Real-time 3D SLAM for Humanoid Robot considering Pattern Generator Information

Olivier Stasse*, Andrew J. Davison†, Ramzi Sellaouti* and Kazuhito Yokoi*
*JRL, CNRS/AIST, ISRI
National Institute of Advanced Industrial and Science Technology
AIST Central 2, Umezono 1-1-1, Tsukuba, Ibaraki, Japan 302-8568
Email: {olivier.stasse,ramzi.sellouati,kazuhito.yokoi}@aist.go.jp
†Department of Computing
Imperial College London
180 Queen's Gate South Kensington Campus SW7 2AZ, UK
Email: ajd@doc.ic.ac.uk

Abstract— **Humanoid robotics and SLAM (Simultaneous Localisation and Mapping) are certainly two of the most significant themes of the current worldwide robotics research effort, but the two fields have up until now largely run independent parallel paths, despite the obvious benefit to be gained in joining the two. The next major step forward in humanoid robotics will be increased autonomy, and the ability of a robot to create its own world map on the fly will be a significant enabling technology. Meanwhile, SLAM techniques have found most success with robot platforms and sensor configurations which are outside of the humanoid domain. Humanoid robots move with high linear and angular accelerations in full 3D, and normally only vision is available as an outward-looking sensor. Building on recently published work on monocular SLAM using vision, and on pattern generation, we show that real-time SLAM for a humanoid can indeed be achieved. Using HRP-2, we present results in which a sparse 3D map of visual landmarks is acquired on the fly using a single camera and demonstrated loop closing and drift-free 3D motion estimation within a typical cluttered indoor environment. This is achieved by tightly coupling the pattern generator, the robot odometry and inertial sensing to aid visual mapping within a standard EKF framework. To our knowledge this is the first implementation of real-time 3D SLAM for a humanoid robot able to demonstrate loop closing.**

I. Introduction

The long term target of our humanoid research is to improve the capabilities of humanoid robot in order to perform a larger set of tasks in normal environments. To date, this has been achieved by mixing autonomous behaviors and human teleoperation, such demonstrated during the EXPO 2005 AICHI JAPAN universal exposition [1]. In this demonstration, the robot was able to precisely detect and grasp a can, throw it in the garbage, take a bag, and put it on a table. The can, the garbage and the table are detected using a model based approach and thus the robot knows its own location only in the object reference frame. In order to increase the capabilities of the robot, in this paper we describe a SLAM algorithm for self-localization which has the capability to permit global motion estimation and environment interaction *autonomously*. This is an application of SLAM which differs somewhat from standard SLAM systems for exploring robots. For teleoperation, autonomous local reactivity is a crucial issue, as well as robustness for dealing with changing human environment. Most importantly in this context, SLAM is a subcomponent of a whole system, and the self-localization and map building is not the main target but a necessary functionality. Therefore a self-localization process for such kind of task should be 3D, real-time, robust and with low CPU cost. For those reasons, this work focus on a local environment. Regarding exploration of large space, and environment mapping the timing constraints are less severe and we think that the large body of work already existing would be pertinent [2].

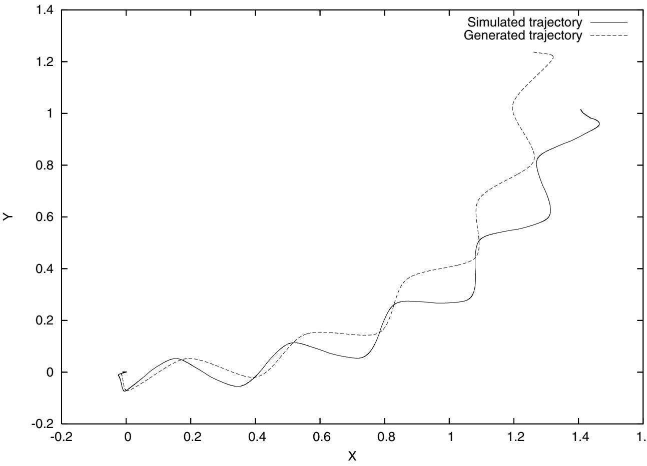


Fig. 1. Drift between generated and simulated waist trajectories

Considering real-time SLAM on humanoid robot they are few other works mainly performed by Thompson and Kagami [3] and Ozawa [4] on H-7 and HRP-2. Their approach is mainly based on 3D visual odometry, and uses dense feature maps to get the position of the camera. A well-known drawback of this incremental approach is the drift created by the accumulation of error. In order to maintain a homogeneous evaluation of the 3D position of a robot, it is mandatory to keep up-to-date information of the correlations between the current state of the camera and the landmarks. Simultaneously building a dense representation of the world and an accurate positioning of the camera seems to be quite difficult to achieve in real-time. Thus in our approach we focus on positioning, and use a sparse representation of the world to obtain an accurate real-time result.

It is worth mention the work of Seara et al. [5] who used the EKF framework beyond self-localization for gaze control of a walking humanoid. This is one of the few pieces of work attempted on coupling walking and vision while performing SLAM. However the main results were done in simulation and in the 2D case.

In this paper, we make use of the monocular SLAM method of Davison [6], but take advantage of the fact that much more is known in advance about the motion of the humanoid than the free camera considered in that work. Specifically we use the planning information from the robot's pattern generator for walking motions, and channel this into the SLAM EKF as observations of the robot's motion. In order to extract as much information as possible to update the Kalman filter, we re-implemented the accurate walking pattern generator described in [7]. We present its implementation, the associated measurement models and its use both in simulation and into a real HRP-2 humanoid robot. It is also shown that the particularity of humanoid pattern-generator creates a parallax effect useful for SLAM. We do not need to use a set of prior-known features as in Davison's approach, because the pattern generator's input gives enough information for completely automatic initialization.

The remainder of this paper is organized as follows. First we review the pattern generator with preview control principles used in this paper. The cause of the drift between the pattern generator and its realization are highlighted. In second we recall the principles of the monocular SLAM used, and detail the used initialization procedure and the measurement model. In third the integration of the information given by the pattern generator, the vision and other sensors is explained. We finally present some experiments.

II. Pattern generator with preview control

We briefly recall the result presented by Kajita in [7]. Most importantly, the equations based on the cart-table model used to generate the ZMP and waist trajectories according to a set of relative steps are given. The resulting ZMP is fed to the stabilizer provided with the robot. Those equations are used to design a measurement model in the SLAM section. Indeed we assume here a walking pattern where the camera is not moving with respect to the waist.

A. The cart-table model

In this approach, the goal is to generate a trajectory of the robot's waist based on a given Zero Momentum Point (ZMP) trajectory. More specially a constraint control is applied to an inverted pendulum such that the mass should move along an arbitrary predefined plane. The subsequent model is called the *Three-Dimensional Linear Inverted Pendulum Model* (3D-LIPM). Given the position and the acceleration of the Center of Mass (CoM) of the inverse pendulum it is possible to compute its ZMP.

The problem of pattern generation is then for a given set of ZMP position to find the CoM trajectory. It is then an inverse problem. From [7], the system can be written in its discretize form as:

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

where

$$\begin{aligned} \mathbf{x(k)} &\equiv [x(k)\ \dot{x}(k)\ \ddot{x}(kT)]^T, \\ u(k) &\equiv \mathbf{u}_x(kT), \\ p(k) &\equiv \mathbf{p}_x(kT), \\ \mathbf{A} &\equiv \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{B} &\equiv \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, \\ \mathbf{C} &\equiv [1\ 0\ -z_c/g] \end{aligned} \tag{2}$$

where $x(k)$ is the CoM position for the x-axis, $u(k)$ is the command on the acceleration of the CoM along the x-axis, z_c the height of the CoM, $p(k)$ the ZMP position, T the sampling period, and g is the gravity constant. As the optimal solution depend upon the future reference, Kajita proposed to apply a *preview control* on this dynamical system. Using the following performance index:

$$J = \sum_{i=k}^{\infty} \{Q_e e(i)^2 + \Delta\mathbf{x}^T \mathbf{Q}_x \Delta\mathbf{x}(i) + R\Delta u^2(i)\} \tag{3}$$

where $e(i) = p(i) - p^{ref}(i)$, the optimal controller which minimizes eq.(3) is given by:

$$u(k) = -G_1 \sum_{i=0}^{k} e(i) - G_2\mathbf{x}(k) - \sum_{j=1}^{N_L} G_p(j) p^{ref}(k+j) \tag{4}$$

where G_1,G_2, and $G_p(j)$ are the gains computed from the weights $Q_e, \mathbf{Q}_x, R$ and the system parameters of eq.(2), and N_L is the discrete size of the preview control window. Because they are some discrepancies between the robot's CoM and the one provided by the cart-table model, Kajita proposed to use a second stage of preview control based on the ZMP computed from the robot's multi-body model.

The waist attitude from the motion model viewpoint is moving only for the yaw. The waist orientation start from the last support foot's orientation to the next support foot's orientation. Its evolution follows a 3-rd order polynomial.

From this brief introduction, it is possible to see that several information are at hand: position, speed, acceleration of the waist, waist attitude, and those for the present time and during the preview control window. From the control viewpoint, the possible source of errors are two-folds: the quality of the ZMP preview control, and the quality of the stabilizer in charge of realizing a desired ZMP. Creating a new motion model from eq.(2) was considered. However the Jacobian computation induced by the command given by eq.(4) will induce to take into account all the preview control window which is made of 320 parameters. In order to keep the computation low, we discarded this option.

B. Drift between planned trajectories and real ones

The previously described pattern generator has been implemented and tested on the humanoid robot HRP-2 in simulation using the OpenHRP simulator [8] and in real experiments. It was used in conjunction with the commercial stabilizer shipped with the robot. In simulation the result shown that the robot walk without perturbation in the forward direction. However while turning, a momentum effect around the z-axis creates a perturbation which makes the feet slip in the external direction of the turn. The result is depicted in Fig 1. To compensate for this effect an arm motion heuristic was implemented. It creates a momentum along the z-axis which compensates for the one created by the lower limbs turning motion. It should be stressed that the hard-coded contact parameters for the simulation are conservative and insure that if the algorithm works on simulation it will work in reality. However they are over-pessimistic regarding the true behavior of the robot.

From the simulation, it is clear that the robot drift mostly while turning. In this case, small circles would be the most likely to trigger discrepancies. Another particular case is the sideway walking. Because the robot's feet are rectangular, in the sideway direction the ZMP has less room to evolve, and is more likely to reach unstable position. For this reason, two circular motions were designed as canonically representative of those problems but also likely to be used for applications. In the first one the robot walks tangently along a circle, while in the second the robot walks perpendicularly to the circle. The first experiment is intended for a searching behavior, whereas the second is aimed to object reconstruction.

In the first case, we preprogrammed the motion for a 0.75 m radius quadrant. The experiment was run 10 times and the results are a 4 mm average deviation in X (1.6 for the standard deviation), and 23 mm average deviation in Y (3.4 for the standard deviation). There was almost no drift regarding the orientation. The single support phase was 0.78 s while the double support phase was 0.02 s. The maximal height for the foot trajectory was set to 0.07 m. The step length was set to 0.15 m. Those are standard parameter values for the HRP-2 robot's walking.

In the second case, we also preprogrammed the motion to perform a 0.75 m radius circle. At first, the same walking parameters were used, and we experienced strong sliding on the landing foot. The resulting drift was 240 mm in X and 600 mm in Y with a 20 degrees of drift for the orientation. In a second time, the single support time was set to 0.75, and the double support time to 0.05. The maximal height of the foot trajectory was lowered to 0.04 m. The resulting drift was therefore 160 mm for X, 40 mm for Y and 15 degrees for the orientation. The step length was set to 0.1 m in order to decrease the dynamical effects.

III. SLAM Using Vision

Since the early work of Davison and Murray [9] which produced the first real-time SLAM system with vision as the primary sensor, there have recently been a growing number of successful demonstrations of the use of vision in SLAM in various very different scenarios — and especially in robot systems not accessible to other sensing modalities.

Systems of particular note include that of Jung and Lacroix [10], which applied a stereo vision SLAM approach to achieve wide range localization and mapping for ground-based and airship robots. Kim and Sukkarieh [11] used monocular vision in combination with inertial sensing to map ground-based targets from a dynamically manoeuvring UAV. Bosse *et al.* [12] used omni-directional vision in combination with other sensors in their ATLAS mapping framework, making particular use of lines in a man-made environment as consistent bearing references. Most recently Eustice *et al.* [13] have used a single downward-looking camera and inertial sensing to localize an underwater remote vehicle and produce detailed seabed reconstructions.

Although not a SLAM system capable of long-term mapping a loop closure, the real-time visual odometry system of Nistér *et al.* [14] deserves mention as it demonstrates the state of the art in the use of frame-to-frame visual matching for highly accurate motion estimation.

The second author Davison [6] demonstrated that real-time 3D SLAM in an indoor environment can be performed with a single camera and no additional sensing at all if sensible assumptions are made about motion dynamics and an active approach to feature measurement is implemented. In the following section we summarizes the key points of this approach.

A. Single Camera SLAM

In Davison [6], the second author demonstrated SLAM at 30Hz for a hand-waved camera, building on-line a sparse map of features to serve as localization landmarks. In this approach, a full-covariance Extended Kalman Filter (EKF) approach is used, storing the estimated state and covariance of the system at any instant as follows:

$$\hat{\mathrm{x}} = \begin{pmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \end{pmatrix} , \ \mathrm{P} = \begin{bmatrix} \mathrm{P}_{xx} & \mathrm{P}_{xy_1} & \mathrm{P}_{xy_2} & \dots \\ \mathrm{P}_{y_1x} & \mathrm{P}_{y_1y_1} & \mathrm{P}_{y_1y_2} & \dots \\ \mathrm{P}_{y_2x} & \mathrm{P}_{y_2y_1} & \mathrm{P}_{y_2y_2} & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix} . \quad (5)$$

Explicitly, the camera's state vector $\mathbf{x}_v$ comprises a metric 3D position vector $\mathbf{r}^W$, orientation quaternion $\mathbf{q}^{WR}$, velocity vector $\mathbf{v}^W$ and angular velocity vector ω^R (a total of 13 parameters). Feature states $\mathbf{y}_i$ are 3D position vectors.

For the widest possible range of application, Davison assumes that odometry was not available, and in the EKF prediction step a model for smooth motion anticipates Gaussian-distributed perturbations $\mathbf{V}^W$ and $\mathbf{\Omega}^R$ to the camera's linear

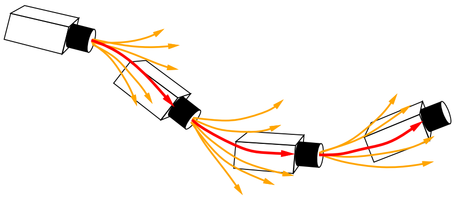


Fig. 2. Visualization of the “constant velocity” model for smooth motion from [6].

and angular velocity at each time-step — modelling motion with a generally smooth character. The explicit process model for motion in a time-step Δt is:

$$\mathbf{f}_v = \begin{pmatrix} \mathbf{r}^W_{new} \\ \mathbf{q}^{WR}_{new} \\ \mathbf{v}^W_{new} \\ \omega^R_{new} \end{pmatrix} = \begin{pmatrix} \mathbf{r}^W + (\mathbf{v}^W + \mathbf{V}^W)\Delta t \\ \mathbf{q}^{WR} \times \mathbf{q}((\omega^R + \mathbf{\Omega}^R)\Delta t) \\ \mathbf{v}^W + \mathbf{V}^W \\ \omega^R + \mathbf{\Omega}^R \end{pmatrix} \quad (6)$$

Fig. 2 illustrates how this models potential deviations from a constant velocity trajectory. Implementation requires calculation of the Jacobians of this process function with respect to both $\mathbf{x}_v$ and the perturbation vector (not presented here).

The features used in the map are natural points of high image interest detected using the operator of [15] and saved as square image template patches. Fig. 3 shows the type of image regions typically detected, corresponding mainly to corners or well-localized small objects. When a feature is first initialized, measurement from a single camera position provides good information on its direction relative to the camera, but its depth is initially unknown beyond potentially very weak prior information on the typical depths of objects in the scene. A semi-infinite 3D line is therefore initialized into the SLAM map, with end-point at the camera optical centre and direction derived from the image measurement: the 3D location of the feature lies somewhere along this line. The parameters describing the line have Gaussian-distributed uncertainties and corresponding entries in the SLAM covariance matrix, but to represent the non-Gaussian uncertainty in depth a discrete particle probability distribution is initialized along this coordinate with an initial flat profile representing complete uncertainty. As the camera moves and subsequent images are acquired, each particle hypothesis for depth is repeatedly tested and their probabilities evolve. Fig. 3(b) illustrates image search in a set of overlapping ellipses corresponding to the particles, and (c) the progression of the depth PDF from flat to a final peak at which point it can be replaced with a Gaussian and the feature fully initialized as a 3D point in the SLAM map. This process can take from 2–10 frames depending on the camera motion and uncertainty.

Fig. 3(d) illustrates active search for fully-initialized features during normal operation. The uncertainty in the relative position of the camera and features is projected into the current image and used to deduce elliptical search regions corresponding to 3 standard deviation confidence intervals within which the features are known to lie with high probability. Expensive normalized correlation search for matches can be restricted to these regions and this gives the algorithm the efficiency necessary for real-time implementation.

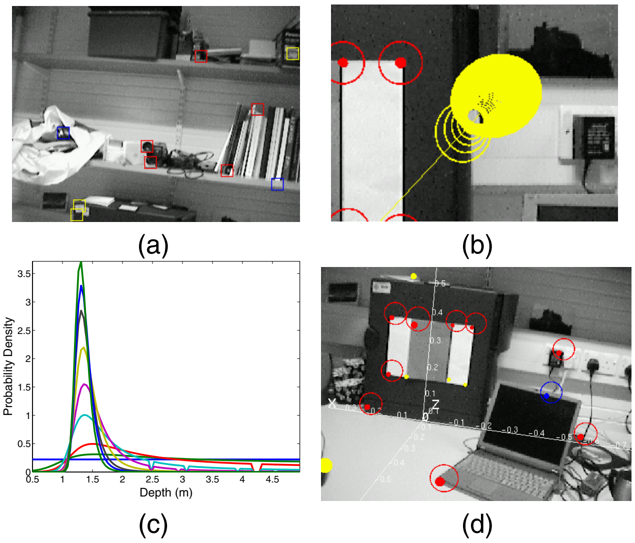


Fig. 3. Feature detection, initialization and matching from [6]: (a) 11 × 11 pixel patches detected a features; (b) searching a set of hypotheses for feature depth which project as image search ellipses; (c) probability distribution over depth refined to a peak over several time-steps; (d) elliptical search regions for mapped features during normal operation.

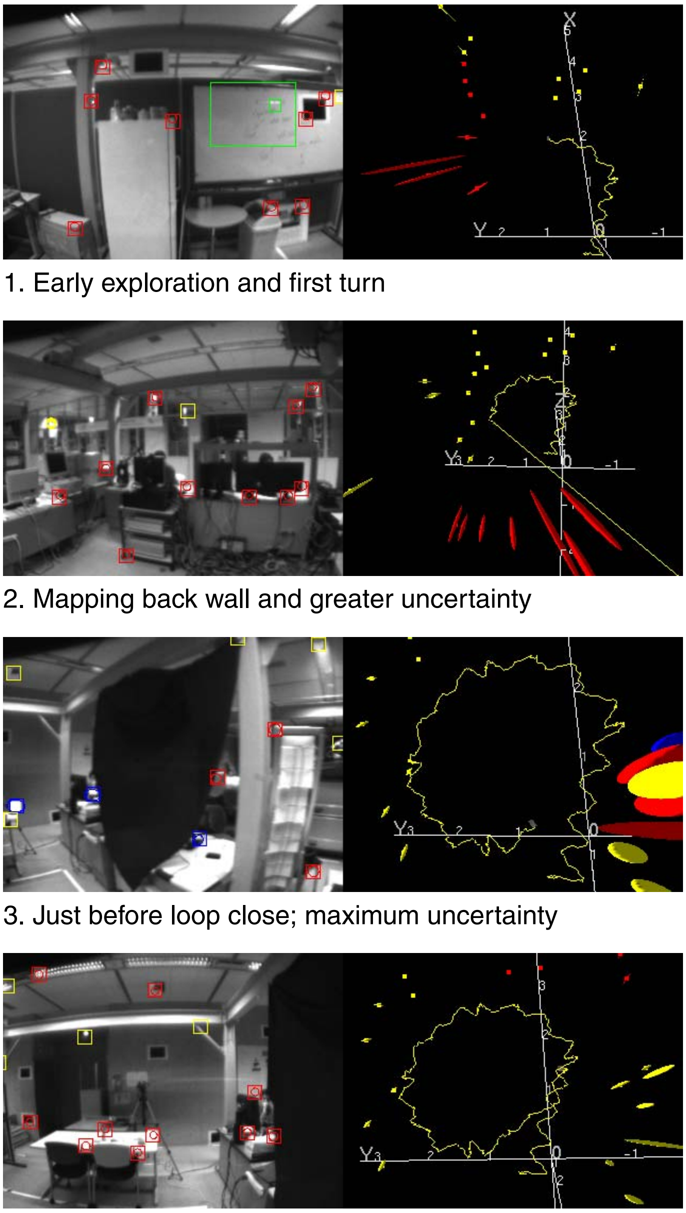
The 3D positions and image descriptions of a small number of features are required to bootstrap the SLAM system, principally to provide information on the overall metric scale of the map and camera motion. All other features are detected automatically and the initialization target can soon move out of the field of view or even be removed. Heuristic map-management criteria are used to decide when to initialize new features: essentially, the requirement is to keep a pre-defined number of features visible from all camera locations. A typical number used is 10; whenever fewer than 10 features are visible new ones are detected and initialized. Importantly, features are not deleted from the map when they leave the field of view, but remain in the map and can be re-observed when the camera moves back and they become visible again. In some cases it is necessary to delete features which are not being reliably matched on a regular basis: some features detected will be frequently occluded or may contain parts of objects at very different depths. These features will lead to failed correlation attempts and can be removed from the map automatically.

IV. HRP-2: Vision and Proprioceptive Sensing

We will now discuss the specific issues in implementing a single camera SLAM approach on the HRP-2 humanoid platform to estimate the motion of a head-mounted camera while mapping scene features.

A. *Internal measurements integration*

Incorporating a robot’s measurements of itself (whether from odometry or inertial sensors) into the SLAM EKF is achieved via a generalized ‘internal measurement’ capability



1. Early exploration and first turn

2. Mapping back wall and greater uncertainty

3. Just before loop close; maximum uncertainty

4. End of circle with closed loop and drift corrected

Fig. 4. Experiment 1: (Vision and gyro) frames from a circular trajectory of radius around 0.75m. The uncertainty in the map can be seen growing until the loop is closed and drift corrected. See the text for more explanations.

in our SLAM codebase. An internal measurement is one which relates only to properties of the camera/robot system itself, and not external features (as it is the case with a visual measurement in SLAM). An internal measurement $\mathbf{h}_v$ (generally a vector of multiple parameters) is modelled by defining function $\mathbf{h}_v(\mathbf{x}_v)$ of the state vector $\mathbf{x}_v$ of the camera. For instance, an odometry measurement of the height of the camera above the ground would give a function where $\mathbf{h}_v$ depends only on the position part $\mathbf{r}^W$ part of the state vector, whereas a measurement of velocity is a function of the camera velocity $\mathbf{v}^W$. The Jacobians $\frac{\partial \mathbf{h}_v}{\partial \mathbf{x}_v}$ of the internal measurements must also be calculated, and measurement uncertainty matrices $\mathtt{R}_v$ assigned, then a standard Kalman update can be used to update the state vector when a measurement is achieved.

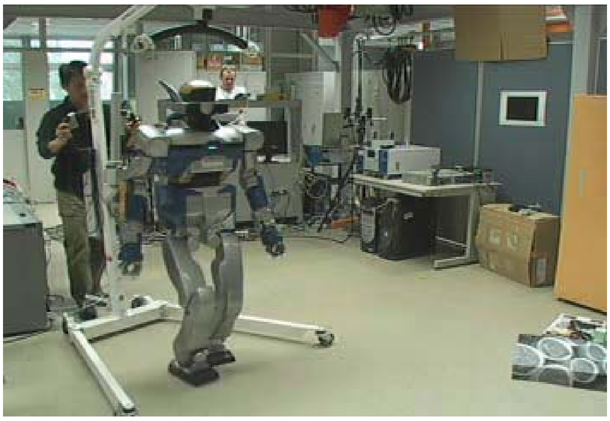


Fig. 5. HRP-2 walking in a circle in Experiment 1. The robot is walking autonomously and the support cradle is only for safety purposes.

During live operation, encoders report the live positions of these axes at 200Hz along with inertial sensors and their actual values are integrated on-board to recover the position of any specific part of the robot's body through the motion.

The outputs of the robot's motion controller are position and velocity sequences integrated over time. Some of these integrated positions are suitable for use in SLAM as they are drift-free quantities with a constant level of uncertainty, such as the height of the head above the ground plane which is constrained by a complete link through the robot's body. The integrated odometry estimate of the robot's forward position after several walking steps, however, would not be useful since this is a quantity in which the uncertainty depends on the detailed previous motion of the robot. Instead we are able to use velocity reports from odometry, since these represent instantaneous estimates.

B. *Pattern generator*

For those experiments the detailed movements of HRP-2 are generated by the pattern generator which receives a stack of foot position. Due to the preview control, here set to 1.6 s, two foot position must be given in advance. The pattern generator then compute every 5 ms the instantaneous desired position, speed and acceleration of the CoM. From this it generates the angle value for all of its 30 degrees of freedom. Based on eq. 2 we extracted from the pattern generator the information on the position, orientation, as well as linear and angular velocities. They are used as a measurement of those all four informations inside the EKF. The standard deviation of the position was set to 2 cm, while the standard deviation for the orientation was set to 0.005 radians. For the same reason than above, and because the pattern generator is only relative to a current pattern of steps, the position and velocity are only local to the motion.

Specifically in the experiments of this paper we chose to use the height of the camera above the ground (z_c in eq. 1, though with other walking patterns or different movements this will not be the case), and the speed of the camera. The standard deviation of the height measurement was assessed to be 3cm, and that of the velocity to be 5cm/s.

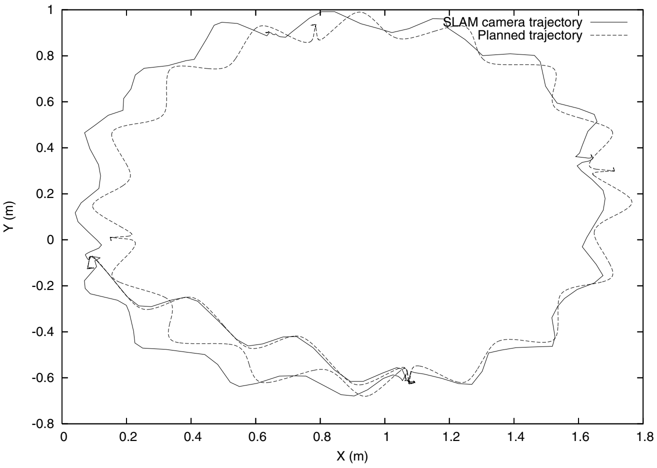


Fig. 6. Difference of trajectories between the planned trajectory and the one reconstructed by SLAM, in the case of walking along a circle. Starting from the origin, the robot first move forward to reach the position $(1, 0.5)$. It then walks around a circle.

C. Gyro

HRP-2 is equipped with a 3-axis gyro in the chest which reports measurements of the body's angular velocity at 200Hz. As with the body odometry, we sampled this at the 30Hz for use within the SLAM filter. We assessed the standard deviation of each element of the angular velocity measurement to be 0.01 rad/s. Since our single camera SLAM state vector contains the robot's angular velocity expressed in the frame of reference of the robot, we can incorporate these measurements in the EKF directly using our internal measurement capability.

HRP-2 is also fitted with a 3-axis accelerometer in the chest but we have not yet successfully incorporated this sensor into the SLAM system because the output is very sensitive to having an accurate orientation estimate. Note that a different method from the internal measurement capability must be used with accelerometer measurements, since acceleration terms do not appear in our state vector — the method of choice here is to fold the acceleration readings into the camera motion model.

D. Vision

As standard, HRP-2 is fitted with a high-performance forward-looking trinocular camera rig, providing the capability to make accurate 3D measurements in a focused observation area close in front of the robot, suitable for grasping or interaction tasks. In previous work on SLAM using vision, it has been shown that by contrast a wide field of view is advantageous for localization and mapping. For this and other related work, it was therefore decided to equip HRP-2 with an additional wide-angle camera (field of view around 90°) and use output from only this camera for SLAM. The wide angle camera was calibrated with a one parameter radial distortion model as in [16].

Also we consider the use of the trinoptic system in our localization scenario, several disadvantages came against relying only on it. Because the three cameras have narrow field of view the effective 3D perception space is also narrow. It involves a saccade for almost each features, and the result has a quite strong uncertainty. The interested reader can refer to previous work of the authors for more detailed explanation on how uncertainty has been evaluated [17]. In the case of walking, the advantages of stereo is compensate by the walking motion which induces a parallax effect. The larger field of view of our camera makes also the set of measurements more reliable. Finally as our targeted application is to use our SLAM technique in conjunction with other behaviors, such as object modelling, it might not be possible to perform such saccade. Seara [5] proposed a decision making strategy to choose between several gaze behaviors. Although it is interesting for higher autonomous behavior, this is not always pertinent with the current level of our teleoperation system. The whole vision rig of the robot is on an IEEE1394 bus connecting digitally to the vision processor, and image capture is at 30Hz.

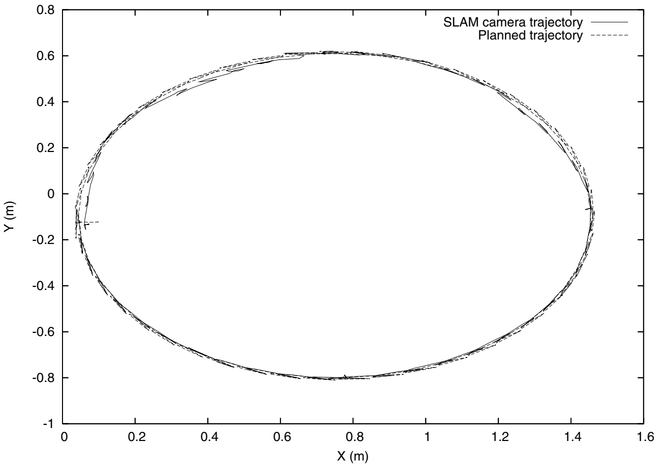


Fig. 7. Difference of trajectories between the planned trajectory and the one reconstructed by SLAM, in the case of walking orthogonal to a circle.

E. Implementational issues

HRP-2 contains two main internal Pentium M 1.8 GHz processors running the Linux operating system, one dedicated to motion control and one available for additional processing such as vision. Our vision processing was implemented to run at frame-rate on this second processor, triggered by capture from the camera in the robot's head. The two processors were synchronized via the NTP protocol over the robot's internal network so that time-stamped odometry and inertial data could be passed to the vision processor.

The vision processor communicated via CORBA and a wireless Ethernet link to a workstation acting as a graphical display and user interface terminal. Here a human was able to monitor the vision system's process in real-time. **When**

only based on vision the single camera SLAM currently requires a small set of features with pre-measured positions to be specified to bootstrap tracking. In our experiments, a set of well-chosen salient features (a mixture of natural features and some hand-placed targets) mostly on the wall in front of the robot's starting position were chosen by hand. It is relatively important that these features are well-spread in the image at the start of motion since they effectively seed the rest of the map, which will be extrapolated from the initial robot motion estimates they provide. However it is shown that those initial features **are not used** when integrating the information provided by the pattern generator.

A technical change was to make the image measurement uncertainty for features scale linearly with distance from the image centre. This reflects our belief that the one parameter radial distortion model fits less well in the image periphery.

V. Experiments

We performed two types of motion for evaluation of the SLAM tracking performance. All the loop-closing experiment, the image processing, pattern generation and sensor processing are performed entirely on-line and autonomously inside the robot.

A. Experiment 1: Walking along a circle

1) Vision and gyro only: In order to solve the scale problem, and only on the case of this specific experiment the robot is set in a given position facing known landmarks. This experiment performed was a real SLAM test, in which the robot was programmed to walk in a circle of radius 0.75m. For safety and monitoring reasons, the motion was broken into five parts with short stationary pauses between them: first a forward diagonal motion to the right without rotation, in which the robot put itself in position to start the circle, and then four 90° arcing turns to the left where the robot followed a circular path, always walking tangentially. The walking was at HRP-2's standard speed, and total walking time was around 20–30 seconds.

Fig. 4 shows the results of this experiment. On the left, the elected landmarks are depicted, the ones surrounded by a red square are tracked and selected landmarks for the EKF update. The yellow landmarks are detected but not selected. On the right, the trajectory of the camera is depicted together with the landmarks position. Their uncertainty is represented by 3D ellipsoid. Classic SLAM behavior is observed, with a steady growth in the uncertainty of newly-mapped features until an early feature can be re-observed, the loop closed and drift corrected. A large number of features are seen to swing into better estimated positions simultaneously thanks to the correlations stored in the covariance matrix.

Also the loop-closing is definitely realized, the quality of the trajectory itself is not sufficient for the targeted application. In the next section, the same experiment integrating the pattern generator is described.

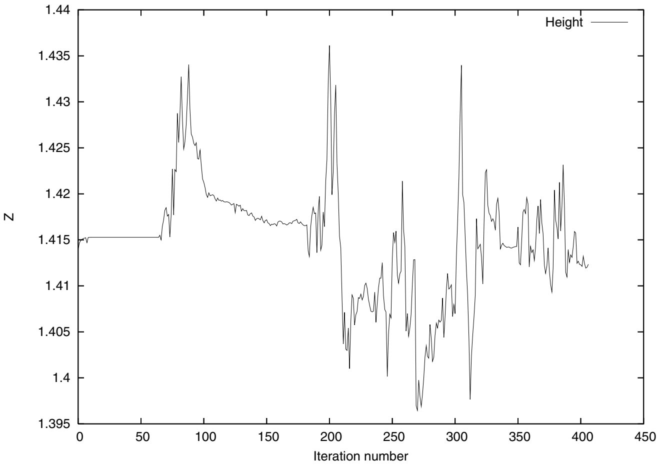


Fig. 8. Height variation for experiment 1. Interval between 50 and 180 corresponds to the forward motion.

2) Vision, pattern generator and sensors integration: Having good odometry and inertial readings means that new features can be initialized quickly. As the information of the pattern generator is quite well realized by the robot in the following set of experiments realized, the initialization phase usually necessary for the vision-only SLAM has been completely ignored. The robot chose by itself the initial landmarks inside the environment. This experiment has been performed 10 times, and in each case loop closure with an average of 10 landmarks have been realized. The reconstructed trajectory is depicted in Fig. 6. This motion is most difficult for vision as the system is almost continuously turning and therefore the views change drastically. Therefore the discrepancy between the planned trajectory and the reconstructed one is becoming larger. However, we clearly see that during the loop-closing the two are finally converging. In comparison during the forward motion the reconstructed trajectory and the planned one match well almost all the time. The final point and the initial point of the circle have the same order of error compared to the real measurements. Moreover, we experimented several ill-chosen visual features discarded because of invalid match between the prediction and the measurement. If this motion shows how the pattern generator might help vision, it is not pertinent to show how the vision might compensate for the pattern generator drift. The second experiment in this regard is more interesting.

B. Experiment 2: Walking perpendicularly to a circle

As shown previously, this motion is realized differently than what was planned by the pattern generator. There is a strong drift in position and orientation between the desired final position, and the one achieved. This motion is however very helpful for vision, as it is mainly translational with few rotation. In this case, we also experimented loop-closing with an initialization not relying on the accurate positioning of initial landmarks. Only 3 landmarks during our experiments were able to close the loop. The reconstructed trajectory is depicted in Fig. 7. In this experiment the robot is starting at the origin and draws a circle facing the center at position

$(0.75, 0)$, and going counterclockwise. Clearly the trajectory is modified by the vision measurement during the second half of the circle, However the final drift of the pattern generator is not well represented in the final reconstructed trajectory along the Y-axis (140 mm of difference) and the orientation, although it is almost correct along the X-axis (30 mm of difference).

VI. Discussion

During the experiments, we experienced often ill-chosen landmarks such as highly-repetitive light-ceiling corners. During the first trial the experiments were conducted using an open space of size $11m \times 25m \times 3m$, to which must be added an open office of size $7m \times 25m \times 3m$. In this case most of the landmarks were chosen at 10 to 12 meters with big uncertainties, and did not allow the algorithm to give meaningful results given the size of the motion. Therefore the space was limited to the one similar to an office $7m \times 11m \times 3m$ as depicted if Fig.(5). Another parameter to take into account is the CPU bandwidth limitation. In the space indicated the covariance matrix computation cost stays in the real-time boundaries : 30 ms with some variations around 40 ms. In this context, the use of all the information allowed to discard the ill-chosen features, and get rid of the landmark initialization. Indeed the pattern generator allow to check if the scale at which the features are perceived is correct. Using only vision SLAM induced too much uncertainties and we could not close loop with the motion of experiment 1.

The evaluation of the height was also checked, and its range of variation is in a 4 cm interval. Also some discontinuities clearly appear, we did not find any immediate link with the foot impact, as shown in Fig. 8. However for the first experiment, the forward motion appears to involve less perturbation than for circling motion.

VII. Conclusions

We have shown that monocular visual 3D SLAM can be achieved almost from the start in real-time and autonomously for small scale environment. The key-point is to integrate the information provided by the pattern generator inside the EKF framework to help the vision process. This integration allows also to get rid of the knowledge of initial landmarks to start the SLAM process. Real experiments with visual loop-closing were achieved. The range of motion used for the experiments was the one used and intended for applications. We will extend this work to longer and more complex behavior to test the capabilities of the system. In the near future we would like to test the performance of the system with more ambitious motions — HRP-2 is capable of lying down on its back from a standing position for instance.

Once reliable visual localization performance in a wide range of scenarios is confirmed, and robust real-time operation is fully achieved, the next step would of course be to close the loop on motion estimation and control — SLAM would enable the robot to move purposively around its environment, revisit places or interest or remember the locations of task-related items. We hope that this will provide a platform for continuing research in increased robot autonomy.

Acknowledgement

The authors would like to thank the JRL group, Pierre-Brice Wieber, Shuji Kajita, and Kenji Kaneko for helpful discussions while implementing the pattern generator.

References

[1] K. Yokoi, N. E. Sian, T. Sakaguchi, H. Arisumi, E. Yoshida, O. Stasse, Y. Kawai, K. Maruyama, T. Yoshimi, and S. Kajita, "Humanoid robot hrp-2 no.10 with human supervision," in *International Symposium on Robotics, ISR*. Tokyo, Japan: IEEE, November 29 - December 1 2005.

[2] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, pp. 111–127, 2003.

[3] S. Thompson and S. Kagami, "Humanoid robot localisation using stereo vision," in *Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots (Humanoids2005), Tsukuba*. IEEE, December 2005, pp. 19–25.

[4] R. Ozawa, Y. Takaoka, Y. Kida, K. Nishiwaki, J. Chestnutt, J. Kuffner, S. Kagami, H. Mizoguchi, and H. Inoue, "Using visual odometry to create 3d maps for online footstep planning," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC2005), Hawaii, USA*. IEEE, October 2005, pp. 2643–2648.

[5] J. F. Seara and G. Schmidt, "Intelligent gaze control for vision-guided humanoid walking: methodological aspects," *Robotics and Autonomous Systems*, vol. 48, no. 4, pp. 231–248, 2004.

[6] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings of the 9th International Conference on Computer Vision, Nice*, 2003, pp. 1403–1410.

[7] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *International Conference on Robotics And Automation, Taipei Taiwan*, September 2003, pp. 1620–1626.

[8] H. Hirohisa, F. Kanehiro, S. Kajita, K. Fujiwara, K. Yokoi, K. Kaneko, and K. Harada, "Experimental evaluation of the dynamic simulation of biped walking of humanoid robots," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan*, September, pp. 1640–1645.

[9] A. J. Davison and D. W. Murray, "Mobile robot localisation using active vision," in *Proceedings of the 5th European Conference on Computer Vision, Freiburg*, 1998, pp. 809–825.

[10] I. Jung and S. Lacroix, "High resolution terrain mapping using low altitude aerial stereo imagery," in *Proceedings of the 9th International Conference on Computer Vision, Nice*, 2003, pp. 946–951.

[11] J. H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 406–411.

[12] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003.

[13] R. M. Eustice, H. Singh, J. J. Leonard, M. Walter, and R. Ballard, "Visually navigating the RMS titanic with SLAM information filters," in *Proceedings of Robotics: Science and Systems*, 2005.

[14] D. Nistér, "Preemptive RANSAC for live structure and motion estimation," in *Proceedings of the 9th International Conference on Computer Vision, Nice*, 2003, pp. 199–206.

[15] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[16] A. J. Davison, Y. G. Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision," in *Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*, July 2004.

[17] B. Telle, O. Stasse, T. Ueshiba, K. Yokoi, and F. Tomita, "Three characterisations of 3d reconstruction uncertainty with bounded error," in *International Conference on Robotics and Automation, ICRA*. Barcelona, Spain: IEEE, April 18-22 2005, pp. 3905–3910.