Motion Capture Sports Tutor

Project Proposal

Third Revision

03/22/2018

Jonathan Fermin
Jonathan Meade
Stacia Near
Christoph Uhl
Michael Vienneau

CSCI-4830 IML4HCI

Prof. Ben Shapiro

Instr. Will Temple

Proposal Contents

Idea and Mission Statement

We will create a virtual personal trainer that observes your performance of correct motions under a coach's supervision, and helps you replicate them later.

Many people are interested in improving their performance at sports techniques, everything from dancing, to boxing, to yoga. Usually this requires expensive, dedicated sessions with a personal trainer, however with our method, the price for improving sports techniques would be more affordable. We seek to develop a method that takes advantage of the skills and insights of professional coaches, but delivers this service directly to your home studio at a low cost.

Using an inexpensive motion capture studio setup, users can compare their motions to the correct form. The idea is to set up the motion capture sensors (3-6 inexpensive cameras) while a coach is present, and have the cameras extract skeletal data to observe the user's correct form under a coach's supervision. The camera input is used to train a computer system using dynamic time warping and classification, to build a database of correct movements. This is an excellent application for machine learning, as gesture detection is a complex problem that is difficult to solve when programming by hand. However, by using machine learning, we can quickly train a system to recognize a wealth of different sports movements.

Another advantage of such a system is its ability to recognize how closely a given movement matches the target movement. The system can give the user live feedback in the form of a precise percentage representing how well they replicated the correct motions. Additionally, instead of forcing the system to guess which movement the user is going for, we will use speech recognition to trigger the system to look for a particular gesture. For instance, let's say we install a more advanced version of the system on a basketball court. The user selects "freethrow" from a UI interface, and the system knows to spend the next minute or so analyzing their movements for this gesture. Rather than confusing the movement for similar throws, the system will know exactly what to look for.

This is a huge benefit for consumers, who can train the system while a coach is present, perhaps during a group lesson. Having the coach provide individual lessons is too expensive, so our tutor will provide follow up training at home in the coach's stead. Because machine learning provides a high level of flexibility in terms of allowing you to retrain the model at your leisure, the user can continue to improve the model that the system considers "correct" each time they visit the coach. This way, the user will be able to see how they consistently improve, and each time their technique gets better, they simply focus on maintaining that improvement until they have a chance to further refine it.

Of course, one of the problems with this is the chance that the user, even with a coach present, may be exhibiting incorrect form. To make our system customizable, we are having it train on the user exhibiting correct form, and relying on that information to help them reproduce the form later on. However, it may be difficult for a beginner to display form that is, in any sense of the word, correct. We don't want to help them form bad habits by having them reproduce their own incorrect movements.

So, the system will be two-tiered. In order to enforce customizability, the user will have the option to record their own movements under a coach's supervision. However, in order to prevent bad habit formations, a model will also be trained on professional movements. It will be more difficult for a user to match these movements perfectly, but it will be a great way for them to measure themselves in comparison to a consistent standard.

Our initial goal is simply to provide auditory feedback on how well the user is performing. Our stretch goal is to create a full blown “virtual coach,” who represents the model gesture. Your movements are motion captured and matched onto a virtual avatar. The virtual avatar moves in real time, and acts as either a mirror, or a way for the user to watch themselves from behind, or from a new angle - depending on their preference settings. Standing next to the user’s virtual avatar (or overlayed over it in three dimensional space), the virtual coach will display the correct movements for the user to follow along with and try to match. In “personal” mode, the virtual coach will display the user’s best attempt at the motion so far, as chosen by a coach from the user’s attempts. In “pro” mode, the virtual coach will replicate the movements of a professional. Using regression or other strategies, we can map from the user’s body type to the professional’s body, to get a more accurate measure of how well the user is copying the professional. This method will still take into account their unique body type and movement style.

Overall, this system will be a huge benefit to hobbyists who don’t have the resources to work with a personal trainer regularly, and professionals who want to practice daily at home while having a good standard to measure themselves against. Paying a professional once to give motion capture data to the system and then distributing it to potentially hundreds of users through a virtual coach is a modern, innovative way to revolutionize personal training.

This project is within reach for us as a small classroom team, because inexpensive in home motion capture is a largely solved problem. Many indie game developers are very interested in this technique, because it allows them to speed the production of their animations. We plan to use the inexpensive motion capture software, Ipisoft, which was developed for this exact purpose. It supports an array of 3-6 PS3 Eyes on tripods as an input setup, and these inexpensive cameras are only $7 each. The PS3 Eyes are a particularly fortunate find, as they were developed for gesture recognition for games, and come with built in microphone arrays for voice commands. They are still readily available for purchase. Ipisoft is marketed towards small teams (who can’t afford the usual multi-thousand dollar motion capture setups), so it is priced reasonably as well. It comes with a 30 day free trial, and is $165 for a three month lease. Overall, our system would be relatively easy and inexpensive for potential customers to set up.

The project is also sufficiently complex for a final project. At its best, the project could be extremely difficult, but it should not be hard to find a balance, as the goal for the end of the semester is simply a working prototype. If difficulties are encountered, there are many ways in which we can simplify the setup. For instance, we will design the system with the intent to customize it to any sport. However, we will simply focus on one that our team is familiar with throughout development. We have discussed using basketball or boxing as our sport of focus, most likely the former because of our familiarity with the techniques. Ideally we would have a coach train with us, but for the purposes of development, or own insights into technique will be sufficient. If we push this to release, we can bring in a coach.

We are excited and hopeful to propose this system for our CU Boulder Interactive Machine Learning Final Project. Please review the details below, and let us know your thoughts.

Challenges

Motion capture requires information greater than the majority of the feature extraction used during our previous coursework. Accelerometers and gyroscopes are generally good for objects with single joints, however motion capture requires more detailed information about 3d space. Therefore, the majority of the challenges will be around gathering our input data, whether that be new technologies, noise, or latency.

Latency will probably become an issue with such a large amount of data and multiple wireless cameras. Possible approaches to this will be to make a wired connection or to abstract the motion capture data into only the important joints. The speed of the actual computer vision library will be a factor as well. Noise is a problem in every project, and the best approach for motion capture can be to make a dedicated rig/suit for people to wear and practice in. This may be outside of the scope of this project, so trying different dynamic time warping algorithms should be the priority when working with markerless motion capture.

The group may have to learn how to use OpenCV, an open source computer vision software, if we choose to create our own motion capture algorithms. The upside to this is that OpenCV has applications with C++, a language everyone knows. OpenCV claims computational speed, so hopefully the challenge of optimization can be solved through decreasing latency. Ipisoft is a motion capture application that can help break through the challenge of interpreting the video and turning it into skeleton data. In order to offer the user feedback, a 3D model will be output in Blender, showing how someone can improve their technique.

One of the more intangible challenges would be related to the philosophy behind basketball and/or boxing. Even at the professional level, a consensus will never arise between what the perfect form is. Therefore, this project’s scope will be limited to whatever the expert says. Eventually, through a genre analysis of basketball shots or defensive boxing techniques, there can be trainers for multiple styles.

Technologies

The main hardware that we will require is an array of 3-6 PS3 Eyes which are small inexpensive cameras dedicated to motion capture along with (possibly) micro bits for added motion data. The PS3 Eyes can be used with the closed source software, Ipisoft. Although this requires payment, it may be feasible for our timeline with the 30 day free trial to see if we like the technology. In the event that this technology does not pan out, we can use OpenCV and C++ to develop our own Motion capture framework for the PS3 eyes or any array of cameras.

The micro bits will be used mainly to send accelerometer data from from the athlete to a central server parsing incoming OSC messages. For this, a variety of tools are available on the wekinator site as well as the class github repo, where we have mostly all experimented with micro bits in one way or another.

To get help, we can look towards documentation for Ipisoft and OpenCV. On top of this, we can look at academic papers on using OpenCV with motion capture and gestural recognition.

For the proof-of-concept, we are hoping to have motion capture working in a way that can send the motion data to a server. We have agreed that this is the hardest part of our project, and as such, is what we are working on first to overcome. If we can get motion capture to work fully, the rest of the project can be dedicated to modeling the motion and adding extra sensors for more information as needed.

Risks

Our project poses several risks and many avenues with which our system could incorrectly identify an individual's form and pose. The largest overall source of failure for our project comes from the motion capture system. Motion capture, especially without advanced equipment and sophisticated, expensive software, is difficult to implement and severely error-prone. While open-source computer vision solutions have gotten more accurate over time, we still face the risk that our system will erroneously identify a user's position. In order to minimize this risk, we will be collecting larger amounts of data than we have for any of our other projects with the overall hope that the increased dataset will allow us to train an increasingly accurate model.

Along with this increased accuracy, however, comes the dangers of overfitting. While we will be implementing a user configuration dialog into our system, we will also be implementing non user-specific data based on our own movements or the movements of the personal trainer we consult. The potential that we will overfit our data to our own body shapes could very likely be high. However, we hope to minimize this risk by using a variety of volunteers and by attempting to shape most of our observations and recordings around each user's specific configuration. During the user configuration setup phase, we will take the largest number of measurements that we can within a reasonable time frame and use that information to fine-tune our model.

Another risk we face arises during the training and testing phase of our project. None of our team members have experience with professional personal training or coaching, therefore we will most likely need to find either online resources on proper athletic form or discuss these topics with a professional in-person. Unless they are willing to donate their time and expertise, time spent consulting a professional will likely be very costly, so we will attempt to gather as much free data as possible in order to minimize the risk of errors appearing in our training set while still keeping initial costs low. This data can be taken from any of the numerous free online exercise videos and tutorials available and can be integrated into out system using ourselves or volunteers to emulate these poses. However, this method of pre-user configuration training will require quite a bit of time and effort to implement.

In a worst-case scenario, our project may actually pose a health hazard if it provides feedback which places the user in a position that threatens their overall well-being. For example, if the system consistently mis-identifies a user's pose and suggests a method to "fix" it which actually proves damaging in the long-term to their spine or back, our project could be rightly blamed for causing an even larger issue than simply their lack of proper posture. Besides the obvious physical harm to the user, any published version of our system could result in a liability to ourselves and our team.

Ethical Discussion

One of our greatest concerns for this project is the need for our system to store personally identifiable information about each user, some of which may be health related due to the nature of our program's design. Even the most seemingly inconsequential health information is governed under strict information storage laws and our data storage policy will have to reflect that. In order to make sure that this information is stored as securely as possible, we will use the most up-to-date database and storage software available. We will also follow a variety of database security checklists put out by the program's developer and other entities, such as NIST, which provides up-to-date information security tips to the public at no cost.

Another large concern is mentioned briefly in the risks section above and involves the possibility of injury to the user while using our system. This possibility, while small, is inherent in any exercise system. In almost all cases, however, the responsibility for the user's safety for such systems rests solely with the user. In order to help our users make educated decisions about when they should stop using the motion capture system or take a break from using it, we will include information with our final product warning potential users about indications that they should cease to use it. These indicators include signs such as physical discomfort, nausea, dizziness, etc. Disclaimer statements such as these are common in commercial exercise systems and will prove valuable to our system in order to protect our team from liability.

Components

Open Source

- Python
- Associated machine learning libraries (SkLearn, etc.)
- Team-developed code
    - Feature extraction and smoothing
    - Data handling and passing
    - Microbit data handling

Closed Source

- Microsoft Kinect and associated software components
- Sony PS Eye

Timeline

Phases	Tasks	Hrs	Due Dates
1 - Concept Phase			
1.1	Choose which use cases to satisfy (Athletic or Accessible)	1	02/05/18
1.2	Finalize which sport to work on	1	02/15/18
2 - Planning Phase			
2.1	Post project proposal on the peer review website	12	03/12/18
2.2	Revise proposal based on feedback. Then, expand your proposal by writing stories about scenarios your users would enact. Make storyboards and sketches to go with them. Start building a very rough prototype that demonstrates the feasibility of your approach, including feature extractors, machine learning approach, actuators (e.g. synthesizers, motors, graphics, etc.), and user interface customization paradigm. All code must live in github. If possible, please also post runnable builds as releases.	12	03/17/18
2.3	In-class status update -- what's happening?	2.5	03/19/18
3 - Design & Development Phase			
3.05	Optional spring break work - get MoCap to work and set up trainer algorithm	20	03/26/18
3.1	Prepare to demo prototype in class - MoCap running and gives audio feedback, first prototype of trainer algorithm. **Have a proof of concept!**	20	04/02/18
3.2	Prepare to demo prototype in class - second prototype of trainer, integrate the working MoCap with a refined algorithm, trainable on any user	20	04/09/18
3.3	Legit **hands-on** demo in class - Any idiot should be able to train this third trainer algorithm themselves	20	04/16/18
3.420	Prepare for end of semester rush	420	04/20/18
3.5 - *** INITIATE FINAL PHASE ***			
3.6	Post draft of final report on the peer review website -	20	04/23/18
3.7	Final Presentation - Prepare our presentation, who will talk about what?	20	04/30/18
3.8	Final Project done - Polish up the baby	20	05/04/18

Minimum Viable Product

There will be multiple parts to the minimum viable product that we expect to deliver for **initial demos**. This is not and is not meant to be a complete list of all goals we hope to accomplish.

Input Hardware

- Kinect for overall joint information.

Initial capture goals

- Basketball free-throw.
- Golf swing.

User interface

- Described in detail in the storyboard below.

User feedback

- Match confidence percentage displayed as text in the “user” mode of the UI.

ML model

- DTW
- Python processing with SkLearn.

User Interaction Storyboard

The overall design and mockups of our UI can be found here.
A screen-by-screen flowchart overview matching the UI mockups can also be found below.
Example Coach/User interaction:

User: "Hey coach, can you help me set up the motion capture to help me improve my form while I'm not there?"

Coach: "No problem, what do I do?"

User: "I don't know, let's start up the application.."

[starts up application]

Application: "Thank you for choosing MoCap Trainer! To start, choose either 'Record New Motion' or 'Train a Motion'. "

[screen shows 2 buttons 'Record' and 'Train' -- if no previous recordings, train is greyed out]

[coach selects 'Record']

Coach: "What now?"

Application: "Please add a new motion"

[List of clickable 'plusses' appears to indicate function to add a new motion -- if previous motions have been recorded, they will be listed here]

Coach: "Okay I will add a new motion."

[Clicks a plus, the application asks for a name-input]

Coach: "Let's call it 'Basketball free-throw'"

[Presses enter, UI window for calibration pops up]

Coach: "Okay, stand in the middle in front of the black curtain and spread your arms out."

[User spreads arms out]

Coach: "Hold for 3 seconds."

[Calibration window is replaced by Coach start/stop screen]

Application: "Successfully calibrated! In order to record the perfect motion, the coach needs to press 'start' when the user performs the desired motion, when the motion is over, press 'stop'. Give some buffer time, I will be able to find the motion."

Coach: "Alright, please get into position"

User: "Okay"

[User tries best to perform the motion correctly]

Application: "If the motion was performed well and you think it can be used as training data, please click 'add', otherwise, please click 'repeat'. We will do this 10 times [progress bar pops up], the motions added will work as training data, so please make sure they are correct."

Coach: "Hmmm, I think you need to keep your elbow closer to your body, try again"

User: "Okay"

[Tries again]

Coach: “Perfect, we will use this one! Onto the next, ready?”

. [Continues until done]

.

.

[User by themselves now]

User: “Time to do some training!”

[Starts up application]

Application: “Thank you for choosing MoCap Trainer! To start, choose either ‘Record New Motion’ or ‘Train a Motion’. ”

[screen shows 2 buttons ‘Record’ and ‘Train’ -- if no previous recordings, train is greyed out]

[User chooses ‘Train’]

Application: “Choose the motion you’d like to train.”

[User chooses ‘Basketball free-throw’]

Application: “Okay please calibrate”

[User stands with hands spread out]

[Countdown initiates: “3… 2… 1… Click! Calibrated!” is hearable in the background]

Application: “Please yell ‘now’, to start, and ‘stop’ to stop. Don’t worry about timing, I will be able to recognize your motion”

[Performs motion]

[Percentage of accuracy is displayed on the screen and remains there]

Application: “Good motion, but try to keep your elbow closer to your body!”

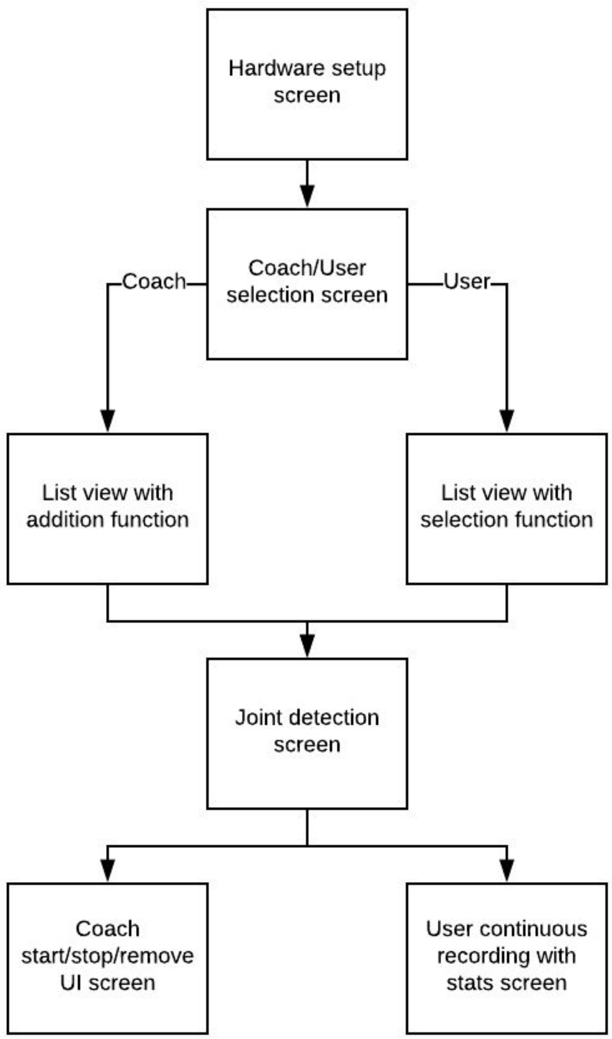
[User performs another motion]

[Percentage updates]

User: “Wow! This works so well! Thank you MoCap trainer!”

[Continues to train until the end of days]

User Interface Flow



Example Personas

The main audience of our project is individuals who want to improve their skills in a particular sport, but do not have the financial resources or the scheduling flexibility to meet with a personal trainer or coach on a regular basis. These individuals may have some experience with the sport already, but it is not required because of the nature of the training mechanism. Due to the user-oriented nature of the setup, our system is applicable to a very wide range of individuals with little to no requirements in regards to athletic ability, age, gender, etc. Users could range from teenagers eager to improve their free-throw style to senior citizens looking to stay active while using specific motions to improve their physical health. The fact that our system contains no pre-trained information makes it perfect for anybody looking to improve their sports posture. In order to train the model on a specific user, they should go through the motion with a coach/trainer. This will enable the system to custom-tailor a solution to each user individually. This will still be cheaper than regularly meeting with a trainer, as the user can decide when the model needs proper updating with a coach (the user can choose to never retrain the model, or to retrain every day → more customizability).

Research Bibliography

Live Document Link

Grove, R. (2012, October). *Ipi soft dmc version 2.0: Affordable motion capture for everyone.* Retrieved from https://www.renderosity.com/ipi-soft-dmc-version-2-0-affordable-motion-capture-for-everyone-cms-16365

Ipisoft. (2014, February). *Quick start guide for multiple playstation eye cameras configuration.* Retrieved from http://wiki.ipisoft.com/Quick_Start_Guide_for_Multiple_PlayStation_Eye_Cameras_Configuration

Ipisoft. (2018). *Ipi soft motion capture for the masses.* Retrieved from http://ipisoft.com