Intelligent Service Robotics manuscript No.
(will be inserted by the editor)

A Closed-loop Approach for Tracking a Humanoid Robot Using Particle Filtering and Depth Data

[*]Pablo A. Martínez · [*]Xiao Lin · Mario Castelán · Josep Casas · Gustavo Arechavaleta

Received: date / Accepted: date

Abstract Humanoid robots introduce instabilities during biped march that complicate the process of estimating their position and orientation along time. Tracking humanoid robots may be useful not only in typical applications such as navigation, but in tasks that require benchmarking the multiple processes that involve registering measures about the performance of the humanoid during walking. Small robots represent an additional challenge due to their size and mechanic limitations which may generate unstable swinging while walking. This paper presents a strategy for the active localization of a humanoid robot in environments that are monitored by external devices. The problem is faced using a particle filter method over depth images captured by an RGB-D sensor in order to effectively track the position and orientation of the robot during its march. The tracking stage is coupled with a locomotion system controlling the stepping of the robot towards a given oriented target. We present an integral communication framework between the tracking and the locomotion control of the robot based on the Robot Operating System (ROS), which is capable to achieve real time locomotion tasks using a NAO humanoid robot.

Keywords Humanoid Robot · Tracking · RGB-D sensor · Particle Filter · ROS

[*]The first two authors contributed equally.

P.-A. Martínez(✉) · M. Castelán · G. Arechavaleta
Robotics and Advanced Manufacturing Group,
Centro de Investigación y de Estudios Avanzados del IPN,
Saltillo, Coahuila, 25900, México.
E-mail: pablo.martinezglz@cinvestav.mx

M. Castelán
E-mail: mario.castelan@cinvestav.mx

G. Arechavaleta
E-mail: garechav@cinvestav.mx

Xiao Lin · Josep Casas
Image Processing Group,
Department of Signal Theory and Communications,
Universitat Politècnica de Catalunya,
Barcelona, 08034, Spain.
E-mail: xiao.lin@estudiant.upc.edu

Josep Casas
E-mail: josep.ramon.casas@upc.edu

1 Introduction

Localization is a classical and well studied problem in robotics. A wide range of sensor devices and methodologies have been used to face this problem in order to accurately estimate the robot pose. Localizing a robot is a crucial step in any application for which the robot must interact with the environment, i.e. for tasks such as navigation, grasping and obstacle avoidance. A challenging instance of this problem is humanoid robot localization since displacements are generated as a by-product of a complex kinematic structure making contact with the ground surface. It is well-known that such locomotion behavior implies inaccurate odometry estimation and important drift at short distances. In this sense, it is necessary to close the pose estimation and motion execution loop by incorporating the estimation process within the control scheme.

Moreover, humanoid robots are designed to perform human-like tasks, such as biped locomotion and human interaction in man-made environments. In this regard, the robot should predict the behavior of the human while performing its self-localization. The prediction of human motion intention has been a major axis of research in both robotics and neuroscience communities. The particular problem of goal-directed locomotion in humans consists of studying the underlying motion patterns such that the shape of human trajectories is recovered. In [3], a database of human walking trajectories is studied and, as a result, a control model of human locomotion is suggested based on optimal

control tools. Also, some inverse optimal control formulations have been introduced in [32] and [30]. In [4], a statistical model of human walking is reported. The common concern of all these works is the prediction of human walking paths according to a spatial goal on the plane in terms of position and orientation coordinates.

It is then required to estimate the humanoid robot position and orientation to arrive at the meeting point with a precise body orientation. There exist several works in human-humanoid interaction pointing out the importance of socially acceptable robot motions [37]. In this sense, the robot orientation at the meeting point should maximize safety and visibility criteria. The objective is to improve human comfort by positioning the robot in the human field of view. In this context, the robot is considered as a service assistant, not only an autonomous machine. Thus, social rules and protocols must be considered when the robot moves [38].

Commonly, an intelligent room is equipped with several sensors to perceive the moving agents, e.g. humans and robots. In this work, we take advantage of the perception capabilities available in these scenarios to cope with the humanoid localization problem. Particularly, we explore the idea of using an external RGB-D sensor, to estimate the position and orientation of a walking humanoid robot to perform locomotion tasks. These sensors are cheap and popular and can be incorporated into closed spaces in a relatively easy manner. We focus our study on providing the robot with localization capabilities, which can be coupled with any strategy that requires an efficient and accurate humanoid tracking tool. It is worth mentioning that tracking a humanoid is not a trivial task, for which the state of the art has invested efforts on using landmarks in both the robot and the scenario, gathering previous knowledge of the world such as 3D maps and constraining the motion of the robot to its upper articulations, since problems in localization are mainly caused by the robot walking. We also consider that accuracy at the level of centimeters must be achieved for tasks performed in a reduced space.

Motivated by the above reasons, this paper uses an RGB-D sensor to estimate the position and orientation of a humanoid robot. This sensor is located on the ceiling with a top-down field of view. The estimation process is based on a particle filter and naturally coupled with the humanoid locomotion control to accurately reach the meeting point given by a predefined position and orientation on the plane.

The main contributions of this article are:

- A depth-based tracker that is able to accurately follow the locomotion behavior of a humanoid robot.
- A control scheme that considers reaching a target position and orientation while updating linear and angular velocities in accordance with the current localization of the humanoid robot.
- A ROS communication framework linking the depth tracker and the humanoid locomotion.

The paper is organized as follows: Section 2 presents a review of the related work; Section 3 describes the problem of tracking and localizing a humanoid robot using the depth data from an RGB-D sensor and a particle filter implementation to face this problem; Section 4 describes a control scheme that incorporates the estimated robot pose within an active localization approach developed on the ROS integration framework; experimental results are then provided in Section 5; finally concluding remarks and future work are presented in Section 6.

2 Related work

This section has been divided into two parts. The first part presents work related to the humanoid robot localization, emphasizing the applicability of using internal as well as external sensors to track the locomotion of the robot. The second part describes the relevant work related to tracking humans in the context of smart rooms.

2.1 Humanoid robot localization

The walking process of legged humanoid robots generally produces noisy motions due to the effects of biped locomotion. For example, situations such as joint backlash or foot slip with the floor generate an inaccurate execution of motion tasks. As a consequence, it is important to keep the robot localized in the environment. In order to face this problem, visual sensors have been mostly used as the main source of input data for tracking systems. To track the motion of the robot, the sensors can be mounted inside the humanoid. One of the reasons to use built-in cameras or range sensors is the weight constraint related to the limited payload of the humanoid robot. Alternatively, sensors may be adapted onto the humanoid or as external sources in order to extend the sensing capabilities if more integral systems are needed.

2.1.1 Built-in sensors

The problem of estimating the humanoid robot pose by visual data has been faced applying different approaches. One of these is the Extended Kalman Filter (EKF), which allows the integration of multiple sensors. In [40] an EKF was proposed where data from proprioceptive sensors and the walking pattern generator was fused with the vision system in order to obtain the robot motion estimation. This method has been successfully applied on the human-size HRP-2 robot in a SLAM methodology able to build a map of sparse 3D

points for localizing the robot in indoor environments [9]. The EKF methodology has been also tested using a small-size humanoid robot. For example, in [29], the EKF prediction step was performed relating the torso and joints velocities of a NAO robot by differential kinematics. To this end, the visual data was processed using the Parallel Tracking and Mapping (PTAM) approach [19] which emulates a 3D visual sensor. The EKF correction step of [29] considered fusing the provided camera pose with data from the inertial unit measure mounted in the chest of the robot. More recently, a novel visual-based localization approach using bundle adjustment [42] over the HRP-2 humanoid robot was presented in [2]. Later, a monocular localization framework [1] was proposed to predict the visibility of 3D points in a previously built sparse 3D map using stereo visual SLAM.

A common feature between the above approaches is that they do not include information about the localization of the robot in order to perform the locomotion tasks, i.e., the instantaneous position and orientation of the robot is not considered within a control module for reaching a specific goal. Although filter-based approaches allow the integration of multiple sensors, visual information (features, cues, etc.) becomes the core of the method if an accurate odometry is required. In this sense, filter based approaches rely on a robust visual-based localization system in order to maintain a successful tracking of the humanoid robot. Currently, PTAM has proved to be one of the most reliable visual-based localization system when enough cues are available. It has been integrated into an active localization method to control the locomotion of a humanoid robot [21,22]. Thereby, it is possible to guide the trajectory of the robot surrounding an object of interest in order to estimate its geometry from a monocular video sequence acquired while the robot is walking [22]. Furthermore, a method to control the march of the robot towards directions that are favorable for visual based localization was presented in [21]. Here, a set of statistical criteria are used for the analysis of the 3D map and reprojected 2D points for targeting the robot towards directions of rich visual information. Unfortunately, for PTAM to work accurately, the system requires an initialization step which is sensitive to the physical distance between the first two images captured by the monocular camera.

Monte-Carlo techniques have been also tested over the NAO platform fusing data from a laser scanner, an inertial measurement unit and joint encoders in order to estimate the robot pose [18]. These approaches were later improved by including new observation models based on the visual data from a monocular camera [28]. Depth sensors have also been considered, for instance, in [20] an RGB-D sensor was mounted over the head of a humanoid robot in order to solve for its 6D torso pose. Here, the observation model was able to integrate depth data. The proposed approach was tested extensively in a real-world environment, including climbing stairs, when a further monocular-based observation model was applied in order to increase localization accuracy. This work was inspired by the octree representation of [43], which allows the successful construction of a dynamic map for real-time collision-free path planning tasks. However, it requires an accurate initialization of the static world and, as a consequence, a previous process of dense 3D scanning and modeling of the navigating space becomes essential. Recent filtering modalities successfully integrate depth data either to reduce the drift in the humanoid localization process [11] or to enlarge the region where the robot is able to be localized [6].

Other techniques are more focused on object-centered localization for obstacle avoidance. For example, in [5], a laser scanner was mounted on the hip of the robot in order to construct a height map representation of the environment. This representation provides a grid that is helpful for fitting planes and identifying obstacles by using the height information contained in the cells of the grid. Also, in [24], a GPU implementation model-based approach was proposed to track the 6D pose of objects in order to localize a camera mounted on the head of an HRP-2 robot. To this end, the dimensions of the objects needed to be known in order to successfully localize the robot and perform the required tasks.

2.1.2 External sensors

A Navigation system using an external sensor to track a small size humanoid robot pose was presented in [15]. Here, 3D features and virtual visual servoing (VVS) [8] were successfully integrated. An RGB-D sensor was used as an external device to get depth features to be combined with image features rendered from a CAD model of the robot. The pose estimation process was based on the difference between the rendered and real image features. Although this approach may be applied to a variety of robotic platforms, the method struggled to accurately estimate the position and orientation of the robot when it ranged outside ± 10 cm and $\pm 10^{\circ}$ respectively.

In [25], a calibrated arrangement of retro-reflective markers was placed over the head of the HRP-2 robot in order to facilitate the problem of multi-camera tracking. The aim of this approach was to continuously solve for the extrinsic parameters of the camera of the robot in order to generate the reconstruction of the environment. This reconstruction was divided into floor and obstacles. The plane of the floor was incrementally updated during the walking of the robot and the obstacle segmentation was based on color. A path planner that operates at the level of the footstep was used for an autonomous navigation task. Later, retro-reflective markers were also incorporated for tracking objects in the scene. The method was named Naviga-

tion Among Movable Obstacles (NAMO) [41]. These approaches provide accurate methodologies to recover the full pose of the robot in real time at the cost of carefully calibrating retro-reflective landmarks along both the robot and the scene.

2.2 Tracking humans in smart rooms

Visual-based human tracking is stated as the process of detecting and tracking a human body over a sequence of images. The process is challenging due to the large variation in human appearance and motion, as well as changes in camera viewpoints. If estimating the whole body pose over time is also considered, the problem scales from tracking to Human Motion Analysis.

A taxonomy of human motion was presented in [31], where two main approaches were introduced: model-base and model-free. Model based approaches use a predefined human body model for pose estimation [35]. Here, the process consists of modeling (constructing the likelihood function) and estimation (finding the most likely pose given the likelihood surface). The second approaches establish a direct relation between image observation and pose, since an *a priori* human body model is not available.

Human motion analysis could be considered as an specific application of real time object tracking. In [45] a suitable categorization of the tracking methods was presented describing the steps to build an object tracker. Visual human tracking approaches can be classified into three categories according to different types of data used: color, depth and the fusion of both.

Color based tracking methods commonly follow the same framework. Foreground appearance is modeled with color or texture information, then the foreground model is matched in successive frames in order to find the correspondence between them. For example, in [7] the object localization is formulated as a gradient optimization problem using a color histogram regularized by spatial masking with an isotropic kernel, while in [27] an adaptive color-based particle filter was used. These models are not reliable enough in some situations like sudden illumination changes and unexpected occlusions, which usually happen in real time tracking.

The advantage of RGB-D consumer depth sensors such as *Microsoft Kinect* or *Asus Xtion* provides an affordable way to acquire depth information at good resolution and low cost. This has fostered the presentation of depth or depth and color-based real time tracking methods in the last years. The intuitive way to exploit depth information is to perform tracking directly on the depth images, which takes the advantage of the shape cues given by depth value. For example, the Histogram of Oriented Depth (HoD) has been successfully applied for human tracking [39], performing an Histogram of Oriented Gradient (HoG)-like method on depth data. HoD locally encodes the direction of depth changes as a shape feature making the model more robust to illumination issues. Background modeling has also been used in approaches when depth data is available. Hansen *et al.* [16] proposed to build the background model on the joint distribution of depth and intensity information, which accommodates changes in both domains. Then, clusters of pixels significantly different from the background model are tracked by an Expectation Maximization (EM) algorithm. In [44], only depth data were used for human tracking. Here, by means of Chamfer distance, a binary head-shape template is matched against an edge map extracted from the depth array. Once the head is located, the whole body contour is extracted and separated from the background and tracking is achieved using the motion of the person.

The other way to deal with depth data is to reconstruct the 3D scene and perform tracking in 3D. Works in this scope usually focus on more complicated problems such as skeleton tracking [12, 13]. The advantage of this 3D tracking is that most of the data used is measured in real world metrics, facilitating the parameter setting in tracking systems. Unfortunately, due to the higher complexity when dealing with 3D data, these methods often require GPU implementations in order to run in real time.

Tracking humans using a depth sensor from a bird-eye view was presented in [34] and [26]. The approach presented in [34] does not rely on background modeling for foreground/background segmentation, but it is based on a feature descriptor to maximize the detection of a discriminatively trained head-shoulder classifier. Moreover, the approach proposed in [26] uses particle filter implementation over depth data for human tracking. The main application is to analyze the behaviors of the costumer (the human is not walking) during their buying activities within the shelves, simulating the installation of the depth sensor in the ceiling of a supermarket. Therefore, their proposed model only considers the upper part of the body (head, shoulders and arms) and it is separated in two: one 2D model, representing head and shoulders, for the estimation of the person localization and one 3D model that determines the arms motion fitting some geometrical primitives (cylinders for arms and forearms, elliptic cylinder for torso and rectangular planes for hands).

3 RGB-D tracking-localization

Commanding a humanoid robot to move and reach a target point may lead to inaccuracies. This usually occurs due to the fact that the controller does not know a possible drift in the actual stepping of the robot. In order to address this problem, we propose a scheme where the robot navigates while

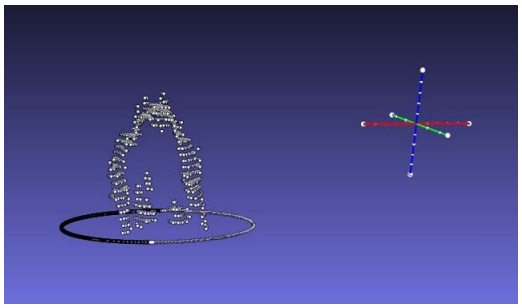


Fig. 1: **Example of applying the proposed RGB-D tracking-localization approach.** Point cloud extracted in the robot model with projected points on the *xy*-plane and Ellipse fitting.

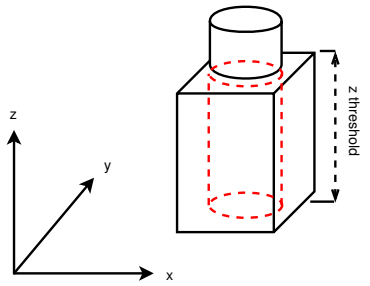


Fig. 2: **The robot model.** The robot model is designed as the combination of a cuboid and a cylinder.

an external depth sensor obtains the data to track it. The RGB-D sensor provides a depth map with real world distances between the visible humanoid surface and the sensor for each pixel on the image, which makes it possible to partially reconstruct the 3D point cloud of the scene, which enriches the information that we can exploit during the tracking process. This is also beneficial in estimating orientation, because of the sharper object boundaries provided by depth maps in comparison with color images.

The main task for the tracker is to estimate position and orientation of the robot by analyzing the point cloud of each frame, then publish them in real time through the communication system [33], so that the controller can subscribe to this information, adjust the movement of the robot accordingly and reach the target with higher accuracy.

3.1 Particle filter implementation

The tracking method used in this paper is based on particle filtering. A zenithal RGB-D sensor is used to capture depth data to track the robot from a top view of the scene.

In this section, the proposed robot tracking system is described. The initial position and orientation are provided by the user with the help of a graphical interface where they can choose these data over the current RGB image of the observed scenario. Therefore, the particle filter is manually initialized on the original position and orientation of the robot. Then, the tracker runs on a particle filtering process in order to estimate the position of the robot. After the position is obtained, the region of the robot will be separated from the background around the estimated position in the depth image. Finally, the orientation is estimated based on the extracted region by using ellipse fitting as shown in Figure 1.

3.1.1 Position

Like most particle filter algorithms, we recursively approximate the posterior probability density function $p(\mathbf{x}_t|y_t)$ for the current state of the model $\mathbf{x}_t$ by evaluating the likelihood of the observation y_t according to a set of weighted particles $\left\{\mathbf{x}_t^i, w_t^i\right\}$. Here $\mathbf{x}_t^i$ stands for a sample point in the state space for the i-th particle at time t and w_t^i is its associated weight. Each of these particles represents a random sample propagated by the prior probability density from the last approximation at time $t-1$ by a dynamic model. The whole process basically consists of four steps.

- Resampling: N Particles $\left\{\mathbf{x}_t^i, w_t^i\right\} \sim p(\mathbf{x}_t|y_t)$ in the last iteration at time t are resampled based on the resampling strategy in order to make sure the resampled particles $\left\{\mathbf{x}_t^{i'}, 1/N\right\}$ get distributed in a more reasonable way. The resampling strategy in our method is Sample Importance Resampling [14] (SIR), which means that particles are selected according to their weight. Particles with large weights are duplicated, and low weight particles are more probably deleted for keeping the total number of particles unchanged.
- Propagation: The resampled particles are propagated to generate a new set of particles $\left\{\mathbf{x}_{t+1}^i, w_{t+1}^i\right\} \sim p(\mathbf{x}_{t+1}|y_t)$ based on the dynamic model $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$, which describes how the system changes over time. In our method, a Gaussian function is used.
- Weighting: In order to approximate the true posterior distribution using discrete particles, we need to weight them with a likelihood function which represents the correspondence between the model and the new observation. These weights should be normalized so that the sum of them equals 1. In our approach, we define the observation as the number of points in it, which is counted within a θ_t degree rotated robot model centered on the position of the particle in 3D space, where θ_t is the estimated orientation in the last iteration. The robot model is designed as the combination of a cuboid and a cylinder as show in Figure 2.
  Figure 1 shows a point cloud of the robot captured from the top view, in which the head, hands and shoulder are the visible parts. The geometric model (Figure 2) is employed to match the point cloud with this shape in 3D space. Therefore, in each observation, the points with

their z-coordinates higher than the z-threshold should be fitted into the cylinder of the model, representing the point cloud of the robot head. Points with their z-coordinates lower than the z-threshold should be fitted in the cuboid area, representing the point cloud of the shoulder and hands. The weight of each observation is then computed by counting the number of points correctly fitted in the geometric model. The model of the robot gives some cues about the shape of the robot, which makes it more robust to the perspective changes when the robot moves.

The likelihood function is then defined as in equation (1), where y_t denotes the observation and q^i the probability mass associated with each i-th particle

$$q^i = \frac{p(y_t|\mathbf{x}_t^{i'})}{\sum_{i=1}^{N} p(y_t|\mathbf{x}_t^{i'})}, \tag{1}$$

and the weight for each particle equals its likelihood:

$$w_{t+1}^i = q^i \tag{2}$$

– Estimation: The estimation result is obtained by performing the weighted average of particles $\{\mathbf{x}_{t+1}^i, w_{t+1}^i\} \sim p(\mathbf{x}_{t+1}|y_{t+1})$:

$$\mathbf{x}_{t+1} = \sum_{i=1}^{N} w_{t+1}^i \cdot \mathbf{x}_{t+1}^i \tag{3}$$

3.1.2 Orientation

The orientation of the robot is designed to be estimated separately from the previous step with the purpose of reducing the dimension of the state space and increasing the computation speed of the tracking system. The speed of change of the orientation is small enough to be estimated directly by a target region extraction and ellipse fitting process without strongly affecting the overall tracking accuracy.

The weighted summation of the locations of all the particles shows the estimated position $\mathbf{x}_{t+1}$ obtained in the previous step. Then a larger window compared to the observation window (the geometric model) used in the location estimation process is chosen to extract the point cloud of the robot centered at the estimated position. The extracted point cloud is projected to the xy-plane, which roughly represents the shape of the robot from the top view. Principal Component Analysis (PCA) is employed to estimate the direction with largest data variation among these projected 2D points. This direction is treated as the major axis of the ellipse and the minor axis of the ellipse which is perpendicular to the major axis indicates the estimated orientation. Besides, we assume the orientation of the robot changes continuously over time which means the estimated orientation θ_t at time t should be within the interval $[\theta_{t-1} - \theta_h, \theta_{t-1} + \theta_h]$ where θ_h represents a threshold for maximal orientation changing in a time slot.

It is worth mentioning that fitting ellipses has been explored for tracking the position of the head of walking humans in a multiple sensor environment, where cameras and laser range finders are combined. The work, presented in [23] also benefits from using a particle filter for tracking the motion of the humans.

As far as other filters such as Kalman are concerned, in a linear system with Gaussian noise, the Kalman filter is optimal. In a system that is nonlinear, the Kalman filter can be used for state estimation, but the particle filter may give better results at the price of additional computational effort. In a system that has non-Gaussian noise, the Kalman filter is the optimal linear filter, but again the particle filter may perform better [36]. In our situation, since the noise in the tracking process is non-Gaussian, which mainly comes from the bipedal walking, it is more reasonable to follow the particle filter framework than the standard Kalman filter.

Data: Localization $\mathbf{x}_{CoM}^{ref}$ at current time k,
orientation θ_c^{ref} at current time k,
target position $\mathbf{x}_t$,
target orientation θ_t.
Result: Reference linear velocity $\dot{\mathbf{x}}_{CoM}^{ref}$ at time $k+1$,
reference angular velocity $\dot{\theta}_c^{ref}$ at time $k+1$.
while $\mathbf{x}_{CoM}^{ref}$ *outside stopping region* **do**
  $\dot{\mathbf{x}}_{CoM}^{ref} = -\lambda_x(\mathbf{x}_{CoM}^{ref} - \mathbf{x}_t)$
  $\dot{\theta}_c^{ref} = -\lambda_\theta(\theta_c^{ref} - \theta_t)$
  Apply a WPG given $(\dot{\mathbf{x}}_{CoM}^{ref}, \dot{\theta}_c^{ref})$
  Generate locomotion with inverse kinematics
end

Algorithm 1: The robot performs the locomotion task of reaching a target position and orientation.

4 Locomotion control

Once the localization of the robot has been computed from the RGB-D tracker, the task to reach a target position with a specific orientation may now be formulated in a locomotion framework. For this task, we propose a locomotion control that directs the next position and orientation of the robot to lie along the path that minimizes the distance between the current localization and a given target in terms of position and orientation.

Let $\mathbf{x}_{CoM}^{ref} = [x_w, y_w]^T$ be the reference position of the center of mass (CoM) on the xy-plane and θ_c^{ref} the orientation angle, both provided by the tracker described in Section 3. The current state of the robot is defined by the pair $(\mathbf{x}_{CoM}^{ref}, \theta_c^{ref})$ and the given target state is defined as $(\mathbf{x}_t, \theta_t)$.

The reference linear velocity of the CoM, $\dot{\mathbf{x}}_{CoM}^{ref}$ is computed considering a proportional control based on the distance between the current estimate of the robot's CoM po-

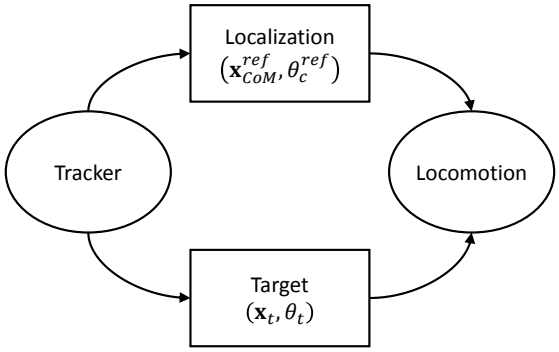


Fig. 3: **ROS framework.** Following the ROS programming paradigm, the tracker and the locomotion control are coded as nodes (elipses) while the common topics shared between them are the localization and the target (rectangles).

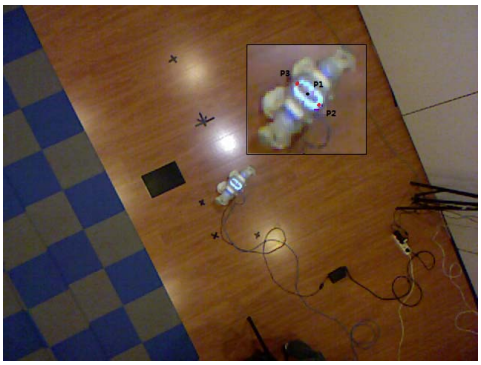


Fig. 4: **Ground truth marking.** Three points are labelled for each color image in which the black point $p_1(x_1, y_1)$ represents the position of the robot, and the vector p_3-p_2 stands for its orientation on the image.

sition and the computed target position. Likewise, for the reference angular velocity, $\dot{\theta}_c^{ref}$, the difference between the current and target orientation is used. Therefore, the errors

$$\mathbf{e}_x = \mathbf{x}_{CoM}^{ref} - \mathbf{x}_t \quad \text{and} \quad e_\theta = \theta_c^{ref} - \theta_t$$

are regulated by imposing the exponential convergences

$$\dot{\mathbf{e}}_x = -\lambda_x \mathbf{e}_x \quad \text{and} \quad \dot{e}_\theta = -\lambda_\theta e_\theta,$$

where λ_x and λ_θ are constant proportional gains. This procedure is performed while the robot does not reach the end of the trajectory and is formally described in Algorithm 1.

The input of the walking pattern generator (WPG) is given by $\dot{\mathbf{x}}_{CoM}^{ref}$ while the output considers a dynamically stable trajectory of the CoM, the position of the foot in contact and the next footstep placement. The WPG solves quadratic programs with a predefined time horizon as it is proposed in [17]. In this case, the reference orientation θ_c^{ref} is used to express the inequality constraints that define the admissible region to place the next footstep. The computation of the joint trajectories of the robot from the WPG outcome is based on the real-time inverse kinematics method suggested in [10].

4.1 ROS integration framework

In order to provide an integral experimental framework, the previously explained tracker and locomotion approaches were integrated using Robot Operating System [33]. This platform is convenient when the communication between different systems is required. In our case, the RGB-D camera serves as the main sensor for the tracker system, while the physical actuators of the humanoid robot are directly affected by the locomotion control system. The ROS platform can be also thought of as a programming tool that helps integrating sensors and actuators that work separately but require a communication protocol to interact and function integrally.

For achieving the required communication, ROS is organized into a protocol that uses nodes, messaging and topics. Figure 3 depicts the messaging strategy designed for communicating the tracker and locomotion systems in our experimental application. In the figure, the tracker and locomotion systems are shown within an elipse and represent two nodes. The purpose of these nodes is to carry out the main processes of the system, i.e. the tracker node estimates the position and orientation of the robot at a current time while the locomotion node determines the immediate linear and angular velocities that need to be applied to control the robot in order to get closer to the target state.

As far as the two topics of our system are concerned, these are depicted as rectangles and are referred to as localization and target topics. The function of a topic can be regarded as that of receiving the messages published by the different nodes of the system. The localization topic receives the current state of the robot published by the tracker node while the target topic receives the target localization of the the current task. Note how, although the target state remains fixed during a single experiment, the tracker node publishes this value as it is manually selected from the graphical interface of our application.

Finally, the arrows appearing in Figure 3 represent the messages that are passed between nodes and topics. In this regard, it is worth mentioning that the tracker node is aimed at publishing messages onto both the localization and target topics, the locomotion node is subscribed to these topics, i.e. it reads every message published on the topics by the tracker node.

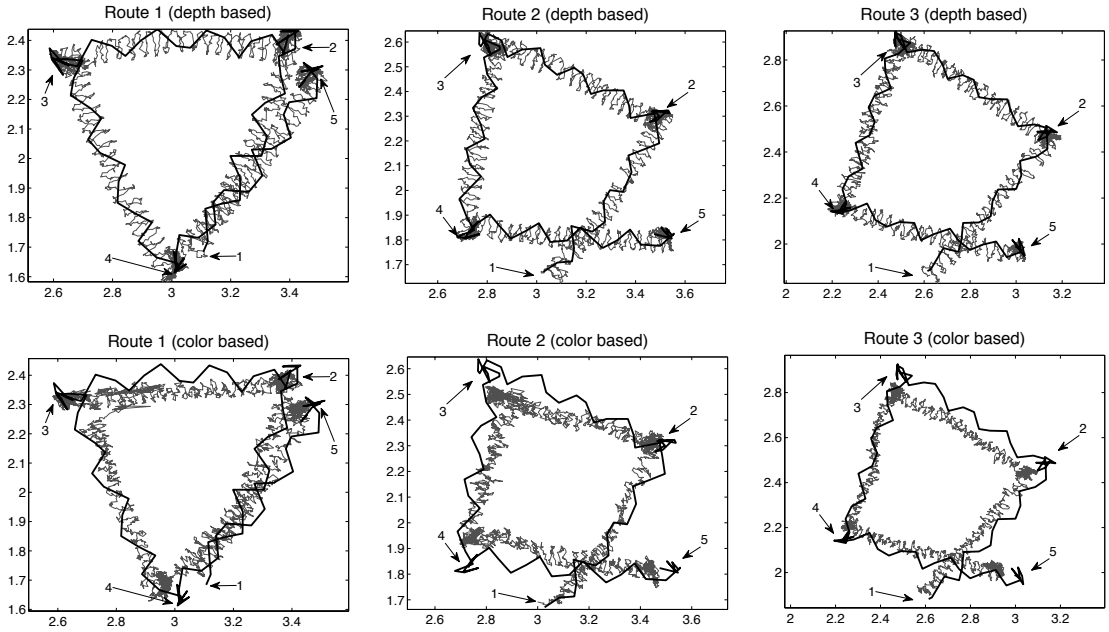


Fig. 5: **Tracking results.** Three routes are shown column-wise. Each route was performed twice, considering depth (top) and color (bottom) information. The ground truth appears with a thick black line while the tracked trajectory appears in gray. The numbers in the diagrams indicate the locations the robot had to visit, starting at location 1 and stopping at location 5. As no control was considered in this experiment, the robot did not reach the desired places with accuracy.

5 Results

This section presents the experimental evaluation of the approach applied on several navigation tasks performed by a humanoid robot. Our first experiment is related with only tracking the motion of the humanoid robot, i.e., no closed loop is considered here. The main idea underlying this experiment is to assess the performance of the particle filter when either depth or color are chosen for driving the filter. The principal conclusion for this experiment is that depth information greatly outperforms color. For this reason, the second experiment includes the evaluation of the proposed control scheme only considering the depth-based particle filter.

5.1 Recording the ground-truth

We have labeled the ground truth position and orientation of the robot along the trajectories for the purposes of quantitative verification on the accuracy of the tracker in our navigation system. Since the experiments are performed in real time, color and depth images used for marking the ground truth are captured every specific number of frames during the tracking process, in order to avoid potential interference with the realtime tracking and control system. Specifically, as Figure 4 shows, three points are labelled for each color image in which the black point $p_1(x_1, y_1)$ represents the position of the robot, and the vector p_3-p_2 stands for its orientation on the image. With these points labelled on the color images, depth values can be obtained from its corresponding depth image. These data is then transformed to world coordinates according to camera calibration parameters, which results in the actual position and orientation of the robot in the world reference frame. Note that, for the first experiment we have used a minimal separation of forty frames during the video sequence for keeping the marking task manageable. With respect to the second experiment a minimal separation of six frames was used. The tracking and control system has an approximate response of four published messages (localization) per second. Also, the image recording process takes time that could affect the system rate, for this reason not all images are saved.

5.2 Experiment 1: tracking

The aim of this experiment is to explore the benefit of using depth information over color information for the particle filter. To this end, we compared three similar routes where the robot was commanded to visit five different locations. Note that no closed loop was considered in this experiment, therefore no correction on error-to-target was observed during the march of the robot. Figure 5 presents the graphical results of the tracking, where the ground truth is depicted in thick black and the resulted tracked trajectory is shown in light gray. The visual inspection of the figure reveals that

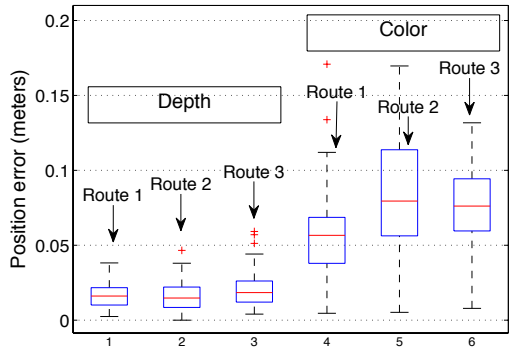


Fig. 6: **Position error.** The figure shows the error in position for the three routes of Experiment 1. Note how depth information significantly outperforms color information.

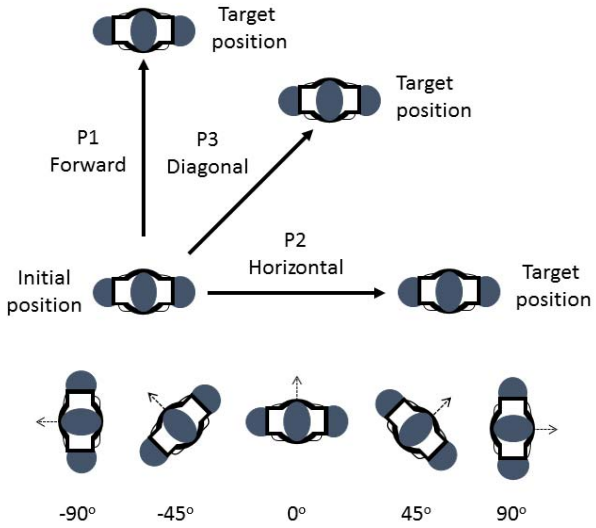


Fig. 7: **Experimental setting (position and orientation).** The initial position and the three final positions of the robot are shown at the top. Target positions are 1 m away from the initial position. Forward, diagonal and horizontal senses are observed in each experiment. The five different orientations at the three final position that the robot has to reach in each experiment are shown at the bottom, ranging from $-90°$ to $90°$.

depth information is more suitable for the success of the particle filter in tracking.

An explanation for this is that the generated particles in the color-based tracker are weighted by comparing the color histogram of the observation from this particles with the source color histogram of the target obtained in the first frame. The color histogram contains no geometric information about the robot, which makes it less robust to perspective changes than the geometric model exploited in the depth-based tracker. For instance, the color based tracker prefers to track the visible part of the robot, which may not coincide with the correct location. Besides, the particle weighting method used in the depth-based tracker consists of simply counting the number of points fitted in the model, which means lower complexity than computing the color histogram for particles in the color-based method.

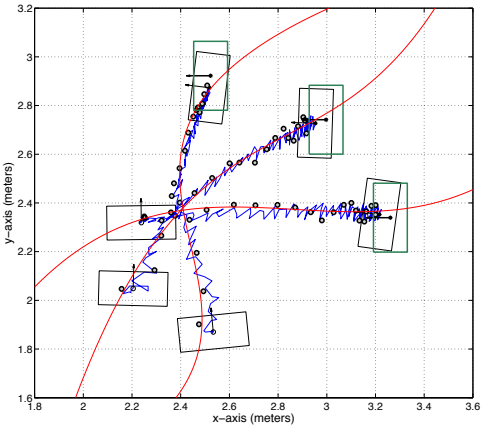
These results are supported by the quantitative results shown in Figure 6. The statistical analysis of the obtained data is presented using box plots. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. The figure shows how the depth-based case performs better in terms of both minimal error (smaller than 0.05 m) and repeatability of results. As far as error in orientation is concerned, no great difference was exhibited between depth and color based trials.The mean error was $4.6°$ with a standard deviation of $3.8°$, meaning that the big majority of the robot states were tracked with an accuracy of at most $10°$ in orientation.

5.3 Experiment 2: tracking and control

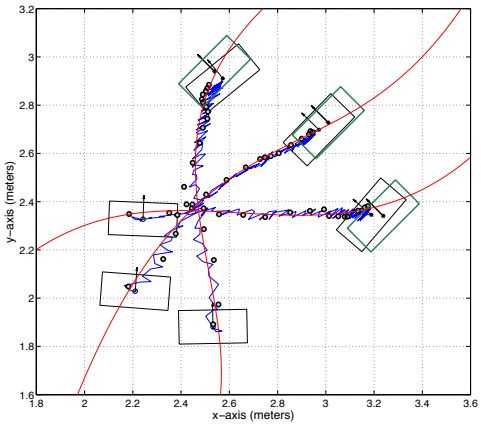
This experiment tests the effect of the depth-based tracker in the proposed control scheme. The experiment is divided into three main navigation tasks related with the final position to be reached by the robot, i.e., the sense of the robot march is performed in forward, horizontal and diagonal modes w.r.t. the initial position. Figure 7 depicts the three position settings. The direction of the paths to be followed is labeled as P1, P2 and P3, respectively for the forward, horizontal and diagonal paths. For all experiments, the distance between the initial and final positions is 1 m.

Providing the accurate position of the robot along its locomotion is not the only relevant feature of a robust tracker. The current orientation needs also be estimated in order to provide the locomotion control with enough information for commanding the robot in more challenging tasks. In this sense, a set of five final orientations was incorporated for each of the three target positions described above, for a total of fifteen experiments (3 final positions $\times$ 5 final orientations).

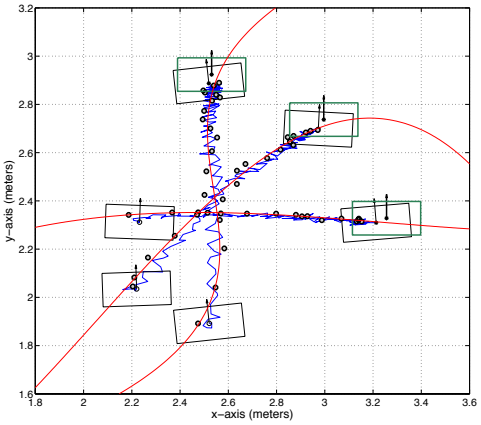
It is important to note that the initial orientation for all experiments is about $0°$, i.e., the robot starting position was always facing towards the forward direction, as shown in Figure 7 (top). In Figure 7 (bottom), the five different orientations required in the target status are shown, ranging from $-90°$ to $90°$. Our system was tested using a kinect as RBG-D sensor and a NAO humanoid robot. The processes ran over a local network formed by one of the servers installed on the smart room (2.8Ghz, 4GB RAM), a laptop (2.53Ghz, 4GB RAM) and the Aldebaran® NAO v4 robot (1.6GHz, 1GB RAM)) with an approximate response of 4 published messages per second.



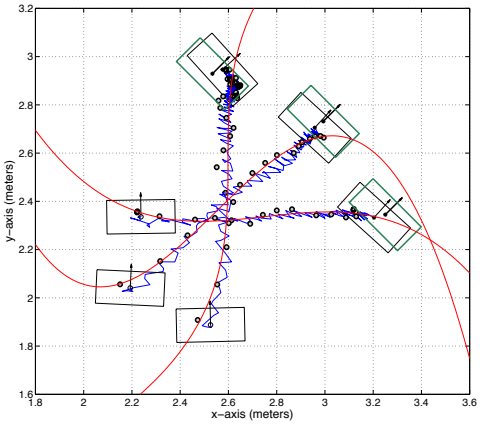
(a) Target orientation $-90°$.



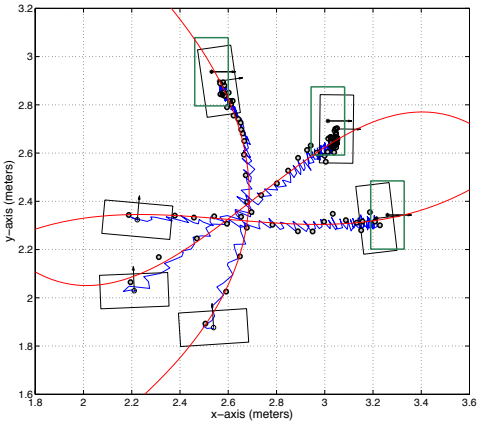
(b) Target orientation $-45°$.



(c) Target orientation $0°$.



(d) Target orientation $45°$.

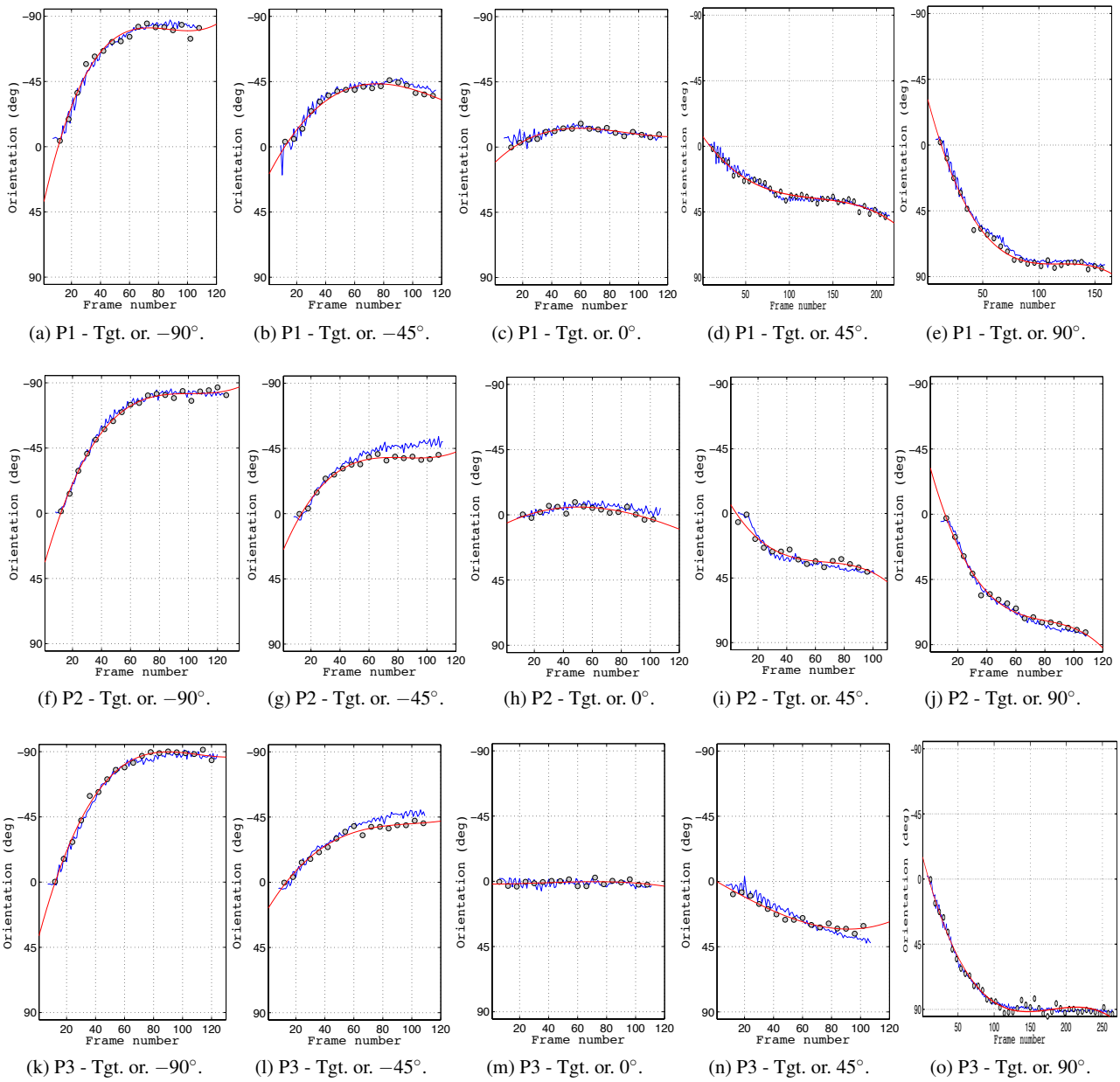


(e) Target orientation $90°$.

Fig. 8: **Qualitative results (position).** The performance of the approach in terms of position is shown in the figure. Blue lines depict the orientation estimated by the tracker while red lines represent the polynomial curves fitted from the ground truth positions (marked with black circles). The black-edged rectangles represent the initial and final positions of the robot, the green-edged rectangles depict the target positions. The small black arrows show the respective orientation (measured w.r.t. x-axis)

5.3.1 Stopping criterion

In order to stop the march of the robot when the target is reached, we considered a Euclidean distance stopping condition of at least 5 cm from the target. This criterion was based on the overlapped area between the bounding box of the robot and the ideal bounding box if the robot is centered at the target. Note how, when the robot is reaching the tar-



(a) P1 - Tgt. or. −90°. (b) P1 - Tgt. or. −45°. (c) P1 - Tgt. or. 0°. (d) P1 - Tgt. or. 45°. (e) P1 - Tgt. or. 90°.

(f) P2 - Tgt. or. −90°. (g) P2 - Tgt. or. −45°. (h) P2 - Tgt. or. 0°. (i) P2 - Tgt. or. 45°. (j) P2 - Tgt. or. 90°.

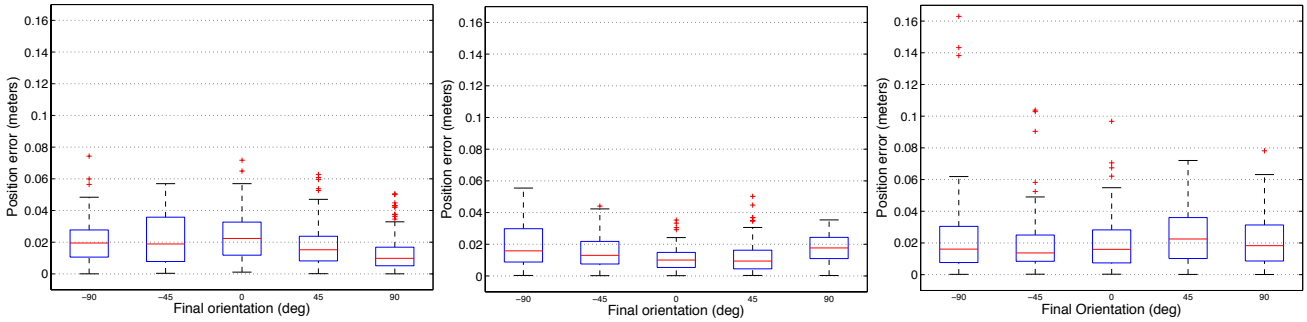
(k) P3 - Tgt. or. −90°. (l) P3 - Tgt. or. −45°. (m) P3 - Tgt. or. 0°. (n) P3 - Tgt. or. 45°. (o) P3 - Tgt. or. 90°.

Fig. 9: **Qualitative results (orientation).** The performance of the approach in terms of orientation is shown. Blue lines depict the orientation estimated by the tracker while red lines represent the polynomial curves fitted from the ground truth positions (marked with black circles).

get, the applied velocity becomes negligible which produces a walking-in-place motion. As a consequence, a noise is induced in the sensor lectures and the tracker node. For this reason, a stopping orientation criterion of at most 10° was chosen based on a supposed resulting drift of 10 cm, which would cause propagating a position error due to divergence in orientation, i.e, without any control that corrects this error.

5.4 Qualitative results

We start our discussion with results regarding the accuracy of the tracker to estimate the position of the robot. To this end, Figure 8 depicts plots showing the aerial view of the fifteen different experiments performed by the robot. The figure is divided into five diagrams. Each diagram presents the three main paths: forward, diagonal and horizontal, followed by the robot in order to reach the target position. Each diagram shows results related with a single target orientation, i.e., in diagram (b) the three main paths regarding the



(a) Position error P1. (b) Position error P2. (c) Position error P3.

Fig. 10: **Statistical analysis of the position errors.** The statistical analysis of the position data is shown in the image. Each diagram shows the position errors in meters related with a single target position (P1 in (a), P2 in (b) and P3 in (c)) combined with the five final orientation.

final orientation of $-45°$ are presented, while diagram (d) depicts those related with the final orientation of $45°$. For all the diagrams, the trajectory followed by the humanoid robot estimated by the tracker system is colored in blue. For comparison, the ground truth positions (depicted by black circles) are fitted to a polynomial curve (in red). The initial and final positions are depicted through black-edged rectangles while the target positions are shown in green. The respective orientations associated with each rectangle (measured w.r.t. x-axis) are represented with black arrows. We decided to use an interpolation method in order to relax the manual ground-truth marking of all the frames in the videos. For the sake of obtaining an error calculation that included all video frames and not only those corresponding to the sampled ground-truth indexes, we used a linear interpolation. Although this strategy is not necessarily accurate it does seem to represent the departure of the tracked states from the general tendency of the marked ground-truth.

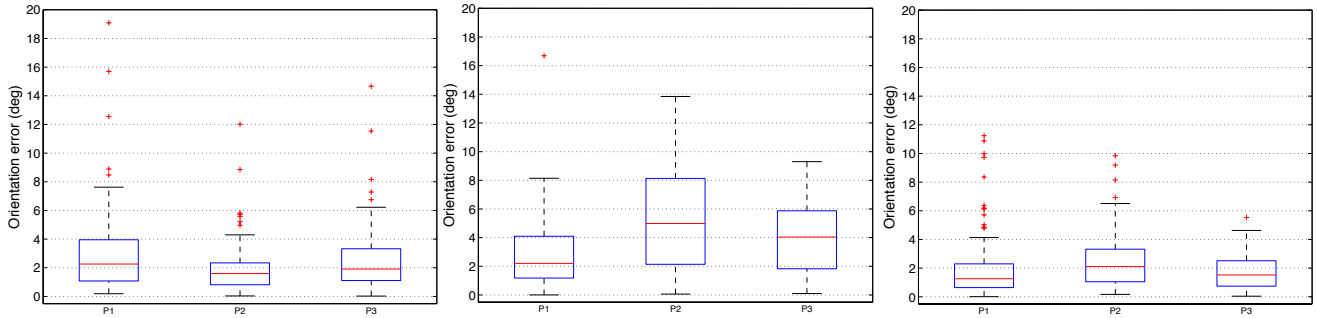
There are several features to note from Figure 8. Firstly, in all the experiments, the final tracked position appears very close to the target, considering the 5 cm stopping condition. The comparison between the estimated trajectory and the ground truth (blue and red lines respectively) show the accuracy of the tracker system. It is important to notice in the trajectories that the natural swinging motion on the small size humanoid robots can be fully recovered. Another relevant feature to note from the figure is the effect of the final orientation in the shape of the tracked trajectory, for example, a visual analysis of the five horizontal paths reveals important differences between the stepping required to achieve each final orientation, as well as the sense of direction the humanoid had to incorporate during its march towards the final goal.

Following with the discussion, the estimated robot orientation is shown, for all the experiments, in Figure 9. The figure is organized in three rows and five columns, corresponding to the three target positions and the five final orientations, respectively. For each diagram, the blue line represents the robot orientation estimated by the tracker system while the ground truth is marked with black circles. The red line represents the polynomial curve fitted from the ground truth observations. It is noticeable how all the experiments perform well in terms of the nodes of the system i.e. the tracker and the humanoid robot control interact coherently. This can be appreciated along the gradual changes of the initial orientation until the desired orientation is reached. Note also how in most of the experiments the number of frames required to reach the position and orientation targets are between 110 and 120. Nonetheless, for three of them (d), (f) and (o) it took more than 150 frames. This can be explained as a consequence of the proportional gain related to controlling the position of the robot. i.e., the 5 cm stopping condition may not be easily reached if the proportional gain has considerably decreased as a consequence of the proximity error with the target position. Our methodology, however, can be coupled with other control paradigms that consider a more sophisticated convergence.

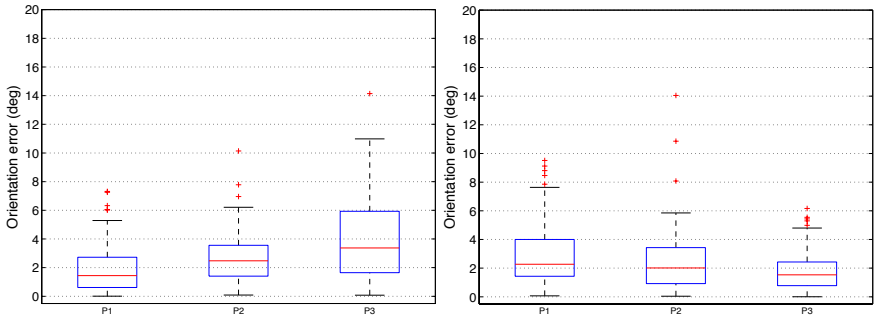
It is worth commenting on path P2 (g), as $-45°$ seems to be a challenging final orientation. This is because the locomotion control causes the robot to be close to the target position yet struggling to reach the target orientation and fulfilling the stopping criteria quickly. For this reason, there is a greater number of frames (near the target position) where the robot appears to be walking almost in the same place (minimal displacement due to a small linear velocity), which causes the tracker to perform less accurately than in other cases.

5.5 Quantitative results

In order to measure the quantitative performance of the proposed approach, we computed the Euclidean error between the values estimated by the tracker and polynomial curves fitted by the ground truth observations. The data correspond-



(a) Orientation error $-90°$. (b) Orientation error $-45°$. (c) Orientation error $0°$.



(d) Orientation error $45°$. (e) Orientation error $90°$.

Fig. 11: **Statistical analysis of the orientation errors.** The statistical analysis of the position data is shown in the image. Each diagram shows the orientation errors in degrees related with a single final orientation ($-90°$ in (a), $-45°$ in (b), $0°$ in (c), $45°$ in (d), $90°$ in (e)) combined with the three final positions.

ing to errors in position is shown with box plots in Figure 10 while quantitative errors in orientation are depicted in Figure 11.

Figure 10 is organized into three diagrams, each of them showing the position errors (in meters) related with a single target position combined with the five final orientations, i.e. diagram (a) shows the five final orientations related to the forward path (P1). In a similar way, Figure 11 depicts five diagrams representing the five orientations combined with the three final positions and, in this case, the orientation errors are shown in degrees.

Let us start the discussion with Figure 10, where we can observe the position error during the trajectory followed by the robot. With the exception of outliers, the recorded error is less than 6 cm for all the experiments, thereby supporting the accuracy of tracking with respect to position. Note also how the presence of outliers does not generally compromise the accurate performance of the system, while most of them are not greater than 10 cm.

As far as Figure 11 is concerned, the orientation error is less than $10°$ for most experiments, excluding a few outliers. It is important to notice that tracking orientation in small size humanoids robots is a challenging problem, compared for instance with wheeled robots considering the same system configuration, i.e. a top view camera-sensor. This may be explained as the outcome of the swinging motion generated during the bipedal walking of such small humanoids, which produce great noise in the data acquired by the sensor.

5.6 Discussion

The results presented in this section correspond to a tracking system that relies on fitting a geometric primitive in order to determine, in real time, the position and orientation of a walking humanoid robot. This means to fit the model proposed in Section 3 according to the physical dimensions of the robot. In this sense, when fitting an ellipse, our methodology can be used for humanoids robots that present an elongated distribution along their shoulders. Although this is the case of most humanoids, some robots such as Atlas of Boston Dynamics, that carries a battery box in its back, may depart from this elliptical elongation, therefore requiring the fitting of a different geometrical primitive.

The depth-based tracking approach proposed in this paper employs a simple yet generic geometric model in the particle filtering process for tracking humanoid objects in the scene. Compared with other target models such as CAD models, the simple geometric model strongly reduces the computational complexity for weighting the particles generated in each frame by intuitively counting the number of

points fitted in the model. Also, this is performed while keeping the module generic enough to track humanoid objects with different sizes and appearances for a further multi-object tracking task. The reason for choosing the Zenithal depth camera as the external camera for our tracking system is that less occlusion and perspective changes are involved from the top view.

The geometric model used in our approach exploits the shape of humanoid objects from the top view in the tracking process and is suitable for improvement with robustness to partial occlusions. The orientation estimation is more sensitive to occlusions since the ellipse fitting method highly relies on the completeness of the point cloud data. Nevertheless, this can be addressed by introducing temporal consistency in the orientation estimation process. Particularly, in our implementation we smooth the estimated orientation along time when the target orientation changes greatly in consecutive frames.

As far as other tracking methods are concerned, we decided to focus solely on our approach due to the nature of the tracking problems approached by the state of the art, i.e., tracking while walking is not completely viable in those methods due to the following constraints: (a) emphasis on tracking the joints and articulations of a rather static robot, (b) evidence of failure when departures from initial and reference orientations occur, (c) relying on numerous landmarks along the humanoid body, head and along the scenario, (d) a previously acquired 3D model of the world, (e) failure on poor textured scenarios (such as wooden and mosaic floors), (f) using more than one external sensor.

It is worth commenting on several aspects related with our approach. Although we have not included more than one robot on scene, the proposed depth-based tracking has the ability to cope with humanoid objects in the scene, since the generic geometric model employed in our approach considers the general shape of humanoid object point clouds from the top view. However, if the direct vision between camera and robot is interrupted or partly interrupted, the tracking performance will be affected. In this case, there will be a smaller amount of generated high weight observations (particles), since the target data provided in this frame is missing. The worst case is when the tracker takes the wrong data as its observation, i.e. taking the occluder as the target, which is a common problem in color-based tracking approaches. However, the proposed depth-based particle filter has robustness to occlusions to some extend. In our approach, the depth information and the geometric model would help the system to distinguish between the occluder and the target so that the observations in the depth-based tracker coincide with the real situation rather than observing the incorrect data. In this manner, the particle distribution before the target is occluded is better protected along the occluding frames.

Finally, the authors believe that the low cost of the RGB-D sensors make our methodology appropriate for installing a monitoring system for humanoids in closed spaces, where human-robot interaction in house environments is needed. Also, combining the capabilities of an external sensor based tracking system with an internal sensor such as the camera of the humanoid robot would increase the localization capabilities of the robot depending on the required task, i.e., walking towards a target could be performed through an external sensor while grasping, manipulating the target could be achieved with the internal or mounted on sensor.

6 Conclusions

A framework facing the active localization and tracking of a humanoid robot has been presented. Using a particle filter over the depth information obtained from an RGB-D sensor the humanoid robot pose (position and orientation) was estimated. The active feature in the localization is carried out by incorporating the pose of the robot into the locomotion control. The integral communication of the system is achieved using ROS, which facilitates applying the proposed framework in navigation tasks. The approach was tested using a Kinect and a NAO humanoid robot with promising results.

The geometric model used in our approach exploits the shape of humanoid objects from the top view in the tracking process and is suitable for improvement with robustness to partial occlusions. The orientation estimation is more sensitive to occlusions since the ellipse fitting method highly relies on the completeness of the point cloud data. Nevertheless, this can be addressed by introducing temporal consistency in the orientation estimation process. Particularly, in our implementation we smooth the estimated orientation along time when the target orientation changes greatly in consecutive frames.

It is worth commenting on the impact of our framework in the field of human-robot interaction, i.e., in terms of accurately localizing the robot for taking decisions in real time about navigating into a room where there are humans to approach. First, our tracking method may complement the applicability of planning algorithms when humans and obstacles appear. Second, our scheme could be easily incorporated within human tracking approaches in order to increase the range of possible interactions of the robot and the human. These ideas are now considered as an extension of the scopes described in this paper.

Acknowledgements This work has been partially developed in the framework of the project TEC2013-43935-R, financed by the Spanish Ministerio de Economía y Competitividad and the European Regional Development Fund (ERDF). Also, the authors would like to thank Mexican Council of Science and Technology (CONACYT) for the PhD studentship of Pablo A. Martínez and the financial support for the sabbatical leave of Mario Castelán.

References

1. Alcantarilla, P., Ni, K., Bergasa, L., Dellaert, F.: Visibility learning in largescale urban environment. In: IEEE International Conference on Robotics and Automation. Shanghai, China (2011)
2. Alcantarilla, P.F., Stasse, O., Druon, S., Bergasa, L.M., Dellaert, F.: How to localize humanoids with a single camera? Autonomous Robots **38**(1-2), 47–71 (2013)
3. Arechavaleta, G., Laumond, J.P., Hicheur, H., Berthoz, A.: An optimality principle governing human walking. IEEE Transactions on Robotics **24**(1) (2008)
4. Castelán, M., Arechavaleta, G.: Approximating the reachable space of human walking paths: a low dimensional linear approach. In: IEEE International Conference on Humanoid Robots, pp. 81–86. Paris, France (2009)
5. Chestnutt, J., Takaoka, Y., Suga, K., Nishiwaki, K., Kuffner, J., Kagami, S.: Biped navigation in rough environments using onboard sensing. In: IEEE International Conference on Intelligent Robots and Systems. St. Louis, MO, USA (2009)
6. Clark, R., Wang, S., Wen, H., Trigoni, N., Markham, A.: Increasing the efficiency of 6-DoF visual localization using multi-modal sensory data. In: Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on, pp. 973–980. IEEE (2016)
7. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(5), 564–577 (2003)
8. Comport, A.I., Marchand, E., Pressigout, M., Chaumette, F.: Real-time markerless tracking for augmented reality: The virtual visual servoing framework. IEEE Transactions on Visualization and Computer Graphics **12**(4), 615–628 (2006)
9. Davison, A., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 1052–1067 (2007)
10. Escande, A., Mansard, N., Wieber, P.B.: Hierarchical quadratic programming: Fat online humanoid-robot motion generation. International Journal of Robotics Research **33**(7), 1006–1028 (2014)
11. Fallon, M.F., Antone, M., Roy, N., Teller, S.: Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing. In: Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pp. 112–119. IEEE (2014)
12. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real time motion capture using a single time-of-flight camera. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 755–762 (2010)
13. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real-time human pose tracking from range data. In: Computer Vision ECCV–2012, pp. 738–751. Springer (2012)
14. Gordon, N.J., Salmond, D.J., Smith, A.F.: Novel approach to nonlinear/non-gaussian bayesian state estimation. In: IEE Proceedings F (Radar and Signal Processing), vol. 140, pp. 107–113. IET (1993)
15. Gratal, X., Smith, C., Björkman, M., Kragic, D.: Integrating 3d features and virtual visual servoing for hand-eye and humanoid robot pose estimation. In: IEEE/RAS International Conference on Humanoids Robots, pp. 240–245 (2013)
16. Hansen, D.W., Hansen, M.S., Kirschmeyer, M., Larsen, R., Silvestre, D.: Cluster tracking with time-of-flight cameras. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW'08, pp. 1–6 (2008)
17. Herdt, A., Diedam, H., Wieber, P.B., Dimitrov, D., Mombaur, K., Diehl, M.: Online walking motion generation with automatic footstep placement. Advanced Robotics **24**(5-6), 719–737 (2010)
18. Hornung, A., Wurm, K., Bennewitz, M.: Humanoid robot localization in complex indoor environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1690–1695. Taipei, Taiwan (2010)
19. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07). Nara, Japan (2007)
20. Maier, D., Hornung, A., Bennewitz, M.: Real-time navigation in 3D environments based on depth camera data. In: IEEE International Conference on Humanoid Robots, pp. 692–697. Osaka, Japan (2012)
21. Martínez, P.A., Castelán, M., Arechavaleta, G.: Vision based persistent localization of a humanoid robot for locomotion tasks. International Journal of Applied Mathematics and Computer Science **26**(3), 669 (2016)
22. Martínez, P.A., Varas, D., Castelán, M., Camacho, M., Marqués, F., Arechavaleta, G.: 3d shape reconstruction from a humanoid generated video sequence. In: Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pp. 699–706. IEEE (2014)
23. Matsumoto, Y., Wada, T., Nishio, S., Miyashita, T., Hagita, N.: Scalable and robust multi-people head tracking by combining distributed multiple sensors. Journal of Intelligent Service Robotics **3**(1), 29–36 (2010)
24. Michel, P., Chestnutt, J., Kagami, S., Nishiwaki, K., Kuffner, J., Kagami, S.: GPU-accelerated real-time 3d traking for humanoid locomotion and stair climbing. In: IEEE International Conference on Intelligent Robots and Systems. San Diego, USA (2007)
25. Michel, P., Chestnutt, J., Kagami, S., Nishiwaki, K., Kuffner, J., Kanade, T.: Online environment reconstruction for biped navigation. In: IEEE International Conference on Robotics and Automation. Orlando, FL, USA (2006)
26. Migniot, C., Ababsa, F.: 3D human tracking in a top view using depth information recorded by the xtion pro-live camera. In: Advances in Visual Computing, pp. 603–612. Springer (2013)
27. Nummiaro, K., Koller-Meier, E., Van Gool, L.: An adaptive color-based particle filter. Image and Vision Computing **21**(1), 99–110 (2002)
28. Obwald, S., Hornung, A., Bennewitz, M.: Improved proposals for highly accurate localization using range and vision data. In: IEEE/RSJ International Conference on Intelligent Robots and System. Vilamoura, Portugal (2012)
29. Oriolo, G., Paolillo, A., Rosa, L., Venditelli, M.: Humanoid odometric localization integrating kinematic, inertial and visual information. Autonomous Robots pp. 1–13 (2015)
30. Papadopoulos, A.V., Bascetta, L., Ferretti, G.: Generation of human walking paths. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1676–1681. Tokyo, Japan (2013)
31. Poppe, R.: Vision-based human motion analysis: An overview. Computer vision and image understanding **108**(1), 4–18 (2007)
32. Puydupin-Jamin, A.S., Johnson, M., Bretl, T.: A convex approach to inverse optimal control and its application to modeling human locomotion. In: IEEE International Conference on Robotics and Automation, pp. 531–536. Minnesota, USA (2012)
33. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
34. Rauter, M.: Reliable human detection and tracking in top-view depth images. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, pp. 529–534 (2013)
35. Siddiqui, M., Liao, W., Medioni, G.G.: Vision-based short range interaction between a personal service robot and a user. Intelligent Service Robotics **2**(3), 113–130 (2009)
36. Simon, D.: Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. Wiley-Interscience (2006)
37. Sisbot, E.A., Marin-Urias, L.F., Alami, R., Siméon, T.: A human aware mobile robot motion planner. IEEE Transactions on Robotics **23**(5), 874–883 (2007)

38. Sisbot, E.A., Marin-Urias, L.F., Broquère, X., Sidobre, D., Alami, R.: Synthesizing robot motions adapted to human presence. International Journal of Social Robotics **2**(3), 329–343 (2010)
39. Spinello, L., Arras, K.O.: People detection in RGB-D data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3838–3843 (2011)
40. Stasse, O., Davison, A., Sellaouti, R., Yokoi, K.: Real-time 3D SLAM for a humanoid robot considering pattern generator information. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 348–355. Beijing, China (2006)
41. Stilman, M., Nishiwaki, K., Kagami, S., Kuffner, J.: Planning and executing navigation among movable obstacles. In: IEEE International Conference on Intelligent Robots and Systems. Beijin, China (2006)
42. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment a modern synthesis. Vision Algorithms: Theory and Practice (1999)
43. Wurm, K.M., Hornung, A., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In: IEEE International Conference on Robotics and Automation, Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation. Anchorage, Alaska (2010)
44. Xia, L., Chen, C.C., Aggarwal, J.: Human detection using depth information by kinect. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, pp. 15–22 (2011)
45. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM computing surveys (CSUR) **38**(4), 13 (2006)