Gait Analysis Instructions

Contents

Record Video

Frontal

Place camera to capture a head-on view of the person walking away from or toward the camera (Fig. 1). Ensure that the entire body is visible in the recording (orienting the camera to portrait view may be helpful). Capture at least one stride (i.e., two heel-strikes) on each limb. The distance from the camera to the person must be known either at the start or end of the walkway. Camera placement and walkway distance is a trade-off between ensuring a long walkway with several strides and ensuring that the person remains large enough for pose estimation. We have used walkways of about 5 m with the camera positioned about 1.5 m behind the start; this is not a universally optimized approach, but the best trade-off given the confines of our laboratory.

Sagittal

Place camera to capture a perpendicular view of a person walking along a walkway; camera should be placed to the side of the *middle* of the walkway (Fig. 1). Ensure that the entire body is visible in the recording (orienting the camera to landscape view may be helpful). Capture at least one stride (i.e., two heel-strikes on each limb. A reference length in the plane of movement must be known (e.g., the length of strips of tape on the floor). Camera placement and walkway distance is a trade-off between ensuring a long walkway with several strides and ensuring that the person remains large enough for pose estimation. We have used walkways of about 5 m with the camera positioned about 3.5 m to the side of the middle of the walkway; this is not a universally optimized approach, but the best trade-off given the confines of our laboratory.

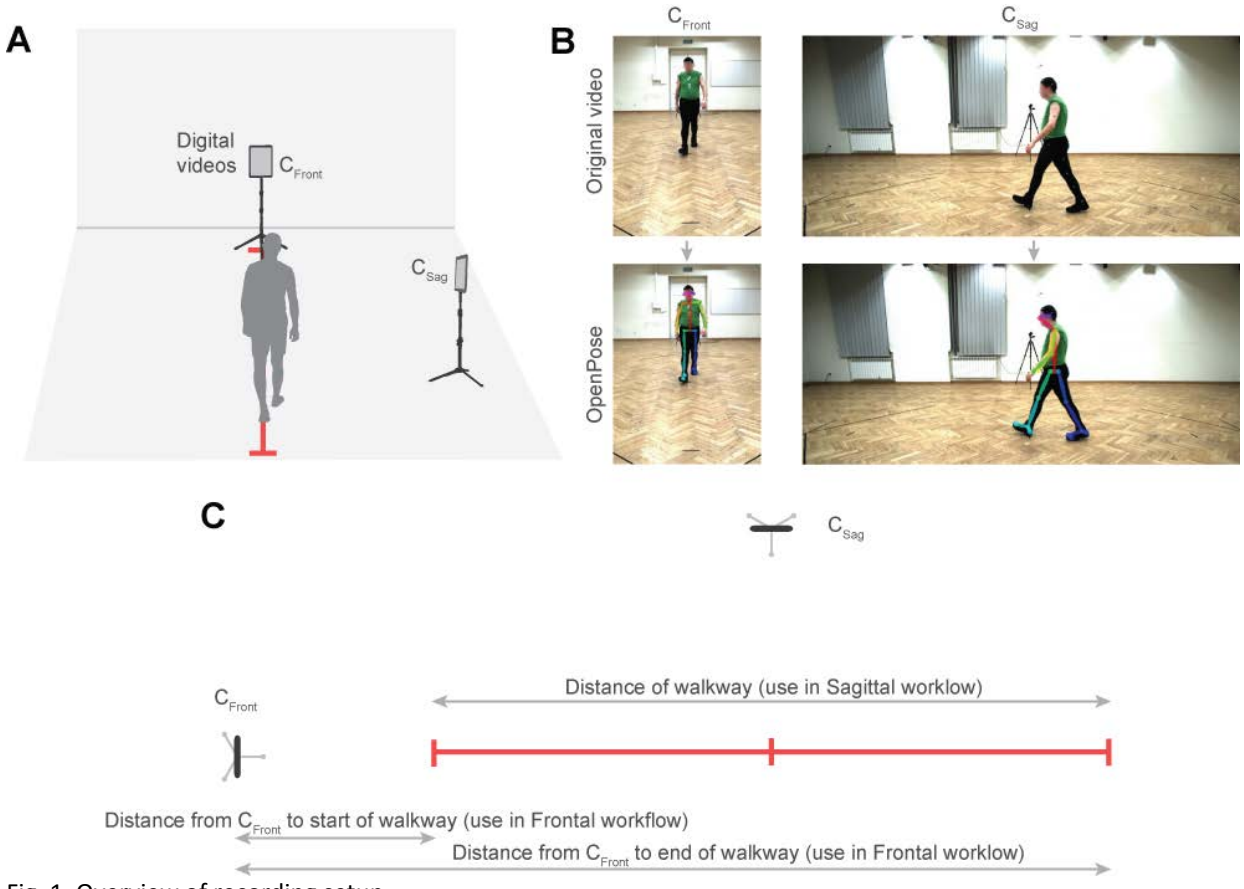

Fig. 1. Overview of recording setup.

Process with OpenPose

Use the demo of OpenPose to automatically track keypoints of visible persons (https://github.com/CMU-Perceptual-Computing-Lab/openpose). For faster processing times use a computer with a GPU. We used the BODY_25 model to track keypoints (see Appendix for overview of keypoints). Use flags to specify various parameters such as video orientation, set limit to only track one person or start and end frames (https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/include/openpose/flags.hpp). The outputs from OpenPose are JSON files for each frame analyzed and a video file in which keypoints are overlaid on the video.

Post-processing in MATLAB

Place the OpenPose outputs (JSON files and video file) in a folder. Ensure that the functions in each MATLAB workflow are in the same folder.

Frontal

1. Script: *OpenPoseGaitAnalysis_front.m*

Run *OpenPoseGaitAnalysis_front.m* in MATLAB; this script calls subsequent functions in succession.

2. Subfunction: *process_openpose_trackPerson_addition_front.m*

The function prompts for you to choose **all** JSON files and the OpenPose output video file (do not choose the original video file). If more than one person has been tracked by OpenPose over the entire video recording, additional functions are automatically called to ensure that only the person of interest is tracked (see Appendix). The function prompts for you to select if the person is walking away from or toward the camera.

3. Subfunction: *correctLegID_openpose_front.m*

This function lets you correct errors in left-right identification of the limbs (Fig. 2). A window opens that includes: various pushbuttons, a panel with time-series data of left and right ankle and shoulder keypoints and an image of the currently chosen frame. You can use the time-series data and the image to identify frames with errors in left-right identification. Use the Data tip or slider to select a specific frame (image updates to reflect chosen frame). Click 'Switch' pushbutton for instances where left-right identifications of limbs are switched. Click 'Delete left/right ID' pushbuttons to delete either left or right keypoints. Click restore pushbuttons to restore either left or right keypoints for one frame or for the entire time-series. Click 'Save' pushbutton when done.

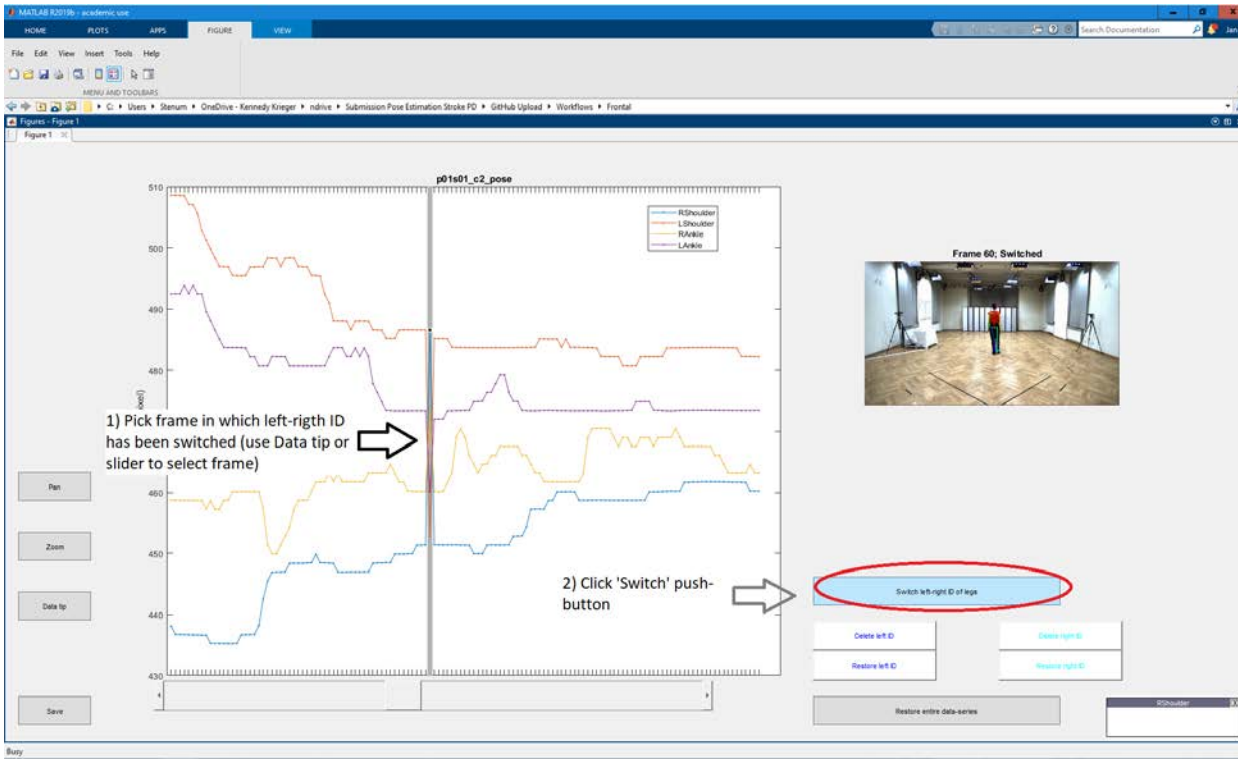

Fig. 2. Frontal workflow *correctLegID_openpose_front.m*.

4. Subfunction: *gapFill_openpose_front.m*

No user input needed. This function gap fills missing data using linear interpolation for gaps up to 0.12 s.

5. Subfunction: *findEvents_openpose_front.m*

This functions automatically finds left and right heel-strike events and lets the user manually inspect and correct as needed (Fig. 3). Events are found from local maxima/minima of the vertical distance between ankle keypoints. Positive peaks are left heel-strikes and negative peaks are right heel-strikes when person is walking away from camera; vice versa when person is walking toward camera. To delete spurious events: use brush to select events and click 'Delete Events'. To create missing events: use cursor to pick frame with missing event and click 'Create Left/Right Heel-Strike'. Click 'Save' when done.

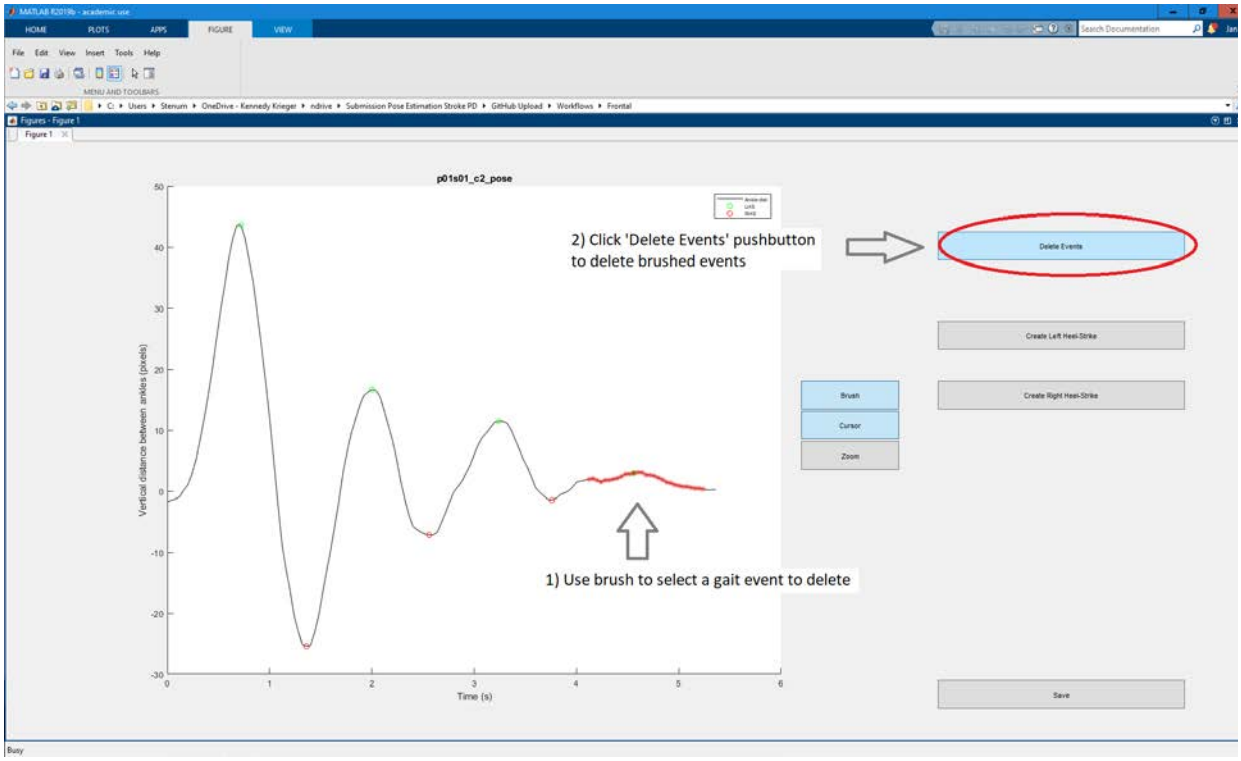

Fig. 3. Frontal workflow *findEvents_openpose_front.m*.

6. Subfunction: *calc_depthChange_front.m*

This function calculates a time-series of depth-change of the person relative to the reference position (Fig. 4). Find the frame in which the person is standing at the known reference depth (distance to camera; 2.33 m in 'Away' example video, 7.48 m in 'Toward' example video). Input the known reference distance measured in meter. Click 'Calculate Depth-Change and Save' when done.

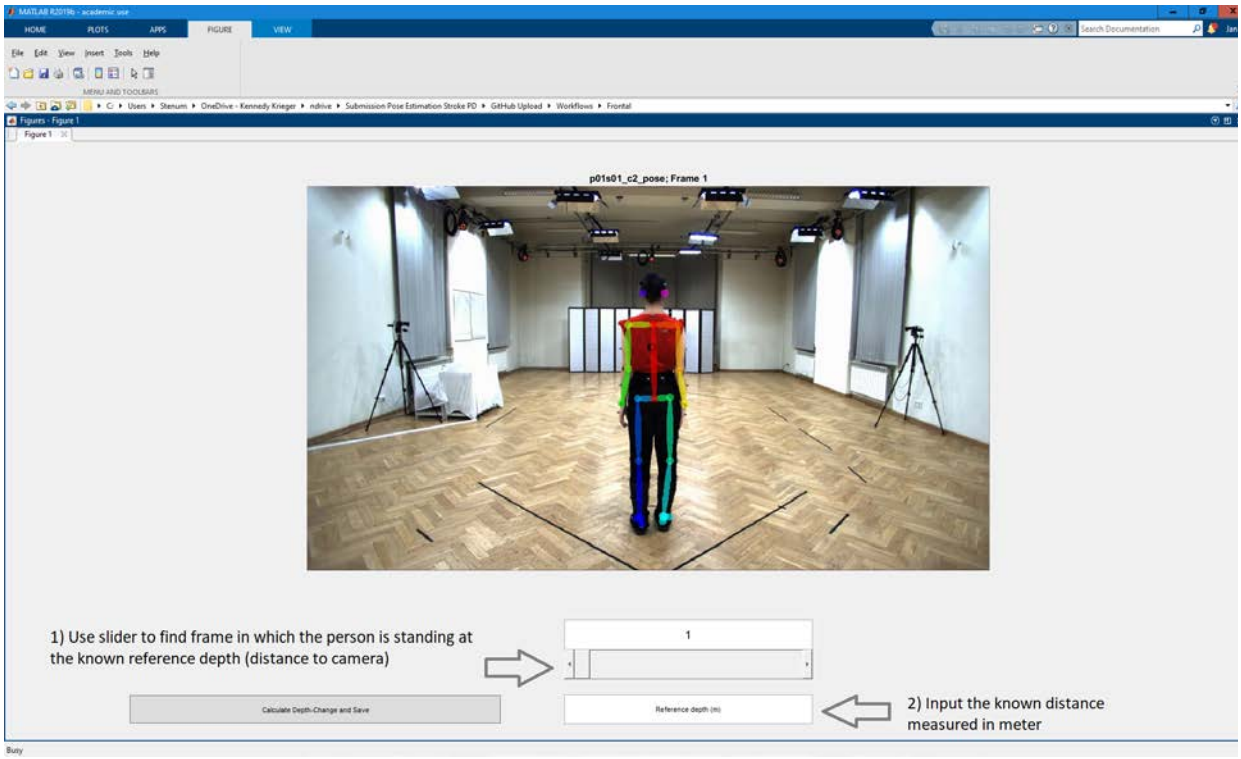

Fig. 4. Frontal workflow *calc_depthChange_front.m*.

7. Subfunction: *calculate_gaitParameters_front.m*

No user input needed. This function calculates spatiotemporal gait parameters: gait speed and left and right step times and step lengths.

8. Contents in saved MATLAB data file

The saved '.mat' file contains 6 structures: 'data_openpose', 'events_openpose', 'frameInfo', gaitParameters', 'openpose' and 'videoInfo'.

data_openpose

Contains the struct 'pose' which contains data matrices: 'data_raw' (raw OpenPose data: person id × frame × keypoint (see Appendix for indices) × coordinate (x,y)), the remaining matrices correspond to successive steps in workflow (frame × keypoint × coordinate). Contains time-series vector, depth-change time-series and the reference depth.

events_openpose

Contains frame numbers of left and right heel-strikes.

frameInfo

Contains metadata indication if multiple person are detected ('multiplePersonsDetected'), number of person detected ('numberPersonsDetected'), frames in which left-right IDs have been switched ('frames_switched'), frames in which left or right IDs have been deleted ('frames_leftClear' and 'frames_rightClear'), frames that have been gap filled ('isGapFilled').

gaitParameters

Contains left and right step times and step lengths for every step and gait speed.

Openpose
Contains information on OpenPose.

videoInfo
Contains MATLAB VideoReader object used to load video frames in the workflow.

Sagittal

1. Script: *OpenPoseGaitAnalysis_sag.m*
Run *OpenPoseGaitAnalysis_sag.m* in MATLAB; this script calls subsequent functions in succession.

2. Subfunction: *process_openpose_trackPerson_addition_sag.m*
The function prompts for you to choose **all** JSON files and the OpenPose output video file (do not choose the original video file). If more than one person has been tracked by OpenPose over the entire video recording, additional functions are automatically called to ensure that only the person of interest is tracked (see Appendix). The function prompts for you to select if the person's walking direction is left-to-right (→) or right-to-left (←).

3. Subfunction: *correctLegID_openpose_sag.m*
This function lets you correct errors in left-right identification of the lower limbs (Fig. 5). A window opens that includes: various pushbuttons, a panel with time-series data of left and right ankle keypoints and an image of the currently chosen frame. You can use the time-series data and the image to identify frames with errors in left-right identification. Use the Data tip or slider to select a specific frame (image updates to reflect chosen frame). Click 'Switch' pushbutton for instances where left-right identifications of limbs are switched. Click 'Delete left/right ID' pushbuttons to delete either left or right keypoints. Click restore pushbuttons to restore either left or right keypoints for one frame or for the entire time-series. Click 'Save' pushbutton when done.

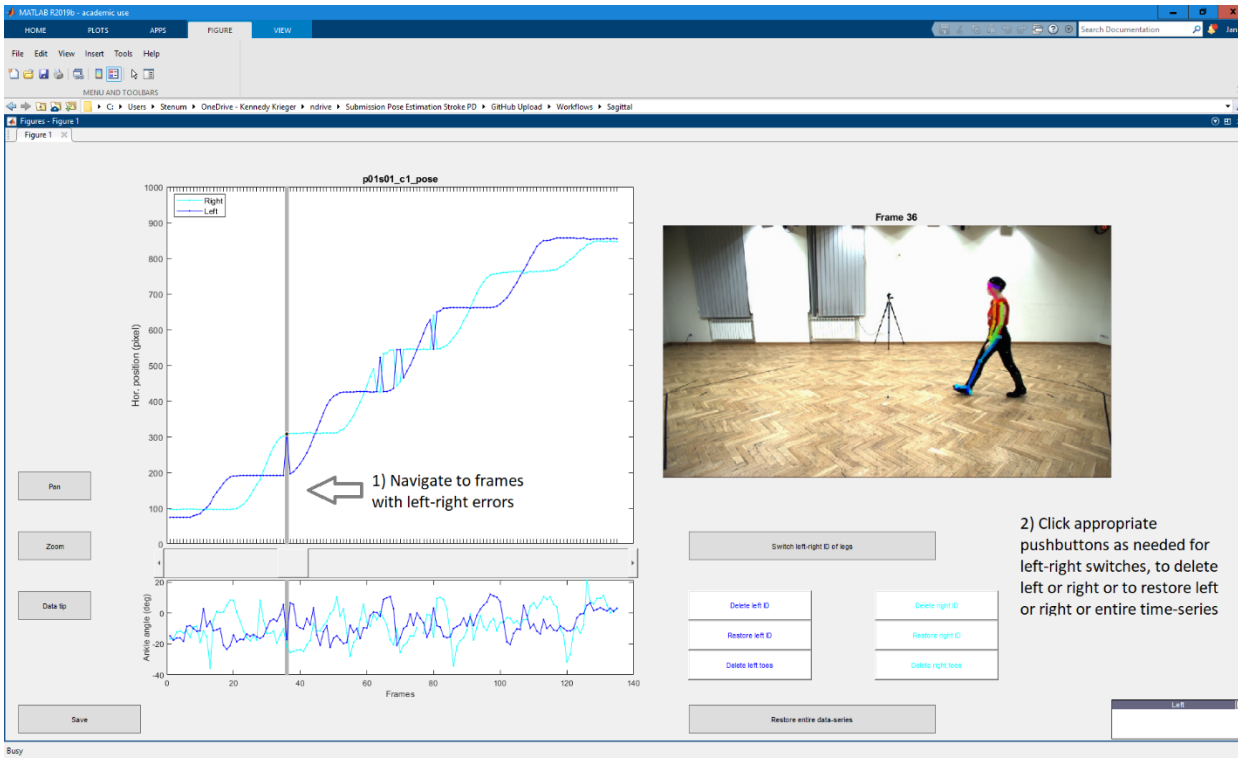

Fig. 5. Sagittal workflow *correctLegID_openpose_sag.m*.

4. Subfunction: *gapFill_filter_openpose_sag.m*

No user input needed. This function 1) gap fills missing data using linear interpolation for gaps up to 0.12 s, 2) low pass filters using 4th order Butterworth filter with cut-off frequency at 5 Hz.

5. Subfunction: *findEvents_openpose_sag.m*

This functions automatically finds left and right heel-strike and toe-off events and lets the user manually inspect and correct as needed (Fig. 6). Events are found from local maxima/minima of the excursions of left and right ankle keypoints relative to the mid-hip keypoint. Positive peaks are left/right heel-strikes and negative peaks are left/right toe-offs. To delete spurious events: use brush to select events and click 'Delete Events'. To create missing events: use cursor to pick frame with missing event and click 'Create Left/Right Heel-Strike'. To visually check if evens have been selected correctly, click 'Summary' which opens a new window with overlaid plots for left and right gait cycles aligned to either heel-strikes or toe-off events; the curves should follow a repeatable pattern for every gait cycle. Click 'Save' when done.

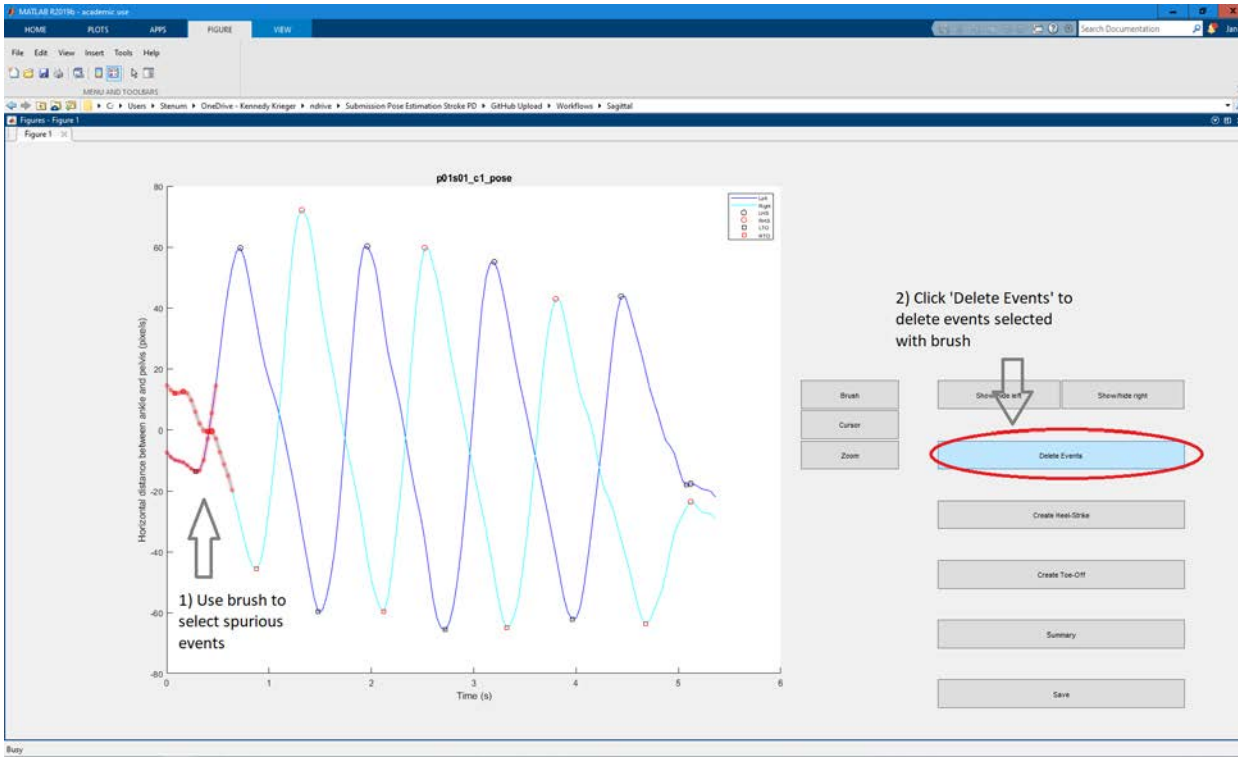

Fig. 6. Sagittal workflow *findEvents_openpose_sag.m*.

6. Subfunction: *extractScaling_openpose_sag.m*

This function opens a window to let you pick the start and end points of a known distance along the walkway in the plane of movement (Fig. 7). Use the cursor to select 2 pixels at the start and end (place one data tip first, then place the second data tip by either holding shift and clicking on the desired location or right-click and select 'Create New Data Tip', then place the second data tip). Input the known distance (6.3 m in the example video). Click the 'Calculate Scaling' pushbutton. Click 'Save' when done.

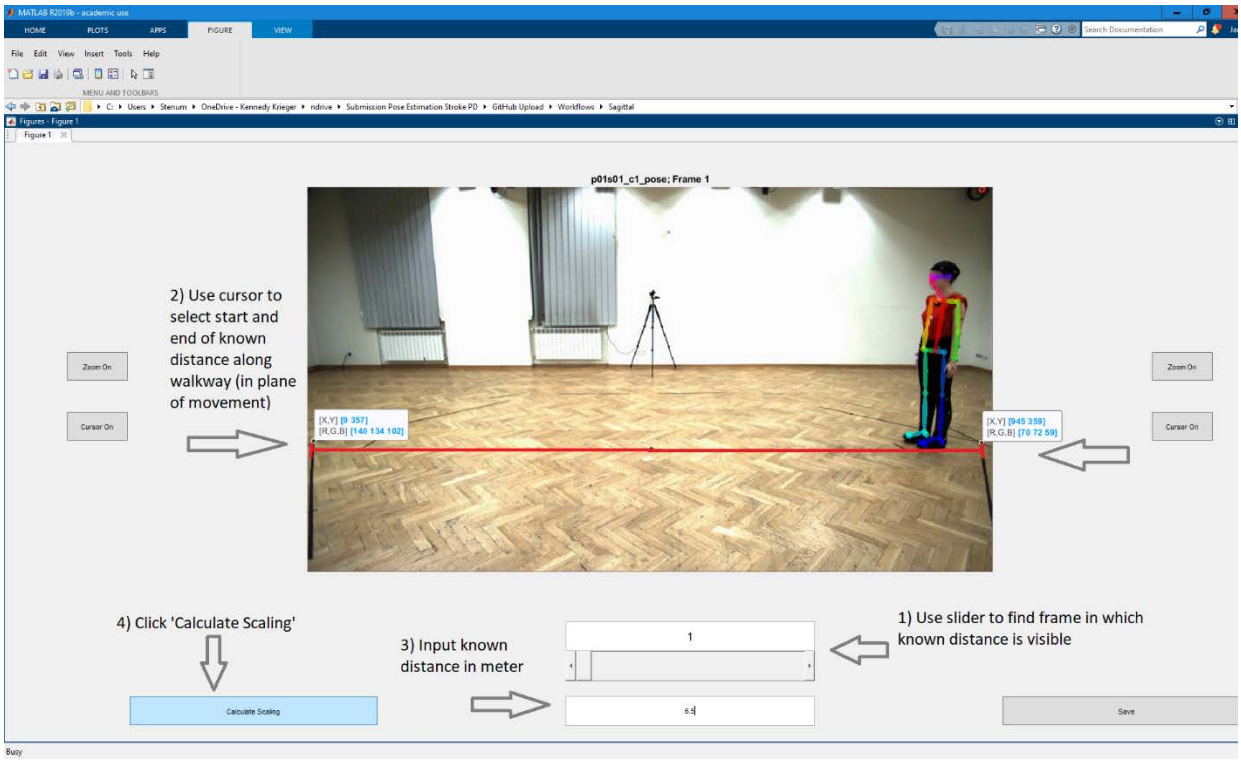

Fig. 7. Sagittal workflow *extractScaling_openpose_sag.m*.

7. Subfunction: *calculate_gaitParameters_jointAngles_sag.m*

No user input needed. This function calculates spatiotemporal gait parameters (gait speed and left and right step times, stance times, swing times, double support times and step lengths) and joint angles (sagittal plane lower limb kinematics at the hip, knee and ankle).

8. Contents in saved MATLAB data file

The saved '.mat' file contains 6 structures: 'data_openpose', 'events_openpose', 'frameInfo', gaitParameters', 'openpose' and 'videoInfo'.

data_openpose

Contains the struct 'pose' which contains data matrices: 'data_raw' (raw OpenPose data: person id × frame × keypoint (see Appendix for indices) × coordinate (x,y)), the remaining matrices correspond to successive steps in workflow (frame × keypoint × coordinate). Contains time-series vector and scaling factor.

events_openpose

Contains frame numbers of left and right heel-strikes and toe-offs.

frameInfo

Contains metadata indication if multiple person are detected ('multiplePersonsDetected'), number of person detected ('numberPersonsDetected'), frames in which left-right IDs have been switched ('frames_switched'), frames in which left or right IDs have been deleted ('frames_leftClear' and 'frames_rightClear'), frames that have been gap filled ('isGapFilled').

gaitParameters

Contains left and right step times, stance time, swing time, double support times and step lengths for every step and gait speed.

jointAngles
Contains 2D sagittal plane lower limb kinematics for left and right hip, knee and ankle.

Openpose
Contains information on OpenPose.

videoInfo
Contains MATLAB VideoReader object used to load video frames in the workflow.

Appendix

OpenPose Keypoints

OpenPose keypoints in the BODY_25 model is given in the table and Fig. A1.

OpenPose Keypoints in MATLAB

Index	Keypoint
1	Nose
2	Neck
3	Right should
4	Right elbow
5	Right wrist
6	Left shoulder
7	Left elbow
8	Left wrist
9	Mid hip
10	Right hip
11	Right knee
12	Right ankle
13	Left hip
14	Left knee
15	Left ankle
16	Right eye
17	Left eye
18	Right ear
19	Left ear
20	Left big toe
21	Left small toe
22	Left heel
23	Right big toe
24	Right small toe
25	Right heel

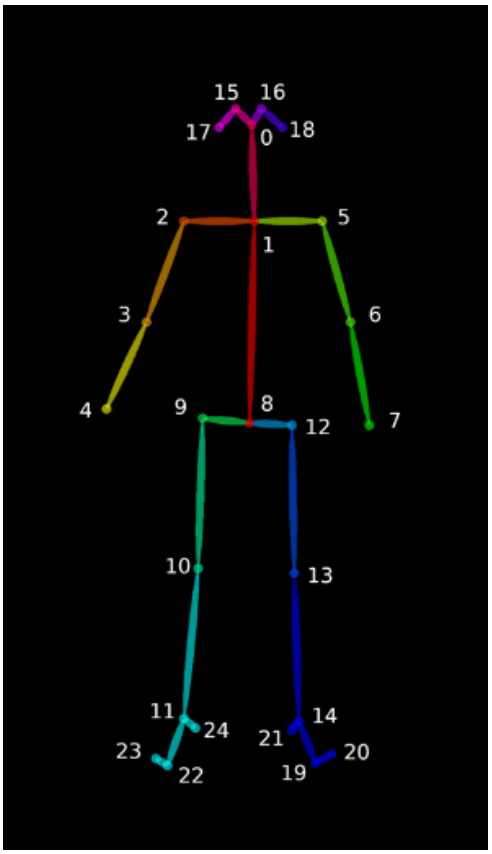

Fig. A1. (https:// github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/output.md)

Person Tracking

If OpenPose has tracked multiple persons, additional subfunctions are called: 1) *trackPerson_openpose_manual_input.m*, 2) *trackPerson_openpose_automatic_tracking.m* and 3) *trackPerson_openpose_visual_inspection.m*.

1. Subfunction: *trackPerson_openpose_manual_input.m*

A window opens that allows the user to select the person of interest to track throughout the entire video (Fig. A2). Use cursor to click on keypoints of the person of interest to select this person. Click 'Set Anchor Point' when the person is selected. Use slider to find frames in which the person of interest is visible. You can use the pushbuttons 'Set start of tracking: frame x' and 'Set end of tracking: frame x' to limit the frames in which to track the person (e.g., if the person is not visible at the start or end of the video). Click 'Save and Continue' when done.

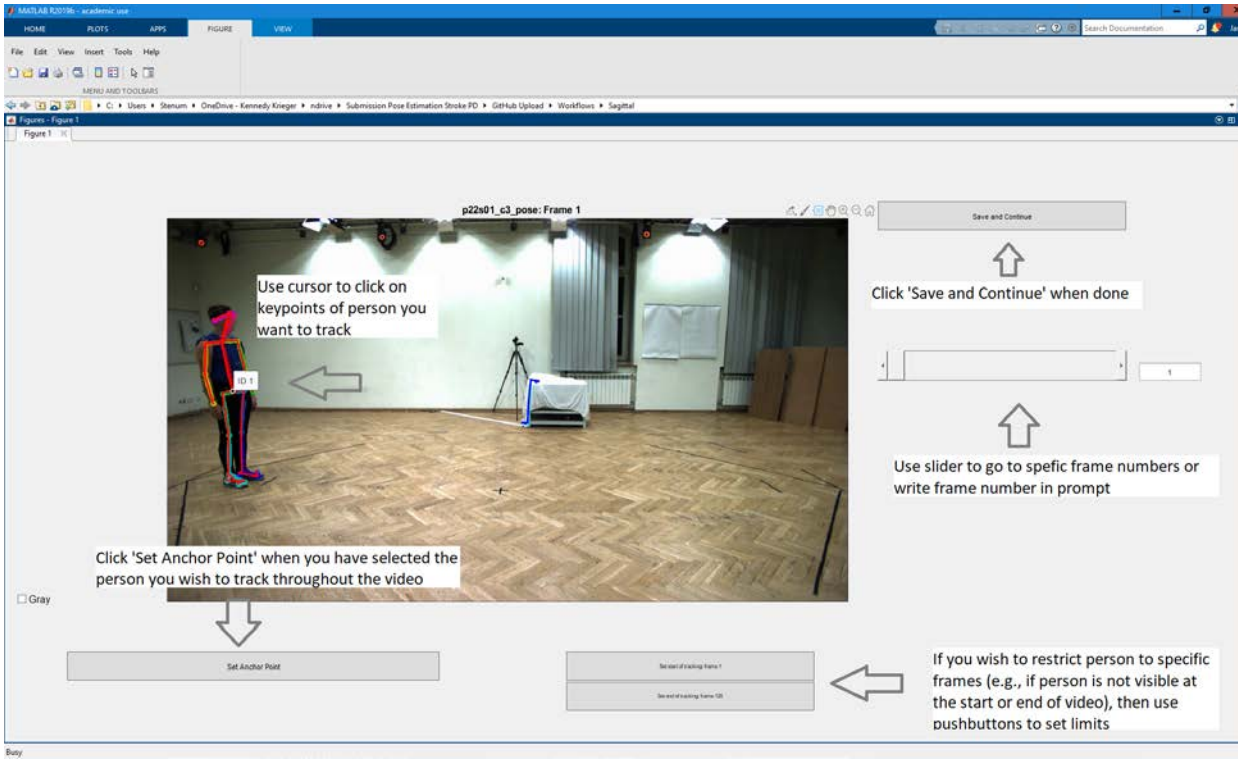

Fig. A2. Additional workflow if OpenPose detects multiple persons *trackPerson_openpose_manual_input.m*.

2. Subfunction: *trackPerson_openpose_automatic_tracking.m*

This subfunctions tracks the person throughout the video based on minimal frame-to-frame distance between keypoints from all the persons tracked.

3. Subfunction: trackPerson_openpose_visual_inspection.m

A window opens that allows the user to visually inspect and correct errors in person tracking (Fig. A3). Use time-series plots and image to inspect person tracking. The left upper time-series plot shows the horizontal coordinate of chosen keypoints (select from list at upper right corner); look to large jumps in the time-series to indicate possible errors in person tracking. The left lower time-series plot shows the mean frame-to-frame pixel distance of all keypoints; small values indicate that location change of keypoints was minimal; spikes may indicate instances with errors in person detection. Select specific frames by clicking the time-series plots with the cursor tool, use slider or type into text box: image updates to reflect chosen frame. To correct a person ID of one frame: use cursor to click person of interest in image (currently tracked person is shown with red keypoints and a bounding box; persons not tracked are shown with white keypoints) and click 'Correct: ID x to y'. To revert the ID of current frame

or all frames click ‘Revert ID of current frame’ or ‘Revert all IDs’, respectively. To delete ID click ‘Delete ID’ which will delete IDs over the range specified in boxes above pushbutton. You can auto track either forward or backward from a chosen frame by clicking ‘Auto Track’ pushbutton: person IDs will be updated based on minimizing frame-by-frame distance between keypoints from the person ID of the current frame. You can set the limits of the auto track in the boxes about the pushbutton. Click ‘Save and Continue’ when done. The remaining frontal or sagittal workflow will resume.

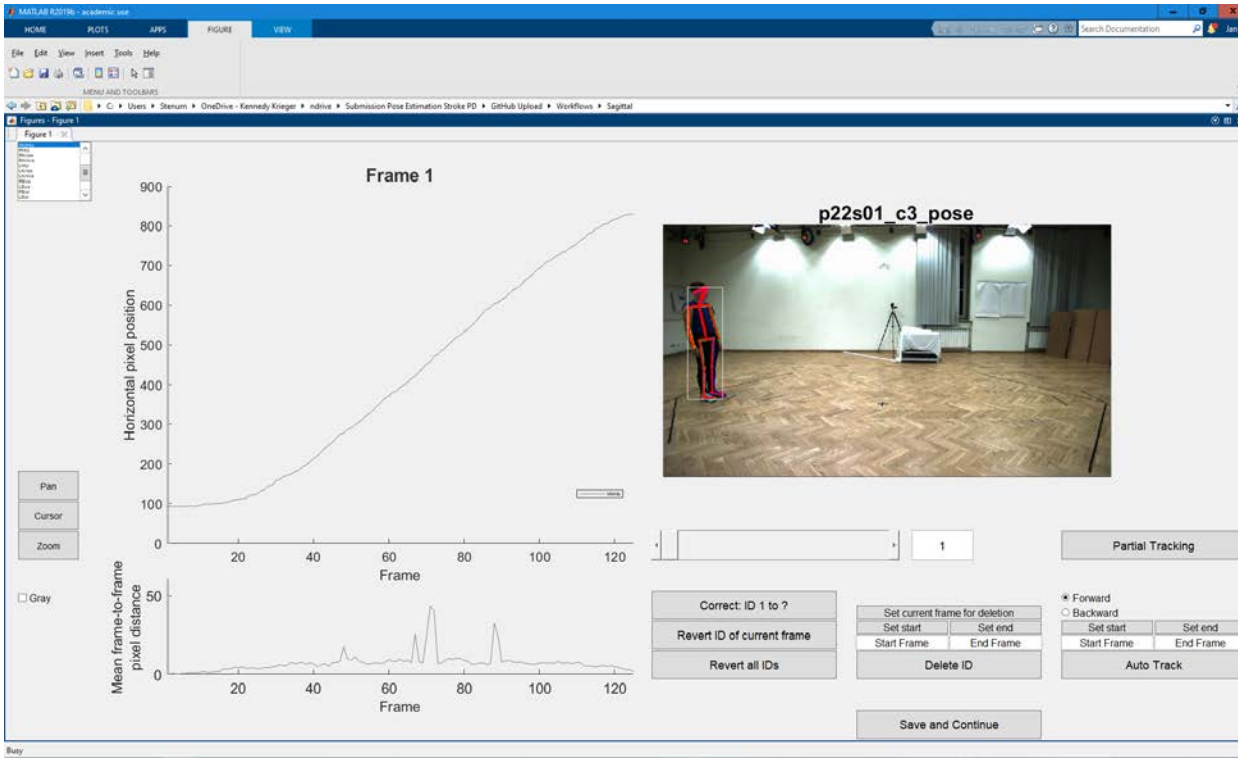

Fig. A3. Additional workflow of OpenPose detects multiple persons *trackPerson_openpose_visual_inspection.m*.