Final Proposal

Aaron Gokaslan, Jonathan Chang

Abstract

The ability to train tasks on a robot through teleoperation expands the versatility and utility of robots in many ways. Current work on falls short in that they have only shown a small subset of simple tasks that we hope to expand upon through our implementation of their algorithm. Our technical approach first involves collecting our own training data. In order to do this, we have created a recorder for Baxter that collects training data as well as a simple neural net for object detection using Tensorflow's object detection API. As for our actual neural net implementation, we decided to construct the net presented by the paper using PyTorch. Baxter can now record training data by collecting depth and RGB images while recording key information about the position and orientation at each time step. Furthermore, Baxter is now able to playback the output from our net which consists of an array of linear and angular velocities. The playback requires the position coordinates and the quaternions for the orientation which can easily be calculated given the recorded orientations.

Introduction

If we can successfully use virtual reality teleoperation data as training data, we now have access to an influx of training data. We could capitalize on the widespread popularity of virtual reality games and tap into the gaming community to contribute training data through interactive games on platforms such as Steam. The payoff of such a platform would be the increase in available training data as well as an increase in scope of many tasks. For example, we would no longer need to physically be at a farm to teach a robot basic farming tasks.

To the best of our knowledge, the limits of today's practice is the paper that we are currently planning on implementing. The paper successfully teaches the robot basic tasks such as pushing and placing an apple on a plate through virtual reality teleoperation with about 30 minutes worth of training data for each simple task. They also achieved chained actions where the robot was able to perform multiple simple tasks in a row. We do not necessarily think that the paper's approach fell short; however, we believe that our goal to be able to eventually train from a simulation improves upon previous approaches.

Our current technical approach first involved using ROS reality to store our training data and to then manually collect training data. With the collected training data, we feed this into our pre-trained neural net which will output a resulting linear and angular velocity which dictates the orientation and the position in the next time step.

From this information, Baxter will move to the required position. The next steps include allowing simultaneous playback of the nets output and Baxter. We aim to produce comparable results as the paper and we are also in the process of asking the team for their pre-trained model/implementation/weights of their model to assist us in the process.

Related Work

The main related work concerns the actual paper we are reproducing: Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation. We seek to extend, reproduce, and improve this prior work. We seek to demonstrate that this can approach can be reasonably reproduced and extended to models such as Baxter and through virtual reality integration, crowdsourcing the data collection for task demonstrate could prove possible.

Technical Approach

Our current technical approach first involved using ROS reality to store our training data and to then manually collect training data. With the collected training data, we feed this into our pre-trained neural net which will output a resulting linear and angular velocity which dictates the orientation and the position in the next time step. From this information, Baxter will move to the required position. The next steps include allowing simultaneous playback of the nets output and Baxter. From the initial tests that we have conducted, we found that the net does well to get the end effectors position accurately; however, the overall orientation of the arm position does not take the environment into account. What I mean by this is that although the end of the arm gets to where it needs to go, the “elbows” do not avoid obstacles such as tables or walls to achieve the outcome. Our approach to this problem may involve incorporating a separate path planning algorithm or trying different inputs to our neural net. More specifically, currently instead of taking in the joint angles/torques, we input the orientation and the quaternion of the end effector into the net. Perhaps if we instead incorporate more information about the joint states during the training, the net would be able to learn a path that does avoid obstructions. Another aspect that we would like to further explore is the positioning of the Kinect. Currently, there exists a level of occlusion in our training data that arises from the camera orientation relative to Baxter. Finding a clearer view may improve the performance of our net as well.

We aim to produce comparable results as the paper and we are also in the process of asking the team for their pre-trained model/implementation/weights of their model to assist us in the process. Finally, once Baxter is able to learn from physical training data, we hope to transfer this process to the Virtual Reality space.

Evaluation

Originally, we had the ambitious goal of fully implementing the paper in question and recreating their results. Currently, we have not yet been able to recreate the results of the paper. Our progress has been slowed be certain architecture choices that we made to supplement the paper. Different from the neural net architecture of the paper, we decided to input the position and orientation (by way of quaternions) to the net instead of the 9 joint states. We have shown on a simple task of pushing a box that our current implementation is able to recreate the end-effector position of the task at hand, but it is still unable to properly plan a valid path that will avoid and take the physical space into account. As mentioned in the Technical Approaches section, we are currently in the process of exploring a couple of valid solutions to improve upon our initial results. The video recordings of our current results can be found in our final presentation.

Conclusion

Our goal for this paper was to be able to have Baxter take training data in the form of RGB and depth pictures, end-effector position (x,y,z), and the end-effector orientation (x,y,z,w - quaternions), and output the linear and angular velocity which will direct Baxter to perform a simple task. The simple tasks could involve motions that involve gripping and moving the location of objects which could be further chained with other pre-trained tasks. Currently we are able to train a simple task on Baxter; however, the resulting output is not without its flaws and requires further exploration to recreate the results of the paper we are implementing.

The most immediate next step would involve integrating all of our individual pieces together. We currently have a playback system and a trained neural net. We aim to connect these two together so that immediately after the net outputs the next step, our playback system could move Baxter accordingly at each time step. After achieving this, we would need to improve the path that Baxter takes in order to achieve the task outputted by the net. More specifically, from the initial tests that we have conducted, we found that the net does well to get the end effectors position accurately; however, the overall orientation of the arm position does not take the environment into account. What I mean by this is that although the end of the arm gets to where it needs to go, the "elbows" do not avoid obstacles such as tables or walls to achieve the outcome. Our approach to this problem may involve incorporating a separate path planning algorithm or trying different inputs to our neural net. More specifically, currently instead of taking in the joint angles/torques, we input the orientation and the quaternion of the end effector into the net. Perhaps if we instead incorporate more information about the joint states during the training, the net would be able to learn a path that does avoid obstructions.

Another aspect that we would like to further explore is the positioning of the Kinect. Currently, there exists a level of occlusion in our training data that arises from the camera orientation relative to Baxter. Finding a clearer view may improve the performance of our net as well. This also plays into the next step that we speak about in the next paragraph of allowing our training data to be sources from virtual reality. We believe that a good intuition about how the position of the camera effects our system would be integral in collecting better training data as well as creating a more intuitive way for virtual reality users to be able to create data.

Work to be done after this would be to convert the collection of the training data from the physical space to the virtual space as shown by the paper. Achieving this would give us all the tools to aim for the goal of crowdsourcing our training data. To further generalize this process, we would also be interested in exploring how other robots react to our system for Baxter. Perhaps we will be able to convert our architecture to be more generalizable to a wider domain of robots.