Learning Dense Correspondence from Synthetic Environments

Abstract—Estimation of human shape and pose from a single image is a challenging task. It is an even more difficult problem to map the identified human shape onto a 3D human model. Existing methods map manually labelled human pixels in real 2D images onto the 3D surface, which is prone to human error, and the sparsity of available annotated data often leads to sub-optimal results. We propose to solve the problem of data scarcity by training 2D-3D human mapping algorithms using automatically generated synthetic data for which exact and dense 2D-3D correspondence is known. Such a learning strategy using synthetic environments has a high generalisation potential towards real-world data. Using different camera parameter variations, background and lighting settings, we created precise ground truth data that constitutes a wider distribution. We evaluate the performance of models trained on synthetic using the Common Objects In Context (COCO) dataset and validation framework. Results show that training 2D-3D mapping network models on synthetic data is a viable alternative to using real data.

Index Terms—Synthetic Data Generation, Skinned Multi-Person Linear (SMPL) Model, Densepose, Semantic Segmentation, Human Modelling.

I. INTRODUCTION

Estimating human shape and pose from a single 2D image and establishing correspondence with a 3D human surface is a hard problem. Challenges from the lack of 3D information in 2D images range from depth ambiguity to occlusion or imaging conditions (e.g. lighting) as well as the difficulty and tediousness of creating (sparse) ground truth data and overall natural variability in human pose and shape. It is a primary task for many applications including virtual reality, human action recognition, augmented reality and human-robot interaction [1], [2]. 2D-3D human mapping involves other difficult problems including object detection, pose estimation and semantic segmentation as auxiliary tasks. Patel et al. [3] introduced a synthetic human dataset for regression analysis containing 3D human scans placed in scenes that performed well on fine tuning the SMPL optimization IN the loop (SPIN) [4] model that fits a human shape model to humans in 2D images. Another synthetic dataset that improves pedestrian detection and tracking was introduced in [5]. Thus, the adaptability and reliability of synthetic data over real data are increasing over time.

There is a substantial body of work trying to solve the problem of mapping a 2D image to a 3D model with dense point correspondence for inanimate objects [6]–[8]. For humans, 3D representations, parametric surface models such as SMPL [9] or Adam model [10] are often used to create controllable 3D surface deformations. Pixel aligned implicit functions have been used for high resolution clothed human digitization by transferring the human geometry onto a 3D model based on a single image. The method is non-parametric, which decreases the bias towards training data [11], however, when complex motions are involved, parametric models such as SMPL can be constrained more easily and provide temporally consistent and realistic animations.

Creating a synthetic dataset is a tedious task that requires extensive designer work and advanced procedures to make it realistic. Available datasets are composed of people doing pre-defined actions, and obtaining a more diverse set of actions still requires significant human effort to create those scenarios [12]. Human avatars can be created in a virtual environment and made to perform procedural actions such that all these can be properly recorded and annotated for various recognition tasks. Data can be generated from actions grounded in motion capture data or from procedurally defined synthetic actions. Probabilistic models can be used to generate action videos based on various parameters including human body shape, action category, basic motions, variations of motions, camera setup, environment and many more [13].

Generative adversarial networks (GANs) can be used for creating artificial data. They were introduced by Goodfellow et al. in 2014 [14]. They combined the concepts of game theory with statistical modelling and came up with a novel approach for generative modelling. GANs are a framework in which two neural networks compete with each other to improve the generation of some distribution. They are constituted of two components, the generator and the discriminator. The generator tries to trick the discriminator by creating better samples that are very similar to the real distribution while the discriminator tries to identify whether the generated image is real or fake. It is thus a trade-off between the generator and the discriminator elements which improves the generation process [15]. The discriminator is trained to maximise the probability of assigning correct label to both training examples and samples from the generator. Thus, it is a two player minimax game between the generator and the discriminator with value function $V(G, D)$ [14]:

$$V(G, D) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log (1 - D(G(z)))]$$

Minimising $V(G, D)$ is the same as maximising the probability of assigning the correct label to both training and generated samples,
treating the problem of dense correspondence. They manually
on to the 3D human model which was a different way of
by humans. They used an inverse mapping of image pixels
in [21] which consists of normal human 2D images annotated
achieve promising results in obtaining accurate correspon-
surface coordinates was proposed in [21] where Mask-RCNN
painting network is used to interpolate a dense supervision
structural complexity and pose variability of human body. A
work is based on humans, there were challenges due to higher
images "in the wild". This work was focused on establishing
to establish dense correspondence between a 3D model and
realistic data points.

Style guided transformations [19] can be performed within
to guide this transformation to create reliable synthetic data.
The binary masks of the object segmentation could be used as a supervision
to transform it to a more realistic image [18]. The binary masks
when both the generator and the discriminator were conditioned on some extra information y. This extra component can be any form of auxiliary information like class labels or even data from other modalities. The conditioning is performed by feeding y into both the discriminator and generator as an additional input layer. Thus, the input to the generator and the discriminator will be concatenated with this extra condition y. Then, the objective function of the two-player minimax game will be value function $V(G, D)$ [16]:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}}[\log D(x)] + E_{z \sim p_z}[\log (1 - D(G(z)))].$$

GANs could be used to minimise the distribution gap between real data and synthetic data [17]. Image to image translation can be performed on artificially generated images to transform it to a more realistic image [18]. The binary masks of the object segmentation could be used as a supervision to guide this transformation to create reliable synthetic data. Style guided transformations [19] can be performed within the required regions of the image to create more diverse and realistic data points.

DenseReg [20] is a framework where CNNs were trained to establish dense correspondence between a 3D model and images “in the wild”. This work was focused on establishing correspondence of human face with a surface model. Since the work is based on humans, there were challenges due to higher structural complexity and pose variability of human body. A 24 part uv field regression to achieve dense correspondence between image pixels and surface points was introduced in [21]. A cascaded extension of Densepose-RCNN and an inpainting network is used to interpolate a dense supervision from sparse supervision signals.

A suitable architecture for performing regression of human surface coordinates was proposed in [21] where Mask-RCNN architecture of [22] was combined with their approach to achieve promising results in obtaining accurate correspondence field for multiple persons in a scene with a real-time inference speed. The Densepose-COCO dataset was introduced in [21] which consists of normal human 2D images annotated by humans. They used an inverse mapping of image pixels on to the 3D human model which was a different way of treating the problem of dense correspondence. They manually

identified human semantic part segmentation maps and sparse uv coordinate mapping for these regions. These uv coordinate mappings establish human correspondence between the human in the 2D image and the 3D human model [9].

A key objective of our work is to reduce the labour cost, ethics and privacy issues, and to eliminate the time sink of manual data labelling. Another advantage of using artificial data is that there are lot of tasks that are very complex for humans to perform and are prone to human error, causing inaccuracies in ground truth data. Synthetic data can be a viable alternative since exact data points can be generated through automated procedures. Over the past decade, synthetic data have emerged as a possible solution for the large amount of labelled data that are required to train deep-learning algorithms. Synthetic data could be a great alternative to address the issues previously mentioned. There are multiple ways for creating artificial data ranging from 3D modelling using rendering engines, simple data augmentation techniques or generative networks. While realism has been a point of contention with synthetic data, a large body of work has been published in relation to improving realism in order to bridge the gap between artificial data and real-world data.

Our contributions are two fold. First, we present a method to automatically generate synthetic data and extract ground truth annotations for the problem of dense 2D-3D human correspondence. The data generated consist of a video sequence of a human performing a mundane action and associated ground truth data for later use in 2D-3D mapping. The ground truth data include human semantic segmentation maps, exact 2D and 3D human pose keypoints, and ground truth 2D pixels to 3D human vertices mapping. Second, we obtained promising results for the problem of dense correspondence mapping using precise synthetic datapoints. With our generated data, we are able to qualitatively improve some areas of human segmentation using the method suggested in [21] by training it only using our synthetic data.

II. METHODOLOGY

In this section, we describe the completely automated method used to generate accurate ground truth 2D-3D mapping data of a human mesh using a synthetic 3D environment where human avatars are made to perform mundane actions based on motion capture (MOCAP) data. Also, we present the methods used to train the dense correspondence model for obtaining correspondences between humans in images and the 3D SMPL
model. The various data post-processing techniques developed in the image domain to create occlusions and improve realism of the visual data are also explained.

A. Avatar Animation and Rendering

In this research, we utilised MOCAP data in order to animate the human avatar in the scene. Part of the MOCAP data were collected at the Queensland Centre for Advanced Technologies. Ten volunteers performed a set of actions (e.g., walking, running, seating) while wearing reflective markers. The 3D position of these markers was captured using 24 synchronised infrared cameras and Qualisys Track Manager was used to reconstruct and animate 3D human skeletons. Additional publicly available MOCAP data were included to increase diversity [23].

SMPL model is used to create procedural human action videos [24]. It is based on the mean human body shape that is deformed in a realistic way using principal component analysis in order to generate large amount of statistically plausible body shapes that have resemblance with 4000 real human shapes which were used to train the model. SMPL [9] is a function \( M(\theta, \beta) \) that uses pose parameters \( \theta \in \mathbb{R}^{72} \) and shape parameters \( \beta \in \mathbb{R}^{10} \) to generate a mesh of 6890 vertices. The variations in shape parameters and pose parameters within a visually plausible range are applied randomly to generate varying distribution of human mesh data with clothing. An example of a male avatar with few shape variations are depicted in Figure 1.

Textures from Surreal dataset [25] are used to transfer textures to the mesh. The texture data is separated into male and female categories and random stitching of semantic parts of human body in texture space are performed to create more textures. The shape parameters, texture maps, camera views and backgrounds were all randomized for each animation. To render realistic human images, we used 3456 random background images from the COCO Densepose train set and the synthetic human model is made to move in 3D space between the cone of camera and image plane based on the MOCAP animations. A varying distribution of human data is thus created in the 3D environment where synthetic humans are programmed to perform random actions for a sequence of frames with varying backgrounds.

The Unity Real-Time Development Platform was used to render all video data. 9 different synthetic cameras were used to create synthetic data where avatar animation was overlayed onto the existing background image that was placed within the cone of the rendering camera. Synthetic camera noise and camera lens distortion was added to the rendered scene to impart more realism. Separate SMPL human models were used for male and female bodies to obtain precise physiological representations. A synthetic light source is placed at a random location closer to the human avatar to create lighting and illumination effects on the human surface in the 3D scene. The intensity and range of the light source were randomly changed within a visually plausible range to create better lighting effects. We used a windows desktop with a single Nvidia Tesla RTX 2080 Graphics Processing Unit (GPU) to render the images. The various stages of our data acquisition and data post processing are described in Figure 2.

B. Data Generation Methodology

The fully automated data generation pipeline used to render synthetic data is as follows:

- A random (statistically realistic) human shape is created using SMPL by randomly varying the shape blend-shapes (within a factor of -2.0 – 2.0 along the principal component vectors, based on testing and observations) for each frame
- Human texture is loaded and mapped onto the model created in step (1) for every frame
- One of the MOCAP animation is fitted onto the human model using pre-configured skeleton mapping
- One of the 9 rendering cameras is randomly selected
- A random background scene is loaded for every frame
- Synthetic camera noise and camera lens distortion are added to the rendered scene
- The animation plays and each frame is saved (fixed frame rate, 15 FPS chosen for this project)
- A maximum threshold for the number of frames that corresponds to the animation length to be played is set.
- A light is placed near the avatar with random range and intensity.
- All the calculations to extract ground truth information are performed for every frame.
- All the frames corresponding to a single animation are saved as images along with their ground truth information.
- Restart from step 1. Until final animation has played.

C. Mesh Vertex Visibility Test in 3D

To achieve dense correspondence, the visible vertices of the mesh for a given view needs to be obtained. The vertex

![Fig. 2. Synthetic data generation pipeline](https://example.com/synthetic-data-generation-pipeline.png)
visibility in 3D is dependent on the camera view $\alpha$ and pose parameters $\theta$. We compute the concave hull of the SMPL human mesh in 3D from the mesh vertices to get precise visible vertex locations of the mesh surface. For a given view, we used the method of Ray casting for each vertex to resolve its visibility. Rays are cast from each vertex locations to the corresponding camera center in order to check if they collide with the concave hull as depicted in Figure 3. The non-colliding vertex locations are considered visible with respect to the given view. If the ray collides with the concave hull, another ray is cast from the colliding coordinate to check whether the surface location is an occluded point in 3d space. Precise world coordinate vertex locations are extracted by comparing the 3d coordinates of the vertex position and collision point. All visible vertices in the world coordinate 3D space are transformed into image space in-order to retrieve visible vertices of the human 3D mesh in the rendered 2D image space. This information helps to achieve human body correspondence between xy locations in the image domain and uv coordinates in the uv space.

$$Visibility(v_i) = F(\alpha, \theta), \quad (3)$$

where $v_i$ denotes 3D vertex coordinate and $\alpha$, $\theta$ are camera parameters, pose parameters respectively.

D. Semantic Region Mapping and IUV Segmentation

We divide the human mesh in SMPL uv space into 24 part atlas uv space as discussed in [21]. We believe that the uv coordinate regression through a continuous uv field would be inappropriate due to the general symmetric nature of the human body. Hence, regression needs to be performed in a smaller uv space where there are more unique pixel features for a particular body part. The whole human mesh is divided into 14 semantic regions like Head, Torso, Lower/Upper Arms, Lower/Upper Legs, Hands and Feet etc. To extract ground truth data, all points in the image space are projected on to the world coordinate space to see if these points collide with the concave hull of the human mesh. All such coordinates of projection in 2D are obtained as a dense mapping of the human mesh in the image space. This information can be used to get precise human segmentation ground truth in image space.

The projection of the visible vertices onto the rendered image plane is then used as ground truth 2D-3D mapping. These are the control points used for regression of the visible part of the human mesh. The uv coordinates in smpl space are transformed to 24 part atlas uv space to perform IUV regression of human body parts. To obtain the semantic region mapping, we project all the screen space coordinates from corresponding camera to check whether the rays hit the concave hull. If a point in screen space when cast hits the field, then the collision point is associated with the closest vertex coordinate in world coordinate space such that a correlation is established between all 2D points in screen space on the human surface and the mesh vertices. This relationship can be used to extract the corresponding uv coordinate for every visible point on human in the image domain. This idea also helps to extract human semantic region mapping in the image domain. The SMPL uv space can be divided into the 14 semantic regions mentioned above using a mapping relation between atlas uv space and SMPL uv space. This mapping in SMPL uv space is used as a look up to extract ground truth semantic region mapping in image space which is shown in Figure 4. The association created between screen space coordinates and uv space coordinates helps to gather precise semantic region information in the image domain.

E. Human Pose Keypoints

According to [21], integration of human pose keypoints as an auxiliary task along with human segmentation improves the uv regression performance. We retrieve the human pose
keypoints from the human MOCAP skeleton used for creating the animation. The movement of the skeleton corresponds to the movement of real people who performed particular actions during mocap data collection. The visibility of a particular part, (e.g., eyes, ears, nose etc) is determined by identifying whether the majority of the human mesh vertices surrounding the part are visible. We also affix some keypoint locations on the mesh surface itself rather than the skeleton keypoint for convenience. 17 human pose keypoints are obtained in 3D overall. The identified 3D keypoint locations are transformed to retrieve the keypoint location in image space. The visibility algorithm mentioned above is utilised to identify whether the keypoint location is visible or not from a given camera view.

F. Occlusion Generation and Alpha Blending

To create occlusions, we used 227 segmented 2D mundane objects which were extracted using a background removal tool. An object was randomly sampled and re-scaled according to the human Region of Interest (ROI) in the scene. Then, the object is positioned over the rendered human in the image. The 2D objects that are used to create occlusion have very sharp edges when positioned on top of the human surface. To provide smooth transition between the edge of the 2D object and the rendered image, a random Gaussian filter and alpha blending are applied over a thick inner and outer boundary of the object. This allows for seamless transition between object edge and background.

G. Scene Harmonisation for Visual Data Generation

To better match the “style” of the human avatar and occluding object with the background, a scene harmonisation method was used [26]. It uses a generative adversarial network conditioned on the ROI (the human mask and the occluded object mask) that transforms the target human pixel features to adapt more with the background style of the image as depicted in Figure 5.

H. Dense Correspondence Regression

The task of estimating dense correspondence between 2D images and 3D surfaces is associated with several primary tasks including human classification, human bounding box estimation and human segmentation. These tasks are necessary for the subsequent method. We used the detectron2 [27] github repository which is well known for object detection and localisation to perform these primary tasks. A 24 part uv field regression was used to achieve dense correspondence between image pixels and surface points. We used the COCO Densepose annotation system to train all the models. The annotation system consists of human segmentation masks, human semantic part segmentation maps and the uv mapping in correspondence with the image space coordinates of the human. We used the Run Length Encoding (RLE) encoded dense masks to represent the 14 semantically meaningful parts of the human body. This includes Torso, Right Hand, Left Hand, Left Foot, Right Foot, Upper Leg Left, Lower Leg Left, Upper Leg Right, Lower Leg Right, Upper Arm Left, Lower Arm Left, Lower Arm Right and Head. To map the IUV correspondence to these parts, we used the 24 part texture space to which the human surface points are transformed. The IUV regression is performed using this space to learn dense correspondence between image space and human surface.

We utilised a fully convolutional neural network that includes a classification and regression branch as mentioned in [20] in order to regress the 2D-3D correspondence. The objective of the classification branch is to identify whether a pixel belongs to the background or one of the 24 body parts defined within the region of interest. Cross-entropy loss is used to train this part of the network. The regression branch is trained using a smooth L1 loss to predict the uv coordinates within each of the 24 parts of the human surface. The coordinate regression at a position i can be formulated as follows:

\[ c^* = \arg \max_c P(c|i), \quad [U, V] = R^C(i) \quad (4) \]

where the index i to the body part c* that has the highest posterior probability is determined by the classification branch. In the second stage, the regressor \( R^C \) places the point i the continuous U,V coordinate parameterization of part c*. 24 regression functions \( R^C \), each predicting 2D uv coordinates within the respective part c are trained using a smooth L1 loss function and the loss is taken into account for a part only if the pixel is predicted within that part by the classification branch.

ROI pooling is performed using the human bounding box calculated based on the human mesh vertex coordinates in the image domain. These features are pooled into a dimension of 256x256 to obtain scale-invariance. The (x,y) coordinates in the image domain that correspond to the human are also transformed to this 256x256 dimension such that they provide precise ground truth information related to the human in the image domain. A multi-task cascaded architecture is used to exploit the different sources of information available for supervision. The aim is to learn human dense 2D-3D correspondence and utilise human segmentation as an auxiliary task to learn these features. The Mask-RCNN architecture is used to learn human segmentation features. Thus, the task
of dense correspondence is learned with a supervision from ground truth segmentation information using the multi-task cascaded architecture.

As discussed in [21], “in-painting” the values of the supervision signals for which there are no ground truth information provides better regression results. A learning based approach in which a teacher network is trained to reconstruct dense supervision signal based on a sparse supervision signal and the input image is used to in-paint missing values. The human mesh vertex coordinates that has correspondence with the uv coordinates are sparse. The in-painting network is used to interpolate dense uv coordinates for the locations where the uv coordinate ground truth is not available. The ROI consisting of human vertex coordinates in the image domain that has a correspondence with the dense uv field is trained to establish dense correspondence between the synthetic human and the SMPL human mesh.

### III. EXPERIMENTS & RESULTS

#### A. COCO Densepose Training

According to our goals, we evaluated the performance of synthetic data that we generated for the problem of dense correspondence estimation. We used the detectron2 github repository for training the dense correspondence model. For all experiments, we used Resnet-50 backbone as the primary feature extractor and the Panoptic head from [28] and DeepLabV3 head from [29]. A hyperparameter free training strategy is followed according to the linear scaling rule from [30]. Comparison between the same model setting trained using real data and synthetic data are discussed in the coming sub-sections. To avoid over-fitting, we initialised the model with weights trained on real data and only the last layers i.e, the part of the model where the IUV estimation occurs, is re-trained. We also used an additional mask-based supervision for training the coarse segmentation parts (14 parts) whose weights are randomly initialised. Thus, the human mask estimation and the IUV dense correspondence estimation is completely learned from our synthetic data. We trained the model for 35,000 iterations with mini-batch size 2, which is almost equivalent to an epoch at a learning rate 0.001 on a single Nvidia Tesla T4 GPU to avoid over-fitting. To train the model, we used 73,671 synthetic human instances containing directly rendered images without occlusion and images post-processed by adding occlusion along with scene harmonisation.

#### B. Evaluation Metrics

We report the standard COCO Densepose evaluation metrics described in [21]. We consider the human bounding box estimation (bbox), human segmentation mask estimation (Segm), Geodesic Point Similarity (GPS) and Geodesic Point Similarity within the correctly predicted human part mask (GPSm) to evaluate the dense correspondence. For all tasks, the mean Intersection over Union (mIoU) is considered with thresholds for each metric that correspond to a good percentage of intersection. The precision averaged over IoU thresholds (AP) is the primary metric followed by AP$_{50}$ and AP$_{75}$ being the selected supplementary metrics. We also compute the average recall (AR over IoU thresholds) followed by AR$_{50}$ and AR$_{75}$ for comparison. AP for any metric is average precision over 10
threshold values that range from 0.5:0.05:0.95 whereas AP$_{50}$ is AP at 0.5 (loose metric) and AP$_{75}$ is AP at 0.75 (strict metric). Similarly, AR for any metric is average recall over 10 threshold values that range from 0.5:0.05:0.95 whereas AR$_{50}$ is AR at 0.5 (loose metric) and AR$_{75}$ is AR at 0.75 (strict metric). The GPS is based on geodesic distances on the template mesh between the ground truth and estimated surface point for the same image points as follows:

$$\text{GPS} = \frac{1}{|P|} \sum_{p_i \in P} \exp\left(-\frac{d(\hat{p}_i, p_i)^2}{2k(p_i)^2}\right) \quad (5)$$

where $d(\hat{p}_i, p_i)$ is the geodesic distance between estimated ($\hat{p}_i$) and ground truth ($p_i$) human body surface points and $k(p_i)$ is a per-part normalization factor, defined as the mean geodesic distance between points on the part. The masked geodesic point similarity (GPSm) is calculated as:

$$\text{GPS}^m = \sqrt{\text{GPS} \cdot \text{I}}, \quad \text{with} \quad I = (\text{M} \cap \hat{M})/(\text{M} \cup \hat{M}) \quad (6)$$

where GPS is the geodesic point similarity metric value and I is the intersection over union between the ground truth (M) and the predicted $\hat{M}$ foreground masks.

C. Quantitative Evaluation

The COCO densepose test set containing 1508 images with 5581 person instances are used for performing quantitative evaluation. Table I shows the results of the dense correspondence model trained with synthetic data. Table II shows the state-of-the-art model results [21]. Table III shows another experiment where the bounding box estimation is performed by the state-of-the-art model while all the remaining metrics are evaluated by the model trained with synthetic data. The results depict that with better known human ROIs, all other metrics showcase improvements.

D. Qualitative Evaluation

We provide some comparative results between the human semantic region estimation by the model trained using synthetic data and estimation performed by the pre-trained model trained on real data in Figure 6. The human pixels inferred within the ROI are then transformed to the SMPL UV space for further comparison of the 2D human image to 3D surface mapping. Figure 7 shows comparison of the human textures inferred by the model trained with synthetic data and the pre-trained model trained using real data.

With the Densepose model trained on our generated synthetic human data, we achieved qualitative improvement in the mask estimation, especially in areas of boundary between human and background. We believe that this improvement stems from the more accurate (exact) ground truth segmentation masks that were generated from the synthetic environment, as described in section II-D. The comparison of semantic region mask estimation between the model trained on real data (from detectron2 github repository [27]) and the model trained with our synthetic data is shown in Figure 6. In addition, qualitative comparison of the textures estimated using models trained using real-only data and the synthetic model showed visual improvements in the estimated texture. Figure 7 shows a comparison of the textures extracted using both models. In Figure 6, the boundaries of the human semantic region segmentations, match more accurately with the outline of the human (e.g., hair, cap) than the SOTA model. In Figure 7, the texture map shows better pixel spread. The gap between the facial regions is reduced when inferred using the model trained on synthetic data, which can be explained by denser training landmark points. Clothing showed better inference with larger correct regions estimated.

E. Experimental Analysis

To evaluate the quality of the synthetic data that we generated, we rendered 3D humans on background images without humans in it. Because the presence of non annotated humans in the background would affect metrics like precision and recall. We used the pre-trained model available on the detectron2 github repository to evaluate its performance on the mentioned synthetic data that we generated, consisting of 1866 synthetic human instances. The performance of the model evaluated on the synthetic data that we generated are reported in the table IV. The results showcase that the model trained on the diverse COCO real data shows good performance on the synthetic data that we generated. This is another evidence that the synthetic data created resemble real data.

| Task  | AP  | AP$_{50}$ | AP$_{75}$ | AR  | AR$_{50}$ | AR$_{75}$ |
|-------|-----|-----------|-----------|-----|----------|----------|
| bbox  | 87.80 | 99.01 | 97.87 | 87.80 | 99.01 | 97.87 |
| GPS   | 58.78 | 97.78 | 70.84 | 63.92 | 98.87 | 80.44 |
| GPSm  | 71.35 | 97.78 | 93.37 | 74.73 | 99.14 | 95.98 |
| Segm  | 85.03 | 99.01 | 98.94 | 86.70 | 99.73 | 99.19 |
IV. CONCLUSION

In this paper, we studied the possibility of training deep-learning models for human 2D-3D mapping using synthetic data, which has the advantage of being dense, precise, unbiased, labour-free and solves issues of data volume and privacy. Utilising a synthetic data generation pipeline for creating ground truth makes it easier to generate more diverse data easily once the method is developed. Initial experiments show promising quantitative and qualitative results that the synthetic model has the potential to improve fine edge accuracy, and to solve the bottleneck of data scarcity and manual ground truth inaccuracies in 2D-3D applications. The current gap between synthetic and real data can cause an overfitting of the model to the synthetic data, which can be solved using advanced domain adaptation techniques. This presents a clear avenue for work on synthetic data for 2D-3D mapping.

REFERENCES

[1] Gul Varol, Ivan Laptev, Cordelia Schmid, and Andrew Zisserman, “Synthetic humans for action recognition from unseen viewpoints,” International Journal of Computer Vision, vol. 129, no. 7, pp. 2264–2287, 2021.

[2] Denis Zherdev, Larisa Zherdeva, Sergey Agapov, Anton Sapochnikov, Artem Nikonorov, and Sergei Chaplygin, “Producing synthetic dataset for human fall detection in ar/vr environments,” Applied Sciences, vol. 11, no. 24, pp. 11938, 2021.

[3] Priyanka Patel, Chun-Hao P Huang, Joachim Tesch, David T Hoffmann, Shashank Tripathi, and Michael J Black, “Agora: Avatars in geography optimized for regression analysis,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13468–13478.

[4] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis, “Learning to reconstruct 3d human pose and shape via model-fitting in the loop,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2252–2261.

[5] Matteo Fabbri, Guillem Braso, Gianluca Maugeri, Oscar Cetintas, Riccardo Gasparini, Aljosa Osep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara, “Motsyn: How can synthetic data help pedestrian detection and tracking?,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10849–10859.

[6] Carl Martin Grewe and Stefan Zachow, “Fully automated and highly accurate dense correspondence for facial surfaces,” in European conference on computer vision. Springer, 2016, pp. 552–568.

[7] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon, “The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation,” in 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012, pp. 103–110.

[8] Tinghui Zhou, Philipp Krakenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros, “Learning dense correspondence via 3d-guided cycle consistency,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 117–126.

[9] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black, “Smpl: A skinned multi-person linear model,” IEEE, 2012, pp. 103–110.

[10] Cesar Roberto de Souza, Adrien Gaidon, Yoann Cabon, Naila Murray, and Antonio Manuel López, “Generating human action videos by coupling 3d game engines and probabilistic graphical models,” International Journal of Computer Vision, vol. 128, no. 5, pp. 1505–1536, 2020.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” Advances in neural information processing systems, vol. 27, 2014.

[12] Ronald Poppe, “A survey on vision-based human action recognition,” Image and vision computing, vol. 28, no. 6, pp. 976–990, 2010.