The University of Calgary
Department of Electrical and Computer Engineering

ENCM 509
Biometric Technologies and Systems Design

Lab # 7

Gait Biometrics: detecting gait abnormalities

1 Introduction

The purpose of this laboratory exercise is to become familiar with gait data acquisition using RGB-Depth camera, extracting a gait cycle and the derived features, as well as detecting gait abnormalities (such as normal versus limp).

Gait analysis aims at various sport or healthcare applications (such as analyzing an athlete gait to improve performance, prevent injuries, or detect abnormalities in a patient's gait after an injury). It is also known to be used to verify the identity of the individual by the way they walk. This experiment concerns only with detecting a particular gait abnormality such as limp.

To study gait, a multiple video camera system are being used, such as 4–8 Vicon camera system, or other high frame-per-second rate cameras. Another way is to use RGB-Depth camera such as MSoft Kinect.The Kinect sensor includes a color camera and a depth sensor, consisting of a time-of-flight projector/receiver. It provides full body 3D motion capture, and MSoft SDK converts it into 3D coordinates of the main joints that form a "skeleton" of a moving human [1].

2 Methodology

In this exercise, we will use the information about the joint of the lower body, specifically, those of the hip, knees, ankles, and feet. The Kinect camera in a laboratory setting performs recording of a subject, starting approximately 4 meters away from the camera. During the recording, a software built into the camera collects the depth data from the subject and computes/tracks their joint positions. These joint positions are stored as Cartesian coordinates x, y, z respective to the position of the camera. From the coordinates of the subjects joints, the gait parameters such as cadence, stride length, and joint angles can be computed in real time.

2.1 Data Collection Procedure

The gait recording will be performed in the Biometric Technologies Laboratory on a "walkway" with two Kinect cameras recording front and side view. To identify a single gait cycle, a "walkway" in the lab is marked with equally spaced lines. An individuals is asked to start walking from the 4-meter mark, using their right foot first. During recording, the gait cycle end time is registered (by the recorder, or user), so that the recorded data contains one gait cycle for that individual.

The recordings are being formatted as the three `.csv` files, the first one contains the Geometric positions, the second one has the Joint angles, and the third one has distances between the ankles. In this exercise, you will be asked to detect and plot one gait cycle using the data from the `.csv` files.

2.2 Gait Cycle

A gait cycle starts with heel strike of one foot and ends with heel strike of that same foot. There are 8 gait cycle phases; in this lab, we will focus on the main three phases of the gait cycle: Heel Strike (HR), Foot Flat (FF), and Toe Off (TO) as shown in Figure 1.

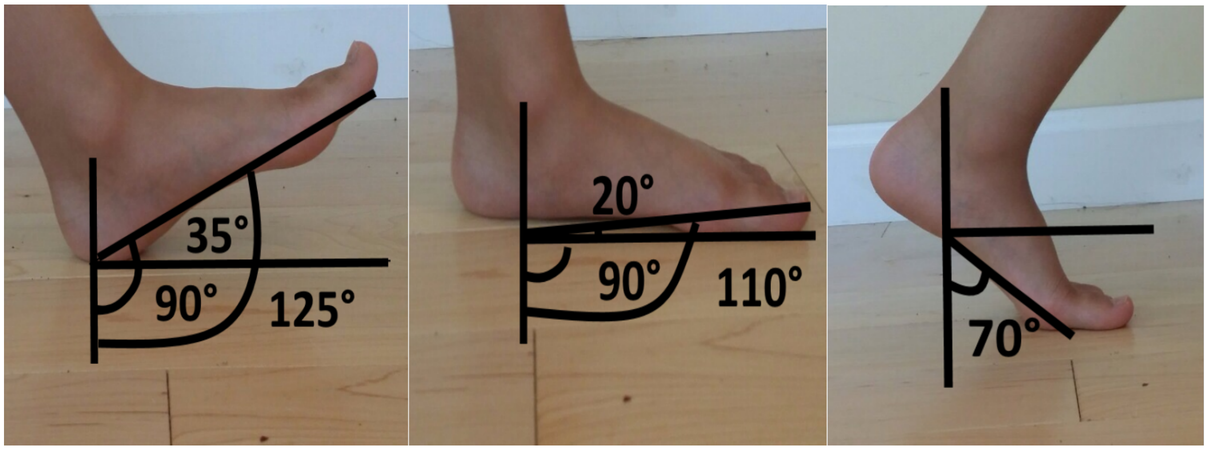


Figure 1: Three Gait Cycle phases using foot angles

There are two approaches we will use here to detect a single gait cycle [2]:

- Joint Relative Angle (JRA).
- Joint Relative Distances (JRD).

You will have up to six recording, so you can repeat the gait cycle detection procedure six time, using both methods each time.

Next, you will calculate Cadence (number of strides per minute) of the gait.

Finally, the last step is to classify the gait to "Normal" and "Abnormal". You will have recording of an individual's normal gate, as well as the same individual will be asked to do two recordings of "Right Limp" and "Left Limp", which shall be classified as "Abnormal". Below, we consider the used classification (Pattern Recognition) techniques.

2.3 Classification of gait by type (normal and abnormal)

In order to distinguish a normal gait from abnormal (such as limp in the left or right leg), we have to apply a classifier that would distinguish these two classes.

Three classifiers will be used in this lab: Linear Discriminant Analysis (LDA), Support Vector Machine (SVM),and K-nearest-neighbors (KNN). They are briefly described below.

- The LDA classification technique [3] is used to classify observations into two or more groups when trained on a sample of known observations that consist of N features. Assuming that data is Gaussian,the mean μ_k and variance σ^2 for each input x for each class k can be computed.The model uses Bayes Theorem to estimate the probability of the output class k given the input x. The discriminate function $D(x)$ for class k given x, μ_k, and prior probability p_k can be written as:

$$D_k(x) = x * \frac{\mu_k}{\sigma^2} - \frac{\mu_k}{2 * \sigma^2} + \ln(p_k) \tag{1}$$

- The purpose of a SVM classifier [4] is to find a hyperplane that divides a data set of n points $(x_1, y_1), ..., (x_n, y_n)$ into two classes. When choosing a hyperplane, the objective is to maximize the margin $\frac{1}{||w||}$ of the training data. For this particular application the data is assumed to be linearly separable, therefore the maximization of the margin $\frac{1}{||w||}$ was computed as follows:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \frac{1}{2}||w||^2 \\ \text{subject to} \quad & y_i(w \cdot x_i + b) \geq 1,\ i = 1, \ldots, m. \end{aligned} \tag{2}$$

- The KNN classification technique [5] utilizes the euclidean distances between training samples to assist with classifying new data. This method is widely used in applications such as image processing, computer vision, and bio-informatics, therefore it made sense to investigate it for this application. To classify a data set of i points x_i, we examined the class most common amongst its k nearest neighbors measured by the distance function $D(x_i, y_i)$.

$$D(x_i, y_i) = \sqrt{\sum_{n=1}^{k} (x_i - y_i)^2} \tag{3}$$

In this Lab, we will use the classifier implementation available in Matlab.

Exercise 1: Detect one Gait Cycle using JRA

In this exercise, you will be using the Joint Relative Angle (JRA) to detect the gait cycle as described below. Note that you will have up to six recordings, therefore, will be detecting the gait cycle for each, thus having a sample of six gait cycles.

1. In Matlab, open the file `GaitCycleJRA.m`.
2. In any text editor, or Excel, open the file `Smoothed_JointAngles.csv`.
3. Copy both columns (Left Knee Angle and Right Knee Angle) into your Workspace, choose Numeric Matrix and change the name to `KneeAngles` and then click import. You will get a matrix of a size $N \times 2$ (for example, 100×2).
4. Repeat the same procedure for the columns `Left Ankle Angle` and `Right Ankle Angle`, and name it `AnkleAngle`and select the first column and you will get a matrix of a size $N \times 1$ (for example, 100×1).
5. Run the code `GaitCycleJRA.m`; this shall produce a graph similar to Figure 2.
6. Record the Minimum Knee Angles using the variables `MinangledegL` and `MinangledegR` generated by the code; the variables are available in the Workspace. You will use these values later in Exercise 4.
7. Record the value of the variables `"FrameStart"`and `FrameEnd` generated by the code; the variables are available in the Workspace. You will use these values later in Exercise 3.

Exercise 2: Detect one Gait Cycle using JRD

In this exercise, you will be using the Joint Relative Distances (JRD) to detect the gait cycle as described below. Note that you will have up to six recordings, therefore, will be detecting the gait cycle for each, thus having a sample of six gait cycles.

1. In Matlab, open file `GaitCycleJRD.m`.
2. Copy your .csv files that contains your JRD values and JRA values for the Knees. They should have a names such as `Subject_N_Seq_M_GaitCycle_N.csv`, and `Subject_N_Seq_M_JointAngles_N.csv`.
3. Paste the copied files into the same folder that contains `GaitCycleJRD.m`.
4. If done correctly, the `.csv` files should appear on the right side of the Matlab interface as shown in Figure 3.

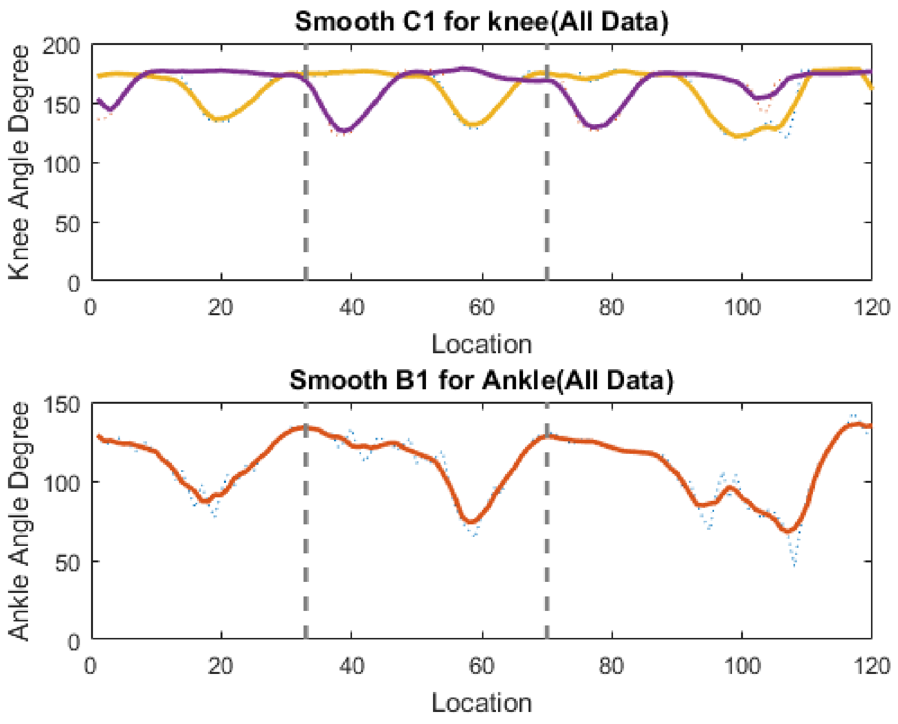
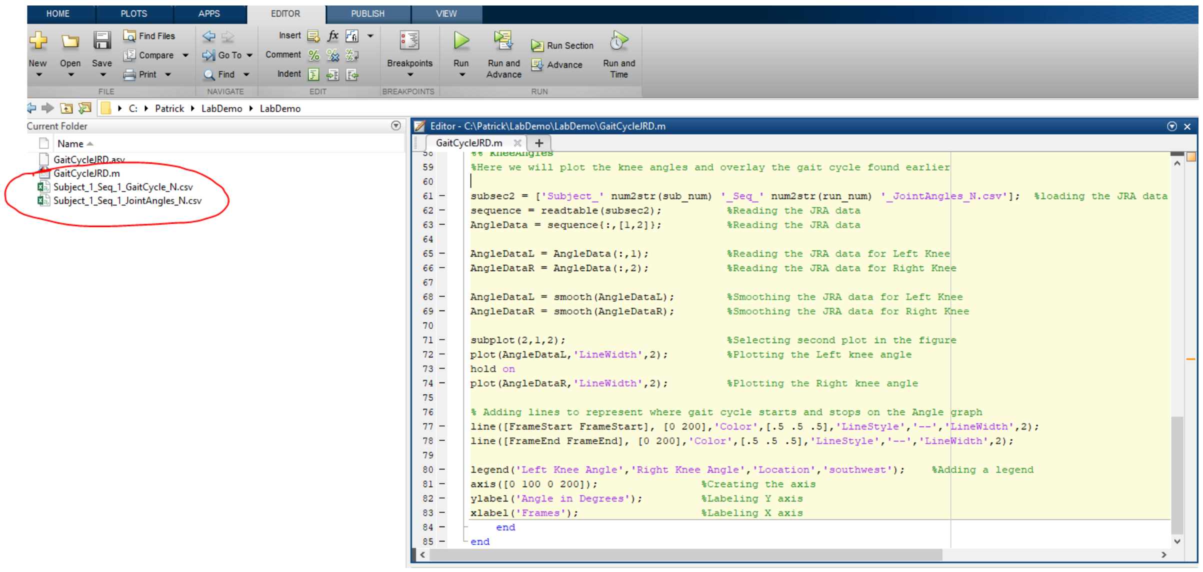


Figure 2: Gait cycle graph built using the JRA method



```
%% KneeAngles
%Here we will plot the knee angles and overlay the gait cycle found earlier

subsec2 = ['Subject_' num2str(sub_num) '_Seq_' num2str(run_num) '_JointAngles_N.csv'];  %loading the JRA data
sequence = readtable(subsec2);           %Reading the JRA data
AngleData = sequence{:,[1,2]};           %Reading the JRA data

AngleDataL = AngleData(:,1);             %Reading the JRA data for Left Knee
AngleDataR = AngleData(:,2);             %Reading the JRA data for Right Knee

AngleDataL = smooth(AngleDataL);         %Smoothing the JRA data for Left Knee
AngleDataR = smooth(AngleDataR);         %Smoothing the JRA data for Right Knee

subplot(2,1,2);                          %Selecting second plot in the figure
plot(AngleDataL,'LineWidth',2);          %Plotting the Left knee angle
hold on
plot(AngleDataR,'LineWidth',2);          %Plotting the Right knee angle

% Adding lines to represent where gait cycle starts and stops on the Angle graph
line([FrameStart FrameStart], [0 200],'Color',[.5 .5 .5],'LineStyle','--','LineWidth',2);
line([FrameEnd FrameEnd], [0 200],'Color',[.5 .5 .5],'LineStyle','--','LineWidth',2);

legend('Left Knee Angle','Right Knee Angle','Location','southwest');    %Adding a legend
axis([0 100 0 200]);                 %Creating the axis
ylabel('Angle in Degrees');          %Labeling Y axis
xlabel('Frames');                    %Labeling X axis
    end
end
```

Figure 3: Loading the GaitcycleJRD and JointAngles files

5. Run the code `GaitCycleJRD.m`; it should produce a graph similar to Figure 4.
6. Record the Minimum Knee Angles using the variables `MinangledegL` and `MinangledegR` generated by the code; the variables are available in the Workspace. You will use these values later in Exercise 4.
7. Record the value of the variables `"FrameStart"`and `FrameEnd` generated by the code; the variables are available in the Workspace. You will use these values later in Exercise 3.

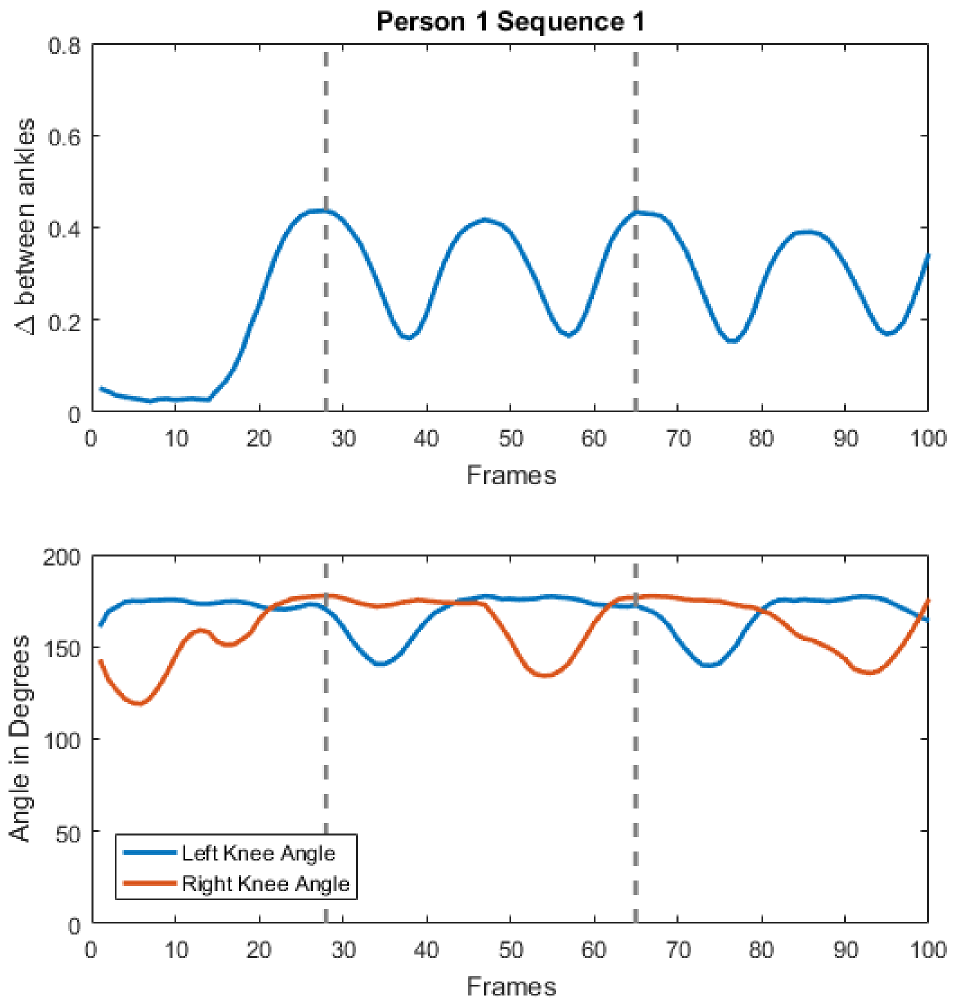


Figure 4: Gait cycle graph built using the JRD method

Exercise 3: Calculating Cadence

Many gait parameters can be extracted using the positions of the joints such as cadence, stride length, and others. In this exercise, you will derive the cadence of one gait cycle. Cadence is the rate at which a human walks, expressed in steps per minute. The average cadence is 100 – 115 steps/min. To calculate the cadence (for the JRA and JRD methods) follow the steps below:

1. Use the `FrameStart` and `FrameEnd` values in the Matlab Workspace that you have recorded them before.This will give you the number of frames (# of frames) that are used to get one gait cycle.

2. Use the equation below to write a simple script in Matlab that would calculate the cadence:

$$Cadence(steps/min) = \frac{2\ steps}{\#of frames} \times \frac{30\ frames}{sec} \times \frac{60\ sec}{min} \tag{4}$$

Here, 30 $frames/sec$ is the frame-per-second rate of the Kinect camera. perform your calculation fr each recorded gait cycle and each gait type (normal, abnormal). You shall have 12 values for cadence (for each gait cycle, and for both normal and abnormal). Analyze the data and estimate the mean and deviation of your sample. Report your findings.

Exercise 4: Classification of gait

In this lab, we will need a data for all six or more records of each gait type (normal, right limp, left limp etc.) for classification in the form of a `.mat` file. You are provided with a sample file `Combined-TData.mat` file located on N: drive.

1. Open Matlab and load the sample file `Combined-TData.mat` (in a later stage, you will use your own file with the recording of the gait) into the Workspace.

2. Go to APPS and clink on `Classification Learner`. This shall created a labelled set of data for further classification.

3. Click on `New Session` at the upper left hand corner and choose `From Workspace`. A new window will open with the name `New Session` with three steps in the sub-menu. In Step 1, choose your file name. In Step 2, in the options (below the step line) make two change: change `Person` and `Sequence` from `Predictor` to `Do not import`. In Step 3, for `Cross-validation`, choose 5. Cross-validation defines a random partition on a set of data of a specified size. This partition is used to define the size of the test and the training sets for validating the used statistical model.

4. Perform the classification using the three different classifiers below:

   - LDA classifier

     Open `Start Session`. In the opened window, click on the drop-down list beside the `Advanced` button and choose `Linear Discriminant`; this will perform an LDA classification. Click the `Train` button. The training results will be presented in the form of the "Scatter Plot" and the "Accuracy". Record and analyze the results.

   - SVM classifier

     Repeat the same for the SVM classifier. In the start menu, in `Advanced`, choose `SUPPORT VECTOR MACHINES` and then choose `Linear SVM`. Perform training, and draw the plots. Analyze the results, and compare with the LDA.

   - KNN classifier

     Repeat the same for the KNN classifier. In the start menu, in `Advanced`, choose `Fine KNN`. perform training and graph plotting. Analyze and compare with the results obtained when using the LDA and SVM.

5. Classification of your data

   In this part, repeat the above classification with you own data. In order to process your own data, you will have to create your own `.mat` file as described below.

   To create your own `.mat` file, copy the sample file `Combined-TData.mat` into Matlab workspace, rename the file (your choice), double click on it and change the old information to the new one. To do so, use the values that you recorded for minimum knee angles, - these are the values `Min_R_Knee_Angle` and `Min_L_Knee_Angle`. For example, for Person labelled 1, you will have 1,2,3,4,5,6,1,2,3,4,5,6 in the `Sequence` field, where the first six are for the JRA method and the other six are for the JRD method. Next, place your values for the Minimum Knee Angles in the section `Min_R_Knee_Angle` and `Min_L_Knee_Angle`. Put your dta in the `Status` field as well. Your new `.mat`' file shall have 12 rows.

6. Perform classification using LDA classifier, as well as SVM and KNN. Compare the results.

Laboratory Report

In this report, include:

- Report on Exercises 1–5, with illustrations where appropriate, (15% each, 75% total).

- Script of the code cadence detection and any new or modified code for the classification procedure (25%).

Acknowledgments

This laboratory exercise was co-developed by MSc students P. Kozlow and A. Noor in the Biometric Technologies Laboratory, ECE Dept., UofC, 2017-2018, using Mictosoft Kinect SDK for Windows and Matlab.

Svetlana Yanushkevich
February 8, 2018

References

[1] Kastaniotis, D., Theodorakopoulos, I., Theoharatos, C., Economou, G., and Fotopoulos, S., “A framework for gait-based recognition using Kinect,” *Pattern Recognition Letters*, Available online 9 July 2015, ISSN 0167-8655, http://dx.doi.org/10.1016/j.patrec.2015.06.020.

[2] Ahmed, F., Paul, P. P., and Gavrilova, M. L., “Joint-Triplet Motion Image and Local Binary Pattern for 3D Action Recognition Using Kinect,” *Proceedings of the 29th International Conference on Computer Animation and Social Agents*, 2016.

[3] McLachlan, G., “Discriminant Analysis and Statistical Pattern Recognition,” John Wiley & Sons, 2005.

[4] Kecman, V., “Learning and soft computing,” MIT Press, 2001, pp. 148-166.

[5] Altman, N. S., “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician 46.3,* 1992, pp. 175-185.