# Feet and legs tracking using a smart rollator

C. Joly, C. Dune, P. Gorce, P. Rives

***Abstract*— Clinical evaluation of frailty in the elderly is the first step to decide the degree of assistance they require. This evaluation is usually performed once and for all by filling standard forms with macro-information about standing and walking abilities. Advances in robotics make it possible to turn a standard assistance device into an augmented device. The existing tests could then be enriched by a new set of daily measured criteria derived from the daily use of standard assistance devices. In this paper we use a smart rollator, equipped with a kinect and odometers, for biomechanical gait analysis. This paper focuses on the method we develop to track the legs and feet position during walking. TODO : a completer**

## I. Introduction

Ageing in society is a worldwide issue that especially impacts northern countries. In France, due to the high care cost and to the limited number of rooms in care institution, the solution that has been chosen by care-givers, frail people and their family is to maintain elderly at home the longest and in the best conditions by giving them an *adapted assistance*.

Clinical evaluation of frailty in the elderly is the first step to decide the degree of assistance they require. This evaluation is usually performed once and for all by filling standard forms with macro-information about standing and walking abilities, e.g. by measuring the time taken to walk $10m$. Advances in robotics make it possible to enhance a standard assistance device by adding sensors and actuators. The existing tests could then be enriched by adding a new set of daily measured criteria derived from the daily use of standard assistance devices. This monitoring will allow to evaluate gait in ambulatory conditions, to measure the evolution of some pathologies, to refine diagnostics and to distinguish autonomy levels. The assistance device is not meant to be an alternative for clinical frailty observation but rather as a complementary tool that gives field information. The data acquired *online* could also be used to control a robotics walker in order to prevent a fall. These new characteristics can extend the use of walkers to more diverse population.

The system used here is a *smart rollator* equipped with sensors, for gait monitoring. The objective is to provide physicians with the features they are used to process to evaluate elderly frailty, while maintaining a low cost and ensuring a good ease of use and by embedding all the sensors on the walker without equipping the patient. And at best, the intelligent walker could deliver others relevant features that will enriched the existing feature set.

This paper focuses on the use of an embedded kinect sensor to segment online the lower limb and estimated. The algorithm returns the pose of the leg and feet with regards to the rollator. The next section gives an overview of existing smart rollators and depicts our smart rollator.. Section III describes the lower limb detection and pose estimation, based on kinect depth map. Section IV focuses on a kalman filtering to refine this estimation. Section V shows experimental results

C.Joly, C. Dune and P. Gorce are with Handibio, EA4322 Université du Sud- Toulon Var, France.

P. Rives is with Lagadic team Inria Sophia Antipolis, France

## II. System description

Depending on the degree of assistance they need, people are prescribed canes, crutches or walkers [1]. The latter can be legged walker or wheeled walkers (rollators). A rollator can be defined as a frame with wheels. It has handles with brakes, and in some case a seat, a basket and a tray (see fig. 1).

Fig. 1. A 4 wheeled rollator

Rollators induce a more natural gait pattern than standard four legged frames. They are designed for people that need less weight bearing [2].

### A. Existing smart rollators

Here is an overview of the the existing smart rollators, focusing on common wheeled walker that connects to a person at the hands. Smart walkers may be used to analyse either the environment or the user's behaviour. Environmental data is dedicated to navigation purpose, such as obstacle avoidance [3], wall following [4], slope compensation [5] or localisation [6], [7]. Even though these functionalities are relevant for people autonomy, especially for the visually impaired, these functionalities are out of the scope of this paper that focuses on gait analyses. A thorough survey on assistance mobility device, focusing on

smart walkers can be found in [8], [9].

Some of the existing Smarts walkers aim at tracking the trajectories of gait features in order to monitor health. The great advantage of such systems is that the user stands at a roughly known position with regards to the walker. Body segment localisation is then made easier.

Walkers can be equipped with force-moment sensors mounted on the walker handles [10], [11], or under the forearm [8], [12] to passively derive some gait characteristics. In both cases it is assumed that the force and moment recorded have cyclic changes reflecting the gait cycle and that these changes depend on basic gait features (cadence, stride time, gait phases). The iWalker [11] quantifies loads exerted through the handles an frame and standards spatio-temporal parameters (such as speed and distance). In [10], a direct comparison between motion capture and force-moment data was studied to detect significant pattern in the force signal. The lateral sway motion of the upper body reflects in peaks in vertical direction and in the corresponding forward moment signal. These peaks coincided with the heel initial contacts and. The forward propulsion force applied by the user is related to the toe-off event from the right and left toe. Finally, the stride (ie. duration of a gait cycle) can be computed from two heel contacts. In [12], a method based on Weighted Frequency Fourier Linear Combiner, is introduced for the same standards gait parameters extraction from force data.

Walker wheel motion measurement can also be used to estimated the user state [3], [13]. The Personal aid for mobility and monitoring project (PAMM) [3] developed health monitoring tools. The PAMM smart walker is an omnidirectional walker design for walking assistance with navigation and monitoring functionalities. Its sensors record user speed and compute the stride-to-stride variability, which have been shown to be an effective predictor of falls. A power spectrum analysis on PAMM's velocity allows to estimate user's stride length and frequency. Besides, the shape of the power spectrum is related to the gait symmetry. Indeed, for a symmetric gait, the energy is located at twice the stride frequency. However, the system can detect asymmetric gait as spectrum with energy located at the stride frequency and at higher frequency. An asymmetrical gait could be an indicator of a physical injury or a minor stroke.

Direct measurement of body segments may be obtained by using ultrasonics sensors or cameras [8], [14]. A vector of ultrasonic sensors can be mounted on the walker to scan the space between the user and the walker and determine coordinate of each leg without adding any marker on the patient [8]. In [14], a camera is mounted on the frame and observes markers on the toes. This marker based toe tracking algorithm allows to calculate step width and provide an accurate assessment of foot placement during rollator use.

The main issue when using that kind of device is that the accuracy of the leg localisation depends strongly on the clothes that the user wears. It is a drawback with regards to method based on odometers or force sensors. The method proposes in [14] by-passes this point by adding markers on the toes. Yet it also by-passes our constraint not to equip the user in order to ensure acceptance and ease of use.

Extrinsic data can then be used to monitor the user's health or to control the walker in order to prevent a fall. The walker-user relative distance can be used to classify the states between a *walking state*, a *stopped state* and an *emergency state* [15]. A laser range finder mounted on the walker acquired the position of the knee with regards to the walker frame. It is assumed that the position of the feet is at the vertical of the knee position, and fixing the human frame in the middle of the feet. User velocity is estimated from the walker velocity obtained by odometers. The *stopped state* occurs when both the walker and the human velocities are null. To distinguish the *walking state* from the *emergency state*, user-walker distance is used. A normal distance distribution is computed to determine the *walking state* based on user data. The robot control tries to bring the user distance right to the mean of the walking state distance distribution.

In [16] the RT-Walker is equipped with laser range finder and perform an estimation of the kinematics of a 7-link human model. The model is used to estimate the position of the user center of gravity (CoG) in 3D. A stable region is determined by analysing the distribution of the C.o.G. position for three subjects with different physiques who walked for 100 seconds with a walker. If the C.o.G is out of the region, the user may fall. The system then brakes enough to compensates for its lightweight and prevent the fall. Notice that the fall detection is restricted to the sagital plane.

In [17], The RT Walker is equipped with vision sensor to classify the user state among four classes : sitting, standing, falling, and walking. The classifier is based on heuristic on the distance between user head, hands and shoulder. Basically, the vision algorithms are based on head tracking, and skin detection. Shoulder detection is performed by finding the higher points of a uniform color region under the head, which seems to lack robustness with regards to environment properties and user clothes.

### *B. Our smart rollator*

In this paper, the system aims at tracking some specific parameters for biomechanical gait analysis, that were chosen in [**?**]:

- step length
- step width
- step frequency
- feet orientation
- heel trajectory
- ankle angle trajectory

Our system is made of a standard 4 wheeled rollator equipped with sensors (fig. 2):

- The Kinect sensor which is viewing the feet of the person.
- Odometers are mounted on the rear wheels to estimate the trajectory of the walker.

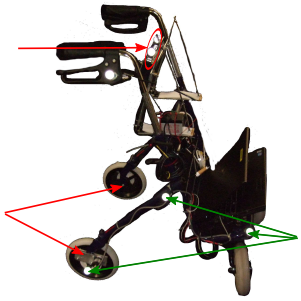


Fig. 2. Our smart rollator : a 4 wheeled rollator equipped with odometers and a kinect sensor

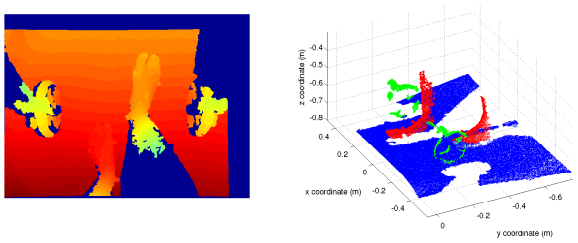



Fig. 3. Left: kinect depth map. The warmer is the color, the larger the depth associated is – Right: 3D points cloud associated to the depth map (for clarity the feet, rollator and ground were segmented).

- A laptop is installed to grab the sensor data
- Finally, motion capture[1] markers are installed in order to compute the ground truth.

In this paper, we focus on the estimation of leg and feet poses in a Frame attached to the Kinect sensor. With such data, some interesting parameters like feet orientation or ankle angle trajectory can be directly computed. This is the main topic of this paper. Computation of step length and width may be done by fusing the results of this paper with odometry (out of scope here). This could be done by integrating the robot motion to computes its pose in a fixed reference frame. Then, the current feet pose could be directly expressed in this fixed reference frame to compute the step length and width.

## III. Algorithm for Kinect processing

The algorithm to process Kinect images is presented in this section. The method aims at fitting a 3D skeleton on the partial kinect data. The 2 legs will be described as two rigid bodies linked with a ball joint (which represent an approximation of the ankle joint).:

- a first segment which links the toe to the heel
- a second segment which has a predefined length starting from the heel in the direction of the leg

To do so, a parametric model will be fitted to each feet and leg. This will be done in four steps:

1) The first step consists in removing from Kinect depth maps the points associated to the ground and those associated to the rollator,
2) The second step consists in making a first segmentation of the feet and legs thanks to a model based method. The model used corresponds to a cylinder (legs) and a plane (feet)
3) The third step consists in optimizing the former segmentation and the model parameters by optimizing both legs and feet model simultaneously, eg. by taking into account the fact that the two legs have the same diameter.
4) Finally, the last step consists in transforming the former 2 body model introduced before. This model will be used as measurements in a Kalman filter (see section IV).

We propose to describe these four steps in the following paragraphs.

### A. Ground and walker segmentation

In our experiments, the walker is used indoors on a flat ground. Since the Kinect sensor is rigidly fixed to the walker, the ground plane will be the same in all images. As a consequence it is possible to compute it once and for all and to consider it as constant. An other solution could consists in detecting the main plane of the 3D points cloud with a RANSAC algortithm. The orientation of the kinect with regards to the ground plane can be used to compute the position of the point in a frame align with the ground. It eases the reading of the point cloud..

The rollator does not move with respect to the Kinect and so is static in the Kinect camera frame. It can be removed from the depth map by using a predefined mask.

In the following, we defined by $\Omega$ the set of the points which do not belong neither to the ground nor to the rollator.

### B. Feet and legs segmentation

The main idea of feet and legs segmentation is based on region growing and inspired by [**?**]. From a current set of points of a member, we look for potential candidates in the neighbouring and then validate it or not. The following paragraphs are dedicated to the presentation of the process. The global algorithm is presented in paragraph III-B.1. Then application for legs and feet segmentation is presented (paragraph III-B.2 and III-B.3).

*1) General algorithm:*

The general algorithm for member (a foot or a leg) segmentation is presented here:

1) Let $\mathcal{M}$ be the set of the current image points belonging to the member and $\mathbf{p}$ a **set of parameters** associated to it
2) Let $\mathcal{N}$ be the set of neighbours associated to $\mathcal{M}$:

$$\mathcal{N} = \{\mathbf{n} \notin \mathcal{M} \mid \exists \mathbf{m} \in \mathcal{M}, \|\mathbf{n} - \mathbf{m}\| \leq s\} \quad (1)$$

where $s$ is a threshold to define (8-10 pixels). Note that in (1) $\mathbf{m}$ and $\mathbf{v}$ are not 3D points but pixels in Kinect image.
3) $\mathcal{N}$ represents the set of new potential candidates. We remove from this set all the points that belongs to the ground or to the walker. Moreover, the points that do

[1] A Qualysis system is used.

not match enough with the current model defined by $\mathbf{p}$ are also removed. Thus, the final candidates are defined by:

$$\mathcal{N}' = \{\mathbf{n} \in \mathcal{N} \mid \mathrm{dist}(\mathbf{n}, \mathbf{p}) \leq s^{pts}\} \cap \Omega \quad (2)$$

where $\mathrm{dist}$ is the function which gives the distance of a point (first argument) to a model (second argument), $s^{pts}$ a threshold and $\Omega$ the set defined in III-A.

4) 2 cases can be distinguished:
   a) $\mathcal{N}' = \emptyset$: there is not any new candidate. The set of points associated to the member is complete. The set $\mathcal{M}$ and the last set of parameters computed $\mathbf{p}$ are returned.
   b) $\mathcal{N}' \neq \emptyset$: in this case, there are new points to add to the model. A new model $\mathbf{p}'$ is computed with the point set:

$$\mathbf{p}' = \mathrm{fit}\left(\mathcal{M} \cup \mathcal{N}'\right) \quad (3)$$

   where $\mathrm{fit}$ is the function that computes a model by fitting the 3D points associated to the set given in argument.
5) The new model $\mathbf{p}'$ is then tested by comparing the mean error to a threshold $s^{mod}$:

$$\mu = \frac{1}{\mathrm{card}\left(\mathcal{M} \cup \mathcal{V}'\right)} \cdot \sum_{\mathbf{m}_i \in (\mathcal{M} \cup \mathcal{V}')} \mathrm{dist}(\mathbf{m}_i, \mathbf{p}) \quad (4)$$

   2 cases can be distinguished:
   a) $\mu > s^{mod}$: the mean error with the new model is too important and so it is rejected. The new points $\mathcal{N}'$ are discarded. The algorithm terminates and returns the set $\mathcal{M}$ and the parameters $\mathbf{p}$.
   b) $\mu < s^{mod}$: the new model is accepted with the complete set of points. We return to **step 1** to make it grow again with the following parameters:

$$\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{N}' \quad (5)$$
$$\mathbf{p} \leftarrow \mathbf{p}' \quad (6)$$

*2) Leg segmentation:*

*a) Model and definition of* $\mathrm{dist}$ *function:*

To segment the legs, a *cylinder model* is used. It is a 5 parameters primitive (4 parameters stand for its axis and one for its radius). The distance considered is the radial distance to the cylinder. Let $\mathbf{m}$ be a Kinect image point whose distance to the cylinder of parameters $\mathbf{p}$ has to be evaluated. It is computed with the following steps:

1) Let $\mathcal{B} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$ be an orthnormal base with $\mathbf{w}$ in the same direction than the cylinder axis and $\mathbf{C}$ a point belonging to this axis. The rotation matrix $\mathbf{R}$ $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ initiale à $\mathcal{B}$:

$$\mathbf{R} = \begin{bmatrix}\mathbf{u} & \mathbf{v} & \mathbf{w}\end{bmatrix} \quad (7)$$

2) Let $\mathbf{M}$ be the 3D point (in the Kinect frame) associated to $\mathbf{m}$ and $\mathbf{M}'$ the coordinates of $M$ in the frame defined by $(\mathbf{C}, \mathcal{B})$:

$$\mathbf{M}' = \mathbf{R}\left(\mathbf{M} - \mathbf{C}\right) \quad (8)$$

Finally, the distance to the cylinder is the difference between its radius (noted $a$ in the following) and the distance between $\mathbf{C}$ and the projection of $\mathbf{M}'$ in the plane $(\mathbf{C}, \mathbf{u}, \mathbf{v})$. With $\mathbf{M}' = [x\ y\ z]^T$, we have:

$$\mathrm{dist}(\mathbf{m}, \mathbf{p}) = |\sqrt{x^2 + y^2} - a| \quad (9)$$

where $\mathbf{p}$ stands for the cylinder parameters (see paragraph **??**) and are used to define $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, $\mathbf{R}$, $\mathbf{C}$ and $a$.

*b) Model parameterization:* :

A cylinder can be defined by the rotation matrix $\mathbf{R}$,the point $\mathbf{C}$ and its radius $a$ which were introduced in the previous paragraph. A rotation matrix has 3 degrees of freedom and can be written has the multiplication of 3 matrices:

$$\mathbf{R} = \mathbf{R_3}(\theta_3) \cdot \mathbf{R_2}(\theta_2) \cdot \mathbf{R_1}(\theta_1) \quad (10)$$

where $\mathbf{R_1}(\theta_1)$ (resp. $\mathbf{R_2}(\theta_2)$, $\mathbf{R_3}(\theta_3)$) stand for the rotation matrix around the first (resp. second, third) canonical axis of angle $\theta_1$ (resp. $\theta_2$, $\theta_3$). $\mathbf{C}$ can be defined by 3 coordinates:

$$\mathbf{C} = \begin{bmatrix}c_x & c_y & c_z\end{bmatrix}^T \quad (11)$$

From (10) and (11), cylinder pose uses 6 parameters. However, it is possible to use only 4 parameters. Using 6 parameters could lead to divergence during optimization. We propose to use only 4 parameters by using the following remarks:

1) Since a cylinder has a revolution symmetry, there is an infinite number of solution to define the base $(\mathbf{u}, \mathbf{v}, \mathbf{w})$. Only the direction of $\mathbf{w}$ is important (it corresponds to the direction of the cylinder); as a consequence, any rotation of $(\mathbf{u}, \mathbf{v})$ around $\mathbf{w}$ can change the results of distance computation. So, we can multiply by the left the matrix $\mathbf{R}$ by any matrix of the form $\mathbf{R_3}(\theta)$ ($\theta \in [0\ 2\pi]$) without modifying the function $\mathrm{dist}$. As a consequence, any value of $theta_3$ is valid. So, we chose to fix it to zero and do not estimate it.
2) Finally, the choice of $\mathbf{C}$ is not unique: every point of the cylinder axis is valid. Thus, there is one degree of freedom to remove. To do it and force they unicity of $\mathbf{C}$, it was chosen to fix the third component ($c_z$, see (11)). Fixing this component is save as soon as the cylinder axis is not perpendicular to the $\mathbf{z}$ axis.[2] Such situation implies that the leg is parallel to the ground, which does appear in our context. So, parameters $c_x$ and $c_y$ can be seen as a function of $c_z$ which is arbitrarily [3]

Finally, each leg is represented by a cylinder which parameters are stored in a 5D vector:

$$\mathbf{p} = \begin{bmatrix}c_x & c_y & \theta_1 & \theta_2 & a\end{bmatrix}^T \quad (12)$$

This representation ensure that there is one and only one possibility to define the point $\mathbf{C}$ and the rotation matrix $\mathbf{R}$.

[2]Which corresponds to the direction perpendicular to the ground, see paragraph **??**

[3]Not really in practice. $c_z$ is chosen so that it is possible to find a good initialisation of $c_x$ and $c_y$ to optimize the minimisation process during fitting.

*c) Definition of* $\mathrm{fit}$ *function:*

The $\mathrm{fit}$ function aims to estimate a vector parameters $\mathbf{p}$ associated to a set of Kinect points $\mathcal{M}$. Soit $\mathbf{f}(\mathcal{M}, \mathbf{p})$ the vector defined by:

$$\mathbf{f}(\mathcal{M}, \mathbf{p}) = \frac{1}{1+\mathrm{e}^{-a}} \cdot \begin{bmatrix} \mathrm{dist}(\mathbf{m}_1, \mathbf{p}) \\ \vdots \\ \mathrm{dist}(\mathbf{m}_N, \mathbf{p}) \end{bmatrix} \tag{13}$$

with $\mathcal{M} = \{\mathbf{m}_i, i \in [1 \dots N]\}$. $\mathrm{fit}$ is then defined by:

$$\begin{aligned} \mathrm{fit}(\mathcal{M}) &= \min_{\mathbf{p}} \left( (\mathbf{f}(\mathcal{M}, \mathbf{p}))^T \cdot (\mathbf{f}(\mathcal{M}, \mathbf{p})) \right) \\ &= \min_{\mathbf{p}} \left( \left(1+\mathrm{e}^{-a}\right)^{-2} \cdot \sum_{\mathbf{m}_i \in \mathcal{M}} \mathrm{dist}^2(\mathbf{m}_i, \mathbf{p}) \right) \end{aligned} \tag{14}$$

In (14), the minimum is computed with the Levenberg-Marquardt method. Initial conditions are fixed as follows:

1) $c_x$ and $c_y$ are initialized with the barycenter of the 3D points cloud. The value of $c_z$ is then *fixed* as the mean of the $z$ coordinates associated to the 3D points cloud ($z$ is not modified during the minimization process).
2) $\theta_1$ and $\theta_2$ are provided thanks to the covariance matrix associated to the 3D points cloud. The axis cylinder $\mathbf{w}$ is assumed to be equal to the eigen vector associated to the largest eigen value (since the length of the leg is larger than the radius cylinder). The parameters $\theta_1$ and $\theta_2$ can the be easily computed.
3) $a$ is initialized to 0.05m.

*Remark 1:* In (13) and (14), $(1+\mathrm{e}^{-a})^{-1}$ is used to penalize solutions with a large radius. Since the 3D points cloud associated to the legs corresponds only to a partial cylinder (around 30deg), the radius is not easily observable and the function to minimize tends to be "plate". Introducing $(1+\mathrm{e}^{-a})^{-1}$ yields in slightly favoring cylinder with smaller radius. Moreover, it can be obserbed that $(1+\mathrm{e}^{-a})^{-1}$ is always between 1 and $1/2$ and can not have a strong influence on the final result or make converge the final result to a radius close to zero.[4]

*3) Foot segmentation:*

*a) Model and definition of* $\mathrm{dist}$ *function:*

For the feet, a planar model is used and is parameterized by its 4 equation parameters:

$$\begin{cases} \mathbf{p} = \begin{bmatrix} a & b & c & d \end{bmatrix}^T \\ \text{avec } \|\mathbf{p}\| = 1 \end{cases} \tag{15}$$

so that every 3D points $[x\ y\ z]^T$ belong to the plane verify:

$$ax + by + cz + d = 0 \tag{16}$$

The function $\mathrm{dist}$ corresponds to the Euclidean distance to the plane. Let $\mathbf{m}$ be an image point associated to the foot and $\mathbf{M} = [x\ y\ z]^T$ the 3D point associated to it:

$$\mathrm{dist}(\mathbf{m}, \mathbf{p}) = \frac{|ax + by + cz + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{17}$$

[4]This is not be the case with a factor such as $\sqrt{a}$ like in [**?**].

*4) Definition of* $\mathrm{fit}$ *function:*

The $\mathrm{fit}$ function is defined as the minimization of the error function to the plane with the constraint $\|\mathbf{p}\| = 1$. Let $\mathcal{M}$ be the set to fit and $\mathbf{g}(\mathcal{M}, \mathbf{p})$ the vector defined by:

$$\mathbf{g}(\mathcal{M}, \mathbf{p}) = \begin{bmatrix} ax_1 + by_1 + cz_1 + d \\ \vdots \\ ax_N + by_N + cz_N + d \end{bmatrix} \tag{18}$$

with:

$$\begin{cases} \mathcal{M} = \{\mathbf{m}_i, i \in [1 \dots N]\} \\ \{\mathbf{M}_i, i \in [1 \dots N]\} \text{the 3D points associated to } \mathcal{M} \\ \forall i \in [1 \dots N],\ \mathbf{M}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \end{cases} \tag{19}$$

So we have:

$$\mathrm{fit}(\mathcal{M}) = \min_{\mathbf{p} \in \mathbb{R}^4,\ \|\mathbf{p}\|=1} \left( (\mathbf{g}(\mathcal{M}, \mathbf{p}))^T (\mathbf{g}(\mathcal{M}, \mathbf{p})) \right) \tag{20}$$

(20) can be exactly solved with Lagrange multiplier method. Let $\mathbf{A}$ be the matrix defined by:

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & z_N & 1 \end{bmatrix} \tag{21}$$

We have:

$$\mathrm{fit}(\mathcal{M}) = \mathbf{vp}(\mathbf{A}^T\mathbf{A}, 1) \tag{22}$$

where $\mathbf{vp}(\mathbf{X}, i)$ stands for the normalized eigen vector associated to the i-th eigen value of $\mathbf{X}$ (In (22), it corresponds to the smallest eigen value).

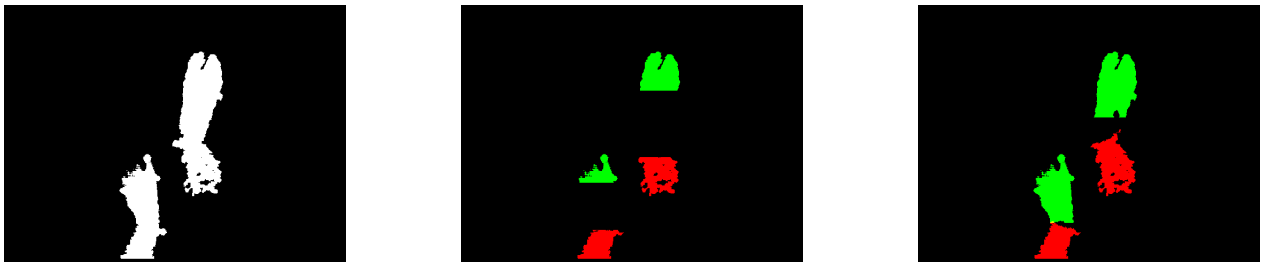
*5) Initialization of the set:*

The global method proposed to segment the feet and the legs needs to provide an initial set of image points for each member. This is done by the following steps and illustrated on Fig. 4 :

1) Firstly, it can be noticed that after removing the ground and the rollator from the Kinect image, there are two "big" sets of points that can be easily segmented and corresponding to the left and right member (Fig. 4a). If there are other small sets due to noise, they can thus be discarded.
2) Then each part contains both the foot and the leg. Because of the geometry of the system, the bottom part corresponds systematically to the leg. Thus, we chose to initialize the leg with the last bottom quarter and the foot with the first bottom quarter (Fig. 4b)
3) Finally, the algorithm makes grow these sets, thus computing the final segmentation (Fig. 4c).

### C. Model optimization

After the segmentation step, a model has been computed for each member. We propose now to optimize these parameters.



(a) After ground and rollator segmentation (b) At initialization) (c) Final result)

Fig. 4. Steps of segmentation

*1) Legs:*

In the following, the two legs will be optimized simultaneously. During the segmentation step, they were optimized independently. So, it was guaranteed that both legs have the same radius. At this step, we propose to estimate both parameters from the 3D points extracted from the segmentation with the constraints that they have the same radius. So, we define a new vector to optimize:

$$\mathbf{p_{legs}} = \begin{bmatrix} c_x^g & c_y^g & \theta_1^g & \theta_2^g & c_x^d & c_y^d & \theta_1^d & \theta_2^d & a \end{bmatrix}^T \quad (23)$$

where:

- $c_x^l$, $c_y^l$, $\theta_1^l$ and $\theta_2^l$ stand for the parameters defining the pose of the cylinder associated to the left leg,
- $c_x^r$, $c_y^r$, $\theta_1^r$ and $\theta_2^r$ stand for the parameters defining the pose of the cylinder associated to the right leg.

Thus, the whole points are fitted in the same optimization process (of course, each point is labeled with its own leg). The methods is roughly the same that during segmentation. Nevertheless, it is interesting to note that the new Jacobian matrix has a block structure with two zero blocks.

*a) Feet:*

Finally, the sets associated to the feet are optimized by using stronger thresholds in order to remove outliers.

### D. Final parameters

The final step consists in transforming the legs and feet parameters into a bone representation. For each side, we have:

1) The heel is computed as the intersection between the axis cylinder and the plane associated to the foot. More precisely, since the plane corresponds to the *top of the foot*, it does not correspond exactly to a point which is higher. Then, a segment starting from this point in the direction of the leg is defined to visualize the leg.
2) The toe we get it by using the foot plane and the 3D points associated. It is computed in steps:
   a) Firstly, the 3D point cloud is projected onto the plane computed previously
   b) Then, the convex hull of this projection is computed with its barycenter.
   c) The vector joining the "hell" to this barycenter provides the foot direction. The further 3D point in this direction define the toe.

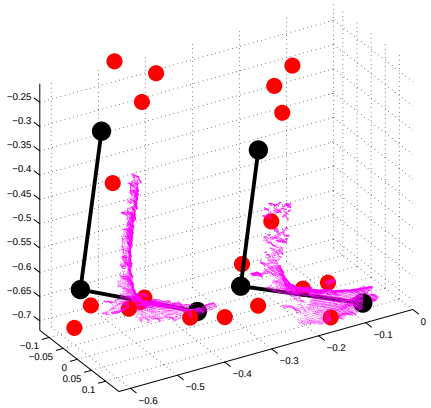


Fig. 5. Example of result provided by the algorithm (black bones). The kinect data are plotted in yellow and the ground truth markers are in red.

An example of the results provided by the algorithm is provided on fig. 5

## IV. Kalman filtering

### A. Motivation

Although the algorithm presented in the previous section can segment well the Kinect points cloud frame per frame, it does without time coherence. So, we chose to add a Kalman filter by side to smooth the trajectory and reject outliers.

### B. Model and evolution matrix

The idea is to define a state with the 3 points defining the two segments. A classical constant speed model is used. For each side, we use the following state vector:

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{X_1}_k^T & \mathbf{X_1}_k^T & \mathbf{X_1}_k^T & \mathbf{V_1}_k^T & \mathbf{V_2}_k^T & \mathbf{V_3}_k^T \end{bmatrix}^T \quad (24)$$

where $\mathbf{X_1}$ (resp. $\mathbf{X_2}$) is the vector representing the 3D Euclidean coordinates of the toe (resp. heel). $\mathbf{X_3}$ stands for the 3D coordinates of the last point of the bones representation. It is defined such that it $\mathbf{X_3} - \mathbf{X_2}$ is in the direction of the cylinder and has a predetermined norm. $\mathbf{V_1}$, $\mathbf{V_2}$ and $\mathbf{V_3}$ are the associated speeds. The evolution matrix is given by:

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{I_{9\times 9}} & \mathbf{I_{9\times 9}}\Delta t \\ \mathbf{0_{9\times 9}} & \mathbf{I_{9\times 9}} \end{bmatrix} \cdot \mathbf{X}_{k-1} \quad (25)$$

### C. Measurement equations

Measurement are provided by the segmentation algorithm. 3 kind of measurement can be distinguished:

1) Direct measurement from the segmentation algorithm
2) Constraint between $\mathbf{X_1}$ and $\mathbf{X_2}$: the norm of the difference is constant but unknown
3) Constraint between $\mathbf{X_2}$ and $\mathbf{X_3}$: the norm of the difference is constant and known

*1) Direct measurements:*
Direct measurements are provided by the segmentation algorithm which provides $\mathbf{X_1}$, $\mathbf{X_2}$ and $\mathbf{X_3}$. The observation matrix is obvious and is made by identity blocks for the positions and zero blocks for the speeds.

*2) Constraint between $\mathbf{X_1}$ and $\mathbf{X_2}$:*
In this work, we assume that the feet are like a rigid body, so that the norm of $\mathbf{X_2} - \mathbf{X_1}$ is constant. Thus, we have:

$$\frac{d}{dt}\left((\mathbf{X_2} - \mathbf{X_1})^T \cdot (\mathbf{X_2} - \mathbf{X_1})\right) = 0 \tag{26}$$

So, we have the following constraint equation:

$$(\mathbf{V_2} - \mathbf{V_1})^T \cdot (\mathbf{X_2} - \mathbf{X_1}) = 0 \tag{27}$$

Since this constraint is almost always verified, a covariance matrix close to zero is associated. This equation is non linear and the observation matrix implies to compute the Jacobian of (27)

*3) Constraint between $\mathbf{X_2}$ and $\mathbf{X_3}$:*
By construction, the norm of $\mathbf{X_3} - \mathbf{X_2}$ is constant and known. So, we have the following constraint:

$$(\mathbf{X_3} - \mathbf{X_2})^T \cdot (\mathbf{X_3} - \mathbf{X_2}) = d^2 \tag{28}$$

where $d$ is known and fixed by advance. Similarly to the previous constraint, a covariance matrix close to zero is associated to (28).

### D. Outlier rejection

Finally, the Kalman prediction can be applied to outlier rejection. The idea is to compare Mahlanobis distance between the prediction of $\mathbf{X_i}_k$ and its measurement provided by the segmentation algorithm, with is the sum of the covariance matrices of the prediction and the measurement. So, we compute the following value for each point $i \in [1 \dots 3]$:

$$\left(\mathbf{x_i}_k^{pred} - \mathbf{x_i}_k^{meas}\right)^T \cdot \left(\mathbf{P_i}_k^{pred} + \mathbf{R_i}^{meas}\right)^{-1} \cdot \left(\mathbf{x_i}_k^{pred} - \mathbf{x_i}_k^{meas}\right) \tag{29}$$

where $\mathbf{P_i}_k^{pred}$ is the covariance associated to the prediction of $\mathbf{X_i}_k$ (i.e. $\mathbf{X_i}_k^{pred}$) and $\mathbf{R_i}^{meas}$ is the covariance matrix associated to the measurement of $\mathbf{X_i}_k$ (i.e. $\mathbf{X_i}_k^{meas}$). The value computed in 29 is computed to a threshold computed thanks to $\chi^2$ distribution. The measurement is rejected if the value is above the threshold.

This algorithm allows us to detect and reject spurious values computed by the segmentation algorithm. For example, if the fitting strongly fails and converge to a time-inconsistent value, the Kalman filtering with outlier rejection is able to reject it.

## V. Results

### A. Presentation of the experiment

Our rollator with Kinect was tested with a person walking during two cycles of walk. A ground truth is provided by the motion capture sensor. We propose in this experiment to compute the evolution of several parameters during the walk:

- The feet angles in the Kinect frame
- The angle between the two segments computed, that we approximate to the ankle angle
- The height of the heel

These parameters provide very useful information about the walk. For example, it is possible to know if the cycle is standard or not. By combining these information with the odometry, it is possible to make statistics about the length of the steps (out of the scope of this paper).

Unfortunatelly, the data collected do not correspond to a full legnth cycle walk. This is due to problem of visibility of the whole leg in the Kinect frame: our Kinect was a bit to close of the member and there was a problem of angle of view. The first problem will be corrected in the future by using a Prime sense[5] adapted for close measurements. The second problem will be corrected by slightly modifying the viewing angle of the sensor. To these reasons, the following results can only be considered as **preliminary results**.

### B. Results analysis

Fig. **??** show the results of error in orientation of the feet. It can be seen that the elevation angle (computed with respect to the ground) is pretty well estimated (less than 5deg of error in most cases). This is an important result that show that the system may be able to pretty well estimate if a foot is on the ground or not.

Concerning the bearing angles, the results are less precise. However, the results are globally consistent and can be used to detect anomalies in the angle feet (see fig. **??**).

The ankle angle estimation is also consistent (fig. **??**) even if it is a bit less precise than the feet elevation angles.

Finally, we wanted to compare the $z$ coordinate of the heel to the $z$ coordinate of the heel computed by our model. However, we know that these two points are not exactly and that the point associated to the heel in our model is higher. To test the consistency of our model, we test if the real heel is in the axis defined by our model (fig **??**). Then, the distance between the two points should be constant.

## VI. Conclusion

## References


[1] B. Joyce and K. R.L., "Canes, crutches and walkers," *J. American Family Physician*, vol. 2, no. 43, pp. 535–542, 1991.

[2] F. Van Hook, D. Demonbreun, and B. Weiss, "Ambulatory devices for chronic gait disorders in the elderly," *J. American Family Physician*, vol. 8, no. 67, pp. 1717–1724, 2003.

[3] M. Spenko, H. Yu, and S. Dubowsky, "Robotic personal aids for mobility and monitoring for the elderly," *IEEE Trans. on Neur. Sys. and Rehab. Eng.*, vol. 14, no. 3, pp. 344–351, 2006.


[5] An other constructor of Kinect-like sensor who sells a small-range certified version of the sensor.

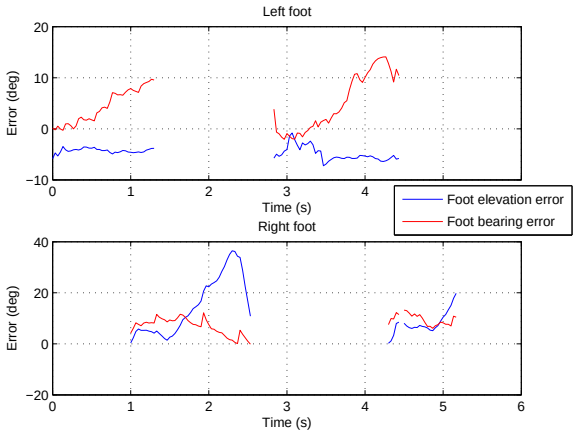



Fig. 6. Orientation errors for the feet

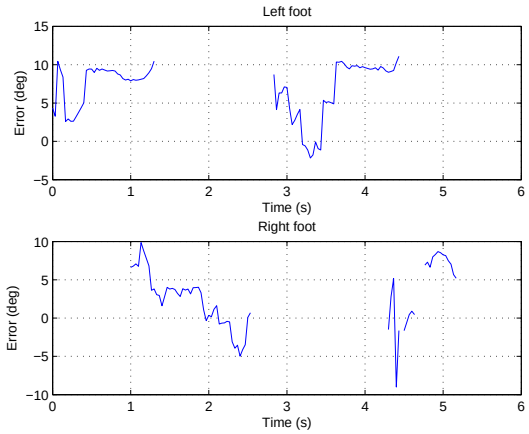



Fig. 7. Error on ankle angle estimation

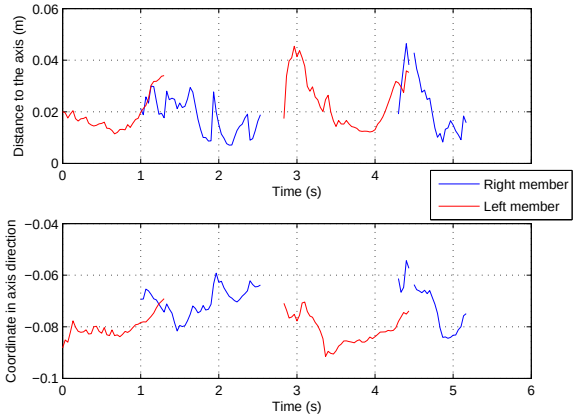



Fig. 8. Difference between real heel position and estimation

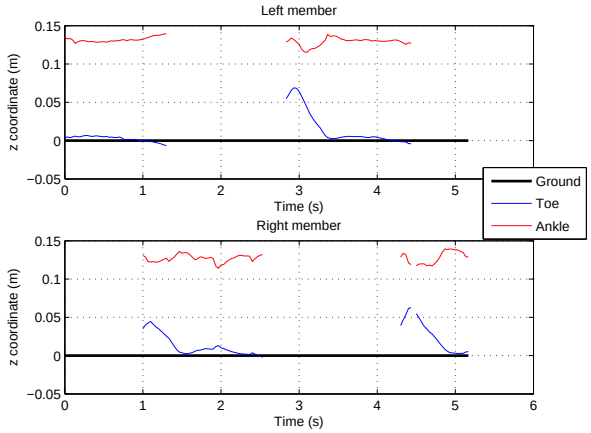



Fig. 9. $z$ trajectory of the toes and "heels" estimated by the algorithm


[4] H. Yu, M. Spenko, and S. Dubowsky, "An adaptive shared control system for an intelligent mobility aid for the elderly," *J. of Auto. Rob.*, vol. 15, no. 1, pp. 53–66, 2003.

[5] Y. Hirata, A. Hara, and K. Kosuge, "Motion control of passive intelligent walker using servo brakes," *IEEE Trans. on Robotics*, vol. 23, no. 5, pp. 981 –990, oct. 2007.

[6] S. Kotani, H. Mori, and N. Kiyohiro, "Development of the robotic travel aid hitomi," *J. of Auton. Robots*, vol. 17, pp. 119–128, 1996.

[7] S. MacNamara and G. Lacey, "A smart walker for the frail visually impaired," in *IEEE Int. Conf. on Rob. and Autom.*, vol. 2, 2000, pp. 1354–1359.

[8] A. Frizera, R. Ceres, J. Pons, A. Abellanas, and R. Raya, "The smart walkers as geriatric assistive device. the simbiosis purpose." in *Int. Conf. of the Int. Soc. for Geron.*, 2008, pp. 1–6.

[9] M. Martins, C. Santos, A. Frizera, and R. Ceres, "Assistive mobility devices focusing on smart walkers: classification and review," *Rob. and Aut. Sys.*, December 2011.

[10] M. e. a. Alwan, "Basic walker-assisted gait characteristics derived from forces and moments exerted on the walker's handles: Results on normal subjects," *Medical Engineering and physics*, vol. 29, p. 380–389, 2007.

[11] J. Tung, "Development and evaluation of the iwalker : An instrumented rolling walker to assess balance and mobility in everyday activitiess," Ph.D. dissertation, University of Toronto, 2010.

[12] A. Frizera, J. Gallego, E. Racon de Lima, A. Abellanas, J. Pons, and R. Ceres, "On line cadence estimation through force interaction in walker assisted gait," in *ISSNIP Biosignals and Biorobotics Conference*, Vitoria, Brazil, 2010, pp. 1–5.

[13] J.-P. Merlet, *New trends in Mechanism Science: Analysis and Design*. Springer, 2012, ch. Preliminary design of ANG, a low cost automated walker for elderly, pp. 529–536.

[14] *Feasibility of objectively assessing foot placement during rollator use*. Toronto Rehab. Res. Day, Toronto, Canada, 2009.

[15] Y. Hirata, A. Muraki, and K. Kosuge, "Motion control of intelligent passive-type walker for fall-prevention function based on estimation of user state," in *IEEE Int. Conf. on Rob. and Autom.*, May 2006, pp. 3498–3503.

[16] Y. Hirata, S. Komatsuda, and K. Kosuge, "Fall prevention control of passive intelligent walker based on human model," in *IEEE/RSJ Int. Conf. on Int. Rob. and Sys.*, sept. 2008, pp. 1222 –1228.

[17] S. Taghvaei, Y. Hirata, and K. Kosuge, "Vision-based human state estimation to control an intelligent passive walker," in *IEEE/SICE Int. Symp. on Sys. Int.*, 2010, pp. 146–151.