Creating a Large-scale Synthetic Dataset for Human Activity Recognition

Ollie Matthews
Princeton University
om3@princeton.edu

Koki Ryu
Princeton University
kokir@princeton.edu

Tarun Srivastava
Princeton University
taruns@princeton.edu

Abstract

Creating and labelling datasets of videos for use in training Human Activity Recognition models is an arduous task. In this paper, we approach this by using 3D rendering tools to generate a synthetic dataset of videos, and show that a classifier trained on these videos can generalise to real videos. We use five different augmentation techniques to generate the videos, leading to a wide variety of accurately labelled unique videos. We fine tune a pre-trained I3D model on our videos, and find that the model is able to achieve a high accuracy of 73% on the HMDB51 dataset over three classes. We also find that augmenting the HMDB training set with our dataset provides a 2% improvement in the performance of the classifier. Finally, we discuss possible extensions to the dataset, including virtual try on and modeling motion of the people.

1. Introduction

A significant challenge in Human Activity Recognition (HAR) is obtaining large amounts of video data on which deep networks can be trained. Labelling and storing video data is significantly more labour and memory intensive than images, and the largest version of the Kinetics dataset which contains 650,000 videos of crowd-sourced data over 700 classes is still two orders of magnitude smaller than ImageNet.

In this paper we attempt to see if this problem can be approached with synthetic data. The recently published MoVi dataset provides joint position data for 90 subjects doing 21 different actions. Combining this data with human body rendering packages, we are able to generate videos of people doing the actions from any angle, and superimpose them onto any background.

The aim is to use this to create a large dataset with a diverse set of angles and camera motions, and to see if models trained on this dataset are able to generalise to real videos.

For evaluation we use the I3D model, which is pre-trained on Kinetics, and test it on the HMDB51 dataset on a set of classes which do not appear in Kinetics.

In particular, our contributions are:

- We introduce a potentially infinite source of data for Human Activity Recognition. We cover three common actions - walking, waving, and sitting down. These actions were chosen for testing due to their appearance in the HMDB51 dataset, but our framework could easily be extended to more actions provided in the MoVi dataset.

- We validate our approach of using synthetic data by fine-tuning the I3D model, pretrained on Kinetics, with a combination of our synthetic data and real videos from the MoVi. We show that our fine-tuned model outperforms the same I3D model fine-tuned with only real videos taken from the HMDB training set.

- Lastly, we provide the ability to extend our framework to any real indoor scene through the use of RGB-D depth map fusion. We reconstruct a variety of indoor scenes using a pipeline based on the work done by Choi et al. and synthesise videos of human subjects performing the chosen actions. Rendering our subjects in such 3D environments was shown to further improve the performance of our framework.

2. Related Work

The use of a synthetic dataset is actively being researched by the deep learning community. Peng et al. crowd-source CAD models online to augment datasets for object detection algorithms. Gaidon et al. have created a Virtual KITTI dataset that generates synthetic videos of cars to study object detection and tracking. There has not, however, been much research into the use of synthetic data for Human Activity Recognition. The only widely available dataset for HAR is SURREAL by Varol et al.
Figure 1: Examples images taken from the SURREAL dataset.

Figure 1 shows example images from their synthetic dataset, found in their paper. These images are far from realistic, and do not make geometrical sense. People are often in impossible positions, as seen in the first image, or out of proportion like in the second image. By generating a 3D model of our subjects within the background scene, we can ensure the results are not geometrically impossible, and the videos produced from these 3D renderings can include realistic effects like shadows and occlusion.

3. Datasets

We make use of a number of datasets for generation and evaluation of our videos. The videos of the generated people are based on MoVi, an opensource dataset of Inertial Measurement Unit (IMU), motion capture and video data for actors performing a range of activities. The 60 female and 30 male actors are of a range of ages, and perform 20 prescribed actions as well as one individually chosen action.

We generate realistic 3D human body meshes based on the captured body poses from MoVi with: SMPL+H [24], which is a parameterized human body model with hand motions; and Dyna [23], which generates soft-tissue deformations on the body model. The 3D points of MoVi converted into the body parameters of SMPL+H and Dyna are stored in the AMASS[18] dataset. We use these parameters to generate human action videos.

As a part of this project, we also reconstruct indoor 3D scenes to use as backgrounds for our synthetic dataset. The underlying idea being that any user can extend this synthetic dataset to their specific purpose by reconstructing their own environment and synthesising more data. As a proof of concept, in this project we reconstructed open-source dataset with readily available RGB-D streams. We reconstructed scenes from the Redwoood Indoor LiDAR dataset [20], the ICL-NUIM dataset [12], the BundleFusion dataset [6], and lastly the TUM RGB-D SLAM dataset [25].

Figure 2: An frame from a challenging video in the HMDB dataset where the subject is not in the center.

We evaluate our results on the HMDB51 dataset, which contains 6,849 clips of 51 different actions mostly extracted from films.

3.1. Challenges in the HMDB Dataset

The HMDB dataset is notoriously challenging, with only 70 videos for training. The videos also come from films, and the person doing the action is often not central in the video.

Figure 2 shows an example of a problem that can occur because of this. The image is a frame taken from one of the ‘hand waving’ videos in HMDB. The frame has a wide aspect ratio, with the man waving only appearing at the right of the screen. The classifier we use requires inputs to have equal aspect ratios, so at training, we have to crop out sections of the frame to be fed into the classifier. For many of the crops, the label will be ‘hand waving’ even though the man waving is not in shot. We deal with this at test time by convolving our classifier across the image and averaging the results, but these videos can harm the performance of the network if they come up during training.

The HMDB dataset also includes videos where the camera is moving during the shot, and others where the subject is occluded. These problems make the dataset hard to train on, and we aim to deal with them in generating our synthetic dataset. The problem of the subject not being in the shot, for example, does not occur when we create the videos ourselves and choose where the subject is placed.

4. Approach

4.1. Rendering

We render realistic human action videos based on the body parameters from AMASS with Trimesh [9] and Pyrender [19]. We create Trimesh objects corresponding to the
human body and the background and add them to a 3D scene model in Pyrender. Pyrender exports the scene as MP4 video.

We convert the body parameters to human body mesh objects with VPoser [21]. We create the background in two different ways. In the first, we create a flat wall as a Trimesh object and add a background image as the texture. In the second, we convert the OBJ file of the living room in ICL-NUIM to a GLB file with a free converter [1]. The GLB format is readable by Trimesh, and can then be imported as a 3D object.

We use a Pyrender scene set up by VPoser to render videos. We modify the positions of cameras and lights in the scene and add the background and human model as objects to it. To make the rendered video realistic, we fine-tune the size and positions of the objects manually.

4.2. Data Augmentation

We render videos based on the actions of 15 subjects in MoVi dataset. For each subject, we choose three actions, ’walking’, ’sitting down’ and ’hand waving’, and generate 10 videos per action. We try different methods to add variation to the multiple videos made from a single pose.

4.2.1 Background Images + Rotation (BG+R)

For this and the next method, we use six images from Google Images to provide the backgrounds. These show three inside and three outside scenes, and are used as texture for the back wall. The human objects are generated with a random rotation from -90° to 90°. We then add the human model in the center of the scene and the wall behind the human, and the scene is rendered as a video. Examples are shown in Figures 3 and 4.

4.2.2 Background Images + Rotation, Resizing and Rotation (BG+R²T)

In 4.2.1 the human model is centered in the generated videos. Since the subject is not necessarily in the center of the shot in a real video, we apply scaling and translation to the human poses to generate another set of videos. For each action, we choose random variables

\[ s \in [0.7, 1.3], \quad x \in [-0.5, 0.5], \quad y \in [-0.1, 0.1] \]

If we denote by \( h \) the height of the human body in the video, for each scene we change the coordinates of the body by \( (h \cdot x, h \cdot y) \), then scale the body by \( s \). Examples are shown in Figures 5 and 6.

4.2.3 3D Model + Rotation (3D+R)

In all of the following methods, we use a 3D object as the background of the generated videos. We create a human pose 3D mesh with the same method as the one used in 4.2.1, then place this human body in a fixed point of the living room scene from the ICL-NUIM. To add some variation to the background, we pick a random color for the background from 13 colors: ’pink’, ’purple’, ’cyan’, ’red’, ’green’, ’yellow’, ’brown’, ’blue’, ’offwhite’, ’white’, ’orange’, ’grey’, and ’yellow’.

As described in 4.2.1 we also pick a random angle from the range of -90° to 90°. Instead of rotating the human body by this angle, we rotate the camera and the light around the body by the angle so that each rendered video shows the different part of the living room.

Finally, we change the position of the light and the angle of view of the camera to make the human pose look clear in the closed object. Figures 7 and 8 show some frames from these.

4.2.4 3D Model + Camera Motion (3D+M)

Having a full 3D model allows us to add camera motion to the video. We choose random variables \( x_1 \) and \( x_2 \) from the
range of -0.5 to 0.5, and choose \(y_1\) and \(y_2\) from the range of -0.1 to 0.1. We move the camera and the light in the video so that the coordinates of the body is \((x_1 \cdot h, y_1 \cdot h)\) at the beginning of the video and \((x_2 \cdot h, y_2 \cdot h)\) at the end.

We also choose two angles, \(\theta_1\) and \(\theta_2\), from the range of -90° and 90° and rotate the camera and the light from \(\theta_1\) to \(\theta_2\). Figure 9 shows a sequence of frames from one of these videos, where the camera motion can be seen.

4.2.5 Reconstructed 3D Model + Rotation (R3D+R)

In this final method, we use 3D reconstructions of scenes to lead to more realistic images, and give users the ability to cater the synthetic dataset to their specific purpose. Given that there are several commercially available structured light sensors available, users could reconstruct their own scenes for use in the synthetic dataset. A good example would be human activity recognition in sports - if a user wished to identify human actions in a school basketball court, they could reconstruct the scene and generate videos for that purpose.

We use the Python implementation of the Open3D library to reconstruct scenes from a sequence of RGB-D frames. In real life, users will not have precise camera pose data, so our 3D scene reconstructions were performed without ground truth pose data. We begin by aligning the RGBD frames by using the in-built 'compute RGB-D odometry' function that estimates 6D rigid body motion between two frames. The input to this function is a rough estimate of the alignment provided by a 5 point RANSAC algorithm. With these estimates, we perform pose optimisation which results in fragments of the scene. A global posegraph is then computed after performing multiple registration on the scene fragments. Multiple registration is performed again to refine the posegraph, and lastly these fragments are integrated, resulting in a .ply file.

The human can then be placed into the .ply file in the same way as in Section 4.2.4. Figure 10 shows two of the generated scenes, and Figures 11 and 12 show frames from the final videos.

4.3. Optical Flow

To calculate optical flow, we use the TV-L1 algorithm. This is also used in the I3D model’s original paper as it provides a dense and accurate output. We use the OpenCV implementation, and Figure 13 shows an example of an output for a 3D+M video.

5. Dataset Evaluation

To evaluate our dataset, we fine-tune a pre-trained video classifier on our videos and test it on the HMDB dataset. The classifier used is the I3D classifier, pre-trained on the original version of Kinetics which contains at least 400 videos per class for 400 classes.
5.1. Classes

We evaluate over classes that fit two criteria: they appeared in both MoVi and HMDB, and did not appear in Kinetics. The second condition ensures that our pre-trained model had no advantage in predicting certain activities, and restricts us to three classes - ‘walking’, ‘sitting down’ and ‘hand waving’.

5.2. Model

The I3D model utilises a two-stream 3D Convolutional Neural Network (CNN) approach based on Inception-v1 [13]. Before being trained on Kinetics, the classifier is bootstrapped from 2D CNNs trained on ImageNet.

We use a publicly available version of the I3D model [7] with the final layer removed to extract 1024-dimensional features from RGB and optical flow videos. The model is fully convolutional, and provides around 4 feature vectors per second of video.

For both the RGB and optical flow features, we connect a fully connected layer to the features to a 3-dimensional output with a softmax applied. A dropout rate of 0.5 is applied to the features. The class probabilities from both networks are then summed to provide combined probabilities. For testing, we also average the probabilities over the temporal dimension of the video.

5.3. Data Pre-Processing

Before being fed into the I3D model, the videos need to be converted to RGB and optical flow arrays. The I3D model takes videos at 25 frames per second and $224 \times 224$ pixels. While our videos can be generated under these constraints, the HMDB videos need to be resampled to 25 fps and reshaped. The HMDB videos do not have equal aspect ratios, so we resize them - keeping the aspect ratio constant - so that the vertical dimension is 224 pixels, and then extract an equally spaced set of frames spanning the width of the videos.

The RGB and flow arrays are also centered and normalised. As suggested in the original I3D paper, the optical flows are truncated at ±20 prior to this.

5.4. Training

To train the fully connected layer, we use an Adam optimiser [15] with a learning rate of $10^{-4}$ for 300 epochs. Each video has multiple feature vectors over the time dimension, and we train the fully-connected layer individually on each of these with a categorical cross-entropy loss function.

6. Results

We use the first suggested split of the HMDB dataset for evaluation, which gives 70 training videos and 30 testing videos for each class. We then evaluate the test accuracies for the RGB network, the optical flow network, and the two networks combined.

6.1. Ablation Study 1: Comparison of Augmentation Methods

Table 1 show the accuracy achieved when training the classifier on the subset of videos from each augmentation technique listed in Section 4.2.

| Augmentation Method | RGB | Optical Flow | Combined |
|---------------------|-----|--------------|----------|
| BG+R                | 0.58| 0.60         | 0.67     |
| BG+R ‘T’            | 0.53| 0.68         | 0.68     |
| 3D+R                | 0.58| 0.70         | 0.66     |
| 3D+M                | **0.59**| 0.56   | 0.67     |
| R3D+R               | **0.59**| 0.70   | **0.69** |

Table 1: Test accuracies for model trained on individual augmentation methods.

The simplest background with rotation method performs well in RGB and combined, but underperforms in optical flow, indicating that it is important to have different sizes.
of models for the optical flow network to learn to be size-invariant. Varying the size and location of the subjects helps to deal with this, but the RGB accuracy decreases.

Using the living room model as a 3D scene leads to better accuracy in the individual networks. This can be explained by the fact that in this augmentation method the humans are not necessarily centered, which helps enforce size and location invariance in the network, but the range of sizes is not as large as in BG+R^2T which could explain the better accuracy in RGB and optical flow. Since the videos are rendered from a full 3D model, they are also more realistic geometrically. Adding camera motion improves RGB performance, but dramatically decreases the performance of the optical flow network. This could be due to the camera motion being too fast, which causes non linear behaviours if the optical flow values truncation at the limits described in Section 5.3. It could also be because it trains the classifier to always expect camera motion while in reality this is only the case in some videos.

The best performance overall is from using the reproduced 3D scenes. These scenes have realistic backgrounds, and the proportions and shadows are geometrically correct. As seen in the feet of the figure in Figure 8, these videos also can have occlusion, which is not present in any of the other videos and often occurs in HMDB51.

### 6.2. Full Dataset Evaluation

|          | RGB Weight | Flow Weight |
|----------|------------|-------------|
| BG+R     | 1          | 0           |
| BG+R^2T  | 0          | 1           |
| 3D+R     | 1          | 1           |
| 3D+M     | 1          | 1           |
| R3D+R    | 1          | 1           |
| HMDB51   | 8          | 3           |

Table 2: The weights applied to each set of videos in COMBW (note HMDB51 is only included in COMBW+HMDB).

Table 3 shows the results from evaluating the model on the combined augmentation methods. In COMB, all of the augmentation methods are given equal weight. In COMBW, they are weighted differently, with weights shown in Table 2 applied to each set of videos. These weights were informed by the results of Section 6.1 with augmentation methods which lead to poor classifiers being left out. Note that the camera motion is included in training the optical flow network even though it leads to poor performance in our ablation study. It turns out there was a 1% reduction in accuracy when it was omitted, which implies that while not all of the training videos should have camera motion, it is an important factor to include.

HMDB shows the results from training on HMDB alone, and COMBW+HMDB shows the results of training on HMDB combined with COMBW. We weight HMDB more heavily than the other sets, in particular on the RGB training, since it contains less samples and the classifier performance is so much higher when trained on it. However we do not pretend the weights chosen are optimal - they were found through trial and error, and could potentially be improved upon in future work.

|          | RGB   | Optical Flow | Combined |
|----------|-------|--------------|----------|
| COMB     | 0.54  | 0.74         | 0.70     |
| COMBW    | 0.59  | 0.75         | 0.73     |
| HMDB     | 0.78  | 0.77         | 0.81     |
| COMBW+HMDB | 0.78 | 0.82         | 0.83     |

Table 3: Test accuracies for model trained on combined datasets.

Using the entire dataset leads to significant improvements in performance compared to only using certain parts. This shows that the different augmentation methods work well in conjunction, for example while using an entire dataset with camera motion is not effective, including this with the other augmentation techniques makes for a more robust classifier.

The RGB performance on COMB is poor, however, and it seems that the BG+R^2T set has a significant detrimental effect on the performance of the RGB network as removing it in COMBW dramatically improves performance. The flow network also sees moderate improvements when the BG+R set is removed and more emphasis is put on the R3D+R scenes. The final accuracy of the COMBW classifier of 0.73 is very impressive seeing as the classifier has never seen any of the videos in HMDB before.

Training on HMDB leads to significant improvements in particular in the RGB network, which is not entirely surprising. While we try to make the RGB videos as realistic as possible, they are still based on reconstructions and animations. The optical flow, on the other hand, can largely be reproduced in the synthetic videos regardless of how realistic the actual videos are.

Using a combination of our dataset and HMDB improves the optical flow network performance by 5%, and the combined performance of the classifier by 2%. This shows that our data is realistic enough, at least in terms of optical flow,
to bolster smaller datasets and improve performance by increasing the training set size.

Figure 14 shows the combined performance of the I3D model trained on each of the datasets, for each class.

The main source of confusion in the classifiers is between 'walking' and 'hand waving'. Since these both mostly happen from an upright position, they look more similar than 'sitting down', on which the accuracies are close to 1.0. The generated videos seem to struggle in particular with walking, which we attribute to two problems. The first is that walking shots often have close ups on the upper body of the person, and our generated videos do not have this. The second is that all of our models are moving on a constant point in their scenes, since there was no readily available position information in the dataset. This means that images like that shown in Figure 15 where there is a close up and all the motion is in the background of the scene are difficult for the model trained on our dataset. In contrast, the model performs well on walking videos where the entire body is in the frame, as seen in Figure 16.

The I3D network trained on the generated videos has higher accuracy on hand-waving than that trained on HMDB. This could be in part because the classifier is more biased towards hand waving rather than the classifier being better, but using the generated videos to augment the HMDB training set does improve performance on hand waving, with only a slight fall in walking accuracy.

Figures 17 and 18 show two more examples of predictions from the classifier trained on our dataset. The classifier is able to accurately recognise the person sitting in Figure 17 even though they are occluded by the desk, but struggles with the person who is waving in Figure 18 where only the upper part of their body is shown, and they look like they could be sitting down.

6.3. Ablation Study 2

To justify the complexity of generating the synthetic, we try training the I3D model on the human body models without any background (HBM-B), and on the raw videos of the subjects performing the activities from the MoVi dataset (RV). The results are shown in Table 4.

|         | RGB | Optical Flow | Combined |
|---------|-----|--------------|----------|
| HBM-B   | 0.51| 0.67         | 0.65     |
| RV      | 0.58| 0.66         | 0.68     |

Table 4: Test accuracies for model trained on ablation datasets.

The results indicate that including a background to the scene is important for RGB classification, but has less of an effect on the optical flow network. This is unsurprising as the stationary backgrounds should not have an effect on the optical flow output. The optical flow on our synthetic dataset is still higher, which shows that the occlusion and camera motion effects in the video generation are important for performance.

Using the training videos achieves high RGB performance and optical flow performance, but not as high as our generated videos. While the videos are more realistic, they do not offer the same range of viewing angles as the generated videos, and do not include other augmentation techniques like occlusion and camera motion.
7. Conclusion

In this paper, we have shown that it is possible to train a high performing classifier with a purely synthetic dataset. Moreover, we have shown that synthetic data can be used alongside real data to improve on performance, with particular improvements in optical flow processing.

These results serve as a proof of concept, implying that synthetic data is a valuable tool in human activity recognition. By being able to generate infinite permutations of humans in environments, we are able to provide a rich dataset for training even if the videos are not completely realistic. We believe that with more sophisticated video augmentation techniques, the improvements could be even more dramatic. In particular, more realistic videos could allow for improvements in the RGB stream as well as the optical flow stream.

8. Future Work

We identify some key areas in which our video generation could be improved.

More variation in the videos. We found that many of the videos in the test set had close-ups on the upper body of people. Including these into our data would be easy, and could improve performance. We could also include more occlusion by actively placing objects in the 3D scenes between the camera and the human.

Adding clothing to the subjects. An obvious technique to make our subjects more realistic would be to generate them with clothes. The work done by Lassner et al. [17] achieves exactly this aim. They have built a deep learning generative model for people in clothing. They train models on a large image base build on top of the Chictopia 10K dataset. During our initial attempts at building our dataset, we ran into issues with their current codebase and had to opt for SMPL based models instead. Using their generative model could be a promising step towards more realistic videos.

Implementing movement. The humans we generated are all fixed on one spot in their scenes, not moving during the video. This was done because the movement information is not directly provided in the MoVi dataset, but if movement were inferred from the IMU sensors and videos, our generated people could be made to walk across the scenes instead of walking on the spot. This would make for more realistic optical flow as well as RGB frames.

Automating human placement. Placing the humans in the 3D scenes was a very time consuming job as the rotation and translation parameters needed to be tuned one by one. This limited the number of scenes we could recreate. The process could be made semi-autonomous by implementing a GUI where a user could rotate the scene and place the human in acceptable locations. The possible human placement locations could also be automatically inferred from surface normal and geometric information in the scenes. A better improvement still would be to be able to automatically locate ‘suitable’ objects in the environment so that the people do not sit in mid air.

Generating videos at train time. Since the videos were time consuming to generate, we generated them all before training and trained on each video multiple times. One potential advantage of a synthetic dataset is that the videos could be generated before each training epoch, to discourage the model from overfitting by never showing it the same video twice.

As well as improving the dataset, we could see how well our dataset works for other classifiers, and try to improve on state-of-the-art performance by augmenting with our videos.

References

[1] AnyConv. Anyconv file converter. https://anyconv.com/ Online; accessed 7 May 2020.
[2] G. Bradski. The OpenCV Library. Dr. Dobb’s Journal of Software Tools, 2000.
[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. arXiv preprint arXiv:1907.06987, 2019.
[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6299–6308, 2017.
[5] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5556–5565, 2015.
[6] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. ACM Transactions on Graphics 2017 (TOG), 2017.
[7] deepmind. kinetics-i3d. https://github.com/deepmind/kinetics-i3d 2019.
[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.
[9] Dawson-Haggerty et al. trimesh.
[10] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. CoRR, abs/1605.06457, 2016.

[11] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F Troje. Movi: A large multipurpose motion and video dataset. arXiv preprint arXiv:2003.01888, 2020.

[12] A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In IEEE Intl. Conf. on Robotics and Automation, ICRA, Hong Kong, China, May 2014.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In 2011 International Conference on Computer Vision, pages 2556–2563, 2011.

[17] Christoph Lassner, Gerard Pons-Moll, and Peter V. Gehler. A generative model for people in clothing. In Proceedings of the IEEE International Conference on Computer Vision, 2017.

[18] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In The IEEE International Conference on Computer Vision (ICCV), Oct 2019.

[19] Matthew Matl. Pyrender.

[20] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In ICCV, 2017.

[21] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2019.

[22] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Exploring invariances in deep convolutional neural networks using synthetic images. CoRR, abs/1412.7122, 2014.

[23] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. Dyna: A model of dynamic human shape in motion. ACM Transactions on Graphics, (Proc. SIGGRAPH), 34(4):120:1–120:14, Aug. 2015.

[24] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia), 36(6), Nov. 2017.

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremer. A benchmark for the evaluation of rgb-d slam systems. In Proc. of the International Conference on Intelligent Robot Systems (IROS), Oct. 2012.

[26] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. CoRR, abs/1701.01370, 2017.

[27] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. arXiv:1801.09847, 2018.