Supplementary Material
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

This document includes the following:

1. Examples of filtered images and human keypoint images from our MannequinChallenge dataset (using our data generation pipeline) and examples of demonstrating our proposed depth cleaning approach, (see Section 3 in the paper).
2. Mathematical details of our depth prediction models (described in Section 4 in the main paper).
3. Implementation details of our training and experiments.
4. Qualitative comparison to parametric human model fitting.

1. Dataset

Examples of filtered images Figure 1 and Figure 2 show examples of images filtered out by our data creation pipeline from the raw MannquinChallenge video clips. These examples include images captured by fisheye cameras, and images with large regions of synthetic background or moving objects.

Examples of human keypoints images Figure 3 shows example images of human keypoints predicted by Mask-RCNN [4]. For visualization purpose, we perform morphological dilation to original keypoint image to make each keypoint location more visisble. Moreover, we use different color to visualize different human joints keypoints.

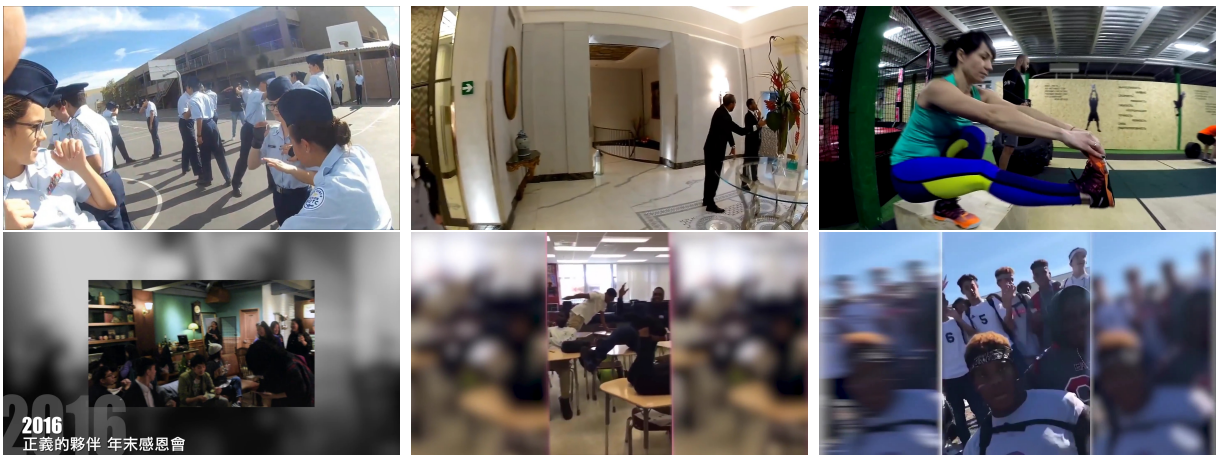


Figure 1: **Examples of filtered images**. First row shows the images captured by fisheye cameras and second row shows the images with synthetic background;

Figure 2: **Examples of filtered images**. Each column depicts an example of filtered images from our pipeline due to moving objects.

Figure 3: **Examples of keypoints images**. Top rows depicts examples of original images and bottom row depicts corresponding human keypoint images with different color indicating different human joints.

Depth cleaning examples Figure 4 shows examples of our depth cleaning method for MVS depth, as described in Section 3 of the paper. The regions circled in yellow show MVS depth with and without our proposed depth cleaning method based on Equation 1 in the paper. Our cleaning method removes incorrect depth values. These depth maps serve as supervision in training, thus careful filtering has large impact on our performance, as demonstrated in our TUM RGBD experiments.

2. Derivations and additional mathematical details

We provide detailed derivations of our inputs to the depth prediction model (Section 4 in the paper).

Suppose we have relative camera pose $R \in SO(3), \mathbf{t} \in \mathbb{R}^3$ from source image view I^s to reference image view I^r with common intrinsic matrix K estimated from visual SfM system. In addition, we also compute forward flow F from I^r to I^s, and backward flow B from I^s to I^r. Let $\mathbf{p}$ denote the 2D pixel position in I^r, and $\mathbf{p}' = \mathbf{p} + F(\mathbf{p})$ the corresponding 2D pixel position in I^s that is warped by $F(\mathbf{p})$; we denote such positions in either $\mathbb{R}^2$ pixel space or $\mathbb{R}^3$ homogeneous space based on context.

2.1. Depth from motion parallax

We estimate our initial input depth from optical flow and camera motion using Plane-plus-Parallax (P+P) representation [6]. Note that P+P is typically used to estimate the structure component of the scene with respect to a reference plane, either a scene plane or a virtual one. In our case, we use P+P as means to cancel out the relative camera rotation and to efficiently invert the flow field to a depth map. Therefore, we set the reference plane to be at infinity.

Let Π denote a real or virtual planar surface, and let d'_Π denote the distance between camera center of source image I^s and plane Π, H is distance between the 3D scene point corresponding to 2D pixel $\mathbf{p}$ and Π. It can be shown (See Appendix of [6] for complete math derivations) that

$$\mathbf{p} = \mathbf{p}_w + \frac{H}{D_{\text{pp}}(\mathbf{p})}\frac{t_{[3]}}{d'_\Pi}\mathbf{p}_w - \frac{H}{D_{\text{pp}}(\mathbf{p})d'_\Pi}K\mathbf{t} \tag{1}$$

$$= \mathbf{p}_w + \frac{H}{D_{\text{pp}}(\mathbf{p})d'_\Pi}(t_{[3]}\mathbf{p}_w - K\mathbf{t}) \tag{2}$$

where $D_{pp}(\mathbf{p})$ is initial estimated depth at $\mathbf{p}$ in reference image I^r; $t_{[3]}$ is the third component of translation vector $\mathbf{t}$, and $\mathbf{p}_w$ is the 2D image point in I^r that results from inverse warping the corresponding 2D pixel $\mathbf{p}' = \mathbf{p} + F(\mathbf{p})$ in I^s by a homography A:

$$\mathbf{p}_w = \frac{A\mathbf{p}'}{\mathbf{a}_3^T\mathbf{p}'} \tag{3}$$

$$\text{where } A = K(R + \mathbf{t}\frac{\mathbf{n}'^T}{d'_\Pi})K^{-1}$$

where $\mathbf{a}_3^T$ is the third row of A, and $\mathbf{n}'$ is normal of plane Π with respect to the camera of source image I^s. Note that the original paper [6] divides the P+P representation into two cases depending on whether $T_z = 0$, but we combine these two cases into one equation shown in Equation 2 by simple algebraic manipulations.

Now, if we set plane Π at infinity, using L'Hôpital's rule, we can cancel out H and d'_Π and obtain following equations:

$$\mathbf{p} = \mathbf{p}_w + \frac{t_{[3]}\mathbf{p}_w - K\mathbf{t}}{D_{\text{pp}}(\mathbf{p})} \tag{4}$$

$$D_{pp}(\mathbf{p}) = \frac{||t_{[3]}\mathbf{p}_w - K\mathbf{t}||_2}{||\mathbf{p} - \mathbf{p}_w||_2},$$

$$\text{where } \mathbf{p}_w = \frac{A'\mathbf{p}'}{\mathbf{a}_3'^T\mathbf{p}'} \text{ and } A' = KRK^{-1}$$

We use P+P representation to estimate initial depth because we found it more efficient and robust for dense depth estimation compared with standard triangulation methods, which are usually used with sparse correspondences. Equation 4 can also be extended to multiple frames with importance weights by formulating it as a weighted least square problem.

2.2. Confidence

Recall the confidence value at each pixel $\mathbf{p}$ in the non-human (environment) regions $\mathcal{E}$ of the image is defined as:

$$C(\mathbf{p}) = C_{lr}(\mathbf{p})C_{ep}(\mathbf{p})C_{pa}(\mathbf{p}) \tag{5}$$

C_{lr} is a confidence based on left-right consistency between the estimated forward and backward flow fields. That is, $C_{lr}(\mathbf{p}) = \max\left(0, 1 - r(\mathbf{p})^2\right)$, where $r(\mathbf{p}) = ||F(\mathbf{p}) + B(\mathbf{p}')||_2$ is the forward-backward optical flow warping error.

C_{ep} gives low confidence to pixels where the flow field and the epipolar constraint disagree [3]. Specifically, $C_{ep}(\mathbf{p}) = \max\left(0, 1 - (\gamma(\mathbf{p})/\bar{\gamma})^2\right)$, where geometric epipolar distance $\gamma(\mathbf{p})$ is defined as:

$$\gamma(\mathbf{p}) = \frac{|\mathbf{p}'^T \mathbf{F} \mathbf{p}|}{\sqrt{(\mathbf{F}\mathbf{p})_{[1]}^2 + (\mathbf{F}\mathbf{p})_{[2]}^2}} \tag{6}$$

where $\mathbf{F} = K^{-T}[\mathbf{t}]_\times R K^{-1}$ is the fundamental matrix and $(\mathbf{F}\mathbf{p})_{[i]}$ is the ith element of $\mathbf{F}\mathbf{p}$.

$C_{pa}(\mathbf{p})$ is a confidence based on parallax angles: $C_{pa}(\mathbf{p}) = 1 - \left(\frac{\min(\bar{\beta}, \beta(\mathbf{p})) - \bar{\beta}}{\bar{\beta}}\right)^2$ [11], where $\beta(\mathbf{p}) = \cos^{-1}\left(\frac{\mathbf{b}(\mathbf{p})\mathbf{b}(\mathbf{p}')}{||\mathbf{b}(\mathbf{p})||_2 ||\mathbf{b}(\mathbf{p}')||_2}\right)$, and $\mathbf{b}(\mathbf{p}) = K^{-1}\mathbf{p}$ is a bearing vector at $\mathbf{p}$ in I^r, and $\mathbf{b}(\mathbf{p}') = K^{-1}\mathbf{p}'$ is a bearing vector at $\mathbf{p}'$ in I^s.

2.3. Losses

Our loss is computed on log-space depth values and consists of three terms (Section 4.3 in the paper):

$$\mathcal{L}_{\mathsf{si}} = \mathcal{L}_{\mathsf{MSE}} + \alpha_1 \mathcal{L}_{\mathsf{grad}} + \alpha_2 (\mathcal{L}_{\mathsf{sm}}^1 + \mathcal{L}_{\mathsf{sm}}^2). \tag{7}$$

Scale-invariant MSE. $\mathcal{L}_{\mathsf{MSE}}$ denotes the scale-invariant mean square error (MSE) adopted from [2]. 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. Intuitively, it penalizes differences in the ratio of depth between two predicted depth values relative to the same ratio in the ground truth:

$$\mathcal{L}_{\mathsf{MSE}} = \frac{1}{N} \sum_{\mathbf{p} \in I} R(\mathbf{p})^2 - \frac{1}{N^2} \left(\sum_{\mathbf{p} \in I} R(\mathbf{p}) \right)^2 \tag{8}$$

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

Multi-scale gradient term. We use a multi-scale gradient term to encourage smoother gradient changes and sharper depth discontinuities in the predicted depth images [10]:

$$\mathcal{L}_{\mathsf{grad}} = \sum_{s=0}^{S-1} \frac{1}{N_s} \sum_{\mathbf{p} \in I_s} \left(|\nabla_x R_s(\mathbf{p})| + |\nabla_y R_s(\mathbf{p})| \right) \tag{9}$$

where subscript s of R_s and I_s indicates scale s and N_s is 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 a simple smoothness term at multiple scales based on the first- and second-order derivatives of images [12]:

$$\mathcal{L}_{\mathsf{sm}}^1 = \sum_{s=0}^{S-1} \frac{1}{N_s 2^s} \sum_{\mathbf{p} \in I_s} \exp(-|\nabla I_s(\mathbf{p})|) |\nabla \log \hat{D}(\mathbf{p})| \tag{10}$$

$$\mathcal{L}_{\mathsf{sm}}^2 = \sum_{s=0}^{S-1} \frac{1}{N_s 2^s} \sum_{\mathbf{p} \in I_s} \exp(-|\nabla^2 I_s(\mathbf{p})|) |\nabla^2 \log \hat{D}(\mathbf{p})| \tag{11}$$

We create $S = 5$ scale image pyramids using nearest-neighbor down-sampling for both multi-scale gradient and smoothness terms.

2.4. Error Metrics

Recall in Section 5 of our main paper, we measure 5 different error metrics based on scale-invariant RMSE (si-RMSE). Here we provide definition of each error metric. Notice we can use similar algebraic manipulations to those proposed in [9] to evaluate all terms in time *linear* in the number of pixels.

Recall that $\hat{D}$ is the predicted depth and D_{gt} is the ground truth depth, and we define $R(\mathbf{p}) = \log \hat{D}(\mathbf{p}) - \log D_{\text{gt}}(\mathbf{p})$. Recall we also define human regions as $\mathcal{H}$ with N_h valid depth, non-human (environment) regions as $\mathcal{E}$ with N_e valid depth, and full image region as $I = \mathcal{H} \cup \mathcal{E}$ with $N = N_e + N_h$ valid depth.

Specifically, **si-full** measures si-RMSE between all pairs of pixels, giving the overall accuracy across the entire image and is defined as follows:

$$\textbf{si-full} = \frac{1}{N^2} \sum_{\mathbf{p}\in I} \sum_{\mathbf{q}\in I} \Big((\log \hat{D}(\mathbf{p}) - \hat{D}(\mathbf{q})) - (\log D_{\text{gt}}(\mathbf{p}) - D_{\text{gt}}(\mathbf{q})) \Big)^2 \tag{12}$$

$$= \frac{1}{N^2} \sum_{\mathbf{p}\in I} \sum_{\mathbf{q}\in I} (R(\mathbf{p}) - R(\mathbf{q}))^2 \tag{13}$$

$$= \frac{1}{N^2} \sum_{\mathbf{p}\in I} \sum_{\mathbf{q}\in I} R(\mathbf{p})^2 + R(\mathbf{q})^2 - 2R(\mathbf{p})R(\mathbf{q}) \tag{14}$$

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

$$= \frac{2}{N^2} \left(N \sum_{\mathbf{p}\in I} R(\mathbf{p})^2 - \sum_{\mathbf{p}\in I} R(\mathbf{p}) \sum_{\mathbf{q}\in I} R(\mathbf{q}) \right) \tag{16}$$

si-env measures pairs of pixels in non-human regions $\mathcal{E}$, giving accuracy of the environment, and is defined as:

$$\textbf{si-env} = \frac{1}{N_e^2} \sum_{\mathbf{p}\in \mathcal{E}} \sum_{\mathbf{q}\in \mathcal{E}} \Big((\log \hat{D}(\mathbf{p}) - \hat{D}(\mathbf{q})) - (\log D_{\text{gt}}(\mathbf{p}) - D_{\text{gt}}(\mathbf{q})) \Big)^2 \tag{17}$$

$$= \frac{1}{N_e^2} \sum_{\mathbf{p}\in \mathcal{E}} \sum_{\mathbf{q}\in \mathcal{E}} (R(\mathbf{p}) - R(\mathbf{q}))^2 \tag{18}$$

$$= \frac{2}{N_e^2} \left(N_e \sum_{\mathbf{p}\in \mathcal{E}} R(\mathbf{p})^2 - \sum_{\mathbf{p}\in \mathcal{E}} R(\mathbf{p}) \sum_{\mathbf{q}\in \mathcal{E}} R(\mathbf{q}) \right) \tag{19}$$

si-hum measures pairs where one pixel lies in the human region $\mathcal{H}$ and one lies anywhere in the image, giving accuracy for people, and is defined as :

$$\textbf{si-hum} = \frac{1}{NN_h} \sum_{\mathbf{p}\in \mathcal{H}} \sum_{\mathbf{q}\in I} \Big((\log \hat{D}(\mathbf{p}) - \hat{D}(\mathbf{q})) - (\log D_{\text{gt}}(\mathbf{p}) - D_{\text{gt}}(\mathbf{q})) \Big)^2 \tag{20}$$

$$= \frac{1}{NN_h} \sum_{\mathbf{p}\in \mathcal{H}} \sum_{\mathbf{q}\in I} (R(\mathbf{p}) - R(\mathbf{q}))^2 \tag{21}$$

$$= \frac{1}{NN_h} \sum_{\mathbf{p}\in \mathcal{H}} \sum_{\mathbf{q}\in I} R(\mathbf{p})^2 + R(\mathbf{q})^2 - 2R(\mathbf{p})R(\mathbf{q}) \tag{22}$$

$$= \frac{1}{NN_h} \left(N \sum_{\mathbf{p}\in \mathcal{H}} R(\mathbf{p})^2 + N_h \sum_{\mathbf{q}\in I} R(\mathbf{q})^2 - 2 \sum_{\mathbf{p}\in \mathcal{H}} R(\mathbf{p}) \sum_{\mathbf{q}\in I} R(\mathbf{q}) \right) \tag{23}$$

$$\tag{24}$$

Furthermore, **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, and is defined as

$$\textbf{si-intra} = \frac{1}{N_h^2} \sum_{\mathbf{p}\in\mathcal{H}} \sum_{\mathbf{q}\in\mathcal{H}} \Big((\log \hat{D}(\mathbf{p}) - \hat{D}(\mathbf{q})) - (\log D_{\text{gt}}(\mathbf{p}) - D_{\text{gt}}(\mathbf{q})) \Big)^2 \tag{25}$$

$$= \frac{1}{N_h^2} \sum_{\mathbf{p}\in\mathcal{H}} \sum_{\mathbf{q}\in\mathcal{H}} (R(\mathbf{p}) - R(\mathbf{q}))^2 \tag{26}$$

$$= \frac{2}{N_h^2} \left(N_h \sum_{\mathbf{p}\in\mathcal{H}} R(\mathbf{p})^2 - \sum_{\mathbf{p}\in\mathcal{H}} R(\mathbf{p}) \sum_{\mathbf{q}\in\mathcal{H}} R(\mathbf{q}) \right) \tag{27}$$

si-inter measures si-RMSE between pixels in $\mathcal{H}$ and in $\mathcal{E}$, or human accuracy w.r.t. the environment and is defined as:

$$\textbf{si-inter} = \frac{1}{N_e N_h} \sum_{\mathbf{p}\in\mathcal{H}} \sum_{\mathbf{q}\in\mathcal{E}} \Big((\log \hat{D}(\mathbf{p}) - \hat{D}(\mathbf{q})) - (\log D_{\text{gt}}(\mathbf{p}) - D_{\text{gt}}(\mathbf{q})) \Big)^2 \tag{28}$$

$$= \frac{1}{N_e N_h} \sum_{\mathbf{p}\in\mathcal{H}} \sum_{\mathbf{q}\in\mathcal{E}} (R(\mathbf{p}) - R(\mathbf{q}))^2 \tag{29}$$

$$= \frac{1}{N_e N_h} \sum_{\mathbf{p}\in\mathcal{H}} \sum_{\mathbf{q}\in\mathcal{E}} R(\mathbf{p})^2 + R(\mathbf{q})^2 - 2R(\mathbf{p})R(\mathbf{q}) \tag{30}$$

$$= \frac{1}{N_e N_h} \left(N_e \sum_{\mathbf{p}\in\mathcal{H}} R(\mathbf{p})^2 + N_h \sum_{\mathbf{q}\in\mathcal{E}} R(\mathbf{q})^2 - 2 \sum_{\mathbf{p}\in\mathcal{H}} R(\mathbf{p}) \sum_{\mathbf{q}\in\mathcal{E}} R(\mathbf{q}) \right) \tag{31}$$

$$\tag{32}$$

3. Implementation Details

We use FlowNet2.0 [5] to estimate optical flow because we found it handles large displacements well and preserves sharp motion discontinuities. We use Mask-RCNN [4] to generate human masks and optionally 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. We normalize human keypoints between 0 and 1 before we feed them into network, if needed.

We adopted the networks architecture of [1] except that we replace all the nearest neighbor upsampling layers with bilinear upsampling layers since we found such simple modification could produce sharper depth boundaries while slightly improve performance. We refer readers to [1] for full details of network architectures.

Our network predicts log depth in both 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 40 and 60 percentile of valid input $\log D_{\text{pp}}$. During inference, we normalize input log-depth by subtracting the median of $\log(D_{pp})$. Additionally, during training, we randomly set to zero the initial input depth and confidence (with probability 0.1) to tackle the potential situation where input depth is not available (e.g. camera is nearly static or estimated optical flow is completely incorrect) in inference stage. When we input human keypoints into network, 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.

For our experiments we train our networks for 20 epochs from scratch using the Adam [8] optimizer with initial learning rate of 0.0004 and we halve the learning rate every 8 epochs. During training, we firstly downsample all images to a resolution of 532x299, 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 512x288. We set hyperparameters in our loss terms $\alpha_1 = 0.5, \alpha_2 = 0.05$ based on our validation set. For the experiments on the TUM RGBD dataset, we downsample ground truth to 512x384 and perform morphological erosion with radius 2 to ground truth for all evaluations since we found depth from RGBD sensor is not well aligned with image edges due to synchronization and small regions of depth captured by depth sensors are usually attributed to outliers due to sensor noise. Additionally, we downsample images to 512x384 for our network, and we downsample input images to provided default image resolutions for other state-of-the-art single-view and motion stereo models (since we found input default resolutions always produce the best performance for other methods) and upsample their depth predictions to 512x384 before we measure the error metrics.

4. Human Mesh Reconstruction

We provide a qualitative comparison to a state-of-the-art parametric human model fitting approach [7] on one of our videos. As can be seen in Figure 5, the model fitting fails to capture the complex poses of the limbs of the the human. Parametric model fitting also does not capture fine details such as clothes and hair.

References

[1] 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.

[2] 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.

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

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

[5] 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)*, pages 2462–2470, 2017.

[6] 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. Springer, 1996.

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

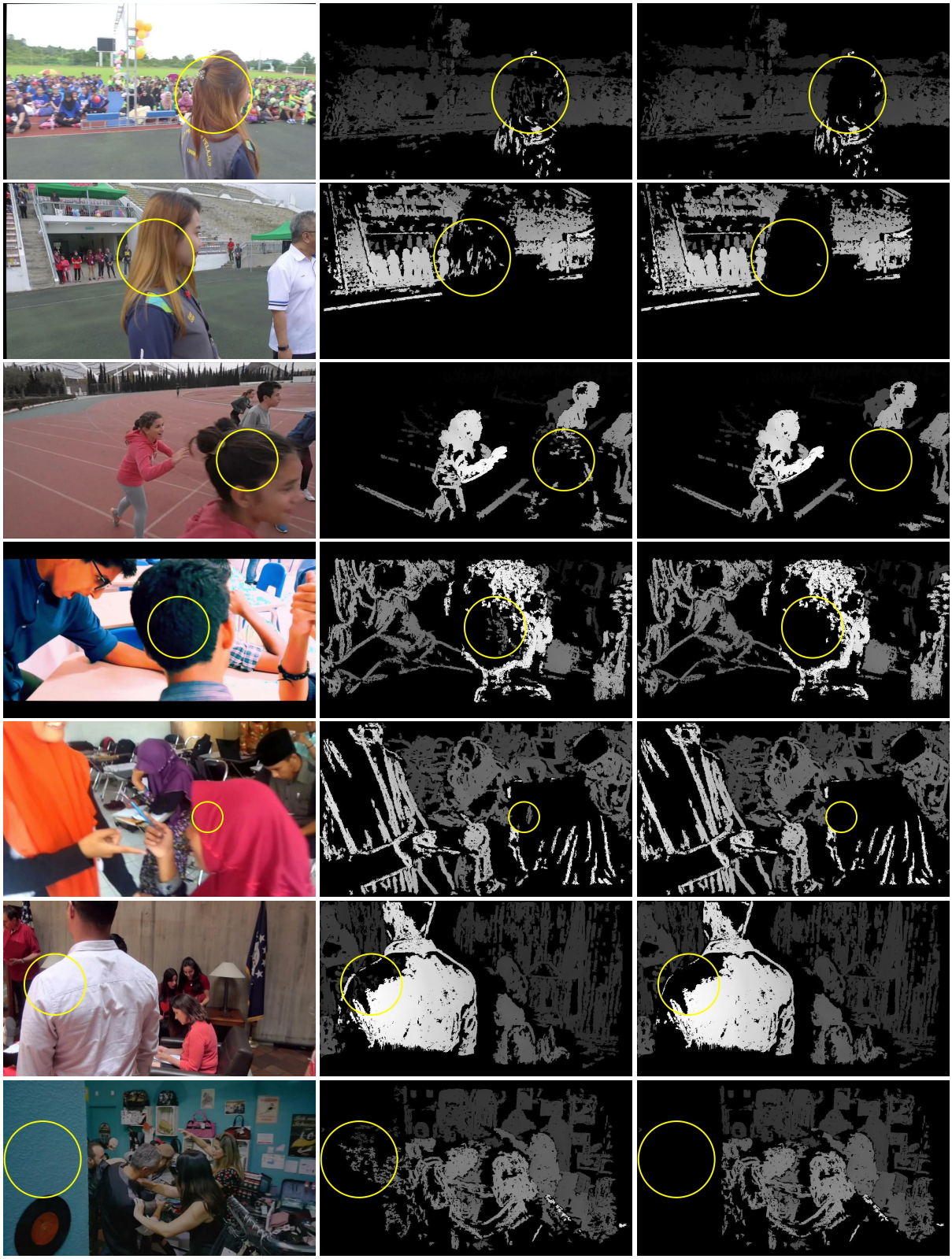
[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[9] Z. Li and N. Snavely. Learning Intrinsic Image Decomposition from Watching the World. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

[11] 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.

[12] 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)*, June 2018.



(a) Image (b) D_{MVS} w/o depth cleaning (c) D_{MVS} w/ depth cleaning

Figure 4: **Effects of proposed depth cleaning method.** See regions circled in yellow. Proposed depth cleaning method using Eq. (1) in our main paper removes outliers of MVS depth D_{MVS} significantly.

(a) (b)

Figure 5: Human shape and pose estimation [7]: (a) the projected mesh outline marked in white on top of the image; (b, left) view of the reconstructed mesh from the camera direction, (b, right) second view of the reconstructed mesh.