

000 054
001 055
002 056
003 057
004 058
005 059
006 060
007 061
008 062
009 063
010 064
011 065
012 066
013 067
014 This document includes the following: 068
015
016 1. Examples of filtered images and human keypoint images from our MannequinChallenge dataset (using our data 070
017 generation pipeline) and examples of demonstrating our proposed depth cleaning approach, (see Section 3 in the paper). 071
018
019 2. Mathematical details of our depth prediction models (described in Section 4 in the main paper). 072
020
021 3. Implementation details of our training and experiments. 073
022
023 4. Qualitative comparison to parametric human model fitting. 074
024
025
026

1. Dataset

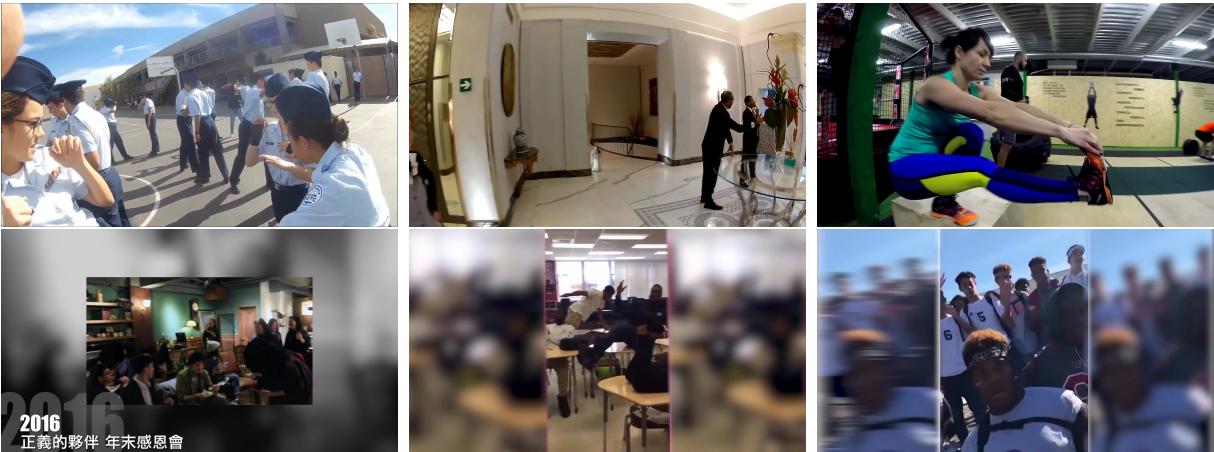
027 **Examples of filtered images** Figure 1 and Figure 2 show examples of images filtered out by our data creation pipeline from 028 the raw MannquinChallenge video clips. These examples include images captured by fisheye cameras, and images with large 029 regions of synthetic background or moving objects.
030
031 **Examples of human keypoints images** Figure 3 shows example images of human keypoints predicted by Mask-RCNN [4].
032 For visualization purpose, we perform morphological dilation to original keypoint image to make each keypoint location more
033 visisble. Moreover, we use different color to visualize different human joints keypoints.
034
035 
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053

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.

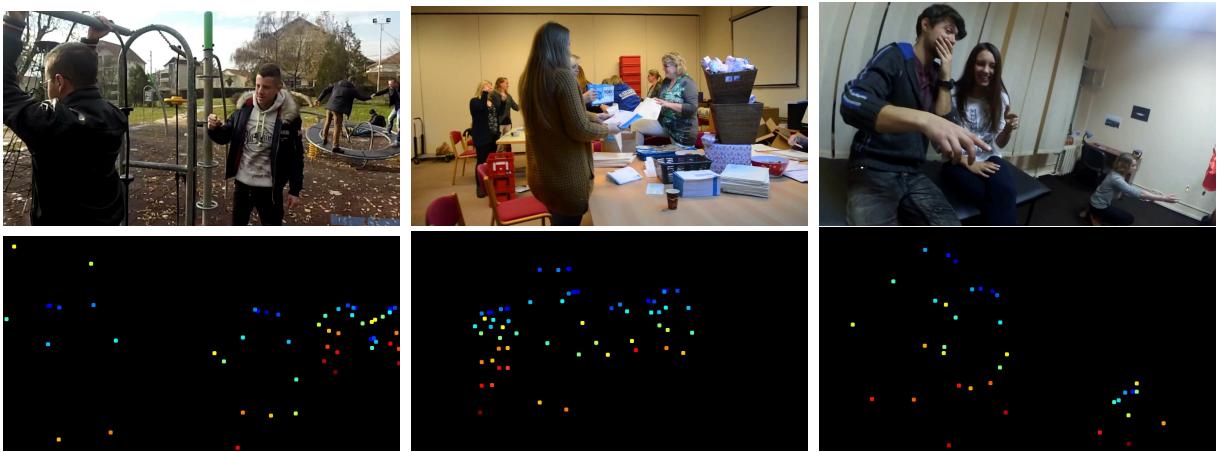


Figure 3: **Examples of keypoints images.** Top rows depicts examples of original images and bottom row depicts corresponding human keypoint images with different color indicating different human joints.

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.

216 **2. Derivations and additional mathematical details** 270
217218 We provide detailed derivations of our inputs to the depth prediction model (Section 4 in the paper). 271
219220 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 273
221 common intrinsic matrix K estimated from visual SfM system. In addition, we also compute forward flow F from I^r to I^s , 274
222 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 275
223 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 276
224 context. 277
225226 **2.1. Depth from motion parallax** 278
227228 We estimate our initial input depth from optical flow and camera motion using Plane-plus-Parallax (P+P) representation [6]. 279
229 Note that P+P is typically used to estimate the structure component of the scene with respect to a reference plane, either a 280
230 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 281
231 invert the flow field to a depth map. Therefore, we set the reference plane to be at infinity. 282
232233 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 283
234 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] 284
235 for complete math derivations) that 285
236

237
$$\mathbf{p} = \mathbf{p}_w + \frac{H}{D_{pp}(\mathbf{p})d'_{\Pi}} \frac{t_{[3]}}{d'_{\Pi}} \mathbf{p}_w - \frac{H}{D_{pp}(\mathbf{p})d'_{\Pi}} K\mathbf{t} \quad (1)$$
 286
238

239
$$= \mathbf{p}_w + \frac{H}{D_{pp}(\mathbf{p})d'_{\Pi}} (t_{[3]}\mathbf{p}_w - K\mathbf{t}) \quad (2)$$
 287
240

241 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 293
242 \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 294
243 homography A : 295
244

245
$$\mathbf{p}_w = \frac{A\mathbf{p}'}{\mathbf{a}_3^T \mathbf{p}'} \quad (3)$$
 296
246

247
$$\text{where } A = K(R + \mathbf{t} \frac{\mathbf{n}'^T}{d'_{\Pi}})K^{-1}$$
 297
248

249 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 302
250 original paper [6] divides the P+P representation into two cases depending on whether $T_z = 0$, but we combine these two 303
251 cases into one equation shown in Equation 2 by simple algebraic manipulations. 304
252253 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: 305
254

255
$$\mathbf{p} = \mathbf{p}_w + \frac{t_{[3]}\mathbf{p}_w - K\mathbf{t}}{D_{pp}(\mathbf{p})} \quad (4)$$
 306
256

257
$$D_{pp}(\mathbf{p}) = \frac{\|t_{[3]}\mathbf{p}_w - K\mathbf{t}\|_2}{\|\mathbf{p} - \mathbf{p}_w\|_2}, \quad (5)$$
 307
258

259
$$\text{where } \mathbf{p}_w = \frac{A'\mathbf{p}'}{\mathbf{a}_3'^T \mathbf{p}'} \text{ and } A' = K R K^{-1}$$
 312
260

261 We use P+P representation to estimate initial depth because we found it more efficient and robust for dense depth estimation 315
262 compared with standard triangulation methods, which are usually used with sparse correspondences. Equation 4 can also be 316
263 extended to multiple frames with importance weights by formulating it as a weighted least square problem. 317
264265 **2.2. Confidence** 318
266267 Recall the confidence value at each pixel \mathbf{p} in the non-human (environment) regions \mathcal{E} of the image is defined as: 320
268

269
$$C(\mathbf{p}) = C_{lr}(\mathbf{p})C_{ep}(\mathbf{p})C_{pa}(\mathbf{p}) \quad (5)$$
 322
270

324 C_{lr} is a confidence based on left-right consistency between the estimated forward and backward flow fields. That is, 378
 325 $C_{lr}(\mathbf{p}) = \max(0, 1 - r(\mathbf{p})^2)$, where $r(\mathbf{p}) = \|F(\mathbf{p}) + B(\mathbf{p}')\|_2$ is the forward-backward optical flow warping error. 379
 326 C_{ep} gives low confidence to pixels where the flow field and the epipolar constraint disagree [3]. Specifically, $C_{ep}(\mathbf{p}) =$ 380
 327 $\max(0, 1 - (\gamma(\mathbf{p})/\bar{\gamma})^2)$, where geometric epipolar distance $\gamma(\mathbf{p})$ is defined as: 381
 328
 329

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

333 where $\mathbf{F} = K^{-T}[\mathbf{t}] \times R K^{-1}$ is the fundamental matrix and $(\mathbf{F} \mathbf{p})_{[i]}$ is the i^{th} element of $\mathbf{F} \mathbf{p}$. 387
 334
 335 $C_{pa}(\mathbf{p})$ 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 $\beta(\mathbf{p}) = \cos^{-1} \left(\frac{\mathbf{b}(\mathbf{p}) \mathbf{b}(\mathbf{p}')}{\|\mathbf{b}(\mathbf{p})\|_2 \|\mathbf{b}(\mathbf{p}')\|_2} \right)$, 389
 336 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 . 390
 337
 338

2.3. Losses

339 Our loss is computed on log-space depth values and consists of three terms (Section 4.3 in the paper): 393
 340

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

343 **Scale-invariant MSE.** \mathcal{L}_{MSE} denotes the scale-invariant mean square error (MSE) adopted from [2]. This loss term computes 397
 344 the squared, log-space difference in depth between two pixels in the prediction and the same two pixels in the ground-truth, 398
 345 averaged over all pairs of valid pixels. Intuitively, it penalizes differences in the ratio of depth between two predicted depth 399
 346 values relative to the same ratio in the ground truth: 400
 347
 348

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

352 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. 406
 353
 354 **Multi-scale gradient term.** We use a multi-scale gradient term to encourage smoother gradient changes and sharper depth 407
 355 discontinuities in the predicted depth images [10]: 408
 356
 357

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

360 where subscript s of R_s and I_s indicates scale s and N_s is the number of valid pixel at scale s . 414
 361
 362 **Multi-scale, edge-aware smoothness terms.** To encourage smooth interpolation of depth in texture-less regions where MVS 415
 363 fails to recover depth, we add a simple smoothness term at multiple scales based on the first- and second-order derivatives of 416
 364 images [12]: 417
 365
 366

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

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

372 We create $S = 5$ scale image pyramids using nearest-neighbor down-sampling for both multi-scale gradient and smoothness 426
 373 terms. 427
 374
 375
 376
 377

432	2.4. Error Metrics	486
433	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 <i>linear</i> in the number of pixels.	487
434	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.	488
435	Specifically, si-full measures si-RMSE between all pairs of pixels, giving the overall accuracy across the entire image and is defined as follows:	489
436		490
437	$\begin{aligned} \text{si-full} &= \frac{1}{N^2} \sum_{\mathbf{p} \in I} \sum_{\mathbf{q} \in I} \left((\log \hat{D}(\mathbf{p}) - \hat{D}(\mathbf{q})) - (\log D_{\text{gt}}(\mathbf{p}) - D_{\text{gt}}(\mathbf{q})) \right)^2 \end{aligned} \quad (12)$	491
438		492
439		493
440		494
441		495
442	$= \frac{1}{N^2} \sum_{\mathbf{p} \in I} \sum_{\mathbf{q} \in I} (R(\mathbf{p}) - R(\mathbf{q}))^2$	496
443		497
444		498
445	$= \frac{1}{N^2} \sum_{\mathbf{p} \in I} \sum_{\mathbf{q} \in I} R(\mathbf{p})^2 + R(\mathbf{q})^2 - 2R(\mathbf{p})R(\mathbf{q})$	499
446		500
447		501
448		502
449		503
450	$= \frac{1}{N^2} \left(N \sum_{\mathbf{p} \in I} R(\mathbf{p})^2 + N \sum_{\mathbf{q} \in I} R(\mathbf{q})^2 - 2 \sum_{\mathbf{p} \in I} R(\mathbf{p}) \sum_{\mathbf{q} \in I} R(\mathbf{q}) \right)$	504
451		505
452		506
453		507
454	$= \frac{2}{N^2} \left(N \sum_{\mathbf{p} \in I} R(\mathbf{p})^2 - \sum_{\mathbf{p} \in I} R(\mathbf{p}) \sum_{\mathbf{q} \in I} R(\mathbf{q}) \right)$	508
455		509
456		510
457	si-env measures pairs of pixels in non-human regions \mathcal{E} , giving accuracy of the environment, and is defined as:	511
458		512
459	$\text{si-env} = \frac{1}{N_e^2} \sum_{\mathbf{p} \in \mathcal{E}} \sum_{\mathbf{q} \in \mathcal{E}} \left((\log \hat{D}(\mathbf{p}) - \hat{D}(\mathbf{q})) - (\log D_{\text{gt}}(\mathbf{p}) - D_{\text{gt}}(\mathbf{q})) \right)^2$	513
460		514
461		515
462	$= \frac{1}{N_e^2} \sum_{\mathbf{p} \in \mathcal{E}} \sum_{\mathbf{q} \in \mathcal{E}} (R(\mathbf{p}) - R(\mathbf{q}))^2$	516
463		517
464		518
465	$= \frac{2}{N_e^2} \left(N_e \sum_{\mathbf{p} \in \mathcal{E}} R(\mathbf{p})^2 - \sum_{\mathbf{p} \in \mathcal{E}} R(\mathbf{p}) \sum_{\mathbf{q} \in \mathcal{E}} R(\mathbf{q}) \right)$	519
466		520
467		521
468	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 :	522
469		523
470		524
471	$\text{si-hum} = \frac{1}{NN_h} \sum_{\mathbf{p} \in \mathcal{H}} \sum_{\mathbf{q} \in I} \left((\log \hat{D}(\mathbf{p}) - \hat{D}(\mathbf{q})) - (\log D_{\text{gt}}(\mathbf{p}) - D_{\text{gt}}(\mathbf{q})) \right)^2$	525
472		526
473		527
474	$= \frac{1}{NN_h} \sum_{\mathbf{p} \in \mathcal{H}} \sum_{\mathbf{q} \in I} (R(\mathbf{p}) - R(\mathbf{q}))^2$	528
475		529
476	$= \frac{1}{NN_h} \sum_{\mathbf{p} \in \mathcal{H}} \sum_{\mathbf{q} \in I} R(\mathbf{p})^2 + R(\mathbf{q})^2 - 2R(\mathbf{p})R(\mathbf{q})$	530
477		531
478		532
479	$= \frac{1}{NN_h} \left(N \sum_{\mathbf{p} \in \mathcal{H}} R(\mathbf{p})^2 + N_h \sum_{\mathbf{q} \in I} R(\mathbf{q})^2 - 2 \sum_{\mathbf{p} \in \mathcal{H}} R(\mathbf{p}) \sum_{\mathbf{q} \in I} R(\mathbf{q}) \right)$	533
480		534
481		535
482		536
483		537
484	Furthermore, si-hum can further be divided into two error measures: si-intra measures si-RMSE within \mathcal{H} , or human	538
485		539

540 accuracy independent of the environment, and is defined as 594
 541
 542

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

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

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

551 **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: 605
 552
 553

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

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

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

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

$$(32)$$

567 3. Implementation Details

568 We use FlowNet2.0 [5] to estimate optical flow because we found it handles large displacements well and preserves sharp 622 motion discontinuities. We use Mask-RCNN [4] to generate human masks and optionally human keypoints. The predicted 623 masks sometimes have errors and miss small parts of people, so we apply a morphological dilation operation to the binary 624 human masks to ensure that the masks are conservative and include all the human regions. We normalize human keypoints 625 between 0 and 1 before we feed them into network, if needed. 626

573 Our networks architecture is similar to that of [1] except that we replace all the nearest neighbor upsampling layers with 627 bilinear upsampling layers since we found such simple modification could produce sharper depth boundaries while slightly 628 improve performance. We refer readers to [1] for full details of network architectures. 629

574 Our network predicts log depth in both training and inference stages. During training, we randomly normalize the input 630 log-depth before feeding it to the network by subtracting a value sampled from between the 40 and 60 percentile of valid 631 input log D_{pp} . During inference, we normalize input log-depth by subtracting the median of log(D_{pp}). Additionally, during 632 training, we randomly set to zero the initial input depth and confidence (with probability 0.1) to tackle the potential situation 633 where input depth is not available (e.g. camera is nearly static or estimated optical flow is completely incorrect) in inference 634 stage. When we input human keypoints into network, we also use the depth from motion parallax D_{pp} with high confidence 635 ($C_{lr} > 0, C_{ep} > 0$ and $C_{pa} > 0.5$) at these locations as ground truth if MVS depth D_{MVS} is not available. 636

575 For our experiments we train our networks for 20 epochs from scratch using the Adam [8] optimizer with initial learning 637 rate of 0.0004 and we halve the learning rate every 8 epochs. During training, we firstly downsample all images to a resolution 638 of 532x299, use a mini-batch size of 16, and perform data augmentation though random flips and central crops so that input 639 image resolution to the networks is 512x288. We set hyperparameters in our loss terms $\alpha_1 = 0.5, \alpha_2 = 0.05$ based on 640 our validation set. For the experiments on the TUM RGBD dataset, we downsample ground truth to 512x384 and perform 641 morphological erosion with radius 2 to ground truth for all evaluations since we found depth from RGBD sensor is not well 642 aligned with image edges due to synchronization and small regions of depth captured by depth sensors are usually attributed to 643 outliers due to sensor noise. Additionally, we downsample images to 512x384 for our network, and we downsample input 644 images to provided default image resolutions for other state-of-the-art single-view and motion stereo models (since we found 645 input default resolutions always produce the best performance for other methods) and upsample their depth predictions to 646 512x384 before we measure the error metrics. 647

648	4. Human Mesh Reconstruction	702
649		703
650	We provide a qualitative comparison to a state-of-the-art parametric human model fitting approach [7] on one of our videos.	704
651	As can be seen in Figure 5, the model fitting fails to capture the complex poses of the the human. Parametric	705
652	model fitting also does not capture fine details such as clothes and hair.	706
653		707
654	References	708
655	[1] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In <i>Neural Information Processing Systems</i> , pages	709
656	730–738, 2016. 6	710
657	[2] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In <i>Neural</i>	711
658	<i>Information Processing Systems</i> , pages 2366–2374, 2014. 4	712
659	[3] R. Hartley and A. Zisserman. <i>Multiple view geometry in computer vision</i> . Cambridge university press, 2003. 4	713
660	[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In <i>Proc. Int. Conf. on Computer Vision (ICCV)</i> , 2017. 1, 6	714
661	[5] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of Optical Flow Estimation With Deep	715
662	Networks. In <i>Proc. Computer Vision and Pattern Recognition (CVPR)</i> , pages 2462–2470, 2017. 6	716
663	[6] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In <i>Proc. European Conf. on Computer Vision</i>	717
664	(ECCV), pages 17–30. Springer, 1996. 3	718
665	[7] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In <i>Computer Vision and Pattern</i>	719
666	<i>Recognition (CVPR)</i> , 2018. 7, 9	720
667	[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. <i>CoRR</i> , abs/1412.6980, 2014. 6	721
668	[9] Z. Li and N. Snavely. Learning Intrinsic Image Decomposition from Watching the World. In <i>Proc. Computer Vision and Pattern</i>	722
669	<i>Recognition (CVPR)</i> , 2018. 5	723
670	[10] Z. Li and N. Snavely. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In <i>Proc. Computer Vision and</i>	724
671	<i>Pattern Recognition (CVPR)</i> , 2018. 4	725
672	[11] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In <i>Proc.</i>	726
673	<i>European Conf. on Computer Vision (ECCV)</i> , pages 501–518, 2016. 4	727
674	[12] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In <i>Proc.</i>	728
675	<i>Computer Vision and Pattern Recognition (CVPR)</i> , June 2018. 4	729
676		730
677		731
678		732
679		733
680		734
681		735
682		736
683		737
684		738
685		739
686		740
687		741
688		742
689		743
690		744
691		745
692		746
693		747
694		748
695		749
696		750
697		751
698		752
699		753
700		754
701		755



(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.

864
865
866
867
868
869
870
871
872
873874
875
876
877
878
879
880
881882
883
884
885
886
887
888889
890
891
892
893
894
895
896897
898
899
900
901
902
903904
905906
907

908

909

910

911

912

913

914

915

916

917

918
919
920
921
922
923
924
925
926
927928
929
930
931
932
933
934935
936
937
938
939
940
941
942943
944
945
946
947
948
949
950951
952
953
954
955
956
957958
959960
961

962

963

964

965

966

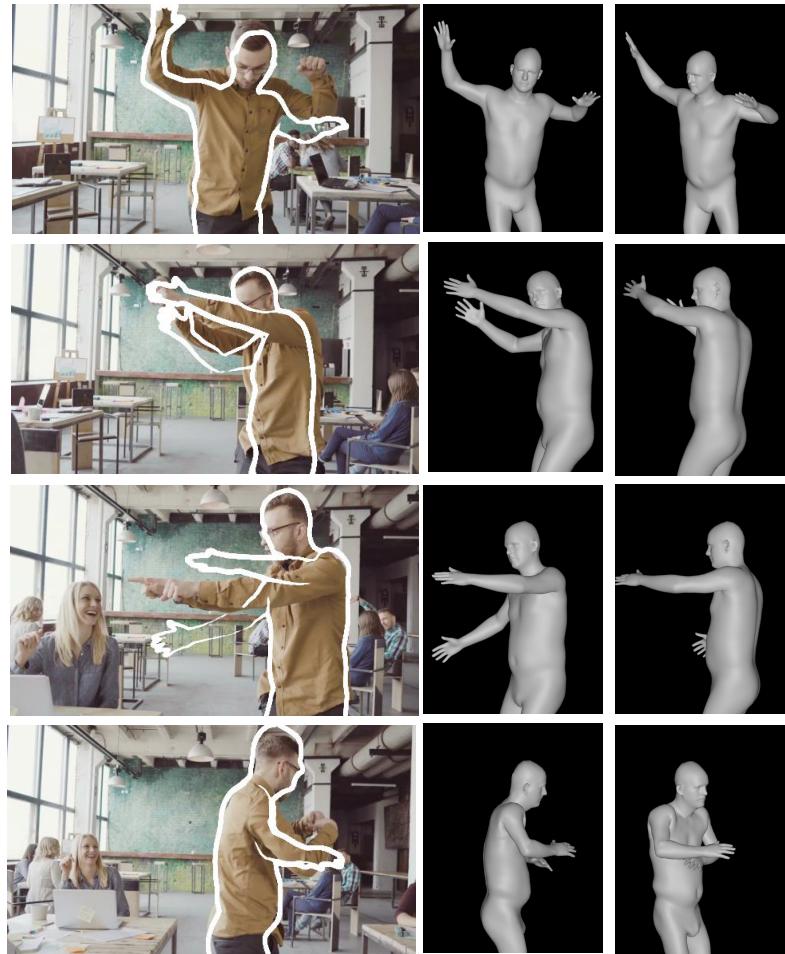
967

968

969

970

971



(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.