Open access • Journal Article • DOI:10.1007/S10514-012-9312-1

How to localize humanoids with a single camera — Source link

Pablo F. Alcantarilla, Olivier Stasse, Sébastien Druon, Luis M. Bergasa ...+1 more authors

Institutions: University of Auvergne, Centre national de la recherche scientifique, University of Alcalá, Georgia Institute of Technology

Published on: 01 Jan 2013 - Autonomous Robots (Springer US)

Topics: Bundle adjustment, 3D reconstruction, Humanoid robot, Visibility (geometry) and Monocular vision

How to Localize Humanoids with a Single Camera?

Alcantarilla Pablo, Olivier Stasse, Sébastien Druon, Bergasa Luis M., Frank Dellaert

Autonomous Robots manuscript No.
(will be inserted by the editor)

How to Localize Humanoids with a Single Camera?

Pablo F. Alcantarilla · Olivier Stasse · Sebastien Druon · Luis M. Bergasa · Frank Dellaert

Received: date / Accepted: date

Abstract In this paper, we propose a real-time vision-based localization approach for humanoid robots using a single camera as the only sensor. In order to obtain an accurate localization of the robot, we first build an accurate 3D map of the environment. In the map computation process, we use stereo visual SLAM techniques based on non-linear least squares optimization methods (bundle adjustment). Once we have computed a 3D reconstruction of the environment, which comprises of a set of camera poses (keyframes) and a list of 3D points, we learn the *visibility* of the 3D points by exploiting all the geometric relationships between the camera poses and 3D map points involved in the reconstruction. Finally, we use the prior 3D map and the *learned visibility* prediction for monocular vision-based localization. Our algorithm is very efficient, easy to implement and more robust and accurate than existing approaches. By means of visibility prediction we predict for a query pose only the highly visible 3D points, thus, speeding up tremendously the data association between 3D map points and perceived 2D features in the image. In this way, we can solve very efficiently the Perspective-n-Point (PnP) problem providing robust and fast vision-based localization. We demonstrate the robustness and accuracy of our approach by showing several vision-based localization experiments with the HRP-2 humanoid robot.

Keywords Vision-Based Localization · Visibility Prediction · Humanoid Robots · Locally Weighted Learning · Bundle Adjustment

Pablo F. Alcantarilla
ISIT-UMR 6284 CNRS, Université d'Auvergne, Clermont Ferrand, France
E-mail: pablofdezalc@gmail.com

Olivier Stasse
LAAS, CNRS, Toulouse, France
E-mail: ostasse@laas.fr

Sebastien Druon
LIRMM, University Montpellier II, Montpellier, France
E-mail: druon@lirmm.fr

Luis M. Bergasa
Department of Electronics, University of Alcalá, Madrid, Spain
E-mail: bergasa@depeca.uah.es

Frank Dellaert
School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA
E-mail: dellaert@cc.gatech.edu

1 Introduction

In this paper, we consider the problem of real-time localization for humanoid robots using a single camera as the only sensor. In order to obtain fully autonomous robots an accurate localization of the robot in the world is much more than desirable. Furthermore, if we can obtain an accurate localization in real-time, we can use the remaining computational resources to perform other important humanoid robotic tasks such as planning (Perrin et al, 2010), 3D object modeling (Foissote et al, 2010) or visual perception (Bohg et al, 2009).

Indeed, many humanoid robotics applications will be benefited from an accurate and fast localization of the robot. For a robust localization, we can choose between different alternatives: One option is to estimate simultaneously the localization of the robot and the map of the environment, yielding the well-known Simultaneous Localization and Mapping (SLAM) problem in the robotics community (Durrant-White and Bailey, 2006). However another possible option is to dedicate more computational resources in the construction

of a persistent map, and then use this map for long-term localization or navigation purposes. In this way, we can take advantage of the prior map of the robot's environment learning different parameters ranging from *visibility prediction* (Alcantarilla et al, 2011), 3D object reconstruction (Stasse et al, 2007) to scene understanding (Li et al, 2009).

However, accurate vision based localization for humanoid robots is still a challenging problem due to several aspects such as: noisy odometry, innacurate 3D data, complex motions and motion blur originated due to fast robot motion and to the large jerk caused by the landing impact of the feet. In addition, humanoids usually can not be assumed to move on a plane to which their sensors are parallel due to their walking motion (Hornung et al, 2010). Therefore, compared to wheeled robots, there is still open research for reliable localization of humanoid robots.

In the particular case of humanoid robots, it is very important that the sensors are light-weight and small. Humanoids should be stable under all possible motions, and heavy sensors can compromise this stability. Besides, not all sensors are suitable for humanoid robots. For example not all laser scanners can be mounted on humanoid platforms, especially the heavy ones such as the SICK LMS-221. Only small laser range sensors (e.g. Hokuyo URG-04LX) are suitable for humanoid robotics applications (Kagami et al, 2005). However, the main problem of these small laser range sensors is the limited distance range (up to 4 m for the Hokuyo URG-04LX). All these reasons make cameras an appealing sensor for humanoid robots: they are light-weight and cheaper than laser scanners. In addition, stereo cameras can also provide higher distance ranges (depending on the the stereo rig baseline). Moreover, most of advanced commercial humanoids platforms are already equipped with vision systems. However, there have been only limited attempts at vision-based localization for humanoid robots.

In this work, we show that is possible to obtain a real-time robust localization of a humanoid robot, with an accuracy of the order of cm just using a single camera and a single CPU. Prior to localization we compute an accurate 3D map of the environment by means of stereo visual SLAM techniques. For building an accurate 3D map we use stereo vision for one important reason: we can measure in directly the scale of each 3D point thanks to the computation of a dense disparity map between the two images, which is a well-studied problem for stereo vision (Scharstein and Szeliski, 2002). In this way we can solve the main drawback of monocular SLAM approaches, recovering the scale of a map due to observability problems in recovering 3D information from 2D projections. Once we have obtained a 3D map of the environment, we perform monocular vision-based localization using the 3D map as a prior. Hence, for localization experiments we can avoid the dense disparity map computation, which in certain occassions can be an important time-consuming operation, and perform robust and efficient real-time localization just using a single camera and a 3D map as a prior.

To satisfy all these demands, we firstly build a 3D map of the environment using stereo visual SLAM techniques and Bundle Adjustment (BA) (Triggs et al, 1999; Mouragnon et al, 2009). Inspired by recent works in visual SLAM, we propose to use a stereo visual SLAM algorithm combining local and global BA to obtain accurate 3D maps with respect to a global coordinate frame. Then, these maps can be used later for monocular vision based localization or navigation. In this way, 3D points and camera poses are refined simultaneously through the sequence by means of local BA, and when a loop closure is detected, the residual error in the reconstruction can be corrected by means of global BA adding the loop closure constraints.

Once we have a 3D map of the environment, we would like to use this map for different robotics applications such as localization, planning or navigation. Vision-based localization in a large map of 3D points is a challenging problem. One of the most computationally expensive steps in vision-based localization is the data association between a large map of 3D points and 2D features perceived by the camera. Then, matching candidates are usually validated by geometric constraints using a Random Sample Consensus (RANSAC) framework (Bolles and Fischler, 1981). Therefore, we need a smart strategy to sample the large database of 3D points and perform an efficient data association between the 3D map points and perceived 2D features by the camera. Given a prior map of 3D points and perceived 2D features in the image, our problem to solve is the estimation of the camera pose (with known intrinsic parameters) with respect to a world coordinate frame. Basically, this problem is known in the literature as the Perspective-n-Point (PnP) problem (Lu et al, 2000; Ansar and Danilidis, 2003).

For solving efficiently the PnP problem, we propose to use the *visibility prediction* algorithm described in (Alcantarilla et al, 2011). Visibility prediction exploits all the geometric relationships between camera poses and 3D map points in the prior 3D reconstruction. Then, during vision-based localization experiments we can speed-up tremendously the data association and robot localization by predicting only the most highly visible 3D points given a prior on the camera pose. In this way, we can solve the PnP problem in an efficient

and fast way, reducing considerably the number of outliers in the set of 3D-2D correspondences.

In (Alcantarilla et al, 2010), the visibility prediction idea was successfully used for estimating the pose of a hand-held camera in cluttered office-like environments. Results were quite satisfactory taking into account that no appearance descriptor was considered in the data association process between the 3D map points and perceived 2D features. In this work, we use the visibility prediction algorithm in the context of humanoid robot localization, but adding more capabilities due to the use of appearance information. In our map, each 3D point is also described by a low-dimensional descriptor vector that encodes appearance information. By means of appearance information, we can easily perform fast robot re-localization for those cases where the robot gets lost or is kidnapped.

The paper is organized as follows: In Section 2 we review the different approaches regarding humanoid robots localization and their main limitations. The stereo visual SLAM algorithm is explained in Section 3. Then, we describe in Section 4 how to learn the visibility of 3D points given a prior 3D reconstruction. In Section 5, we explain the main steps of our monocular vision-based localization algorithm. In Section 6 we show extensive localization experiments with the HRP-2 robot. Finally, we present main conclusions and future work in Section 7.

2 Related Work

Most humanoid robotic platforms have vision systems. Cameras seem to be an appealing sensor for humanoid robotics applications: they are small, cheap and lightweight compared to other sensors such as laser scanners. However, there have been only limited attempts at vision-based localization, whereas more interesting results have been obtained using laser scanners as the main sensor (Stachniss et al, 2008; Hornung et al, 2010).

Ozawa et al. (2007) proposed to use stereo visual odometry to create local 3D maps for online footstep planning. They validate their algorithm, performing several experiments with biped robots walking through an obstacle-filled room, while avoiding obstacles. The main drawback of this approach is the drift created by the accumulation of errors from visual odometry systems (Nistér et al, 2004; Kaess et al, 2009). In addition, this approach lacks the ability to close loops and the local nature of the obtained 3D maps prevents the maps from life-long mapping. Within the visual odometry context, Pretto et al. (2009) proposed a framework robust to motion blur. Motion blur is one of the most severe problems in grabbed images by humanoid robots, specially for the smaller ones, making vision-based applications challenging.

Michel et al.In (2007) proposed a real-time 3D tracking for humanoid robot locomotion and stair climbing. By tracking the model of a known object they were able to recover the robot's pose and to localize the robot with respect to the object. Real-time performance was achieved by means of Graphic Processing Units (GPUs). The main limitations of this approach are that it is extremely dependent on the 3D object to track and that the 3D model is relatively small. However, it can be useful for challenging humanoid robots scenarios such as stairs climbing.

Kwak et al. (2009) presented a 3D grid and particle based SLAM for humanoid robots using stereo vision. The depth data from the stereo images was obtained by capturing the depth information of the stereo images at static positions in the environment, measuring the distance between these positions manually. This tedious initialization and the computational burden introduced by the grid matching process, prevents the system from mapping more complex environments and from real-time performance, which is of special interest in humanoid robots applications.

Davison et al. (2007) showed succesful monocular visual SLAM results for small indoor environments using the HRP-2 robot. This approach, known as MonoSLAM, is a monocular Extended Kalman Filter (EKF) vision-based system, that allows to build a small map of sparse 3D points. This persistent map permits almost drift-free real-time localization over a small area. However, only accurate results are obtained when the pattern generator, the robot odometry and inertial sensing are fused to aid the visual mapping into the EKF framework as it was shown in (Stasse et al, 2006). The fusion of the information from different sensors can reduce considerably the uncertainty in the camera pose and the 3D map points involved in the EKF process, yielding better localization and mapping results. Although in most of occasions, odometry in humanoid robots can be estimated only very roughly (Hornung et al, 2010).

The main drawback of EKF-based approaches is the limited number of 3D points that can be tracked, apart from divergence from the true solution due to linearization errors. As it has been shown in several works (Dellaert and Kaess, 2006; Strasdat et al, 2010) non-linear optimization techniques such as BA or Smoothing and Mapping (SAM) are superior in terms of accuracy to filtering based methods, and allow to track many hundreds of features between frames. BA is a very popular and well-known technique used in computer vision, and in particular for Structure-from-Motion (SfM) problems. A complete survey on the basis of BA meth-

ods can be found in (Triggs et al, 1999). More recent works focus in the scalability of BA in large-scale environments (Byröd and Åström, 2010; Jian et al, 2011). BA has been successfully employed in different problems such as augmented reality (Klein and Murray, 2007) or large-scale mapping for mobile robot platforms (Konolige and Agrawal, 2008; Mei et al, 2010).

One of the most successful monocular SLAM approaches is the Parallel Tracking and Mapping approach (PTAM) (Klein and Murray, 2007). PTAM was originally developed for augmented reality purposes in small workspaces and combines the tracking of many hundred of features between consecutive frames for an accurate camera pose estimation and non-linear optimization of the map. The map optimization uses a subset of all camera frames of special importance in the reconstruction (keyframes) to build a 3D map of the environment. Recently, (Blösch et al, 2010) showed a vision-based navigation approach for micro aerial vehicles (MAVs) that uses PTAM for accurate pose estimates. The main limitations of PTAM are that it does not scale well with larger environments and that it is necessary to simulate a virtual stereo pair to initiliaze the algorithm. This initialization is carried out in order to estimate an approximate depth of the initial 3D points. Then, new 3D points will be triangulated according to previous reconstructed keyframes. This initialization procedure plays a very important role in the final quality of the 3D map and results can differ substantially from real ones if this stereo initialization is not accurate enough as shown in (Wendel et al, 2011). Therefore, in order to avoid these problems we propose to use our own stereo visual SLAM algorithm to build an accurate 3D map of the scene and then perform efficient and fast monocular vision-based localization with visibility prediction. In Section 6.3.3 we compare our monocular vision-based localization algorithm to the PTAM approach under one experiment done with the HRP-2 robot.

3 Stereo Simultaneous Localization and Mapping

Figure 1 depicts an overview of the main components of our stereo visual SLAM system. Notice, that in this work we are mainly interested in using stereo visual SLAM for computing a 3D map, that will be used later for visibility learning and monocular vision-based localization. Therefore in this section, we briefly review the main components of our visual SLAM module.

3.1 Preprocessing

Before performing any visual SLAM processing, the stereo rig calibration parameters (intrinsics, extrinsics) are obtained in a camera calibration setup. We used a chessboard pattern of known dimensions as a calibration object, and around twenty pairs of images were taken for the calibration. The stereo rig was calibrated offline using the Camera Calibration Toolbox for Matlab (Bouguet, 2008b).

Given the calibration parameters, we perform distortion correction and stereo rectification (Hartley, 1999; Bouguet, 2008a) for the input images. Stereo rectification simplifies considerably the stereo correspondences problem and allows to compute dense disparity maps. In this work, we use the method proposed in (Konolige, 1997) for computing the disparity maps, since it offers a good compromise between speed and performance. Notice here that more modern stereo disparity methods can also be used (Hirschmüller, 2008; Geiger et al, 2010).

After stereo rectification, we obtain a new set of calibration parameters, where the left and right cameras have the same focal length f and principal point (u_0, v_0). The rotation matrix between cameras R_{LR} is the identity matrix, and the translation vector T_{LR} encodes the baseline B of the rectified stereo rig. Now, considering an ideal stereo system, the depth of one 3D point can be determined by means of the following equation:

$$Z = f\frac{B}{u_R - u_L} = f \cdot \frac{B}{d_u} , \quad (1)$$

where d_u is the horizontal pixel disparity. Given the depth Z and the stereo image projections of the point in both images (u_L, u_R, v) (notice that in a rectified stereo $v_L = v_R = v$) the rest of the coordinates of the 3D point with respect to the camera coordinate frame can be determined as:

$$X = \frac{Z \cdot (u_L - u_0)}{f} , \quad (2)$$

$$Y = \frac{Z \cdot (v - v_0)}{f} , \quad (3)$$

3.2 Stereo Visual Odometry

Visual odometry (Nistér et al, 2004; Kaess et al, 2009) is as its heart a pose estimation problem, that allows to estimate the relative camera motion between two consecutive frames. In addition, visual odometry can be implemented very efficiently in real-time and can be used to obtain good priors for the camera poses and 3D points that can be optimized later in a BA procedure.

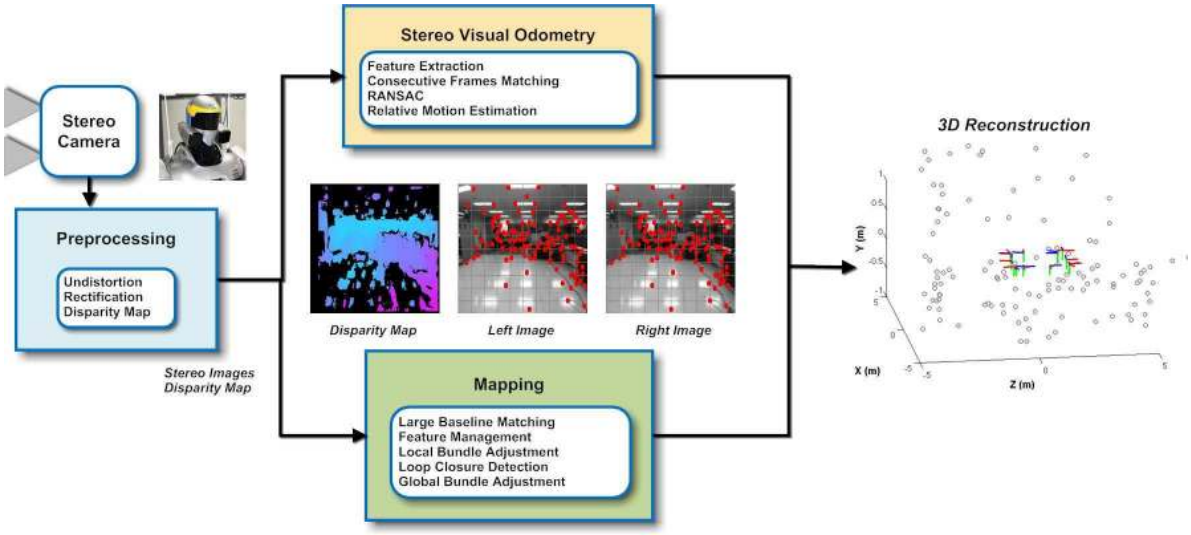


Fig. 1 Stereo visual SLAM system overview: First, we undistort and rectify the stereo images given the stereo rig calibration parameters. Then, for the left image of the stereo pair, we detect 2D features of interest and associated descriptor vectors that encode appearance information. By performing stereo visual odometry between two consecutive frames, we estimate the relative camera motion between frames. Finally, the accumulated relative camera motion is transformed into a global coordinate frame, and a set of selected camera poses (keyframes) and 3D points are refined in a local BA procedure. When a loop closure is detected, we reduce the residual re-projection error by means of global BA adding the loop closure constraints.

We estimate the relative camera motion by matching features between two consecutive frames. Features are detected by using the Harris corner detector (Harris and Stephens, 1988) at different scale levels. We detect features only for the left image of the stereo pair. Then, we find the correspondences of the 2D features in the right image by accessing the disparity map and compute the 3D coordinates of the point with the stereo geometry equations (see Equations 1, 2, 3). Finally, we have a set of M stereo features at frame t, $\mathcal{F}_{t_i} = \{(u_L, u_R, v)_i, h_i\}$ with $i = 1 \dots M$.. The 2D point (u_L, v) is the location of the feature in the left image and (u_R, v) is its corresponding location in the right view. In addition, we also store for each stereo feature $\mathcal{F}_{t_i}$ the 3D coordinates of the reconstructed point h_i with respect to the camera coordinate frame at that time instant t.

For each detected 2D feature in the left image we also extract a descriptor vector that encodes the appearance information of a local area centered on the point of interest. Similar to SURF (Bay et al, 2008), for a detected feature at a certain scale, we compute a unitary descriptor vector of dimension 16 to speed-up the descriptor and matching computations. We use the upright version of the descriptors (no rotation invariant) since upright descriptors perform better in scenarios where the camera only rotates around its vertical axis, which is often the case of humanoid robots applications. For simplicity, we do not use any kind of spatial or Gaussian weighting. Even though, this descriptor dimension may seem relatively small, matching is robust enough for obtaining accurate and fast vision-based localization as we will show in our experimental results section.

Once we have computed the features descriptors, we find the set of *putatives* (set of correspondences) between the stereo features from the current frame $\mathcal{F}_t$ and the previous one $\mathcal{F}_{t-1}$ by matching their associated list of descriptors vectors. In order to reduce matching ambiguities we only try to match descriptors between consecutive frames in a circular area of a fixed radius centered on the detected feature in the current frame. In our experiments, a fixed radius of 15 pixels is enough for finding the set of putatives between two consecutive frames considering an image resolution of 320×240 pixels.

After finding the set of putatives between two consecutive frames we estimate the relative camera motion using a standard two-point algorithm in a RANSAC setting by minimizing the following cost function:

$$\underset{R^t_{t-1}, \mathbf{t}^t_{t-1}}{\arg\min} \sum_i \left\| z_{i,t} - \Pi \left(R^t_{t-1}, \mathbf{t}^t_{t-1}, h_{i,t-1} \right) \right\|_2 , \qquad (4)$$

where $z_{i,t} = \{(u_L, u_R, v)_i\}$ are the set of 2D measurements of a stereo feature at time t and Π is a function that projects a 3D point $h_{i,t-1}$ (referenced to the camera coordinate frame at time $t-1$) to the image coordinate frame at time t assuming the pin-hole camera model. This projection function Π involves a rotation R^t_{t-1} and a translation $\mathbf{t}^t_{t-1}$ of 3D points between both coordinate frames and a projection onto

the image plane by means of the stereo rig calibration parameters. The resulting relative camera motion is transformed to a global coordinate frame and then used by the mapping management module. We use the Levenberg-Marquardt (LM) algorithm (Marquardt, 1963) for all the non-linear optimizations.

3.3 Bundle Adjustment and Map Management

By means of stereo visual odometry, we estimate the relative camera motion between consecutive frames. When the accumulated motion in translation or rotation is higher than a fixed threshold we decide to create a new keyframe. This keyframe, will be optimized later in an incremental local BA procedure. While initializing a new keyframe, we store its pose with respect to a global coordinate frame, the set of detected 2D features, associated appearance descriptors and respective 3D points. In addition, we also store its visibility information, i.e. the list of 3D points that are visible from that keyframe. This information will be used later in the visibility learning procedure as explained in Section 4. In our experiments, we add a new keyframe when the accumulated translation or rotation is higher than 0.15 m and 5° respectively.

BA provides an iterative optimization of the camera poses and 3D points involved in the reconstruction. Roughly speaking, BA is a non-linear least squares problem and consists in the minimization of the sum of squared reprojection errors. In general, BA has a $\Theta\left(N^3\right)$ time complexity, being N the number of variables involved in the optimization problem (Hartley and Zisserman, 2000). This time complexity becomes a computational bottleneck for incremental SfM or visual SLAM approaches that have real-time constraints. Therefore another alternatives that can reduce this time complexety are necessary. In addition, it is also important to have an initial estimate of the parameters close to the real solution (Schweighofer and Pinz, 2006). In our work we obtain a robust initialiation of the structure and motion by means of the stereo visual odometry algorithm described in Section 3.2.

For optimizing simultaneously the set of camera poses and 3D points in real-time, we use the incremental local BA approach described in (Mouragnon et al, 2009). We use a sliding window BA over the last N_k keyframes, optimizing only the camera parameters of the last n_k cameras. With respect to the 3D points, only those 3D points that are visible in the last n_k cameras are optimized. In this way, 3D points and camera poses are refined simultaneously through the sequence. Optimal values for these parameters are typically $n_k = 3$ and $N_k = 10$, see (Mouragnon et al, 2009) for more details. In this work, we use the Sparse Bundle Adjustment (SBA) package (Lourakis and Argyros, 2009) as the basis for our local BA implementation. SBA exploits the inherent sparsity structure of the problem and is widely used in the computer vision community (Snavely et al, 2006; Agarwal et al, 2009).

We perform an *intelligent management* of features into the map, in order to produce an equal distribution of feature locations over the image. While adding a new feature to the map, we also store its associated appearance descriptor and 3D point location. Then, we try to match the feature descriptor against detected new 2D features on a new keyframe by matching their associated descriptors in a high probability search area. In this way, we can create for a map element, *feature tracks* that contain the information of the 2D measurements of the feature (both in left and right views) over several keyframes. Then, this information is used as an input in the local BA procedure. Features are deleted from the map when the mean re-projection error in the 3D reconstruction is higher than a fixed threshold (e.g. 3 pixels).

By means of appearance based methods, loop closure situations can be detected. We try to match the set of descriptors from the current image to the stored descriptors from previous keyframes, but only taking into account those keyframes that are inside a small uncertainty area around the current camera location. We also check for geometric consistency by means of epipolar geometry. This geometric consistency check is very important and almost guarantees that there will be no false positives, even using a very low inlier threshold (Konolige and Agrawal, 2008). Even simple, our method can detect very efficiently loop closure situations although incremental *Bag of Visual Words* methods can be also used (Cummins and Newman, 2008; Angeli et al, 2008). However, loop closure situations are very few in normal humanoid robots scenarios, since usually these scenarios are laboratory-based and relatively small. Once a loop closure is detected, the residual error in the 3D reconstruction can be corrected in a global BA step. Normally, due to the fact that the scenarios are relatively small, the accumulated drift or error is very small and therefore few iterations are necessary in the global BA step.

4 Visibility Prediction of known 3D Points

In the visibility prediction problem, we are interested in the posterior distribution of the visibility v_j for a certain 3D point x_j given the query camera pose θ, denoted as $P(v_j|\theta)$.

The visibility of known 3D points can be approximated by using a form of lazy and memory-based learning technique known as *Locally Weighted Learning* (Atkeson et al, 1997). This technique is a simple memory-based classification algorithm and can be implemented very efficiently. The idea is very simple: given the training data that consists of a set of reconstructed camera poses $\Theta = \{\theta_1 \dots \theta_N\}$, the 3D point cloud $X = \{x_1 \dots x_M\}$ and a query camera pose θ, we form a locally weighted average at the query point and take that as an estimate for $P(v_j|\theta)$ as follows:

$$P(v_j|\theta) \approx \frac{\sum_{i=1}^{N} k(\theta, \theta_i) \cdot v_j(\theta_i)}{\sum_{i=1}^{N} k(\theta, \theta_i)} \ , \tag{5}$$

where the function $k(\theta, \theta_i)$ is a kernel function that measures the similarity between two camera poses, and the function $v_j(\theta_i)$ just assigns a real value equal to 1 for those cases where a certain 3D point x_j is visible by a camera pose θ_i and 0 otherwise. In the end, the main problem is finding an appropriate kernel function $k(\theta, \theta_i)$ that captures correctly the similarity between two camera poses, emphasizing similar ones and deemphasizing very different camera poses.

The kernel function is learned by combining the Gaussian kernel and Mahalanobis distance. More in detail, we need to learn the kernel parameters from the training data, by fitting the kernel function to a set of target values. These target values y_{ij} are defined as the mean of the ratios between the intersection of the common 3D points with respect to the number of 3D points visible to each of the two cameras:

$$y_{ij} = \frac{1}{2} \cdot \left| \frac{|X_i \cap X_j|}{|X_i|} + \frac{|X_j \cap X_i|}{|X_j|} \right| \ , \tag{6}$$

Finally, the expression of the kernel function that measures the similarity between two camera poses is:

$$k_{ij} \equiv k(\theta_i, \theta_j) = \exp\left(- \left\| \mathbf{A}(\hat{\theta}_i - \hat{\theta}_j) \right\|_2 \right) \ , \tag{7}$$

where $\mathbf{A}$ is a $n \times n$ matrix, being n the number of cues used in the proposed metric. In this work, each camera pose is parametrized by means of a vector $\hat{\theta}_i = \{T_i, R_i\}$ (3D vector for the translation and 4D unit quaternion for the rotation). For simplicity, we just use two cues in the proposed metric: difference in camera translation and dot product between cameras viewing directions vectors, capturing efficiently the differences between camera poses due to changes in translation and orientation. Even though we only use two cues in the metric, the proposed framework allows to incorporate more cues in the metric such as RGB histograms, local appearance descriptors, disparity information, etc.

The visibility posterior can be approximated by just considering the K Nearest Neighbors (KNNs) of the current query pose θ_t. As a consequence, once we find the KNNs of the current query pose, we only need to predict the visibilities for the subset of map elements which are at least seen once by these KNNs. Then, we can set the visibilities to be zero for the rest of map elements. Finally, we obtain the locally weighted K nearest neighbor approximation for the visibility posterior as follows:

$$P(v_j = 1|\theta) \approx \frac{\sum_{i=1}^{K} k(\theta, \theta_i{}^{v_j=1})}{\sum_{i=1}^{K} k(\theta, \theta_i)} \ , \tag{8}$$

where only the nearest K samples of the query pose $\Theta^K = \{\theta_1 \dots \theta_k\}$ are considered.

5 Monocular Vision-Based Localization

Now, once we have obtained a 3D map of the environment (by using the stereo visual SLAM algorithm described in Section 3), we are interested in exploiting that map for common humanoid robot tasks such as navigation or planning, while providing at the same time an accurate robot localization. For this purpose, obtaining a real-time and robust vision-based localization is mandatory. Given a prior map of 3D points and perceived 2D features in the image, our problem to solve is the estimation of the camera pose with respect to the world coordinate frame, i.e. the PnP problem.

The PnP problem is a thoroughly studied problem in computer vision (Lu et al, 2000; Ansar and Danilidis, 2003). In general, even with a perfect set of known 3D-2D correspondences, this is a challenging problem. Although there exist some globally optimal solutions such as (Schweighofer and Pinz, 2008) that employ Second Order Cone Programs (SOCP), the main drawback of the current globally optimal solutions to the PnP problem is the computational burden of these methods. This makes difficult to integrate these algorithms for real-time applications such as the ones we are interested with humanoid robots.

Our main contribution for solving the PnP problem efficiently, is using the output of the *visibility prediction* algorithm (given a prior on the camera pose) to predict only the most highly visible 3D points, reducing considerably the number of outliers in the set of correspondences. In this way, we can make the data association between 3D map points and 2D features easier, thus

speeding-up the pose estimation problem. Figure 2 depicts an overall overview of our vision-based localization approach with visibility prediction. To clarify, the overall vision-based localization algorithm works through the following steps:

1. While the robot is moving, the camera acquires a new image from which a set of image features $Z_t = \{z_{t,1} \dots z_{t,n}\}$ are detected by a feature detector of choice. Then, a feature descriptor is computed for each of the detected features. Notice, that even any kind of feature detector and descriptor may be used, it is necessary that both detector and descriptor are the same and have the same settings as in the map computation process described in Section 3.
2. Then, by using the visibility prediction algorithm, a promising subset of highly visible 3D map points is chosen and re-projected onto the image plane based on the estimated previous camera pose θ_{t-1} and known camera parameters.
3. Afterwards, a set of putative matches C_t is formed where the i-th putative match $C_{t,i}$ is a pair $\{z_{t,k}, x_j\}$ which comprises of a detected feature z_k and a map element x_j. A putative match is created when the Euclidean distance between the appearance descriptors of a detected feature and a re-projected map element is lower than a certain threshold.
4. Finally, we solve the pose estimation problem minimizing the following cost error function, given the set of putative matches C_t:

$$\arg\min_{R,\mathbf{t}} \sum_{i=1}^{m} \|z_i - K\,(R \cdot x_i + \mathbf{t})\|_2 \ , \tag{9}$$

where $z_i = (u_L, v_L)$ is the 2D image location of a feature in the left camera, x_i represents the coordinates of a 3D point in the global coordinate frame, K is the left camera calibration matrix, and R and t are respectively the rotation and the translation of the left camera with respect to the global coordinate frame. The PnP problem is formulated as a non-linear least squares procedure using the LM algorithm implementation described in (Lourakis, 2004). The set of putative matches may contain outliers, therefore RANSAC is used in order to obtain a robust model free of outliers.

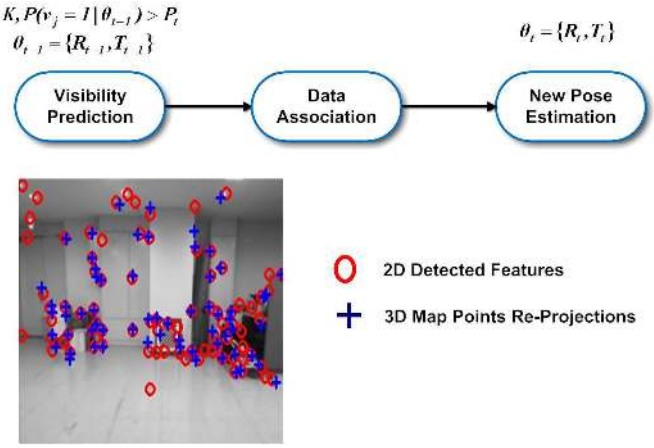


Fig. 2 The input for the visibility prediction algorithm is the latest camera pose θ_{t-1}, the number of KNNs (K) and a probability threshold P_t. Only the highly visible 3D map points are re-projected onto the image plane of the left camera, and a set of putative matches between 2D detected features and map elements is formed. Then, the PnP problem is solved yielding the localization of the robot with respect to a world coordinate frame θ_t at time t.

5.1 Initialization and Re-Localization

During the initialization, the robot can be located in any area of the map. First, we need to find a prior camera pose to initialize the vision-based localization algorithm. For this purpose, we compute the appearance descriptors of the detected 2D features in the new image and match this set of descriptors against the set of descriptors from the list of stored keyframes from the prior 3D reconstruction. In the matching process between the two frames, we perform a RANSAC procedure forcing epipolar geometry constraints. We recover the camera pose from the stored keyframe that obtains the highest inliers ratio score. If this inliers ratio is lower than a certain threshold, we do not initialize the localization algorithm until the robot moves into a known area yielding a higher inliers ratio. At this point, we are confident about the camera pose prior and initialize the localization process with the camera pose parameters of the stored keyframe that has the highest score.

Eventually, it may happen that the robot gets lost due to bad localization estimates or that the new camera pose is rejected due to a small number of inliers in the PnP problem. In those cases, we perform a fast re-localization by checking the set of appearance descriptors of the robot's new image against only the stored set of descriptors of the keyframes that are located in a certain distance of confidence around the last accepted camera pose estimate.

Notice here, that other re-localization frameworks can been adapted to our vision-based localization system such as (Williams et al, 2007; Checklov et al, 2008). Williams et al. (2007), use randomised tree classifiers for fast feature matching during relocalization, whereas Cheklov et al. (2008), focus more on feature indexing on space and scale to facilitate matching during relocalization. In our experiments we obtained good relocalization results by just checking the appearance descriptors in a certain area of confidence around the previous cam-

era pose estimate and forcing epipolar geometry constraints. The epipolar geometry check is very important and almost guarantees that there will be no false positives as shown in (Konolige and Agrawal, 2008).

6 Results and Discussion

In this section, we show several localization experiments conducted on the HRP-2 humanoid robot. The HRP-2 humanoid platform is equipped with a high-performance forward-looking trinocular camera rig and a wide angle camera. The wide-angle camera is normally used for grasping or interaction tasks, providing the capability to make accurate 3D measurements of objects located very close to the camera. In this work, we only consider the two cameras that are attached to the ears of the robot. These two cameras have a baseline of approximately 14.4 cm and an horizontal field of view of 90° for each of the cameras.

For the stereo visual SLAM algorithm, we use both the left and right cameras, considering the left camera as the reference one in the 3D reconstruction process. Then, during monocular vision-based localization experiments we just consider only the left camera for the pose estimation problem. This is possible, since we use a prior 3D map and therefore we can perform vision-based localization with a single camera. Figure 3 depicts an image of the HRP-2 stereo rig settings. The height of HRP-2 is 155 cm in standing up position and the total weight is about 58 kg. More detailed specifications of this humanoid platform can be found in the work by Kaneko et al. (2004). We created two different datasets of common humanoid robotics laboratory environments. The first dataset is called *Tsukuba*, and it was done at the Joint Robotics Laboratory, CNRS-AIST, Tsukuba, Japan. This dataset comprises of different sequences for the evaluation of the monocular vision-based localization algorithm under the assumption that a prior 3D map is known. In particular, in this dataset we have different robot trajectories (square, straight) and challenging situations for the localization such as robot kidnapping, people moving in front of the robot and changes in lighting conditions. For this dataset, we performed experiments with an image resolution of 320×240 and a frame rate of 15 frames per second. The main motivation of using that image resolution is that in this dataset we focused more on achieving real-time localization results while at the same time obtaining robust pose estimates.

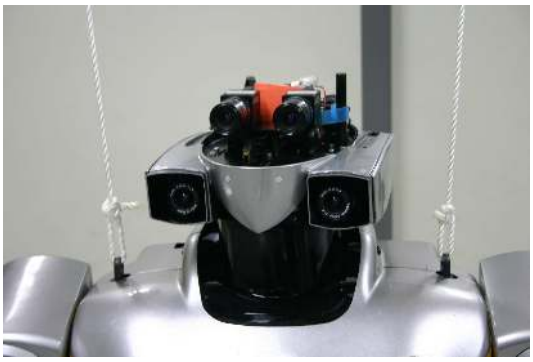


Fig. 3 HRP-2 stereo rig settings. In this work we consider the two cameras attached to the ears that have a baseline of approximately 14.4 cm.

The second dataset called *Toulouse* was done at the Gepetto Robotics and Artificial Intelligence laboratory, LAAS/CNRS, Toulouse, France. For this dataset, we performed experiments with an image resolution of 640×480 and a frame rate of 15 frames per second. By using a higher resolution, computation times will be higher than for the Tsukuba dataset, however we can expect some improvement in localization accuracy and quality of the 3D reconstruction. In addition, in this dataset we chose that resolution to perform a fair comparison against PTAM. Originally, PTAM stores a three level scale-space pyramid representation of each frame, being the level zero an image resolution of 640×480, and the coarsest level 80×60 pixels. In this dataset we have different robot trajectories (square, straight, circular) and also difficult scenarios such as people moving in front of the robot and some changes in the environment. We provide some datasets from our experiments in the website: `http://www.robesafe.com/personal/pablo.alcantarilla/humanoids.html`. The datasets include stereo calibration parameters and ground truth information for evaluation. The ground truth was obtained either by MOCAP or by stereo visual SLAM with global BA.

Figure 4 depicts some of the extracted keyframes for two different sequences from the Tsukuba and Toulouse datasets respectively. It can be observed that in some areas of the two different environments, there is a lack of texture due to the presence of walls (Figure 4(c)), fluorescent and natural lighting (Figure 4(d,h)) and foliage (Figure 4(h)).

In order to evaluate the accuracy of our vision-based localization algorithms, we compare our localization results against ground truth measurements for some of the sequences from the Toulouse dataset. We obtained ground truth information by using a Vicon motion capture system [1]. The Vicon motion capture system is

[1] For more information, please check the following url: http://www.vicon.com/

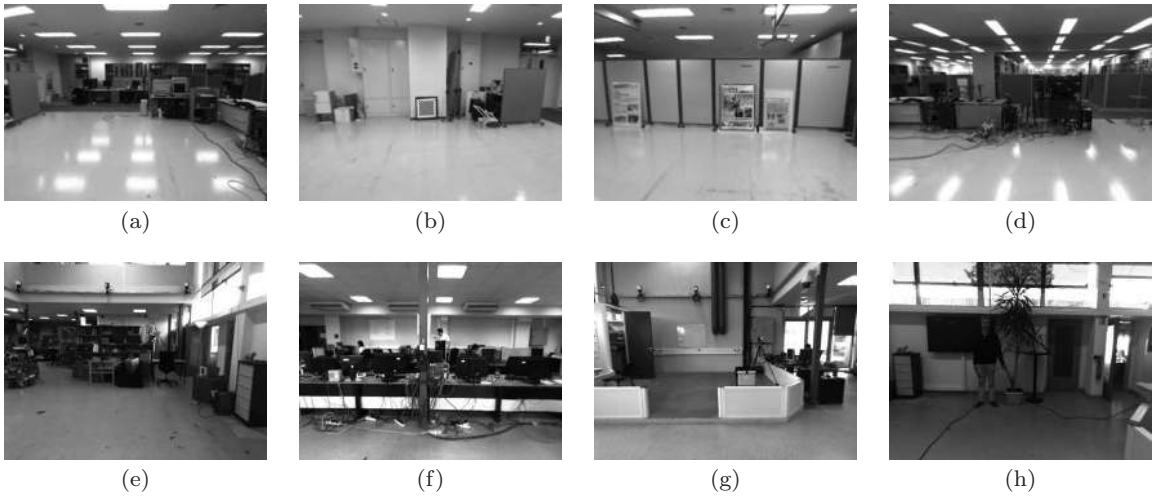


Fig. 4 Some keyframes of the reconstructed environments. The environments are typical from humanoid robotics laboratories. (a)-(d) Four extracted keyframes from one reconstruction from the Tsukuba dataset. (e)-(h) Four extracted keyframes from a sequence from the Toulouse dataset.

a state-of-the-art infrared marker-tracking system that offers millimeter resolution of 3D spatial displacements.

We used the pattern generator described in (Stasse et al, 2008) to perform a set of pre-computed sequences of interest. Due to noisy odometry, there exists a discrepancy between the desired trajectory and the real one. This is the reason why in the sequences the robot was not able to fully close the loop in some of the planned trajectories.

Firstly, we show the accuracy of our stereo visual SLAM algorithm in Section 6.1. We stand out the accuracy of our approach by comparing our trajectory estimates with respect to the ground truth obtained by the motion capture system. Then, we show monocular vision-based localization results with visibility prediction in Section 6.2 and Section 6.3, both for the Tsukuba and Toulouse datasets respectively. Finally we show a timing evaluation for the two different datasets in Section 6.4.

6.1 Stereo Visual SLAM Accuracy

For a robust localization of the robot, we compute an accurate 3D map of the environment by means of the stereo visual SLAM algorithm described in Section 3. In addition, since the visibility prediction algorithm described in Section 4 depends on the number of camera poses that are present in the prior 3D reconstruction, this reconstruction should comprise of enough camera viewpoints and map 3D points to perform an efficient long-term localization.

Figure 5 depicts a comparison of the obtained trajectory for a circular 3 m diameter sequence by our visual stereo SLAM algorithm and the ground truth collected from of a Vicon motion capture system. We can observe that the estimated trajectory is very approximate to the motion capture data. It can also be observed that in some parts of the sequence the motion capture system missed to compute reliable pose estimates, mainly because the retro-reflective marker attached to the robot's waist was partially occluded.

Due to the mentioned discrepancy between the desired trajectory and the real one performed by the robot, the robot was not able to close the loop in this sequence. Therefore, there is a drift between the initial and end position of the robot in the sequence. Figure 6 depicts the final 3D map and keyframes obtained with our stereo visual SLAM system. One can clearly appreciate a circular trajectory of 3 m diameter.

Table 1 shows information about the latest robot's pose in the sequence both for the stereo visual SLAM and motion capture system. According to those results we can observe that the absolute error at the end of the sequence was about 2 cm in the Y axis and 10.80 cm and 18.80 cm for the X and Z axes respectively. The error increased at the end of the sequence, mainly because in the last part of the sequence the robot was facing a challenging low-textured environment. Figure 20(b) depicts one keyframe extracted from this area.

Figure 7 depicts another comparison of our stereo visual SLAM against motion capture data. In this case, the robot performed a 3 m straight line sequence. For this sequence we had always good visibility conditions between the retro-reflective marker attached to the robot and the motion capture camera. We can observe again that both trajectories are very similar. Table 2 shows information of the latest robot's pose in the sequence both

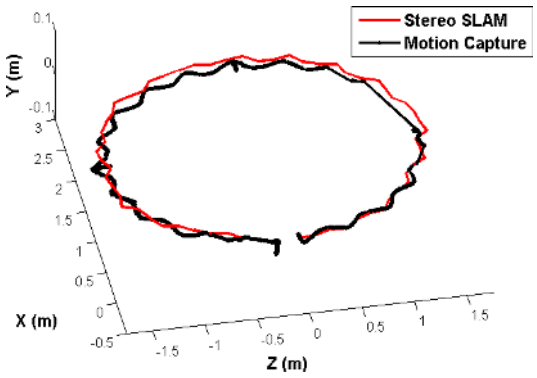


Fig. 5 Camera trajectories for the circle 3 m diameter sequence from the Toulouse dataset. The estimated trajectory of our stereo visual SLAM algorithm is depicted in red, and the ground truth trajectory obtained by the motion capture system is depicted in black. Best viewed in color.

Table 1 Comparison of stereo Visual SLAM and motion capture (MOCAP) camera trajectories for a circular 3 m diameter sequence from the Toulouse dataset.

Camera Pose **E**lement	Final Position Stereo SLAM	Final Position MOCAP	Error $\lvert\epsilon\rvert$ (m)
X (m)	-0.1322	-0.0242	0.1080
Y (m)	0.0018	0.0297	0.0279
Z (m)	-0.5142	-0.3262	0.1880

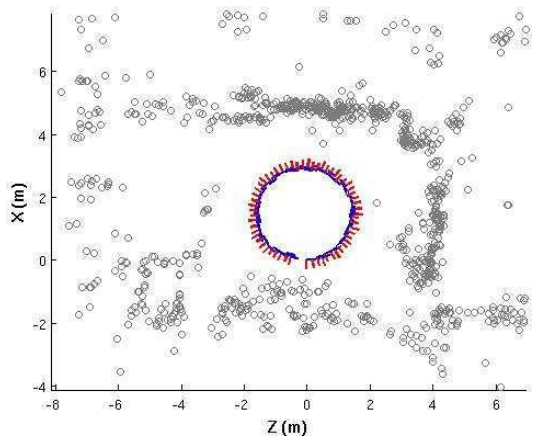


Fig. 6 Stereo Visual SLAM results: Final 3D map and set of reconstructed keyframes for the circle 3 m diameter sequence from the Toulouse dataset.

for the stereo visual SLAM and motion capture system. This time, we can observe that the trajectory estimates for our vision-based method are pretty accurate about the order of few cm. The estimated trajectory length of our method for this sequence was 3.0934 m and for the motion capture system the estimated length was 3.0833 m.

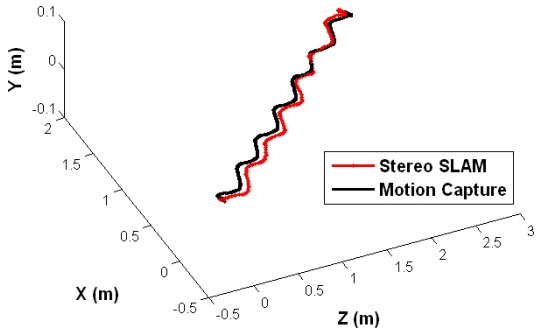


Fig. 7 Camera trajectories for the straight 3 m sequence from the Toulouse dataset. The estimated trajectory of our stereo visual SLAM algorithm is depicted in red, and the ground truth trajectory obtained by the motion capture system is depicted in black. Best viewed in color.

Table 2 Comparison of stereo Visual SLAM and motion capture (MOCAP) camera trajectories for a straight 3 m length sequence from the Toulouse dataset.

Camera Pose Element	Final Position Stereo SLAM	Final Position MOCAP	Error $\lvert\epsilon\rvert$ (m)
X (m)	-1.8357	-1.7780	0.0577
Y (m)	0.0000	-0.0033	0.0033
Z (m)	2.5260	2.5190	0.0070

6.2 Localization Results: Tsukuba Dataset

In this section we evaluate the accuracy and robustness of our monocular vision-based localization algorithm with visibility prediction under different robot trajectories and scenarios. In the Tsukuba dataset, the experiments were performed considering an image resolution of 320×240 and a frame rate of 15 frames per second. For the visibility prediction algorithm we considered the following input parameters of the algorithm: $K = 10$ and $P_t > 0.20$. We chose a threshold value of 2 pixels in the RANSAC process, for determining when a putative match is predicted as an inlier or outlier in the PnP problem.

6.2.1 Square 2 m Size Sequence

In this sequence, the robot performed a 2 m size square in a typical humanoid robotics laboratory. This sequence was designed for capturing different camera viewpoints both in translation and orientation. Firstly, we built a 3D map of the environment by using the stereo visual SLAM algorithm described in Section 3 and performed visibility learning. The resulting 3D map comprises of 935 points and 75 keyframes.

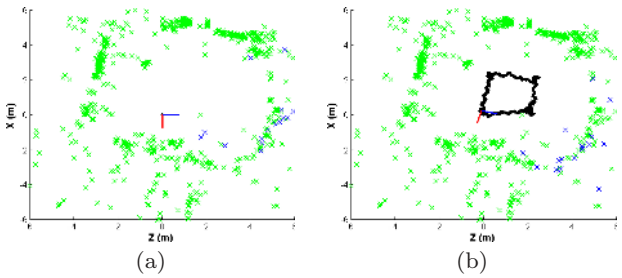
At the start of the sequence, we placed the robot at the origin of the map, and then by using the pattern generator, the robot performed a square of 2 m size. We measured manually the final position of the robot,

Table 3 Square 2 m size monocular vision-based localization results (Tsukuba dataset).

Camera Pose Element	Start Position	Final Position
X (m)	0.0000	0.2320
Y (m)	0.0000	0.0000
Z (m)	0.0000	-0.0092
q_0	1.0000	0.9905
q_x	0.0000	0.0034
q_y	0.0000	0.1375
q_z	0.0000	0.0050

and this position was $(X = 0.14, Y = 0.00, Z = -0.02)$ in meters. Due to the existing drift between the planned trajectory and the real one, the robot was not able to close the loop itself. Then, we validated our vision-based localization algorithm with visibility prediction under a similar square sequence.

Figure 8 depicts the initial and final position of the robot, and the performed trajectory. Table 3 shows the obtained localization results using visibility prediction for this square sequence. According to the results we can see that the localization accuracy is very good, about the order of cm. The differences with respect to the real trajectory for the final position are very small 9 cm, in the X coordinate and about 7 cm in the Z coordinate. While the robot was walking the pattern generator fixed the Y coordinate always to the same value. Therefore, in the PnP problem we add this constraint to speed-up the process, although our algorithm can deal with 6DoF.

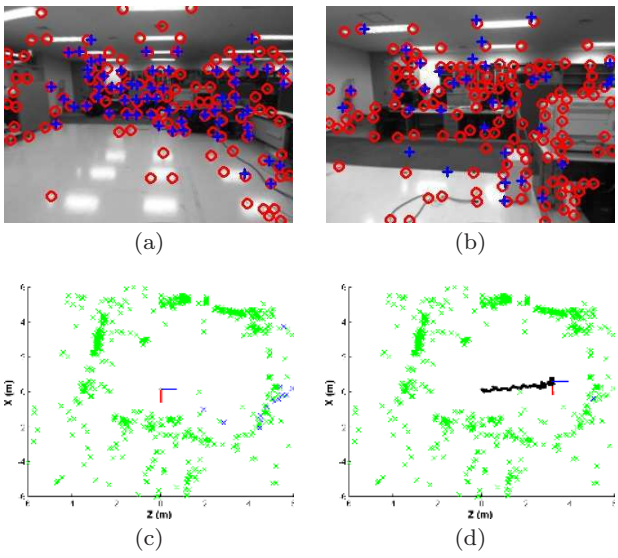


(a) (b)

Fig. 8 Square 2 m size localization results Tsukuba dataset. (a) and (b) depict the initial and final position of the robot and the performed trajectory in the sequence. In these two images the robot trajectory is depicted in black, the visible 3D points are depicted in blue and the rest of 3D points in green. Best viewed in color.

6.2.2 Straight Line 3 m Length Sequence

In this experiment we validated our vision-based localization algorithm under new camera viewpoints that were not captured during the map computation process. The visibility prediction algorithm depends on the number and locations of the keyframes in the prior 3D reconstruction. Therefore, the PnP problem is more difficult to solve in those areas where we have a small density of keyframes. Data association is also more challenging as well, due to the fact that the appearance of new perceived 2D features may not be captured properly by the stored descriptors of the map elements. For this purpose, we planned a sequence in which the robot started in a known position in the 3D map and moved in a straight line of 3 m length. Since in the prior 3D map we have only keyframes in a square 2 m×2 m area, in this experiment we have 1 m length without keyframes. In this new area we should expect from the visibility prediction algorithm lower visibility probabilities for the predicted 3D points than in a well-mapped area where we can have a higher number of keyframes. Figure 9 depicts the initial and final position of the robot in the sequence, and their associated image views with detected 2D features and 3D map re-projections. Table 4 shows the localization results for this experiment.



(a) (b)

(c) (d)

Fig. 9 Straight line 3 m localization results Tsukuba dataset. (a) and (b) depict the initial and final associated image views of the sequence. The red circles are the detected 2D image features, whereas the blue crosses represent the re-projection of predicted visible 3D points. On the other hand, (c) and (d) depict the initial and final position of the robot and the performed trajectory in the sequence. Best viewed in color.

In this sequence, we measured manually the final position of the robot which was 3.0 m in the Z direction and 0.23 m in the X direction. Compared to the obtained localization results we can observe that we have a higher absolute error in the X axis of 33 cm than

Table 4 Straight line 3 m length monocular vision-based localization results (Tsukuba dataset).

Camera Pose Element	Start Position	Final Position
X (m)	0.1191	0.5644
Y (m)	0.0000	0.0000
Z (m)	0.0045	3.1633
q_0	1.0000	0.9994
q_x	0.0000	0.0196
q_y	0.0000	-0.0293
q_z	0.0000	0.0038

in the Z axis, which is about 16 cm for this sequence. These errors are reasonable acceptable, since this area was not captured properly in the map and therefore the PnP problem and data association were more difficult to solve.

Figure 10 depicts information about the inliers ratio and number of RANSAC iterations for the square and straight sequences. As expected, we can observe in Figure 10(c) and Figure 10(d) how the inliers ratio decreases and how the number of RANSAC iterations increases for the straight sequence from the frame 450 approximately. This is because at that point of the sequence the robot started to move into a new area, and therefore both the PnP problem and data association were more difficult to solve. In contrast, the mean inliers ratio for the square sequence 0.9558 is higher than for the straigth sequence one 0.8744. Also, the number of RANSAC iterations is smaller for the square sequence case 2.3808 than for the straight one 7.8144.

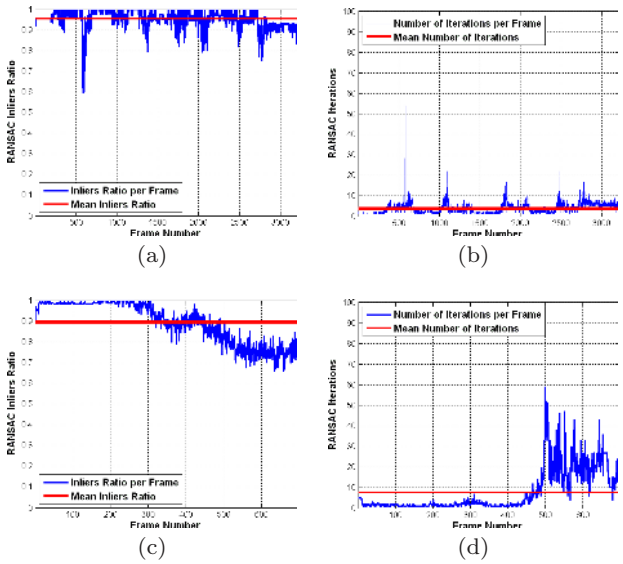


Fig. 10 Comparison of localization results of the square 2 m size sequence versus straight 3 m sequence, using a prior 3D map and visibility prediction (Tsukuba dataset). (a) Inliers ratio % and (b) Number of RANSAC iterations for the square 2 m size sequence. (c) Inliers ratio % and (d) Number of RANSAC iterations for the straight 3 m sequence.

6.2.3 People Moving in front of the Robot

In typical humanoid robotics laboratories is common that while the robot is performing different tasks in the environment, people may pass close to the robot, occluding some areas of the map or even performing human-robot interaction (Dominey et al, 2007). In all the mentioned situations it is important that the robot is always localized correctly in the environment.

In this experiment we placed the robot at the origin of the map and planned a straight sequence of 1 m length while some people were walking in front of the robot, without occluding completely the camera field of view. Even though moving people or objects can occlude some areas of the image and the 3D map, we were able to obtain reliable pose estimates. Outliers are rejected either for apperance information in the data associtions step or by means of RANSAC. Roughly speaking, as long as we have two 2D-3D good correspondences we can estimate the robot's pose. Figure 11(a,b) depicts two frames of the sequence where we can appreciate two persons performing common tasks such as going to the printer or picking up the chessboard pattern. At the same time the students were walking in the environment, the robot was moving 1 m straight from its initial position. Figure 11(c) depicts the initial position of the robot in the experiment, whereas Figure 11(d) depicts the final position.

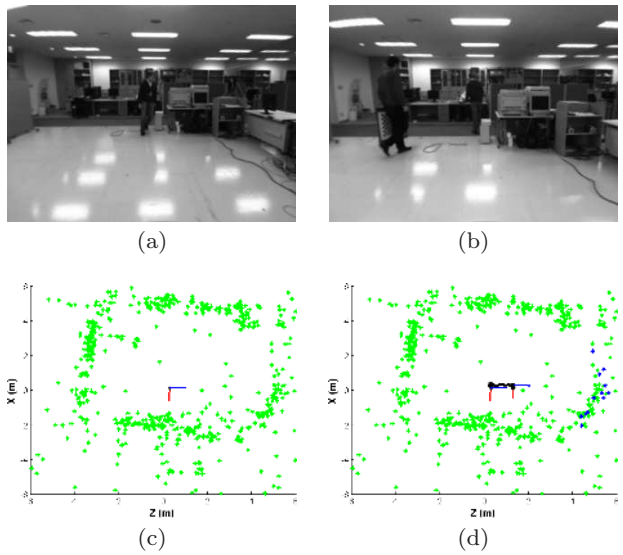


Fig. 11 People moving in front of the robot. The robot performed a 1 m straight line sequence while at the same time some students were walking in the environment, occluding some 3D map points. Best viewed in color.

6.2.4 Robot Kidnapping

In this experiment, the robot was in a known location and then it was suddenly kidnapped, obstructing completely the camera field of view. Although, in the previous sequences (square, straight) the robot did not get lost, it may happen that eventually the robot gets lost if it moves into a new area or the robot is kidnapped as happened in this experiment. In this occasion, for kidnapping recovering, we used the re-localization procedure described in Section 5.1. This re-localization procedure takes an average of 25.27 ms per frame. When the robot was kidnapped we moved the robot 1.40 m to the left, and let the system to re-localize itself.

Figure 12 (a) and (b) depict the moment of kidnapping and after kidnapping. We can observe in (a) that even a large area of the image is occluded we are still able to obtain some good 2D-3D correspondences, and therefore localization estimates. Figure 12 (c) and (d) depict the location of the robot when the kidnapping was going to start and after kidnapping respectively.

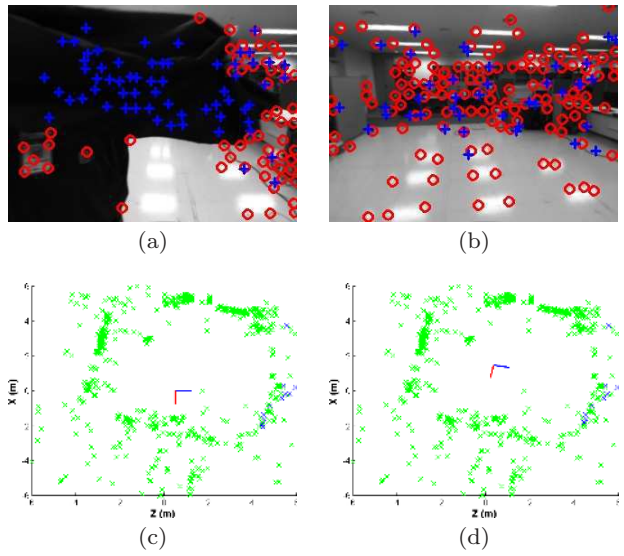


Fig. 12 Robot kidnapping experiment. (a) The moment when the robot was kidnapped. Notice that even a large area of the image is occluded, we are still able to find good 2D-3D correspondences (b) After kidnapping, the robot is re-localized (c) Robot location in the moment of the kidnapping (d) Robot location after kidnapping. Best viewed in color.

6.2.5 Localization Robustness against changes in Lighting Conditions

In this experiment we wanted to evaluate the robustness of our vision-based localization approach and the quality of the reconstructed 3D map against changes in lighting conditions. Even though, most of humanoid robots operate under indoors controlled lighting conditions, it may happen that under special circumstances lighting conditions can change drastically. Invariance to changes in lighting is even much more important for outdoor scenarios where robots have to explore the same area during different hours of a day. Therefore, it is important that even if the lighting conditions change, the localization of the robot in the environment must be robust and accurate.

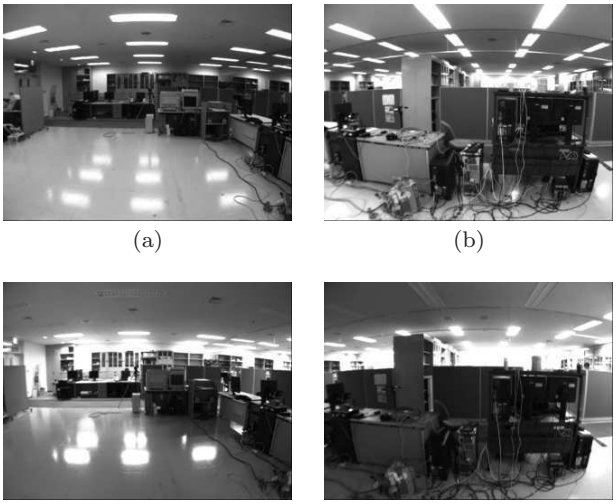
For evaluating the quality of our vision-based localization framework against changes in lighting, the robot performed a square 2 m size trajectory with low-intensity lighting conditions, using a prior 3D map that was obtained in normal lighting conditions. Local image descriptors exhibit some invariance against changes in lighting. Invariance to contrast can be achieved by turning the descriptor into a unit vector. For example in (Mikolajczyk and Schmid, 2005), local image descriptors are evaluated under different image transformations including illumination changes. However, not only the descriptor invariance is important, it is also necessary that the feature detector exhibits high repeatability against these changes. If the feature is not detected, it is not possible to match a 3D map element with the corresponding 2D feature, making the data association more challenging.

Figures 13 depicts two frames of the environment with normal lighting conditions, where the prior 3D reconstruction was done. Figures 13(c,d) depict two frames of approximately the same places of the same environment but under low-intensity lighting conditions. It can be observed the difference in contrast between the two images of the same place under different lighting conditions. Figure 14(a) depicts the square 2 m size performed by the robot under low-intensity lighting conditions. Figure 14(b) shows the inliers ratio score per frame for the experiment. At the beggining of the sequence the inliers ratio score was small. This was because during the initial frames of the sequence the system was trying to obtain a stable pose initialization. Once the initialization process converged, the inliers ratio score increased and the localization was stable.

6.3 Localization Results: Toulouse Dataset

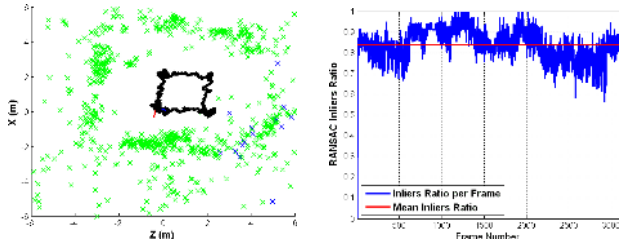
For this dataset, we performed experiments considering an image resolution of 640×480 and a frame rate of 15 frames per second. We also compare our monocular vision-based localization results with respect to the PTAM approach.

Firstly, we obtained a prior 3D reconstruction of the environment from a square 3 m size sequence that



(a) (b) (c) (d)

Fig. 13 (a-b) Two frames of the sequence under normal lighting conditions where the prior 3D map was obtained. (c-d) Two captured frames at approximately the same positions as (a-b) but considering low-intensity lighting conditions. Notice the difference in contrast between the image pairs (a-c) and (b-d) respectively.



(a) (b)

Fig. 14 Evaluation of localization robustness against changes in lighting conditions. (a) The square 2 m size trajectory performed by the robot (b) Inliers ratio % per frame.

was done by the robot. From this prior reconstruction, visibility was learned and this visibility prediction was used for testing the algorithm under different scenarios. The resulting 3D map comprises of 1768 points and 97 keyframes. In general, the set of experiments from this dataset are more challenging than the ones from the Tsukuba dataset. This is mainly because to moving people and some challenging low-textured areas.

6.3.1 Square 3 m Size Sequence

In this experiment we evaluated our localization framework in a square sequence but including dynamic objects such as people. These dynamic objects were not present in the map sequence, and therefore in the evaluation sequence, these objects can occlude some visible 3D points. We considered again the same input parameters for the visibility prediction algorithm as in the Tsukuba experiments, i.e. $K = 10$ and $P_t > 0.20$. However, in order to cope with the new image resolution we chose a threshold value of 4 pixels in the RANSAC process.

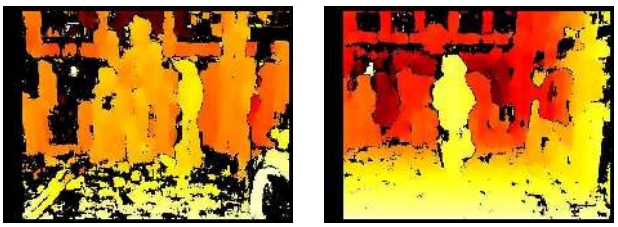
Figures 15(a,b) depict two frames from the sequence where some people are walking in the environment occluding some visible 3D points from the prior map. Figure 15(a) depicts one particular area of the sequence in which vision-based localization is challenging. This is due to the fact that in this area there is a lack of highly textured features. Most of the features are detected in the foliage or around the windows. Due to this lack of texture, the resulting 3D reconstruction in this area can contain higher errors than in other more textured areas of the sequences, since the disparity maps obtained during the map computation are much more sparser and noisier than in other areas. Furthermore, the person walking occludes some predicted visible 3D points. Then, when the robot moved to another more textured area (Figure 15(b)), the localization algorithm was able to find correct pose estimates even in the presence of people occluding some predicted visible 3D points. Figure 16 depicts the two associated

(a) (b)

Fig. 15 (a) In this area localization is more difficult, mainly due to the lack of textured features. (b) Even though there are some persons walking in the environment occluding some visible 3D points, the algorithm is able to find correct pose estimates without problems. The red circles are the detected 2D features, the blue crosses represent the re-projection of predicted visible 3D points. The set of inliers putatives after solving the PnP problem are represented by cyan rectangles, whereas the outliers are represented as yellow rectangles. Best viewed in color.

disparity maps for the frames shown in Figure 15. As mentioned before, it can be observed how the disparity map is sparser and noisier for the low-textured area (Figure 16(a)) than for the textured one (Figure 16(b)).

Figure 17(a) depicts the performed square 3 m size trajectory done by the robot. In order to stand out the accuracy of the localization per area, the trajectory is depicted in a typical *cool* color space. In this case, the value in the color space is the inliers ratio per frame in the PnP problem. This inliers ratio can be interpreted



(a) (b)

Fig. 16 Disparity maps of two frames. Disparity is coded by using a *hot* color space representation. In this representation, close 3D points to the camera are depicted in yellow, whereas far points are depicted in red. Best viewed in color.

as an indicator of how good was the localization or how easy to solve was the PnP problem. Other quantities could have been used as for example the covariance result from the PnP problem. We can observe that the inliers ratio tend to decrease when the robot was facing the area depicted by Figure 15(a). After this area, the localization was more robust and the inliers ratio increases. In average the inliers ratio per frame was 0.76 for this sequence.

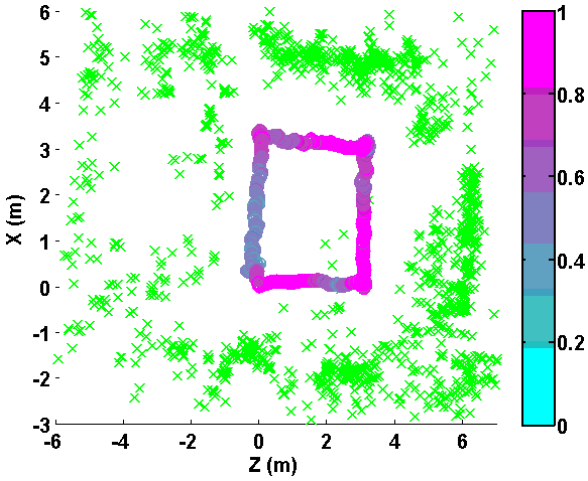


Fig. 17 Square 3 m Size Localization Results. The trajectory is coded considering a *cool* color space by means of the inliers ratio per frame in the PnP problem. Best viewed in color.

6.3.2 Circle 3 m Diameter Sequence

Now, we evaluate localization considering very different viewpoints from the ones that were captured in the map sequence. In particular, the robot performed a circular 3 m diameter sequence, including very different viewpoints that were not captured in the prior 3D reconstruction. In addition, this experiment was done in a different day than the prior 3D reconstruction. Therefore, there are some changes in the environment, such as for example boxes or a tripod placed in different positions from the original map sequence. Introducing changes in the environment, implies a more difficult localization since our map and localization assume rigid SfM. For example, Figure 18 depicts one example of these changes in the environment. We consider the following input parameters for the visibility prediction algorithm: $K = 10$ and $P_t > 0.05$. The probability threshold is reduced in this case, since in this scenario we had very different camera viewpoints than the ones captured in the map computation sequence and therefore the weights given by the learned kernel function will be much lower. Figure 19(a) depicts the performed circular 3 m diameter sequence done by the robot. The trajectory is depicted again considering a *cool* color space coded by means of the inliers ratio per frame in the PnP problem. Again we can observe that the lowest inliers ratios were obtained when the robot was facing the low-textured area depicted by Figure 15(a). In average the inliers ratio per frame was 0.49 for this sequence. Although the inliers ratio in this scenario is smaller compared to the square sequence, we need to take into account that viewpoints are very different compared to the map sequence and that some changes in the environment were introduced. Despite of these facts, we are able to obtain a robust localization in real-time, as we will show in the timing evaluation section.

(a) (b)

Fig. 18 (a) An image from the map computation sequence (b) An image from approximately the same place as image as (a) but for the circle sequence. Since this sequence was captured in a different day than the map one, there are some changes in the environment, e.g. tripod, chair and white box.

6.3.3 Comparison to PTAM

In this section we compare our localization results with respect to PTAM under the same circular sequence from the previous experiment. At the beginning of the sequence, we obtained good results with PTAM, since it was able to estimate an accurate camera trajectory. However, when the robot performed pure rotation steps in a low-textured area the pose estimation error increased considerably and PTAM had problems adding

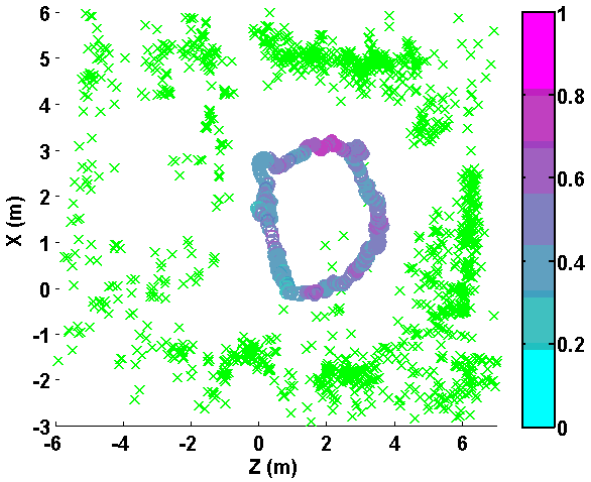


Fig. 19 Circle 3 m Diameter Localization Results. The trajectory is coded considering a *cool* color space by means of the inliers ratio per frame in the PnP problem. Best viewed in color.

new 3D points to the map. Figure 20(a) depicts one frame where PTAM tracking was successful. However, when the robot moved to a low-textured area (Figure 20(b)) PTAM tracking got lost.

(a) (b)

Fig. 20 PTAM Tracking Results: (a) One frame of the circular sequence where PTAM tracking was successful (b) One frame where PTAM tracking had severe problems and new 3D map points can not be added to the map.

Figure 21 depicts a comparison of the estimated robot trajectory considering PTAM, the motion capture system and monocular vision-based localization results with a prior 3D map. For the monocular vision-based localizaton results with visibility prediction, we consider two different prior maps: one obtained from a square sequence as described in Section 6.3.2 and another one obtained from the circular 3 m diameter sequence. We can observe that PTAM obtained good trajectory estimates at the beginning of the sequence, but as soon as the robot was doing pure rotation steps the error increased considerably. It can also be observed that the monocular localization results with a prior 3D map obtained a very similar trajectory compared to the motion capture system.

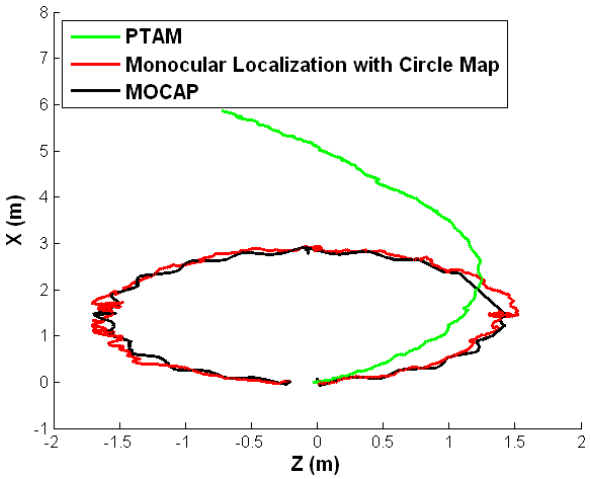


Fig. 21 Comparison to PTAM Localization Results. Best viewed in color.

Now, we will explain in detail the main reasons why our proposed vision-based localization system obtained much better results than PTAM. Notice here, that we need to distinguish between the map computation process and the localization with a prior map step. In our system we use stereo and monocular vision depending on the algorithm stage. In contrast, PTAM always uses monocular vision for the mapping and posterior localization with the computed map.

– **Mapping**:

In the context of humanoid robots, most of robotics platforms are already equipped with stereo vision systems, and therefore, we think it is preferable using a stereo-vision framework for building an accurate 3D map of the environment than a monocular one. Then, once the map is computed, we have shown that is possible to perform an efficient monocular vision-based localization. In general, stereo-vision systems will always be more accurate than monocular ones, since they use more information thanks to a second camera and they do not suffer from unobservability problems of recovering the scale of a 3D point from 2D image projections.

PTAM performance is highly dependent on the initialization module and localization results vary considerably according to this initialization as reported in (Wendel et al, 2011). The initialization in PTAM is done by simulating a virtual stereo pair. Then, the initial 3D points are obtained from triangulation between 2D image correspondences in the two images. The virtual stereo pair simulation may be tedious to be performed by a humanoid robot using a single camera, since a pre-defined walking motion should be done at the beginning of each sequence in order to simulate a virtual stereo providing enough baseline. However, in the case of stereo-vision, this initialization is directly

obtained from the two views and stereo rig calibration parameters.

– **Localization in a Prior 3D Map**:

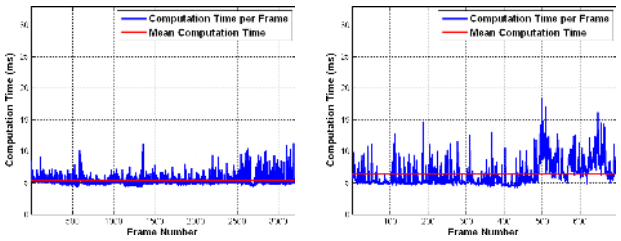
Assuming that we took special care in the map computation process and PTAM was able to compute an accurate 3D map of the environment, our monocular vision-based localization framework also provides several benefits with respect to PTAM localization module such as accuracy, speed and scalability. In particular, PTAM does not perform any kind of visibility prediction. At each time a new frame is acquired from the camera, a prior pose estimate is generated from a motion model. Then, map points are projected into the image according to the prior camera pose estimate. Similar to our approach, after data association a cost function is minimized and the camera pose estimate is updated.

The above implies that PTAM can not deal with occlusions since it assumes a *transparent world* and needs to back-project each 3D point onto the image plane. This can be computationally expensive for very large 3D reconstructions and prone to failure as demonstrated in (Alcantarilla et al, 2010, 2011). On the other hand, thanks to the use of the visibility prediction algorithm, we can perform an efficient data association that takes into account occlusions (the algorithm *learns* the occlusions) and allows to estimate an accurate camera pose even in the presence of large-scale and cluttered 3D environments.

6.4 Timing Evaluation

We show a timing evaluation of our vision-based localization algorithm for both the Tsukuba and Toulouse datasets. All timing results in this section were obtained on a Core i7 2.87GHz desktop computer using a single CPU. Notice that even though the HRP-2 has an onboard computer, this computer is not powerful enough to run advanced computer vision algorithms. However, the images from the robot can be send to an external more powerful computer by an Ethernet or Wifi (802.11 n 5 Ghz) links.

Figure 22 depicts timing results for the localization experiments of the square and straight sequence from the Tsukuba dataset. We can observe that in average the mean computation time for the square sequence, 5.35 ms, was slightly smaller than for the straight one, 6.49 ms. For a faster localization, we only detect 2D features at the finest scale-space level. In the environment we carried out our experiments, we observed that with one single-scale level we have enough amount of features to perform robust localization.



(a) (b)

Fig. 22 Monocular Vision-Based Localization Timing Evaluation Tsukuba Dataset: (a) Computation times per frame for the 2 m square sequence (b) Computation times per frame for the 3 m straight sequence.

Table 5 Monocular vision-based localization mean computation times per frame (Tsukuba dataset). For this dataset the image resolution was 320×240 pixels.

Localization Step	Square Time (ms)	Straight Time (ms)
Initialization	1636.86	1641.99
Undistortion and Rectification	0.76	0.87
Feature Detector	2.43	2.62
Feature Descriptor (16)	0.94	1.04
Re-Projection 3D Points	0.12	0.14
Data Association	0.10	0.09
Pose Estimation	1.00	1.72
Total per Frame	**5.35**	**6.49**

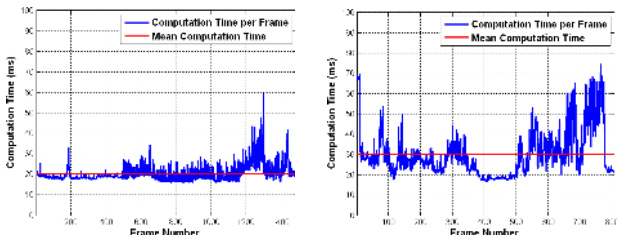
Table 5 shows mean computation times for the analyzed experiments, but describing timing evaluation for the main steps involved in the localization algorithm. In general, most time consuming steps per frame are feature detection, descriptors computation and pose estimation. Initialization only takes place during the first frame or an initial transitory time of the sequence until the robot detects that it is in a known area with high confidence.

Figure 23 depicts timing results for the localization experiments of the square and circular sequence from the Toulouse dataset. For the square sequence we obtained a mean computation time per frame of 20.31 ms. For the circular sequence the computation time is higher (30.36 ms). This is mainly because the PnP problem was more difficult to solve, due to the fact that viewpoints are very different from the ones captured in the map sequence and the changes in the

Table 6 Monocular vision-based localization mean computation times per frame (Toulouse dataset). For this dataset the image resolution was 640 × 480 pixels.

Localization Step	Square Time (ms)	Circular Time (ms)
Initialization	2540.93	2723.15
Undistortion and Rectification	3.36	2.95
Feature Detector	10.28	9.81
Feature Descriptor (16)	2.79	2.15
Re-Projection 3D Points	0.29	0.28
Data Association	0.55	0.51
Pose Estimation	3.02	14.64
Total per Frame	**20.31**	**30.36**

environment also make the PnP problem more challenging.

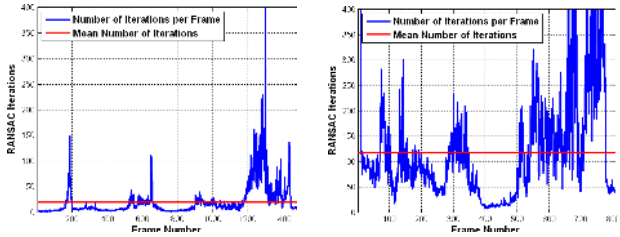


(a) (b)

Fig. 23 Monocular vision-based localization timing evaluation Toulouse dataset: (a) Computation times for the 3 m size square sequence (b) Computation times for the 3 m diameter circular sequence.

Table 6 shows mean computation times per frame for both sequences of the Toulouse dataset. Since in the Toulouse dataset we are using a 640×480 image resolution, the feature detection and description steps are more time consuming than for the Tsukuba dataset. In the same way, since the image resolution is higher, the detected number of 2D features is also higher and therefore the PnP problem has a higher number of putative correspondences. In those areas where we have enough textured features, the PnP problem is solved very fast in real-time. However, in some particular areas where it may be difficult to find good 2D-3D correspondences the PnP problem can take more time to be solved efficiently (e.g. low-textured areas of the circular sequence).

In general, with a small image resolution 320×240 we can obtain accurate localization results in few ms. With a higher resolution such as 640×480 the localization results can be very accurate, although the computation time will also increase considerably. For all the analyzed experiments, mean computation times per frame are below real-time demands (30 Hz). If certain applications have some time restrictions, one can always fix a smaller threshold for the number of iterations of the RANSAC step. Usually if the set of putative matches is good, only few iterations are necessary to solve the PnP problem efficiently. Figure 24 depicts the number of RANSAC iterations for the square and circular sequence from the Toulouse dataset. In those experiments we fixed a maximum threshold of 400 iterations in the RANSAC process.



(a) (b)

Fig. 24 Monocular vision-based localization timing evaluation Toulouse dataset: (a) Number of RANSAC iterations per frame for the 3 m size square sequence (b) Number of RANSAC iterations per frame for the 3 m diameter circular sequence.

7 Conclusions and Future Work

In this paper, we have presented a vision-based localization algorithm that works in real-time (even faster than 30 Hz) and provides localization accuracy about the order of cm. We first build a 3D map of the environment by using stereo visual SLAM techniques, and perform visibility learning over the prior 3D reconstruction. Then, for fast vision-based localization we use visibility prediction techniques for solving the PnP problem and obtaining the location of the robot with respect to a global coordinate frame. We measured the accuracy of our localization algorithm by comparing the estimated trajectory of the robot with respect to ground truth data obtained by a highly accurate motion capture system. We also compared our algorithm with respect to other well-known state of the art SfM algorithms such as PTAM, showing the benefits of our approach.

In this work, we have mainly put our focus in real-time vision-based localization. However, we think that

the accuracy in localization can be increased if we fuse the information from our vision-based localization with the odometry information of the robot. Also the image resolution and length of the descriptors can be increased, but the price to pay is higher computational demands, that may prevent the algorithm from real-time performance. In the near future, we will study the localization performance with respect to different combinations of feature detectors-descriptors for humanoid robotics applications, similar as the study performed in (Gil et al, 2010) for visual SLAM settings.

In addition, we plan to perform a novel control architecture for humanoid robots where the current vision-based localization is used by the robot controller and planner trajectory. In this way, we think humanoid will be able to perform complex tasks in challenging robotics scenarios where an accurate localization of the robot is necessary. For that goal, we think that the fusion of vision-based algorithms with inertial sensors can be of interest as proposed in (Strelow and Singh, 2004).

Acknowledgments.

This work has been financed with funds from the Ministerio de Economía y Competitividad through the project ADD-Gaze (TRA2011-29001-C04-01), as well as from the Comunidad de Madrid through the project Robocity2030 (CAM-S-0505/DPI/ 000176). The authors would also like to thank the Joint French-Japanese Robotics Laboratory (JRL), CNRS/AIST, Tsukuba,Japan.

References

Agarwal S., Snavely N., Simon I., Seitz S.M., Szeliski R. (2009) Building Rome in a Day. In: Intl. Conf. on Computer Vision (ICCV)

Alcantarilla P.F., Oh S., Mariottini G., Bergasa L.M., Dellaert F. (2010) Learning visibility of landmarks for vision-based localization. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Anchorage, AK, USA, pp 4881–4888

Alcantarilla P.F., Ni K., Bergasa L.M., Dellaert F. (2011) Visibility learning in large-scale urban environment. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Shanghai, China

Angeli A., Filliat D., Doncieux S., Meyer J.A. (2008) Fast and Incremental Method for Loop-Closure Detection using Bags of Visual Words. IEEE Trans. Robotics 24:1027–1037

Ansar A., Danilidis K. (2003) Linear pose estimation from points or lines. IEEE Trans. Pattern. Anal. Machine Intell. 25(4):1–12

Atkeson C., Moore A., Schaal S. (1997) Locally weighted learning. AI Review 11:11–73

Bay H., Ess A., Tuytelaars T., Gool L.V. (2008) SURF: Speeded up robust features. Computer Vision and Image Understanding 110(3):346–359

Blösch M., Weiss S., Scaramuzza D., Siegwart R. (2010) Vision based MAV navigation in unknown and unstructured environments. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Anchorage, AK, USA

Bohg J., Holst C., Huebner K., Ralph M., Rasolzadeh B., Song D., Kragic D. (2009) Towards grasp-oriented visual perception for humanoid robots. Intl. J. of Humanoid Robotics 6(3):387–434

Bolles R., Fischler M. (1981) A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In: Intl. Joint Conf. on AI (IJCAI), Vancouver, Canada, pp 637–643

Bouguet J. (2008a) The calibration toolbox for Matlab, example 5: Stereo rectification algorithm (code and instructions only)

Bouguet J. (2008b) Documentation: Camera Calibration Toolbox for Matlab. URL `www.vision.caltech.edu/bouguetj/calib\_doc/`

Byröd M., Åström K. (2010) Conjugate gradient bundle adjustment. In: Eur. Conf. on Computer Vision (ECCV)

Checklov D., Mayol-Cuevas W., Calway A. (2008) Appearance based indexing for relocalisation in real-time visual SLAM. In: British Machine Vision Conf. (BMVC), Leeds, UK

Cummins M., Newman P. (2008) FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. Intl. J. of Robotics Research 27(6):647–665

Davison A.J., Reid I.D., Molton N.D., Stasse O. (2007) MonoSLAM: Real-time single camera SLAM. IEEE Trans. Pattern. Anal. Machine. Intell. 29(6): 1052–1067.

Dellaert F., Kaess M. (2006) Square Root SAM: Simultaneous localization and mapping via square root information smoothing. Intl. J. of Robotics Research 25(12):1181–1203

Dominey P., Mallet A., Yoshida E. (2007) Progress in programming the HRP-2 humanoid using spoken language. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Rome, Italy

Durrant-White H., Bailey T. (2006) Simultaneous localization and mapping SLAM: part 1. IEEE Robotics and Automation Magazine 13(3):99–110

Foissote T., Stasse O., Wieber P., Escande A., Kheddar A. (2010) Autonomous 3D object modeling by a humanoid using an optimization-driven next-best-view formulation. Intl. J. of Humanoid Robotics 7(3):

407–428.

Geiger A., Roser M., Urtasun R. (2010) Efficient large-scale stereo matching. In: Asian Conf. on Computer Vision (ACCV), Queenstown, New Zealand

Gil A., Mozos O., Ballesta M., Reinoso O. (2010) A comparative evaluation of interest point detectors and local descriptors for visual SLAM. Machine Vision and Applications 21(6):905–920

Harris C., Stephens M. (1988) A combined corner and edge detector. In: Proc. Fourth Alvey Vision Conference, pp 147–151

Hartley R. (1999) Theory and practice of projective rectification. Intl. J. of Computer Vision 35:115–127

Hartley R., Zisserman A. (2000) Multiple View Geometry in Computer Vision. Cambridge University Press

Hirschmüller H. (2008) Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern. Anal. Machine. Intell. 30(2):328–341

Hornung A., Wurm K., Bennewitz M. (2010) Humanoid robot localization in complex indoor environments. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), Taipei, Taiwan

Jian Y.D., Balcan D., Dellaert F. (2011) Generalized subgraph preconditioners for large-scale bundle adjustment. In: Intl. Conf. on Computer Vision (ICCV), Barcelona, Spain

Kaess M., Ni K., Dellaert F. (2009) Flow separation for fast and robust stereo odometry. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Kobe, Japan

Kagami S., Nishiwaki K., Kuffner J., Thompson S., Chestnutt J., Stilman M., Michel P. (2005) Humanoid HRP2-DHRC for autonomous and interactive behavior. In: Proc. of the Intl. Symp. of Robotics Research (ISRR), pp 103–117

Kaneko K., Kanehiro F., Kajita S., Hirukawa H., Kawasaki T., Hirata M., Akachi K., Isozumi T. (2004) Humanoid robot HRP-2. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), New Orleans, USA, pp 1083–1090

Klein G., Murray D.W. (2007) Parallel tracking and mapping for small AR workspaces. In: IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR), Nara, Japan

Konolige K. (1997) Small vision system: Hardware and implementation. Proc. of the Intl. Symp. of Robotics Research (ISRR) pp 111–116

Konolige K., Agrawal M. (2008) FrameSLAM: from bundle adjustment to real-time visual mapping. IEEE Trans. Robotics 24(5):1066–1077

Kwak N., Stasse O., Foissotte T., Yokoi K. (2009) 3D grid and particle based SLAM for a humanoid robot. In: IEEE-RAS Intl. Conference on Humanoid Robots, Paris, France, pp 62–67

Li L., Socher R., Fei-Fei L. (2009) Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA

Lourakis M.A. (2004) levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. [web page] `http://www.ics.forth.gr/~lourakis/levmar/`

Lourakis M.A., Argyros A. (2009) SBA: A software package for generic sparse bundle adjustment. ACM Trans. Math Software 36(1):1–30

Lu C., Hager G., Mjolsness E. (2000) Fast and globally convergent pose estimation from video images. IEEE Trans. Pattern. Anal. Machine Intell. 22(6):610–622

Marquardt D. (1963) An algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics (SIAP) 11(2):431–441

Mei C., Sibley G., Cummins M., Newman P., Reid I. (2010) REAL: A system for large-scale mapping in constant-time using stereo. Intl. J. of Computer Vision 94(2): 198–214

Michel P., Chestnutt J., Kagami S., Nishiwaki K., Kuffner J., Kanade T. (2007) GPU-accelerated Real-Time 3D Tracking for Humanoid Locomotion and Stair Climbing. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), San Diego, CA, USA, pp 463–469

Mikolajczyk K., Schmid C. (2005) A performance evaluation of local descriptors. IEEE Trans. Pattern. Anal. Machine Intell. 27(10):1615–1630

Mouragnon E., Lhuillier M., Dhome M., Dekeyser F., Sayd P. (2009) Generic and real-time structure from motion using Local Bundle Adjustment. Image and Vision Computing 27:1178–1193

Nistér D., Naroditsky O., Bergen J. (2004) Visual Odometry. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)

Ozawa R., Takaoka Y., Kida Y., Nishiwaki K., Chestnutt J., Kuffner J. (2007) Using visual odometry to create 3D maps for online footstep planning. In: IEEE Intl. Conference on Systems, Man and Cybernetics, Hawaii, USA, pp 2643–2648

Perrin N., Stasse O., Lamiraux F., Yoshida E. (2010) Approximation of feasibility tests for reactive walk on HRP-2. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Anchorage, AK, pp 4243–4248

Pretto A., Menegatti E., Bennewitz M., Burgard W., Pagello E. (2009) A visual odometry framework robust to motion blur. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Kobe, Japan

Scharstein D., Szeliski R. (2002) A taxonomy and evaluation of dense two-frame stereo correspondence al-

gorithms. Intl. J. of Computer Vision 47:7–42
Schweighofer G., Pinz A. (2006) Fast and globally convergent structure and motion estimation for general camera models. In: British Machine Vision Conf. (BMVC)
Schweighofer G., Pinz A. (2008) Globally optimal O(n) solution to the PnP problem for general camera models. In: British Machine Vision Conf. (BMVC)
Snavely N., Seitz S.M., Szeliski R. (2006) Photo Tourism: Exploring image collections in 3D. In: SIGGRAPH
Stachniss C., Bennewitz M., Grisetti G., Behnke S., Burgard W. (2008) How to learn accurate grid maps with a humanoid. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Pasadena, CA, USA
Stasse O., Davison A., Sellaouti R., Yokoi K. (2006) Real-time 3D SLAM for a humanoid robot considering pattern generator information. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), Beijing, China, pp 348–355
Stasse O., Larlus D., Lagarde B., Escande A., Saidi F., Kheddar A., Yokoi K., Jurie F. (2007) Towards Autonomous Object Reconstruction for Visual Search by the Humanoid Robot HRP-2. In: IEEE-RAS Intl. Conference on Humanoid Robots, Pittsburg, USA, pp 151–158
Stasse O., Verrelst B., Wieber P., Vanderborght B., Evrard P., Kheddar A., Yokoi K. (2008) Modular architecture for humanoid walking pattern prototyping and experiments. Advanced Robotics, Special Issue on Middleware for Robotics–Software and Hardware Module in Robotics System 22(6):589–611
Strasdat H., Montiel J.M.M., Davison A. (2010) Real-time monocular SLAM: Why filter? In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Anchorage, USA
Strelow D., Singh S. (2004) Motion estimation from image and inertial measurements. Intl. J. of Robotics Research 23(12):1157–1195
Triggs B., McLauchlan P., Hartley R., Fitzgibbon A. (1999) Bundle adjustment – a modern synthesis. In: Triggs W., Zisserman A., Szeliski R. (eds) Vision Algorithms: Theory and Practice, Springer Verlag, LNCS, pp 298–375
Wendel A., Irschara A., Bischof H. (2011) Natural landmark-based monocular localization for MAVs. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Shanghai, China, pp 5792–5799
Williams B., Klein G., Reid I. (2007) Real-time SLAM relocalisation. In: Intl. Conf. on Computer Vision (ICCV)