Smart Walker Design for Clinical Rehabilitation

Ben Naets*, Evelien De wachter*, Rik Lambrechts*, Joris Vaneyghen†,
Maaike Fobelets†‡, Catherine Middag†, Ruben Debeuf§¶, Eva Swinnen§¶ and Ben Minnaert*
*Odisee University College of Applied Sciences, Ghent, Belgium
†Dep. Health Care, Brussels Expertise Centre for Healthcare Innovation,
Erasmus Brussels University of Applied Sciences and Arts, Brussels, Belgium
‡Faculty of Medicine and Pharmacy, Dep. of Public Health,
Biostatistics and Medical Informatics Research group, Vrije Universiteit Brussel , Brussels, Belgium
§Rehabilitation Research Group, Dep. of Physiotherapy, Human Physiology and Anatomy, Vrije Universiteit Brussel, Belgium
¶Brussels Human Robotics Research Cetenter (BRUBOTICS), Vrije Universiteit Brussel, Belgium

***Abstract*—Walkers increase the ability to walk independently and safely but can - when used incorrectly - lead to dangerous situations and injuries. Therefore, it is important that during the rehabilitation injured and/or elderly people learn how to walk with a walker. Currently, patients are dependent of their healthcare professionals to assist them and correct their gait pattern if needed, which limits them in when and where they can train walking. In view of this problem, a smart walker was designed which assists the patient by monitoring feet placement and alerting when incorrect usage is detected. In this way, the patient can train independently. This work describes the design of the smart walker, consisting out of a depth camera, RGB cameras and ultrasonic sensors. These sensors serve as input for a computer model which returns audiovisual feedback to the user.**

***Index Terms*—rehabilitation, walker, AI, single board computer, image processing**

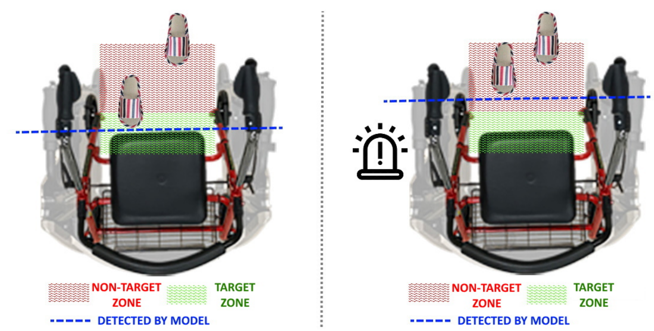

Fig. 1. Foot placement - Walking too far behind the walker. Recommended usage (left) and incorrect usage (right) where an audiovisual alert is given.

I. Introduction

As healthcare has improved in recent years, more people are surviving after injury or disease. This also results into more patients who need to rehabilitate. However, the number of healthcare professionals has not increased proportionally [1], although personal contact between physiotherapists and their patients is necessary to facilitate healing during clinical rehabilitation. Smart devices and the application of artificial intelligence (AI) may enable patients to perform the necessary rehabilitation exercises outside therapy time, i.e. without the healthcare professional.

In this context, a smart walker for use in clinical rehabilitation was designed. During rehabilitation from certain diseases or from surgery, walkers serve as an important walking aid [2]–[5]. The goal is to develop a device which is tailored to the needs of the healthcare professionals and patients, enabling the patient to train independently while important parameters - which are indicated by the therapist - are monitored. It was shown [6], [7] that adding smart technologies to a walker can not only improve the quality of life for the patient, but also prevent harmful falls. For example, sensors can detect leg positions and gait events [8], [9] and a strain gauge-based instrumentation system can address the changes in upper extremity kinetics that occur with the use of a walker [10]. Also robotic walkers are tested with standing and walking assistance function [11].

One of the main indicators of recommended usage of a walker is the foot placement of the user with respect to the walker, which serves as the main detection objective of the device. The walker can be used by each patient without the need of a healthcare professional to apply on-body sensors on the feet or legs. An important parameter is the cost of the total system, which will be optimized by the use of low-cost smart devices and developing custom AI models.

This paper describes the first version of the smart walker - which will be used for data-acquisition, AI-model development and pilot tests with a test group of patients - including the detection features, sensors, feedback options and initial vision models. The data and user experiences gathered by this initial version will be used in the following iterations to determine the final design of the walker, optimizing sensors and AI-models for low-cost smart devices, reducing cost and improving usability.

This work is supported by Flanders Innovation & Entrepreneurship VLAIO (TETRA-project HBC.2020.2533 “SAIRE: Smart Devices and Artificial Intelligence for Rehabilitation”).
Corresponding author: B.N., email: ben.naets@odisee.be

978-1-6654-2209-3/22/$31.00 ©2022 IEEE

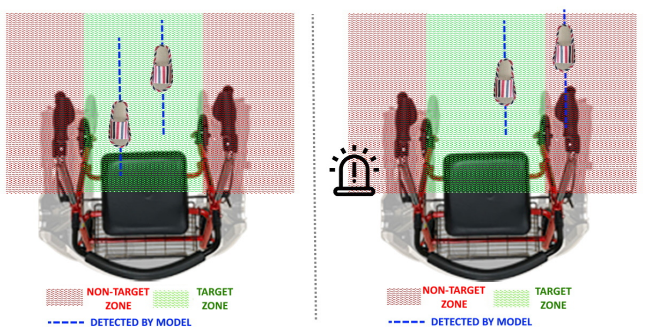

Fig. 2. Foot placement - Walking out of center. Recommended usage (left) and incorrect usage (right) where an audiovisual alert is given.

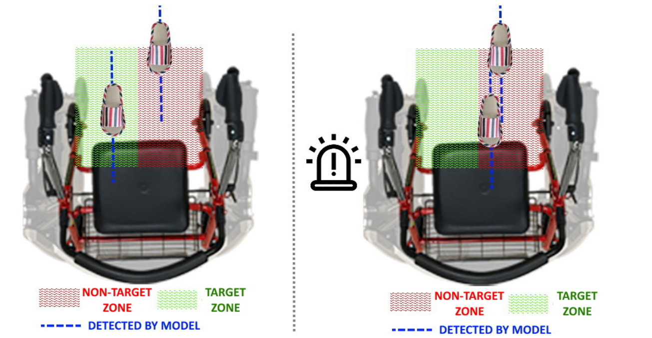

Fig. 3. Foot placement - Catwalk. Recommended usage (left) and incorrect usage (right) where an audiovisual alert is given.

II. Smart features objectives

The goal of our prototype is enabling patients to train independently; e.g., the smart walker can teach patients to train a recommended gait pattern, based on the feedback of the device. In order to determine which parameters to prioritize, a panel of professionals in rehabilitation was consulted and a set of features was determined for the smart walker. The following sections give an overview of parameters which our developed smart walker can detect.

A. Foot placement - Walking too far behind the walker

The center of gravity should be between the wheels and support legs of the walker. If the patient walks behind the walker, the center of gravity will shift behind the back support legs and the risk of the walker sliding away is increased, resulting in a possible fall of the patient.

Two zones relative to the back support legs are created: a target zone, and a non-target zone (Figure 1). When walking, the patient should always place the moving foot in the target zone. If this is not the case, the patient is walking behind the walker and will be notified by an audiovisual signal.

B. Foot placement - Walking out of center

A common error by patients when taking turns is walking besides the walker. This results in uneven forces on the walker handles. As a result, the walker can rotate or slip away, again resulting in a possible fall of the patient.

Three zones relative to the center of the walker are created: a center target zone, and a non-target zone at each outer side of the walker (Figure 2). The feet of the patient should be in the target zone at all times. If not, the smart walker will detect this and notify the patient.

C. Foot placement - Crossing feet a.k.a. catwalk

When the user crosses his/her feet when walking - such as on a catwalk - the joints of the user are submitted to extra forces, possibly leading to new injuries. Furthermore this results in bad stability due to the feet being on one line instead of at shoulder width apart.

Two zones relative to the center of the walker are created: a left zone and a right zone (Figure 3). The left/right foot should always be in the corresponding zone. If not, the patient will be notified.

D. Other useful detections

The panel of professionals in clinical rehabilitation listed other useful detections, i.e.:

- The detection of step frequency, step length and foot roll.
- Fall detection.
- Left/right pressure distribution on the handles.

These detections were not considered in the first prototype of our smart walker.

III. System setup

The current version of the smart walker is a first prototype version. It will be used for practice tests at rehabilitation centers with patients and healthcare professionals. From the feedback of these actors, further iteration models will be developed. This section describes the hardware setup of the current model. Figure 4 gives a schematic overview of the components of the system and their connections.

The central core of the smart walker is a single board computer (type: Jetson Nano) for the data-acquisition and processing. A touchmonitor (Waveshare 13.3 inch HDMI LCD Display) is used as User Interface and provides audiovisual feedback to the user. Power is delivered by a Litionite Tanker 90 W / 50000 mAh Powerbank, which allows for multi hour testing. A DC-DC step down voltage regulator (12 V - 5 V) is used as voltage regulator.

Two ultrasonic sensors HC-SR04 are applied to estimate distance. Two Neopix LED sticks provides visual feedback to the patient. The ultrasonic sensors and LED sticks are controlled by an Arduino Uno microcontroller.

An (expensive) depth camera D435i and two (cheap) Raspberry Pi Camera V2 are applied as input devices to monitor the foot placements. The purpose of the depth camera is to acquire high-quality data to train AI-models which then can run on the cheaper camera module. Further experimental testing will determine whether or not the depth camera is necessary in practice environments.

The cameras can be mounted at two different positions on the walker:

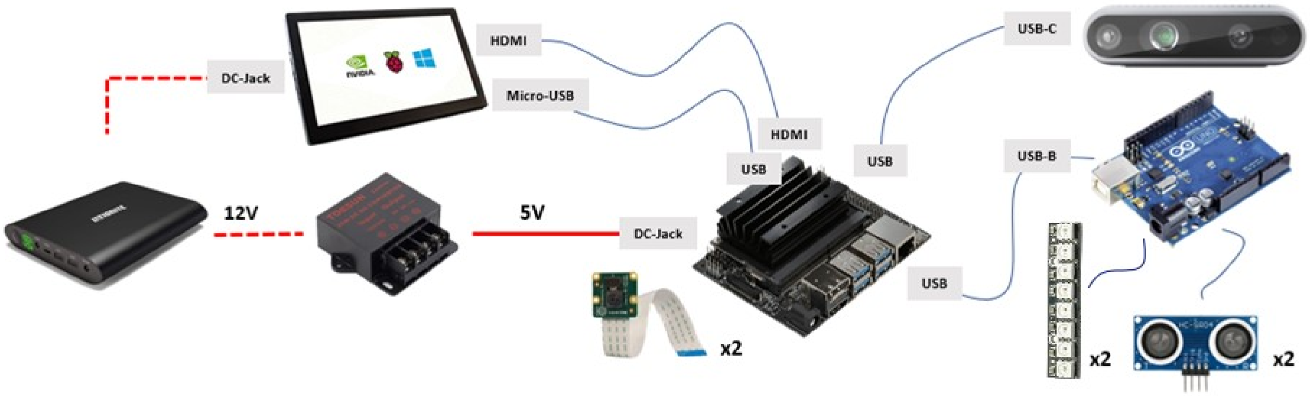

Fig. 4. Smart walker: system topology and connections of the hardware components.

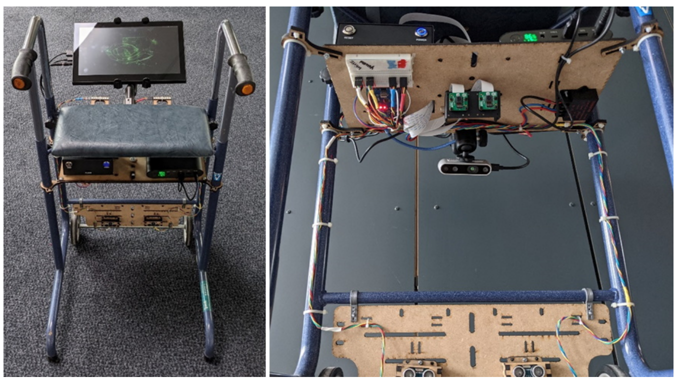

Fig. 5. Bird's-eye view configuration.

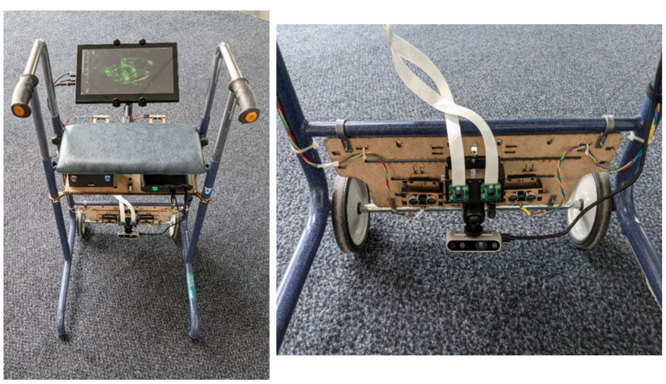

Fig. 6. Worm's-eye view configuration.

- *Bird's-eye configuration*: a top-down view on the feet of the patient and the back legs of the walker (Figure 5).
- *Worm's-eye configuration*: a bottom-up view on the feet and legs of the patient (Figure 6).

The user interface (UI) ties the different program modes together and enables easy data-acquisition and testing of the software. Table I gives an overview of the UI control options.

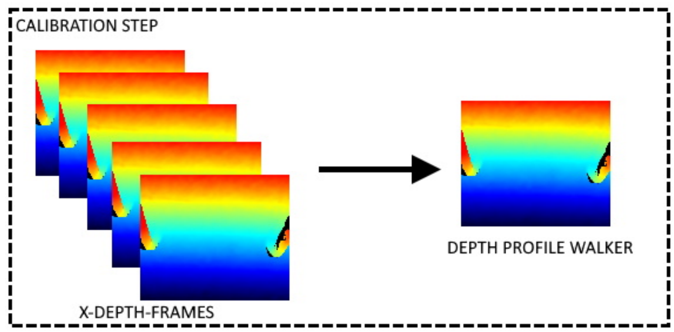

Fig. 7. Calibration pipeline of the walker depth profile.

IV. Detection methods

One of the main challenges of the smart walker is the development of a model which can detect the position of the feet relative to the walker and -moreover- is performant enough to run on the limited resources of a single board computer. We now describe three different possible approaches: depth calibration, a Pix2Pix model and a Tiny YoloV4 model.

A. Foot detection using depth calibration

This method uses the depth image from the D435i camera and consists out of 2 main steps:

1) *Calibration of the walker depth profile*: At the start of the program, a number of frames will be taken to create a depth profile of the walker (Figure 7). At this stage no feet or other objects may be present in the field of view of the depth camera.
2) *Foot detection*: After calibration, each depth frame is subtracted with the depth profile of the walker. The resulting frame represents the objects not present during calibration (Figure 8). Two threshold functions are applied to this frame: one removes the pixels which only differ by a couple of millimeter in order to reduce noise; the other one removes the pixels which are further away than the region of interest. A contour detection -

TABLE I
USER INTERFACE CONTROL OPTIONS

Depth camera D435i: *Activate*: Activate the camera *Display stream*: Display the RGB camera stream *Display depth*: Display the depth camera stream *Analyse*: Run the detection model on the D435i data and augment the RGB camera stream *Display FPS*: Display the frames-per-second on the camera streams *Display line*: Draw a fixed line on the camera streams which can be used as a target for the patient to step over *Archive*: Save the D435i streams in a bag-file	**Commands:** *Save*: Save the current configuration *Save & Launch*: Save the current configuration and launch the application *Update*: Update the settings while the application is running *Stop*: Stop the application without closing the UI *Quit*: Stop the application and close the UI *Upload*: Upload the recorded files to the cloud *Clean*: Remove uploaded files *Move*: Move recorded files to an external USB drive
Ultrasonic sensors: *Activate*: Enable the ultrasonic sensors *Display*: Display the ultrasonic measurements in a separate UI *Archive*: Save the measurements in a csv-file	**Raspberry Pi cameras:** *Activate*: Activate the Raspberry Pi cameras *Archive*: Save the Raspberry Pi camera streams in mp4-files
Metronome: *Activate*: Enable the metronome with the desired beats per minute	**Audio-Feedback:** *Activate*: Enable audio feedback
LED-Feedback: *Activate*: Enable visual feedback with Neopix	

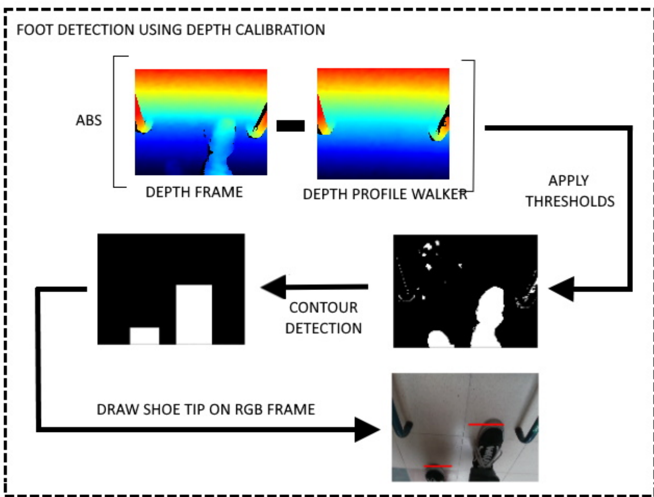

Fig. 8. Foot detection pipeline using the depth camera.

excluding contours which are too small to be a foot - extracts the feet from this image and the bounding box. From the resulting bounding box, the tip of the shoe or foot is determined and compared with the target zone.

B. Foot detection using Pix2Pix model

It is possible to extract the feet with only the RGB camera. The first approach is by using a Pix2Pix GAN model [12] which returns a feet-mask of the input image. The training dataset is automatically generated by the foot detection model using depth calibration. This method enables fast and easy (re-)generating of datasets (Figure 9).

The output of the model can be used to detect the contours and their bounding boxes in order to find the position of the feet. The resulting model works good on a standard desktop but is - with its size of 200 MB - not suitable for a single board computer such as the Jetson Nano (Figure 10).

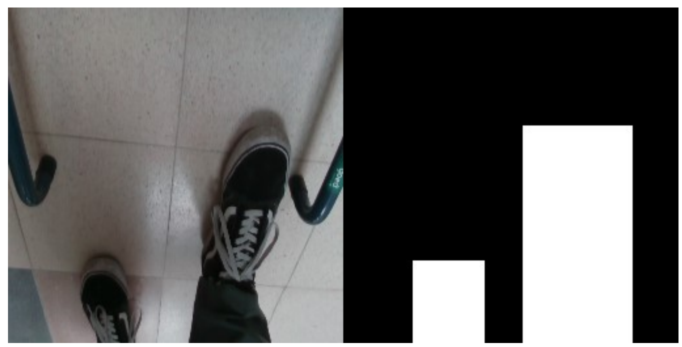

Fig. 9. Training image using a Pix2Pix GAN model, generated from depth calibration model.

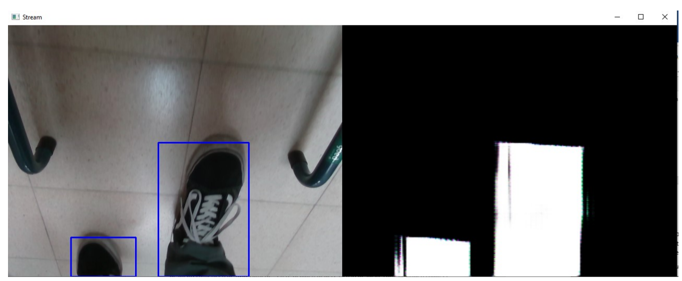

Fig. 10. Output feet detection image from the Pix2Pix model with post processing.

C. Foot detection using Tiny YoloV4 model

Given the limited capabilities of single board computers, a more suitable model for finding the position of the feet is an object detection and positioning model such as the YOLO network [13]. This network bypasses the feet-mask step and directly gives the position and bounding box of each detected foot (Figure 11). The training dataset can be autogenerated from the foot detection model using the Pix2Pix model or/and depth calibration.

A trained Tiny YoloV4 model converted to tensorflow-lite

Fig. 11. Output feet detection image from the YoloV4 Tiny model.

has a size of 20 MB. Further improvements can be made by converting the data to a tensorRT model optimized for running on e.g., a Jetson Nano. For comparison, the benchmark for a Tiny YoloV3 model on a Jetson Nano with tensorRT is 25 fps [14], which is sufficient for the smart walker.

V. CONCLUSION

A first prototype version of a smart walker for clinical rehabilitation was developed. The walker detects foot placement, i.e. walking too far behind the walker, out of center, or crossing feet. The different detection models show that there are multiple options to detect the position of the feet with the equipped sensors on the smart walker. By applying further model optimization - choice of model and building for specific hardware - these models can run on the constraint hardware of the walker. The *foot detection using depth calibration* shows how depth data can be used to automatically generate datasets for vision models which only use the RGB image, possibly making the depth camera unnecessary in further iterations of the walker and thus reducing cost and the number of components.

REFERENCES

[1] W. Sermeus, L. H. Aiken, K. Van den Heede, A. M. Rafferty, P. Griffiths, M. T. Moreno-Casbas, R. Busse, R. Lindqvist, A. P. Scott, L. Bruyneel *et al.*, "Nurse forecasting in europe (rn4cast): Rationale, design and methodology," *BMC nursing*, vol. 10, no. 1, pp. 1–9, 2011.

[2] S. M. Bradley and C. R. Hernandez, "Geriatric assistive devices," *American family physician*, vol. 84, no. 4, pp. 405–411, 2011.

[3] M. M. Martins, C. P. Santos, A. Frizera-Neto, and R. Ceres, "Assistive mobility devices focusing on smart walkers: Classification and review," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 548–562, 2012.

[4] M. Martins, C. Santos, A. Frizera, and R. Ceres, "A review of the functionalities of smart walkers," *Medical engineering & physics*, vol. 37, no. 10, pp. 917–928, 2015.

[5] C. Nave and O. Postolache, "Smart walker based iot physical rehabilitation system," in *2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI)*. IEEE, 2018, pp. 1–6.

[6] O. Postolache, J. M. D. Pereira, V. Viegas, L. Pedro, P. S. Girão, R. Oliveira, and G. Postolache, "Smart walker solutions for physical rehabilitation," *IEEE Instrumentation & Measurement Magazine*, vol. 18, no. 5, pp. 21–30, 2015.

[7] T. Kikuchi, T. Tanaka, S. Tanida, K. Kobayashi, and K. Mitobe, "Basic study on gait rehabilitation system with intelligently controllable walker (i-walker)," in *2010 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2010, pp. 277–282.

[8] M. Martins, A. Frizera, R. Ceres, and C. Santos, "Legs tracking for walker-rehabilitation purposes," in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. IEEE, 2014, pp. 387–392.

[9] C. Nave, Y. Yang, V. Viegas, and O. Postolache, "Physical rehabilitation based on smart walker," in *2018 12th International Conference on Sensing Technology (ICST)*. IEEE, 2018, pp. 388–393.

[10] R. A. Bachschmidt, G. F. Harris, and G. G. Simoneau, "Walker-assisted gait in rehabilitation: a study of biomechanics and instrumentation," *Ieee Transactions on neural systems and Rehabilitation Engineering*, vol. 9, no. 1, pp. 96–105, 2001.

[11] D. Chugo, T. Asawa, T. Kitamura, S. Jia, and K. Takase, "A rehabilitation walker with standing and walking assistance," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 260–265.

[12] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[13] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[14] (2021, Jan) Jetson nano: Deep learning inference benchmarks. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks