Report Number: WUCSE-2012-37

2012

Building a Skeleton of a Human Hand Using Microsoft Kinect

Jed Jackoway

The goal of the project was to reconstruct the skeleton of a Microsoft Kinect user's hand. Out of the box, Kinect reconstruct the skeleton of users' bodies, but it only does large joints, such that the hand is given a location on the general skeleton, but the specifics of the fingers and fist are not actually calculated.

Department of Computer Science & Engineering

Washington University in St. Louis
School of Engineering & Applied Science

2012-37

Building a Skeleton of a Human Hand Using Microsoft Kinect

Authors: Jed Jackoway

Abstract: The goal of the project was to reconstruct the skeleton of a Microsoft Kinect user's hand. Out of the box, Kinect reconstruct the skeleton of users' bodies, but it only does large joints, such that the hand is given a location on the general skeleton, but the specifics of the fingers and fist are not actually calculated.

Type of Report: MS Project Report

Jed Jackoway

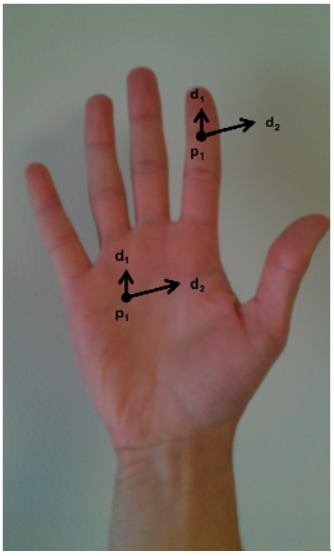
Building a Skeleton of a Human Hand Using Microsoft Kinect

The goal of the project was to reconstruct the skeleton of a Microsoft Kinect user's hand. Out of the box, Kinect reconstruct the skeleton of users' bodies, but it only does large joints, such that the hand is given a location on the general skeleton, but the specifics of the fingers and fist are not actually calculated.

The uses for this are fairly widespread: the most apparent use would be to improve Kinect as a gaming device. With the hand skeleton in addition to the general body skeleton, games could factor in fine finger motion to gameplay. Tracking fingers would also allow Kinect to do many other things that it currently cannot, such as building user interfaces that are used through gesture control. We also though that the technology would allow for games to be created which encouraged patients with limited hand mobility to complete their rehabilitation exercise.

In order to do this, we decided to emulate the process Microsoft used to get the Kinect to track whole human skeletons. The general process is to look at each pixel in a depth image from Kinect and label that pixel with the part of the body it belongs to. From there, they use all of those labels to decide on the general location of each body part, and from that, a skeleton is constructed.

Microsoft's technique for labeling body parts was to use a pre-trained decsision tree. To gather data for this decision tree, they used motion capture data. Using that motion capture data, they generated even more data and ultimately ended up with about 300,000 depth images on which to train their tree. The decision tree was trained using single pixel features; each feature made a decision about a single pixel, not a group of pixels. A feature was defined by two vectors, U and V. These vectors represent pixels nearby to the pixel in question. The actual *value* of the feature is equal to the depth at the pixel plus vector U minus the depth at the pixel plus vector V. This gives a single value based on nearby pixels.



An Example depth feature on both the finger and the palm.

The decision tree is trained using these features, which can be calculated quite quickly at runtime in order to label pixels in real time. Once all the pixels are labeled, a mean-shift filter is used to actually construct the skeleton.

To track hands, we attempted to emulate this process, except that we trained our tree on hands instead of whole bodies. The first major hurdle we encountered was collecting data.

We had neither motion capture nor computer generation available to us. What we ultimately came up with was to collect actual data with the Kinect of users wearing colored gloves. Theses gloves would provide us with a basis to automatically label each part of the hand for training.

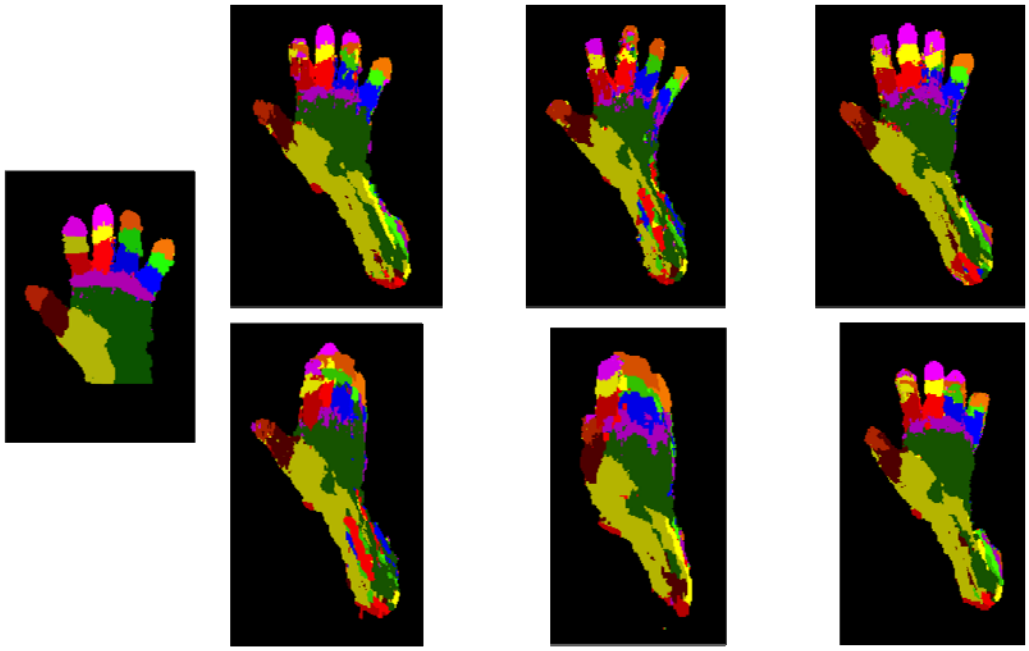
The actual process of designing colored gloves ultimately took months. Our first attempt was to use tie die to color white gloves. However, the tie die process took many hours and resulted in gloves with dull colors that ran together. Following these, we used paint markers to color the gloves, which resulted in much brighter colors that did not run together.

All 16 paint marker colors on a single glove

In order to reliably distinguish our colors, we could not use 18 different colors for our 18 labels, as this would make it very hard to distinguish between colors. We instead opted to use only 6 different colors, and use them on three different gloves. This way, by knowing what glove we were looking at and what color, we could easily determine what label we were looking at.

After capturing and cleaning data using these gloves, we used it to train a decision tree. Our initial decision tree had two problems. The first is that it was overfitting, which we solved by relaxing the constraints on it. The second is that our U and V vectors for features were much too small, so we allowed them to be larger. This resulted in a tree that could label with substantially more accuracy.

To go from this to a skeleton, we simply used a mean-shift filter, like Microsft (http://research.microsoft.com/pubs/145347/BodyPartRecognition.pdf). The filter works very well, and the only times it results in inaccuracy are when the labeling is inaccurate.



Final labeled hands

The results of our project are quite good. We have demonstrated that labeling a hand with Kinect is possible. We've also determined ways to make it even better, the primary of which would be to use substantially more data.