Cloning Outfits from Real-World Images to 3D Characters for Generalizable Person Re-Identification: Appendix

Yanan Wang[1] yanan.wang.cs@gmail.com
Xuezhi Liang[2] xz.liang.cs@gmail.com
Shengcai Liao[1*] scliao@ieee.org

Inception Institute of Artificial Intelligence (IIAI)[1]
Mohamed bin Zayed University of Artificial Intelligence[2]
Masdar City, Abu Dhabi, UAE

A. Introduction

Due to space limits, we are not able to explain everything in detail in the main paper. In this Appendix, we further present more details of our implementations, and demonstrate more illustrations to explain our design choices. Besides, we present more experimental results for further understanding.

All methods used and designed in the project are listed in Tab. A, including existing methods, adapted methods, and the proposed methods. For example, we adapted some existing methods in the person image pre-processing stage to help cherry-pick best-viewing person images and determine clothes positions, categories and clothes keypoint locations. At the same time, we also propose some new methods, such as Registered Clothes Mapping, Homogeneous cloth Expansion and Similarity-Diversity Expansion, to achieve the goal of mapping real clothes to virtual people. Finally, we create the ClonedPerson dataset that can improve the generalization performance of person re-identification.

B. Person Image Pre-Processing

B.1. Pedestrian detection

Our target is to clone the full-body outfits from real-world person images to virtual 3D characters. However, considering the variety of clothing images in real life, we need to avoid images of standalone clothes and incomplete person images. Therefore, we apply a person detection model, Pedestron [3], to detect qualified person images. We keep the original configuration of the Pedestron [3] and set the detection threshold to be 0.8 to avoid images of standalone clothes and incomplete person images. Different situations of person images are shown in Fig. A. Furthermore, we set the area of the detected bounding boxes to be at least 20% of the input image to remove low-resolution persons and some false positives. The detected person images are cropped for the following pose detection procedure.

Characters in some existing synthetic datasets are dressed in random collocation, such as in RandPerson [11] and UnrealPerson [12]. However, random collocation sometimes creates incongruous characters, as shown in Fig. B. The left side shows person images and characters created by the proposed cloning method. The right side shows characters created by randomly combining some upper-body and lower-body clothes. We can see that the collocation on the right is inconsistent. Therefore, the use of person detection in localizing full-body person images is also for the purpose of cloning the full-body outfits from real-world person images to virtual 3D characters. As a result, the proposed method follows the original collocations of real-life persons, and so the sample distributions of our data would be more consistent with real-life persons.

B.2. Person view qualification by pose detection

After person detection, another problem is that person images may have different viewpoints, such as frontal view, back view, and side view. Furthermore, the frontal view images are divided into two situations: occluded and non-occluded. For our purpose, back-view, side-view, and occluded front-view images are all incomplete displays of clothes, so they are regarded as noisy data. To reduce these noisy data, we use person pose estimation model for automatic judgment. Specifically, we apply the HRNet [9] model from MMDetection [1] to do person pose estimation. It is trained on the COCO dataset [6]. HRNet predicts 17 body keypoints and their visibility probabilities, from which we use 12 keypoints on the body, including shoulders, elbows, hands, hips, knees, and feet. According to the positions of the shoulders, back-view images could be classified. With the width-to-height aspect ratio of the upper body, side-view images could be distinguished. Based on the position of hands and elbows, we can identify occluded

*Shengcai Liao is the Corresponding Author.

Method	Category	Notes
Person Detection	Existing	Pedestron [3]
Pose Detection	Adapted	We used the existing HRNet [9] model from MMDetection [1]. Specific rules are designed based on the detected keypoints to cherry-pick best-viewing person images.
Clothes Detection and Classification	Adapted	We trained a model based on Faster-RCNN [8] with the annotated clothes bounding boxes and categories in DeepFashion2 [2].
Clothes Keypoint Detection	Adapted	We annotated clothes keypoints and trained a model based on PIPNet [4].
Registered Clothes Mapping	Proposed	We annotated clothes keypoints on regular UV maps, detected clothes keypoints on person images, and applied the perspective homography method to warp real clothes texture to UV maps.
Homogeneous Cloth Expansion	Proposed	A new method is proposed to find a homogeneous area as large as possible on clothes images.
Similarity-Diversity Expansion	Proposed	A new method is proposed to scale up virtual character creation.

Table A. Methods used and designed in the ClonedPerson pipeline.

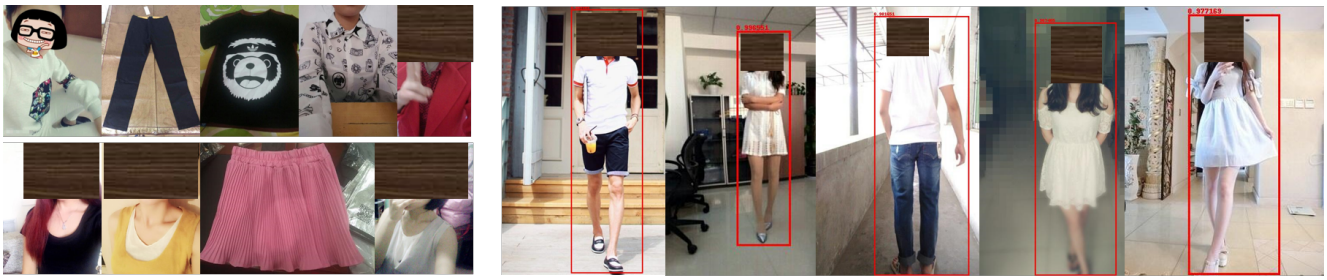


Figure A. Examples of person detection results.

images. The definition and locations of the specific keypoints used in our pipeline are shown in Fig. C(1). Then, we can classify different situations according to the following rules:

1) **Back view**. The right shoulder ($P6$) is on the right side of the left shoulder ($P5$) on the image.

2) **Side view**. The width to height aspect ratio W/H of the person's upper body is less than 0.3.

3) **Occluded**[1]. Any hand or elbow point ($P7$, $P8$, $P9$, $P10$) is in the upper body area (the area surrounded by $P6$, $P5$, $P11$, and $P12$) or the lower body area (the area enclosed by $P12$, $P11$, $P13$, and $P14$).

For the width-to-height aspect ratio of the upper body (Rule 2), as shown in Fig. C(1) and Fig. C(3), we consider the Euclidean distance between shoulders ($P5$ and $P6$) as the upper-body width W, and that between the center of the shoulder (the middle point of $P5$ and $P6$) and the center of the butt (the middle point of $P11$, and $P12$) as the height H. Then, we select qualified frontal-view images with $W/H \geq 0.3$.

For the judgment of occlusion (Rule 3), since the detected points are not on the edge of the body, we define the upper-body and lower-body areas by expanding the surrounding points. First, we define each area's width according to the top corner-point distance of that area. Specifically, as shown in Fig. C(4), w_1 is the width of the upper-body area, and w_2 is the width of the lower-body area. Then, we extend the upper-body area and lower-body area horizontally by $w_1^{'}$=$0.1\times w_1$ and $w_2^{'}$= $0.1\times w_2$, respectively.

As shown in Fig. C, Fig. C(1) is a qualified frontal-view and non-occluded image. According to the position of the shoulders (Rule 1), Fig. C(2) is a back-view image. Fig. C(1) shows an example of $W/H \geq 0.3$, while Fig. C(3) is a side-view image because $W/H < 0.3$ (Rule 2). Fig. C(4) is classified as an occluded image based on the

[1]Note that only self-occlusion is considered here. Though, with the visibility probabilities predicted by HRNet we can also infer occlusion by other objects, this is not yet considered in the current pipeline.

Figure B. Examples of different combinations of upper-body and lower-body clothes. Left: the proposed cloning of the full-body outfits, in their original collocations. Right: random combination.

(1) Qualified. (2) Back-view. (3) Side-view. (4) Occluded.

Figure C. Different viewpoints judged by pose detection. (1) A qualified image, where the left shoulder $P5$ is on the right side of the image, $W/H \geq 0.3$, and hands are not in the body area. (2) A back-view image, where the left shoulder $P5$ is on the left side of the image. (3) A side-view image, where $W/H < 0.3$. (4) An occluded image, with hands in the body area.

position of hands (Rule 3).

Unqualified images of person views may cause some common problems for the proposed cloning method, as shown in Fig. D. For example, characters generated from back-view images may contain hairs (Fig. D(1)). Characters created from side-view images may have strange textures (Fig. D(2)). Besides, clothes occluded by hands may cause the generated characters containing ghost hands (Fig. D(3)). Therefore, the proposed person view qualification step by pose detection is useful to get qualified frontal-view and non-occluded images, and thus facilitate the cloning of clean clothes.

B.3. Clothes and keypoint detection

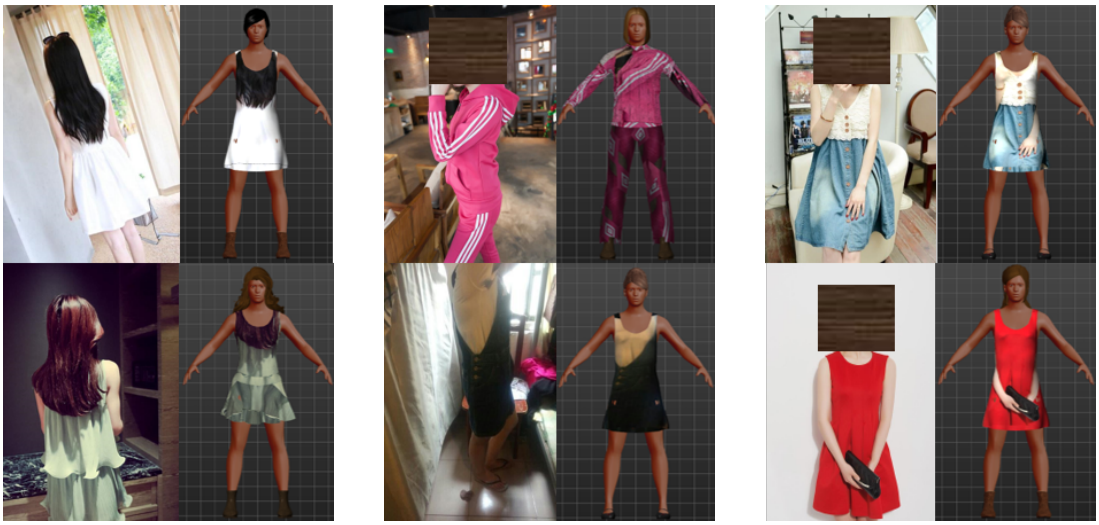
Through Appendix B.1 and Appendix B.2, we obtained images that contain persons' entire bodies and are completely visible. To achieve the mapping from real-world image to virtual character, we need to get the clothes position and type, and positions of the clothing keypoints in the image. Therefore, we further train two models: the clothes detection model and their corresponding key points detection model.

Fig. E(1) shows the types of clothes we use, and the red points display positions of keypoints. The clothing models include eight models (long sleeves, short sleeves, sleeveless, trousers, shorts, skirts, short dresses, and long dresses). After obtaining the labeling information of the clothes keypoints, the clothes detection models and the keypoint detection models are trained separately for eight clothes models. The clothes detection model is based on the faster RCNN [8] which predicts the bounding box localization and clothes category jointly. The keypoint detection model is based on PIPNet [4] without Neighbor Regression Module. Finally, we detect all pose qualified images and get the clothing category and the keypoint locations of clothes.

C. Registered Clothes Mapping

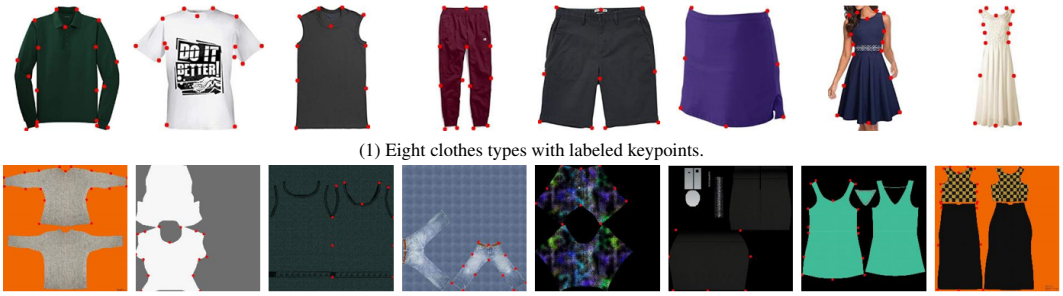
With 3D clothes models available in the MakeHuman community, we obtain some clothes models with regular UV maps, where clothes appear in regular shapes and structures, as Fig. E(2) shows. With these regular UV maps, it is possible to apply Registered Clothes Mapping to map real-world clothes textures to virtual models. However, the structure of some UV maps is not clear, so we need a way to find out its structure.

Changing the UV map will change the appearance of the



(1) From back-view images (2) From side-view images (3) From occluded images

Figure D. Characters created from images of different person views.



(1) Eight clothes types with labeled keypoints.

(2) Regular UV maps where clothes appear in regular shapes and structures.

(3) Irregular UV maps.

Figure E. Different types of clothes and UV texture maps of the corresponding 3D clothes models.

3D model because there is a correspondence between the UV map and the model. As Fig. F shows, firstly, we use a pure black image as the UV map, and get the model's front-view image as a reference image. Next, a 50×50 white square is used to traverse the UV texture map and get many corresponding front-view images as response images. Then, by comparing these response images and the reference image, we could find out which area in UV maps would be mapped to the front of the model. Finally, by stacking these squares, we can get the approximate area of the texture on the front of the model. Fig. F shows some frontal areas founded by this method. Accordingly, different region division and keypoint labeling and mapping rules are designed according to different structures of the UV maps.

Multi-view strategy. Note that We aim at developing a general system that requires only one single image, as multi-view images are not always available. However, when multi-view images are available as inputs, it is quite straightforward to integrate them into different parts of regular UV maps. An example is shown in Fig. H.

D. Homogeneous Cloth Expansion

As discussed in the main paper, to generate clothes textures for irregular UV maps, and textures on regular UV maps corresponding to invisible person parts, we further design a homogeneous cloth expansion method to find a homogeneous area on clothes as a realistic cloth cell, and expand the cell to fill the UV map. Fig. I shows some examples of the optimized cloth cells by the proposed algorithm. From these examples, we can see that the proposed

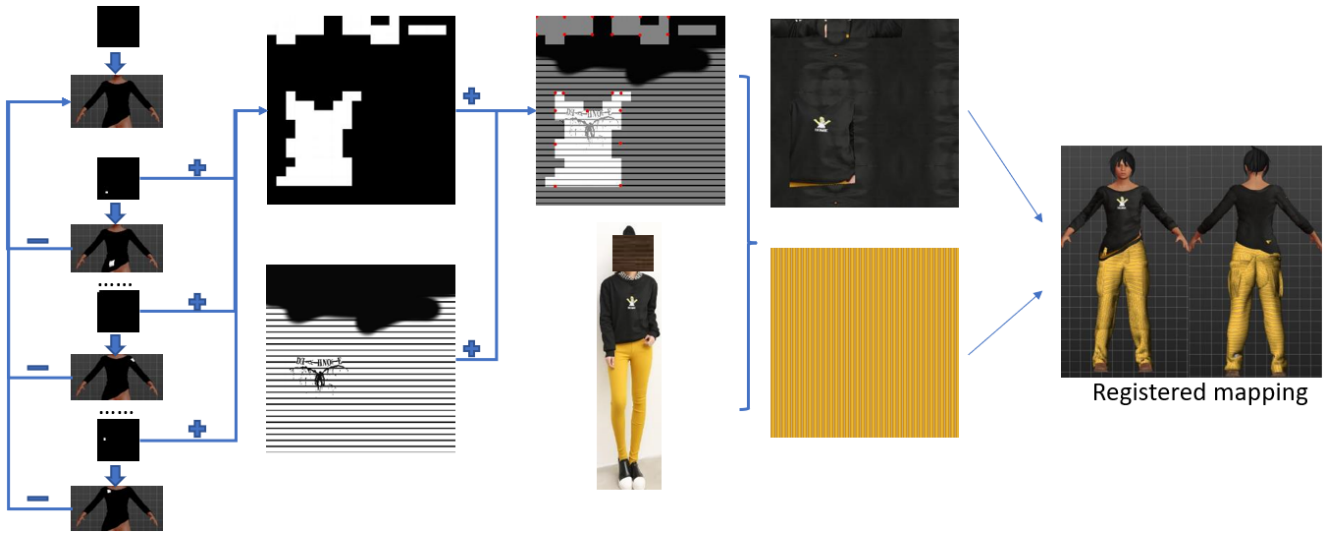


Figure F. Find out clear structure in UV maps.

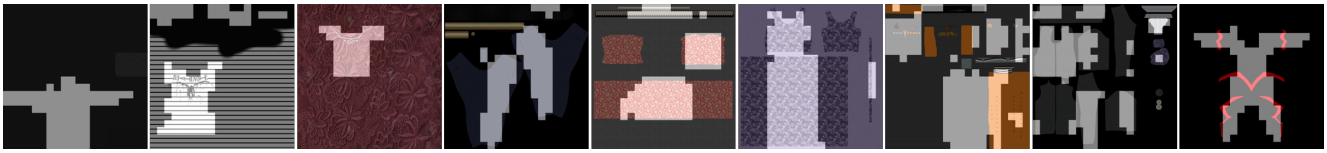


Figure G. Examples of founded frontal areas in UV maps.

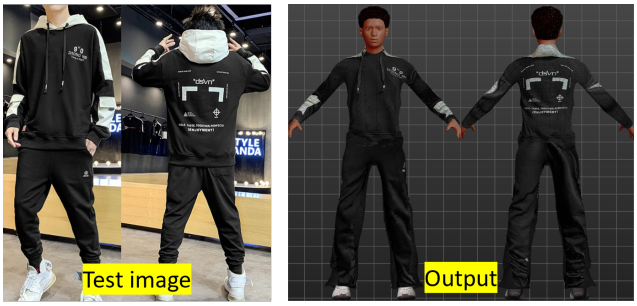


Figure H. Character generated by multi-view images.

method is able to find a homogeneous cloth patch as large as possible.

Besides the proposed homogeneous cloth expansion method, given a cropped cloth cell, a simple way to create a UV map is to resize the cloth cell directly as a UV map, as proposed in RandPerson [11], and also used in UnrealPerson [12]. However, simply resizing the cloth cells may result in blur textures and unrealistic patterns. For example, Fig. J shows a comparison between resizing and the proposed expansion methods. As can be seen, characters created by the proposed expansion method have more realistic textures, while those created by resizing are usually blur. Besides, textures created by resizing usually do not match the pattern scale of the original clothes, and thus are not able to represent the original clothes. This can also be observed from synthesized images of UnrealPerson . In contrast, the proposed expansion method usually has a better consistency of pattern scales.

E. Unity3D Simulation and Rendering

As for the rendering process, we follow RandPerson [11] for the Unity3D environment settings, including the scenes, the configuration of camera networks and character movements, video capturing, and image cropping. In addition, we implement some adjustments to improve the rendering:

Camera filter. We set post-processing effects for some cameras to increase the imaging variations and make the data more diverse. Post-processing effects include color grading, bloom, grain, and vignette provided in Unity3D.

Actions. To make the generated data closer to the real-world data, we remove the running and uncommon walking actions in RandPerson. Instead, we include the situation of hanging out in place, allowing the character to stand in place and move hands or turn around, enriching the data diversity.

Scenes and cameras. The number of cameras in each scene should be expanded to increase rendering efficiency and viewpoint diversity. Since some scenes in RandPerson are too small to expand cameras, we select five out of 11

Figure I. Examples of optimized cloth cells by the proposed algorithm.

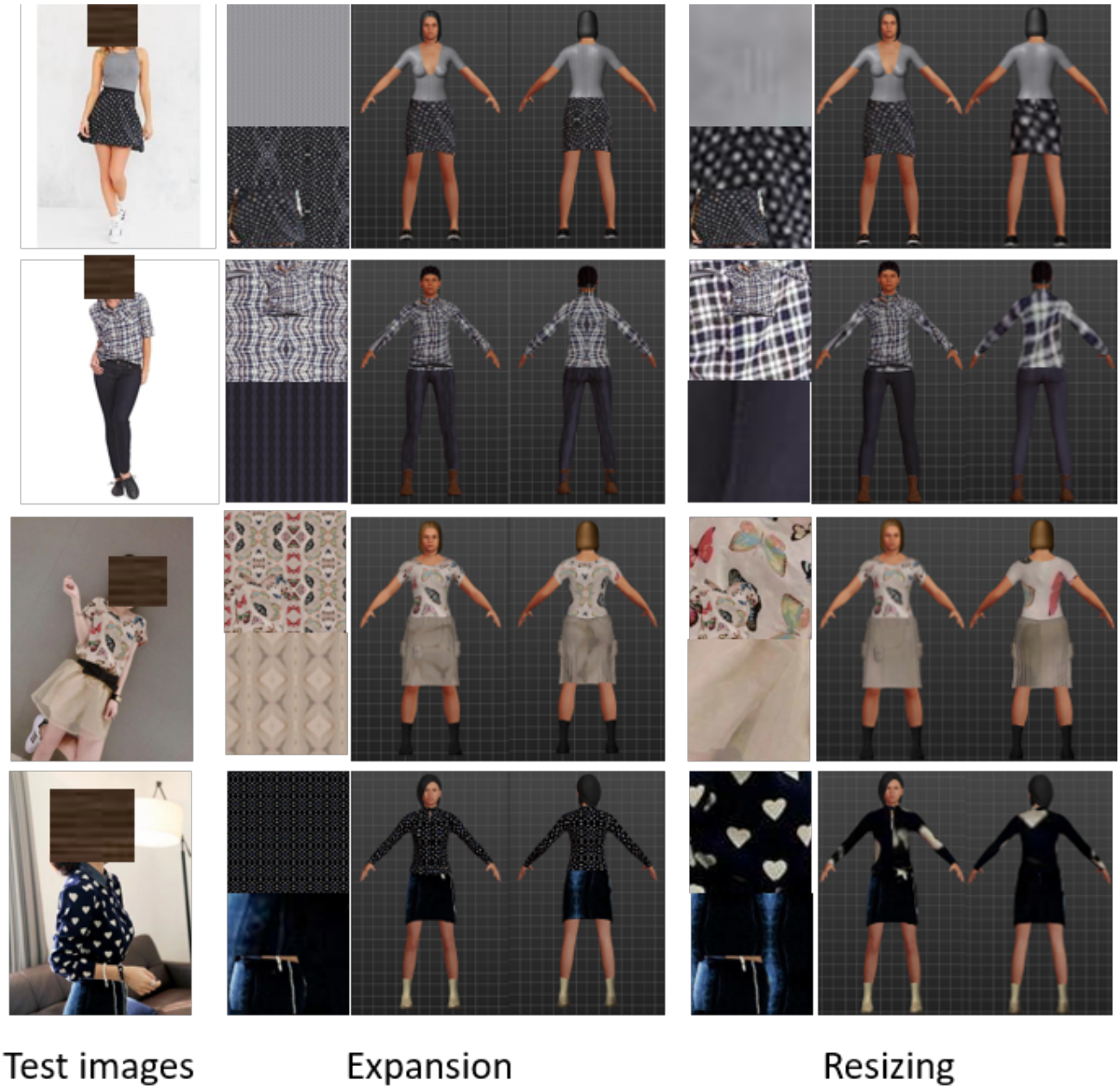


Figure J. Comparison of expansion and resizing methods in generating UV maps and characters.

scenes in RandPerson (scene2, 3, 5, 6, and 10) and create a new scene ourselves to get more complex lighting. We expand the number of cameras in each scene to four, making each scene's proportion in the database more balanced. In total, RandPerson uses 19 cameras in 11 scenes, while we use six scenes with 24 cameras. Fig. K shows the six scenes with 24 cameras we use.

Image cropping. We make further improvements with RandPerson's image cropping strategy by introducing random disturbances to the cropping. Cropped persons in RandPerson are mostly complete and well-aligned. However, there are many incomplete and misaligned person images in real-world datasets. Therefore, we make random disturbances to the cropping to simulate partially visible and misaligned person images. Specifically, let the width and height of the original image be W and H, respectively. For each image, with a probability ρ=30% we randomly choose to further crop the image. Then, for the selected image with further cropping, we remove the top 0-$0.1H$ part of the image randomly, and remove the bottom 0-$0.5H$ part randomly. Then we randomly use one of the three strategies (left side only, right side only, and both sides) to remove some content randomly in 0-τW of the original image, where the side rate τ=0.3 by default. Fig. L illustrates the process and some cropped examples. Tab. B shows the results of using different cropping strategies.

With the above setup, the generated 3D characters are imported into Unity3D environments to render and crop person images.

F. ClonedPerson Dataset

An automatic pipeline is described in the main paper to directly clone the whole outfits from real-world person images to virtual 3D characters. Fig. M shows some examples of 3D characters in ClonedPerson. For the whole process of creating the ClonedPerson dataset, the specific information in each step is detailed as follows. For each image, person detection needs 0.28s, pose detection needs 0.15s, clothes and keypoint detection needs 0.23s, clothes mapping and 3D creation needs 27.65s, and Unity3D rendering needs 16.5s. Therefore, for the whole pipeline each image costs 44.8s in total.

For training clothes detection, we use 191k diverse images of 13 popular clothing categories from DeepFashion2 [2]. The clothes keypoint detection training data is composed of DeepFashion2 and crawled clothes images, in which we annotate 17k images manually. After removing the invalid images, we finally select about 10k images of eight clothing categories that we use in this paper to train the clothes detection and clothes keypoint detection models.

For cloning clothes from real-world person images to virtual characters, we use images from both DeepFashion [7] and DeepFashion2, with a total of 409k images as our source data. By employing person detection, 146k person images are selected which contain detected persons. Then, 83k images are qualified by viewpoint judgment employing pose detection. Among them, 65k images are successfully detected with clothes bounding boxes and categories, as well as clothes keypoint positions.

In the clustering stage, we use eps=0.4 to remove 29k images due to repeating persons with the same outfits. Then, we set eps=0.5, and obtain 968 clusters with 6,340 images to create characters. Among the valid 968 clusters, we use all of the clusters and select seven images in each cluster to create our 3D characters. Since some clusters are less than seven images, finally, we get 5,621 person images as inputs and create 5,621 characters accordingly by the proposed method. After rendering and cropping, we obtain 887,766 images for the 5,621 virtual persons, and this forms our ClonedPerson dataset. Among them, we use 763,953 images from 4,826 characters for training, and 123,813 images of 795 characters for testing.

Besides person re-identification, our data can also be used for other tasks e.g. person detection, person keypoint detection, multi object tracking (with videos), multi-camera multi object tracking, etc. Fig. N shows some examples of person keypoint detection on real-world images with a model trained on the ClonedPerson dataset, with automatically recorded keypoint annotations.

G. EXPERIMENTS

G.1. Comparison of Different Cropping Strategies

Tab. B shows the performance of different cropping strategies with the cropping probability ρ and side rate τ as introduced in Appendix E. Firstly, we only change the cropping probability ρ. From the results shown in Tab. B, it can be observed that the best result is achieved with ρ=30%. Then, we keep ρ=30%, and change the side rate τ. Finally,

Prob. ρ	Side Rate τ	#ID	#Images	Rank-1	mAP
0	0	4,826	763,953	45.7	26.7
10%	0	4,826	763,953	48.9	29.9
20%	0	4,826	763,953	48.7	29.8
30%	0	4,826	763,953	49.0	30.1
40%	0	4,826	763,953	48.8	30.0
50%	0	4,826	763,953	48.8	29.9
30%	10%	4,826	763,953	49.7	31.1
30%	20%	4,826	763,953	51.2	32.3
30%	30%	4,826	763,953	52.1	33.4
30%	40%	4,826	763,953	51.3	32.7

Table B. Results of different cropping strategies with the cropping probability ρ and side rate τ.

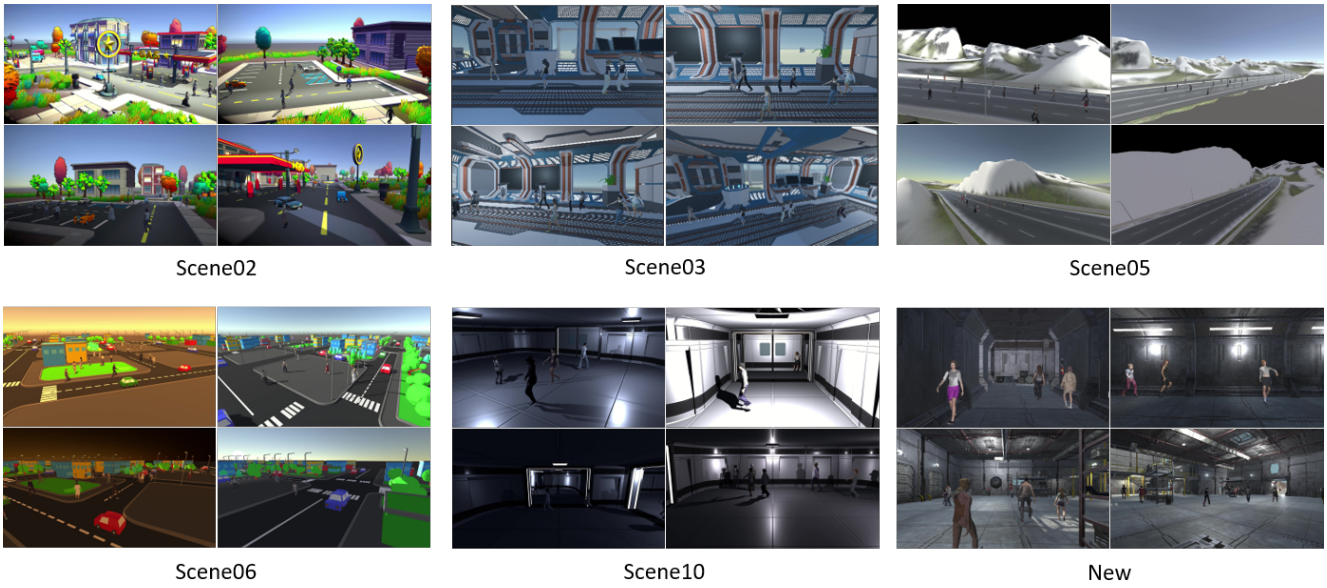


Figure K. Unity3D virtual environments utilized in this work.

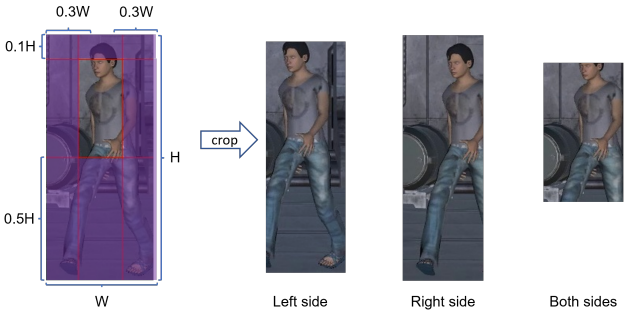


Figure L. Illustration of image cropping. Based on the result of the RandPerson's image cropping strategy, the occluded area on the left image shows the possible range of our random removals, and the images on the right are three examples of the cropped results.

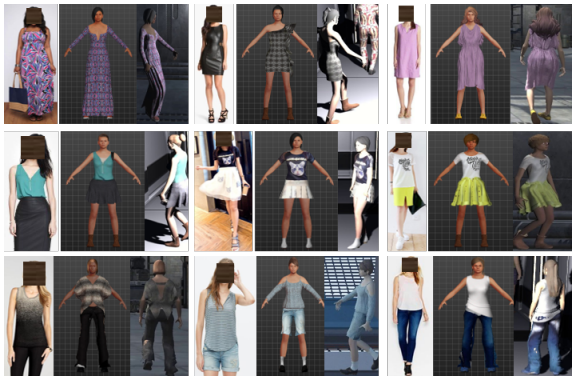


Figure M. Examples of 3D characters in ClonedPerson. Each group contains input image, generated 3D character, and rendered person image.

from Tab. B it can be observed that it achieves the best performance with the cropping probability ρ=30% and the side rate τ=30%. Therefore, the two values are kept as default values.

G.2. Comparison to Existing Datasets

Due to space limits of the main paper, we report the detailed results of different datasets for different tasks in Tab. C.

H. Limitations

As summarized below, this research leaves some aspects for improvements.

(1) Limited virtual character clothes models. The models we used are from the MakeHuman community, where the available models are limited. Because of the limited models, the categories of clothes can be applied are thus limited.

Figure N. Illustrations of keypoint detection on real-world images with a model trained on the ClonedPerson dataset.

Method	Dataset	#ID	#Imgs	CUHK03-NP		Market-1501		MSMT17		Average	
				Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
QAConv	RandPerson	8,000	1,801k	17.8	16.0	74.5	46.9	40.6	14.0	44.3	25.6
	RandPerson	8,000	132k	16.8	15.1	75.9	45.9	40.8	13.8	44.5	24.9
	RandPerson*	8,000	1,239k	20.5	20.1	81.6	56.4	46.8	17.6	49.6	31.4
	UnrealPerson	6,799	1,256k	19.2	17.2	80.0	56.1	46.0	17.5	48.4	30.3
	UnrealPerson	3,000	120k	18.8	17.8	80.6	55.9	**49.5**	**19.3**	49.6	31.0
	ClonedPerson	4,826	763k	**22.6**	**21.8**	**84.5**	**59.9**	49.1	18.5	**52.1**	**33.4**
TransMatcher	Market [5]	-	-	22.2	21.4	-	-	47.3	18.4	-	-
	MSMT17 [5]	-	-	23.7	22.5	80.1	52.0	-	-	-	-
	RandPerson	8,000	1,801k	21.2	18.7	77.6	49.6	45.3	16.4	48.0	28.2
	RandPerson	8,000	132k	18.4	16.9	77.3	49.0	44.3	15.8	46.7	27.2
	RandPerson*	8,000	1,239k	22.9	22.9	83.6	58.0	51.4	20.9	52.6	33.9
	UnrealPerson	6,799	1,256k	21.8	19.7	81.1	60.2	44.8	18.4	49.2	32.8
	UnrealPerson	3,000	120k	21.4	19.6	81.6	59.4	**52.0**	**21.6**	51.7	33.5
	ClonedPerson	4,826	763k	**25.4**	**24.4**	**84.8**	**62.3**	51.6	20.8	**53.9**	**35.8**
SpCL	RandPerson	8,000	132k	3.9	4.7	83.4	67.2	53.7	27.2	47.0	33.0
	UnrealPerson	3,000	120k	4.2	5.3	86.5	71.7	**55.2**	**28.4**	48.6	35.1
	ClonedPerson	4,826	75k	**11.7**	**12.0**	**88.0**	**72.7**	49.3	24.2	**49.7**	**36.3**

Table C. Results with different datasets for different tasks. RandPerson* means an adapted RandPerson dataset rendered with the same settings of the ClonedPerson.

(2) Limited data source. We mainly use images from DeepFashion and DeepFashion2 datasets to create our virtual characters. This makes the data source not diversified enough. We show a distribution of the DeepFashion and DeepFashion2 images in Fig. O. We use the same model trained on MSMT17 by QAConv 2.0 to compute similarity scores between images, and draw a sample distribution by t-SNE [10]. By this plot, we can find that clothes in DeepFashion datasets are not diversified enough. For example, most of the images are summer clothes in white or black. Therefore, we need to exploit more data sources in the future.

(3) Only clothes considered. The proposed method only clones clothes from person images, but is not capable of high-fidelity reconstruction of 3D models from person images. However, our motivation is to create diversified characters with realistic clothing and create a dataset for improved generalization. High-fidelity reconstruction is challenging and not efficient for our purpose. On the other hand, high-fidelity reconstruction of identifiable biometric signatures, e.g. faces, may also raise privacy concerns.

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155, 2019. 1, 2

[2] Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. CVPR, 2019. 2, 7

[3] Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, and Ling Shao. Generalizable pedestrian detection: The elephant in the room. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11328–11337, 2021. 1, 2

[4] Haibo Jin, Shengcai Liao, and Ling Shao. Pixel-in-pixel net: Towards efficient facial landmark detection in the wild. International Journal of Computer Vision, Sep 2021. 2, 3

[5] Shengcai Liao and Ling Shao. Transmatcher: Deep image matching through transformers for generalizable person re-identification. Advances in Neural Information Processing Systems, 34, 2021. 9

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 1

[7] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. 7

Figure O. Distribution of DeepFashion and DeepFashion2 images, made by t-SNE [10].

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28:91–99, 2015. 2, 3

[9] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 1, 2

[10] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008. 9, 10

[11] Yanan Wang, Shengcai Liao, and Ling Shao. Surpassing real-world source training data: Random 3d characters for generalizable person re-identification. In Proceedings of the 28th ACM International Conference on Multimedia, pages 3422–3430, 2020. 1, 5

[12] Tianyu Zhang, Lingxi Xie, Longhui Wei, Zijie Zhuang, Yongfei Zhang, Bo Li, and Qi Tian. Unrealperson: An adaptive pipeline towards costless person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11506–11515, 2021. 1, 5