Human Pose Estimation using Google Tango

Victor Vahram Shahbazian
Assisted: Sam Gbolahan Adesoye
Co-assistant: Sam Song
March 17, 2017
CMPS 161 – Introduction to Data Visualization
Professor Alex Pang

Abstract

For the first time in history, through the advancements of AR technology, it is now possible to track and understand objects in real space using a Tango enabled Android device. For the purpose of simplicity, this project focuses primarily on tracking and understanding human poses by training a machine learning algorithm to understand what shapes in real space are human.

1 Introduction

1.1 Motivation

The rapid advancement of mobile technology has allowed for many new forms of applications arise, bringing new ways a handheld device can entertain, connect, and learn. Now yet again the mobile platform is advancing, allowing for a new range applications to be developed. Unlike previous iterations of applications, the technology progressing the advancement of the platform is Augmented Reality.

Google has developed a groundbreaking new field of technology by incorporating a depth sensing camera with their Android based mobile devices. The technology is called Google Tango, and through Lenovo, has already developed a consumer grade mobile devices. There are many applications already available that can observe static objects in real space. The objective of this project is to expand Tango's understanding to organic moving objects, such as humans, in real time. This would allow many developers to use this technology to create new applications for the mobile platform in ways that are yet to be seen. The ability for mobile platforms to be able to observe organic moving shapes can bring many changes in the way that people interact with their phones. As a simple example, with Tango it is possible to create an application that overlays the human anatomy onto a person. This could be used in the medical field allowing a real-time demonstration of x-rays or any other form of internal scan. To conclude, there is a reasonable amount of motivation to the development of Human Pose Estimation through Google Tango as this technology has never been available to the extent that it would be once this project is completed.

1.2 Related Works

[] Microsoft's research paper on how they implemented Human Pose Estimation using the Xbox Kinect. Much of the ideas for our implementation comes from this paper. The similarities of the paper to our project include the use of a depth sensing camera, the use of Random Forest Decision Trees, and the generating of synthesized data to train the trees.

[] Sam's research paper on Human Pose Estimation. Much of the information provided within this paper is also available from Sam's paper as Sam has already done research in regards to developing a solution for real-time pose estimation using a Tango enabled device.

[] A paper containing an implementation of key point detection that can run in real time on a standard computer. The paper describes collecting intensities along a circle with the currently analyzed pixel x at the center of the circle. The radius of the circle is a chosen value with a set of THETA's allowing for the selection of diametrically opposing points.

2 Technical Detail

There are several parts to the implementation of this project. To get to the point where a Tango enabled Android device can understand human poses, several steps need to be taken. These steps include researching on several deep learning frameworks and trying to port them to android, labeling body parts of thousands of different poses, using the labeled body parts to train the deep learning algorithm, and finally connecting the program to Tango to allow the machine learning algorithm to estimate human poses in real time.

The true proponent that makes this project possible is the ability to compare the difference of distance between pixels. Depth sensing cameras seem to be a relatively new technology which has not hit a mainstream level of understanding. Microsoft was the first to bring this technology to consumer level by expanding their Xbox gaming console to include the additional hardware known as the Kinect. [] Their research paper describes in detail their implementation of Human Pose Estimation. The limitations of this technology coming from Microsoft is that it is only available through a gaming platform limiting its use by mobility and genre. With the release of Google Tango, these limitations are now lifted. Smart phones with depth sensing cameras are now available for purchase thus altering the reason for owning a depth sensing camera as well as the convenience of it being mobile.

Random Forest Decision Tree
We have decided to use a Random Forest decision tree because it is both great for making a decision on multiple features and for being able to run efficiently on a GPU. The way the Random Forest tree works is that for its input, given pixel x, a set of 15 features are selected and processed, resulting in an output of 15 different percentages determining whether pixel x is indeed a part of a human pose and which of the 25 labeled body parts it is.

Based on a predetermined patterns it takes in a feature and tries to match it with a pattern. This results in a percentage of how similar it is to previously trained data. The value that is being judged is the maximum difference in offset between differing pixels. These are the 15 features between pixels. At each point of the tree all 15 are being compared to and decided on based on a given threshold. Depending on whether it is above or below a given threshold, the tree then transitions left or right. When the tree reaches a leaf node,

Depth comparison feature.

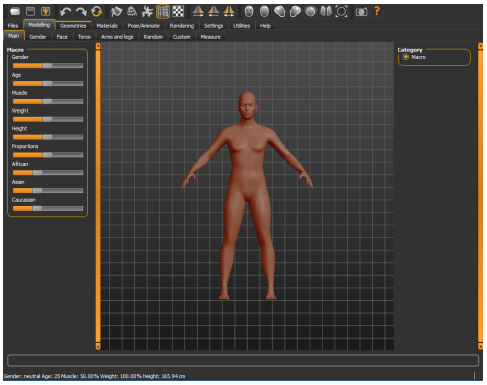
Where is the depth pixel x in image I. The equation is made to be depth invariant meaning that the depth of pixel x is being used to normalize the offsets u and v.

Determining which part of the body the pixel is coming from allows for the localization of joints, discussed further down in this paper. To make the algorithm knowledgeable about human poses, thousands of labeled human poses need to be used as training data for the algorithm. This topic is further discussed within the section Labeling.

Once the Random Decision Trees are trained, they can then to be used to estimate the differences of gray scale at a radial distance away from the current pixel being processed.

Labeling

Training the algorithm requires a vast amount of depth images with a person in front of the camera with various poses. If this were to be done manually it would take a considerable amount of time. Due to the abundance of graphical tools available, it is much faster to simply create and render 3D models of humans. Several tools are required to accomplish this, fortunately all are free to use. The two major applications in use for creating human models are Make Human and Blender. They don't work intuitively, but with a simple plugin added to blender, the port became seamless.

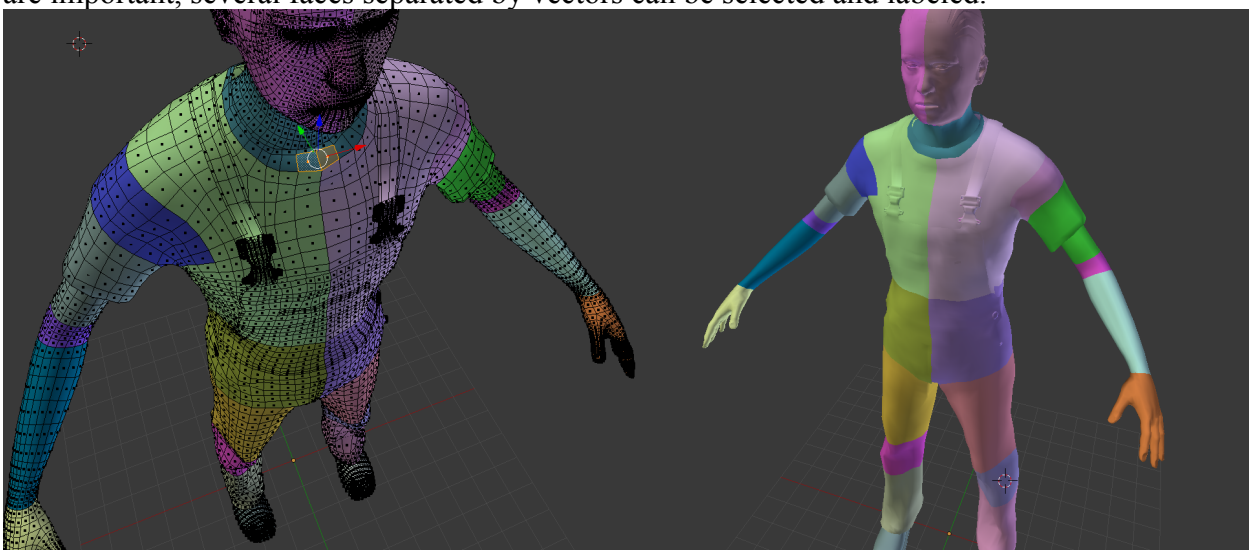


This first of these applications is Make Human.

The algorithm, there are several parts. First part vector features. For each pixel of each frame adjacent pixels are selected radially at a distance of r.

It is a simple tool used to generate a human model. When the application is first run, a blank model is generated. To create different types of bodies, the attribute of each model is tweaked creating people of different types such as a man, a woman, a little girl, a fat man, an old lady, and so on. The more diverse the body types are the better the machine can be trained.

The next tool used for training the algorithm is Blender. Blender is 3D graphics and animation software. Blender makes it easy to import the models created by Make Human. Once the models are important, several faces separated by vectors can be selected and labeled.



To train the algorithm we have labeled 25 body parts. This includes the head, neck, shoulders, elbows, top of arms, bottom of arms, hands, feet, knees, top of legs, bottom of legs, top left and right of body, and bottom left and right of body. The final step is to render these models as images in different poses, to do this we use CMU MoCap scripts.

Results

The project has yet to be completed. In its current phase, the Random Forest tree is being trained by using our labeling data. The process has resulted in several failures in the form of hardware and driver issues impeding the progress of development.

Sam used the resource http://citrisdance.soe.ucsc.edu and was able to compile and test a basic example of human pose estimation. Upon completion he then tried to train the algorithm using the data we generated. This lead to a disk quota issue as the size of data required for training exceeded the limitation provided by Citris Dance.

Another issue that arose was the due to the under-powered graphics cards being used by Sam to train and test the data. The card he was previously using only had 386 stream processors. To resolve the inefficiency of the GPU, a new GPU was purchased, an AMD RX 480 RS which has 2048 stream processors, 8GB of GDDR5 video ram, and a core clock speed of 1288MHz. The card has yet to be used, but the specifications of the card seem promising and will certainly provide faster results than the previously used GPU.

With the new card, another problem came about with the driver support on the Linux platform for that particular card. Using the card on the Linux system resulted in errors that were not previously present and would break at seemingly random points. The conclusion to this problem is that there is a driver error. To solve this problem, Sam has proposed to set up a new Operating System, presumably Windows, and then run the compilation again.

Conclusion

Working on the project has given me a better sense of what it takes to develop a new innovative piece of software. My expectations for assisting Sam where a bit misaligned with reality as previously I had no conception of what it would take to estimated posing using a depth sensing camera. Furthermore, this project has been my introduction to machine learning algorithms as it is the Random Forest Tree which allows to build a system that looks through thousands of different data sets to come to a reasonable conclusion on the various shapes of a human in space.

Along with the machine learning algorithm, the depth sensing camera is the second ingredient that makes this project possible. The reason why an RGB camera does not work is because sometimes the distinction of color between a person and a background object can be negligible, making it hard to distinguish the boundaries of a person. With the depth sensing camera, this is no longer as issue as the depth of objects in space have clear boundaries so long as they are not directly next to an object of the same distance.

Another failed expectation is of how sporadic the collection of data really is. Watching Sam do the actual research revealed the level of uncertainty that arose from need to complete the project in the best way possible. Much of the uncertainty I witnessed was directed towards understand the best deep learning framework to use in conjunction with Android. At the time of writing this paper, I was informed by Sam that he concluded on using OpenCL with a library known as Padenti.

Overall this project has been an interesting experience and one I am glad to have been a part of.

References

[1]Microsoft Research Paper https://www.microsoft.com/en-us/research/publication/real-time-human-pose-recognition-in-parts-from-a-single-depth-image/

[2]Microsoft Video https://www.youtube.com/watch?v=QPYf6pXe_4Q&t=1949s

[] Sam's Paper
https://classes.soe.ucsc.edu/cmps261/Fall15/projects/gadesoye/final/AdesoyeSamuel.pdf

[] Towards Recognizing Feature Points using Classification Trees, EPFL Technical Report IC/2004/74. Focused on chapter 5, Key Point Detection.

[] Explanation of cnnDroid https://arxiv.org/pdf/1511.07376v2.pdf

[] Weka Tutorial http://machinelearningmastery.com/how-to-run-your-first-classifier-in-weka/

[] GPU acceleration of random forest trees http://www.ais.uni-bonn.de/theses/Benedikt_Waldvogel_Master_Thesis_07_2013.pdf

[] OpenCV Random Forest Tree http://docs.opencv.org/2.4/modules/ml/doc/random_trees.html#

[] CNNdroid for porting of Caffe to Android https://github.com/ENCP/CNNdroid

[] Link to Padenti https://github.com/mUogoro/padenti

[] Link to main page of OpenCL https://www.khronos.org/opencl/