Pose Estimations for Quadrupeds from Video

CS249R Final Paper | Stan Chang, Nao Ouyang, Juspreet Singh Sandhu, Akash Shah

Abstract

We follow the “Skills From Video” (SFV) paper and use pose estimation on videos of dogs to create a reference motion (an animated quadruped skeleton model) without using any expensive motion capture equipment. SFV created control policies for humanoid robots from amateur gymnastics videos; we aim to create quadruped control policies using videos of dogs doing tricks. We present a stereo quadruped dataset, and outline the future workflow to use DeepMimic to turn these reference motions into robust physical controllers. Our files are online at

https://github.com/nouyang/cs249r_finalproject

Introduction

The DeepMimic paper came out in 2018 and represented the state-of-the-art for taking advantage of reference motions for training physical (in simulation at least) policy in high degree-of-freedom continuous action spaces. However, their training data was collected from expensive and complicated motion capture (mocap) equipment with detailed post-processing. [11,12] Followup research from the Skills From Video [8,9] team (which included members of the original DeepMimic paper) realized the possibilities of using monocular video to develop pose estimates for humanoid systems.

DeepMimic was an improvement on existing reinforcement learning-based policies by enabling a user to combine action policies with a goal function to achieve desired, natural, and smooth movements from a robot. Prior to the development of reinforcement learning based approaches, the state-of-the art drew on the use of sophisticated physical and / or constrained kinematics models to dictate the motions available for a simulated figure. While accurate, these models require extensive upfront work and remained constrained to the pre-programmed motions. Thus, DeepMimic showed it was possible to learn impressive and robust policies for complex robots and/or animation characters.

The work done by the Skills from Video team [8,9] pushed the usefulness of the DeepMimic process further, by allowing the use of monocular video from consumer cameras to use as the reference motion. However, even with these latest developments, research has been focused on humanoid systems for both training and outcome datasets.

As a result, we decided to focus on quadrupeds for our project. Quadrupeds have different motion constraints and abilities including the option to engage in different pacing methods such as trotting and cantering. A paper titled ‘Locomotion Skills for Simulated Quadrupeds’ [15] documents how skeletal and muscular structure affects the range of motion available to quadrupeds. We set out to explore the possibility of using YouTube videos to train quadruped systems through motion reconstruction and pose estimation.

Related Work

We relied on a few key research papers for our project: “SFV: Reinforcement Learning of Physical Skills from Videos” [8,9], “DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills” [11,12], “DeepLabCut: markerless pose estimation of user-defined body parts with deep learning” [18], and “Locomotion Skills for Simulated Quadrupeds” [14,15]. In addition we used Deep Lab Cut, which is a python library to bridge our pose estimation needs.

The Skills from Video team started by using both 2D point and 3D mesh pose estimates from videos and applied training that minimizes differences between these two datasets. Additionally, they augmented pose estimations by:

- Tracking continuous physical actions across frames
- Limiting changes over time
- Enabling agent training from starting states distributed similarly to the raw videos
- Augmenting datasets by collecting additional data from rotated videos

The DeepLabCut algorithm produces a 2D point location estimator after training on a hundred or so manually annotated frames. Running the trained estimator on stereo video (a pair of videos of the same scene from slightly different perspectives) creates two distinct pose estimate trajectories. The two estimates can then be combined with triangulation to produce a 3D estimate.

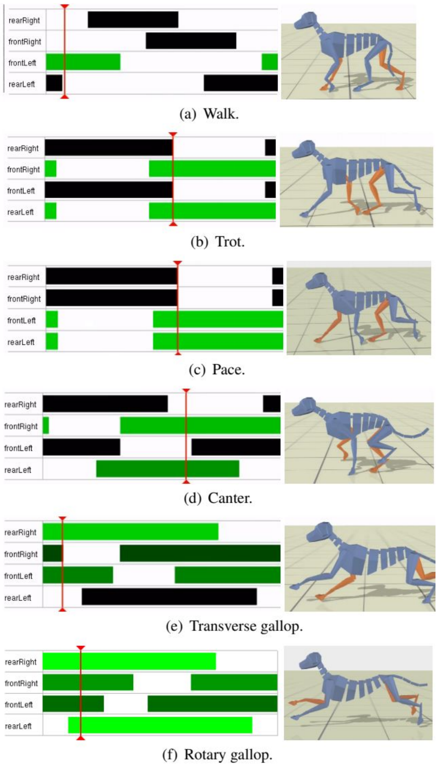
In Deep Mimic, the authors sought to augment motion of simulated systems to achieve particular goals. The input include video(s) of smooth motion performed by these robotic systems as well as cost function(s) that drive the system to achieve particular goals. The motion videos are developed through either mocap, sophisticated physics and kinematic models, or manually keyframed as described above. Using these as source material to drive the system toward achieving a set objective enabled much more natural and smooth motion. More, multiple different motions from different reference motions could be stitched together automatically to achieve a much wider range of objectives.

The paper on quadruped systems focuses on modeling aspects of their physiology. It looks at both manually recreating the motion of quadrupeds once the skeleton models are created as well as using motion capture data, and possibly augmenting it with monocular video manually annotated with joints and other important points in keyframes, to generate pose estimates. While useful as a general background, we look at automate much of the work proposed by this paper.

Methodologies

Setup

With our project, we initially set out to extend the Deep Mimic [12] paper by augmenting its use of professional motion-capture data with the much larger repository of lower-quality YouTube videos. We would develop 3D reference motions by applying off-the-shelf human pose-estimation algorithms to videos of humans dancing. Feeding these into Deep Mimic would then provide us with output target angles for each of the joints as well, which were then achieved with PD controllers deciding the final torques applied at each joint.



(a) Walk.

(b) Trot.

(c) Pace.

(d) Canter.

(e) Transverse gallop.

(f) Rotary gallop.

Figure 01: Manually set motion for quadrupeds. Image from [15]

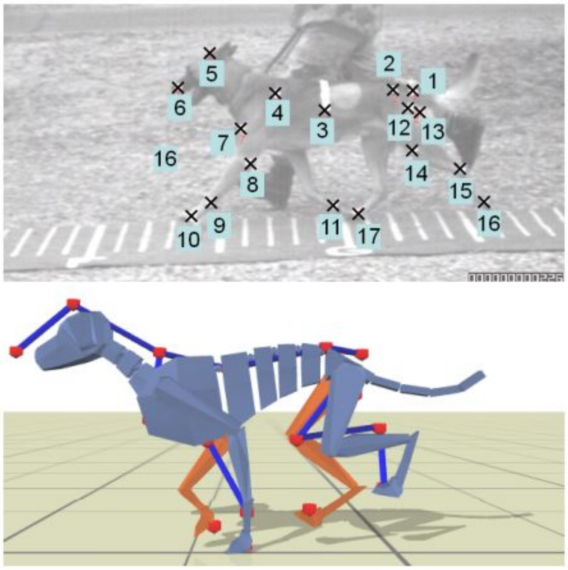


Figure 02: Video data for capture of reference motions. Image from [15]

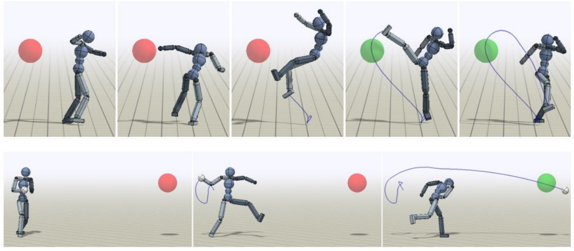


Figure 03: When trying to teach the system to spin-kick or throw a ball, much more natural motions resulted due the use of Deep Mimic. Image from [12]

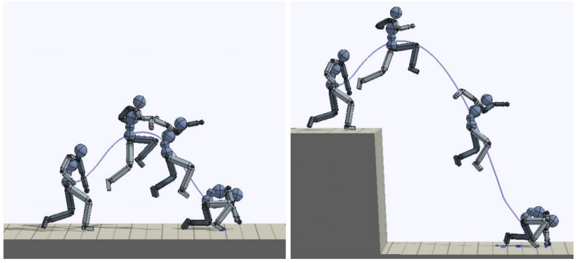


Figure 04: Motion of a linear leap was effective transferred to the robotic system jumping off a ledge. [12]

Our research drew us to a follow-up paper, Skills from Video [9], developed later that year that covered much of the work we had planned to do. To learn from existing work, we sought to implement their code and replicate their output.

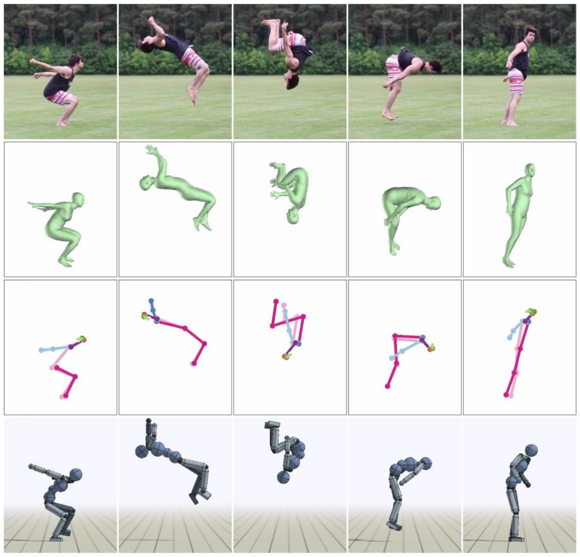


Figure 05: Process Flow from the Skills from Video paper [9]

Seeking to leverage the existing SFV work, we planned to start with a new target motion (e.g. 'moonwalk' dance move), find several YouTube videos, and use their motion reconstruction code [10] to generate output that could be used by the Deep Mimic repository [13].This, again, proved non-trivial. The motion reconstruction library had dependencies on other algorithms (e.g. 2D pose, 3D mesh) which were poorly setup: hardcoded folder paths, lacking functional examples. While each individual deep learning algorithm had their own pre-trained models and datasets, many of these were removed from the repository haphazardly when licensing requirements changed.

Unfortunately, similar to many systems projects, we ran into file format incompatibility issues that were further complicated by this work involving 3D and joint rotation data. Digging further, we realized that pose-estimations were not homogenous. They could be in 2D or 3D. They could be represented as individual points, points connected as a skeleton, or a complete 3D mesh. They could be specified in global or relative coordinate systems. They could be specific as joint rotations, and further as 3D joint rotations or simplified into 1D joint rotations when possible (e.g. knees, elbows). The angles themselves could be specific in Euler angles, axes-angles, or quaternions.

Even though the motion reconstruction library was created by the same research group, it only output a biovision hierarchy (BVH) file [10] which was not directly usable by the Deep Mimic library. Though we found tools [7] that would enable us to transform the BVH files into the required animation format for Deep Mimic. An important limitation is that these tools assume a humanoid skeleton, which created problems for us that we discuss later.

Dance Dance Robot

With the motion reconstruction library transforming videos into BVH files, converters changing them into the custom animation format, and Deep Mimic outputting the physical controllres, we were ready to learn dance moves. The aforementioned file issues took out the first step, so we focused on the second and third.

We found BVH files of humans dancing from the SFU MOCAP library [5]. We planned to take one of these BVH files that included a single reference motion clip. The reinforcement learning agent would explore different joint angles to achieve that motion, generate O(10^6) trajectories, and learn from that. If this worked, we would then start with YouTube videos and work through the entire pipeline described above. We would definitely learn from this effort, but it would only replicate existing work. Therefore we focused on our second possible project, extending the work of Skill From Video to also apply to quadrupeds.

Quadrupeds

We were interested in training a Deep Mimic dog using YouTube videos of dogs doing tricks [Appendix]. This was particularly interesting to us since mocap data was even more limited for quadrupeds when compared to the already sparse libraries for humans. This effort turned out to be more complicated than we expected. Starting backwards through the process described above, we explain our work below.

Training the Deep Mimic character, our final step, was made easier by a recent development. For unclear reasons, dog motion data was added on November 5, 2019 to the Deep Mimic repository despite no mention of it in the paper [33]. With this, we had a character skeleton to train on and an example of the format required by Deep Mimic.

Transforming the BVH file into Deep Mimic animation format (specified as a 1D or 3D rotation for each joint) meant porting the code for humanoid skeletons over to function for dog skeletons. While this is straightforward conceptually, it would involve non-trivial time and resources. Additionally when trying to set up Deep Mimic, we found that the release was solely intended for replication. It did not provide functionality to import custom reference motions.

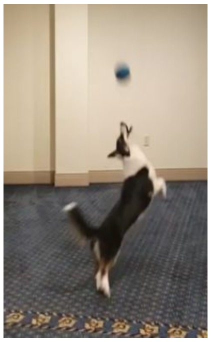
Therefore, we decided to focus on the earlier part of the pipeline, coercing pose estimates out of youtube video. Looking further, we found two candidates that were clearly intended for community development: Deep Pose Kit [20,21,22] and Deep Lab Cut [17,18,19].

These two candidates were selected after an exhaustive search of work on animal pose estimation [23]. In general, we found that animal pose estimation was not nearly as full featured as humanoid pose estimation. Deep Pose Kit and Deep Lab Cut are targeted at animal behavior and neurology researchers looking to track animal motions. Although Deep Lab Cut can create 3D tracking it required calibration of camera parameters using a standard checkerboard procedure. Such calibration matrices cannot be retroactively added to YouTube videos.

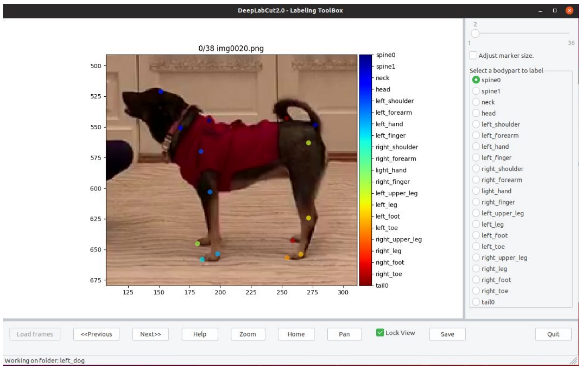
Alternatively, we could have used the Ani Pose wrapper to take care of this calibration requirement and handle using multiple cameras for the same motion, as a wrapper around Deep Lab Cut. However, we were unable to find such a dataset in the YouTube corpus. As a result, we created our own dataset, as we will mention shortly.

We also investigated Deep Pose Kit, another pose estimation algorithm tailored for animals. Deep Pose Kit trains and runs inference faster than Deep Lab Cut, it is limited to 2D point tracking. In the end, we decided that Writing a 3D quadruped pose estimator that used monocular video without calibration was beyond the scope of this project.

We were able to pull together a small dataset of stereo video from two friends, Chuya and her dog Kara, as well as Kenn and his dog Aka. Two smartphones were set up, a calibration checkerboard was filmed, and then the dogs did tricks, making sure to stay in-frame. Note the slight difference in framing below.



We are now in the process of annotating the dataset with ground-truth pose data and training a pose estimator to create a ground-truth for the full length of the videos.



Working on folder: left_dog

In parallel, we investigated existing mocap repositories for quadruped data. We found a few. For example, the Deep Mimic team themselves used a lion dataset which required paid commercial software. Free mocap data was available in c3d and trc formats only [2,24], which was not feasibly convertible into BVH format.

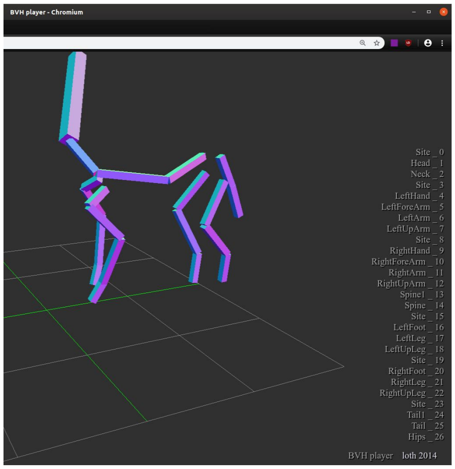


Figure 06: Canine mocap BVH data in a Chromium browser. [25,27]

Finally, we found ~50 mocap BVH canine files from the AI4Animation paper [25,26]. Previewing these in a Chromium browser [27], we found a pleasing variety of gaits but no tricks. Our steps would be to develop a convertor that would transform BVH files into the Deep Mimic animation format for dog skeletons. We could then train the Deep Mimic dog using these mocap files. This would require matching the dog skeleton with points in the BVH file.

Conclusion

To create pose estimates, using the abundance of publicly available dog videos is a promising alternative to the scarce motion capture data of quadrupeds. Animal pose estimation lags that of humans, so no 3D pose estimate from existing online videos is possible. But 3D reference motions are still easy to create by taking “stereo video” and we collect a few paired and calibrated videos (thanks to friends). Since the pipeline to go from video to the DeepMimic input format (local joint rotations) requires many steps (e.g. BVHToDeepMimic works for humans but must be rewritten for quadrupeds), as does installing and running the physics engine for DeepMimic, training a full policy is left for future work.

Next Steps

The ability to train physical robots from video is a powerful tool; imagine a world where pet robots could be easily customized with different personalities. Temperaments of different breeds or even individual dogs could be matched. This would be a significant leap forward in designing robots that have characters that humans easily relate to. These could then better assist the elderly population that may otherwise be unable to take care of pets, and studies have described decreased loneliness has a strong correlation to improved outcomes for many ailments. Of course this would require several steps beyond ones outlined in this paper; indeed, we are unsure if Deep Mimic PD controllers have ever been applied to physical robots successfully.

Figure 07: Non-humanoid system mimicking the motion of humans. [28]

Another interesting possibility is the ability to transfer characteristic motion between different body shapes, known as 'rig-to-rig' transfer in the world of animation. For instance, Pixar used this successfully for their popular 'lamp' animation [28]. The Deep Mimic paper showed us the possibility of training different robot morphologies to achieve similar reference motions.

Although we did not achieve some of the steps we thought would be easy, we learned a lot and had a lot of fun. Our main contribution is/will be a repository of calibrated stereo video of dogs doing tricks. We are also in the process of adding annotations, training a new DeepLabCut pose estimator, and creating a full pose estimate dataset for the entire length of each video. This dataset could be used in the future by turning it into a DeepMimic reference motion and applying it to the dog model to teach a robot dog to do tricks. The dataset can be found online at https://github.com/nouyang/cs249r_finalproject, and we welcome contributions.

References

1. "Functional Body Mesh Representation,, A Simplified ...." 1 Jan. 2016, http://www.naturalspublishing.com/download.asp?ArtcID=10858. Accessed 9 Dec. 2019.

2. "Library - Mocap Club." http://www.mocapclub.com/Pages/Library.htm. Accessed 9 Dec. 2019.

3. "Free Assets - Ziva Dynamics." https://zivadynamics.com/promos/free. Accessed 9 Dec. 2019.

4. "Teaching a Bop Bag to Stand Up with Simplified DeepMimic ...." 9 Aug. 2018, https://3deeplearner.com/teaching-a-bop-bag-to-stand-up/. Accessed 9 Dec. 2019.

5. "SFU MOCAP." http://mocap.cs.sfu.ca/. Accessed 9 Dec. 2019.

6. "Biovision BVH." https://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html. Accessed 9 Dec. 2019.

7. "CreativeInquiry/BVH-Examples - GitHub." https://github.com/CreativeInquiry/BVH-Examples. Accessed 9 Dec. 2019.

8. "SFV: Reinforcement Learning of Physical Skills ... - (Jason) Peng." https://xbpeng.github.io/projects/SFV/index.html. Accessed 9 Dec. 2019.

9. "SFV: Reinforcement Learning of Physical Skills from Videos." https://xbpeng.github.io/projects/SFV/2018_TOG_SFV.pdf. Accessed 9 Dec. 2019.

10. "akanazawa/motion_reconstruction: Motion ... - GitHub." https://github.com/akanazawa/motio

n_reconstruction. Accessed 9 Dec. 2019.

11. "DeepMimic: Example-Guided Deep ... - (Jason) Peng." https://xbpeng.github.io/projects/DeepMimic/index.html. Accessed 9 Dec. 2019.

12. "DeepMimic: Example-Guided Deep Reinforcement Learning ...." https://xbpeng.github.io/projects/DeepMimic/2018_TOG_DeepMimic.pdf. Accessed 9 Dec. 2019.

13. "xbpeng/DeepMimic: Motion imitation with deep ... - GitHub." https://github.com/xbpeng/DeepMimic. Accessed 9 Dec. 2019.

14. "Locomotion Skills for Simulated Quadrupeds - UBC Computer ...." https://www.cs.ubc.ca/~van/papers/2011-TOG-quadruped/index.html. Ac.cessed 9 Dec. 2019.

15. "Locomotion Skills for Simulated Quadrupeds - UBC Computer ...." http://www.cs.ubc.ca/~van/papers/2011-TOG-quadruped/paper.pdf. Accessed 9 Dec. 2019.

16. "Locomotion Skills for Simulated Quadrupeds - UBC Computer ...." https://www.cs.ubc.ca/~van/papers/2011-TOG-quadruped/index.html. Accessed 9 Dec. 2019.

17. "DeepLabCut — adaptive motor control lab - Mathis Lab." http://www.mousemotorlab.org/deeplabcut. Accessed 9 Dec. 2019.

18. "Using DeepLabCut for 3D markerless pose ... - bioRxiv." 24 Nov. 2018, https://www.biorxiv.org/content/10.1101/476531v1. Accessed 9 Dec. 2019.

19. "AlexEMG/DeepLabCut - GitHub." https://github.com/AlexEMG/DeepLabCut. Accessed 9 Dec. 2019.

20. "jgraving (Jake Graving) / Repositories · GitHub." https://github.com/jgraving?tab=repositories. Accessed 9 Dec. 2019.

21. "jgraving/DeepPoseKit: a toolkit for pose estimation ... - GitHub." https://github.com/jgraving/DeepPoseKit. Accessed 9 Dec. 2019.

22. "DeepPoseKit, a software toolkit for fast and robust animal ...." 1 Oct. 2019, https://elifesciences.org/articles/47994. Accessed 9 Dec. 2019.

23. "Deep learning tools for the measurement of animal ... - arXiv." https://arxiv.org/pdf/1909.13868. Accessed 9 Dec. 2019.

24. "Convert a C3d and TRC files to BVH - Blender Stack Exchange." https://blender.stackexchange.com/questions/30835/convert-a-c3d-and-trc-files-to-bvh. Accessed 9 Dec. 2019.

25. "sebastianstarke/AI4Animation: Bringing Characters to ... - GitHub." https://github.com/sebastianstarke/AI4Animation. Accessed 9 Dec. 2019.

26. "AI4Animation/ReadMe.txt at master · sebastianstarke ... - GitHub." https://github.com/sebastianstarke/AI4Animation/blob/master/AI4Animation/Assets/Demo/SIGGRAPH_2018/ReadMe.txt. Accessed 9 Dec. 2019.

27. "BVH player." http://lo-th.github.io/olympe/BVH_player.html. Accessed 9 Dec. 2019.

28. "Mapping Rigs-to-Rigs with Neural Nets - 3DeepLearner.com." 29 May. 2019, https://3deeplearner.com/mapping-rigs-to-rigs-with-neural-nets/. Accessed 9 Dec. 2019.

29. "Dene33/video_to_bvh: Convert human motion from ... - GitHub." https://github.com/Dene33/video_to_bvh. Accessed 9 Dec. 2019.

30. "bvhtodeepmimic · PyPI." 8 Aug. 2019, https://pypi.org/project/bvhtodeepmimic/. Accessed 9 Dec. 2019.

31. "BartMoyaers/BvhToDeepMimic: Convert .bvh files ... - GitHub." https://github.com/BartMoyaers/BvhToDeepMimic. Accessed 9 Dec. 2019.

32. “xbpeng/DeepMimic: dog3d.txt” https://github.com/xbpeng/DeepMimic/blob/master/data/characters/dog3d.txt. Accessed 9 Dec. 2019

Appendix

Dog trick videos on youtube:

- Wave:
  - https://www.youtube.com/watch?v=lzYYaQeB9zE
  - https://www.youtube.com/watch?v=o3_QnUuVadE
  - https://www.youtube.com/watch?v=X4GYhIa648s
- Take a bow:
  - https://www.youtube.com/watch?v=KKr5gBvyjzM
  - https://www.youtube.com/watch?v=Qszx6laEzoU
  - https://www.youtube.com/watch?v=pT7-CHam2HU
- Shake paw:
  - https://www.youtube.com/watch?v=G3-hec29wII
  - https://www.youtube.com/watch?v=CRoDTUkzVpU
  - https://www.youtube.com/watch?v=iPCCHGxL_Gk

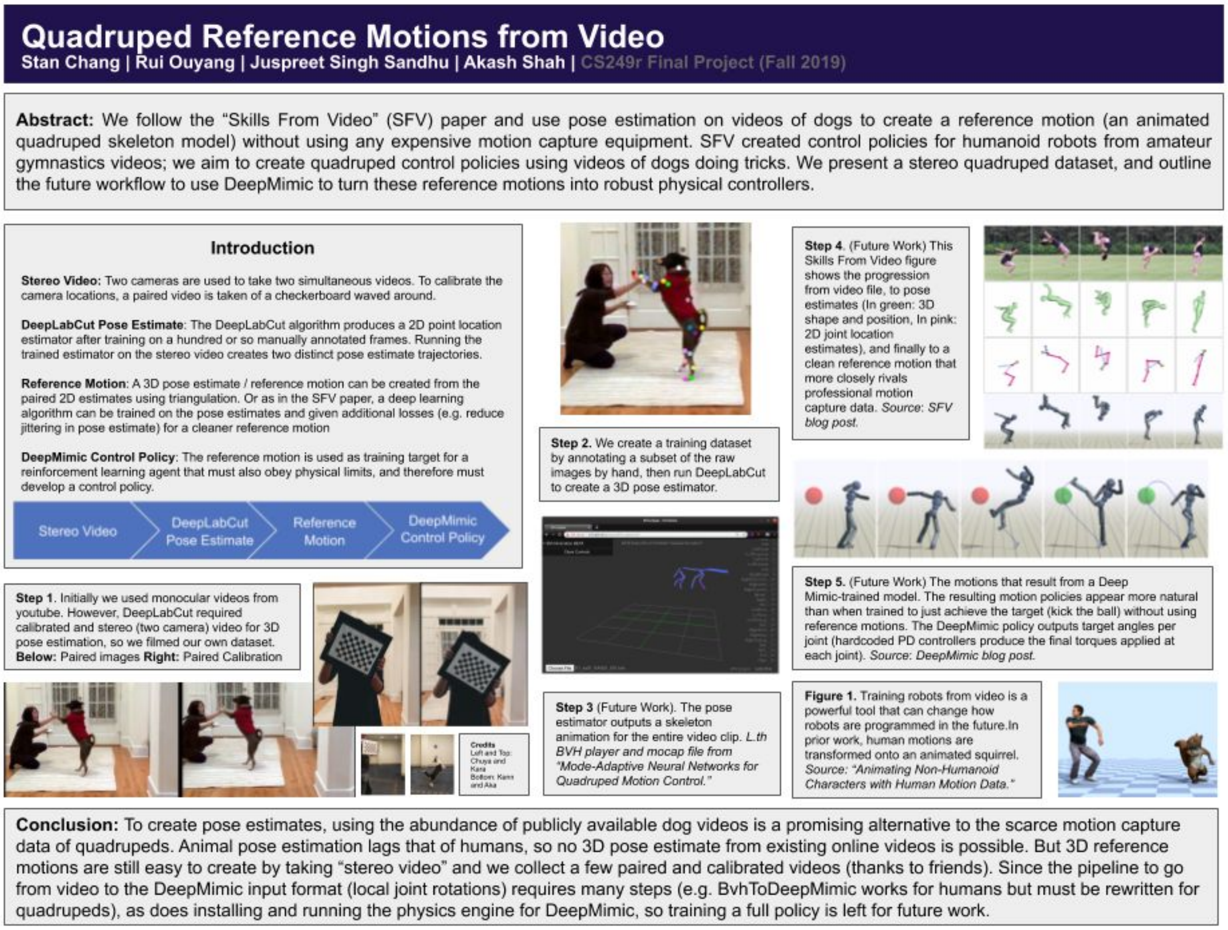
Our dataset:

https://github.com/nouyang/cs249r_finalproject

Appendix (cont'd)

Final Poster:

https://github.com/nouyang/cs249r_finalproject/blob/master/poster.png



Quadruped Reference Motions from Video

Stan Chang | Rui Ouyang | Juspreet Singh Sandhu | Akash Shah | CS249r Final Project (Fall 2019)

Abstract: We follow the "Skills From Video" (SFV) paper and use pose estimation on videos of dogs to create a reference motion (an animated quadruped skeleton model) without using any expensive motion capture equipment. SFV created control policies for humanoid robots from amateur gymnastics videos; we aim to create quadruped control policies using videos of dogs doing tricks. We present a stereo quadruped dataset, and outline the future workflow to use DeepMimic to turn these reference motions into robust physical controllers.

Introduction

Stereo Video: Two cameras are used to take two simultaneous videos. To calibrate the camera locations, a paired video is taken of a checkerboard waved around.

DeepLabCut Pose Estimate: The DeepLabCut algorithm produces a 2D point location estimator after training on a hundred or so manually annotated frames. Running the trained estimator on the stereo video creates two distinct pose estimate trajectories.

Reference Motion: A 3D pose estimate / reference motion can be created from the paired 2D estimates using triangulation. Or as in the SFV paper, a deep learning algorithm can be trained on the pose estimates and given additional losses (e.g. reduce jittering in pose estimate) for a cleaner reference motion

DeepMimic Control Policy: The reference motion is used as training target for a reinforcement learning agent that must also obey physical limits, and therefore must develop a control policy.

Step 1. Initially we used monocular videos from youtube. However, DeepLabCut required calibrated and stereo (two camera) video for 3D pose estimation, so we filmed our own dataset. **Below:** Paired images **Right:** Paired Calibration

Credits
Left and Top: Chuya and Kara
Bottom: Kami and Aka

Step 2. We create a training dataset by annotating a subset of the raw images by hand, then run DeepLabCut to create a 3D pose estimator.

Step 3 (Future Work). The pose estimator outputs a skeleton animation for the entire video clip. *L.th BVH player and mocap file from "Mode-Adaptive Neural Networks for Quadruped Motion Control."*

Step 4. (Future Work) This Skills From Video figure shows the progression from video file, to pose estimates (In green: 3D shape and position, In pink: 2D joint location estimates), and finally to a clean reference motion that more closely rivals professional motion capture data. *Source: SFV blog post.*

Step 5. (Future Work) The motions that result from a Deep Mimic-trained model. The resulting motion policies appear more natural than when trained to just achieve the target (kick the ball) without using reference motions. The DeepMimic policy outputs target angles per joint (hardcoded PD controllers produce the final torques applied at each joint). *Source: DeepMimic blog post.*

Figure 1. Training robots from video is a powerful tool that can change how robots are programmed in the future.In prior work, human motions are transformed onto an animated squirrel. *Source: "Animating Non-Humanoid Characters with Human Motion Data."*

Conclusion: To create pose estimates, using the abundance of publicly available dog videos is a promising alternative to the scarce motion capture data of quadrupeds. Animal pose estimation lags that of humans, so no 3D pose estimate from existing online videos is possible. But 3D reference motions are still easy to create by taking "stereo video" and we collect a few paired and calibrated videos (thanks to friends). Since the pipeline to go from video to the DeepMimic input format (local joint rotations) requires many steps (e.g. BvhToDeepMimic works for humans but must be rewritten for quadrupeds), as does installing and running the physics engine for DeepMimic, so training a full policy is left for future work.