TSBB11 – GAIT

Technical Documentation

Magnus Selin (magse761) Kevin Kjellén (kevkj515)

Rolf Lifvergren (rolli107) Christoffer Malmgren (chrma018)

John Stynsberg (johst529)

December 17, 2017

Contents

1 Introduction

Gait analysis is an important tool for medical applications to estimate and analyze walking patterns of patients. It may be used to detect stride abnormalities or verifying how well a new prosthesis is working for an amputee.

Systems that exist today are either dependent on motion capture, depth cameras, pressure mats, wearable sensors [6] or labor-intensive manual tracking [4]. These systems can be cumbersome to set up or expensive, in many cases both. For a clinic with less resources to be able to set up a gait analysis system none of the above mentioned methods are feasible, and a cheaper alternative is needed. That is the goal of this work: a system that can perform gait analysis with nothing but a video camera and a standard computer.

For instructions on how to install and use the software, see the User Guide[1].

1.1 Problem formulation

In this project two problems are treated: gait analysis from video with and without markers. The restrictions on the input videos are presented in Table 1.

Table 1: An overview of the constraints for video with and without markers.

Requirement	Marker	Markerless
Video has RGB color channels	X	X
Camera is stationary	X	X
Subject walks in a straight line	X	X
Subject walks perpendicularly to camera direction	X	X
Subject walks into frame	X	X
Subject walks out of frame	X	X
Left/right heel & toe are marked with circular colored markers	X	
Background should not share color with markers	X	
Background should not share visual appearance with the foot of the subject		X
Scene is well lit	X	X

From these videos the gait parameters in Table 2 are extracted for both feet.

[1] TSBB11 — GAIT User Guide

Table 2: Output gait parameter

Parameter	Detail
Duty factor (%)	Ratio of stride cycle in which the foot is on the ground.
Stance duration (sec)	Average duration of the period that the foot is on the ground.
Swing duration (sec)	Average duration of the period that the foot is not on the ground.

2 Related work

Similar work was done by Castelli et al. [3]. They used model based tracking of socks and underwear to estimate positions of feet and pelvis. This project's work differ from theirs since our system only extracts gait data from feet movement and tries to integrate completely markerless tracking.

Goffredo et al. [5] has developed a method where they use a controlled environment in which the background is a single color, enabling extraction of the subject's silhouette using a simple color thresholding scheme. The different body parts are then identified by assuming known body proportions. The study analyses hip and knee angle over the entire gait cycle. Different to their work, the system in this project primarily focuses on investigating the duty factor.

Saboune and Charpillet [8] have worked on a solution on markerless tracking with an ordinary consumer video camera to detect if an elderly person shows symptoms that (s)he is likely to fall. The entire system is supposed to be located in the subject's home and surveillance him or her without the video stream ever leaving the building. Only the gait parameters that can hint if a fall is likely to occur, should be transmitted. The system uses foreground segmentation and represent the body with a 3D-model with 19 joints connecting 17 segments. The state is represented with a interval particle filter in which a particle has 31 degrees of freedom.

3 Method

To solve the problem of automatic gait analysis in videos three different keypoint extraction methods are used. The goal of every method is to extract either the coordinates of the toe and heel, or one coordinate per foot, oftentimes the ankle. The methods used are:

- Detection of colored markers on the feet of the subject.
- Detection without markers with a convolutional neural network (foot/toe/heel detector).
- Detection with help of OpenPose, a pre-existing full-body pose estimation library.

The detection methods are described further in sections 3.2, 3.3 and 3.4, respectively.

The information about keypoint detections is then used for tracking with a constant-velocity model and estimation of gait parameters, using a method more or less independent of which detection method that is used. Each method thus only produces a list of possible detections for each frame. A separate system then links detections to tracks and performs the gait analysis. This tracking method is described in Section 3.5 and the parameter estimation is described in 3.6.

3.1 System overview

A flow chart of the system is given in Figure 1.

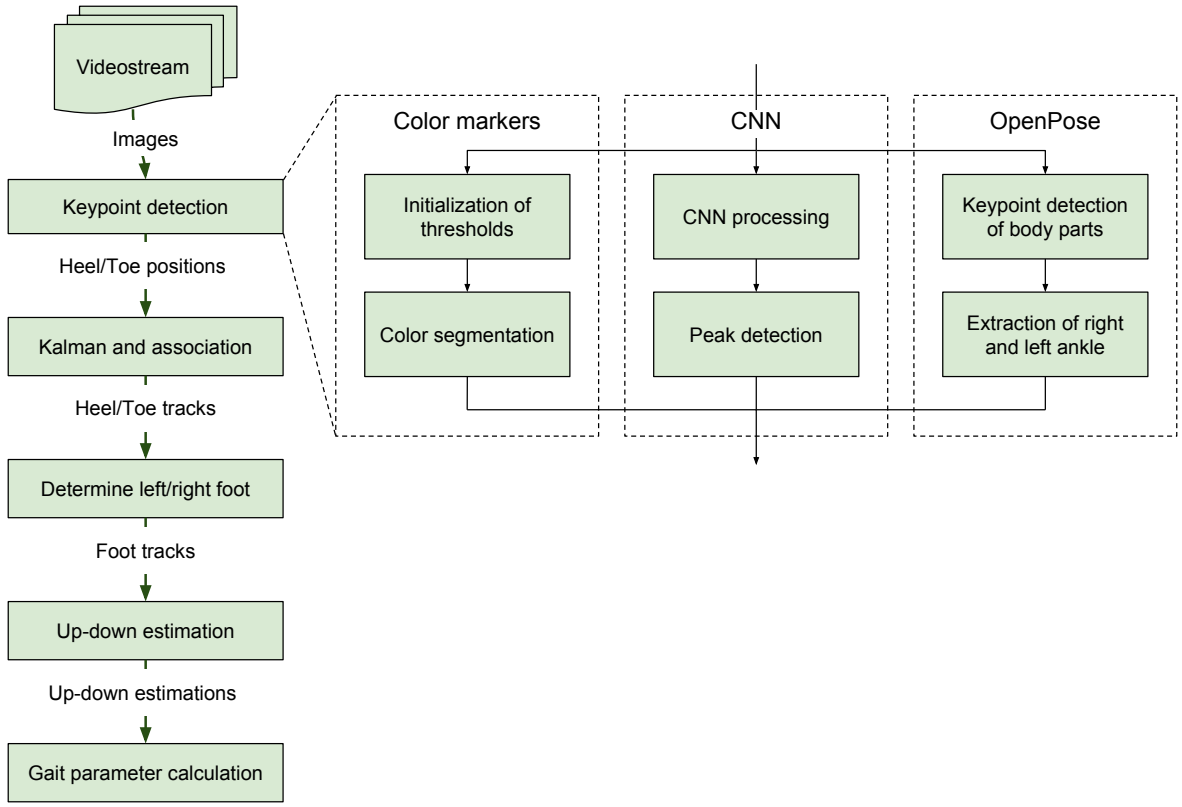

Figure 1: System flowchart. Boxes describe system modules. Arrows describe information flow between modules.

3.2 Detection with color markers

The solution used to detect the colored markers is as follows.

1. The user inputs the threshold levels that when applied on the HSV-transformed image isolates a desired marker. An interactive interface aids the user in deciding which threshold levels are appropriate.

2. Threshold every frame in the video, using the aforementioned levels and a blob-detector on the thresholded image to extract positions.

To increase the robustness of the blob detector, which finds roughly circular areas, some additional image processing is done on the thresholded image. First median filtering is done to close any holes in the binary image. After this, low pass filtering with a Gaussian kernel is performed to further smooth rough shapes.

3.3 Detection with a Convolutional Neural Network

The convolutional neural network (CNN) is trained to detect heels and toes (keypoints) from a set of around 1500 image patches from videos where the positions of the keypoints are annotated manually, using an annotation tool for this purpose. Using this tool, the user would click on the heel and toe each frame of the video. These pixel coordinates are then stored.

The ground truth is given as a sharp Gaussian around each keypoint to the network. The idea behind this is to give it some robustness against noise. The inspiration of outputting a Gaussian comes from the MOSSE filters [1]. The CNN that is employed has the following structure:

1. 64 3x3 filter kernels with stride 1 and padding 1. (3 input channels)
2. Rectified linear unit activation layer
3. 64 3x3 filter kernels with stride 1 and padding 1. (64 input channels)
4. Rectified linear unit activation layer
5. 1 3x3 filter kernel with stride 1 and padding 1. (64 input channels)

Input to the network is an RGB image (i.e. a region of interest from a frame of video) and the output is a single-channel image encoding where the maxima are expected to occur at the keypoints. Since the network contains only convolutional and activation layers, the network can in principle[2] operate on an input of any size, resulting in an output of the same size.

To save time during training, the two first convolutional layers use weights from the network E in [9] (also known as VGG19); only the last layer is initialized and trained from scratch.

3.4 Detection with OpenPose

The OpenPose library[3] is an implementation of the methods described in [2]. It provides a solution for full-body pose estimation by estimating heat maps of the positions of a number of keypoints on the body using a convolutional neural network. The heat maps are iteratively improved using *affinity fields*, also generated by a convolutional neural

[2]This is only true in principle, since large input sizes are demanding of hardware.

[3]`https://github.com/CMU-Perceptual-Computing-Lab/openpose`

network. Affinity fields describe the spatial relationship between keypoints, and so helps remove false detections that cannot produce a valid pose.

Note that this library does not provide estimations of the heel and toe but only gives a single keypoint for the ankle of each foot. Compared to the marker-version the ankle coordinate produced by OpenPose is rather noisy, but it still requires no special treatment. Next, this will be explained this in detail.

3.5 Tracking and filtering

The detection stage outputs foot detections from every frame in the video. The goal of the "Kalman and association" step is to group these positions into tracks for each keypoint, so that analysis can be done on the motion of a keypoint.

The problem of associating keypoints between frames is solved by comparing the similarity between a Kalman prediction and a detection. This measure of similarity uses distance between the keypoints, and in case of color tracking, the hue of the keypoints as well. If an associate is not found for a measurement in the next frame, the prediction is used in its place, predicting with a constant-velocity model. These predicted points are then used for association in future frames. If a track goes without matching with a real measurement for several frames, it is considered finished. The Kalman filter also acts as a smoother of the detections.

3.6 Estimation of Gait Parameters

In this section a description is given of how the different gait parameters are calculated.

3.6.1 Foot track association

When the tracks have been formed they are compared pairwise to determine if they belong to the same foot. This comparison is done by measuring the variance of the distance between the tracks in each frame, normalized by their average distance. This variance should be very low for tracks belonging to the same foot. All tracks will join groups of either 1 or 2 tracks. The two groups with the best score are selected as feet tracks.

For groups with two tracks a labeling of heel and toe is also done. This is done simply by checking which track is in front of the other according to the direction of movement, which is explained in section 3.6.2.

3.6.2 Estimation of Walking Direction

In order to differentiate between the left and right foot the system needs to know which direction the subject is walking in. This is done by fitting a first-order polynomial to the horizontal position of each keypoint track as a function of time and examining the sign of the slope.

If the slope of the polynomial is positive the direction is in the positive horizontal direction, and if the slope is negative the direction is in negative horizontal direction. Each track then gets to vote for which direction it thinks the movement is, weighted with its track score.

3.6.3 Estimation of left/right foot

Which foot is the left and which is the right is estimated by the use of the walking direction as well as which foot has the highest average of missed detections. If the subject is walking to the right in the video the most occluded foot will be the left. This is reasonable since the foot further away from the camera will be occluded by the other foot in some frames, while the closest foot never will.

3.6.4 Estimation of gait

The system estimates for each frame whether each foot is in contact with the ground. For this the horizontal position x as a function of frame number t is used. The detrended horizontal position

$$\overline{x}(t) = at + b$$

can be interpreted as an estimation of the position of the subjects body, where

$$a, b = \arg\min_{a,b} \sum_{t} \left(x(t) - at - b\right)^2 .$$

The foot is assumed to touch the ground when it has a lower velocity than the body in the forward-horizontal direction, i.e.

$$\text{footDown}(t) = \begin{cases} 1, & x'(t) < \overline{x}'(t), \\ 0, & \text{otherwise.} \end{cases}$$

The estimation of the derivative x' is done by convolving x with the derivative of a Gaussian.

All gait parameters are then estimated using footDown(t) in combination with information about the video's frame rate according to table 2.

4 Evaluation and results

The following section describes the evaluation of the different parts of the system the results it produced. The quantitative measurements used are described in table 3. It should be noted that the ground truth is generated by hand and that it is often not obvious which frame the feet hit or leave the ground; sub-frame precision is not present in the ground truth nor in the estimations. This also means some care should be taken if one would try to transform the measurements to seconds.

Table 3: Validation measurements.

		Detector	
Measurement	Details (Averages are per frame and foot)	Color	OpenPose
False detection	Average number of frames where system predicts a foot position despite there being no person in frame	0.0363	0.0588
Missed detection	Average number of frames where system does not predict a foot position despite there being a person in frame	0.0211	0.0299
Up/down misclassifications	Average number of frames where the system provides an incorrect foot position	0.0482	0.0624
Total error	Any of the errors above	0.1057	0.1510
Mean error per up/down transition:	Mean number of frames misestimation of when the foot leaves/hits the ground	2.23	4.95

4.1 Evaluation data

To evaluate the system, some videos were recorded using a smartphone camera. The videos were annotated with ground truth data using tools developed for this purpose. The tool can be used to annotate heel and toe positions, and whether the foot is touching the ground. Seven videos was used for evaluating with the OpenPose detector and five for the color detector.

This is the same data that is used for training the CNN. Which was not evaluated.

4.2 Detection with color markers

No direct quantitative evaluation metric is given here, since the performance of the tracker is highly dependent of the thresholds described in section 3.2. However, in tests the performance of the color tracker is never thought of as a limitation of the overall system performance. To get a hint of how well it performed one may look at *False alarm* and *Missed detections* in table 3, which are dependent on detector performance.

Problems arise if a marker has a color very similar to the background or other objects very close to the foot. This might make the detection linker mistake the background for a foot marker and the entire track is lost.

4.3 Detection with a Convolutional Neural Network

The output from the CNN when ran on a small part of the foot of a test subject is shown in Figure 2, with the maximum marked with a blue `x`.

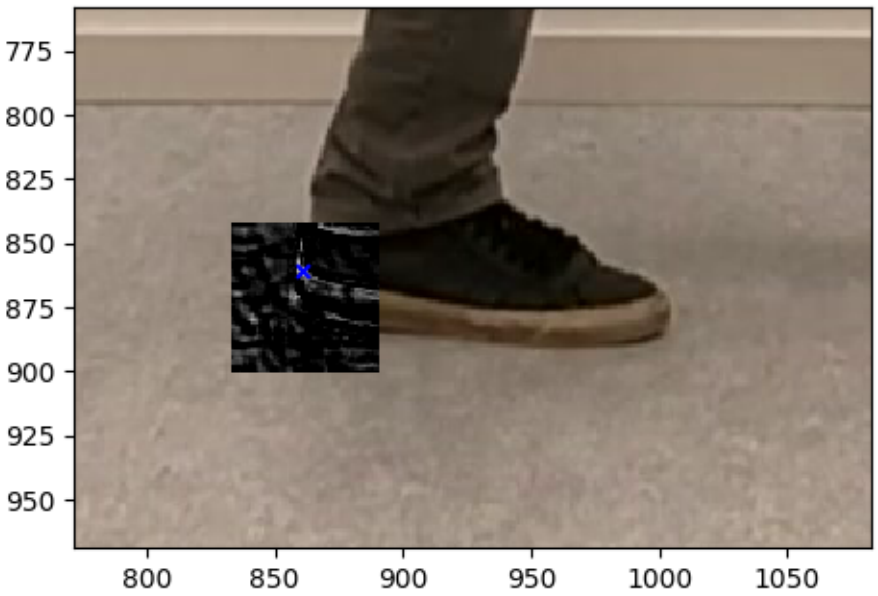

Figure 2: A foot with the output of the CNN overlayed over the heel.

We have successfully managed to train a CNN to output a maximum on the heel in a small image patch. However, this CNN has not yet been used to build a working detector,

mainly because of time constraints. While the sample in Figure 2 correctly locates the heel, the results were sporadic; the training and validation error were still decreasing in the training and this indicates that too little data is available to train on. While the method seems promising this project did not have the data and/or good enough network structure to make it a reliable detector.

4.4 Detection with OpenPose

Figure 3 shows the output of OpenPose on a sample frame from a markerless video. The keypoint estimations that OpenPose provided for the feet of the subject were of satisfying quality.

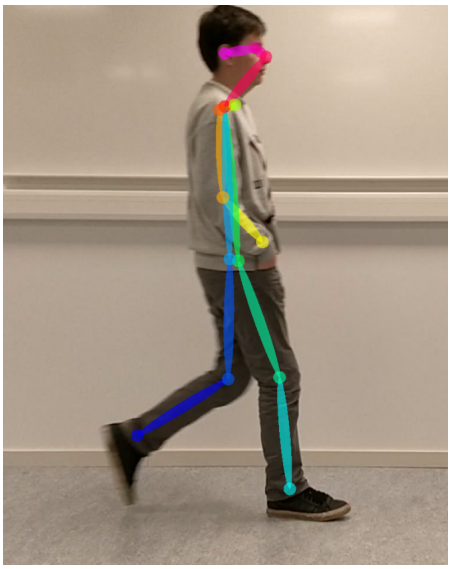

Figure 3: OpenPose used on an image in the test data

Evaluations of OpenPose performance can be seen in COCO's keypoint challenge[4]. The ankle position OpenPose computes is more noisy than that of for example marker tracking, but as a Kalman filter is smoothing the sequence this noise does not seem to affect the results notably. To get a hint of how well it performed one may look at *False alarm* and *Missed detections* in table 3, which are dependent on detector performance. While OpenPose in general gets worse scores than marker tracking this is likely because the current way of extracting up/down estimations is better suited for the toe or heel, rather than the ankle position.

[4] `http://cocodataset.org/`

4.5 Estimation of Gait Parameters

The performance of the gait parameter estimation was evaluated by evaluating the up-down estimation, since all parameters are dependent on it.

False alarm and *Missed detections* give some hits of the performance of the Kalman and associations step. *Average error* is estimating the performance of the gait estimation.

5 Discussion

Both OpenPose-tracking and, assuming the user has chosen markers with a distinct color and chosen good thresholds, marker-tracking, perform quite well; the only limitations are in the estimations for whether the foot is in the ground or not. The tracking of the foot itself is very rarely the limiting factor. However, more accurate estimates of the heel and toe positions would likely theoretically enable more accurate estimations of whether or not the foot is touching the ground.

One disadvantage of using OpenPose is that only the ankle position is obtained. In that way the video cannot be analyzed in standard heel-down and toe-up fashion. This problem would be solved if OpenPose would include the toe and heel in a new version or if another pose estimation library is. Since OpenPose is open source, one option could have been to modify the library as a part of the project. Instead, we chose to develop our own solution based on a CNN for this purpose.

Another disadvantage of OpenPose not having heel-down and toe-up data is that the angle of the foot touching the ground and leaving the ground cannot be measured. This measure can be of interest when doing gait analysis.

One perk with OpenPose is that it is markerless, so it is quicker to use and does not have requirements on the colors of the background.

Since the only input is a video the system has no knowledge of absolute distance. Hence it cannot estimate gait parameters which demand length measurements such as step length, walking speed and foot velocity. This might limit the systems usability compared to other solutions such as pressure mats and motion capture, where some sort of real world reference is present. This could be solved by including an object of known size in the scene or performing a camera calibration in some other way, but this is currently not supported.

6 Conclusion

A system for gait analysis was designed and implemented. It tracks feet and supports both videos with markers and without.

Table 3 shows the result we achieved, where the mean error per frame is around 0.1057 using the colortracker, and 0.1510 using OpenPose. This consolidated error measure is not as reveling of the systems functionality as the individual errors shown in table 3, so look at those for a clearer picture.

The accuracy is good enough to make a simple gait analysis. However, the system needs more tuning if it is to be used in applications where the really small differences are important. Nevertheless the project has verified that gait parameter estimation from a mono camera is possible.

7 Future outlooks

Vision based methods for gait analysis are getting more common and hence the data sets are growing [7]. Data sets containing gait data for different diseases are also getting bigger [7]. An interesting future application would be if one could do predictions based on abnormalities in the gait parameters. Such an application could be used for assisting medical diagnostics.

Gait analysis on animals is also useful. It would be interesting to see if the marker system preforms well on different animals and if we can implement a markerless system which work on animals as well.

Regarding tracking of keypoints, the approach of using a deep convolutional network could definitely provide better results than were achieved in this project. It is possible that these could be achieved with tweaking of hyperparameters, such as tweaking the learning function, sizes of batches, trying different loss functions etc. If these don't provide anything satisfying then maybe a different architecture should be used.

References

[1] David S Bolme et al. "Visual object tracking using adaptive correlation filters". In: (2010), pp. 2544–2550.

[2] Zhe Cao et al. "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: *CVPR*. 2017.

[3] Andrea Castelli et al. "A 2D Markerless Gait Analysis Methodology: Validation on Healthy Subjects". In: Volume 2015 (May 2015).

[4] James Gardiner et al. "Crowd-Sourced Amputee Gait Data: A Feasibility Study Using YouTube Videos of Unilateral Trans-Femoral Gait". In: *PLOS ONE* 11.10 (Oct. 2016), pp. 1–10. DOI: 10.1371/journal.pone.0165287. URL: https://doi.org/10.1371/journal.pone.0165287.

[5] Michela Goffredo, John N Carter, and Mark S Nixon. "2D Markerless Gait Analysis". In: *IFMBE Proceedings Vol. 22* (Nov. 2008).

[6] Mendez-Zorrilla A. Muro-de-la-Herran A Garcia-Zapirain B. "Gait Analysis Methods: An Overview of Wearable and Non-Wearable Systems, Highlighting Clinical Applications". In: *Sensors* (2014). ISSN: 3362-3394. DOI: 10.3390/s140203362.

[7] Chandra Prakash, Rajesh Kumar, and Namita Mittal. "Recent developments in human gait research: parameters, approaches, applications, machine learning techniques, datasets and challenges". In: *Artificial Intelligence Review* (Sept. 2016). ISSN: 1573-7462. DOI: 10.1007/s10462-016-9514-6. URL: https://doi.org/10.1007/s10462-016-9514-6.

[8] Jamal Saboune and François Charpillet. "Markerless Human Motion Capture for Gait Analysis". In: *CoRR* abs/cs/0510063 (2005). arXiv: cs/0510063. URL: http://arxiv.org/abs/cs/0510063.

[9] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). arXiv: 1409.1556. URL: http://arxiv.org/abs/1409.1556.