Int. J. Appl. Math. Comput. Sci., 2016, Vol. 26, No. 3, 669–682
DOI: 10.1515/amcs-2016-0046

VISION BASED PERSISTENT LOCALIZATION OF A HUMANOID ROBOT FOR LOCOMOTION TASKS

PABLO A. MARTÍNEZ [a,*], MARIO CASTELÁN [a], GUSTAVO ARECHAVALETA [a]

[a] Robotics and Advanced Manufacturing Group, Research Center for Advanced Studies
National Polytechnic Institute (CINVESTAV), Ramos Arizpe, Coahuila, 25900, Mexico
e-mail: {pablo.martinezglz,mario.castelan,garechav}@cinvestav.mx

Typical monocular localization schemes involve a search for matches between reprojected 3D world points and 2D image features in order to estimate the absolute scale transformation between the camera and the world. Successfully calculating such transformation implies the existence of a good number of 3D points uniformly distributed as reprojected pixels around the image plane. This paper presents a method to control the march of a humanoid robot towards directions that are favorable for visual based localization. To this end, orthogonal diagonalization is performed on the covariance matrices of both sets of 3D world points and their 2D image reprojections. Experiments with the NAO humanoid platform show that our method provides persistence of localization, as the robot tends to walk towards directions that are desirable for successful localization. Additional tests demonstrate how the proposed approach can be incorporated into a control scheme that considers reaching a target position.

Keywords: robot localization, monocular vision, humanoid locomotion.

1. Introduction

Robot localization constitutes a classical problem in robotics. There exist a body of methods that have been mainly developed for wheeled robots. Different sensors can be used, according to the task assigned to the robot, for perceiving the environment and the robot's internal state. Vision sensors have been widely used as systems to measure the incremental spatial displacements of the robot relative to a given inertial frame attached to the world (Royer *et al*., 2007). A variety of techniques in visual odometry have been suggested for stereo and monocular vision systems (Scaramuzza and Fraundorfer, 2011). However, in most scenarios, visual odometry is not sufficient to correctly estimate the pose of the robot. Commonly, bundle adjustment methods are simultaneously executed to apply sequences of local and global (if necessary) corrections. This process selects reliable image features to be incorporated to the sparse map representing the environment. Filtering-based methods also solve the problem as reported in the literature (Durrant-Whyte and Bailey, 2006).

Paradoxically, while the success of classical approaches is far from being directly extrapolated to walking machines, humanoid robots mainly rely on vision systems in order to perceive the environment and resemble human capabilities. In particular, monocular vision is preferred for small-sized humanoids that are certainly constrained to be equipped with lightweight, low cost and low-energy consumption devices. Additionally, humanoid robot localization while walking turns to be a complex problem due to discrepancies in time among sensor readings. Specifically, the orders of magnitude from the acquired frequency signals differ for each sensor and the rate of divergence from the walking reference trajectory is high for small distances.

A shared feature among visual based localization techniques is that they have been proposed to solve the localization problem regardless of the humanoid locomotion controller. In other words, the robot is asked to walk in accordance with a predefined control input, and during the motion execution the localization module estimates the pose of the robot (e.g., Stasse *et al*., 2006). As a result, the estimation process in these cases does not communicate with the humanoid walking module.

In addition to the common problems arising from the jerky camera movements because of the stepping

*Corresponding author

impacts and the blurring continuously appearing on the acquired images, it is important for the robot to maximize the probability to be localized in the near future while walking. This requirement translates to an active localization formulation where, at each step, the next control input for a given time horizon should consider visibility criteria to direct the humanoid walking. In this sense, an active topological localization strategy has been proposed by Ido *et al*. (2009) to compute the next action at a given time horizon based on a sequence of reference images. Recently, a similar strategy, suggested by Delfin *et al*. (2014), considers predefined reference images to guide the humanoid walking towards a target image by applying a sequence of continuous visual servo control laws. Unfortunately, qualitative localization does not suffice to fulfill requirements of applications where spatial localization is important.

In this context, our work proposes an active monocular localization method relying on meaningful visibility criteria that are used to control either the heading of the humanoid or its foot stepping direction while walking, in order to maintain it spatially well localized with respect to an absolute reference frame.

The main contributions of this article are as follows:

- a novel approach for locomotion tasks that includes an active persistent localization module based on visual cues,
- a set of statistical criteria for the analysis of the 3D map and reprojected 2D points that is useful for targeting the robot towards directions of rich visual information,
- a control scheme that considers reaching a target position while updating linear and angular velocities in accordance with visual criteria.

The paper is organized as follows. Section 2 provides an overview of the previous work related to this paper. Section 3 sketches the main problem of this research, motivating a scheme for persistent localization in terms of a simultaneous localization and mapping (SLAM) application. Section 4 describes how to select promising directions for walking, based on angles spanned by the covariance of the 3D world map. Similarly, Section 5 uses the covariance analysis of the 3D points reprojected onto image views in order to define the visibility criteria. Section 6 describes a control scheme that incorporates the persistent localization approach for the purposes of reaching a target position. Experimental results are depicted in Section 7, and finally conclusions are outlined in Section 8.

2. Related work

This section has been divided into three parts regarding the nature of the different visual based localization and SLAM methodologies. The first part describes work on fusing sensors by filtering, the second part analyses the influence of bundle adjustment in approaches that exploit 2D and 3D geometric relationships, and the third part discusses the recently introduced contributions on RGB-D SLAM techniques.

In the context of this paper, it is important to clarify the difference between the terms 'localization' and 'SLAM.' The former is related to strictly determining the current position of the camera with respect to an absolute reference frame, while the latter refers to building a 3D world map. For monocular localization schemes, however, this difference may become less sharp. It has been shown (Scaramuzza and Fraundorfer, 2011) how, for monocular schemes, the most feasible way for achieving localization in term of the scale of the physical world is by solving the perspective-from-n-points (PnP) problem. The solution for PnP implies reprojecting a set of 3D world points onto the camera plane in order to determine correspondence and finally recover motion. For this reason, monocular localization schemes are usually attached to a 3D map. In this sense, SLAM can be thought of as the task, while localization can be regarded as a tool for successfully reaching that task.

2.1. Filtering. Probabilistic methods, such as the extended Kalman filter (EKF), have been widely used in mobile robotics (Skrzypczyński, 2009). However, there is a relatively small number of SLAM techniques for humanoids. Monocular visual SLAM by means of the EKF method has been successfully applied on the human-sized HRP-2 robot building a map of sparse 3D points that allows localization on small indoor scenarios (Davison *et al*., 2007). This method is capable of reaching real-time performance; however, it requires sophisticated initialization techniques and data from sources other than vision systems such as proprioceptive sensors and walking pattern generators in order to obtain accurate motion estimation (Stasse *et al*., 2006).

Recently, the EKF method for humanoid localization has been tested on the NAO platform (Oriolo *et al*., 2016). In this case, the EKF fuses data obtained from the parallel tracking and mapping (PTAM) software (Klein and Murray, 2007) and the inertial unit attached to the robot. In particular, the localization and mapping problems are first solved by means of a real-time structure from motion (SfM) technique with PTAM, where the outcome is then used as a 3D visual sensor. The second estimation phase uses an EKF where the prediction stage is performed by differential kinematics to relate the torso and joint velocities. Then, the correction stage uses the

camera pose estimate from PTAM and inertial data to refine robot localization. An important aspect of the method is the use of the pressure sensors attached to the feet to activate the other sensor readings. This is consistent with the experiments presented by Ido *et al*. (2009) to overcome the acquisition of blurred images caused by the impacts between the feet and the ground. The undesired effects produced by the lateral movements are studied by Oriolo *et al*. (2013). The authors propose a vision-based feedback controller for trajectory tracking. However, the experiments do not clarify the importance of the inertial measurements to better localize the robot.

Other techniques such as Monte Carlo have also proved useful in global localization problems by fusing visual and range data (Obwald *et al*., 2012). Although filtering allows the integration of multiple sensors (Stasse *et al*., 2006), rich visual information must be available if an accurate odometry is sought (Hornung *et al*., 2010). Thus, visual cues become an important source for localizing a humanoid robot, and filter based approaches need to rely on a robust visual odometry system in order to remain successful.

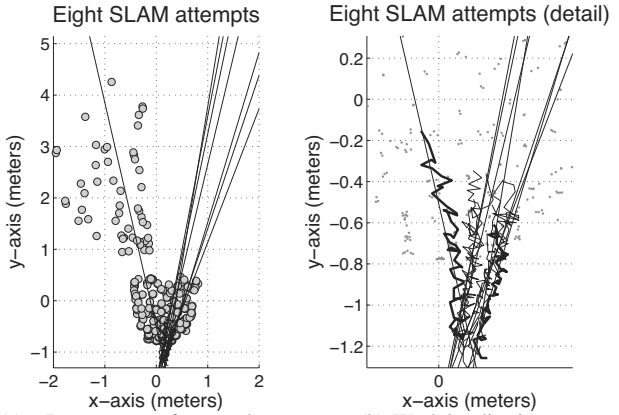
2.2. Image reprojections and bundle adjustment. According to Dellaert and Kaess (2006) as well as Strasdat *et al*. (2010), if accurate localization is required, bundle adjustment methods are more suitable than filtering because the latter is prone to linearization errors as well as unable to deal with a great number of features tracked between frames. Bundle adjustment has been widely used in computer vision (Triggs *et al*., 1999) as it iteratively adjusts camera poses and 3D points through an optimization strategy by minimizing the image reprojection error. Being non-linear, the optimization is usually formulated as a Levenberg–Marquardt problem (Hartley and Zisserman, 2004), nonetheless, its computational complexity of $\mathcal{O}(N^3)$ is an issue in real-time approaches. The incremental local bundle adjustment proposed by Mouragnon *et al*. (2009) reduces complexity using a windowed bundle adjustment over the last number of frames, i.e., optimizing only the last camera poses and the visible 3D points.

The linear systems that Levenberg–Marquardt needs to solve for bundle adjustment have a sparse block structure. In order to exploit this property for reducing the computational cost, a sparse bundle adjustment package was developed by Lourakis and Argyros (2009) as a library. Other platforms such as PTAM represent a more integral tool due to their ability to track hundreds of features, perform both local (incremental) and global bundle adjustments and grow the 3D map when new keyframes appear. These tasks may be computed in parallel resulting in real-time applications. For monocular localization, an initialization that simulates a stereo pair to approximate the depth of the initial 3D points is crucial to obtain feasible results.

Recently, a visual based localization approach that benefits from robust bundle adjustment was introduced by Alcantarilla *et al*. (2013). Here, a sparse 3D map is previously built using stereo visual SLAM, and a visual criterion is later incorporated into a monocular localization framework which predicts the visibility of 3D points. Unlike filtering based strategies, this method takes advantage of the geometric dependency between the 3D map and the camera poses. While this approach is inspired for solving PnP in a fast and robust way, it requires the knowledge of the 3D structure of the navigating space. One alternative could be using intersections between straight lines on the floor (Santana and Medeiros, 2012); unfortunately, not all working spaces exhibit linear patterns useful for robust localization.

2.3. RGB-D SLAM. The growing popularity of RGB-D sensors has allowed the recent development of what is now called RGB-D SLAM systems. The first of these schemes was proposed by Henry *et al*. (2012) as an attempt to pose the iterative closest point (ICP) algorithm (Segal *et al*., 2009) in terms of RGB-D platforms. In their work, visual odometry was solved using a cost function which linearly combined sparse 2D image features and 3D points. Recently, Kerl *et al*. (2013b) developed a probabilistic framework for RGB-D based visual odometry. Here, a photo-consistency error was measured between all pixels of consecutive images in order to compute the *a-posteriori* likelihood of the camera motion. The idea was extended with visual SLAM capabilities by Kerl *et al*. (2013a), who added a depth error to the cost function in order to achieve scene reconstruction and loop closure. More recently, Endres *et al*. (2014) approached a geometric solution including robust matching of visual features using the sensor input as landmark positions in order to compute the 3D-to-3D relations for camera motion estimation. Here, a beam-based environment measurement model is used to penalize unlikely pose estimates, and the octree-based mapping framework OctoMap (Hornung *et al*., 2013) is employed to represent the environment.

Although the above methods have proved successful in SLAM tasks, they have been tested over databases observing smooth transitions and approximately constant velocities such as a hand-held camera and a wheeled robot motion (Sturm *et al*., 2012). Unfortunately, as a humanoid robot march implies constant swinging, the risk of sudden changes in the motion of the camera may compromise the applicability of these approaches. Still, Maier *et al*. (2012) proposed an integrated navigation framework that considers localization, obstacle mapping and collision avoidance using an RGB-D camera mounted on top of the head of a NAO robot. To this end, an internal map is represented through an octree (Wurm *et al*., 2010),



(a) Departures from the planned path.

(b) Weak localization.

Fig. 1. Persistent localization. Results of eight attempts for a humanoid robot to walk on a straight line. Departures from the planned path are shown along the xy-plane in (a) as lines projected towards infinity. The concept of persistence of localization is illustrated in (b) as a close-up view of the eight trials.

while the pose of the robot is estimated using Monte Carlo localization based on depth information. Although an octree-based representation allows the successful construction of a dynamic map for real-time collision-free path planning tasks, an accurate initialization of the static world is required and, as a consequence, the previous process of dense 3D scanning and modeling of the navigating space becomes essential.

3. Persistent localization

In order to illustrate the main problem addressed in this paper, the following experiment was performed. The robot was given the task of walking 1 m along a straight line path. The initial orientation of the robot was approximately parallel to the y-axis. The head was locked to ensure a full alignment of the camera with body orientation. An initial map of the world was first computed from two camera views, approximately displaced 10 cm along the x-axis in order to simulate a stereo pair to approximate the depth of the initial 3D points. During the path, monocular camera localization was performed in a 3D-to-2D fashion using PTAM, while the locomotion control of the robot performed the given walking task, i.e., in a decoupled manner. This experiment was repeated eight times, positioning the robot at the same initial position for each trial, trying to preserve the same starting conditions in the experiment.

Figure 1 presents the results of the experiment. In the figure, 3D points in the world are displayed in (a) as circles projected onto the xy-plane. For each repetition, a line was fitted to the sequence of estimated camera positions in order to estimate the tendency of the robot to depart towards the left or the right side of the y-axis. The fitted lines corresponding to the eight trials reveal strong departure towards the right side with respect to the initial position and orientation of the robot. This is a consequence of one of the main problems present in biped robots while walking: foot slippage on the floor. A detailed view of the performed paths is provided in (b), where the 3D map is displayed as small points for clarity of visualization. The most and least successful paths, in terms of *persistence of localization*, are also highlighted in (b). In the context of this paper, the term 'persistence' indicates how long the robot is able to keep itself localized in the world, based on current visual information and a dynamically generated map of 3D points. In this sense, the most successful attempt is the one oriented towards the left side of the y-axis, while the worst one is probably the most biased towards the right. Although the effective length of march was approximately 1 m, only one trial proved persistently localized during the complete march, since the average length for the rest of the trials was 0.77 ± 0.11 m, indicating that the robot was no longer localized before completing the task.

The concept of persistence of localization is strongly related to the presence of reprojected 3D world points onto camera views originated during the robot march. The visual features are computed using the FAST detector (Rosten and Drummond, 2005), and are possible candidates to match the reprojections of the 3D world points to finally find the 2D-to-3D transformation required to localize the camera of the robot in terms of world coordinates. For both successful and failed attempts, the camera views at the initial steps of the robot have a relatively similar distribution of reprojected 3D points, from which it can be assumed that the robot was successfully localized in all attempts during its first few steps. However, once the robot stopped finding rich visual cues, it started to "get lost."

There are several conclusions to highlight from the analysis of Fig. 1. First, active control of the robot is of great importance if a previously planned path is to be guaranteed. There is certainly a limited utility in having a walking entity that is not able to acknowledge its place in the world. Second, even if the path is predefined and a visual based localization module is at hand, external factors such as poor visual features, the roughness of the floor and an existing bias in the march of the robot, to mention some, may prevent the robot from finding walking directions that are ideal for prevailing thorough localization along its march. Third, visual criteria need to be incorporated into a localization scheme that allows the robot to actively correct its orientation while seeking safe directions in order to keep localization persistent. Finally, there is a compromise between localization and the task,

and the robot should be able to decide whether to stop walking if localization is put at risk, even when the task is not fully completed.

To finish the current section, it is worth noting that assessing localization without considering feedback has been a common practice in the literature (Hornung *et al*., 2010; Maier *et al*., 2012; Obwald *et al*., 2012; Alcantarilla *et al*., 2013; Hornung *et al*., 2014). Specifically, a previously computed path is performed by the robot, where localization is not an active part for achieving the desired path. In this sense, localization and motion are usually decoupled, and this section has introduced the problem of not considering an active localization scheme during the navigation task.

4. Calculating promising directions

This section describes a method of selecting promising directions so as to ensure convenient conditions to achieve persistent localization. Here, the underlying idea consists of decomposing the points in the 3D map into principal components indicating directions where concentration of 3D points occur. It is important to recall that the error propagated on the mapped point features and the camera poses are minimized by the process of bundle adjustment (Mouragnon *et al*., 2009) as a key component of PTAM. This minimizes the effect of spacial uncertainty of the 3D map points and the calculated camera poses.

Let $\mathbf{X} = [\mathbf{x}_i, \mathbf{z}_i]$ be the $n \times 2$ matrix containing the Cartesian positions of the 3D map points, in camera coordinates, reprojected onto the xz-plane, i.e., the walking plane. Under this assumption, the 3D points need to be transformed from world coordinates into camera coordinates. That is to say, the world is aligned with z, the optical axis of the camera, and only those n points in front of the camera are considered for reprojection. In order to find convenient directions for localization, the following orthogonal diagonalization is applied:

$$\mathbf{X}^T\mathbf{X} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T, \tag{1}$$

where the matrix $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2]$ contains the two eigenvectors of the row space of $\mathbf{X}$ and $\mathbf{\Lambda}$ is a diagonal matrix with the two eigenvalues of $\mathbf{X}^T\mathbf{X}$.

We are only interested in the angle of the leading eigenvector $\mathbf{p}_i$ related to the greater eigenvalue, as it dictates the direction of greater variability of the 3D point cloud in front of the camera. This angle can be easily calculated as $\theta_m = \tan^{-1}(p_i(2)/p_i(1))$, and it will be used to divide the 2D point cloud into two subsets from which two new orientation angles, θ_l and θ_r, will be calculated. The idea is to have at least three promising directions calculated from the dispersion of the cloud points on the xz-plane: one central direction, one directed towards the left side, and one towards the right side of the main variation.

The left and right subsets of points can be estimated from the matrix $\mathbf{X}' = \mathbf{P}\mathbf{X}^T$. Geometrically, this multiplication means rotating the data in $\mathbf{X}$ around the axis defined by their principal variation. From the set of points stored in matrix $\mathbf{X}'$, the second row is used to form the Boolean vector $\mathbf{t}$, whose elements are defined as

$$t_i = \begin{cases} 0 & \text{if} \quad x'_{(i,2)} \leq 0, \\ 1 & \text{otherwise}, \end{cases} \tag{2}$$

where $x'_{(i,2)}$ is the second element of the i-th column of the rotated matrix $\mathbf{X}'$. The values in $\mathbf{t}$ are used to filter out the map points towards the left and the right side of the principal variation of all the points in the map. This operation can be done with the following pair of equations:

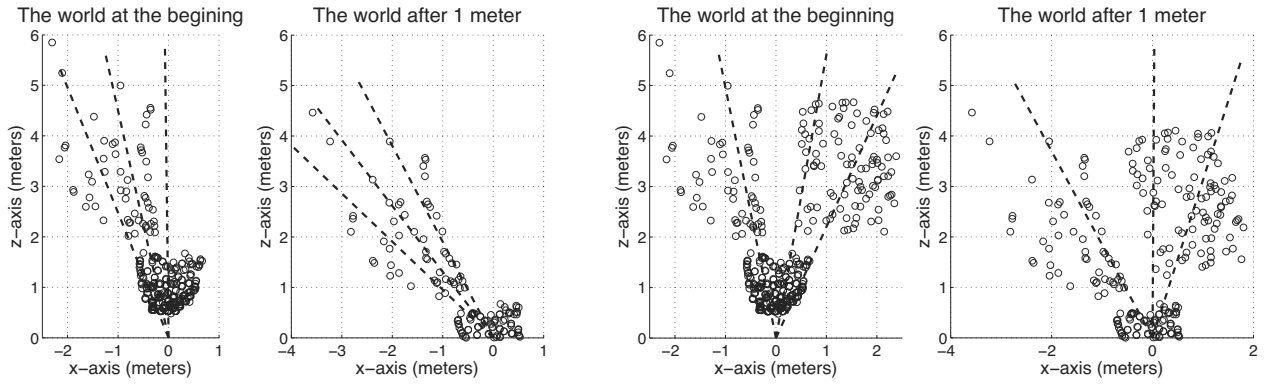
$$\mathbf{X}'_l = (\mathbf{I}\mathbf{t})\mathbf{X}'^T \quad \text{and} \quad \mathbf{X}'_r = (\mathbf{I}\neg\mathbf{t})\mathbf{X}'^T, \tag{3}$$

where $\mathbf{I}$ is the identity matrix and the symbol "$\neg$" stands for logical not. Once all zero rows have been filtered out from subsets $\mathbf{X}'_l$ and $\mathbf{X}'_r$, inverse rotation is required in order to define the map points into their original coordinate frame, i.e., $\mathbf{X}_l = \mathbf{P}^T(\mathbf{X}'_l)^T$ and $\mathbf{X}_r = \mathbf{P}^T(\mathbf{X}'_r)^T$. These two subsets are finally orthogonally decomposed so as to find their principal directions. This is done in a similar fashion as in Eqn. (1), i.e., $\mathbf{X}_l^T\mathbf{X}_l = \mathbf{P}_l\mathbf{\Lambda}_l\mathbf{P}_l^T$ and $\mathbf{X}_r^T\mathbf{X}_r = \mathbf{P}_r\mathbf{\Lambda}_r\mathbf{P}_r^T$. From these two factorizations, the leading eigenvectors can be taken from $\mathbf{P}_l$ and $\mathbf{P}_r$ to respectively find the leading orientation angles θ_l and θ_r.

Figure 2 depicts the main idea of this process. The figure is divided into two scenarios. Each one describes two situations: the robot at the beginning of the march and the robot after having approximately walked 1 m. The pair of camera localizations used in this figure were taken from the most successful march shown in Fig. 1. For the purposes of illustrating the horizon in front of the robot, the projected 3D maps are expressed in camera coordinates, rather than in world coordinates.

Scenario 1 is shown in (a) and presents the same 3D map as that used in Fig. 1. The three main directions are shown with dashed lines. Note how, for both situations of Scenario 1, the three promising directions lead to walking orientations towards the most concentrated number of 3D points of the world map, i.e., discarding any orientations related to the right side of the y-axis.

For Scenario 2, shown in (b), 100 random points were synthetically generated towards the right side of the map. It is noticeable from this scenario that the richness in 3D information determines new orientations considered promising for the purposes of localization. This observation motivates the introduction of visibility criteria that can be applied in order to select the most promising orientation angle among the three extracted from the 3D map. These criteria are in fact related to the



(a) Scenario 1. (b) Scenario 2.

Fig. 2. World in front of the camera. The coordinates of the figure are expressed in terms of the camera. The figure is split into two scenarios. Each scenario describes two situations: the robot at the beginning of the march and the robot after having approximately walked 1 m. The pair or camera localizations used in this figure were taken from the most successful march shown in Fig. 1. Random points were synthetically generated towards the right side of the map for Scenario 2. The dashed lines depict the three possible orientations $(\theta_l, \theta_m, \theta_r)$.

2D reprojections of the map points onto the camera image plane and are explained in depth in the next section.

As a final note, recall that the selection of promising directions is calculated from a statistical analysis of the 3D map points and it is therefore not deterministic, which leads to two main observations. On the one hand, the principal direction cannot simply be considered the most suitable for keeping the robot localized. On the other hand, incrementing the number of promising directions would generate oversampling and the sampled directions would become statistically meaningless.

5. Visibility criteria: Where to go

Once the possible directions are calculated, we search for future horizons by synthetically generating three camera views with directions θ_m, θ_l and θ_r, and a predefined translation expressed in world units. This is done by calculating the camera projection matrix, which is obtained as

$$\mathbf{P}_{\text{cam}} = \mathbf{K}[{}^c\mathbf{R}_w|{}^c\mathbf{t}_w], \tag{4}$$

where $\mathbf{K}$ is the matrix of intrinsic camera parameters, ${}^c\mathbf{R}_w$ is the rotation matrix and ${}^c\mathbf{t}_w$ is the translation vector, both expressed from the camera axis to the world axis. The idea underlying the synthetic generation of future horizons is to rely on actual reprojections of the 3D map points on the image plane, according to Eqn. (4), in order to determine whether a direction is actually promising for prevailing thorough localization. It is important to note that only those points reprojected within the limits of the image size are taken into account for calculations.

We proposed three visual criteria to be combined into a weighted optimization scheme in order to provide the goodness of fit for each possible direction. The first criterion is related to the *eccentricity* of the reprojected points. Let the pair $(\bar{u},\bar{v})$ be the mean values for all the reprojected points located at pixel positions (u, v) along the x and y axis of the image, respectively. The eccentricity of a view is defined as the scalar

$$e = \sqrt{(u_c - \bar{u})^2 + (v_c - \bar{v})^2}, \tag{5}$$

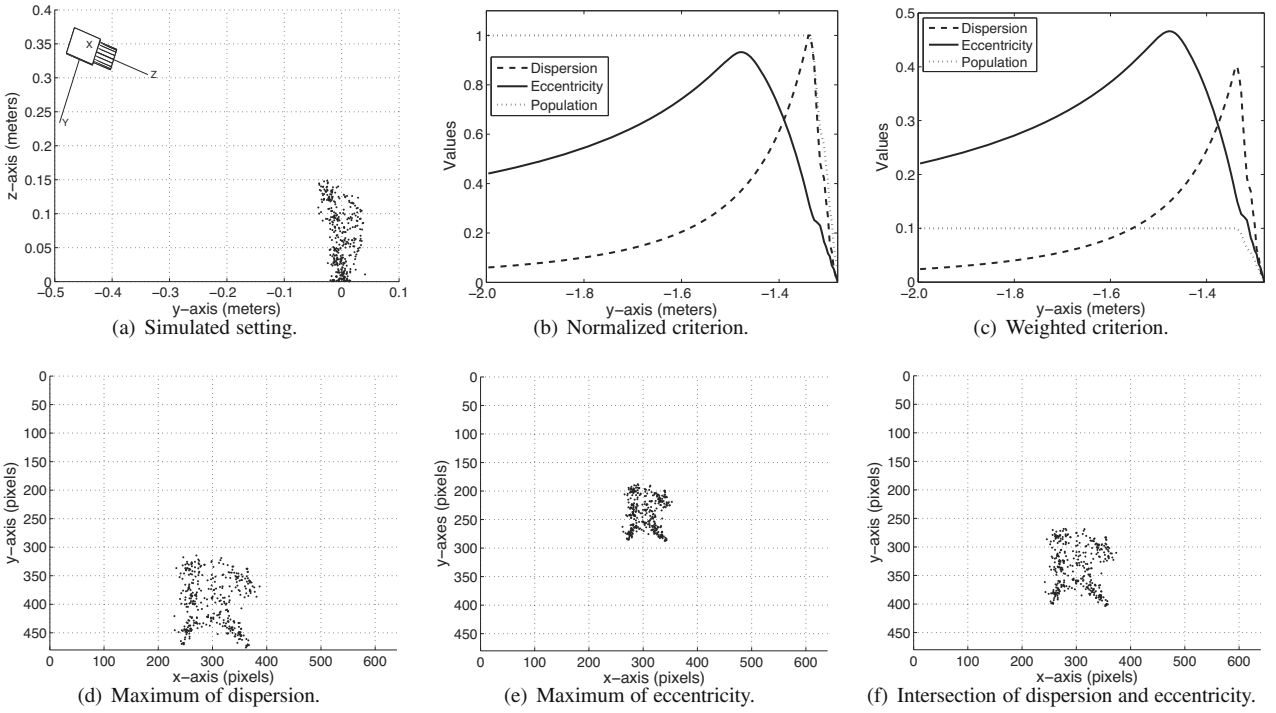
where the image center has coordinates (u_c, v_c). The eccentricity criterion is intended to explain the displacement of the center of mass of the set of all 3D points in sight from the center of the image.

The second criterion is related to the *dispersion* of the reprojected points around the image plane. The greater the dispersion, the greater the probability that a point in 3D will have a match within a set of detected image features. In other words, it is desirable that points in the 3D world map are reprojected along all directions in the image plane.

In order to calculate a degree of dispersion, the product of the eigenvalues $\lambda_1\lambda_2$ of the symmetric matrix $\mathbf{U}^T\mathbf{U}$ is calculated through its determinant. The matrix $\mathbf{U}$, containing the set of visible reprojected points in the image, has been previously centered about the mean values $(\bar{u}, \bar{v})$, and its two eigenvalues span a quadratic form whose area can be related to the dispersion of the set. The proposed measure of dispersion is defined as the proportion of the area of this quadratic form with respect to the total number of pixels as

$$d = \frac{\pi\sqrt{\det(\mathbf{U}^T\mathbf{U})}}{4u_c v_c}. \tag{6}$$

The third and final criterion, referred to as the *population*, is simply the number of reprojected 3D points



(a) Simulated setting. (b) Normalized criterion. (c) Weighted criterion.

(d) Maximum of dispersion. (e) Maximum of eccentricity. (f) Intersection of dispersion and eccentricity.

Fig. 3. Selecting weights of criteria. The figure shows in (a) the simulated setting for the robot's virtual camera approaching a cloud of points, in (b) the normalized criteria registered as a function of the distance from the camera to a cloud of points, in (c) the behavior of criteria after having being weighted by the selected values. Views of the projected points for maximum values of dispersion and eccentricity are respectively shown in (d) and (e), while the image corresponding to the intersection point between these two is depicted in (f).

for each of the three generated images, as views with a greater number of reprojected points from the 3D map are preferred over those with a fewer number of 3D point reprojections. The scalar p will be used to represent this criterion.

Finally, the three visibility criteria e, d and p are used under a weighted maximization scheme in order to assign a goodness of fit to each of the promising directions calculated in the previous step. The goodness of fit is defined by the vector $\mathbf{f} = \mathbf{C}\mathbf{w}$ described by

$$\begin{bmatrix} f_l \\ f_m \\ f_r \end{bmatrix} = \begin{bmatrix} e_l & d_l & p_l \\ e_m & d_m & p_m \\ e_r & d_r & p_r \end{bmatrix} \begin{bmatrix} w_e \\ w_d \\ w_p \end{bmatrix}, \tag{7}$$

where $\mathbf{C}$ is the visibility matrix containing the eccentricity, dispersion and population criteria evaluated over the three directions $(\theta_l, \theta_m, \theta_r)$ and the columns of $\mathbf{C}$ are normalized to length one. The vector $\mathbf{w}$ weighs the contribution of each visibility criteria. The weights w_e, w_d and w_p affect the eccentricity, dispersion and population, respectively. The best orientation angle is finally chosen in accordance with the maximum goodness of fit from the triplet (f_l, f_m, f_r).

5.1. Selecting the weights. The weighing values are selected in an experimental way. Specifically, we developed a simulation that allowed us to evaluate the behavior of each criteria as a function of the distance between a virtual camera and a cloud of points corresponding to an object of interest. This cloud of points was previously acquired using the monocular vision-based locomotion control proposed by Martínez *et al*. (2014), which may be suitable for the purposes of persistent localization applications. Roughly, the simulation experiment is as follows: the camera is located at $2\,\mathrm{m}$ apart from the object and is moved in a forward direction every $1\,\mathrm{cm}$ until the camera is at a distance of $20\,\mathrm{cm}$ from the object's centroid. A lateral view of the described setting is shown in Fig. 3(a).

Using the intrinsic parameters and the virtual poses of the camera (180 for this experiments), the reprojected 3D points are estimated at all positions and the corresponding values of dispersion, eccentricity and population computed over each virtual image. The behavior of the criteria is shown in Fig. 3(b) for normalized (unweighed) values. The diagram reveals that the population does not appear to vary with time and that it starts decreasing once the dispersion criterion has reached its maximum value. Note how eccentricity

presents a rather continuous descent in comparison with dispersion, while its peak value is also reached sooner. The virtual images of reprojected points corresponding to the maximum values of dispersion and eccentricity are respectively shown in Figs. 3(d) and (e), and the intersection between these two criteria is depicted in Fig. 3(f). From these diagrams, it is to note how the maximum dispersion value risks both criteria of population and eccentricity. For this reason, the selected weight values are aimed at avoiding those cases where a high concentration of reprojected points appear around a relatively small image region. In other words, eccentricity and dispersion must be granted greater importance than the population to favor views that contain a good number of reprojected points spread around the image. Additionally, a normal distribution of reprojected points may lead to an unbiased 3D-to-2D sampling, thus supporting the numerical stability of the direct linear transformation (DLT) method commonly applied when solving the PnP problem in monocular visual odometry. Considering the above facts, we have decided using the following weights in the rest of our experiments: 0.5 for eccentricity, 0.4 for dispersion and 0.1 for population.

5.2. Simulation example. Figure 4 presents a visual analysis of the proposed criteria. The left and right panels of the figure illustrate Scenarios 1 and 2 explained in the previous section, respectively. For both the scenarios, the cases labeled "the world at the beginning" in Fig. 2 are shown. The aim of Fig. 4 is to illustrate the different views the robot would have faced if it had translated 1 m from its initial position, but keeping the three most promising directions for localization, i.e., those provided by the principal directions of the 3D map world ahead of the robot. In this sense, the candidate views provide visibility information related to future localizations of the robot. In the figure, eccentricity is shown as a dashed line between the center of the image and the mean point, while dispersion is illustrated as the ellipse surrounding the reprojected 3D points. It is important to notice that the ellipse only appears partially because, for the cases shown in the figure, it is bigger than the image size. As far as population is concerned, this criterion can be perceived as the amount of reprojected 3D points appearing along each synthetic view.

In order to complement the visual analysis of Fig. 4, Table 1 presents the quantitative results obtained from each visibility criterion. The maximum values for the three criteria are highlighted in both scenarios. Interestingly, the table reveals that Scenario 1 has little trouble to determine the middle orientation as the best fit for future localizations. Nonetheless, the complexity of Scenario 2 shows the importance of eccentricity in the goodness of fit, since the left orientation is chosen as the best over a (nearly as good) right orientation. Note how

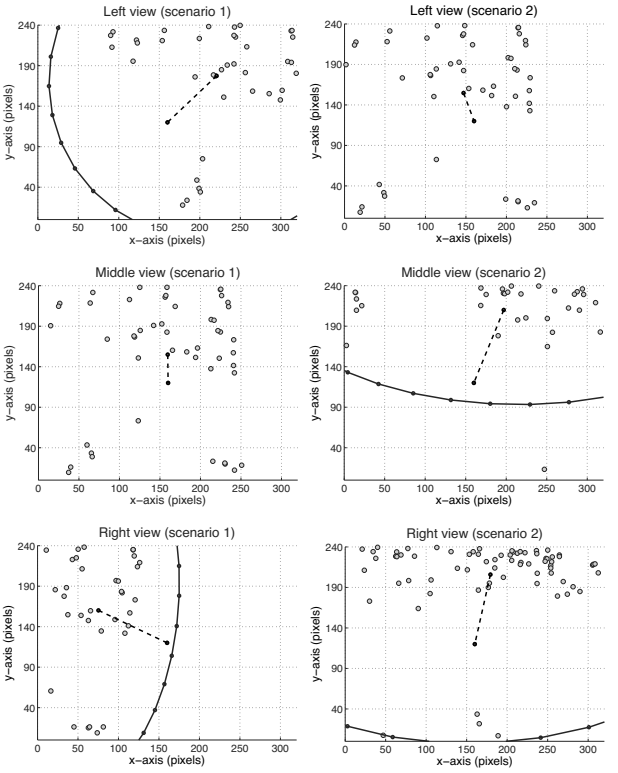


Fig. 4. Future in sight. Views of reprojected 3D map points are presented for Scenarios 1 (left panels) and 2 (right panels) of Fig. 2. The left, middle and right orientation views are depicted row-wise. Eccentricity is shown as a dashed line between the center of the image and the mean point. Dispersion is illustrated as the ellipse surrounding the reprojected 3D points. The number of points represent population.

the weights used to evaluate the goodness of fit benefit the eccentricity over the dispersion due to the sensitivity of the covariance matrix to atypical values. An example of this phenomenon can be observed in the right view of Scenario 2 in Fig. 4, where three outliers appear at the bottom of the image.

It is important to note that the proposed weighing supports the overall quality of the tracking points in each frame. According to Klein and Murray (2007), the computation of this quality is based on the fraction of feature observations which have been successfully corresponded. In this sense, the greater the corresponding 3D-to-2D features, the greater the possibility of incorporating more points into the 3D world map and thus keeping persistence of localization. For virtual views, however, it is impossible to determine the fraction of points that will be successfully corresponding: therefore the population criterion, on its own, does not provide a guarantee for safely localizing the robot. For this reason, the proposed weighing scheme has been aimed at keeping a greater number of evenly spread points

Table 1. Results on predicted views. Calculated values for eccentricity, dispersion and population are shown for the two scenarios depicted in Fig. 2.

Scenario	Eccentricity	Dispersion	Population	Goodness
1: Left	162	1.71	39	0.51
1: Middle	**245**	**2.67**	**49**	**0.76**
1: Right	156	0.84	34	0.38
2: Left	**233**	2.69	49	**0.69**
2: Middle	154	1.44	32	0.39
2: Right	175	**2.93**	**71**	0.62

(dispersion) in sight (eccentricity).

6. Footstep generation to reach a target position

Once the final orientation angle θ has been selected that maximizes the goodness of fit, the next goal is to modify the march of the robot towards convenient directions. As θ is given in terms of the x-axis of the camera, it should be expressed with respect to the principal camera axis z, i.e., $\theta_d = \pi/2 - \theta$. Note that the forward movement of the robot will always happen along the z-camera axis, as the head of the robot has been locked to be fully aligned with its body. In this sense, the relationship between the angular velocity ω and the desired θ is only up to a scaling factor; in other words, the computation of the angular velocity ω is derived by regulating the error function $e = \theta_d$. Typically, the convergence of the error is given by an exponential decrease of the form $\dot{e} = -\lambda e$ with λ as a constant gain. This implies that $\omega = \dot{e}$.

In this section we introduce the integration of our persistent localization scheme into a goal-oriented locomotion task, i.e., a scenario where a humanoid robot is required to walk from a source position and orientation to a target position on flat terrain within an open space. First, we briefly recall the main ingredients of a walking pattern generator (WPG), then we describe how our localization scheme provides, at each instant of time, the necessary input data for the reactive WPG to automatically solve the footstep placements, i.e., feasible positions and orientations of the feet to perform the next step while maintaining the dynamic balance of the robot.

6.1. Walking pattern generation. In the work of Kajita *et al*. (2003) the cart-table model is introduced to capture the main dynamic effects of a biped robot in terms of a linearized system of the zero moment point (ZMP). The input of the problem is the reference trajectory of the ZMP deduced from the predefined footsteps and the outcome should be the corresponding center of mass (CoM) trajectory. To find a solution, the authors proposed to apply a linear quadratic regulator using predicted information within a time window. In this case, the discretized version of the simplified dynamical system is of the form

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k), \\ p(k) &= \mathbf{c}\mathbf{x}(k), \end{aligned}$$

and the involved variables are defined as

$$\begin{aligned} \mathbf{x}(k) &= \left[x(kT)\ \dot{x}(kT)\ \ddot{x}(kT)\right]^T, \\ u(k) &= u_x(kT), \\ p(k) &= p_x(kT), \\ \mathbf{A} &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & T \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, \\ \mathbf{c} &= \begin{bmatrix} 1 & 0 & -h/g \end{bmatrix}, \end{aligned}$$

where T is the sampling period and x stands for the CoM position towards the x-axis (forward motion) since the analysis of the lateral motion (y-axis) is identical. The jerk of the CoM is represented through $u_x = \dddot{x}$, while p_x describes the position of the ZMP. The height above the ground and the norm of the gravity force correspond to the variables h and g, respectively. Equation (8) can be seen as a set of linear equality constraints to be satisfied. Thus, the problem can be written as a quadratic program (QP) where the jerk of the CoM is minimized together with the difference between the current position of the ZMP p and its reference p_r such that

$$\min_{\mathbf{u}(k)} \frac{\alpha}{2} \left\|\mathbf{u}(k)\right\|^2 + \frac{\gamma}{2} \left\|\mathbf{p}(k+1) - \mathbf{p}_r(k+1)\right\|^2. \quad (8)$$

The dynamics are computed recursively over the time interval of length NT:

$$\mathbf{p}(k+1) = \left[p(k+1)\ \ldots\ p(k+N)\right]^T, \quad (9)$$

with $\mathbf{u}(k) = \left[u(k), \ldots, u(k+N-1)\right]^T$, where N represents the future steps with respect to a determined sample k (i.e., the discretized time window).

This method has been successfully implemented in the NAO humanoid robot as its core WPG (Gouaillier *et al*., 2010). Note how the predefined foot step placements need to be provided by a foot step planner in order to extract the reference trajectory of the ZMP. Herdt *et al*. (2010) cope with this problem by regulating the velocity of the CoM to a desired mean value $\dot{\mathbf{x}}_\mathbf{r}$. Considering the current position of the foot on the ground, $\mathbf{x}_c(k)$, the positions of the following steps $\mathbf{x}_f(k)$ are adapted automatically using the selection matrices $\mathbf{S}_c(k+$

1) and $\mathbf{S}(k+1)$. The QP is then rewritten as

$$\min_{\bar{\mathbf{u}}(k)} \frac{\alpha}{2} \left\|\bar{\mathbf{u}}(k)\right\|^2 + \frac{\beta}{2} \left\|\dot{\bar{\mathbf{x}}}(k+1) - \dot{\bar{\mathbf{x}}}_r(k+1)\right\|^2 + \frac{\gamma}{2} \left\|\mathbf{p}(k+1) - \mathbf{p}_r(k+1)\right\|^2, \quad (10)$$

where $\dot{\bar{\mathbf{x}}}(k+1) = [\dot{\mathbf{x}}(k+1) \ldots \dot{\mathbf{x}}(k+N)]^T$ is the velocity of the CoM while $\dot{\bar{\mathbf{x}}}_r$ represents its reference velocity, $\bar{\mathbf{u}}(k) = \left[\mathbf{u}^T(k),\, \mathbf{x}_f^T(k)\right]^T$, α, β and γ constitute weighing parameters for the jerk, the CoM velocity error and the ZMP position error, respectively, and $\mathbf{p}_r(k+1) = \mathbf{S}_c(k+1)\mathbf{x}_c(k) + \mathbf{S}(k+1)\mathbf{x}_f(k)$.

A great advantage of this formulation is the possibility to incorporate linear equality and inequality constraints at will. In particular, to complete the above formulation, it is necessary to define the geometrical limits for the foot step placements by considering a polygonal area to be coherent with joint limits, self collision avoidance, etc. Therefore, the position of the ZMP should be constrained to remain within the polygonal area represented by linear inequalities. However, it is important to note that the set of inequalities depends on the orientation of the foot step. Thus, they are linear with respect to the position of the foot step but nonlinear with respect to its orientation. As a consequence, there is no feasible way to solve QP for reorienting the foot steps (Herdt *et al.*, 2010).

Algorithm 1. Target-driven framework. A current 3D map, a target position given in world units and the localization of the camera of the robot at time t are required.

Require: 3D map, localization at time t, target position $\mathbf{x}_t$.
1: initialization
2: **while** reprojected 3D points $\geq$ threshold **and** $\mathbf{x}_{CoM} \neq \mathbf{x}_t$ **do**
3: $\dot{\mathbf{x}}_r = -\lambda(\mathbf{x}_{CoM} - \mathbf{x}_t)$
4: horizon $h = \int \dot{\mathbf{x}}_r dt$
5: Calculate promising directions $[\theta_l, \theta_m, \theta_r]$
6: f_l = evaluate (θ_l,h)
7: f_m = evaluate (θ_m,h)
8: f_r = evaluate (θ_r,h)
9: **Apply** the reactive WPG with $(\dot{\mathbf{x}}_r, \max(f_l, f_m, f_r))$
10: **end while**

6.2. Algorithm outline. According to the previous discussion, the proposed persistent localization plays an important role in automatically reorienting the next foot steps to favor successful localization, while the reference velocity of the CoM $\dot{\mathbf{x}}_r$ is computed considering a proportional control based on the distance between the current estimate of the robot's CoM position $\mathbf{x}_{CoM} = [x\ y]^T$ and a given target position $\mathbf{x}_t = [x_t\ y_t]^T$. Therefore, the error $\mathbf{e} = \mathbf{x}_{CoM} - \mathbf{x}_t$ is regulated by imposing an exponential convergence $\dot{\mathbf{e}} = -\lambda\mathbf{e}$, where λ is a constant proportional gain. The control scheme considers a variable horizon h by integrating the reference velocity within a known fixed time. As a result, the humanoid is able to perform the locomotion task for reaching a target position while navigating along promising orientations to maximize the success of its localization in indoor environments. The process is described in Algorithm 1, where the input data provided by PTAM are the localization at time t and the 3D map points. This does not compromise the technical operation of PTAM, since the proposed algorithm only uses the information as an input, without performing further changes on its data structures. Algorithm 1 might be regarded as an application example of persistent localization for a specific navigation task. A graphical example of the application of Algorithm 1 for a given task is additionally depicted in Fig. 5.

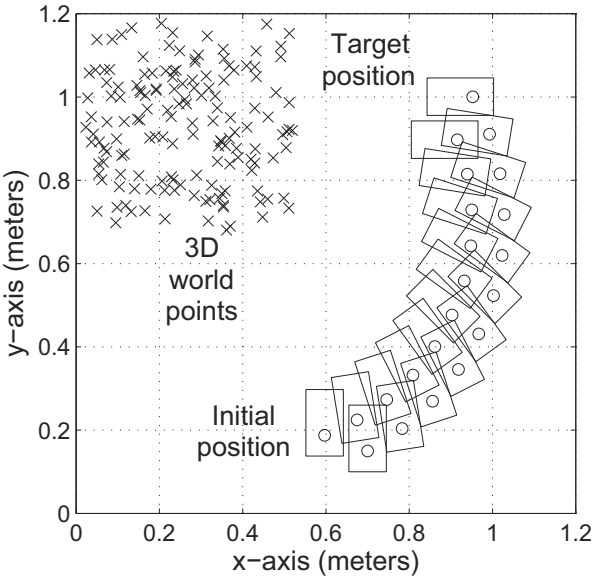
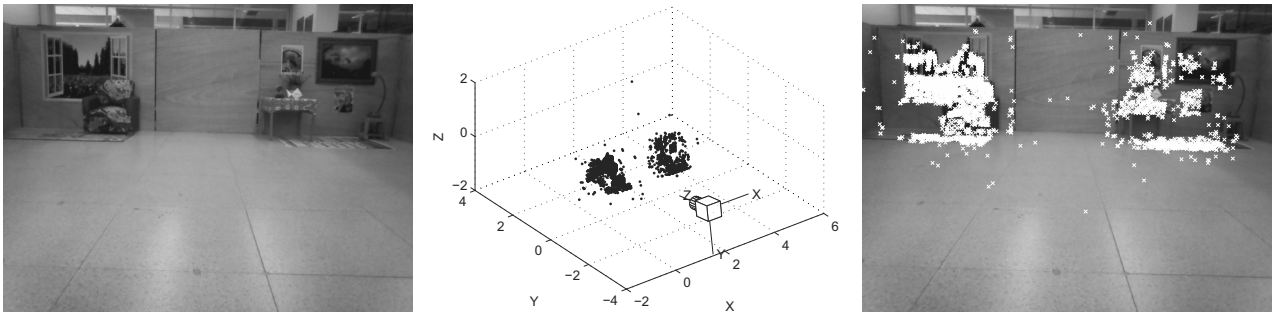


Fig. 5. Graphical example of a given task. A target position is required as an input for Algorithm 1, which is used for controlling the locomotion of the robot in order to reach a target position. Due to persistent localization, the robot is capable of remaining safely localized while keeping oriented towards sources of rich visual information.

7. Results

This section presents experimental evaluation after incorporating the persistent localization scheme on the NAO humanoid robot. We show experimental evaluation when the control depicted in Algorithm 1 is applied for two tasks: reaching a fixed target position and tracking



(a) Setting. (b) 3D map points. (c) 3D map points reprojected.

Fig. 6. Experimental setting. In (a) the setting used to evaluate the performance of the persistent localization scheme on the NAO humanoid robot is shown. In (b) the extracted and mapped points features are depicted as black dots while a camera drawing represents the pose of the robot camera. In (c) the reprojected 3D map points over the image plane of (a) are depicted as white crosses overlaid over the image.

a predefined path, i.e., a time-variant target. Figure 6(a) shows the setting used to evaluate the performance of the persistent localization scheme on the NAO humanoid robot, the image was captured with the camera mounted on the robot. In Fig. 6(b) the extracted and mapped points features are depicted as black dots while a camera drawing represents the pose of the robot camera. The reprojected 3D map points over the image plane of (a) are depicted as white crosses in (c). The setting has been designed to have two main clusters of points in order to test the algorithm in the presence of a bias. For each experiment the robot is initialized looking at either the right or the left cluster. This is due to PTAM limitations, as the camera needs to be no less than $1\,\mathrm{m}$ away from the observed scene in order to achieve good estimation of the initial 3D map.

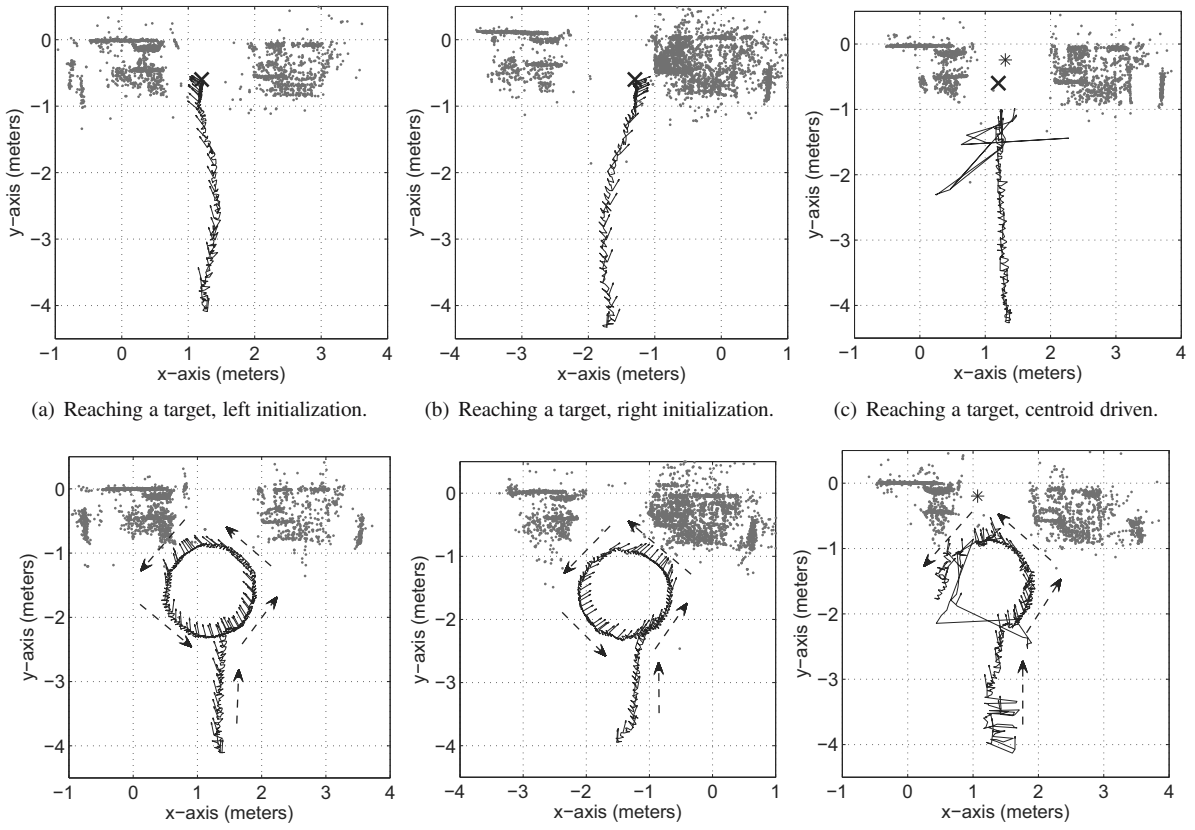
Once the robot has been initialized, it is moved $4\,\mathrm{m}$ away from the wall, approximately in the middle of the two point clusters. The results are depicted in the first row of Fig. 7, where (a) shows the left side initialization trajectory performed by the robot to reach a cross-marked target; similarly, results related to the right side initialization are shown in (b). The black arrows indicate samples of the robot orientation (with respect to the x-axis) at a particular position. For comparison, the centroid of the 3D points projected on the xy-plane is used to control the orientation of the robot. This is to compare the effect of persistent localization with the information provided by a fixed statistical parameter commonly used to describe central tendency in sets of points. The outcome of using the centroid (depicted with an asterisk) is shown in (c). Here, it is important to note that the robot lost the localization when, in the field of view of the robot camera, there were not enough points to maintain the robot localized. Note also how the coordinate axis varies through the different experiments, due to the varying initialization recorded by PTAM for each experiment.

It is possible to apply the proposed approach if the target is time-variant, i.e., for the purposes of path tracking. The experiment is designed to follow a circular path. Here, the immediate target to be reached is the point along the circumference corresponding to the next $3\,\mathrm{degrees}$ from the current position of the robot. The left and right side initialization cases are respectively shown in Figs. 7(d) and (e), while the result of tracking by keeping an orientation led by the centroid is shown in (f).

Let us start the analysis of Fig. 7 with the forward task. From the visual analysis of (a) and (b) it is clear how the task was performed successfully for both initializations towards the left and the right side of the scenario. Note how the orientations at the end of the march appear biased towards the placement of the visual features, corroborating that orientation is driven by visual data convenient for persistent localization. From (a) and (b) it can be seen that the robot approached both targets with similar accuracy. As far as the circular path tracking tasks are concerned, in (d) and (e), the orientations of the robot during the trials appear more pronounced than in the forward tasks. This is because the robot needed to keep the visual features in sight so as to remain successfully localized. As the trials where a centroid was used to control the orientation of the robot (i.e., without persistent localization), it is clear from the diagrams shown in (c) and (f) that the generated paths did not benefit from healthy visual based localization and the visual information disappeared from camera views, causing the robot to fail any localization attempt. The video showing the results of our experiments is available at `https://sites.google.com/site/gustavoarechavaleta/vlochum`.

8. Conclusions

A method aimed at incorporating persistence in visual based localization schemes has been proposed. The main idea of the method is to update the angular velocity of



(a) Reaching a target, left initialization. (b) Reaching a target, right initialization. (c) Reaching a target, centroid driven.

(d) Path tracking, left initialization. (e) Path tracking, right initialization. (f) Path tracking, centroid driven.

Fig. 7. Experiments. Examples of navigation tasks are shown when our method is applied in (a), (b), (d) and (e). Results of using only a locomotion controller and a statistical centroid (marked with an asterisk) for controlling orientation are presented in (c) and (f). The cross indicates the location of the targets to be reached while the black arrows represent samples of the robot orientation (with respect to the x-axis) at every 20 positions. The dotted line arrows illustrate the direction of the robot trajectory. The gray dots represent the 3D points from the final map generated during the experiment plotted on the xy-plane.

a humanoid robot using directions provided by both the current estimated 3D map and its future reprojections on the image plane. Experiments have demonstrated that our algorithm actively corrects the orientation of the robot while preserving camera views that benefit localization. Given its simplicity, a target-driven framework that uses persistent localization has been additionally formulated, and experiments demonstrate that the robot is able to perform absolute localization in order to reach a desired target. It is worth mentioning that the ultimate effect of persistent localization is focused on the orientation of the robot, which is pulled towards visually rich regions, thus attempting to preserve numerically stable 3D-to-2D monocular visual odometry.

In future work it would be interesting to investigate the outcome of our method for more complex navigation tasks, i.e., following a sequence of targets to perform SLAM. Also, It could be worth including a term that uses 3D information to detect unexplored regions that represent good candidates for incrementing the current 3D map. In addition, incorporating strategies to improve the robustness of PCA may deal with its sensitivity to outliers in the 3D points distribution. Finally, incorporating the persistence localization approach in goal-oriented locomotion tasks considering that the orientation of the head is not locked with the body may lead to applications with a strong focus on visual attention, and can also be of interest in obstacle avoidance path planning.

References

Alcantarilla, P.-F., Stasse, O., Druon, S., Bergasa, L.-M. and Dellaert, F. (2013). How to localize humanoids with a single camera?, *Autonomous Robots* **38**(1–2): 47–71.

Davison, A., Reid, I.-D., Molton, N.-D. and Stasse, O. (2007). Monoslam: Real-time single camera SLAM, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6): 1052–1067.

Delfin, J., Becerra, H.M. and Arechavaleta, G. (2014). Visual path following using a sequence of target images and smooth robot velocities for humanoid navigation, *IEEE-*

RAS International Conference on Humanoid Robots, Madrid, Spain, pp. 354–359.

Dellaert, F. and Kaess, M. (2006). Square root SLAM: Simultaneous localization and mapping via square root information smoothing, *International Journal of Robotics Research* **25**(12): 1181–1203.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: Part I, *IEEE Robotics and Automation Magazine* **13**(2): 99–110.

Endres, F., Hess, J., Sturm, J., Cremers, D. and Burgard, W. (2014). 3-D mapping with an RGB-D camera, *IEEE Transactions on Robotics* **30**(1): 177–187.

Gouaillier, D., Collette, C. and Kilner, C. (2010). Omni-directional closed-loop walk for NAO, *IEEE-RAS International Conference on Humanoid Robots, Nashville, TN, USA*, pp. 448–454.

Hartley, R.I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*, Cambridge University Press, New York, NY.

Henry, P., Krainin, M., Herbst, E., Ren, X. and Fox, D. (2012). RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments, *International Journal of Robotics Research* **31**(5): 647–663.

Herdt, A., Holger, D., Wieber, P.-B., Dimitrov, D., Mombaur, K. and Moritz, D. (2010). Online walking motion generation with automatic foot step placement, *Advanced Robotics* **24**(5–6): 719–737.

Hornung, A., Osswald, S., Maier, D. and Bennewitz, M. (2014). Monte Carlo localization for humanoid robot navigation in complex indoor environments, *International Journal of Humanoid Robotics* **11**(02), Article ID: 1441002.

Hornung, A., Wurm, K. and Bennewitz, M. (2010). Humanoid robot localization in complex indoor environments, *IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan*, pp. 1690–1695.

Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C. and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees, *Autonomous Robots* **34**(3): 189–206.

Ido, J., Shimizu, Y., Matsumoto, Y. and Ogasawara, T. (2009). Indoor navigation for a humanoid robot using a view sequence, *International Journal of Robotics Research* **28**(2): 315–325.

Kajita, S., Kanehiro, F., Fujiwara, K., Harada, K., Yokoi, K. and Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point, *IEEE International Conference on Robotics and Automation, Taipei, Taiwan*, pp. 1620–1626.

Kerl, C., Sturm, J. and Cremers, D. (2013a). Dense visual SLAM for RGB-D cameras, *IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan*, pp. 2100–2106.

Kerl, C., Sturm, J. and Cremers, D. (2013b). Robust odometry estimation for RGB-D cameras, *IEEE International Conference on Robotics and Automation, Karlsruhe, Germany,* pp. 3748–3754.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces, *6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07), Nara, Japan*, pp. 225–234.

Lourakis, M. A. and Argyros, A. (2009). SBA: A software package for generic sparse bundle adjustment, *ACM Transactions on Mathematical Software* **1**(36): 1–30.

Maier, D., Hornung, A. and Bennewitz, M. (2012). Real-time navigation in 3D environments based on depth camera data, *IEEE International Conference on Humanoid Robots, Osaka, Japan*, pp. 692–697.

Martínez, P.A., Varas, D., Castelán, M., Camacho, M., Marques, F. and Arechavaleta, G. (2014). 3D shape reconstruction from a humanoid generated video sequence, *IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain*, pp. 699–706.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F. and Sayd, P. (2009). Generic and real-time structure from motion using local bundle adjustment, *Image and Vision Computing* **8**(27): 1178–1193.

Obwald, S., Hornung, A. and Bennewitz, M. (2012). Improved proposals for highly accurate localization using range and vision data, *IEEE/RSJ International Conference on Intelligent Robots and System, Vilamoura, Portugal*, pp. 1809–1814.

Oriolo, G., Paolillo, A., Rosa, L. and Vendittelli, M. (2013). Vision-based trajectory control for humanoid navigation, *IEEE-RAS International Conference on Humanoid Robots, Atlanta, GA, USA*, pp. 113–123.

Oriolo, G., Paolillo, A., Rosa, L. and Vendittelli, M. (2016). Humanoid odometric localization integrating kinematic, inertial and visual information, *Autonomous Robots* **40**(5): 867–879.

Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking, *10th IEEE International Conference on Computer Vision, Beijing, China*, Vol. 2, pp. 1508–1515.

Royer, E., Lhuillier, M., Dhome, M. and Lavest, J.M. (2007). Monocular vision for mobile robot localization and autonomous navigation, *International Journal of Computer Vision* **74**(3): 237–260.

Santana, A.M. and Medeiros, A.A.D. (2012). Straight-lines modelling using planar information for monocular SLAM, *International Journal of Applied Mathematics and Computer Science* **22**(2): 409–421, DOI: 10.2478/v10006-012-0031-8.

Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry, *IEEE Robotics and Automation Magazine* **18**(4): 80–92.

Segal, A., Haehnel, D. and Thrun, S. (2009). Generalized-ICP, *in* J. Trinkle *et al.* (Eds.), *Proceedings of Robotics: Science and Systems*, The MIT Press, Cambridge, MA.

Skrzypczyński, P. (2009). Simultaneous localization and mapping: A feature-based probabilistic approach, *International Journal of Applied Mathematics and Computer Science* **19**(4): 575–588, DOI: 10.2478/v10006-009-0045-z.

Stasse, O., Davison, A., Sellaouti, R. and Yokoi, K. (2006). Real-time 3D SLAM for a humanoid robot considering pattern generator information, *IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, pp. 348–355.

Strasdat, H., Montiel, J. and Davison, A. (2010). Real-time monocular SLAM: Why filter?, *IEEE International Conference on Robotics and Automation, Anchorage, AK, USA*, pp. 2657–2664.

Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems, *IEEE International Conference on Intelligent Robots and Systems, Vilamoura, Portugal*, pp. 573–580.

Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A. (1999). Bundle adjustment a modern synthesis, *in* B. Triggs *et al.* (Eds.), *Vision Algorithms: Theory and Practice*, Lecture Notes in Computer Science, Vol. 1883, Springer-Verlag, London, pp. 298–372.

Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C. and Bur-gard, W. (2010). OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems, *IEEE International Conference on Robotics and Automation, Anchorage, AK, USA*.

Pablo A. Martínez obtained his B.Sc. from the Technological Institute of Fresnillo, Zacatecas (ITSF), in 2007. He obtained his M.Sc. and Ph.D. in robotics and advanced manufacturing from the Research Center for Advanced Studies of the National Polytechnic Institute (CINVESTAV-Saltillo) in 2010 and 2016, respectively. During his Ph.D. he was engaged in research on visual-based active localization and 3D reconstruction for humanoid robots. His research interests are related to the area of computer vision for robotics applications.

Mario Castelán obtained his B.Sc. from the University of Veracruz (UV) in 1999 and his M.Sc. in artificial intelligence from the University of Veracruz and the National Laboratory of Advanced Informatics (LANIA) in 2002. He obtained his Ph.D. in computer science from the University of York, UK, in 2006. Currently, he is a full-time researcher at the Robotics and Advanced Manufacturing Research Group of CINVESTAV-Saltillo. His research interests are focused on 3D shape analysis and statistical learning for computer vision and robotics applications.

Gustavo Arechavaleta received his M.Sc. degree from the Monterrey Institute of Technology (ITESM), Mexico, and his Ph.D. degree from the University of Toulouse, France, in 2003 and 2007, respectively. During his Ph.D. he was engaged in research on motion planning for anthropomorphic mechanisms and on the computational principles of movement neuroscience via optimal control. He has been a researcher in the Robotics and Advanced Manufacturing Group at CINVESTAV, Mexico, since 2008. His current research interests include humanoid motion generation and perception, human locomotion, and trajectory optimization.

Received: 18 September 2015
Revised: 13 March 2016
Accepted: 4 April 2016