A Computer Vision-Based System for Stride Length Estimation using a Mobile Phone Camera

Wei Zhu[1], Boyd Anderson[1,2], Shenggao Zhu[1,2], Ye Wang[1,2]
[1]School of Computing, National University of Singapore, Singapore
[2]NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore
juilangchu@icloud.com,{boyd,shenggaozhu}@u.nus.edu, wangye@comp.nus.edu.sg

ABSTRACT

Conditions such as Parkinson's disease (PD), a chronic neurodegenerative disorder which severely affects the motor system, will be an increasingly common problem for our growing and aging population. Gait analysis is widely used as a noninvasive method for PD diagnosis and assessment. However, current clinical systems for gait analysis usually require highly specialized cameras and lab settings, which are expensive and not scalable. This paper presents a computer vision-based gait analysis system using a camera on a common mobile phone. A simple PVC mat was designed with markers printed on it, on which a subject can walk whilst being recorded by a mobile phone camera. A set of video analysis methods were developed to segment the walking video, detect the mat and feet locations, and calculate gait parameters such as stride length. Experiments showed that stride length measurement has a mean absolute error of 0.62 cm, which is comparable with the "gold standard" walking mat system GAITRite. We also tested our system on Parkinson's disease patients in a real clinical environment. Our system is affordable, portable, and scalable, indicating a potential clinical gait measurement tool for use in both hospitals and the homes of patients.

Keywords

Gait Analysis; Computer Vision; Mobile Camera; Video Analysis; Parkinson's Disease; Movement Disorder

1. INTRODUCTION

The world population has been growing rapidly in recent decades, with more than seven billion people living worldwide[1]. Along with this growing trend, the world population is aging at an unprecedented rate, and this brings profound implications for many facets of human life[2]. One of the most serious implications is age-associated chronic diseases that deteriorate the quality of life of elderly people, such as neurodegenerative disorders including Parkinson's disease, Huntington's disease, and Amyotrophic Lateral Sclerosis.

Parkinson's disease (PD) affects an estimated seven to ten million people worldwide, and its incidence increases with age[3]. PD is characterized by decreased motor control abilities, causing symptoms which typically include bradykinesia (slowness of movement), rest tremors, rigidity, and postural and gait impairment. Currently there is no cure for PD, and PD patients rely on medicine such as Levodopa to control the symptoms.

As a chronic disease, PD progresses slowly and its onset is usually difficult to detect. It has a long progression time, which can be divided into different stages. Diagnosis of the PD stage of a patient is an important clinical task that also affects the treatment of the patient. In rural or community hospitals where neurologists and specialized medical equipment are not available, doctors have to diagnose a patient's PD stage based on experience and some simple tests. Therefore, there is a great need for inexpensive, accessible, and objective tools that can be used for PD diagnosis in rural hospitals or hospitals in developing countries.

Mobility assessment is a standard non-invasive PD test, which aims to assess the degree of mobility impairment via the quantification of limb movement. Specifically, gait analysis is widely used to evaluate lower motor performance. Some of the fundamental gait measures include stride time and stride length, as well as the variability of these measures. Experiments have shown that PD patients usually have reduced cadence and increased variation during walking. For example, a study showed that PD patients had a much larger step time variability (7%) than healthy controls (4%) [6].

Clinical gait measurement normally uses pressure sensitive walking mats [17] or motion-capture cameras or camera arrays [19]. These devices are costly and often require technical expertise to use, and thus are difficult to scale up. For those motion-capture systems, usually highly specialized cameras are used, such as Vicon cameras[4]. However, cheap, general-purpose cameras on mobile phones are now ubiquitous. Many smartphone (e.g., iPhone 6S, Samsung Galaxy S5 and newer) cameras are capable of recording videos at 4K resolution at 30 frames per second (FPS) or higher. These modern smartphones have become a potential platform for camera-based gait analysis.

[1]https://ourworldindata.org/world-population-growth/
[2]http://www.un.org/esa/population/publications/worldageing19502050/

[3]http://www.pdf.org/en/parkinson_statistics
[4]http://www.vicon.com/

ASSETS '16, October 23-26, 2016, Reno, NV, USA

DOI: http://dx.doi.org/10.1145/2982142.2982156

2. RELATED WORK

There are many studies that focus on analyzing PD patient's activities using different methods and tools. Inertial sensors such as accelerometers and gyroscopes have been used to record patient motion [23, 12, 33, 5]. A mobile phone application [18] was introduced to quantify the Timed-up-and-go test which is common for PD assessment. It also uses the inertial sensors of the mobile device to capture the signals of the patient's gait. One study [23] designed a biometric suit with inertial sensors to simultaneously measure the gait oscillation from eight major joints (knees, hips, elbows and shoulders) of a human body. Another inertial measurement system was developed with several IMUs attached to the subject's feet [12]. These studies use multiple wearable inertial sensors to collect motion data, and then calculate gait parameters based on the collected data, which is not very convenient, especially for severe PD patients. Moreover, the data collected from inertial sensors is not straightforward to analyze, requiring complicated algorithms to extract even simple gait parameters.

Many studies also use specialized cameras for motion analysis. Microsoft Kinect cameras are commonly used in marker-less tracking systems which provide both a color and depth image. There are some studies which use these images to extract motion parameters. For example, a Kinect camera was used in a study [8] to track the center of mass of a person. Another study [22] focuses on the effects of playing Kinect Adventures on the postural control of patients with Parkinson's disease. These studies use relatively expensive and specialized cameras to track human motion, which require simple environments to avoid noise caused by background objects and lighting. A disadvantage of marker-less tracking systems is that their accuracy is affected by complicated movements. This is not ideal for tracking PD patients with abnormal gait patterns.

There are some other commercial gait measurement tools based on pressure sensors, such as GAITRite[5] [17] (a mat embedded with pressure sensors) which records foot steps when the subject walks on the mat. It does not require the subject to wear any special sensors on the body. However, it is unable to detect foot movement above the mat and thus can not obtain the full picture of the gait cycle. Moreover, GAITRite is too expensive to be widely adopted in the homes of patients or even some hospitals.

Based on the above analysis, this paper presents a computer vision (CV) based gait analysis system using a smartphone camera. It records a video of a subject walking on a mat, and calculates the subjects stride times and lengths. The stride length measurement accuracy is comparable with the "gold standard" walking mat GAITRite. A set of video analysis algorithms are developed to segment the videos, locate the mat and shoes, and estimate stride lengths. The detailed system description and algorithms will be presented in the following sections.

3. SYSTEM OVERVIEW

Due to clinical constraints, the proposed system needs to meet certain usability requirements. We placed a strong emphasis on portability, ease of use, cost reduction, and accuracy. As our target users are rural hospitals, hospitals in developing countries, and home users, we require the system to cause minimal interference to the patient and work in a wide range of environments with differing lighting conditions, backgrounds, and walking styles. The basic idea of the system is to use video footage of patients walking on a specially designed mat (see Figure 1 & 2) to automatically locate the foot position on the mat and then estimate the stride lengths of the patient. The system should be a complete solution for gait analysis, providing important gait information and statistical analysis. To this end, the system accepts a recorded video of a patient walking on a mat and outputs estimations of stride lengths.

[5]http://www.gaitrite.com

Figure 1: System test environment. Note the smartphone on a tripod and the mat on the ground.

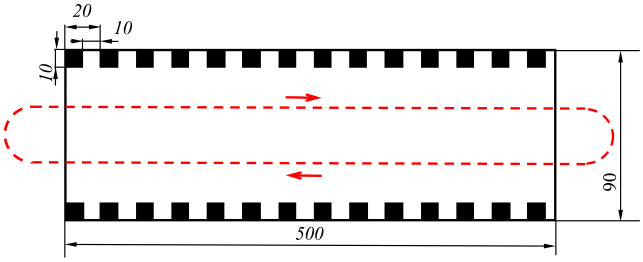

Figure 2: Walking mat. All sizes in the figure are in cm. The red dashed line denotes the walking path in our experiment.

The pipeline of the system is presented in three main components: Mat Extraction, Shoe Detection and Stride Length Estimation. Figure 3 illustrates how the components are connected, with a detailed work-flow for each component. The first component (Mat Extraction) analyses the recorded video to find sections where the patient is not in the frame. It uses these sections to extract the relative orientation of a specially designed mat to the camera and acts as a ruler for later stride length estimation. The second component (Shoe Detection) uses the same video and selects the sections where the patient is walking on the mat. This component finds the closed contour of the shoe using a selection of computer vision and Computer Aided Geometric Design (CAGD) algorithms. The third component (Stride Length Estimation) uses this closed shoe contour in combination with the previously extracted mat information to map the shoe contour position to real world units and thus estimate stride length. This component uses a mapping function which relates pixels to meters, and uses dif-

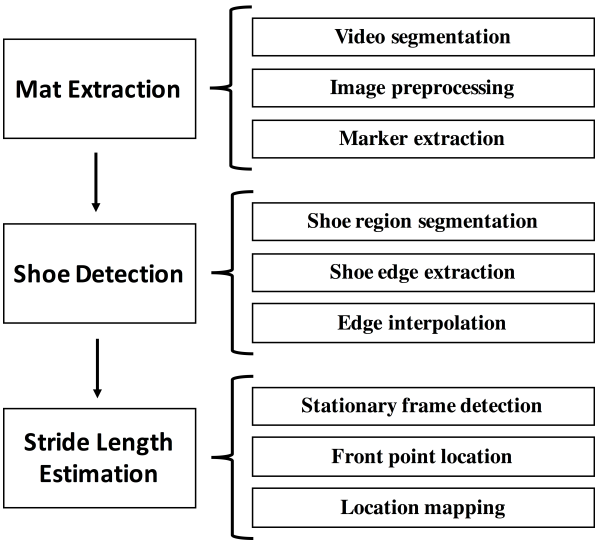

Figure 3: System Pipeline

ferent strategies to determine where the front of the shoe is (even under perspective change).

To keep within a modest budget, our system consists of a Samsung S5 phone, a printed PVC walking mat, and a pair of shoes which have distinct and uniform colors. The camera is placed on a tripod and pointed at the mat (See Figure 1). We record 1080p video at 30fps and then transfer the video to a computer running our gait analysis software. In the next three sections we will detail the three components of the system.

4. MAT EXTRACTION

In order to extract stride lengths from a video we will need some point of reference to figure out the scale and orientation of the recorded environment. Therefore we design a simple mat which acts as this reference point, giving us both scale, and relative orientation of the mat to the camera. It also serves as a ruler for stride length estimation. In this section, we will describe the mat used in our system, and then discuss the method for automatically extracting perspective information from it.

4.1 Mat Design

We designed our mat to be durable, low cost, simple, and to not cause undue dizziness or vertigo in those walking across it. The mat consists of alternating black and white markers printed on the long edges of a 0.9 m × 5.0 m sheet of PVC material. The markers, each 10 × 10 cm in size, are used to calculate the orientation of the mat and provide a ruler in the final system component (Section 6) for stride length estimation. The middle of the mat is clear in order to both simplify shoe detection in Section 5 and to stop the subject getting vertigo from the high contrast edges. The completed mat design is shown in Figure 2.

We selected PVC material as it is easy to find in printing shops around the world and can be printed to any length or width. It is also durable, easy to clean, and heavy enough that it will sit flat on the ground. Furthermore, this material is non-reflective and therefore it will not interfere with video recording.

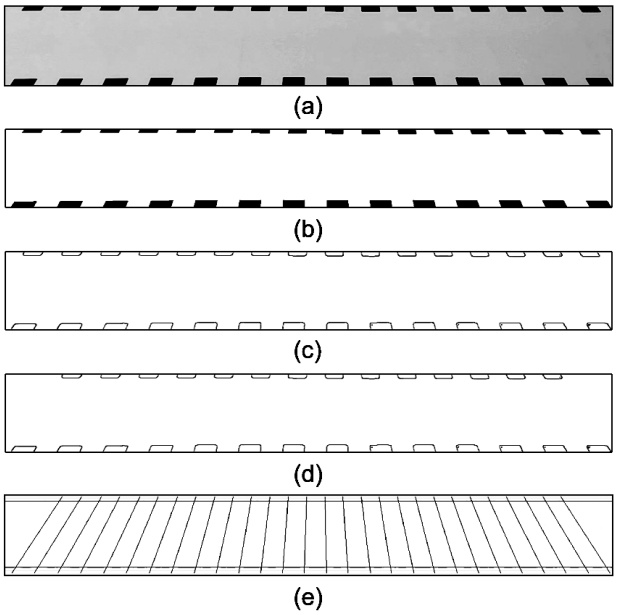

Figure 4: Mat extraction stages. (a) Original cropped mat image from video. (b) Binary mat image. (c) Marker contours. (d) Paired markers. (e) Perspective lines.

4.2 Perspective Information

In our system a smartphone camera is placed on a tripod and pointed towards the mat, but this position may not always be exactly the same. Therefore the goal of this component is to find the relative size, location, and orientation of the mat automatically so that the camera position does not have to be re-calibrated every time it is set up or bumped.

Before we do any processing on the recorded video it must be segmented into multiple sections, those with walking (walking videos) and those without (mat videos). This is performed using a simple background subtraction approach to detect motion [31, 4]. In this section we will process the mat videos, and in the next section we will use the walking videos. We define the perspective information as the extracted set of paired markers which make up the mat. From this set of paired markers we can ascertain the relative orientation of the mat to the camera and estimate distances in our real world co-ordinate system.

A procedure overview is detailed in Algorithm 1. The procedure for extracting perspective information from a video is automatic with the exception of the following step. First, we manually crop the frames of the mat video to include only the mat itself using a simple UI. Figure 4a shows the cropped image. We have not automated this step because it needs to be able to handle different conditions and environments and it is possible that the mat could blend into the background. Therefore we ask the user to specify the location of the mat. This step reduces both overall processing time and the impact of other objects in the frame. From this step onward, the system is completely automatic.

Next, we use a few basic image processing algorithms [21] to prepare the image for analysis. First we use the histogram equalization method to compensate for differing lighting.

Then we use use a basic Gaussian filter to remove the noise in the middle of the empty mat. Lastly, we transform the color image to a binary one using a threshold method. Figure 4b shows the processed image from the video.

With this binary image we now locate, count, and pair off the mat markers. For simplicity sake, we will define the bottom of the mat as the edge of the mat that is closest to the camera, and conversely the top of the mat is the edge furthest away from the camera. The idea is straightforward: we count the number of markers in the bottom of the original mat and then align these markers to the markers in the top of the mat. We use the Canny operator [2] to find the contours of all of the markers (see Figure 4c).

Due to the perspective of the camera, we find that there are a few markers at the top of the mat that have no corresponding markers on the bottom of the mat and therefore these need to be excluded. Here, we choose the line that passes through the left edge of the bottom first marker and shift the line 5 to 10 pixels to left. All top markers to the left of this line are removed from the binary image of the mat. Likewise, we do this on the right using the bottom rightmost marker, and therefore only the markers which can be paired remain. Figure 4d shows the mat after all of the markers have been paired, with the redundant markers removed.

Finally we choose critical points from each marker's left and right edges to draw the line between the bottom and top markers. We take the top and bottom marker edges, and then we calculate the line of best fit that connects them. This is repeated for all pairs of markers. To improve the success of this method we only use the edge points between 5% and 95% of the total height of the marker, thus removing any influence of the corner of the marker. Figure 4e shows the final perspective information as displayed on the mat.

Algorithm 1 Mat Extraction Algorithm

```
procedure MAT–EXTRACTION
    Crop frame of mat video
    Perform histogram equalization operation
    Remove noise using Gaussian Filter
    Convert frame to binary image
    Use Canny operator to extract edges of markers
    Count the number of markers
    Exclude unpaired markers
    Connect paired markers with a line of best fit
end procedure
```

To get high accuracy perspective information we recommend the following steps. Ensure the camera is stable and record in as close to a uniform lighting environment as possible. It is also important to fine tune the parameters for the techniques used to get a clear contour of the edge of the markers. Even with this calibration we may not always get a continuous contour of the edge, in this event we can use the technique described in Section 5.2 to find a continuous contour. Next, we describe the process to detect the shoe and find its contour.

5. SHOE DETECTION

This component analyses the sections in a video where the patient is walking on the mat whilst wearing uniformly colored shoes. The goal here is to find the contour of both shoes to be used in the final component to estimate stride length. In this section, we describe the procedure for shoe detection. The procedure involves detecting the regions of the shoes based on the color of the shoes, and then extracting the closed contour of the shoes. To make the computation easily parallelizable, the video is first decomposed into a sequence of images, and shoe detection is performed on each image independently.

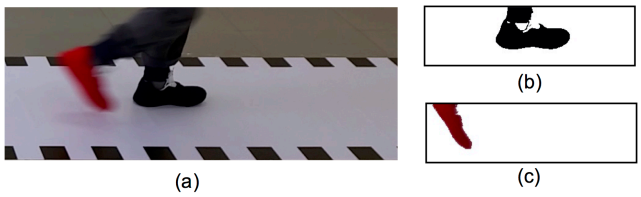

Figure 5: Shoe region segmentation. (a) Original frame. (b) Left shoe region. (c) Right shoe region.

5.1 Shoe Region Segmentation

First, each source image is cropped to the same size and location as the empty central region of the mat. This step removes both the background and the mat markers from the frame. Next, we need to reduce the noise from the cropped image. There are many sources of noise such as those caused by shadows under the feet, lighting differences, and objects that have fallen on the mat during walking (such as dust and hair). Here we choose the median filter, because our goal is to extract the contour of the shoe, and the median filter performs better at preserving edges compared to simple average filters. After applying the median filter to the image, most of this noise is removed.

In order to track both feet, our system needs the colors of the left and right shoe. In our experiments, we use red for the right shoe and black for the left, as shown in Figure 5a. While it is not necessary for the feet to use these exact colors, we recommend that the feet colors have high contrast with each other and the mat. This is because the lighting conditions may vary over a recording or even vary within the same walk. This causes the shoe color in the image to change significantly. Thus using shoe colors which are sufficiently different from the mat color will produce the best results.

Given the color of each shoe, we can segment the shoe region from the image based on the similarity between the shoe color and the color of each pixel. The shoe region is formed by the pixels which have a high color similarity (using a predefined threshold) to the shoe color. Figure 5b and 5c show the segmented left and right shoe regions, respectively. In our system, we adopt the commonly used CIE94 algorithm to compare two colors [16].

5.2 Contour Detection

After finding the shoe regions, we need to find the closed contour of each shoe, which is used to determine the exact location of the shoe. To obtain an accurate contour of each shoe, we first detect the edges of the shoe, which can be achieved using the Canny operator [32, 1, 20, 27]. For our system we choose to modify the normal operator to achieve a better edge detection result. In a normal Canny operator, an image is first filtered using a Gaussian filter before finding the intensity gradient of the image. However, the Gaussian filter is not very good at preserving the edges. There-

Algorithm 2 Shoe Detection Algorithm

```
1: procedure SHOE DETECTION
2:     Crop frame of walking video
3:     Detect edges using modified Canny operator
4:     Use edge traversing to find edges belonging to shoe
5:     while largest gap between edges > ϵ do
6:         Lower edge detecting threshold for operator
7:         Detect edges using modified Canny operator
8:         Use edge traversing to find edges
9:     end while
10:    Interpolate edges to get the closed contour
11: end procedure
```

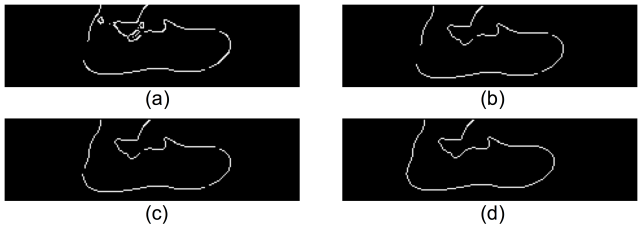

Figure 6: Modification on Canny operator result. (a) Original detected result. (b) Edge traversing result. (c) Reducing gap size. (d) Interpolation result.

fore, to achieve better edge detection results, we modify the Canny operator by instead using the bilateral filter [26, 29] to remove noise, as it performs better at edge-preservation. Although computationally expensive, it is possible to use general purpose GPU computing techniques to speed up the Canny operator [11, 25] even on mobile hardware.

Next we use our modified Canny operator to detect the edges of the shoe (see Figure 6a). We find that the initial edges obtained by the Canny operator cannot guarantee a closed contour of the shoe region. Instead, there are often many gaps in the front of the shoe. This makes it difficult to find the front point of shoe and will cause large errors in the estimation of shoe location. One of the solutions for this problem is to adjust the parameters of Canny operator, but this method usually causes other side effects (e.g., too many unwanted edges) and is not suitable for the automation of the whole processing procedure. Another solution is to use Computer Aided Geometric Design (CAGD) algorithms [9, 24] to modify the initial edges.

Since we cannot always obtain a closed contour using the edges extracted by the Canny operator, we need to use fitting methods to fill in the gaps between edges to make it closed. For this task we use B-spline curves, which have the advantages of being continuous and allowing fine local control. For the B-spline interpolation curve, the main question is how to calculate the so called control points on the shoe edges [15, 13, 30, 14]. Since there are many different types of shapes on the contour, we cannot simply use a uniform function to choose the feature points [28]. In this paper, we propose a new algorithm to find the feature points from all of the discrete points located on the shoe edges. Then we use non-uniform cubic B-spline interpolation method to obtain a closed shoe contour.

Edge traversing methods normally come from two main strategies. One is based on Run-length [7], and the other is based on Chain Code methods [3]. Regardless, both of them have the disadvantage of repeatedly outputting some edges or missing some interior edges. Our method is based on 3D reconstruction techniques used in medical imaging [28] which can quickly extract a complex contour in one scan. After the edge traversing process, we get a single pixel-width contour as shown in Figure 6b. If we find large gaps between adjacent edges, we can dynamically adjust the parameters of the Canny operator to decrease the gaps to be smaller than a predefined threshold ϵ. The trade off is we will also extract more redundant edges in the process. The improved edges extracted are shown in Figure 6c.

To fill the gaps among the edges to form a closed contour, we used an interpolation algorithm based on the non-uniform cubic B-spline curve. First, we need to find the feature points from an edge line. As illustrated in Figure 9c, for the $j-$th point (i.e., a pixel) P_j in an edge line, two of its neighbor points P_{j-s} and P_{j+s} are selected, where s is a predefined neighbor range (e.g. $s = 4$). We proposed a new method for easy estimation of the vertex curvature at P_j, which is calculated as the perpendicular distance (H_j) from P_j to the line formed by P_{j-s} and P_{j+s}. If H_j is less than a preset threshold, P_j is selected as a candidate of the feature points. Then, a non-minimum suppression method [28] is adopted to choose the final feature points from the candidates. After obtaining the feature points, we apply the bi-directional accumulated arc length method [24] for parameterization in B-spline interpolation. Finally, control points are computed and non-uniform cubic B-spline interpolation curves are generated to connect the edges to form a closed contour [24]. The final closed contour is shown in Figure 6d. An overview of the procedure can be seen in Algorithm 2.

6. STRIDE LENGTH ESTIMATION

This component finds the front point of the closed shoe contour and maps it to real world units using the previously extracted mat information. It also separates each stride from continuous walking and thus estimates stride length. An overview of the procedure can be seen in Algorithm 3.

Algorithm 3 Stride Length Estimation Algorithm

```
1: procedure STRIDE LENGTH ESTIMATION
2:     Find stationary frames
3:     Find front point on shoe contour
4:     Convert from pixels to cm using mapping function
5:     Estimate stride length
6: end procedure
```

6.1 Stride Detection

Stride length is the displacement between two successive foot strikes on the ground of the same foot. We denote the video frame that separates two adjacent strides as a "stationary frame" (i.e., the frame in the middle of the stationary stance phase). Therefore, the stride length can be calculated as the distance traveled by the same foot at two adjacent stationary frames.

We propose two methods to find the stationary frames. In the first method, we take advantage of the nature of walking as there are alternating periods when the foot is stationary on the ground and periods when the foot is moving above the

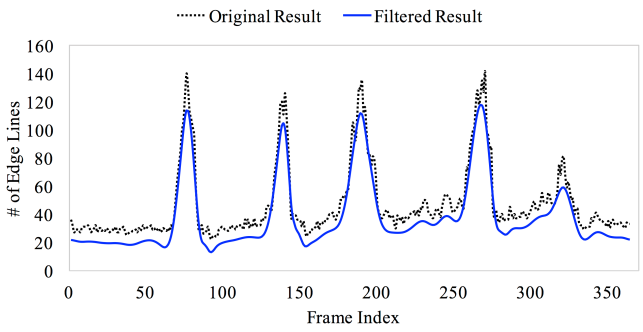

Figure 7: Stationary frame detection based on the number of edge lines in each frame.

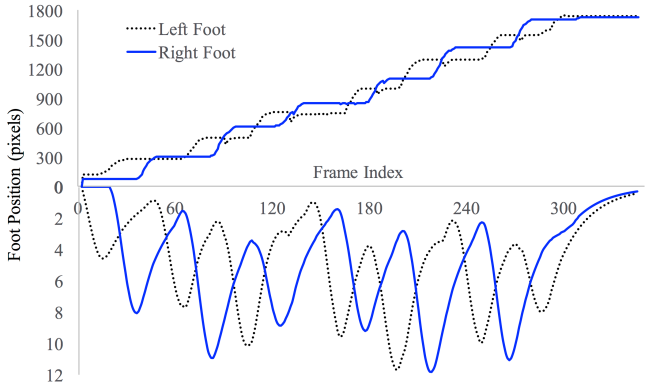

Figure 8: Front point position on foot contour

ground. This leads to our key observation that the frames are less blurred when the foot is stationary and more blurred when the foot is in motion. As noted earlier, the Canny operator will output more edge lines if the image is blurred. So we can use this feature to find the number of edges in each frame of the whole image sequence. The dashed line in Figure 7 illustrates the detected number of edges in every frame. We can easily find, for one foot, that there are a few peak points. The middle point between two adjacent peaks is therefore the frame in which the foot is on the ground (i.e., the stationary frame). To facilitate peak detection, we apply a low-pass filter on the original signal to smooth out the waveform (the solid line in Figure 7).

This stationary frame detection method is simple and effective. However, it may be affected by other noise in the frame. For example, if the clothing of the subject is still moving after the foot is stationary, the captured frame can still be blurred. If the subject is walking at a very slow speed, the frame will not be blurred much, which leads to less dominant peaks in the waveform of the edge lines. Nevertheless, with well-tuned parameters in filtering and peak detection, this method worked successfully in stride detection in our experiments.

The second method we propose for stationary frame detection is based on the position of the foot in the frames. We can take the rightmost point in the shoe contour as the position of the foot. The upper panel of Figure 8 shows the foot position (in pixels) in each frame. We can observe that the foot position remains level when the foot is stationary on the ground, and it increases along a rising edge when the foot moves forward. The foot furthest away from the camera has small fluctuations in its foot position waveform which are caused by the foot closest to the camera occluding it as it moves by. These small fluctuations can be smoothed out using a low-pass filter.

To separate the strides, we take the first order difference of the foot position signal, resulting in an approximately periodic waveform with clear peaks and troughs. This difference waveform is then low-pass filtered, as shown in the lower panel of Figure 8. As before, a peak detection algorithm is used to find the local minimum points in the difference waveform, which correspond to the stationary frames. Unlike the first method, this method is more robust to slow walking.

6.2 Mapping Function

For each stationary frame, we find the front point in the shoe contour, which is used to represent the foot position at pixel-level. First, we obtain the $x-$coordinate (along the mat direction) of each point in the shoe contour. We also use the edge traversing method to find the front point in the shoe contour, which is the rightmost point for left-to-right walking, or the leftmost point for right-to-left walking.

Once we obtain the front points, we use a mapping function to transform the front point position (x) at pixel-level into real world position (d) in cm. There are several choices for the mapping function. A basic algorithm uses the average distance of a pixel (d_p), which is calculated as the total mat length (in cm) divided by the total pixels of the mat. The real position of the front point is obtained as $d = xd_p$. However, the above algorithm ignores the perspective differences for the pixels.

To mitigate these problems, we designed a new mapping function (referred to as Method-1 in Section 7), which is illustrated in Figure 9a. This method first finds two marker edge lines (L_1 and L_2) that are closest to the front point. The real position of these two lines (d_1 and d_2 respectively) are known from the markers. Then the perpendicular distances (λ_1 and λ_2) from the front point to the two lines are calculated at pixel level. The real position is derived proportionally as

$$d = \frac{\lambda_2 d_1 + \lambda_1 d_2}{\lambda_1 + \lambda_2}. \quad (1)$$

Since the two marker lines L_1 and L_2 are generally not parallel due to the perspective of the camera, we designed a second mapping function (referred to as Method-2 in Section 7), as illustrated in Figure 9b. If L_1 and L_2 are parallel, it reduces to the first mapping function. Otherwise, we assume L_1 and L_2 intersect at point Q. Similarly, we estimate the real position based on the proportion of the arc length α_1 (formed by line PQ and L_1) and arc length α_2 (formed by line PQ and L_2), where the arc lengths are proportional to the `sin` of the angle θ_1 (between line PQ and L_1) and θ_2 (between line PQ and L_2), respectively.

$$d = \frac{\alpha_2 d_1 + \alpha_1 d_2}{\alpha_1 + \alpha_2} = \frac{d_1 \sin\theta_2 + d_2 \sin\theta_1}{\sin\theta_1 + \sin\theta_2}. \quad (2)$$

Another strategy which provided an improvement in accuracy was to compensate for the small errors in the man-

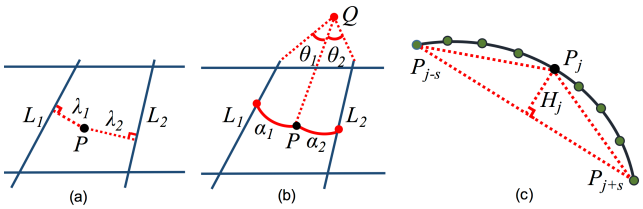

Figure 9: (a) Mapping function using perpendicular distance. (b) Mapping function using arc length. (c) Estimation of vertex curvature.

Figure 10: Finding the front point. (a) Original method. (b) Finding a small region in the front of the contour.

ufacture of the mat. Instead of using the assumed width of each marker we measure them individually, and directly use the real distance rather than our print specifications and then apply the Method-2 approach. This method is denoted as Method-3 in Section 7.

6.3 Front Point Selection

The previous analysis used a naive approach to find the front of the shoe, taking the maximum point on the shoe contour (see Figure 10a). Certainly, this maximum point on the contour may not always be the front most point on the shoe as the contour can change under rotation. To compensate for this issue we propose a new iterative testing method to find a better estimate for the location of the front of the shoe.

This method is based on the assumption that the true front of the shoe exists in the small region enclosed in the front of the foot contour. We first use a threshold to define a small region near the maximum point on the contour. Next, we use the mapping function to calculate the position in real units of all the points inside this region. Finally, we pick the rightmost point in this region as the front point on the foot. Figure 10b shows the area that we use in this method. This refined front point selection method in combination with Method-3 is labelled as Method-3+R in Section 7.

7. SYSTEM ACCURACY

Now that we have a system for estimating stride length we must quantify the accuracy of its output. To test our system we designed some imitation experiments in different environments. To do this we must compare our CV stride length estimates against ground truth measurements. We aligned a ruler to the edge of the mat and took a high quality photo for every step taken using a secondary camera and then directly measured the stride length. Whilst this ground truth measurement is simple and inexpensive, it is also cumbersome and requires a lot of busywork.

We invited five subjects to mimic PD patients with different walking requirements. The accuracy of the system was tested in one room, over multiple days, at numerous times during the day. This caused a large variation of lighting and coloring on the mat. The non-PD subjects were shown videos of PD patients and were asked to imitate their movements: failing to initiate walking (in PD fields known as "start hesitation"), walking with asymmetric gait patterns, and simulating the common "freezing of gait" symptoms. In this experiment, we collected 128 video sessions with 382 strides. Table 1 shows the accuracy of all three methods. It can be seen that Method-2 outperforms Method-1, and Method-3 outperforms both. Additionally, using the improved front point selection approach in combination with Method-3 further decreases the error (Method-3+R in Table 1). We found the mean absolute error of the stride length of both feet to be 0.62 cm for Method-3+R, which is comparable with "gold standard" tools such as GAITRite.

8. CLINICAL TEST

Now that we have an estimate of the accuracy of our system, we try to investigate its practicality and usefulness in practice. Therefore we tested our system in a real world clinical environment using real patients with severe gait issues.

8.1 Procedure

We recruited 55 elderly subjects, 44 PD patients and 11 healthy controls with help from Huashan Hospital located in Shanghai, China. The PD patients came from four severity groups according to the Hoehn and Yahr Scale (HY) [10]. The subjects recruited in our experiment were classified into HY1 (10), HY2 (14), HY3 (10), HY4 (10) and healthy controls (11) by neurologists. Our inclusion criteria only allowed subjects who (1) were 50 to 75 years old, (2) were able to walk by themselves without help, (3) had no other serious diseases, (4) were able to understand and sign the consent forms. This test had IRB approval and was performed in accordance with hospital ethics board requirements.

The procedure in this test involved a subject walking from left to right, turning around at the end and walking back as shown in the Figure 2. We did not limit the duration for walking, and encouraged subjects to walk for as long as they were able such that we could get a minimum of thirty strides on each foot. One "session" was defined as walking from one side to the other side (Left to Right or Right to Left). Subjects would repeat this process numerous times.

8.2 Results

We obtained 98 videos consisting of 1947 walking sessions from the 55 subjects over a period of two weeks. In our test, we processed all videos and our system was able to correctly identify every shoe contour in comparison with human annotated data. It also correctly counted every stride even in severe patients with abnormal cadence or gait patterns. Finally, it correctly extracted the walking mat in all videos. While we were unable to test the stride length accuracy on these patients due to the cumbersome nature of our ground truth measures, we were able to test the practicality of using the system in a real world scenario. Importantly, we could detect all subject's shoe contours even under very different lighting conditions, gait styles, gait abnormalities, and with subjects who wore a wide array of different clothing.

Table 1: Absolute stride length error between ground truth and estimations from different methods

	Left Foot				Right Foot			
	Method-1	Method-2	Method-3	Method-3+R	Method-1	Method-2	Method-3	Method-3+R
Max (cm)	11.36	4.10	2.77	1.75	12.14	4.50	4.07	1.87
Min (cm)	1.45	0.65	0.01	0.04	0.06	0.04	0.15	0.06
Mean (cm)	5.97	2.52	1.31	0.59	5.83	2.54	1.62	0.65
SD (cm)	2.79	0.84	0.67	0.40	2.96	1.12	0.97	0.48

Table 2: Example of a basic statistical report

Parameters	Left Foot	Right Foot
Basic Statistical Analysis		
Mean Stride Length (cm)	73.69	73.66
Mean Stride Time (s)	0.96	0.96
# of Strides	75	74
Stride Length		
Standard Deviation (cm)	11.14	9.5
Coefficient of Variation	0.15	0.13
Stride Time		
Standard Deviation (s)	0.06	0.05
Coefficient of Variation	0.15	0.05

To provide information to doctors or clinicians, the system is able output the subject's basic statistical information, including mean, coefficient of variation, and standard deviation of left and right stride lengths and times (an example is shown in Table 2). These results can be used to generate many gait measures of interest to clinicians. The system can be used in combination with other systems as it imposes minimal interference to the subject. Some patients preferred to have someone walk beside them as they performed the walking task. Our system could handle this scenario, meaning that aided walking (for severe PD patients) could also be analyzed.

9. CONCLUSIONS

This paper presented a novel system for accurate estimation of stride lengths of Parkinson's disease patients based on computer vision techniques. It serves the needs of neurologists who want to ascertain PD progress by way of gait analysis, even in rural hospitals or hospitals in developing countries. Our clinical test demonstrated that our system supports various testing environments and is suitable for use in a clinical setting.

Our system has a few limitations, mostly due to the use of the contour of the shoe which is not invariant under rotation. This requires clever techniques to compensate for this perspective shift which does not occur in other motion based systems. We also have a limited recording width as we are constrained by the field of view of the camera, however with the use of higher resolution cameras in modern smartphones and external optics such as fish eye lenses it is possible to overcome this limitation without losing accuracy. Stride times are also limited by the refresh rate of the camera, however modern smartphones can support 60 fps or higher.

The accuracy of the system is dependent on many factors, such as lighting, filming distance, lens type, frame rate, and video quality. In our experiments and trials, we fixed the video quality, frame rate, and filming distance. However, lighting changed throughout recording and despite this we were able to identify the contour of the shoe correctly, in part due to the high contrast colors used between the shoes and mat. Two major errors were identified, those associated with the parallax effect between the shoe and the markers, and the fact that the front part of the shoe contour does not always coincide with the front part of the shoe. In the paper we proposed methods to get around these constraints and limitations.

Despite these limitations our system has many advantages over other commercial systems such as high accuracy, low price, portability, and scalability. The cost of our system is less than $800 USD: $200-$700 USD for a smartphone (which many users already have), $50-$80 USD for the printed mat, and $20 USD for the camera tripod. In contrast, commercial gait measurement systems are orders of magnitude more expensive. GAITRite (a pressure sensitive mat-based system) costs $36,000 USD, Vidcon motion capture systems can retail for $45,000 USD, and APDM (an inertial measurement sensor-based system) costs approximately $10,000-$18,000 USD. As the system only has two main components (the smartphone and the mat), it can be easily transported and set up. The system does not require a lab setting, or clinicians and is therefore suitable for the home.

For future work, we believe the system could be scaled to run entirely on a smartphone as the algorithms have low time complexity and could be computed on mobile GPU hardware. The use of improved mapping functions, perspective modeling, and better front point selection will definitely reduce estimation errors. A study of the stride and step length accuracy on a group of elderly subjects (including PD patients) is also an important step for the validation of the system. Additionally, these same algorithms proposed in this paper could be used to detect the hand movements of PD patients to quantify tremor, another important symptom for PD progression. Finally, testing the system concurrently with commercial systems such as GAITRite is the next step to validating the system.

In conclusion, we have presented a pipeline for a low cost system for gait analysis which is easy to use, portable, and accurate. The use of CV algorithms and smartphone cameras has the potential to make gait analysis cheap, accessible, and ubiquitous even in hospitals in rural areas or in developing countries.

10. ACKNOWLEDGEMENTS

The authors would like to thank Prof. Jian Wang, Miss. Ke Yang, and the entire Huashan team for their assistance in the data collection. This project was partially funded by a NUS Cross-faculty research grant R-252-000-567-133 and a FOE-SOC Joint Collaboration grant R-252-000-588-112.

11. REFERENCES

[1] S. Agaian, A. Almuntashri, and A. Papagiannakis. An improved canny edge detection application for asphalt concrete. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 3683–3687. IEEE, 2009.

[2] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, 1986.

[3] V. Chalana, W. S. Costa, and Y. Kim. Integrating region growing and edge detection using regularization. In *Medical Imaging 1995*, pages 262–271. International Society for Optics and Photonics, 1995.

[4] D. Chinchkhede and N. Uke. Image segmentation in video sequences using modified background subtraction. *International Journal of Computer Science & Information Technology*, 4(1):93–104, 2012.

[5] B. T. Cole, S. H. Roy, C. J. De Luca, and S. H. Nawab. Dynamic neural network detection of tremor and dyskinesia from wearable sensor data. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 6062–6065. IEEE, 2010.

[6] M. F. del Olmo and J. Cudeiro. Temporal variability of gait in parkinson disease: Effectsof a rehabilitation programme based on rhythmic sound cues. *Parkinsonism & related disorders*, 11(1):25–33, 2005.

[7] S. Di Zenzo, L. Cinque, and S. Levialdi. Run-based algorithms for binary image analysis and processing. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 18(1):83–89, 1996.

[8] A. Dubois and F. Charpillet. A gait analysis method based on a depth camera for fall prevention. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 4515–4518. IEEE, 2014.

[9] G. E. Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.

[10] C. G. Goetz, W. Poewe, O. Rascol, C. Sampaio, G. T. Stebbins, C. Counsell, N. Giladi, R. G. Holloway, C. G. Moore, G. K. Wenning, M. D. Yahr, and L. Seidl. Movement disorder society task force report on the hoehn and yahr staging scale: Status and recommendations the movement disorder society task force on rating scales for parkinson's disease. *Movement Disorders*, 19(9):1020–1028, 2004.

[11] H. Gupta and D. S. Antony. Implementation of gaussian and box kernel based approximation of bilateral filter using opencl. In *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, pages 1–5. IEEE, 2015.

[12] D. Korotkin and K. Artem. Inertial measurement system for human gait analysis. In *Proceedings of the 8th International Conference on Body Area Networks*, pages 414–419. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013.

[13] H. Lin, G. Wang, and C. Dong. Constructing iterative non-uniform b-spline curve and surface to fit data points. *Science in China Series: Information Sciences*, 47(3):315–331, 2004.

[14] Y. Lu, K. Shi, J. Yong, H. Gu, and H. Song. A b-spline curve extension algorithm. *Science China Information Sciences*, pages 1–9, 2015.

[15] E. Margolis and Y. C. Eldar. Interpolation with nonuniform b-splines. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 2, pages ii–577. IEEE, 2004.

[16] R. McDonald and K. J. Smith. Cie94-a new colour-difference formula*. *Journal of the Society of Dyers and Colourists*, 111(12):376–379, 1995.

[17] H. B. Menz, M. D. Latt, A. Tiedemann, M. M. San Kwan, and S. R. Lord. Reliability of the gaitrite® walkway system for the quantification of temporo-spatial parameters of gait in young and older people. *Gait & posture*, 20(1):20–25, 2004.

[18] M. Milosevic, E. Jovanov, and A. Milenkovic. Quantifying timed-up-and-go test: A smartphone implementation. In *Body Sensor Networks (BSN), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[19] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2):90–126, 2006.

[20] H. S. Neoh and A. Hazanchuk. Adaptive edge detection for real-time video processing using fpgas. *Global Signal Processing*, 7(3):2–3, 2004.

[21] M. Petrou and C. Petrou. *Image Processing: The Fundamentals*. John Wiley & Sons, Ltd, 2011.

[22] J. E. Pompeu, C. Torriani-Pasin, F. Doná, F. F. Ganança, K. G. da Silva, and H. B. Ferraz. Effect of kinect games on postural control of patients with parkinson's disease. In *Proceedings of the 3rd 2015 Workshop on ICTs for improving Patients Rehabilitation Research Techniques*, pages 54–57. ACM, 2015.

[23] S. Shahid, A. Nandy, S. Mondal, M. Ahamad, P. Chakraborty, and G. C. Nandi. A study on human gait analysis. In *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, pages 358–364. ACM, 2012.

[24] F.-Z. Shi. Cagd & nurbs. *China Higher Education Press*, 2001.

[25] S. Srisuk, W. Kesjindatanawaj, and S. Ongkittikul. Real-time bilateral filtering using gpgpu. In *Applied Mechanics and Materials*, volume 781, pages 568–571. Trans Tech Publ, 2015.

[26] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[27] S. Vasilic, V. Hocenski, et al. Improved canny edge detector in ceramic tiles defect detection. In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pages 3328–3331. IEEE, 2006.

[28] Y. Wang, J. Zheng, H. Zhou, and L. Shen. Medical image processing by denoising and contour extraction. In *Information and Automation, 2008. ICIA 2008.*

International Conference on, pages 618–623. IEEE, 2008.

[29] Q. Yang, N. Ahuja, and K.-H. Tan. Constant time median and bilateral filtering. *International Journal of Computer Vision*, 112(3):307–318, 2015.

[30] L. Zhang, X. Ge, and J. Tan. Least square geometric iterative fitting method for generalized b-spline curves with two different kinds of weights. *The Visual Computer*, pages 1–12, 2015.

[31] S. Zhang, P. Li, L. Wang, and Y. Zhang. Gait image segmentation based background subtraction. In *Intelligent Computing Theories and Methodologies*, pages 563–569. Springer, 2015.

[32] P. Zhou, W. Ye, Y. Xia, and Q. Wang. An improved canny algorithm for edge detection. *Journal of Computational Information Systems*, 7(5):1516–1523, 2011.

[33] S. Zhu, R. J. Ellis, G. Schlaug, Y. S. Ng, and Y. Wang. Validating an iOS-based Rhythmic Auditory Cueing Evaluation (iRACE) for Parkinson's disease. In *Proceedings of the ACM International Conference on Multimedia*, pages 487–496. ACM, 2014.