2.093	2.520	0.772
DeMoN [39]	RGBD+MVS	Yes	0.338	0.302	0.360	0.293	0.384	0.866	0.220
Chen *et al*. [3]	NYU+DIW	No	0.441	0.398	0.458	0.408	0.470	1.004	0.262
Laina *et al*. [17]	NYU	No	0.358	0.356	0.349	0.270	0.377	0.947	0.223
Xu *et al*. [46]	NYU	No	0.427	0.419	0.411	0.302	0.451	1.085	0.274
Fu *et al*. [7]	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
$ID_{\text{pp}}CM$ (w/o d. cleaning)	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 TUM RGBD datasets.** 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. *w/o d. cleaning* indicates the model is trained using raw MVS depth predictions as supervision, without our depth cleaning method. Dataset '-' indicates the method is not learning based. Lower is better for all error metrics.

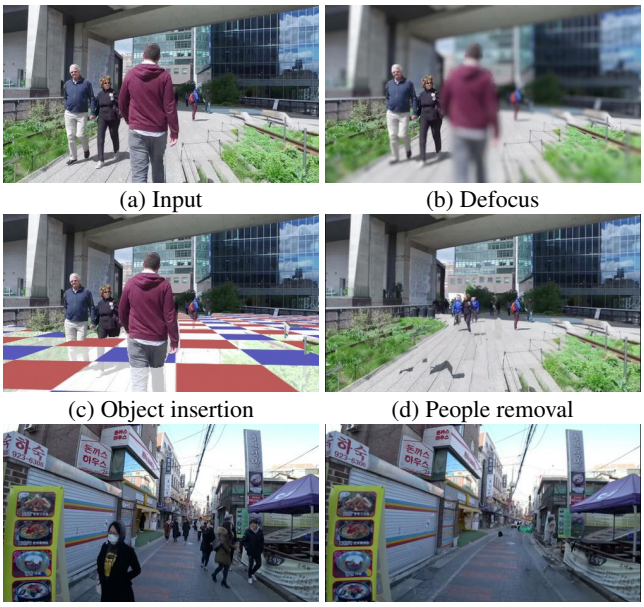
(a) I^r (b) I^s (c) DORN [7] (d) Chen *et al*. [3] (e) DeMoN [39] (f) Ours (full)

Figure 6. **Comparisons on Internet video clips with moving cameras and people.** From left to right: (a) reference image, (b) source image, (c) DORN [7], (d) Chen *et al*. [3], (e) DeMoN [39], (f) our full method.

in the Wild (DIW) datasets [3], and the recent two-frame stereo model DeMoN [39], which assumes a static scene. We also compare with Video-Popup [31], which deals with dynamic scenes. We use the same image pairs for computing D_{pp} as inputs to DeMoN and Video-Popup.

Quantitative comparisons are show in Table 2, where we report 5 different scale-invariance error measures as well as standard RMSE and relative error; the last two are computed by applying a single scaling factor that aligns the predicted and ground-truth depth 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 [31] failed to produce meaningful results due to the challenging camera and object motion. Our full model, by making use of the initial (masked) depth map, significantly improves performance for all the error measures. Consistent with our MC test set results, when we use optical flow as input (instead of initial depth map) the performance is only slightly better than the single-view network. Finally, we show the importance of our proposed "depth cleaning" method, applied to the training data (see Eq. 1). Compared to the same model, only trained using the raw MVS depth predictions as supervision ("w/o d. cleaning"), we see a drop of about 15% in performance.

Fig. 5 shows qualitative comparison between the different methods. Our models' depth predictions (Fig. 5(f-g)) strongly resemble the ground truth and show high level of details and sharp depth discontinuities. This result is a no-



(a) Input (b) Defocus

(c) Object insertion (d) People removal

(e) Input (f) People removal

Figure 7. **Depth-based visual effects.** We use our predicted depth maps to apply depth-aware visual effects on (a, e) input images; we show (b) defocus, (c) object insertion, and (d, f) people removal with inpainting results.

table improvement over competing methods, which often produce significant errors in both human regions (e.g., legs in the second row of Fig. 5), and 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), involving simultaneous natural camera motion and human motion. Our SLAM/SfM pipeline was used to generate sequences ranging from 5 seconds 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 ($ID_{\text{pp}}CMK$) with several recent learning based depth prediction models: DORN [7], Chen *et al*. [3], and DeMoN [39]. 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 [3], 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 Fig. 6, our depth predictions are significantly better than the baseline methods. In particular, DORN [7] has very limited generalization to Internet videos, and Chen *et al*. [3], 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 predicted depth maps depict accurate depth ordering both between people and other objects in the scene (e.g., between people and buildings, fourth row of Fig. 6), and within human regions (such as the arms and legs of people in the first three rows of Fig. 6).

Figure 8. **Failure cases.** Moving, non-human objects such as cars and shadows can cause bad estimates (left and middle, boxed); fine structures such as limbs may be blurred for distant people in challenging poses (right, boxed).

Depth-based visual effects. Our depth can be used to apply a range of depth-based visual effects. Fig. 7 shows depth-based defocus, insertion of synthetic 3D graphics, and removal of nearby humans with inpainting. See the supplemental material for additional examples, including mono-to-stereo conversion.

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. Fig. 7 (d, f) show 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 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 of 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 "frozen" in natural poses. We showed how to obtain reliable depth supervision from such noisy data, and demonstrated that our models significantly improve over state-of-the-art methods.

Our approach still has limitations. We assume known camera poses, which may difficult to infer if moving objects cover most of the scene. In addition, the predicted depth may be inaccurate for non-human, moving regions such as cars and shadows (Fig. 8). Our approach also only uses two views, sometimes leading to temporally inconsistent depth estimates. However, we hope this work can guide and trigger further progress in monocular dense reconstruction of dynamic scenes.

References

[1] 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)*, 2016. ii

[2] 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)*, 2017. ii

[3] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Neural Information Processing Systems*, pages 730–738, 2016. ii, iii, vii, viii

[4] 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)*, 2017. ii

[5] 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. ii

[6] 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*, pages 2366–2374, 2014. ii, v

[7] 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. ii, vi, vii, viii

[8] 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. ii

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

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

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

[12] 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. ii

[13] 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. iv

[14] 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. ii

[15] 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. iv

[16] 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. ii

[17] 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. ii, vi, vii

[18] 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. ii

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

[20] 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. ii

[21] 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. ii

[22] 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. ii

[23] 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. ii

[24] 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. ii

[25] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2015. ii

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

[27] 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. ii

[28] 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)*, 2017. ii

[29] 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. ii

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

[31] 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. ii, vii

[32] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2016. ii

[33] J. L. Schönberger, 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. ii, iii, iv

[34] 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. ii

[35] 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. ii

[36] 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. ii

[37] 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. ii

[38] 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. vi

[39] 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. ii, vi, vii, viii

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

[41] 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. ii, v

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

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

[44] 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. iv

[45] 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. ii

[46] 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. ii, vi, vii

[47] 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)*, 2018. ii

[48] 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. ii

[49] 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. ii

[50] 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. ii

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

[52] 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. ii

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

[54] 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. ii

[55] 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. ii