Abstract

This paper aims to solve the video motion retargeting problem, where we are given a source video and target video, and we want to control the target person’s motion by the source person. While previous methods all require a minutes-long source video, we solve this problem with only one source frame, which greatly relaxes the data constraint of performing motion retargeting. To the best of our knowledge, our work is the first to approach motion retargeting with a single source frame. Experimental results demonstrate that our model is able to achieve the retargeting on the intermediate semantic maps, which is promising for future research on the realistic frames.

1. Introduction

Motion retargeting has been receiving a lot of attention in the past few years [3, 14, 1]. In this setting, we have a source video and a target video, where there are two persons performing different actions. And the goal is to control the target person’s action by the source person, while keeping all other aspects such as the identity, appearance, background, etc. to be unchanged, as illustrated in Fig. 1. Current state-of-the-art methods solve this problem by training a personalized model for the target person to be controlled with minutes-long videos [14, 3]. This makes these models very hard to be applied to new identities, because each time it requires gathering long videos of the new identities, and also re-train the entire model on these videos. However, we argue that all the necessary information of the target video, including appearance, identity and background, can be obtained from one single target frame. In this paper, we propose a framework that learns a generalized model that maps a source driving video and a target frame to the controlled target video. To the best of our knowledge, this is the first work that explores this idea. Limited by the time scope of this project, we started with motion retargeting between intermediate semantic maps, rather than directly manipulating the realistic frames. So far, we have successfully achieved motion retargeting on the semantic maps, and will extend our approach to the realistic video in the future.

Since our goal is to learn a generalized model for all identity and all possible background, we cannot train the entire model on only one video as the strategy used in [3], because in this way, the model will not generalize to other identities and background at all. So a key problem comes out that what supervision should we provide to our model. One natural solution is to find videos of different person dancing to the same song (or performing the exact same action) and align them temporally, then use the source frame at each time step as the corresponding supervision. However, it is very hard to find videos that person doing exactly the same action and also perform precise temporal alignment. In order to tackle this issue, we propose a self-supervised training procedure, which leverages frame pairs inside one video that are apart temporally to supervise the training. This turns out to work very well and our model can successfully transfer the motion.

In summary, our main contributions are as follows:
we are the first to solve the video motion retargeting problem with only one target frame, which greatly relaxes the data constraints;

• we propose a generalized motion retargeting model with a self-supervised training scheme to enable this single-frame motion retargeting;

• experimental results show that our model can successfully generate source-controlled target semantic maps, which paves way for future work on the realistic video.

2. Related Works

2.1. Motion Retargeting

Recently, there has been few works that look into the problem of pose or motion retargeting. Among the most prominent motion retargeting works, Everybody Dance now [3] is able to perform well to produce high quality videos of target person dancing in the desired motion. However, they are able to do the task only in lab settings and require huge amounts of video data for training a person specific GAN. [14] also propose a motion retargeting network but requires they require multiple frames to generate the target video. In contrast to these works, we aim to use just a single image to generate the dancing video for the target person.

2.2. Pose Transfer

In a similar vein as motion retargeting, there has been several attempts to transfer poses between pairs of still images. The PG$^2$ network by Ma et al. [9] has been prominent in the area that inspired several followups that aim to improve upon the results. Adversarial techniques have been proposed to formulate pose transfer as a minimax game [10] [11] [13] [12]. Perceptual Losses to improve the generated image quality has also been proposed by [11]. Other techniques like Multiscale architecture for coarse-fine refinement [10] [11] and foreground background separation [10] have also been shown to improve performance. Notably, disentangled modelling of different human parts and limbs have been shown also to be shown very useful [10] [13].

3. Data Description

Since there is currently no publicly available dance2dance dataset, we collected our own dataset from Youtube. Specifically, we collected 10 dance videos from Youtube, and at 30 / 60 fps only. Also, all videos are chosen to be: (1) Without fast motion or movement to avoid motion blur; (2) Show the full view of the person who is dancing; (3) With minimum camera motion (4) With a static background, and the background is chosen to be as uniform as possible. In total we collected around 86000 frames from all 10 videos.

After we get the raw videos, we ran an off-the-shelf state-of-the-art semantic map extractor [4] to get the semantic maps. Concretely, a semantic map identifies different parts of the body (arms / legs / torso / face / feet, etc). We used semantic maps as an intermediate representation because this is easier for the model to learn a mapping for, and also provides rich information for latter conversion to realistic frames (because the semantic maps provide the body part information). These semantic maps are later used as ground-truth output of the generator, as stated later. We show several frame samples and their corresponding semantic map in Fig. 2.

4. Method

The overall architecture is illustrated in Fig. 3. Basically, the entire framework consists of a pose detector followed by a conditional GAN. Specifically, given a training video $V = \{f_1, f_2, ..., f_t, ..., f_N\}$ (where $N$ is the number of frames in the video), we take a frame at time step $t+1$ as $f_{t+1}$, and pass it through the OpenPose detector [2], after which we get the corresponding pose stick represented by a set of joint coordinates, denoted as $p_{t+1}$. Then we pass $p_{t+1}$ and the frame at $t$, $f_t$ (which is later extracted into frame feature) together into a conditional generator $G$. Concretely, the generator takes $f_t$ as input and is conditioned on $p_{t+1}$. The generator $G$ generates a fake intermediate semantic map $s'_{t+1}$, which is then sent into the discriminator $D$ alongside the ground-truth semantic map $s_{t+1}$ (this is obtained from the state-of-the-art semantic map extractor [4]). In the following sub-sections, we will go through each part in detail.
4.1. OpenPose Pose Extractor

The first part of our pipeline is the off-the-shelf pose detector OpenPose [2]. The goal of this step is to get a set of joint coordinates denoting the pose given a frame of the person. This pose is generally quite noisy however, provide enough supervision in case of our camera setting (upfront, fixed with a static background). Of course, there are still missed detections but since we are using the pose to map to the mid level representation and not to the final image, these missing detections can be taken care of in generator by implicitly using continuity priors learnt over the body. So for example, if a joint is missing in the lib, it can be implicitly filled in by the generator using information from the mirror of that joint in the other limb.

4.2. Objective

As is conventional, The objective of a conditional GAN is formulated as a minimax game between $G$ and $D$. It is constrained such that when $D$ starts dominating, the losses for $G$ pile up and the other way around. Mathematically, can be written as:

$$L_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x,G(x,z)))]$$

Where $G$ tries to minimize this objective against an adversarial $D$ that tries to maximize it, i.e. $G^* = \arg\min_G \max_D L_{cGAN}(G,D)$.

4.3. Conditional Generator

In order to generate the realistic frame (or the intermediate semantic map) at time step $t + 1$, we need to provide the corresponding frame (or the intermediate semantic map) at time step $t$. But since we want the generated output to also be controlled by the target, we use a conditional generator in this case, which is conditioned on the joint coordinates obtained from last step.

The architecture of the conditional generator is shown in Fig. 4, which is an encoder-decoder structured network with skip connections. In such a network, the input is passed through a series of layers that progressively downsample, until a bottleneck layer, at which point the process is reversed. Such a network requires that all information flow pass through all the layers, including the bottleneck. For many image translation problems, there is a great deal of low-level information shared between the input and output, and it would be desirable to shuttle this information directly across the net.

To give the generator a means to circumvent the bottleneck for information like this, we added skip connections, following the general shape of the U-Net [12]. Specifically, we add skip connections between each layer $i$ and layer $n - i$, where $n$ is the total number of layers. Each skip connection simply concatenates all channels at layer $i$ with those at layer $n - i$. 

4323
As stated before, different from pix2pix [6], which uses a non-conditional generator, our generator needs to be condition on the pose from time step \( t + 1 \). Concretely, we modified the generator’s architecture to make it conditional, which results in our conditional generator.

### 4.4. Conditional Discriminator

It is well known that the L2 loss, as well as L1, produce blurry results on image generation tasks [8]. Although these losses fail to encourage high-frequency crispness, in many cases they nonetheless accurately capture the low frequencies. For problems where this is the case, we do not need an entirely new framework to enforce correctness at the low frequencies. L1 will already do.

This motivates restricting the GAN discriminator to only model high-frequency structure, relying on an L1 term to force low-frequency correctness. In order to model high-frequencies, it is sufficient to restrict our attention to the structure in local image patches. Therefore, we use PatchGAN [6] that only penalizes structure at the scale of patches. This discriminator tries to classify if each \( N \times N \) patch in an image is real or fake. We run this discriminator convolutionally across the image, averaging all responses to provide the ultimate output of \( D \).

Such a discriminator effectively models the image as a Markov random field, assuming independence between pixels separated by more than a patch diameter.

### 4.5. Training Strategy

One of our main contributions lies in the self-supervised training scheme, which solves the problem of lack of supervision when training a generalized model. In this subsection, we will explain our training strategy in detail.

Specifically, we first sample a consecutive frame pair (the time difference between the two frames is 1 frame). We do this because later in our video generation step, we generate the video frame-by-frame in an iterative manner. In order to simulate the testing condition, here we let the two sampled frames to be one frame apart from each other. Our goal is to generate the frame at time step \( t + 1 \) given the frame at \( t \) and pose stick from \( t + 1 \). This way, the ground-truth output becomes the frame at \( t + 1 \) while the input is the frame at \( t \).

### 5. Experimental Procedure

In this section, we briefly describe our training procedure and experimental details mentioned in the previous sections.

#### 5.1. Training

As discussed in the previous section, we train a pix2pix GAN with the triplets of training data described by (image of target person at time \( t \), pose of target person at time \( t + 1 \) and the mid level representation of the target person at time \( t + 1 \)). Referring to the Figure 5, the first two images along with the last image forms this triplet of training data. Note that since we use the pose of the target person while training, we have the ground truth image of the target person in our database.

#### 5.2. Optimization

To optimize our networks, we follow the standard approach from [6]: we alternate between one gradient descent step on \( D \), then one step on \( G \). As suggested in the original GAN paper [5], rather than training \( G \) to minimize \( \log(1 - D(x, G(x, z))) \), we instead train to maximize \( \log D(x, G(x, z)) \). In addition, we divide the objective by 2 while optimizing \( D \), which slows down the rate at which \( D \) learns relative to \( G \). We use minibatch SGD and apply the Adam solver, with a learning rate of 0.0002, and momentum parameters \( \beta_1 = 0.5 \), \( \beta_2 = 0.999 \).

#### 5.3. Inference

While inference with start with time \( t = 0 \) image of the person and use the pose of the source at time \( t = 1 \) as the input into the network. Since the network is trained person independently, it can learn to use this pose to transfer to the person and also implicitly do the semantic segmentation of the person. This generated \( t = 1 \) image is then used along with the pose of the source at \( t = 2 \), to generate the mid level semantic segmentation representation of the target at \( t = 2 \) and so and so forth. This recurrent cycle is unrolled until the entire dance can be retargeted.

#### 5.4. Results

Figure 5 show the images from our method on image based pose transfer. As observed, the fake GAN generated semantic segmentation map are very close to the real maps showing that not only the network can learn to transfer poses it can also map from real images to semantic segmentation body maps efficiently. This allows us to train the model in an end to end fashion thereby improving performance compared to using a pretrained frozen body semantic segmentation method like [4].

#### 5.5. Discussion

From the experimental results, our model can achieve motion retargeting on frame-level semantic maps and the results are pretty good. However, we can see that the colors in different regions of the generated semantic maps sometimes can be very different from the ground-truth semantic maps. This is because the input to our generator is the realistic images rather than the semantic maps from the previous time step. In other words, our generator is doing both semantic
map extraction and motion retargeting simultaneously. And because we don’t provide the color information to our generator, it is reasonable that the generator cannot figure the corresponding color scheme out. The reason why we want to train our generator to do both tasks at once lies in that we can train the entire model in an end-to-end way, thus bringing about possible performance boost. On the other hand, this also implies that our model works pretty well even the task is much harder.

6. Conclusion and Future Work

In this paper, we proposed a framework to tackle the video motion retargeting problem from only one source frame. Specifically, we introduced a decoder-encoder structured conditional GAN-based model to solve this task. Furthermore, we came up with a self-supervised training strategy to learn the model. Experimental results show that our model can do the motion retargeting on the intermediate semantic maps very well. However, our work is far from complete, and several future steps include:

- further render the realistic image out based on the intermediate semantic maps
- apply our method iteratively for every time step to get all the necessary frames
- (possibly) apply temporal smoothing to render a better-looking video output

7. Acknowledgements

We thank the instructors Maneesh Agrawala, Ohad Fried and Michael Zollhofer for the wonderful learning experience and all the work you have devoted for the entire quarter. We also want to thank all our CS448V classmates for very helpful and interesting in-class discussion. Particularly, we want to thank Michael Zollhofer for helpful discussion on our project. Thank you all again for this wonderful class!

References


