Real-Time Markerless Gait Analysis System

Corbin Combs, Stephanie Chan, Jonathan Korty, Sedinam Gadzekpo

**Abstract**— This paper proposes a system for analyzing gait for walking and running gait cycles. By using background subtraction methods to segment out the human form, the system can use a distance transform algorithm for model fitting a skeleton onto a human figure, allowing for tracking the range of motion of the joints. The change in angle between the joints are used to classify the gait of the figure in question as walking or running in real-time.

**Index Terms**— *gait, tracking, biometrics, computer vision*, model fitting

—————————— ◆ ——————————

# 1 Introduction

Tracking and identifying something about a person – or even the person themselves - by analyzing their gait is a modality of biometrics. An individual’s gait is an unobtrusive piece of data that is relatively easy to acquire. The fact that each person’s gait can be measured as a unique quality makes this study very applicable to scenarios involving security.

A subject walking can be observed with the subject having little or only assumed knowledge of judgement. Obviously, this preserves the subject’s natural behavior and environment. Their continued natural behavior leads to improved accuracy of analysis, as the actions observed are more likely to be without the intention of altering the data by way of altering behavior. This discrete gathering of data becomes even more useful when considerations of specific properties of a gait are recognized and identified in situations where other biometric systems may prove to be inefficient with regard to the cost of time and energy required for enrollment as well as cooperation of those being analyzed.

## 1.1 Scenario

In building our system, we aimed to use an observed gait to distinguish whether their means of travel was by walking or by running. The only sensor required to capture this data is a simple camera such as the webcam on a laptop. Various methods have been implemented in the past to analyze properties of gait. Segmenting the motion from a frame is always a key point in these systems, as the methods to capture the necessary data while restricting the noise and other information that is in motion along side the person walking.

Our markerless gait analysis system could be used for surveillance in a store to detect shoplifters. In most shoplifting scenarios the shoplifter will be in a hurry to leave the premise in order to get away from the scene. Our system would identify a person fleeing from the scene and alert the correct authorities.

In a similar way, our system could be used in a hallway or alleyway to detect problems or potential danger in the area. If someone is running through the hallway or alleyway, there is a high likelihood that something is wrong and authorities could be alerted and sent to investigate the area further. This would help notify first responders faster.

This could also be used in rehabilitation centers to analyze patients healing progress. This would allow the doctors to observe the changes in the patients gait throughout their rehabilitation. This could give professionals a better insight to a specific person’s gait.

Our markerless gait analysis system may be used to analyze a single person’s gait in order to tell whether the subject is walking or running. There must be a somewhat high contrast between the person and the background so that it may correctly identify the contour of the subject. The whole contour of the subject must be within the frame. The camera must be relatively still in order to capture a moving person. The subject must also be in continuous motion.

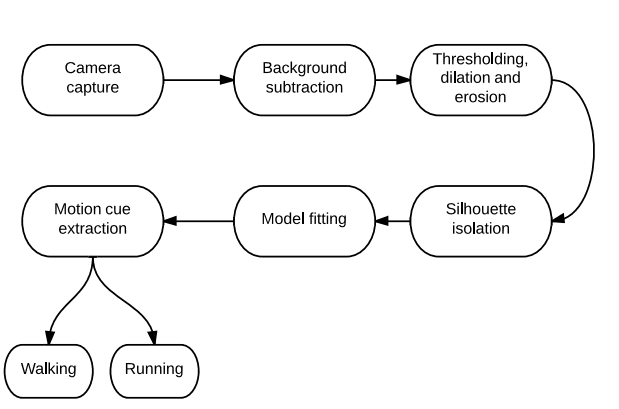
## 1.2 Existing Literature

In the course of our research, we found numerous papers describing various gait tracking and analysis methods. Boyd & Little [7] provide a high level overview of the theory, problems, solutions, and applications of gait as a biometric. Kim & Palik [8] describe a gait recognition algorithm using active shape models. Zhang, Vogler, & Metaxas [9] propose a method of model based gait recognition through the extraction of gait using the Metropolis-Hasting method and Hidden Markov Models to calculate trajectories. Urtasun & Fua[10] describe a method of tracking using 3d temporal models.

For our particular sytem, we found the research of Vignola, Lalonde, & Bergevin [2] as well as Khan, Westin & Dougherty[1] for the model based skeleton fitting system that we had implemented. Our system combines the most pertinent parts of both methods and simplies the skeleton fitting given the constraints we had set for our system.

# 2 Method

A high level overview of the system is shown in Figure 1. Our process begins by capturing movement in a series of frames from a webcam. Then, segmentation is performed by creating a background model using an Adaptive Gaussian Mixture Model [3]. Contours are established, creating a region of interest. Within this region, the distance transform is used as a basis for model fitting and the subsequent joint movements of this model are used to classify the gait cycle of a subject.



*Figure 1: A high level system overview*

## 2.1 Theoretical background

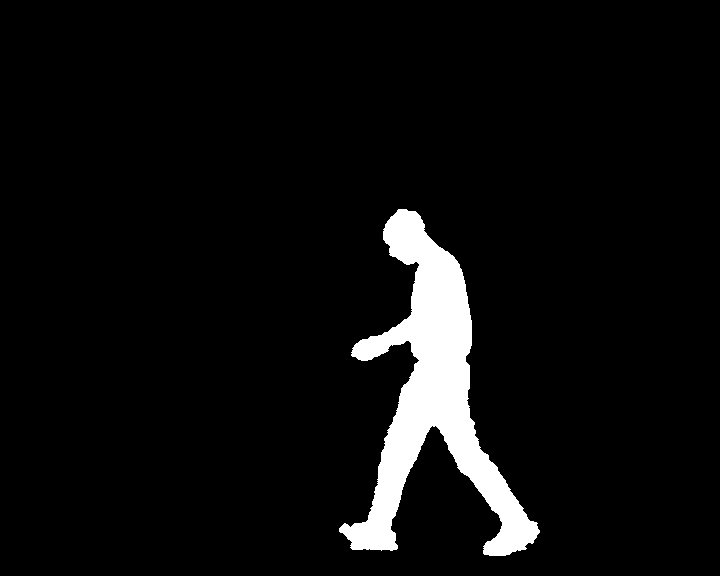
## 2.1.1 Background Subtraction

The segmentation of the motion from the static portion a frame involves the application of a statistical model to a binary image, called Mixture of Gaussians which is described by [3]. This segmentation creates a background model composed of k-groups of colors in a corresponding Gaussian distribution for each color.

The amount of groups are limited to around 3 to 5. Depending on the amount of time a given color has remained in place, or static, the probability of that color belonging to the background increases. That group of color is segmented into its own group. If this segmented group does not change in color or location, meaning the cluster has not grown or shrunk, then that group clusters tighter.

Segments that change over time clusters in a loose fitting fashion. If this happens, then the variation on the cluster size results in a lowered probability of belonging to the background. As time moves on, the baseline of the intended background properties will most likely change. This can be accounted for by implementing an adaptation of the newer properties by checking new, incoming pixel values against the existing segments. If a match is unable to be identified, a new-segmented group is created.

Once the groups are established for the pixels in the frame, the pixels are compared to a threshold, which is the minimum probability that the background is in the frame based on the results from previous frames. Then the smallest value that is bigger than the threshold represents the background model. Any pixel that is more than 2.5 standard deviations away from any of the k distributions the background model is labeled as a pixel belonging to the foreground. In order to control the amount of change to be considered, a constant focused on the amount of time included in the background model is included as a parameter.



*Figure 2: An example of background subtraction* http://users.ecs.soton.ac.uk/drn/ug/compvision/gait/BackgroundSubtraction/index.htmlt

## 2.1.2 Silhouette Isolation

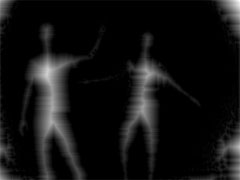
Contours are established on the resulting binary image of segmented motion by analysis of each pixel in the frame and that pixel’s neighboring pixels. The process described by Suzuki and others [Suzuki], outlines an algorithm that determines the relationships between a region – or neighborhood of pixels, and continues to group these overall relationships to a minimized cluster of points, or contour.

Pixel p(x,y) has 8 neighboring pixels. The preceding column and the following column of p(x,y), where there are some pixels, p1(x-1,y) and p2(x+1,y) that account for two of p(x,y)’s neighbors. Then there are points in the preceding row and following row of p, p1, and p2 where there are some pixels, p3(x-1,y-1) and p4(x-1,y+1) corresponding to p1, p5(x,y-1) and p6(x,y+1) corresponding to p, p7(x+1,y-1) and p8(x+1,y+1) corresponding to p2. These neighboring pixels’ binary values are compared to the value stored in the pixel p(x,y). The relationships of each individual set of neighboring pixel are stored as binary values, the values being based on the various criteria established in Suzuki’s method [Suzuki]. Then the updated binary values per pixel in the neighborhood are used to establish clusters or groups. These are then minimized to provide the smallest amount of clusters possible as contours.

The results of the minimized contours are used to provide dimensions of the area that has been detected to contain motion. By creating a container that isolates the area determined to be motion, the area’s dimensions can be used as a baseline for the determination of a local area of interest that optimizes the functionality of the model fitting process applied to this segmented area.

## 2.1.3 Model Fitting

Once a silhouette of a human figure has been properly segmented using the background subtraction and silhouette isolation methods described above, a skeleton model can be fit to the region of interest using a combination of applying a distance transform on the binary image and using the generated distances to fit the joints appropriately.



*Figure 3: Distance transform of human body* *http://www.mat.ucsb.edu/whsmith/xray/xray.jit.distance.html*

The distance transform finds the distance of the nearest zero pixel for every pixel in the binary image produced from the background subtraction. This results in a grayscale image that represents the distance of all the pixels within the body region to the contours of the body. The effect is a rough skeletonization of the body, where the areas of maximum distance from the contours are the regions where an anatomical skeleton would fit. Vignola, Lalonde, & Bergevin proposed a similar method for as a basis for their proposed skeleton fitting method.

Borgerfors [5] describes an efficient algorithm for performing such a transformation. In principle, a distance transform is a global operation, which can be costly to compute. Using the methods in [5], the local distance calculations eventually propagate to show the global distance of the image. For every pixel, a 3x3 mask is used to calculate the nearest zero pixel value horizontally, vertically, and diagonally using the Euclidean distance. These distance calculations are summed to get the overall distance of the pixel.

The distance transform provides a basis for fitting the skeletal joints to the region of interest. Our method was inspired by the method proposed by [2]. Their method requires generating all possible points available given the possible range of motion of a particular joint and then finding the best fit on the highest intensities of the distance transform. However, because our purposes requires speed and is only intended to detect a limited range of motion, we simplified the system to better serve the system.

We will consider only the joints most pertinent for the purpose of distinguishing walking and running in a person’s gait as the following points: The torso, the neck, the two knee joints and the two feet. The positioning of these points of interest can be bounded by constraints posed by the human anatomy.

In [1], Khan, Westin & Dougherty describe the proportions of the human body, which can be used to bound the region of interest for each joint. In particular, we can assume that the neck is at the top 13% of the bounding box, the knee joints are at the bottom 29%, and the feet are set at roughly the bottom 5%. These assumptions allow us to constrain the y-axis of the joints to be fitted.

For both the torso and the neck joint, we can search within the bounding box for the appropriate x-axis to fit to, which is represented by the highest intensity (and in turn, the maximum distance) in the distance transform on that particular y value. We fit these joints using the found x and y values. For the feet and knee joints, we do a similar search but split the bounding box in half since it is assumed that at any given time during the continuous motion of a subject, one knee and foot pair would be in the opposing half of the bounding box.



*Figure 4: Example of contours and joint fitting*

## 2.1.4 Motion Cue Analysis

The fitted joints allows for the capability to track the movement of the joints between frames. The easiest way to measure this is by using the angles formed by the joints similar to the methods used by [2]. In our implementation, we computed the angle created between the feet and the torso for every frame. We stored the last 4 angles in a queue and computed the average change of the angles. This gives a sense of how rapidly a person’s gait changes, signifying a difference in walking and running. Above a certain threshold in the angle change, the person is detected to be running and anything below that signifies a person walking.

## 2.2 Implementation Details

## 2.2.1 System Requirements

Our implementation required the use of a webcam and either Mac OS X or Ubuntu 12.10.

## 2.2.2 Programming Language

Matlab was the first choice in choosing a programming language, but when acquiring video from a webcam, Matlab became very slow only allowing for one frame every one or two seconds. It was then determined that Matlab would not be an optimal solution. It was then decided C++ with OpenCV should be used. After we switched to C++ with OpenCV, we were able to overcome this slow frame rate. Our final decision was to use C++ with OpenCV.

## 2.2.3 Technical Challenges

Achieving real time video acquisition and performing operations on the images produced was very challenging. This was the first project any of us had the requirement of running in real time. After switching to C++ with OpenCV, the real time requirement became much less of a concern.

Finding a background subtraction that could handle subtle camera or background movement and could filter out shadows was the most difficult challenge of this project. We tried many different methods of background subtraction including subtracting the first frame from the current frame, least squares fitting, optical flow, and two different methods of mixture of gaussian background subtractions. After finding out these methods were not as useful as we had hoped, we then began trying different human detection methods. We tried using Haar cascades to detect the human face in order to determine whether or not a blob was a person. This worked well in determining what was a human and what was not but was very inefficient.

Convexity defects were found in order to identify whether a blob was a person or not. These were not as effective as we had hoped because they did not adjust well to the noise of our scene. Histogram of oriented gradients were also used to detect whether a blob was a human or just noise but did not turn out to be very effective and classified a lot of noise as a human. We then decided to go back to background subtraction and used a mixture of gaussians background subtraction.

This was picked because it incorporated shadow detection and would allow for slight camera and background movements.

Another challenge we had to overcome was cross compatibility. This made for code only working on certain machines and code not being able to be ported between all the group members’ machines. This complicated collaboration and took up time figuring out why the code would execute on one machine but not another. In order to correct this we all used UNIX machines and worked with each other to debug one another’s programs.

We also were challenged in that some of the group members did not have a large enough space to set up the camera to capture their whole body. This was challenging because you could not test the system. This meant work could not be done at home and must be done at a different location.

Skeleton fitting was also a challenge. We tried to use methods such as SIFT and SURF to detect and keep track of the feet but this turned out to be very inefficient and not as helpful as we had hoped. In order to solve the skeleton fitting problem we formed boxes around points of interest based on the position of the bounding box around the blob. We then used a distance transform to find the distance of each point in the contour to the edge of the contour.

Another challenge that arose is when we detected walking and running, the skeleton would jump around causing the angle to be inaccurate. This was caused by the noise in the background subtraction and would also happen when someone walked out of the frame. In order to account for this we averaged the change in the angle over multiple frames so that if the angle change in one frame was large it would not skew the data as much. This allowed us to see the trend of what the gait of the subject had. This allowed us to then find a threshold at which the subject was running or walking.

## 2.2.4 Task Distribution

Stephanie began the project by working on background subtraction on a single image in Matlab. She then worked on implementing background subtraction in OpenCV on the video. She calibrated some of the background subtraction methods and experimented with the bgslibrary using the different background subtractions in the library. Once the background subtractions were tested and calibrated, she worked on the template fitting and came up with using the distance transform as the method of finding out how to fit the skeleton on the image. She also helped debug and write the report.

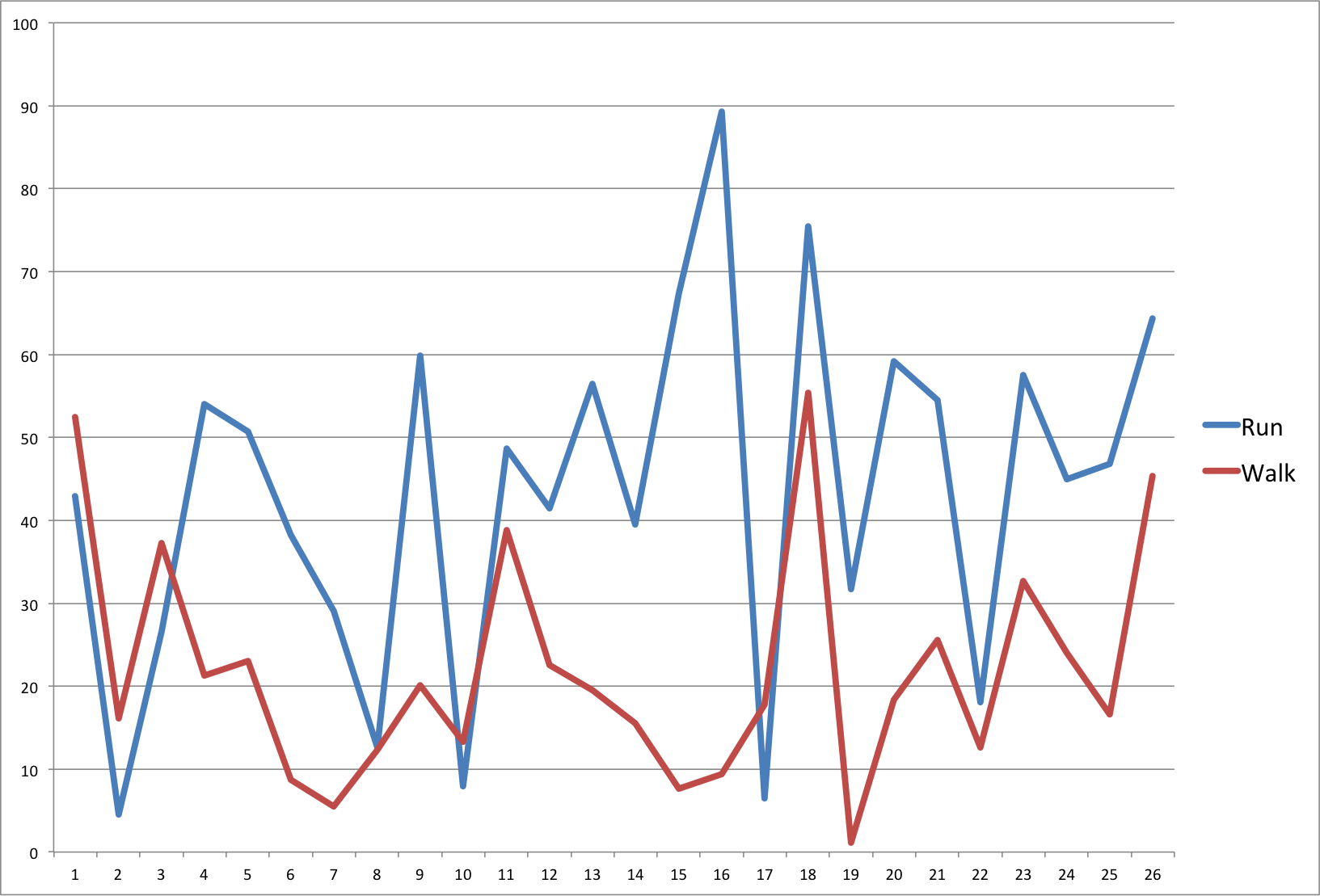
Corbin began working on the project by working on implementing background subtraction of video from a webcam in Matlab. He found this very inefficient and ran at about one frame every one or two seconds. He then began installing C++ with OpenCV. This solved the inefficiency problems and ran very smoothly for background subtraction and most other methods. He also worked on finding a human detection algorithm. He found out about histogram of oriented gradients. He also implemented the gait analysis of whether a person was walking or running. He helped debug and write the report.

Jonathan researched and tested many of the background subtractions. He worked on implementing two types of optical flow and tested it to see if it worked for our application. He then proposed that background subtraction was a better method for our project. He tried to test SIFT and SURF to see if this could help us keep track of the feet or other body parts we needed to track and detect. He found that this was too CPU intensive and slowed the frame rate down. Jonathan helped debug and solve problems with the code. He also helped write the report.

Sedinam researched Hough transforms and found out that that did not work very well for our application. She helped write the introduction of the paper and tried to help with the rest. She also made system overview graph of how our system worked.

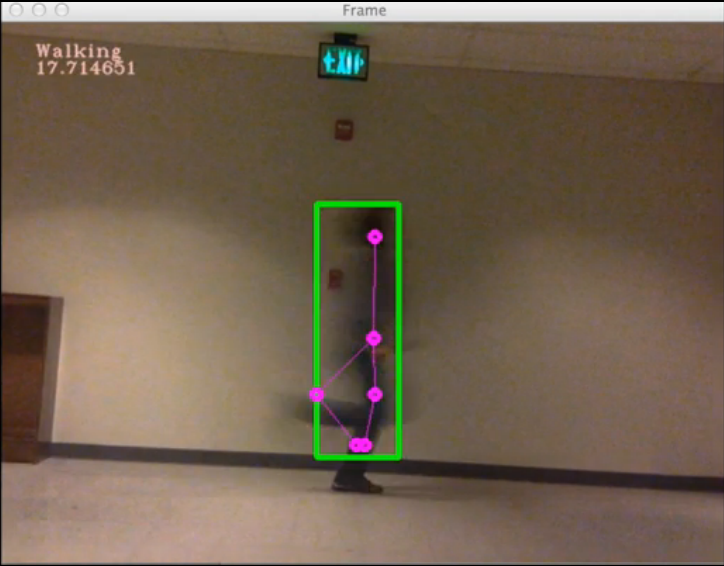
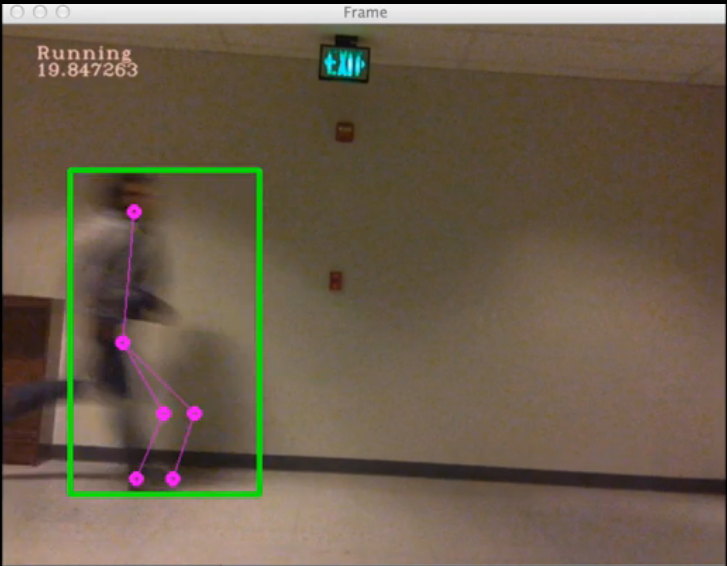
# 3 Results

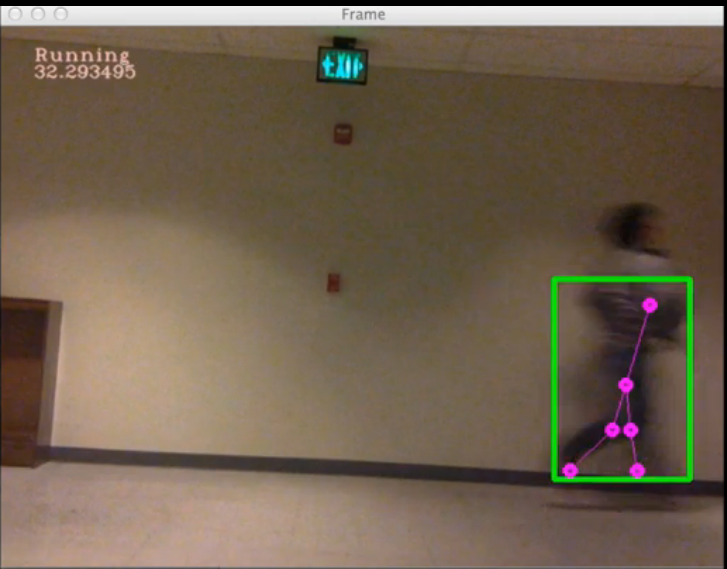
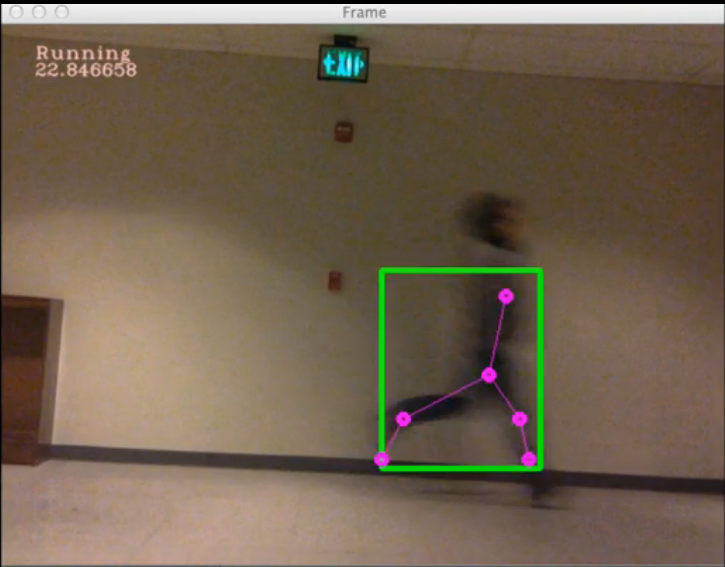
Our markerless gait analysis system was able to detect a person running and walking fairly accurately. The accuracy was challenged when the subject moved off the frame, had part of their body obstructed by another object or their body came in contact with a conflicting background. These were caused by the flaws in the background subtraction used in this method. In Graph 1, you can see that although the change in the angles were averaged from a group of frames, sometimes the data was not clear whether the subject was walking or running.



*Graph 1: Running vs. Walking*

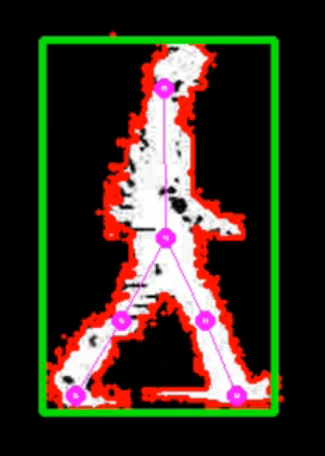
In Sequence 1, you can see that our skeleton would sometimes jump around skewing the data and not give an accurate result. This could help our system become more robust and have less false readings.





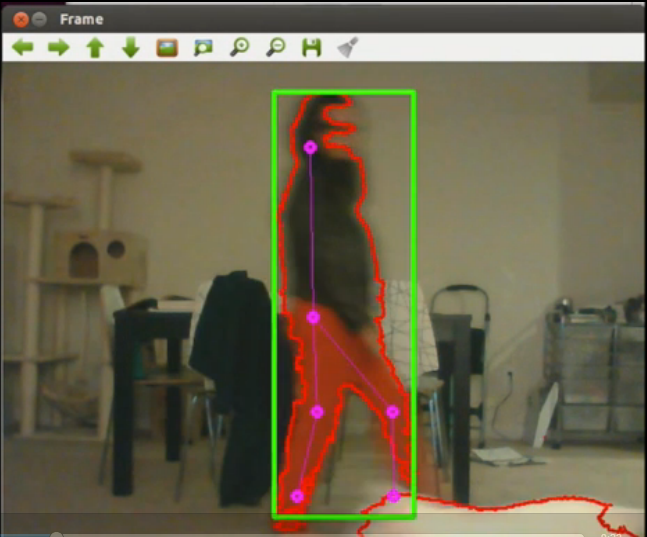
*Sequence 1: Skeleton displacement*

In Image 1, you can see that if we had a better background subtraction the skeleton would fit very well to the contour, thus giving us a better measurement for the change of the angle. With better measurements the more accurate we could determine whether the person was walking or running. This would make our system more effective and useful in its applications.



*Image 1: Better background subtraction*

In Image 2, it can be seen that even with background noise and background motion, the system ignored these factors and still fit to the human contour. The system was fairly effective in keeping only the human contour and not switching to a contour that was not a person.



*Image 2: System with other motion*

Our implementation of our system was able to run at 10 frames per second on a standard laptop. This efficiency was due to the effectiveness of C++ with OpenCV. While with Matlab we could not even get this implementation to run.

# 4 Discussion and conclusions

In this paper, we proposed a method for the markerless gait analysis of walking and running subjects in real-time. By segmenting human regions of interest through background subtraction then performing a distance transform, we were able to fit a model onto a body walking through the frame. The joints on the fitted model can be used to track the movement of a subject over a series of frames and the change in angle of the leg movements indicates if a person is walking or running.

Our system excels in tracking the gait of humans walking continuously through the camera frame against a relatively static and reasonably contrasted background. In our tests, it could continue fitting the model in both running and walking situations, and could adequately classify the type of gait very efficiently with very little lag.

However, our system lacks flexibility for non-moving subjects in the frame due to the nature of the background subtraction used. When a person stands still in the frame, ghosting occurs where certain parts of the person in question is classified as part of the background. The longer a person continues to be in the frame, the noisier the background gets. Low contrast backgrounds also pose a challenge to the system where humans cannot adequately be distinguished from the background. Occlusion also poses a challenge in multi-person scenarios.

In addition, our model fitting implementation is limited in scope. Because we make certain assumptions about the human body, certain positions outside of that range would not be adequately tracked. For instance, arms raised above the head would distort the bounding box and therefore the proportions of the body. The feet can also be raised significantly higher if a person wishes to, which does not fit our assumption that the feet are roughly in the bottom percentage of the bounding box.

Lastly, the motion cue analysis we do could be more robust. Since we only measure the change in angle between the torso and the feet, we are not taking advantage of the specific motions of the joints we are tracking.

Based on these drawbacks, there numerous improvements that can be made for the future progress of this system. A better background subtraction algorithm could greatly improve the results of the rest of the system. In particular, the Grimson method of background subtraction looked promising for cleaner segmentation. In addition, a more sophisticated algorithm for detecting the body proportions as described in [REF] by finding and fitting the range of angles for each joint in order to get a wider range of motion in our model fitting. However, this has the potential drawback of not being performant enough to run efficiently in real time. A more robust range of motions detected by the system can make it extendable to more sophisticate gait analysis for applications like medical rehabilitation purposes.

**References**

[1] M. Taha Khan, 'Motion Cue Analysis for Parkinsonian Gait Recognition', *The Open Biomedical Engineering Journal*, vol 7, p. 1, 2013.

[2] J. Vignola, J. Lalonde and R. Bergevin, 'Progressive human skeleton fitting', pp. 35--42, 2003.

[3] P. KaewTraKulPong and R. Bowden, 'An improved adaptive background mixture model for real-time tracking with shadow detection', *Springer*, pp. 135--144, 2002.

[4] S. Suzuki and others, 'Topological structural analysis of digitized binary images by border following', *Computer Vision, Graphics, and Image Processing*, vol 30, iss 1, pp. 32--46, 1985.

[5] G. Borgefors, 'Distance transformations in digital images', *Computer vision, graphics, and image processing*, vol 34, iss 3, pp. 344--371, 1986.

[6] P. Viola and M. Jones, 'Rapid object detection using a boosted cascade of simple features', vol 1, p. --511, 2001.

[7] Jeffrey E. Boyd and James J. Little. 2003. Biometric gait recognition. In Proceedings of the 1st international conference on Advanced Studies in Biometrics (ASB'03), Massimo Tistarelli, Josef Bigun, and Enrico Grosso (Eds.). Springer-Verlag, Berlin, Heidelberg, 19-42.

[8] Kim, D.; Kim, D.; Paik, J., "Gait recognition using active shape model and motion prediction," Computer Vision, IET , vol.4, no.1, pp.25,36, March 2010.

[9] Rong Zhang, Christian Vogler, and Dimitris Metaxas. 2007. Human gait recognition at sagittal plane. Image Vision Comput. 25, 3 (March 2007), 321-330

[10] Urtasun, R.; Fua, P., "3D tracking for gait characterization and recognition," Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on , vol., no., pp.17,22, 17-19 May 2004