Human Aware Motion Planning for 6 DoF Robot Arm in Painting Application

Group 5: Eric Chen, Emily Wu

***Abstract*—Collaborative robots must operate safely and efficiently alongside humans, especially in dynamic environments like artistic painting. We develop a human-aware motion planning system for a 6-DoF robot arm collaborating with a human artist. Using real-time 3D pose estimation from OpenPose and depth sensing, we capture the artist's motion and predict future poses with the lightweight siMLPe network. Predicted poses are converted into obstacle representations for real-time, collision-aware trajectory planning with MoveIt2. Our system integrates open-source platforms including ROS2, Gazebo, and various learning models to enable adaptive robotic painting. Results demonstrate a strong proof-of-concept for human-aware collaboration, with future work focused on improving prediction robustness, sensor fidelity, and artist-specific data collection.**

***Index Terms*—Collaborative Robotics, Robot Learning, Motion Planning, Human-Robot Interaction**

I. Introduction

How can we predict human motion from a sequence of 3D body poses? This is a useful deep learning application because it can improve decision-making in autonomous driving, provide better tracking of human motions, and improve human-robot interactions. Specifically for our research, we are investigating how a collaborative robot painter can work with a human artist while painting, such as in Figure 1, exploring how generative AI systems can be used in the creative process can help us locate biometric data to explore the meaning of an authentic creative process. To do this, we develop a human-aware motion planner for collaborative painting with a high degree of freedom (DoF) robot arm.

Fig. 1. Artist Painting Alongside Robotic Arm

II. Prior Work

Predicting human joint motion has been extensively studied in the past, with the idea of fusing spatio-temporal information in a sequence-to-sequence task being core concepts.

A. Traditional Methods

Previous works have employed probabilistic models such as Hidden Markov Models (HMMs) to predict human motion by learning transitions between discrete states based on extracted features [6]. Similarly, Gaussian Process Latent Variable Models (GP-LVMs) have been used to capture the underlying low-dimensional structure of human motion sequences by mapping complex pose data into a continuous latent space [5]. While these are effective for modeling periodic or simple motions, they struggle to generalize when more complex and non-repetitive human motions arise. This limitation arises due to their reliance on assumptions about motion regularity and their limited capacity to capture spatial-temporal dependencies in natural human behavior.

B. Deep Learning Architectures

Applications of deep learning to predict human motion include using Recurrent Neural Networks (RNNs), Graph Convolutional Networks (GCN), and Transformers. RNN's sequential nature allows them to model temporal dependencies, but they often suffer from vanishing gradients and long-term dependency issues. There has also been use of encoder-decoder frameworks to embed human poses with the use of long short term memory to update the latent space and predict future motion [2]. In GCN, the human pose is built as a graph to generate a mesh of the human model, and it is trained using a generative adversarial method in which the generator generates a mesh similar to the manifold of a human mesh distribution and the network acts as a supervisor to determine if the mesh is real [4]. Transformers have been applied in other works to fuse spatial and temporal information using the key idea of self-attention to use repetitive motion patterns for prediction [7]. These more complex networks are harder to analyze and modify, as there are many more parameters to train.

III. Methods

A. Open-Source Software

Besides the deep learning networks we utilized, other open-source softwares were used to enable our pipeline to work together. v4l2loopback and FFMPEG were used to create a virtual device in Linux and restream the depth camera's RGB

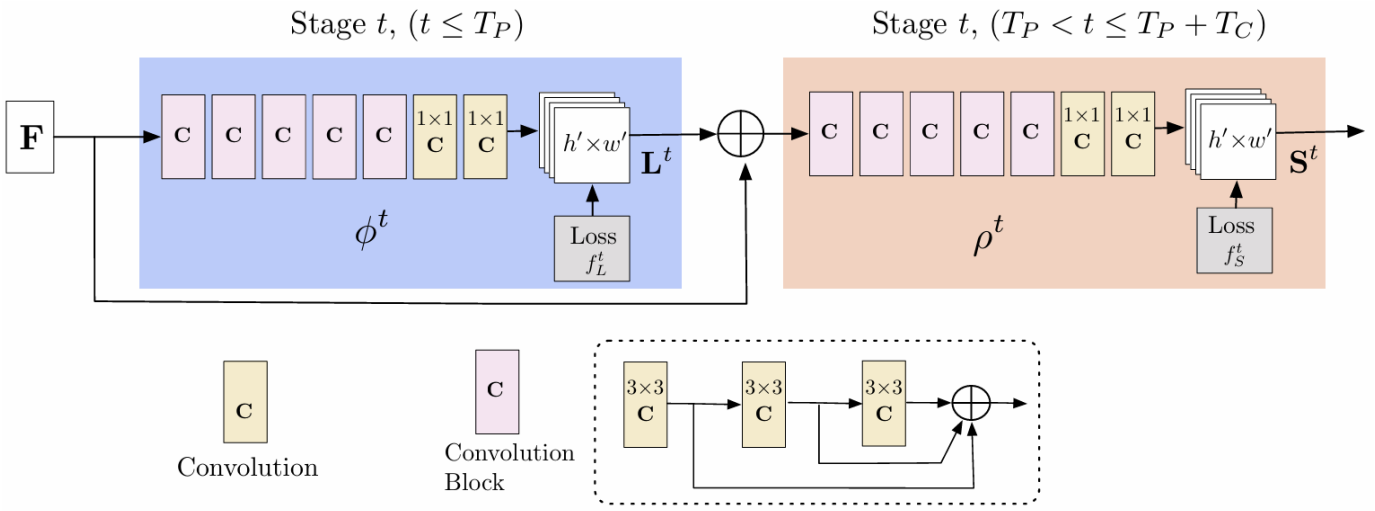

Fig. 2. OpenPose Network Architecture

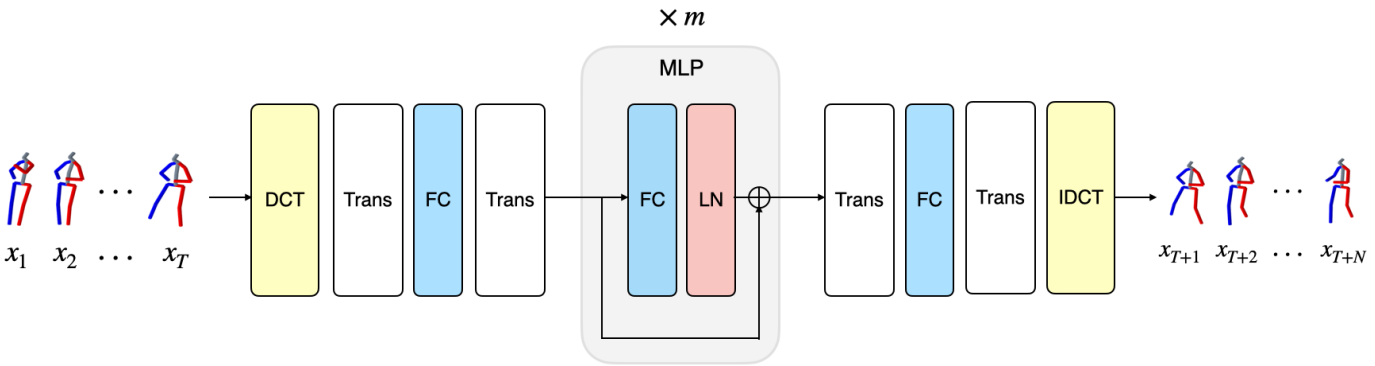

Fig. 3. siMLPe Network Architecture

video for OpenPose to use. Specifically for the Intel Realsense D455 depth camera, we installed the Intel Realsense software development kit (SDK) and librealsense package.

ROS2 Humble allowed us to connect predicted poses to the simulated environment in Gazebo for planning. It managed and connected different data streams. MoveIt2 is a ROS package for plug-and-play robot manipulation, and it includes capabilities in motion planning, manipulation, 3D perception, kinematics, control, and navigation [8]. We also used the XArm ROS 2 package that enables MoveIt for our specific robot model, the UFactory XArm 850.

B. Open Pose

OpenPose is a real-time multi-person keypoint detection library for body joint estimation [1]. The key innovation of the work is the real-time multi-person capability. The network uses part affinity fields (PAFs) to learn to associate body parts with people in an image, achieving high accuracy and realtime performance using a bottom-up approach. This work also provided keypoint detectors in the body, foot, hand, and face. The network architecture is shown in Figure 2, with the first set of stages predicting PAFs and the last set predicting 2D confidence maps of body part locations. In both stages, multiple 7x7 convolutional layers are used leading into 1x1 convolution kernels.

C. SiMLPe

To predict the next position of a human's joints, we used siMLPe, a multi-layer perceptron based network. The key idea in siMLPe is that a human's last pose is similar to future poses, which is intuitive to our understanding of the human range of motion. Therefore, we can let the network predict the residual between the future pose and the last input pose. This model is able to achieve state-of-the-art performance with low mean per joint position error (MPJPE) and 20-60x fewer parameters [3]. Thus, this system is lightweight and efficient, while still being accurate.

This network was trained on the Human 3.6M dataset, a large-scale dataset of human poses and corresponding images captured by motion capture system. During training, sequences of past poses were provided as input, and the network was optimized to predict future poses by minimizing the mean per

joint position error. By focusing on predicting the residuals between the last observed pose and the future pose, siMLPe effectively captures subtle variations in human motion while maintaining low computational complexity, which makes it particularly suitable for real-time applications where both speed and accuracy are critical.

As shown in Figure 3, the network is fairly simple. It takes in a sequence of 3D human poses in the past T timesteps. The poses are then transformed using a discrete cosine transform (DCT) to encode temporal information, transpose layers, and fully connected layers. The fully connected layers operate on the spatial dimension of the transformed motion sequence to account for space. Then, the MLP blocks consisting of fully connected and layer normalization layers are used to merge information across frames. In the siMLPe paper, they found that using 4 MLP blocks was most effective, balancing prediction accuracy and model size. This configuration allowed the model to outperform previous methods while maintaining a simple architecture for real-time applications [3].

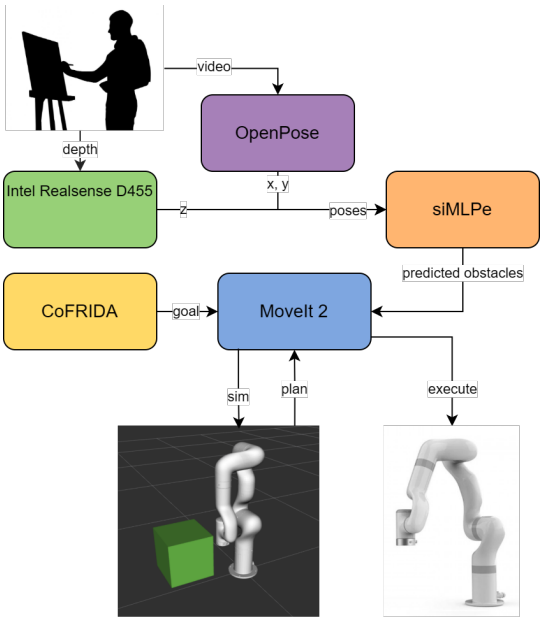

Fig. 4. System Architecture

D. Combined System

In our combined system, we are capturing the artists movements using an Intel Realsense D455 camera. Using OpenPose to track the human joint positions and querying the depth data at joint positions, we obtain the human poses in the past 25 timesteps. This is then sent to our pre-trained siMLPe model to predict the poses in the next 10 timesteps. The predicted poses are reduced more as we take the torso and arm joints to create cylinders or other primitive shapes to represent a safe area around the human pose for the robot to plan around. These obstacles are sent to MoveIt which will simulate the obstacles and robot arm to create safe trajectories. The goal positions for the robot are given from the CoFRIDA node, which plans strokes for the robot to paint using Generative AI [9]. Once a trajectory is found, the robot can execute it in the real world.

IV. Results

We visualize the detected 2D joint keypoints using OpenPose overlaid on the live RGB camera feed in Figure 5. The right side of each image shows the extracted skeleton structure in 2D space. This mapping provides the foundation for accurate 3D human pose reconstruction in our pipeline when combined with depth information.

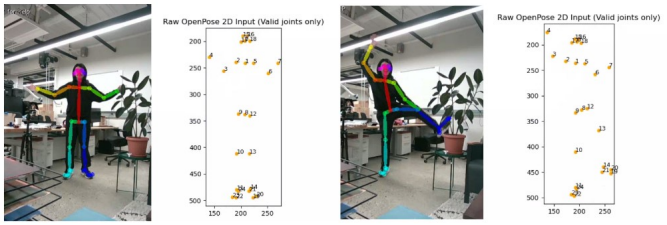

Fig. 5. Real-time Mapping of Joint Positions

We show the predicted future joint trajectories from the siMLPe network over several frames in Figure 6. Each plot visualizes the progression of joint movements based on past observed poses, demonstrating the model's ability to anticipate natural human motion with minimal latency.

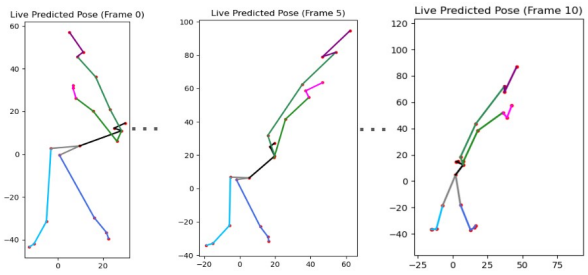

Fig. 6. Real-time Motion Predictions from siMLPe

The predicted human motion from siMLPe were used to generate virtual obstacles around the human body. These obstacles are placed into the motion planning environment in Gazebo, shown in Figure 7, allowing the robot arm to plan and execute collision-free trajectories in real-time alongside a moving human collaborator.

V. Discussion

A. Extension

In this work, we showed a strong proof-of-concept for a motion planner using predicted human motion. Our extension to siMLPe was the integration and connection of multiple disjoint open-source software through modification of systems and/or rerouting data to enable the application of siMLPe in an online robotic system.

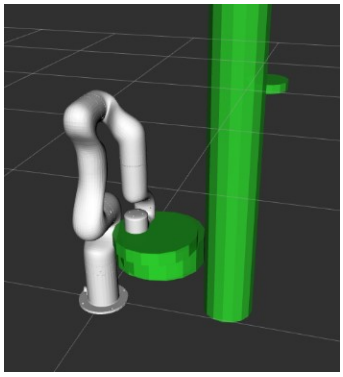

Fig. 7. Online Obstacle Generation from Entire Pipeline

B. Challenges

Some of the challenges we ran into were limited hardware, lack of documentation, inconsistent human skeleton formats, and lack of robustness of networks. For the depth camera, we initially tested out the ZED and the PrimeSense depth cameras. The ZED was promising, however it ended up being deprecated and lacked the human tracking SDK capability, so we turned to the Intel Realsense D455 camera. This camera had fewer SDK capabilities, which led us to adding OpenPose in the first place.

A challenge related to the datasets used in training included lack of documentation of H3.6M keypoint joint labels. This made it more difficult to extract the key body parts we wanted to translate into obstacles. Also, it was difficult to format the OpenPose skeleton for the expected siMLPe skeleton structure. OpenPose was formatted with 25 joints, while siMLPe trains on 22 joints but outputs 32 joints. This indirect translation led to uncertainty in the siMLPe output. There was also a lack of robustness to un-detected keypoint joints. When OpenPose does loses tracking, it will return (0, 0) for the 2D location of that joint. Directly sending this data to siMLPe results in inaccurate predictions.

C. Future Work

In this work, we assumed that an artist's movements are not any different from collected data of human movements, such as walking, talking on the phone, or other everyday activities in the 3.6M Human Dataset. Some future work could include creating a dataset of artist movements while painting using motion capture. Modifying siMLPe's architecture or retraining it for artist motions could also be a way to improve our model for artistic applications. For example, siMLPe does not have any nonlinear activation functions, so attempting to add back some of the complexity of human tracking could improve the model.

Integrating better depth cameras or body tracking modules other than OpenPose could also help our predictions. For measuring the state of a person's joints, the most ideal setup would be using motion capture, which would definitely be applicable to an artist. While OpenPose provided us with accurate and real-time joint measurements, it sometimes found human skeletons in random objects even when no humans were in frame and was somewhat noisy. Validation of predicted pose data from siMLPe against motion capture would also be an important next step to both improving the model and assuring the results we are getting are accurate.

Improving online obstacle generation would also improve this project. Our current simulated environment runs slowly due to the delay in sending obstacles to Gazebo. This would only decrease in performance if we increased the fidelity of the human, so it would be necessary to improve our performance for the best online results. Also, adding rotation of body links using quaternions from the pose data would greatly improve the motion planning of the robot, as the simulation is more accurate to real life.

Finally, training the network to work even with missing joints or faulty sensor measurements would improve the robustness of the system. To do this, we could record OpenPose data and use it for training siMLPe, as well as change the expected joint input format and number of joints to train on. Retraining the network would likely provide a large increase in accuracy of our predictions, especially since the data is collected using the sensors on our real system.

Acknowledgment

We would like to thank the Deep Rob course and staff for providing us with the knowledge of different deep learning architectures. We also want to thank the Robot Studio Lab for providing us with the resources and project to complete the work.

Statement of Contribution

Emily Wu focused primarily on integrating the Intel depth camera with OpenPose for real-time human pose estimation and configuring MoveIt2 for motion planning. Eric Chen focused primarily on implementing and adapting the siMLPe network for future pose prediction. Both team members collaborated closely to integrate the individual components into a unified system. Additionally, both equally to the writing of the poster and final report. Emily also created the project website.

References

[1] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[2] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," 2015. [Online]. Available: https://arxiv.org/abs/1508.00271

[3] W. Guo, Y. Du, X. Shen, V. Lepetit, A.-P. Xavier, and M.-N. Francesc, "Back to mlp: A simple baseline for human motion prediction," *arXiv preprint arXiv:2207.01567*, 2022.

[4] Y. Huang and N. Xiao, "Graph convolutional adversarial network for human body pose and mesh estimation," *IEEE Access*, vol. 8, pp. 215 419–215 425, 2020.

[5] N. D. Lawrence, "Gaussian process latent variable models for visualization of high-dimensional data," in *Advances in neural information processing systems*, vol. 16, 2004.

[6] L. Liu, Y. Jiao, and F. Meng, "Key algorithm for human motion recognition in virtual reality video sequences based on hidden markov model," *IEEE Access*, vol. 8, pp. 159 705–159 717, 2020.

[7] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: Human motion prediction via motion attention," 2020. [Online]. Available: https://arxiv.org/abs/2007.11755
[8] MoveIt2, https://moveit.picknik.ai/humble/index.html.
[9] P. Schaldenbrand, J. McCann, and J. Oh, "Frida: A collaborative robot painter with a differentiable, real2sim2real planning environment," 2022. [Online]. Available: https://arxiv.org/abs/2210.00664