Feature Descriptors for Gait Analysis from Depth Sensors

Ben Crabbe

August 9, 2015

1 Introduction

Gait analysis plays an important part in the treatment and assessment of a number of medical conditions. Presently gait analysis is usually performed through a combination of visual assessment by an experienced physiotherapist, automated methods such as marker based motion capture, pressure sensitive walkways or accelerometers. It requires patients to travel to a gait assessment laboratory which is far from ideal for patients who have difficulty walking.

This problem, and a range of other healthcare challenges, is being tackled through research and development by the SPHERE (a Sensor Platform for Healthcare in a Residential Environment) group in Bristol. An automatic, in home, gait analysis pipeline has been designed [?] which assesses the quality of a subjects movement using inexpensive RGB-D cameras such as the Microsoft Kinect.

Currently this system uses pose information captured by the OpenNI skeleton tracking software, based on the algorithm of [?], to track and record the residents' body configuration using the skeleton tracking software provided for the Kinect. This skeleton tracking software infers the 3D coordinates of each of the body's relevant joints producing a $n_{joints} \times 3$ dimensional vector.

Training data of the This is then processed using a non-linear dimensionality reduction method, Diffusion maps [?], which extracts the non-redundant information and returns a 3 dimensional pose vector containing just the necessary information for determining gait quality. A statistical model of normal gait is built up from training data using this reduced pose representation. Then new data is compared with this model producing a gait quality score on a frame-by-frame basis.

Since this system uses machine learning methods to learn both the best reduced pose representation and the model of normal motion, it can be applied to other types of movement quality assessment such a sports movement optimisation or physiotherapy exercise coaching..

One issue currently limiting the effectiveness of this system is the fragility of the Kinect skeleton tracking software. This software was designed for controlling entertainment/gaming systems with the user viewed frontally, within a range of 1-4m and at a pitch angle of $\sim 0°$. Outside of these conditions skeletons become noisy and unreliable. Typically only a small fraction of data recorded from say a camera attached to the ceiling above the stairs is at all usable. Increasing amount of usable data requires more intrusive camera placement which is to be avoided. The skeleton trackers also perform extremely poorly when props are involved in the scene, for example grasping a banister or a ball often leads to erroneous joint positions for that arm. It also struggles to accurately record sitting/standing motions.

The aim of this project is to develop a tailor made system for determining the reduced pose vector directly from RGB-D footage. This requires a data driven approach since it will enable the flexibility of the rest of the system by being able to record a wide range of motions and should work with an effective accuracy with practical, unobtrusive, camera placements.

Accuracy will be measured by computing the mean squared error (MSE) of the produced pose vectors against the labelled ground truth on a testing subset of the SPHERE staircase 2014 dataset [**?**]. It will be examined as a function of pitch and yaw viewing angles and of range. We will also monitor overall system performance (how well the measured gait quality score matches the score labelled by a trained physiotherapist).

To achieve this we plan to use a convolutional neural network (CNN). CNN's are a supervised learning method for extracting features, such as the pose vector, from images. Given training images labelled with the expected output the network learns to map from the input images to the output by adjusting the free parameters in the network.

The network is arranged in layers with each layer applying some operation to the input. Convolution layers consist of 3D arrays of parameters, called filters, which are applied as a dot product with the pixels in windows of the image; this window is then shifted across the whole image producing a value, or response, each time. The response at each position is stored in a matrix, called a feature map. The feature maps of each filter in the layer are combined to produce the output of the layer which is a 3D array of values, essentially the same format as the input, i.e. an image, except that size of each dimension changes. Other types of layers include pooling layers, which reduce the size of the input, and fully connected layers that are the same as a convolution layer but the window extends across the whole image.

The process of designing the network consists of experimenting with different combinations of layers, and different layer settings e.g. filter size to achieve the best performance. For each test we will need to train the network to measure the performance.

Training a CNN for 2D RGB joint regression tasks has been achieved with as little as 4000 distinct training images [**?**]. The SPHERE staircase dataset contains 42 short sequences, some of which must be reserved for evaluating performance. We plan to make use of publicly available datasets that contain RGB-D footage annotated with Kinect skeletons to bolster our training data. We will also use data augmentation techniques such as taking the mirror images and random crops or adding random noise to each image. We will also use regularisation techniques such as dropout and weight decay which stop the network over training allowing us to make more use of the data we have.

The computational process of both computing the outputs of the network and the training updates require solving many large linear algebra equations, these can be readily parallelised and run on a GPU which increases the speed of training and running the network 30 fold. The implementation of the network is vastly simplified by using one of the open source frameworks for CNNs, we plan to use Caffe [**?**] since it is cited most frequently amongst the papers that have been reviewed. The process of designing the network consists of defining a set of layers in a simple protocol buffer definition (prototxt) file for example a typical convolution layer is

```
layers {
  name: ”conv1”
  type: CONVOLUTION
  bottom: ”data”
  top: ”conv1”
  blobs_lr: 1         # learning rate multiplier for the filters
  blobs_lr: 2         # learning rate multiplier for the biases
  weight_decay: 1     # weight decay multiplier for the filters
  weight_decay: 0     # weight decay multiplier for the biases
  convolution_param {
    num_output: 96    # learn 96 filters
    kernel_size: 11   # each filter is 11x11
    stride: 4         # step 4 pixels between each filter application
    weight_filler {
      type: ”gaussian” # initialize the filters from a Gaussian
      std: 0.01       # distribution with stdev 0.01 (default mean: 0)
    }
    bias_filler {
      type: ”constant” # initialize the biases to zero (0)
      value: 0
    }
  }
}
```

and similarly for other types of layers. This allow for a great deal of experimentation without requiring huge amounts of coding. Therefore although this project can be described as type — since that it involves building something new, using concepts and methods from deep learning, it will actually require far more evaluation and analysis of the performance of these networks than implementation.

To the best of my knowledge this project will be the first time that CNNs will be applied to a 3D human pose estimation task on RGB-D images (although this is not quite this since we find the pose representation rather than the full pose, but the network could be expected to perform this task reasonably well). It will also be a novel combination of CNNs and manifold learning methods since what we are essentially doing is simplifying the difficult task of human pose estimation through the dimensionality reduction stage. We believe that this will make the CNN easier to train and more effective overall since it has far less outputs to specify. If this is proved to be the case it could potentially be applied to other tasks that have been attempted with CNNs such as human action recognition.

The added value of this project, aside from the added value of the SPHERE work, is that this system has the potential to be applied to a wide variety of motions simply by retraining each stage. For an example, it could be used to coach basketball players to take free throws. First we would need to record the skeletons of players taking free throws then process them into a reduced pose representation for this motion. It would not be possible to use the Kinect in this scenario as it would fail to recover the skeleton with a ball in its hand. With our system it would be possible to record the motions using traditional marker based motion capture and process that to produce the pose vector. We would then use this data to train the CNN to extract this pose automatically. Then we would train the models of normal pose and dynamics perhaps using whether the shot was successful or not as indication of whether it is a good or bad pose. This system could then be used on players to assess what specific part of their motions differs with the model of 'successful' motion. This example illustrates how our adaption increases the flexibility of the system over its current state.

2 Kinect Sensor

3 Preprocessing

All data used in this project SPHERE-staircase2014 dataset [**?**])