Project Report : Humanoid Robot Imitation of Human Motion from Instructional Videos

Fatine Boujnouni, Hamza Kabbaj

January 2019

Abstract

This short report resumes an attempt to implement the method of Humanoid Robot Imitation of Human Motion from Instructional Videos [Xue Bin Peng et al.] as a class project of the object recognition MVA class. This project includes both supervised learning and reinforcement learning. It aims to train a robot to learn motions that are performed in any video. In contrast to motion capture used in film making to get a motion from multiple captors, the ingenuity of this method is using Human Mesh Recovery to get poses of the human in the video. It costs less and can be used in any video available on the internet.

1. Introduction

The studied article presents a 3 steps method that makes a robot learn from an instructional video. First, the « pose estimation » step takes in input video frames and outputs a list of joints positions and rotations of the human for each frame. Due to the imperfections of the first steps, the motion resulting from the estimated poses could be incoherent in time and space. The « motion reconstruction » step's goal is to repair these imperfections by minimizing an objective function that insures the time and space coherence and outputs a new reference motion. Finally comes the « Motion Imitation » step that uses reinforcement learning to make the humanoid robot agent reproduce the same motion.

In our implementation, we skipped the motion reconstruction step for time issues. Still we manually made sure that the motion was coherent before passing it the learning step.

In this report, we will introduce individually each of these 3 steps. We will then talk about a mid-step which is mapping between different vector representations used in the first step and the last one. After that we will present how we made our implementation and show the results.

2. Three Steps Method

2.1. Pose estimation

The first step consists of retrieving the vector containing the pose estimation of the human in the RGB images using the HMR code.

The Human Mesh Recovery is an end-to-end framework that takes an RGB image as an input and outputs a set of 3D shape and pose parameters of the SMPL human model. The SMPL (Skinned Multi-person Linear) model is a realistic 3D model of the human body that is based on skinning and blend shapes and is learned from thousands of 3D body scans. It's a kinematic tree composed of a free-floating root joint which corresponds to the pelvis and 23 spherical joints (called also « ball joints »). It has a 10-dimensional shape vector, which determines the 3D skeleton of the human body (the person's limb lengths) and a 72-dimensional vector which corresponds to the 3D positions of each of the 23 spherical joints plus the root joint (23*3 + 3 = 72).

2.2. Motion Reconstruction

Since some poses could be wrongly estimated, the motion reconstruction step aims to improve the coherence of the movement in time and space. Mathematically, a reconstruction loss is defined as an objective function to minimize. The reconstruction loss includes 2D consistency loss, 3D consistency loss and smoothness loss.

2.3. Motion Imitation

This last step allows a humanoid robot agent to imitate the reference motion given by the previous step. In reinforcement learning, this is made possible by an exploration, exploitation paradigm. After a reward is defined, the agent tries different set of movements at each simulation. As he gets rewarded for getting close to the reference motion, the agent starts learning from his previous experience. The reward is the sum of the pose reward, velocity

reward, end-effector reward and deviations penalization reward. The scene is simulated thousands of times until the agent's motion matches the reference motion.

2.4. Mapping

In this project, we are working with two different models that have different pose vectors as illustrated in Figure 1. The mapping consists of finding the corresponding joints between the two representations. And then converting SMPL vectors to quaternions in DeepMimic model. Matching these joints allows the robot to imitate the exact pose predicted by HMR.

3. Implementation

While implementing the three steps method presented previously, we had to develop some intermediate algorithms to make it work with our own examples.

In our tests, we filmed few videos of simple movements that would be easily predicted by HMR. This is to avoid using motion reconstruction.

In steps, we started by framing the video, applying HMR to each frame. Then we convert the resulting SMPL vectors to DeepMimic representation. After validating the motion we launch the training. In case the motion wasn't well predicted, we made manual adjustments.

3.1. Video framing

We started by implementing a script that takes a video as an input and outputs all the frames of that video. We could modify the frequency of selecting frames when the movement is too slow. In this way we avoid having two consecutive frames with the same pose.

3.2. Pose estimation with HMR

After having the set of RGB frames, we apply the HMR code to each one of them. A 72D pose vector is then generated for each frame and concatenated into a list. But before handing it to DeepMimic, the 72D vectors should be mapped to 44D DeepMimic vectors.

3.3. From SMPL to DeepMimic Vector

The first step in the mapping is finding correspondences between the two types of vectors. We had to make multiple tests on each dimension of the SMPL vector in order to figure out which part of the body it represents.

After that we started making correspondences intuitively between the SMPL pose vector and DeepMimic pose vector. One of the challenges was to figure out that joints 13 and 16 should be summed to output the vector that represents a shoulder in DeepMimic.

joint index	SMPL indices	Body parts	Has a matching in DeepMimic vector
0	[0 : 3]	Pelvis	Y
1	[3 : 6]	Left hip	Y
2	[6 : 9]	Right hip	Y
3	[9 : 12]	Belly	Y
4	[12 : 15]	Left Knee	Y
5	[15 : 18]	Right Knee	Y
6	[18 : 21]	Chest 1	N
7	[21 : 24]	Left Ankle	Y
8	[24 : 27]	Right Ankle	Y
9	[27 : 30]	Chest 2	N
10	[30 : 33]	Right Foot	N
11	[33 : 36]	Left Foot	N
12	[36 : 39]	Neck	Y
13	[39 : 42]	Left Shoulder 2	Y
14	[42 : 45]	Right Shoulder 2	Y
15	[45 : 48]	Chin	N
16	[48 : 51]	Left Shoulder	Y
17	[51 : 54]	Right Shoulder	Y
18	[54 : 57]	Left Elbow	Y
19	[57 : 60]	Right Elbow	Y
20	[60 : 63]	Left Wrist	N
21	[63 : 66]	Right Wrist	N
22	[66 : 69]	Left Hand	N
23	[69 : 72]	Right Hand	N

We ended by selecting only the SMPL joints that have their matching in the DeepMimic pose vector. Then we wrote a script that transform the 3D vectors of each of the spherical joints of the SMPL model into a quaternion or an angle, depending on the corresponding joint in the DeepMimic pose vector.

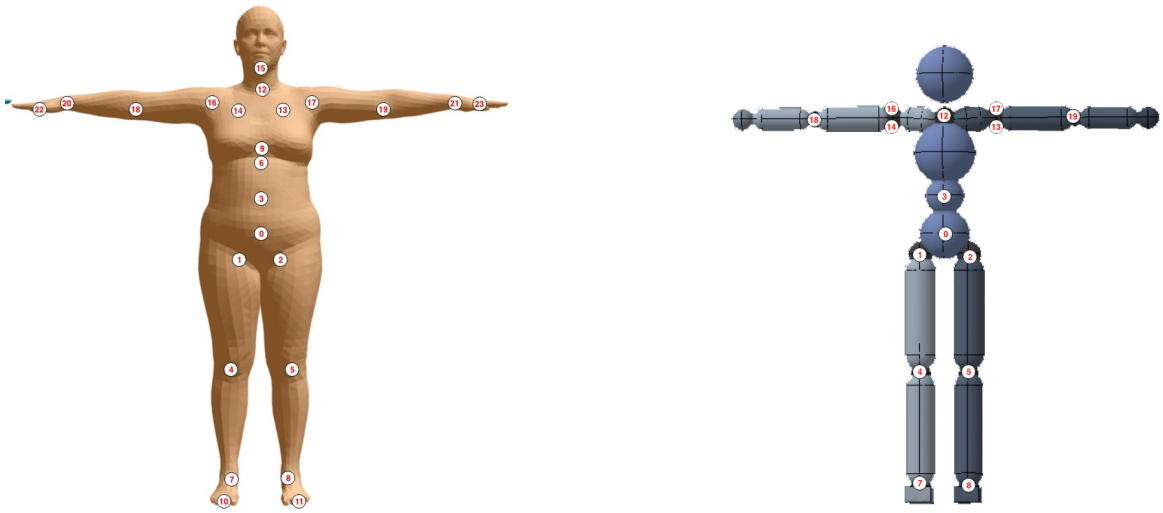
3.4. Motion Validation

Before launching the trainings, we should make sure that the reference motion is acceptable. We could visualize it in the DeepMimic interface. Unfortunately, HMR gave very bad results. So we had to make manual adjustments before moving to the last step.

3.5. Manual Adjustments

Since we didn't use motion reconstruction, we made manual corrections to have a coherent movement as a reference for motion imitation step.

First, we had to manually set values to the root position so that the whole humanoid body can correctly move in the scene.



(a) SMPL joints (b) DeepMimic vectors

Figure 1. Correspondence between the 24 SMPL joints generated by HMR algorithm in left, and the corresponding 13 DeepMimic vectors in right. We should note that shoulder vectors (18/14 and 17/13) are merged into one in each side.

Then, we figured out that for some symmetric movements with respect to the body parts, HMR made false predictions in one of the two sides. So we made a copy of the well predicted body part to the other one.

3.6. Training

After having a well designed reference motion, we set it as an input for DeepMimic in order to learn how to replicate it. Training a single movement takes about 10 thousands iterations and 60 million samples. With a 16 core cpu, the training takes about a day.

4. Results

We made several tests with different movements. Due to poor HMR predictions, we chose to test two simple movements that gave good results after training.

The first motion was raising hands up and down. Like a fly movement. Even though the motion was simple, HMR output was poor. As we can see in Figure 3, legs are not symmetric, and it created undesired rotation. After adjusting manually those artifacts, the results were good. The training algorithm converged in about 8000 iterations (Figure 2).

The second motion was a little more complex than the first one. We wanted to train the robot to have equilibrium one foot. After 10000 iterations we can see that the training curves are still growing.

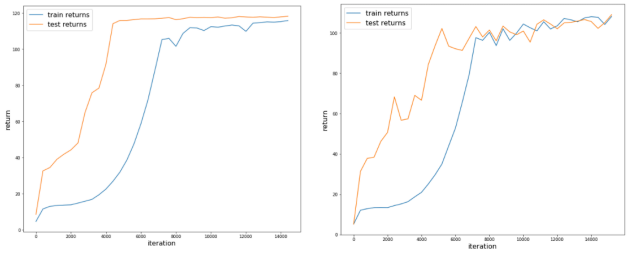


Figure 2. Learning curves of the first motion in left and the second motion in right.

The agent performs perfectly the first motion after the training. But the second motion was a little bit harder to learn. This is due to the imperfect reference motion that was given as input.

5. Limitations and discussion

From our experience, we can tell that the biggest limitation was the results from HMR. As we couldn't do motion reconstruction, we had to manually deal with the various artifacts and incoherences in the generated motion.

With a better HMR model and a well designed motion reconstruction, the 3 steps method would perform better without any manual modification.

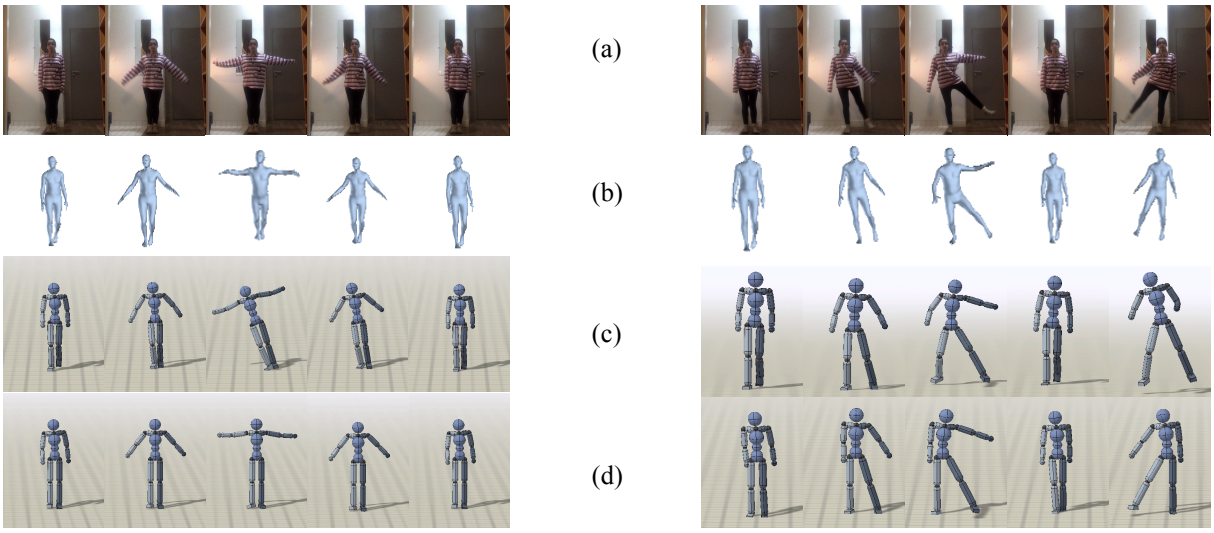


Figure 3. Results of creating the reference motion that would be given to DeepMimic agent. (a) Set of frames extracted from the original video. (b) HMR outputs. (c) Result of mapping SMPL vector to DeepMimic vectors. (d) Manually adjusting artifacts of HMR outputs.

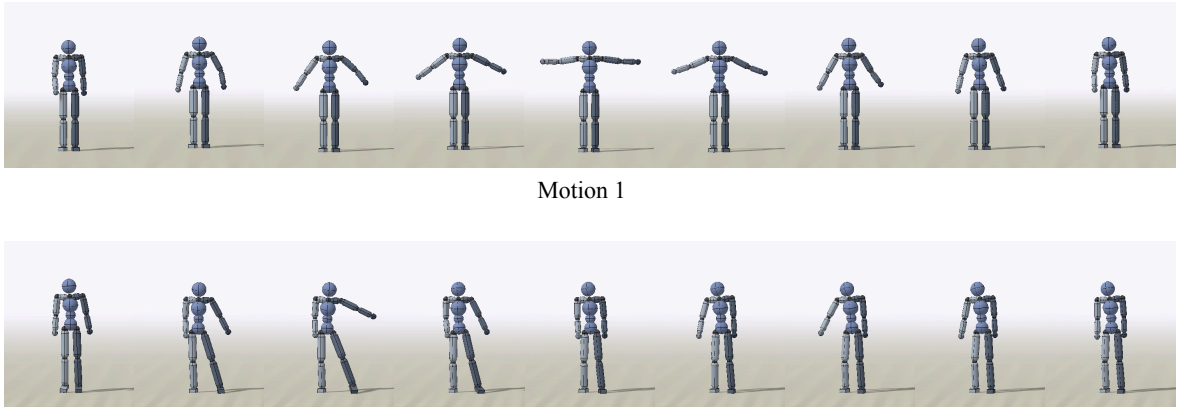


Figure 4. The resulting motions after training upon motion 1 and motion 2

6. References

1. Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine 2018. SFV: Reinforcement Learning of Physical Skills from Videos.
2. Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne 2018. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills.
3. Angjoo Kanazawa, Michael J. Black, David W. Jacobs, Jitendra Malik. End-to-end Recovery of Human Shape and Pose. CVPR 2018.
4. Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. 2015