Img2Motion: Learning to Drive 3D Avatars using Videos

Junying Wang, Weikai Chen and Hao Li

October 11, 2019

Img2Motion: Learning to Drive 3D Avatars using Videos

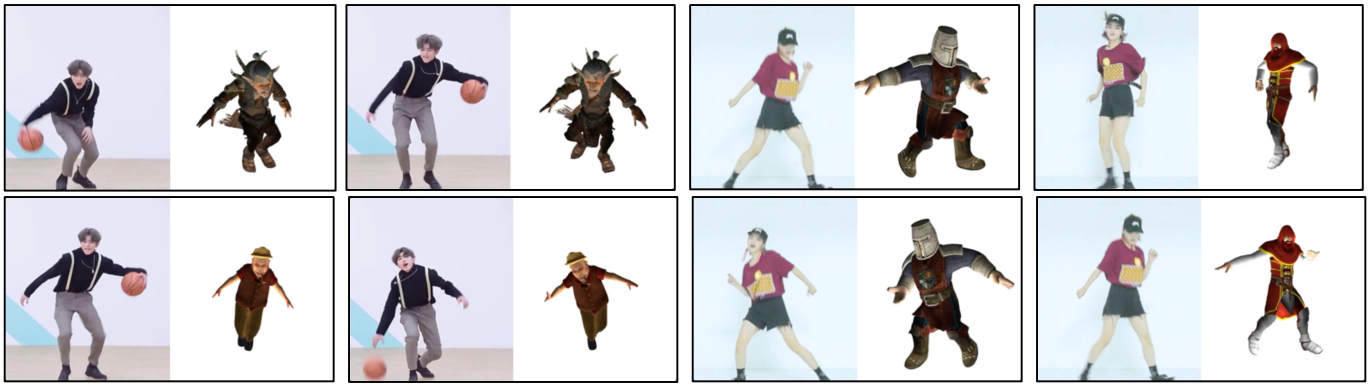

Figure 1: ***Let's create a dance!*** The above pictures show the final outputs of motion retargeting from two-dimensional chromatic videos to three-dimensional digital avatars. For each picture, the left side is the input origial video, based on our system, the right side 3D avatar can generate the same ainmation as the left side 2D character.

Abstract

This paper presents a novel neural network motion retargeting system that drives 3D rigged digital human avatars using videos. We study the problem of building a motion mapping between 2D video and 3D skeletons, in which the source characters can drive the target subjects with varying skeleton structures. In particular, the target 3D avatars may have different kinematic characteristics, e.g. bone lengths, skeleton scales, skeleton topologies, etc. The traditional motion retargeting is between pair to pair characters, especially 2D characters to 2D characters and 3D characters to 3D characters. There is a digital gap of using 2D characters' animations to drive 3D rigged characters. These traditional techniques may not yet be capable of solving motion retargeting from 2D motions to 3D digital human avatars with sparse skeleton motion data. Inspired by these unsolved limitations, we present a pipeline of building neural network motion retargeting system, which can do motion retargeting from 2D videos to 3D rigged digital human avatars. This whole system with the effective pipeline can be used for game implementations, virtual reality system and also can generate a more comprehensive dataset with larger varieties of human poses by animating existing rigged human models.

Keywords—motion retargeting, digital human, 3D pose estimation.

I. Introduction

Motion comes from vision and also beyond vision. In order to animate rigged 3D models, the basic idea is changing the rigging parameters through the whole animation path and generate continuous movement. Generally, there are two common methods to animate 3D avatars: one is keyframe animation, the other method is motion capture. For keyframe animation, traditionally, it totally relies on animators setting the extreme poses of rigged models as keyframes and do interpolations between each keyframes based on certain computer animation software. Though this method can give artists total control, it is an intensive way to drive 3D models which is accompanied by amounts of tedious work. Motion capture is the other technique that is used for capturing more realistic motion sequences. It can extract motion data from real-world people's actions and make a trade-off between motion accuracy and animation effectiveness. And these amounts of variable motion capture files can be used to do motion retargeting. Recently, in order to simplify the animation process and fully take advantage of the comprehensive motion data, there are some related research works about motion transfers, which build neural networks that can reduce the manual animating part and drive the characters in an efficient way. Ruben Villegas[1] presents a neural kinematic networks for unsupervised motion retargeting, which can avoid manual retargeting process and transfer 3D motions from one character to other characters. Jongmin Kim[2] also uses a deep learning approach to solve motion retargeting problem, which mainly focus on solving the motion transfer problem when body sizes and proportion (e.g, arms, legs, torso, and so on) are different. All these research work mentioned above are using training data from motion capture dataset, which already contains complete and comprehensive 3D information, including joint positions and rotations frame by frame. There are also some approaches aiming to do 2D video motion transfer, which build a mapping from 2D video to 2D video that contain human motions. Caroline Chan[3] present an approach that using 2D pose as an intermediate and achieve a video-to-video translation. Kfir Aberman[4] present a new method for retargeting video-captured motion between different human performers with different motion, skeleton, and camera view-angle. These motion retargeting method mainly focus on either 2D-to-2D motion transfer or 3D-to-3D motion retargeting, which means that there is still a gap of motion transfer between 2D video to 3D human avatars.

Specifically, motion transfer aims to be achieved in the same dimensional space, especially two-dimensional space to two-dimensional space or three-dimensional space to three-dimensional space. In this paper, we build an advanced pipeline of neural network motion retargeting system that solves the problem of how to use 2D videos to drive 3D digital avatars. Due to the limitations of 3D motion data sources, we try to use 2D videos as input which is a more general and convenient way to animate these 3D rigged avatars. The whole pipeline can be divided into two individual parts. The first part solves the problem of how to learn from 2D videos that make accurate predictions of 3D motions based on neural network, which belongs to the 3D motion reconstruction category. The second part bridges the tremendous digital gap of how to build a decent mapping between sparse predicted 3D motion data and 3D digital human avatars. This approach is not limited to do motion transfer between human to human, it can also be applied to do retargeting between humans to animals, like cats or dogs, which should have the same skeleton topologies.

With the whole process of our system, the 3D rigged avatars can be driven by 2D wild video motions automatically without setting the joint positions and angles manually. And it can be fully implemented in game industry and brings a new light into digital human avatar animation that can be used for virtual reality environment.

The contributions of our work can be summarized into three parts as follows:

- Estimate the 2D key points from chromatic 2D videos and then obtain the 3D poses by lifting 2D key points to 3D using deep learning approaches.
- Find a highly accurate mapping to retarget the predicted motions to 3D avatars with arbitrary rigging representations.
- After driving these animated 3D avatars successfully, these 3D avatars can generate various poses that can build a recurrent dataset for future training implementation.

II. IMPLEMENTATION

A. *Motivation*

Have you ever played the game - Just Dance. In this game, if you use Nintendo Switch, you need to follow the 2D characters and use the wireless controllers to finish some certain poses, which means you need to follow the 2D characters to do as they do. However, how about let the digital avatars follow our motions and do as I do? We don't need to just dance, let's create a dance!

Figure 2: This is the one of the popular games: ***Just Dance***

Highly accurate motion detections rely on multiple wearable devices, which may bring huge economic burden for personal daily use. Variable game equipment, especially virtual reality handlers, may rely on kinds of wireless controllers to detect players' motions, and use these motions to give vivid visual feedback to the players. Actually this whole process can be simplified with single camera capturing or only using one shot video as the input data. In order to get rid of these tedious equipment demands, we figure out that we can use players' chromatic videos as input and generate the intuitive three dimensional motions to manipulate the human avatars. This process motives us to give the users that in game or virtual reality environment a more effective way to acquire interactive and immersive experience.

The other motivation comes from the dataset preparation. These are many static human avatar dataset, including various rigged human avatars. For data preparation, we use RenderPeople dataset and Mixamo dataset as the original dirven data. If we can find an efficient way to drive these static models, we can acquire large amount of poses, which can compose a larger novel motion dataset, that can be used as recycle data for future training resources. Therefore, the human capture dataset can be significantly enhanced with larger variety of poses by adopting motions from videos in the wild.

B. *Pipelines*

The whole pipeline of building a neural network motion retargeting system can be decomposed into two separate parts. Firstly, we have accomplished an accurate prediction of 3D human motions based on 2D videos. Secondly, we apply the predicted 3D motions to digital human avatars that have different kinematic characteristics.

1. From 2D videos to extract 3D poses

The first part of the pipeline focus on how to do human 3D pose estimation based on 2D videos. There are all sorts of ways can achieve that. In our work, we use Openpose[6] as an intermediate approach to get 2D pose estimation, aiming at

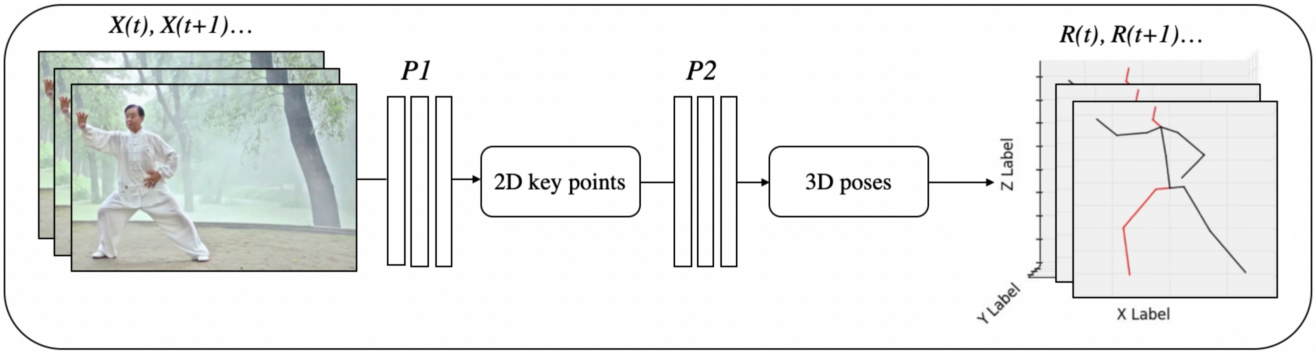

Figure 3: (Top) This is our whole pipeline. We firstly use a 2D key points dectactor (Openpose) P1 to make a 2D key points prediction, and then use P2, 3D human pose estimate model, to get 3D pose. The final step is to achieve motion retargting between two skeleton sets that with different attributes

getting the 2D key points which can be used as training data for 3D human pose estimation. During this process we can generate sufficient 2D key points and save as JSON files with 2D joints (BODY_25).

The second step is using these 2D key points to extract 3D poses. This part belongs to a general research area about 3D human pose estimation. Recently, Dario Pavllo[7] present a decent back-projection semi-supervised training method for human pose estimation. With the implementation of back-projection part, the outputs can also be used as training data when label data is scarce. Inspired by this cycle consistency training approach, we firstly use the pretrained model of this paper to make a prediction of 3D joint positions and do motion retargeting based on these 3D joint information. Then we drive the avatars in our dataset and render these 3D models frame by frame according to their animations. After this process we can get amounts of unlabeled video that can be used to make a prediction of 3D poses, and then map them back to 2D space. This recurrent process can enlarge the dataset, so that we can use the neural network retrain the model and make the model more robust. Based on the retrained model, the final prediction result would be more accurate.

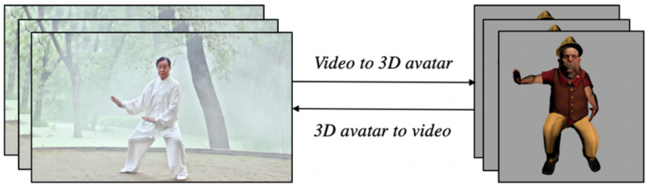

Figure 4: Our system can make an accurate mapping and find an align between input videos and the 3D driven avatars.

In this part, we save all the motion data, including 3D joint coordinates, of all frames as CSV files. Then we map the joints data to rigged skeleton with the internal function of Blender, and output the BVH files, which is considered as the one of the motion capture file formats. The first part of the pipeline aims to generate source motion files from any wild videos.

2. Human motion retargeting based on sparse skeleton 3D motions

By using the retrained model, we can only get 17 or 20 joints. However, the 3D avatars we want to drive have different bone lengths and more joint numbers. In this paper, we figure out a sufficient approach and finish mapping spares joints to dense joints with the support of HumanIK in Maya.

Compared with other neural retargeting models, which can only solve the motion transfer between same joints numbers, retargeting in Maya provides a more generic and systematic way to make up this defect and solve this kind of motion transfer problem.

For the forward kinematics, each joint of the skeleton has a local transform, since joints in the skeleton can form a hierarchy, which is a parent-to-children structure, we can easily get the global transform of each joint by doing matrix calculations. If one joint is the parent of another joint, with the parent-child relationship, the global position of the child joint can be calculated with a matrix multiplication between global parent transform matrix and its own local transform matrix. Each matrix can be transferred by Euler angles. M_{gchild} is the global transform matrix, $M_{gparent}$ is the global parent transform matrix, and M_{lchild} is the local transform child matrix.

$$M_{gchild} = M_{gparent} M_{lchild}$$

With this certain characteristic of FK, the problem of copying motion from one skeleton to the other skeleton can be converted to joint angles and positions transfer, which should be pair to pair transfer. Therefore, forward kinematics method has a big gap when doing motion transfer with sparse joints positions information.

Compared with forward kinematics, for the inverse kinematics solver, each IK joint has a target position which is considered as an end effector. Finding the Euler angles of all the joints set is the final goal of IK method. IK problem can be solved under certain constrained, and the most common way to solve it is by using the Jacobian matrix. By applying constraints, the IK problem can be converted to solving an optimization function:

$$min\frac{1}{2} \parallel J\Delta\theta - \Delta b \parallel^2 + \frac{1}{2}\alpha \parallel \Delta\theta \parallel^2$$

$\Delta\theta$ can be represented by n dimension vecotr that can show the change of the Euler angles. Finding the changed Euler angles is the key to solve IK problem. This IK equation can also be converted to the following equation:

$$(J^T J + \alpha I)\Delta\theta = J^T \Delta b$$

Motion retargeting in Maya is based on the Autodesk HumanIK character solver. Once you have defined your character's skeletal structure using the *HumanIK character structure*, you can transfer animations between skeletons that have the same or different skeletal structures and proportions.

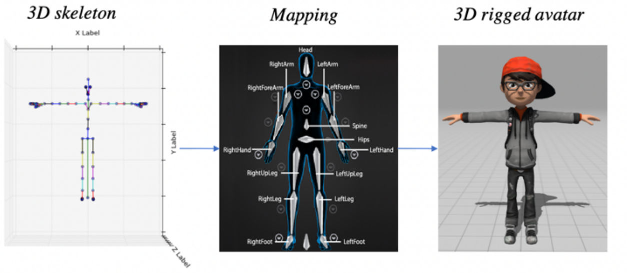

Figure 5: No matter sparse skeletons or densen skeletons, if they have the same topologies, they can be mapped to the same bone set(at least 16 joints).

The quality of the motion retargeting outputs that we expect is mainly based on the effectiveness of the internal logic of HumanIK. The internal logic of HumanIK is that: for different bone number, it can map the bones into a same set, at least 16 joints skeleton structure, since for the human skeleton structure, it has the same skeleton topology. Secondly, since the predicted motion files contain the 3D information of each bone, HumanIK can analyze the movements of the source avatars, and easily copy these baseline information, do matrix calculations and use the inverse kinematics solver to find a final position for the target joint, which can reflect the original motions in decent quality.

III. POTENTIAL APPLICATIONS

Our system can be used for several potential applications. Nowadays most motion detection rely on depth camera like Kinect. With the future improvement, we believe that one interesting application will be that users can free their hands. And it could be used with one RGB camera or smartphone to capture human motion, reconstruct rigged 3D avatars and drive them in real time.

Figure 6: These game euipments are used for pose detections. With the order from left to right and top to down, they are: Nintendo Switch, Wii, Microsoft Xbox and Xbox Kinect.

For the entertainment part, especially in the game industry, it can also be employed commercially. Instead of holding heavy game controllers and VR handlers, players can just record their personal dance videos, use these videos as input and let the 3D rigged digital avatars follow their dance. So rather than follow the dancers, let's create a dance!

IV. LIMITATIONS AND FUTUREWORK

Even if we build a novel neural network motion retargeting system, which can transfer motion from 2D videos to 3D rigged digital human avatars successfully. There are still some limitations: firstly, It is not an end to end framework, the estimated human 3D pose is not accurate. During each step the error will accumulate. Most of the research paper only use sparse 2D key points (22 or 25 joints) to supervise the learning process, which may be insufficient for effective training. We may use the state-of-the-art differentiable renderer (e.g. Soft Rasterizer[8]) to obtain dense pixel-level supervision to obtain stronger loss, which may lead to more accurate predictions. Secondly, Maya HumanIK is not flexible. The IK joints that we need to assign are fixed, and it mainly aims for the same skeleton topology. Hence, it is hard to extend the method to handle more complex 3D models, which has complicated skeleton topologies.

ACKNOWLEDGMENT

Great thanks to Mingming He for her kind writing suggestions. Since this project has been conducted during my summer intern at USC ICT, thanks to all my colleagues and friends for their supports during the whole summer!

REFERENCES

[1] Villegas, R., Yang, J., Ceylan, D. and Lee, H. (2018). *Neural Kinematic Networks for Unsupervised Motion Retargetting*.

[2] Jang, H., Kwon, B., Yu, M. and Kim, J. (2018). *A Deep Learning Approach for Motion Retargeting*.

[3] Chan, C., Ginosar, S., Zhou, T. and Efros, A. (2018). *Everybody Dance Now*.

[4] Aberman, K. , Wu, R. , Lischinski, D. , Chen, B. , & Cohen-Or, D. . (2019). *Learning character-agnostic motion for motion retargeting in 2d.*

[5] Gleicher, M. .(1998). *Retargetting motion to new characters.*
Conference on Computer Graphics & Interactive Techniques. ACM.

[6] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2018). Openpose: *realtime multi-person 2d pose estimation using part affinity fields.*

[7] Pavllo, D. , Feichtenhofer, C. , Grangier, D. , & Auli, M. . (2018). *3d human pose estimation in video with temporal convolutions and semi-supervised training.*

[8] Liu, S. , Chen, W. , Li, T. , & Li, H. . (2019). *Soft rasterizer: differentiable rendering for unsupervised single-view mesh reconstruction.*