Multiview Human Synthesis From a Singleview

Si Wen (06246679), Tiancong Zhou (06247022), Honghao Qiu (06246258)
{wensi, longztc, honq}@stanford.edu

Abstract – We use deep generative models to synthesize multiview images given a single view. The generation process is done in two stages: in the first stage, we train a variational auto-encoder (VAE) [10] to synthesize a new view of the input image; in the second stage, we use a generative adversarial network (GAN) [5] to generate details on the output of the first stage. We evaluate our results using both qualitative and quantitative methods. One potential application is generating multiview images for e-Commerce products.

I. Introduction

Multiview synthesis has been a long standing problem in computer vision. Traditionally, this problem had been attempted using geometry-based approaches. That is, a 3D model is first reconstructed from the input image, and the novel view is generated from that 3D model. However, with recent progress in deep learning, more and more people have attempted to synthesize novel views directly use deep neural networks [25, 22, 27, 28, 6, 11, 16], and have achieved great results.

Our project limits the scope of the multiview synthesis problem to human images, with a focus on the clothing items in these images. Given the recent progress in deep generative models and their amazing results on image synthesize and style transfer [15, 8, 29, 9], we try to tackle this problem using similar approaches. Our goal is to synthesize images of a particular person from any view given a single input view. If successful, this could enable many useful applications in fashion/E-Commerce websites and in the field of photo/video edition and content generation. For example, in Amazon cloth stores we can help provide 360-degree rotation view for customers given a single front-view image taken for the model.

To achieve this goal, we use images of people with different dress in different angles as input data, with both real world multiview fashion dataset from MVC [12] and synthetic human images rendered in 360-degree views. Then we use deep neural networks to generate output images from the inputs in a two-stage process. First, the input image (human body in a specific angle) and the value of target angle (e.g. 90 degree) are fed into a VAE to generate a coarse image output in the target angle, and then this coarse image along with the original input image are both fed into a GAN to generate a fine image in the target angle. Figure 1 further illustrates the idea of this process with a specific example.

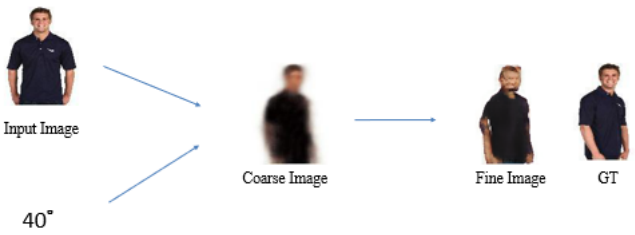

Figure 1: Condition input image is firstly inputed into VAE to generate coarse view image for target angle, then it is inputed into GAN to generate fine view image for target angle.

II. Related Work

A. Geometry-based Multiview Synthesis

A large amount of work has been put into the multiview synthesis problem. Geometry-based approaches are widely used in the past. These approaches either implicitly or explicitly model the geometry of the object in interest, and synthesize novel views from the modeled geometry. Often, these approaches are used when multiple views [4] of the object exist as input (e.g. stereo), so visual correspondence can be found and used as constraints to the algorithm. In the case of a single input view, either manual annotation [3] or additional information [20] is needed for these algorithms to perform well.

B. Deep Learning Approaches

With the recent progress in deep learning, people have started to solve the multiview synthesis problem using deep neural networks. The ShapeNet dataset [2] indexed over 3 million 3D models cross over 3,000 categories, and its corresponding competition has attracted many submissions (e.g. [28, 22, 16]) that uses deep learning to synthesis novel views.

At the same time, there has been an explosion of interest in deep generative models in the past couple years. These models are capable of learning the latent representation of their input dataset and generate new images with interesting variations. These approaches can be grouped into 3 main categories:

- Autoregressive (e.g. PixelRNN [15]): this approach uses a neural network to model the conditional distribution of every pixel in the image given previous pixels (e.g. the pixel to the left or to the top). Its result is very good but it is also extremely computationally intensive given large images.
- Variational Auto-encoder (VAE) [10]: this approach uses a neural network to model the distribution of the input data and tries to maximize the lower bound on its log likelihood.
- Generative Adversarial Network (GAN) [5, 17]: this approach doesn't model the distribution directly but uses adversarial training to reproduce the data distribution.

These methods (particularly GAN) have also been extended to generate new images based on certain types of conditions [14]. For example, Yan et al [24] used conditional generative adversarial network (cGAN) to change facial expression of synthesized images by interpolating specific latent variables and Zhang el al [26] used conditional variational auto-encoder to generate an new image of a person at a particular age. People have also used GANs [8, 29] to perform style transfer tasks with great results.

C. Fashion/Clothing Multiview Synthesis

Even in the space of fashion/clothing image synthesis, there has been a number of approaches using deep generative models. Zhu et al [30] used GAN to synthesize images of people wearing different types of cloth given a source image and a verbal description (e.g. red shirt with blue striped skirt). Zhao et al [27] proposed a VAE+GAN model to synthesize the side and back view of a person given the frontal view, using images from the MVC dataset [12] and DeepFashion dataset [13]. This approach has the strength of combining the ability of VAE to find global appearance/outline and the ability of GAN to fill in fine details. We adopt a similar approach to generate multiview images and extend it to work for any of the 360-degree views.

III. Dataset and Features

Deep learning algorithms require large amounts of training data for the network to learn the latent representation. We used a combination of images of real human from the MVC [12] dataset (160,000 images from over 30,000 clothings items) and images of rendered 3d-models that we generated ourselves. Image data are labeled with the associated angles of the view (e.g. 60 degree). Since the MVC dataset contains images only of front, side, and back views, we label the views as 0, 90, 180, 270 degree angles. We split each dataset into 80/10/10 train/dev/test groups.

We generate the synthetic dataset because the MVC dataset is very limited in its number of views (only front, back, and side) and it's insufficient to learn a model capable of generating 360-degree views from these 4 views (the DeepFashion dataset [13] is more limited in that both left and right side views are labeled simply as "side" view). We also can't use ShapeNet [2] models since it doesn't contain any human models. We try to overcome these limitations by generating 360-degree synthetic images using 3D modeling softwares. Fortunately, Adobe Fuse is a program that allows its users to quickly generate many models with good variation on their body attributes (skin tone, gender, body size, etc) and clothing.

Our synthetic dataset (about 100,000 full-body human images in 360-degree views) is generated using the following approach:

i Create 3D character models in Adobe Fuse. Adobe Fuse provides plenty of hairstyles, faces, cloth and shoes to combine and create 3D characters;
ii Export the model as .obj file and import it into Blender (Blender is a 3D graphics software that could be easily programmed using Python script);
iii We write a Python script to automatically render an image for each of the 360 degree views in Blender. We also vary the camera/lighting placement for variations.

Figure 2 contains some examples of the synthetic images we generated.

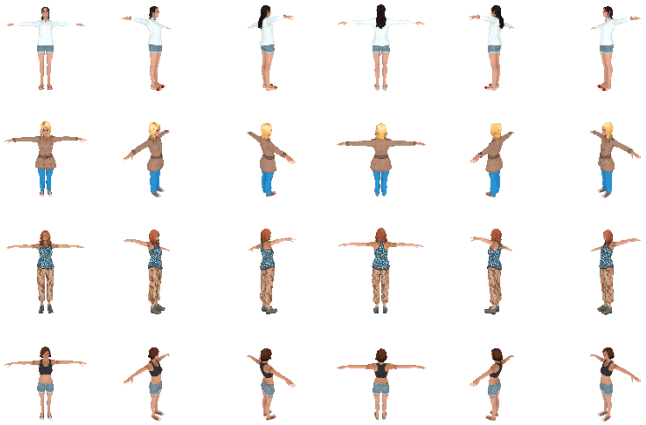

Figure 2: Synthetic Multiview Human Image Samples

We use a number of standard data augmentation techniques to preprocess our input data. We vary the lighting and camera position in Blender. We apply small random crops to the images, and randomly apply light Gaussian blur. Note that we do not perform random horizontal flips as we already have an image at the resulting view.

Since we use batch normalization [7] for all layers, we do not normalize our input image. However, we do normalize

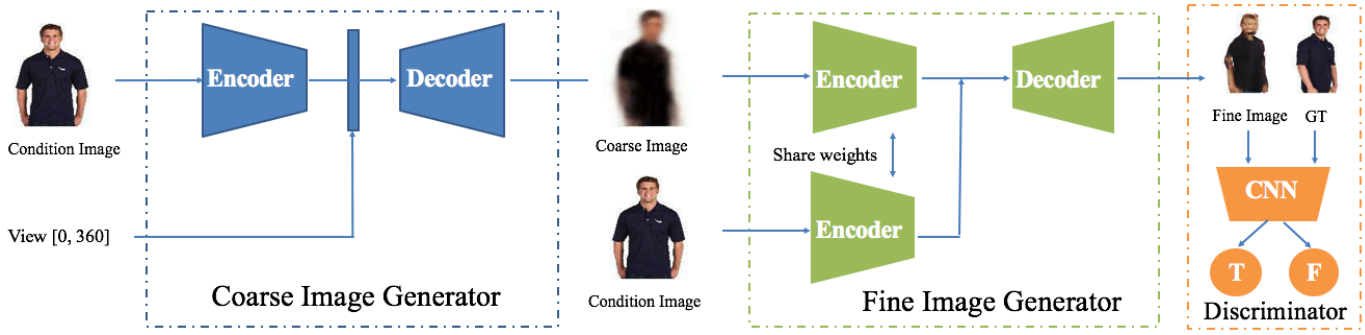

Figure 3: Pipeline Overview

the value of the target angle to [-0.5, 0.5] since it gets concatenated with the latent variables.

Since neural networks take a long time to train, all initial experiments are done using images of size 64x64. The final results are trained on images of 128x128 resolution. We notice significant improvements on the quality of our results when trained using higher resolution images, and we believe this improvement will continue as the resolution goes higher.

IV. Methods

A. Overview

We tackle this problem using deep generative models. We divide our pipeline into two stages: the first stage consists of an input image, a target angle (0 to 360 degrees), and a conditional VAE. The encoder transforms the input image into a latent variable, concatenate it with the target angle, and passes them through the decoder. The decoded image should resemble the input image in the target angle, with noticeable artifact (e.g. blurry, sometimes with incorrect colors). The second stage consists of the original input image, output of the first stage (coarse image), and a conditional GAN. Both images are fed into a hierarchical feature extractor to obtain their latent representations. These latent representations are then concatenated and fed into a generator to obtain the final output. The generator is able to extract low level patterns from the original input image and generate a new image that conforms to the structure of the coarse image with patterns from the original input image. Our network is written in Python using the Tensorflow framework. We trained our models using NVIDIA Tesla GPUs.

The complete pipeline is depicted in Figure 3.

B. Stage 1: cVAE

We implemented the VAE from scratch, going from a simple fully-connected network with two hidden layers to a deep convolutional network consisting of 13 layers. Figure 4 shows the results of some of our intermediate work.

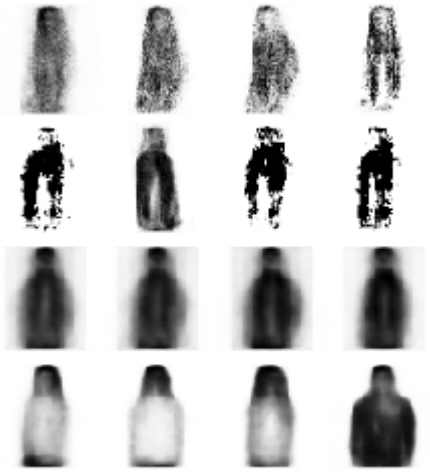

Figure 4: Row 1-4: output of VAEs with (1) fully-connected layers only with insufficient amount of hidden neurons (2) fully-connected layers only with sufficient hidden neurons (3) convolution layers with insufficient amount of filters (4) convolution layers with sufficient amount of filters.

Notice how the number of neurons is extremely important in capturing variation, and that convolution layers are much better at hierarchical representations.

The input is 128x128 pixels. The encoder consists of 6 convolution layers all with 3x3 kernels. The layers have 16, 16, 32, 32, 64, 1 filters respectively. We downsample the input twice with strided-convolutions. In the last layer, we flatten the result into a 1024-dimensional vector and pass it through a fully connected hidden layer to generate parameters to the learned distribution using the parameterization trick [10]. We use these parameters to sample our latent variables from a 400-dimensional Gaussian distribution and concatenate them with the normalized target angle. They then goes through a decoder that has a similar structure but in reverse, with two upsamples in the middle.

The loss function is defined as follows:

$$L(\theta; x, c) = -KL(q_\theta(z|x,c)||p(z|c)) + E_{q_\theta(z|x,c)}[log(p_\theta(x|z,c))] \quad (1)$$

C. Stage 2: cGAN

We attempted to implement a GAN from scratch. However, GAN is notoriously known for its instability and its difficulty to train. Our experience confirms this knowledge. Although the GAN community has proposed many techniques to make the training process easier [18, 1, 9], we decided not to implement one from scratch, but to build on top of an existing implementation [1]. There are several features of this implementation that we find desirable: it uses U-Net, which allows for better generation quality; it uses convolution transpose (fractionally-strided convolutions) instead of bilinear interpolation for upsampling; it uses Leaky ReLU instead of ReLU as activation functions. This implementation worked well without much hyperparameter tuning.

We modified the above implementation so that it takes two 128x128 sized images, passed them through an encoder with shared weights, concatenated their respective latent representations, and finally passed them through the decoder. We did not make any changes to the discriminator. The final generator is a 7-layer U-Net with 64, 128, 256, 512, 512, 512, 512 filters in each of the layers.

V. Experimentation Results

We use both quantitative and qualitative measures to assess the result of our work.

A. Qualitative results

The results of our final model is compared with other approaches and the ground truth (see Figure 5 and Figure 6). Visually, our results have a more complete global appearance as well as more local details (note how the output of cGAN often loses patches of the body). At the same time, our result looks much sharper than the output of cVAE.

B. Quantitative results

Figure 7 summarizes the result of our experimentations.

For quantitative evaluation methods, we use Mean Square Errors (MSE), Structural Similarity Index (SSIM), and the Inception Score (IS) to compare the generated fine image output to target image, as a judgment for our model quality.

[1] https://github.com/affinelayer/pix2pix-tensorflow

Figure 5: Sample images trained and tested using real images in the MVC dataset. Left to right: input image, output from cVAE, output from cGAN, our from our model, ground truth.

Figure 6: Sample images trained and tested using synthetic images. Left to right: input image, output from cVAE, output from cGAN, our from our model, ground truth.

- Structural Similarity Index (SSIM) [23]: SSIM measures the similarity between two images. It's a better metric than the pixelwise error (MSE) as it insensitive to things like lighting conditions and small variation in poses. It's been used in other works on multiview synthesis [16, 27]. It's formally defined as:

$$SSIM(I_x, I_y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

- Inception Score (IS): the Inception Score is adopted from Salimans' work on GANs [18] and is used to measure the quality of different generative models. The idea is that images that contain meaningful objects should have a conditional label distribution with low entropy while a model that generates varied images should have high entropy. The Inception Score is formally defined as:

$$IS(I_x, y) = exp(E_{I_x} D_{KL(p(y|I_x)||p(y))})$$

where I_x is a generated image and y is the label of that image predicted by the Inception model [21].

Methods	SSIM		IS		MSE	
	MVC	Synthetic	MVC	Synthetic	MVC	Synthetic
cVAE Only	0.59 ± 0.10	0.66 ± 0.10	1.67 ± 0.30	2.30 ± 0.44	7.49 ± 0.40	6.95 ± 1.50
cGAN Only	0.63 ± 0.09	0.67 ± 0.13	2.62 ± 0.38	2.34 ± 0.03	6.70 ± 0.58	6.52 ± 1.16
Ours	0.66 ± 0.10	0.68 ± 0.10	2.35 ± 0.45	2.33 ± 0.04	6.62 ± 0.69	6.22 ± 1.02

Figure 7: Experimental Results

For model comparison, we experimented the following 3 models on MVC and our synthetic dataset respectively, and compare their SSIM, IS, MSE measures:

- Using only cVAE.
- Using only cGAN.
- Using our VAE+GAN pipeline.

From the result metrics in the table, we can see that the following trends:

- SSIM obtained by our approach is larger than that of obtained by using VAE or GAN alone, which means our generated output image is more similar to the ground truth;
- IS score for our approach is larger than that of using VAE only, but smaller than IS score using GAN only, this is expected since GAN model alone allows more variation in the output images.
- MSE analysis shows that our MSE is smaller than that of using VAE or GAN alone, this resonates with the result of SSIM and proves that our approach in general has better performance generating images closer to target.
- In general, our model performs better on synthetic data than on real world data, this is because synthetic data is smoother, contains less details, and has similar pose, so the generation is relatively easier.

VI. Future Work

Due to time constraints, there are many things on our roadmap that we haven't gotten to. Some of these things include:

- Improve the GAN: we only used a simple conditional GAN with very little hyperparameter tuning. There is a number of recently published papers that shows various techniques to improve the quality and stability of GANs.
- Improve face synthesis: our model currently handles face poorly since it contains a lot of noticeable details. It is important to be able to recreate these details in a convincing way for good result. There are several promising attempts [25, 6] that focuses solely on facial synthesis that we can follow up on.
- Image background: our current dataset only contains images without any background. It is more difficult to synthesize views when the object of interest is not presented in isolation and we would like to tackle that problem in the future.
- Transfer learning: we hope to use the model trained from our synthetic dataset on real images. Experiments on transfer learning [19] have shown great results for discriminative models. We hope to achieve similar results for generative models.

VII. Conclusion

We use a deep learning based approach to solve the multiview synthesis problem. We use a multi-stage generative model to synthesize a novel view of the input image at a given angle. We performed experiments using real and synthetic datasets and achieved promising results using both qualitative and quantitative evaluation methods.

VIII. Teammate Contributions:

Our teamwork break down:

- Si Wen: Si mainly worked on modeling, including VAE and variants of GAN modeling. He also contributed to the final report.
- Tiancong Zhou: Tiancong mainly worked on the generation of our synthetic dataset and quantitative evaluation of our results.
- Honghao Qiu: Honghao mainly worked on variant of GAN modeling and reporting. He is the main contributor to the reports.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein gan". In: *arXiv preprint arXiv:1701.07875* (2017).

[2] Angel X Chang et al. "Shapenet: An information-rich 3d model repository". In: *arXiv preprint arXiv:1512.03012* (2015).

[3] Tao Chen et al. "3-sweep: Extracting editable objects from a single photo". In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), p. 195.

[4] Yasutaka Furukawa, Carlos Hernández, et al. "Multi-view stereo: A tutorial". In: *Foundations and Trends® in Computer Graphics and Vision* 9.1-2 (2015), pp. 1–148.

[5] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[6] Rui Huang et al. "Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis". In: *arXiv preprint arXiv:1704.04086* (2017).

[7] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International Conference on Machine Learning*. 2015, pp. 448–456.

[8] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *arXiv preprint arXiv:1611.07004* (2016).

[9] Tero Karras et al. "Progressive growing of gans for improved quality, stability, and variation". In: *arXiv preprint arXiv:1710.10196* (2017).

[10] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[11] Alex R Kuefler. "Deep View Morphing". In: *CS 231n, Stanford* (2017).

[12] Kuan-Hsien Liu, Ting-Yen Chen, and Chu-Song Chen. "Mvc: A dataset for view-invariant clothing retrieval and attribute prediction". In: *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM. 2016, pp. 313–316.

[13] Ziwei Liu et al. "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1096–1104.

[14] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[15] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks". In: *arXiv preprint arXiv:1601.06759* (2016).

[16] Eunbyung Park et al. "Transformation-grounded image generation network for novel 3d view synthesis". In: *arXiv preprint arXiv:1703.02921* (2017).

[17] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[18] Tim Salimans et al. "Improved techniques for training gans". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2234–2242.

[19] Ashish Shrivastava et al. "Learning from simulated and unsupervised images through adversarial training". In: *arXiv preprint arXiv:1612.07828* (2016).

[20] Hao Su et al. "3D-assisted image feature synthesis for novel views of an object". In: *arXiv preprint arXiv:1412.0003* (2014).

[21] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[22] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. "Multi-view 3d models from single images with a convolutional network". In: *European Conference on Computer Vision*. Springer. 2016, pp. 322–337.

[23] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[24] Xinchen Yan et al. "Attribute2image: Conditional image generation from visual attributes". In: *European Conference on Computer Vision*. Springer. 2016, pp. 776–791.

[25] Junho Yim et al. "Rotating your face using multi-task deep neural network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 676–684.

[26] Zhifei Zhang, Yang Song, and Hairong Qi. "Age Progression/Regression by Conditional Adversarial Autoencoder". In: *arXiv preprint arXiv:1702.08423* (2017).

[27] Bo Zhao et al. "Multi-View Image Generation from a Single-View". In: *arXiv preprint arXiv:1704.04886* (2017).

[28] Tinghui Zhou et al. "View synthesis by appearance flow". In: *European Conference on Computer Vision*. Springer. 2016, pp. 286–301.

[29] Jun-Yan Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *arXiv preprint arXiv:1703.10593* (2017).

[30] Shizhan Zhu et al. "Be Your Own Prada: Fashion Synthesis with Structural Coherence". In: *arXiv preprint arXiv:1710.07346* (2017).