# Adapting Style Transfer Model to Customized Datasets

Siddhant Arora
*Department of Computer Science*
*University of Texas at Austin*
Austin, Texas
siddhant.arora@utexas.edu

Rohit Neppalli
*Department of Computer Science*
*University of Texas at Austin*
Austin, Texas
rohit.neppalli@utexas.edu

Amal Babu
*Department of Computer Science*
*University of Texas at Austin*
Austin, Texas
amalbabu@utexas.edu

Kunal Mody
*Department of Computer Science*
*University of Texas at Austin*
Austin, Texas
kunalmody@utexas.edu

***Abstract*—Generating realistic human motion is a problem that has been worked on for decades, with many different approaches... We aim to create a model that when given an example of a singular human motion, can generate a motion that is similar to the example it was given. To do this, we will use a style transfer approach that takes a base motion and uses the human example as style data to apply over the base motion. We hope that this model will create output that is unique to the human example it was fed and we expect that the stylistic human motion will be better interpreted by human subjects than a more generalized approach.**

***Index Terms*—style transfer, neural network, generative adversarial network, human-robot interaction, virtual reality, motion synthesis**

## I. Introduction

Many different approaches to the task of motion synthesis for VR humans, that is the generating of original motion data from some input data. These include Physically valid statistic models by Wei et al[3], a GAN-based approach by Wang et al[5], and finally a style transfer-based approach by Aberman et al [4]. Style transfer, however, is unique in that you get output that features individual characteristics of the input motions, rather than just an aggregation of learned motions from all the input data. If you were to provide motion of a human performing a certain action or moving a certain way as one of the inputs, that characteristic would be retained to a large extent in the output of the model. This also enables much work and research, as it allows for a person to be put into an environment as a VR agent by feeding the model that person's walking motion.

We intend to perform a Human-Robot Interaction (HRI) study that compares stylistic VR human avatar motion against a more generalized human motion. We hypothesize that the motion generated from one human, retaining that human's stylistic characteristics, will be much easier for a human observer to interpret compared to a motion generated from the aggregate of many humans, which will not retain the stylistic characteristics of any of the humans it was trained on. By stylistic elements, we mean characteristics that are unique to that human - i.e. gait, stride length, hunching over, etc. We believe that retaining the unique stylistic elements of a human's motion creates a more humanistic and interpret-able motion to a human observer. To do this, we use the style transfer model introduced by Aberman et al to generate motion with the stylistic elements of one human. We configure the model so that it will work with the data dimensions of the VR setup at the BWI Lab at UT Austin.

After this, we will set up an experiment that compares how human surveyors interpret the motion of the style transfer model and a general model; this experiment will test our hypothesis described above, and also outline any future works that can build upon the working model, either by making the stylized output motion even more recognizable (more accurately reflect training data) or by adding some new functionality.

## II. Acronyms / Definitions

1) BVH: Biovision Hierarchy (File format to provide motion capture data, providing the skeleton of the character as a hierarchy and the corresponding motion data for each joint)

2) CMU MoCap Dataset: motion capture database created by Carnegie Melon University. Contains BVH files that we train our model with.

## III. Aberman Model

For the problem of motion synthesis, we decided to use the style transfer to create realistic human motion. There are other models of motion synthesis: statistical models, physically valid statistical models, and GANs. Our focus is largely on GANs and style transfer. All of these models have one thing in common- they are trained on a large amount of data and and create motion output. Style transfer also follows this principle, by using the training data to create new motion. However, it is different in that GAN outputs a singular new motion, while style transfer models create a model that will take any 2 motion inputs and apply a style transfer between the two. This is useful for 2 reasons.

First, the model is able to retain style elements. Since GANs use lots of data to create one motion, any individual styles that the input data might have had will most likely get washed out do to the large amounts of data that was used to train it. While style transfer models still needs large amounts of data, it is able to retain the style of the 2 inputs that are used to actually create the final motion animation.

Second, and specifically to our experiment, we are able to define the 2 input motions in separate ways. One motion will be called the *content motion* which is motion that represents how the final animation will walk through physical space. The second will be the *style motion* which represents the style in which the human is walking. An example of a content motion would be a right turn, while a style motion would be an angry walk. When the style transfer model is applied, the angry style should be transferred onto the right turn, resulting in an angry right turn. This is useful for the HRI aspect of the study as we can measure how humans react differently to the same content motion but with different style i.e an angry right turn vs a sad right turn. This is also useful because only one right turn needs to be recorded, and any style can be applied to it.

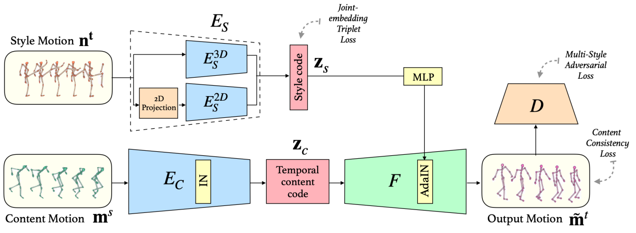
Style Motion $\mathbf{n}^t$
$E_S$
$E_S^{3D}$
2D Projection
$E_S^{2D}$
Joint-embedding Triplet Loss
Style code
$\mathbf{z}_s$
MLP
Multi-Style Adversarial Loss
$D$
$\mathbf{z}_c$
Content Consistency Loss
$E_C$
IN
Temporal content code
$F$
AdaIN
Content Motion $\mathbf{m}^s$
Output Motion $\tilde{\mathbf{m}}^t$

Fig. 1: The process of the model to create the ouput motion after receiving base and style motion inputs.

To meet the requirements specified above we chose the Aberman model. The Aberman model uses the same idea of content motion and style motion. It allows support for applying style over the desired motion in physical space. It also creates relatively smooth motion, especially when Jacobian Inverse Kinematics is applied to smooth the animation. For this model, the input is 3D pose data about each joint, per frame that was recorded. This pose data is in the form of a BVH file, the semantics of which will be discussed later.

The model works as such: the style input will get passed through a style encoder, $E_S$, which creates a style code. The style code is then passed through a perceptron layer the output of which is passed in through the decoder. The content motion passes through a content encoder $E_C$, which creates a content code. This content code acts as the second input to the decoder, F, which will decode the two motions and apply the style transfer, outputting our final style transfer animation. Adversarial loss will be calculated from this output. Motion smoothing can be done to the animation through inverse kinematics after the final output.

## IV. Approach 1

### A. Data Transformation

One primary difference between the Aberman model and our model is the number of joints used in the training and testing due to limitations of the VR lab here at UT Austin. The Aberman model uses the CMU MoCap database [7] and re-targets it towards the CMU standard of 31 joints (a total of 90+ channels). Since the skeleton is also re-targeted, all distances between joints (offsets) are also the same. To transform the dataset into something we can use in conjunction with our lab, we first use the re-targeted database provided by Aberman et al. and process it to reduce the number of joints from 31 down to 21 using a recursive tree-node deletion implementation. We essentially parse each BVH file and remove joints, all while re-calculating the offset distances between the joints to make sure it matches the required skeleton. After doing this, we had to edit many configurations of the model that referenced the new skeleton.

Figure 2 shows the difference in output between the original model and our adapted model. It can be seen that joints such as the foot and elbow were removed to conform to our specifications. Another piece of transformation work was going from the VR data available through the BWI lab to the BVH format that the model works with. Their data comes in the form of a CSV that lists information such as $x, y, z$ position and $x, y, z, w$ rotation values for each joint that they are tracking. We are able to use an existing python library called bvh-toolbox [https://pypi.org/project/BVHtoolbox/] to convert this CSV format to BVH without doing the transformation manually or writing the script ourselves. This

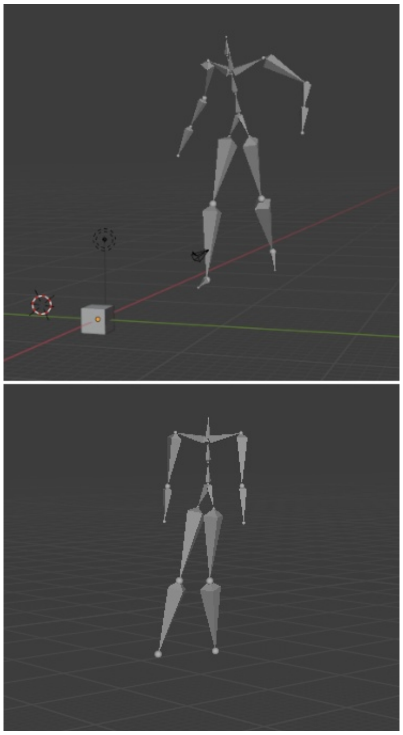

Fig. 2: The top image shows model output without changing the number of joints (31). The bottom image shows the edited skeleton, effectively reducing the number of joints to 21.

library is able to take 3 CSV files of hierarchy, rotation, and position for joints and combine them together into a complete BVH file. Since all the data we need is store their CSV, it was simple to reformat this data and then use the csv2bvh function to get a working BVH file.

### B. Trained Model

We used the model created by Aberman et al [4] for the style transfer. However, the number of joints that this model uses is not configurable. We modified the model to one that could be applied to the joints that the BWI lab at UT Austin can track using it's VR system. The pre-trained model provided by Aberman et al. took an input of 31 joints, with each joint having 3 channels except the root joint, which has 6 channels. The model works such that the training and testing data must have the same number of channels. Similarly the content motion and style motion must have a direct mapping of joints.

To accommodate for the capabilities of the VR system at the BWI Lab at UT Austin, we had to make the model configurable to to number of joints that the VR system could track (21 compared to the 31 of the Aberman Model). The Encoder and Decoder were modified so that the outputs would match the new data dimensions. Finally, we used the re-targeted data sets to train our updated model. This updated model still follows the process outlined in Figure 1, however; no changes were made to the overall implementation of the model.

### C. Results

The resulting model that we generated after 300,000 iterations is able to make motion that looks humanoid; however, it currently does not always follow the style very well in comparison to the model from the Aberman paper. We see training errors of anywhere from a factor of 2 to 3, while the Aberman model produces motion with an error of close to a few hundredths. Due to the lesser amount of joints present in our model, the existing model had to be undergo parameter tuning to replicate the same degree of accuracy as the original model (though there is still room for improvement).

In general, we found that many specific style-content motion pairs are not compatible (for instance, angry style overlaid on sad content), which led to incorrect, though intriguing motion. As seen in Figure 3, the model seems to learn a jumping motion although both the style and content source have no visible jumping animations. Other styles do produce valid output, however, as seen in Figure 4, which represents an happy style over a base strutting motion. Non-conflicting styles generally produced output that mimicked human-like motion. Examples of valid style-motion pairings would include a base motion that had to do with pure movement and a style representing emotion.

However, we also noticed that turning motions were preserved through the motion synthesis. In another example, Figure 5, the base strutting motion conducts a right turn at the end of the clip, which is also present in the resulting style transfer clip. The success of this implied that different styles are transferable over the turning motions, which will allow us to replicate turning motions using style motions that we record ourselves. This concept is also very helpful in setting up VR Hallway experiments (as mentioned below).

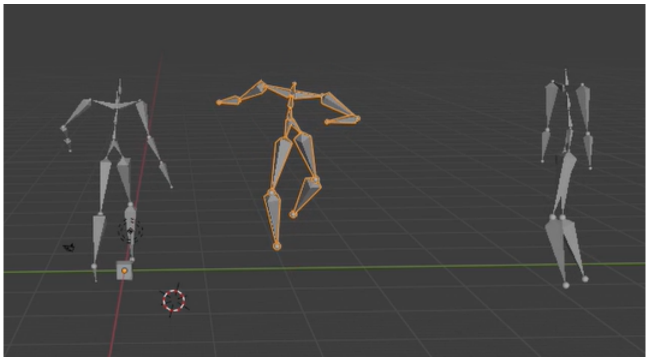

Fig. 3: One style-base combination produces a "dance-like" movement instead of a traditional walking animation.

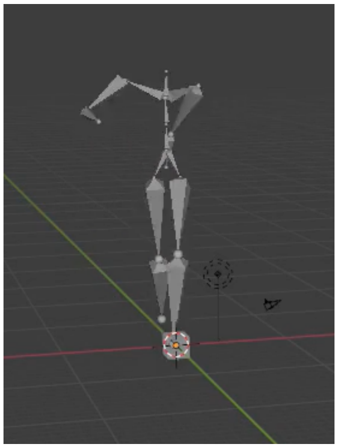

Fig. 4: An example of accurate style transfer produced by our existing model.

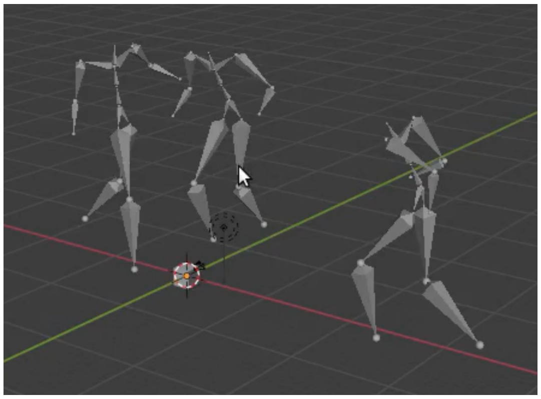

Fig. 5: Content motion that turns is visible in the stylized output as well. In this image, the style and content base motion are shown on the left, while the output is on the right.

Though the output of the model is not what we intended, we've been able to attribute this error to one of three causes:

1) Iterations: the newer model is trained on an "smaller" dataset due to the reduced amount of joints, which doesn't let the model train to convergence. We suspect that we can modify the learning rate or increase the number of training iterations.
2) Loss Function: The current aggregate loss, as described by:

$$\mathscr{L} = \mathscr{L}_{\text{con}} + \alpha_{\text{adv}}\mathscr{L}_{\text{adv}} + \alpha_{\text{reg}}\mathscr{L}_{\text{reg}} + \alpha_{\text{joint}}\mathscr{L}_{\text{joint}} + \alpha_{\text{trip}}\mathscr{L}_{\text{trip}}$$

   has pre-defined constant values ($\alpha$) that may be specifically hyper-tuned for the 31-joint model. For our model to work, we may have to find better hyperparams.
3) Joint Flexibility: the lack of joints present in the animation is simply not enough to mimic realistic human motion.

## V. Approach II

After working with the Virtual Reality trackers, we realized that extracting joint position and rotation data was going to be difficult. Furthermore, this system proved to be a challenge in tracking the number of joints we needed. As a result, we looked to another option, a body tracking software which could record a greater number of joints and made extracting the joint data much more simpler.

The Microsoft Azure Kinect was a good fit for these specifications. We created a recording of a participant walking into the frame for a couple seconds, and were able to quickly gather the position and rotation data of each joint (about 20 which matched the CMU skeleton). We first created a CSV with this motion data to help us create the file format supported by the model. After creating the BVH file (format supported), we tried to render the motion data in Blender, hoping to see an avatar walking. Instead, what we saw was a jumbled mess of joints all out of place. We realized one key difference in how Kinect and the BVH file represent the joint data. Kinect represents each joint in a local coordinate system specific to that joint. As a result, the x, y, and z axes do not translate directly from one joint to another. Figure 6 illustrates the local coordinate systems for each joint, and how they compare to the global frame compatible with the BVH.

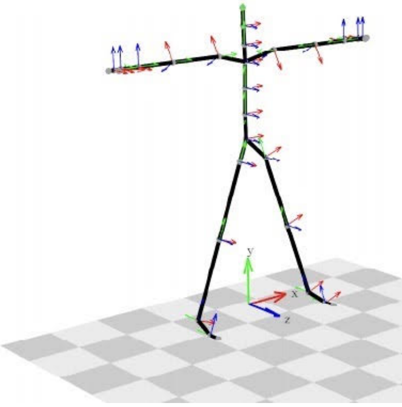

Fig. 6: Local coordinate frame per joint as presented by Kinect.

BVH, on the other hand, has a vastly different approach to representing the joints. First, it doesn't store the position data of every joint; instead, it stores just the root's (hips). In addition, it stores the offsets from one joint to another (identical to bone length); with this information, along with how the joint rotated (stored in a rotation matrix), the BVH can recursively calculate the positions of all joints. As a result, the position data received from Kinect is basically unneeded. The way BVH views the joint data is also different from the Azure Kinect; instead of having a local coordinate system for each joint, BVH has a global frame. This way, every joint has the exact same x,y, and z axes, and translation between two joints is straightforward. Figure 7 visualizes this, and it can be seen there is no difference between any joint coordinate systems; they all follow one singular structure.

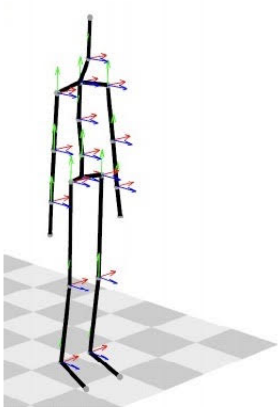

Fig. 7: Global coordinate system for all joints as presented by BVH.

We realized moving from the Kinect data to the BVH required changing the orientation of the local coordinate systems to match the global one. Some systems were simple: flipping one of the axes from positive to negative or vice versa seemed to do the trick. Others, however, were much more involved; the entire system had been rotated, so the axes just didn't line up with the BVH coordinate frame. We had to rotate the local system, but how to do this? First, we can't rotate just the joint positions because BVH doesn't support position data. Second, the joint data given by Kinect is merely a rotation matrix relative to the camera's coordinate frame. This matrix tells us how to translate from the Kinect's camera coordinate system to the local coordinate system of the joint. As a result, there was nothing for us to actually rotate except for this matrix, which already described a rotation. Figure 8 shows the camera's frame of reference, and looking back at Figures 6 and 7, it can be seen that every frame, whether it be the local ones presented by Kinect or the global one fitting to the BVH is different; this made re-configuring the data challenging.

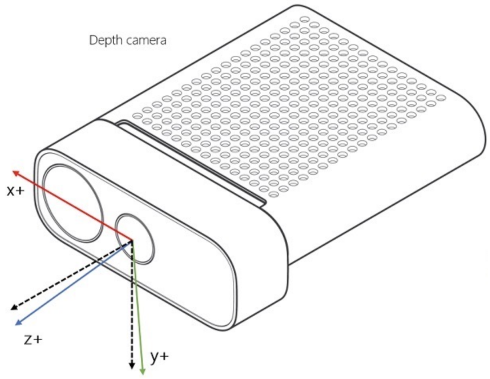

Fig. 8: Coordinate system presented by Azure Kinect camera.

## VI. Experiments

With this model, which is compatible with the VR motion capture from trackers available in the BWI Lab, we hope to make several advancements that build on this work and conduct a few experiments which hopefully validate our hypothesis of stylized motion generation being more recognizable than a traditional model (simply creating motion without factoring in the human characteristics).

### *Hallway Passing*

The BWI Lab currently does a great deal of research into hallway passing of humans with robots within VR and real-world situations. We will setup an experiment that compares our generated VR human motion with another existing approach to the same task, such as the statistical model [2]. Participants would experience both types of generated VR human motion and provide responses such as which model looks more human-like in its walking, which felt easiest to navigate around when walking past in the hallway, etc. By doing this, we aim to show that having an actual human motion as reference through utilizing recorded data creates more realistic or approachable human characters in a VR setting than the statistical model, which doesn't factor in the unique walking style of its training data.

### *Motion Differentiation*

Style transfer was chosen as the approach for motion generation due to style transfer's ability to retain individual characteristics of style and content inputs. However, we cannot simply take it for granted that for motion generation one can necessarily notice the individual characteristics of the inputs. So, we will setup an experiment that will test whether participants can tell who some outputted motion is based off of given 3 different input motions from 3 different people. The goal is to see whether people can actually tell motions apart and if certain characteristics make it more or less likely to uniquely identify someone. Going along the same lines as this is seeing whether participants can or cannot tell between a VR agent generated through this aproach and a regular human walking in VR. By doing all this, we hope to show how generated motions can uniquely be identified and thus be useful in applications such as research or video games, where motions being differentiable would make a big difference.

### *A. Remaining Work*

For both experimental designs discussed above, a new generalized model is needed to compare results against. We will need to train a model that will take lots of walking data from many different humans and use it to create an aggregate, generalized motion. After increasing the accuracy of our model, we can start using the VR system at the BWI Lab to record stylistic data for our model. That data must first be retargeted to the standard CMU Mocap skeleton so that it has the proper dimensions for the model and can be used as an appropriate style. Optionally, we could record our own base motion to ensure the most neutral base result possible. Finally, we need an appropriate survey for human subjects to fill to assess the performance of the two models. We plan on asking short and simple questions to obtain the most honest responses from the individuals, such as which model they found the most ’human’ and what could be improved to make the model more recognizable.

We will be setting up the experiments through Unity. BVH files are not natively supported by Unity, so we cannot directly import it. However, BVH can be converted to an FXH format, which can be loaded into Unity. More research needs to be done, however, about integrating an FXH asset into a regular Unity environment within VR.

## VII. Future Work

To make this research flexible for any future work by others, we will work on making the project easily generalizable for any use case of joints. Currently, we have a script that can modify the BVH files to use our needed joints, but within the code we still needed to make manual changes to the skeleton CMU.yml file that defines the skeleton as a whole as well as the tree structure of joints. This is likely a task that can easily be automated as it is a matter of removing values from an array and updating indices in another part of the file accordingly. After such an implementation, we hope users can merely enter the number of joints they are utilizing as part of command line arguments when training the model.

Another future extension of this work is motion planning to get a VR character that uses this Style Transfer approach to follow a specified path from point A to point B. Currently, we can generate motion, short or long in duration depending on the baseline data, in a particular direction or style, but not coherently enough to navigate an environment. Such an endeavor would involve choosing the most appropriate style motion at a given instance and seamlessly combining many of them together. Although this may be outside of the range of possibility for the time left that we have to work on this research, we are proposing this idea so future works may be able to achieve this functionality or at least generate a new idea influenced by this design.

## VIII. Conclusion

We predict that the style transfer model will create virtual reality agents that will not copy the style of the human they were trained on, but instead imitate the *style* of the human they were trained on. We believe that this is key in creating realistic human motion; instead of one general walking motion, agents should learn motion from individual humans. We predict that humans who interact with these agents in virtual reality will feel more comfortable interpreting their motion than standard VR avatar motion. In general, our goal is to create inconsistent motion that varies depending on the human it was trained on; we expect that these inconsistencies will produce naturalistic walking motions.

## Acknowledgment

We would like to thank Professor Justin Hart for providing helpful guidance throughout this challenging process. We would also like to thank Yijia Weng for helping us navigate the enormous codebase of the style transfer repository; her advice was hugely impactful to the work we accomplished, and her development of the codebase itself made it very simple to discuss challenges and possible solutions. Lastly, we would like to thank members of the BWI Lab for helping to understand how the VR tracking system is setup and how to capture our own data to test our model.

## References

[1] N. Tsoi, M. Hussein, J. Espinoza, X. Ruiz, and M. Vázquez, “SEAN: Social Environment for Autonomous Navigation,” Arxiv https://arxiv.org/pdf/2009.04300.pdf (accessed September 10, 2021).

[2] Qi Wang. Statistical Models for Human Motion Synthesis. Modeling and Simulation. Ecole Centrale Marseille, 2018. English. ffNNT : 2018ECDM0005ff. fftel-02071347f

[3] X. Wei, J. Min, and J. Chai, “Physically valid statistical models for human motion generation,” in ACM Transactions on Graphics, vol. 30, Issue 3, Article No.: 19, pp. 1–10.

[4] K. Aberman, Y. Weng, D. Lischinski, D. Cohen-Or, and B. Chen, “Unpaired Motion Style Transfer from Video to Animation,” in ACM Transactions on Graphics, vol. 39, Issue 4, Article No.: 1, pp. 1–10.

[5] Qi Wang, Thierry Artières. Motion Capture Synthesis with Adversarial Learning. Intelligent Virtual Agents, Aug 2017, Stockholm, Sweden. ffhal-01691463f

[6] Holden, D, Saito, J & Komura, T 2016, 'A Deep Learning Framework for Character Motion Synthesis and Editing', ACM Transactions on Graphics, vol. 35, no. 4, 138. https://doi.org/10.1145/2897824.2925975

[7] The data used in this project was obtained from mocap.cs.cmu.edu