

Appearance-based localization for mobile robots using digital zoom and visual compass

N. Bellotto^{a,*}, K. Burn^b, E. Fletcher^b, S. Wermter^b

^a *Human Centred Robotics Group, Department of Computer Science,
University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK*

^b *Centre for Hybrid Intelligent Systems, School of Computing and Technology,
University of Sunderland, St Peter's Way, Sunderland SR6 0DD, UK*

Abstract

This paper describes a localization system for mobile robots moving in dynamic indoor environments, which uses probabilistic integration of visual appearance and odometry information. The approach is based on a novel image matching algorithm for appearance-based place recognition that integrates digital zooming, to extend the area of application, and a visual compass. Ambiguous information used for recognizing places is resolved with multiple hypothesis tracking and a selection procedure inspired by Markov localization. This enables the system to deal with perceptual aliasing or absence of reliable sensor data. It has been implemented on a robot operating in an office scenario and the robustness of the approach demonstrated experimentally.

Keywords: appearance-based localization; digital zoom; visual compass; Markov localization.

1. Introduction

In mobile robotics, localization plays a fundamental role for the navigation task, since it is necessary for every kind of path-planning. In order to achieve a goal, an autonomous mobile robot must be able to localize itself within the environment where it is acting and relatively to the target destination.

The main objective of this article is to illustrate the implementation of a new map-based localization system for a mobile robot operating in an indoor environment where it is not necessary to know the exact, absolute position. Instead, a topological localization is the most appropriate solution. We developed a new visual place recognition algorithm that does not need any specific landmark. In particular, the novelty introduced by such algorithm is the use of digital zooming to improve the capability of recognizing places. The same algorithm is also used for reconstructing panoramic images from the place of interest, combining a sequence of snapshots taken with the camera. Such images, together with approximate coordinates of the topological locations, form the map used by the robot. Furthermore, when the robot is located in one of the mapped places, it can also estimate its absolute orientation using vision, thanks to an original visual compass system. The place recognition process is then followed by a procedure that resolves cases of perceptual aliasing or absence of reliable sensor information. The system keeps track of a set of hypotheses and for each update step chooses the most plausible with an approach inspired by Markov Localization. From experiments carried out in a typical office scenario, the method shows to be robust even in case of dynamic environments and locations poor of features.

* Corresponding author.

E-mail address: nbello@essex.ac.uk (N. Bellotto).

The remainder of the article is structured as follows: in Section 2 we report a brief literature review; Section 3 and Section 4 describe respectively the place recognition and the multiple hypothesis localization; in Section 5 we present some experimental results and finally we conclude in Section 6 with a summary and some recommendations.

2. Related work

In recent years there have been increasing numbers of robot applications where localization is an essential part of the navigation system. Well known examples include the tour-guide robots Rhino and Minerva [1, 2], or the robot-waiter Alfred [3], which used different approaches and sensors for localization. With Rhino, for example, perceptions were based upon sonar and laser range sensors, whilst Minerva made use of lasers plus an additional camera directed towards the ceiling, so the observed scene was mostly static. In contrast, Alfred used vision with artificial landmarks to recognize places of interest.

Other localization approaches making use of vision have been presented in recent years. Gini and Marchi [4] used a robot equipped with a unidirectional camera pointing ahead and towards the floor. Their basic hypothesis was that the floor had a uniform texture so that after camera calibration, it was possible to reconstruct a local map from images. Localization was then the result of a comparison between the current local map and a pre-recorded global map. The solution of Dao *et al.* [5] was based on a natural landmark model and a robust tracking algorithm. The landmark model contained sets of three or more natural lines such as baselines, door edges and linear edges of tables or chairs. The localization depended on an algorithm that allowed the robot to determine its absolute position from a single landmark.

Several recent approaches have made use of Monte Carlo localization [6, 8]. It has been demonstrated that this technique is reliable and, at the same time, keeps the processing time low. Indeed, Monte Carlo localization has been successfully applied in the RoboCup four-legged league, where the Sony dog's hardware has critical limitations. For example, Enderle *et al.* [6] implemented a Monte Carlo approach for vision-based localization that made use of sporadic features, extracted from the images of the robot's unidirectional camera. The probability of being in a certain location was calculated against an internal model of the environment where the robot moved. Experiments proved that the method was reliable enough, even with a restricted number of image samples, and was improved drastically by increasing the number of features. Tests in a typical office environment were also promising.

Menegatti *et al.* described another application of Monte Carlo localization in the RoboCup context [7]. In this case, the video input came from an omni-directional sensor; the images were processed in a way to simulate a laser scanner, using the distances from points with color intensity transitions. Even here the localization system made use of an internal representation of the football field. Ulrich and Nourbakhsh also used an omni-directional camera for topological localization [8]. They presented an appearance-based place recognition scheme that used only panoramic vision, without any odometry information. Color images were classified in real-time with nearest-neighbor learning, image histogram matching and a simple voting scheme. Andreasson and Duckett illustrated another system in [9] that performed topological localization using images from an omni-directional camera. Their method searched for the best matching place among a database of pre-recorded panoramic images, extracting and using modified SIFT features [10]. An interesting approach was also the context-based visual recognition implemented by Torralba *et al.* [11], which made use of low resolution images from a wearable camera to extract texture features and their spatial layout. Training was done with hand-labeled image sequences taken in the environment to

Pre-print version accepted for publication on *Robotics and Autonomous Systems*, Vol. 56, Issue 2, pp.143-156, February 2008
map, then localization was performed using two parallel Hidden Markov Models (HMMs) for both place recognition and categorization, the latter useful also for the identification of unexplored environments.

Numerous techniques have finally been devised to resolve the ambiguity that arises in sensory perception, irrespective of the device employed. No observation indeed is immune from noise and errors, originating in both the sensor and the surrounding environment. A wide range of localization systems have been tested and compared in the works of [12-14], covering methods based on Extended Kalman Filtering (EKF), Markov Localization (ML), a combination of the two (ML-EKF), Monte Carlo Localization (MCL) and Multi Hypotheses Localization (MHL). The results of these experiments have been used as a basis for motivating the most suitable localization approach for our application.

3. Place recognition

In this section, we describe a new method to recognize a position amongst a finite set of possible locations. This set is basically a topological map of the environment provided by the user and each place is identified by a point in the Cartesian space and a panoramic image of the scene observed from that point. The procedure is based on the comparison of a new image, taken by the robot’s camera, with all the panoramic images of the map. A measure of the match’s quality is assigned to each comparison using a novel image-matching algorithm (or *IMA*). Basically, this process constitutes the place recognition, which is an essential part of our localization.

3.1 Image matching algorithm

Typically, for indoor environments, most of the relevant changes occurring in an image are due to objects or people moving with respect to a horizontal plane. A person walking, a chair moving, a door opening or closing: all of these examples can be thought as “columns” moving horizontally along an image of the original scene. The algorithm described in this section arises from this simple consideration. The principal idea is to divide the new image into several column regions, called “slots”, and then compare each of them with a stored image of the original scene.

Consider the new image I_{new} , single channel, of width W_{new} . This is divided into N_s slots having width $W_{slot} = W_{new} / N_s$. One slot is referred to as $slot_n$, with $n = 1, \dots, N_s$. Also, consider a reference image I_{ref} , single channel, of width $W_{ref} \geq W_{new}$. The images I_{new} and I_{ref} have the same height. A region of I_{ref} , delimited by the pixel columns c_{left} and c_{right} , is referred to as $I_{ref}[c_{left}, c_{right}]$; the columns c_{left} and c_{right} belong to this region. The two image structures are illustrated in Fig. 1.

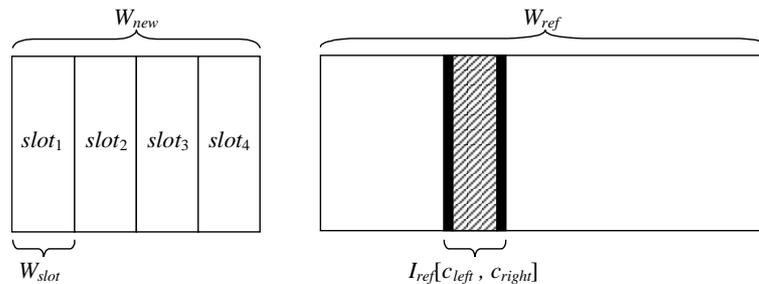


Fig. 1 Example of I_{new} (divided into four slots) and I_{ref}

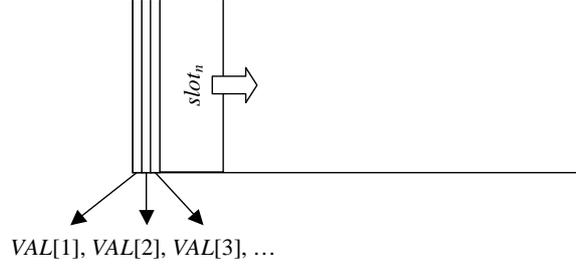


Fig. 2 Slot of I_{new} shifted and compared along I_{ref} by NCC

The measure of the similarity between a slot of the new image and a region of the stored one is given by a function based on the Normalized Correlation Coefficient [15] and called NCC . Given a new $slot_n$ and a reference image I_{ref} , the NCC matching function compares $slot_n$ with all the regions $I_{ref}[c, c + W_{slot} - 1]$, where $c = 1, \dots, W_{ref}$ (if $slot_n$ falls over the right bound of I_{ref} , it restarts from the beginning) After each comparison, a value between 0 and 1 is stored inside an array VAL of length W_{ref} , as explained also in Fig. 2 (note that the original Normalized Correlation Coefficient varies between -1 and 1 , so we actually rescale it to fit between 0 and 1). For example, if the slot's width is $W_{slot} = 10$, the assignment $VAL[5] = 0.7$ means that the similarity between $slot_n$ and the region $I_{ref}[5, 15]$ measures 0.7.

The actual IMA can be divided into two parts: the first apply NCC to find, for each pixel column, which is the slot that matches best; the second determines the position that gives the maximum match for the whole sequence of N_s slots. The algorithm is described by the pseudo-code in Table 1 (note that the highlighted “**else if**” condition is for the reconstruction of panoramic images explained in Section 3.2).

Table 1 Image Matching Algorithm (IMA)

```

/* first part: slot matching */
VAL[W_ref] = {0, ..., 0}
MATCH_SLOT[W_ref] = {0, ..., 0}
MATCH_VAL[W_ref] = {0, ..., 0}
for n = 1 to N_s
    NCC(slot_n , I_ref , VAL)
    for c = 1 to W_ref
        if VAL[c] > MATCH_VAL[c]
            MATCH_SLOT[c] = n
            MATCH_VAL[c] = VAL[c]
        end if
    end for
end for
/* second part: best match extraction */
BEST_MATCH = 0
COL = 1
for c = 1 to W_ref
    SUM = 0
    for n = 1 to N_s
        i = c + (n - 1) W_slot
        if MATCH_SLOT[i] = n
            SUM = SUM + MATCH_VAL[i]
        else if MATCH_VAL[i] = 0.5
            SUM = SUM + SUM / (n - 1)
        end if
    end for
    if SUM > BEST_MATCH
        BEST_MATCH = SUM
    end if
end for

```

```

    COL = c
  end if
end for
BEST_MATCH = BEST_MATCH / Ns
END

```

3.2 Panoramic image

For every place in the environment, a panoramic image can be also reconstructed using the *IMA* algorithm. Initially, the panorama is just a black image and the relative similarities returned by *NCC* measure exactly 0.5. With a simple “**else if**” condition in the second part of the *IMA*, highlighted in Table 1, this situation can be handled and used for the correct insertion of a new image in the panoramic view. Basically, whenever a slot is compared with a black zone, the assigned matching-value is the mean of the previous comparisons. Of course this is valid only if a sequence of snapshots, taken during a clock-wise rotation, is inserted in the exact order, from left to right. The input images are also filtered using a Contrast Limited Adaptive Histogram Equalization (CLAHE) [16] in order to increase the number of distinguishable features for scenes not well illuminated. The insertion of new images continues until the whole panorama is filled. An example of reconstruction is shown in Fig. 3.

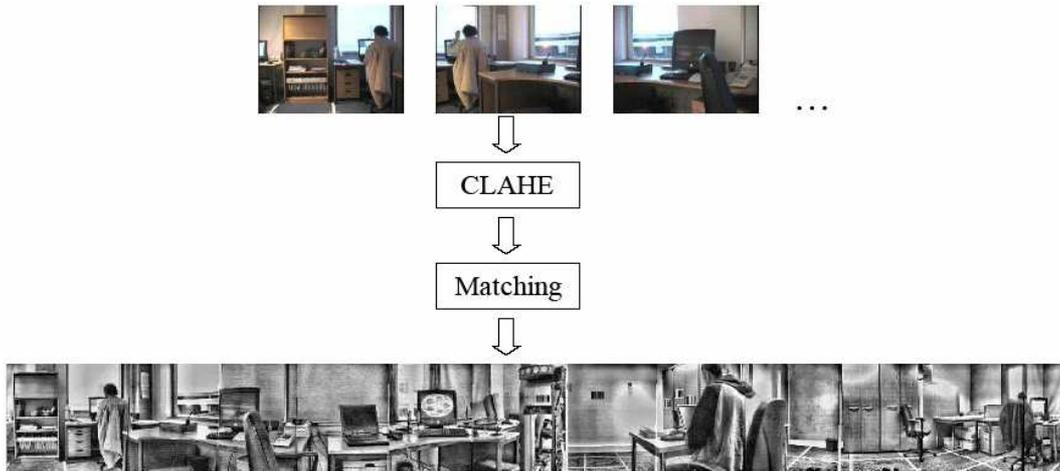


Fig. 3 Panoramic image reconstruction

3.3 Heading angle extraction

An important feature of the *IMA* is the capacity to extract the position, inside a panoramic image, where a new snapshot matches best. This position is given by the value *COL* (see Table 1), which is the left pixel column of the region on I_{ref} where the best match occurs. If $COL = 1$ corresponds to the zero direction on a panoramic image having width W_{ref} (and considering a clock-wise versus), then the displacement angle α of the camera is simply given by the following expression:

$$\alpha = 2\pi \cdot \frac{COL - 1}{W_{ref}} \quad (1)$$

Therefore, if all the panoramic images have been reconstructed with a common angle of reference, α can be used to estimate the robot’s heading. Its precision is normally good enough to be used as a “visual compass” and correct the odometry’s heading angle, as explained in Section 4.6 and demonstrated experimentally in Section 5.2.

3.4 Enhancement with digital zooming

The place recognition method described so far suffers from the problem of sensitivity to the distance from the original point where the panoramic image has been constructed. This means that, moving the robot away from that point, the output of *IMA* quickly decreases. To solve this problem, digital zoom was integrated to enlarge the detection area.

Digital zoom can be implemented using bilinear interpolation and explained starting from the well known pin-hole camera model [17]. This is shown in Fig. 4a for a given object of height H and distance $D = X - x_0$ from the camera, for which the next relations can be written:

$$\frac{h}{f} = \frac{H}{D} \quad \frac{\rho \cdot h}{f} = \frac{H}{D - s(\rho, D)} \quad (2)$$

where h is the height projected on the image plane, f is the focal length of the camera, ρ is the zoom factor and $s(\rho, D) = x' - x_0$ is the “virtual” shift from the original position. After simple passages, the latter can be expressed as follows:

$$s(\rho, D) = D \cdot \left(1 - \frac{1}{\rho}\right) \quad (3)$$

Given a panoramic image of a place at position $P_0(x_0, y_0)$ and moving the robot along a rectilinear path on an interval $[x_0 - \Delta x, x_0 + \Delta x]$, *IMA* returns values that can be approximately represented with a centrally peaked distribution, as empirically showed in Section 5.3. To expand the interval where the *IMA*'s output is higher, the input image from the camera can be digitally zoomed. More precisely, after a normal comparison, the image is zoomed-in and compared again, then zoomed-out and compared once more. Theoretically, including these new comparisons means adding a couple of new peaked curves to the original one. In order to have the same absolute shift $|s(\rho_{in}, D)| \equiv |s(\rho_{out}, D)|$ for both the zoom-in and the zoom-out, the following relationship can be easily derived (note that $\rho_{in} > 1$ and $0 < \rho_{out} < 1$):

$$\rho_{out} = \frac{\rho_{in}}{2\rho_{in} - 1} \quad (4)$$

The combination of the three *IMA*'s outputs is shown in Fig. 4b, where $x_{Zin} = x_0 + s(\rho_{in}, D)$ and $x_{Zout} = x_0 + s(\rho_{out}, D)$. The actual output considered for place recognition is the maximum of the three curves, as specified also by the pseudo-code in Table 2.

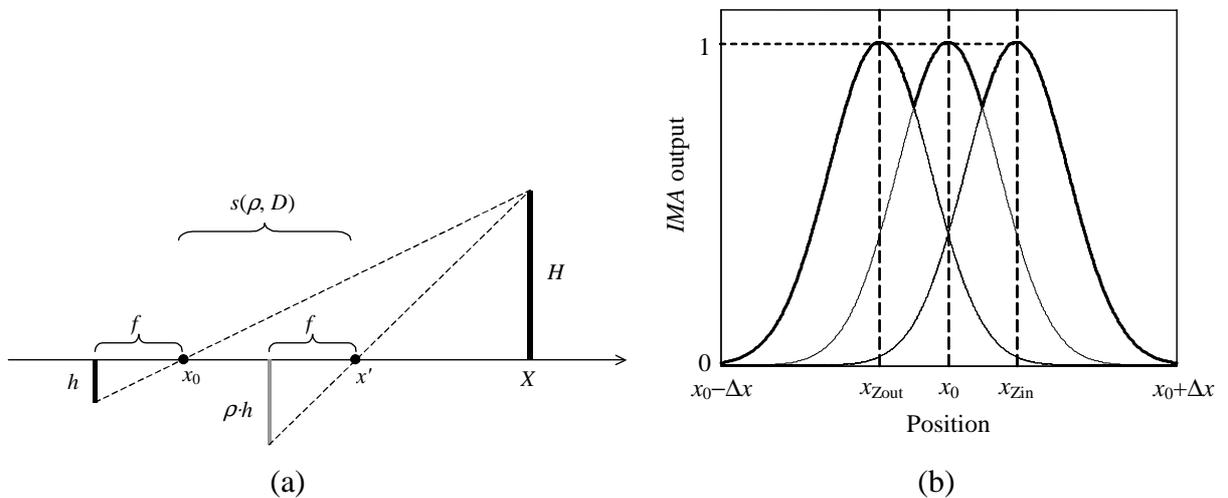


Fig. 4 Virtual shift model and *IMA*'s output using digital zoom

Table 2 Digital zoom enhancement for the *IMA*

```

/* image matching without zoom */
I ← Inew
M = IMA(I, Iref)
BEST_MATCH = M
/* image matching with zoom-in */
I ← zoom-in of Inew
M = IMA(I, Iref)
if M > BEST_MATCH
    BEST_MATCH = M
end if
/* image matching with zoom-out */
I ← zoom-out of Inew
M = IMA(I, Iref)
if M > BEST_MATCH
    BEST_MATCH = M
end if
END

```

Unfortunately, in a real environment things are more complicated – scenes (and objects within them) always have different distances from the point from which they are observed. The width of the curve in Fig. 4 could be altered significantly changing the direction of observation since the distance of a new scene can be different from a previous one, influencing therefore x_{Zin} and x_{Zout} . Because the virtual shift in (3), for a fixed zoom factor ρ , changes linearly with the distance D , the region where the recognition holds can be presumed to depend somehow on the shape of the room. For example, consider an observation point \mathbf{P} within a small empty room, as illustrated in Fig. 5a. The crosses indicate the displacements given by the zoom-in when observing in the direction of the arrows; the squares are the relative displacements for the zoom-out. In Fig. 5b the two sets of points for the zoom-in and the zoom-out, obtained by a full rotation about \mathbf{P} , are represented by the solid and the dashed squares respectively. If we fix a proper threshold on Fig. 4, over which the *IMA* output is considered valid (for example 0.5), we can draw a region for \mathbf{P} where the recognition holds, as illustrated in Fig. 5c. This region is given by the union of two rectangular areas, one for the zoom-in and one for the zoom-out, which are extensions of the previous in Fig. 5b. Indicating also the robot's position and orientation with a versor, the zoom-out rectangle contains all versors pointing to \mathbf{P} , while the zoom-in rectangle contains all versors pointing in the opposite direction.

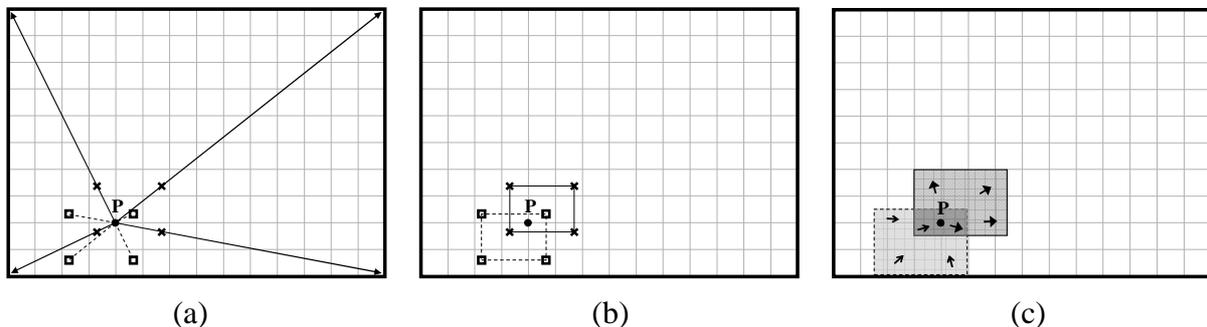


Fig. 5 Place recognition with digital zoom

The observations above suggest some care must be taken when choosing the places to recognize (topological nodes of the map) and the zoom factor to use. In particular, if the nodes are too close to each other or if the zoom is too much, the risk of overlaps amongst

them and the probability of perceptual aliasing increase. Moreover, since the zoom-based recognition works best when observing along those directions passing through the area’s point of reference (point \mathbf{P} in Fig. 5), a small zoom factor is preferable. In this way, the limited area extension increases the probability to be correctly aligned.

4. Multiple hypothesis localization

The main problem using image-based place recognition for localization arises when two or more places look very similar and are therefore difficult to distinguish. This is known as perceptual aliasing and affects not just vision-based applications, but many other systems employing sensors that provide information about the perceived world (e.g. sonar, laser, etc.). It occurs frequently in indoor environments with similar rooms and furniture like offices.

The *IMA* procedure, described in the previous Section 3.1, is normally able to distinguish different places because it considers a significant amount of information coming from the vision input. Nevertheless, cases of perceptual aliasing may occur because of occlusions or changes in the scenes originally memorized. To handle this kind of uncertainty, we adopt an algorithm inspired by Markov localization [18]. It starts with a series of hypotheses generated by the place-recognition procedure and then chooses the most likely according to the previous hypotheses and to the last robot’s movement.

4.1 Notations and assumptions

Let the state (i.e. position) of the robot at time t be represented by a triplet $\langle x_t, y_t, \varphi_t \rangle$, where x_t and y_t are the Cartesian coordinates of the robot’s location and φ_t is its heading angle. The couple (x_t, y_t) belongs to a finite set of two-dimensional points, which is the topological map. The heading angle φ_t has continuous values inside the interval $[0, 2\pi)$. Therefore, the entire set S of possible states contains an infinite number of elements.

To make the problem computationally feasible, certain assumptions are imposed. It is assumed that the probability distribution at time t of the robot being in a certain position $\langle x_t, y_t, \varphi_t \rangle$ is completely contained in a sub-set $S_t \subset S$. The elements of S_t are all the positions for which the *IMA*, at time t , returns a matching-value higher than a certain threshold, plus an additional “virtual” position given by the odometry. That is, the real position of the robot is always supposed to be one of those recognized by the place recognition or calculated using odometric information; this is justified by the fact that, most of the times, the correct position is in effect one of the best recognized with the *IMA*. The number of possible states so generated is limited by the nodes of the topological map; therefore, S_t is a numerable set.

In the following sections, the set S_t is referred to with the letter D and called the *destinations’* set (elements $d \in D$). The set S_{t-1} is referred to with the letter O and called the *origins’* set (elements $o \in O$). The set D of destinations at time t becomes then the set O of origins at time $t + 1$. Also, to distinguish the “local” probability distribution from the one used in Markov localization, the word *belief* is substituted with *activity*, as in [19]. The beliefs $Bel(s_t)$ and $Bel(s_{t-1})$ become then the activities $Act(d)$ and $Act(o)$ respectively (activity of the destination and activity of the origin).

4.2 Virtual destination hypothesis

The assumption of considering only the destinations given by the last observation, i.e. the *IMA* output, would be too restrictive. To be sure that the set of estimated positions contains in effect the right one, the threshold on the visual-recognition with the *IMA* should be very high. This would limit the possibility of considering good hypotheses just because some changes in the environment, temporary or permanent, have reduced their distinctiveness. On the other hand, a low threshold increases the number of possible destinations but also the probability to choose the wrong one. Even worse is the case when none of the current hypotheses are correct. To handle this kind of ambiguity, sometimes a “zero hypothesis” is used when all the other hypotheses are considered wrong. In [20], for example, the authors have a finite set of hypotheses generated by new observations, updated simultaneously using Kalman filters. The zero hypothesis is used to close the probability space and is kept up-to-date considering the uncertainty of the observations. When the probability of such a hypothesis is higher than the others, the robot is in a state of indecision.

In our approach, it was found useful to insert a “virtual destination”, that is, a topological node of the map that is near to the position given by the odometry. In practice, the virtual destination is the closest node, in terms of Euclidian distance, to the previous winning destination plus the last odometric displacement. The heading angle of this new hypothesis is also given by the odometry. The term “virtual” is used because it is assigned a matching-value, like all the other destinations generated by an observation, and then treated the same way. The assigned matching-value is equal to the threshold chosen for generating the other hypotheses, as if an additional place was recognized by the *IMA* with the least acceptable match. Finally, at the next update step, the “virtual-destination” becomes the “virtual-origin”.

4.3 Action model

The first component of Markov localization is the action model. Using the notation introduced before and simply calling a the action a_{t-1} , the model can be written as follows:

$$P(s_t | s_{t-1}, a_{t-1}) \equiv P(d | o, a) \quad (5)$$

This expresses the probability that a destination d is reached by performing the action a from the origin o . This probability is estimated taking into account the location and the heading angle of the robot. The action a is simply the displacement given by odometry. In this work, no sophisticated models were used for handling the cumulative errors typical of odometry; indeed, its information is always relative to the previous estimated state and corresponds to a short path. Therefore, it is considered reliable enough for being directly used in our action model, as described below.

Let Q_o be the position of the origin o (with heading angle φ_o) and Q_a the position reached from Q_o after the execution of a . Also, Q_d is the position of the destination hypothesis d . Using the quantities illustrated in Fig. 6, the action model is calculated as follows:

$$P(d | o, a) = g_l(\Delta l) \cdot g_\varphi(\Delta \varphi) \quad (6)$$

where $g_l(\Delta l)$ and $g_\varphi(\Delta \varphi)$ are two Gaussians:

$$g_l(\Delta l) = \frac{1}{\sqrt{2\pi} \cdot \Delta l_{\max}} \cdot e^{-\frac{\Delta l^2}{2\Delta l_{\max}^2}} \quad g_\varphi(\Delta \varphi) = \frac{1}{\sqrt{2\pi} \cdot \Delta \varphi_{\max}} \cdot e^{-\frac{\Delta \varphi^2}{2\Delta \varphi_{\max}^2}} \quad (7)$$

The quantities Δl_{\max} and $\Delta \varphi_{\max}$ are respectively the maximum Δl and $\Delta \varphi$ calculated between the current origin and all the destination hypotheses.

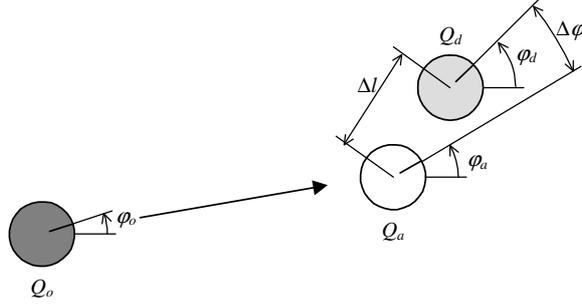


Fig. 6 Parameters for the action model

4.4 Sensor model

In many localization systems, the environment is sensed through low-dimensional devices, like sonar or laser, for which accurate models are already available [21, 22]. Other approaches instead use vision to calculate the robot's position with respect to some particular features. In [7], for example, an omni-directional image is processed using a ray-tracing method, simulating a laser range sensor that returns distances of chromatic-transition features. Even in this case, an accurate model is provided, the parameters of which are extracted by a modified EM algorithm [23] applied to a set of 2000 sample images. There are also other approaches where the sensor models are learned with neural networks using data from both vision and sonar [24-26].

The data given by our image-based place recognition, that is, the *IMA*'s matching-value, differs from all the above-mentioned approaches. The sensor model is implicitly "included" in the pre-recorded panoramic image, in a way conceptually similar to [27]. Ideally, a new image would return 1 in case of perfect match with a portion of the panorama and would decrease to 0 as the match deteriorates. Therefore, given the current state, the probability of the observation can be considered the matching-value calculated by the *IMA*. With the notation introduced earlier and calling v the observation v_t , the sensor model can be written as follows:

$$P(v_t | s_t) \equiv P(v | d) = MATCH(d) \quad (8)$$

where $MATCH(d)$ is the value of the variable *BEST_MATCH* in Table 1 (or Table 2, if enhanced by digital zoom) for the destination d .

4.5 Update of the activities

Activities are updated with the same formulae of Markov localization, but taking into account our previous assumptions. Thus, given a set of destinations $d \in D$ and origins $o \in O$, the procedure for calculating new activities, using (6) and (8), is the following:

$$1) \text{ Prediction:} \quad P'(d) = \sum_{o \in O} P(d | o, a) Act(o) \quad (9)$$

$$2) \text{ Update:} \quad P''(d) = P(v | d) P'(d) \quad (10)$$

$$3) \text{ Normalization:} \quad Act(d) = \frac{P''(d)}{\sum_{d \in D} P''(d)} \quad (11)$$

4.6 Odometry reset and visual compass

An important role in the selection of the current destination is played by odometry. Indeed, the prediction step (9) makes use of the action model (6), which strongly depends on the odometry's information. This is reset every time an update of the topological position has been performed. In general, the fact that the topological area is reasonably small, if compared to the distance between two consecutive destinations, reduces the effect of the error introduced by the reset. On the other hand, the advantage is significant, since it fixes a limit to the cumulative error of the dead reckoning.

The heading angle has a double importance: it affects directly the action model and, since related to the internal frame of reference of the robot, it influences also the parameter Δl . In many applications, instead of considering the heading angle computed using encoders, an external magnetic compass is mounted on the robot [27, 28]. This has the advantage of being independent from the cumulative errors of odometry, since it gives an absolute direction for North. Unfortunately, such a device is not immune from errors, which are mainly due to metallic objects in the proximity of the robot.

In our approach, another original solution was chosen. Since for every new environment an up-to-date map and fresh panoramic images are needed, the latter are always constructed starting from the same direction. This permits the robot to recover its absolute heading using equation (1), like if provided with a "visual compass". A similar approach, based instead on an omni-directional camera, has been recently implemented also in [29]. To limit the error cases of using a wrong panoramic image, the odometry's heading angle is corrected only when the matching-value of the estimated destination is higher than a given threshold.

4.7 Localization algorithm

In order to reduce the computational expense, the whole localization algorithm is executed only after the robot has moved a certain distance or has rotated through a minimum angle. This also has the advantage of effectively generating new different destinations (i.e. different states), reducing instances of failure. The localization algorithm is summarized in the pseudo-code of Table 3. The value ε_M is the threshold used for extracting the destinations with the best matching-values; ε_φ is the threshold on the matching-value for correcting the odometry's heading ($\varepsilon_\varphi \geq \varepsilon_M$); both the quantities ε_M and ε_φ are determined empirically. The destination with the higher activity is d^* , which corresponds to the estimated state $\langle x^*, y^*, \varphi^* \rangle$; finally, o^* is the most likely origin, that is, the d^* extracted at the previous time-step.

Table 3 Localization algorithm

```

Calculate the location given by  $o^*$  plus odometry displacement
Find the topological position  $d_0 \in D$  closer to such a location
Use IMA to compare the new image with all the panoramic images in the map
Extract all the possible destinations  $d \in D$  with matching-value  $MATCH(d) > \varepsilon_M$ 
if no new destinations are generated with IMA
    Return the topological position  $d_0$ 
    END
end if

for each  $d \in D$ 
    for each  $o \in O$ 
        Calculate  $\Delta l$  and  $\Delta \varphi$ 
        Keep track of  $\Delta l_{max}$  and  $\Delta \varphi_{max}$ 
    end for
end for

```

```

for each  $d \in D$ 
    Calculate  $P(d \mid o, a)$  with (6)
    Update the activity (9, 10)
    Keep track of the destination  $d^*$  with the higher activity
end for

Normalize the activities (11)

Reset the odometry's coordinates

if  $MATCH(d^*) > \varepsilon_\phi$ 
    Set the odometry's heading to the angle  $\phi^*$ 
end if

The destinations become the next origins,  $O \leftarrow D$ 

Return  $d^*$ 

END

```

5. Experimental results

In this section we present the results of experiments conducted in the Neuro-Robots Laboratory at the University of Sunderland. This consists of a room approximately $6 \times 6 \text{m}^2$, with typical office furniture, and an adjacent corridor connected through a small hall. Along two sides of the office there are large windows, resulting in particularly challenging light conditions. The robot used is an ActivMedia PeopleBot (Fig. 7) provided with a perspective camera and an on-board computer Pentium III 700MHz with 256MB of RAM. Grey-scale images with resolution 72×58 pixels were used and the number of slots chosen for *IMA* was 8, with thresholds $\varepsilon_M = 0.5$ and $\varepsilon_\phi = 0.6$. The whole localization system, implemented in C++ without any particular optimization, worked in real-time on the robot's computer. The topological position was recovered whenever the robot moved 0.5m or rotated 10° ; the maximum update frequency was 2Hz, which is normally adequate for the tasks of a service robot. We mapped up to 15 locations for the experiments here presented, but the system was still fast enough in other tests with more than 20 different locations. Our approach is therefore feasible for real-time localization in small and medium indoor environments, although larger areas could also be covered if more recent and fast hardware was available.



Fig. 7 Mobile robot used for the experiments

5.1 Place recognition performance

In this section we present results of the matching algorithm applied to panoramic images. Fig. 8 shows a panoramic image reconstructed from snapshots taken in the center of the laboratory, using the procedure illustrated in Section 3.2. In particular, we used 12 snapshots taken at intervals of $\sim 30^\circ$. Note that the robot can be rotated quickly, therefore the whole panorama's reconstruction takes less than 1 minute; the algorithm indeed is capable of aligning the sequence of snapshots correctly, even if the angle step varies of several degrees.



Fig. 8 Panoramic image of reference

A few moments later, after the panorama was recorded and the software reset, the robot was made to perform a complete rotation on the same point, approximately at $10^\circ/\text{s}$. The relative output of the *IMA* is the solid line in Fig. 9. It can be seen that the match has a mean value greater than 0.8. The worst cases, for which *IMA* returned a value of approximately 0.7, correspond to the cupboard ($\sim 100^\circ$, right part of the panorama) and the shelves ($\sim 350^\circ$, left part of the panorama). This is probably due to a combination of imprecision in the panoramic image and errors caused by changes in the perspective. On the same graph, it is also illustrated the output of a second turn, when the person seating in front of the desk moved away. The relative change can be observed on the dashed line of Fig. 9, where the output decreases at about 270° (direction where the person was). It is important to note that, even if the output decreased, the position inside the panoramic image, relative to the best match, was still correct, and so it was the heading angle of the visual compass.

To test the robustness of the place recognition, we performed a similar experiment the day after using the same panoramic image. Also, to make the experiment more challenging, during the observation a person was walking around the robot about one meter far. The result is shown in Fig. 10, where the new matching output (dashed line, relative to the one-day old panoramic image) is compared to the previous one (solid line). Despite the small decrease due to different light conditions and objects' position, the main loss of quality is due to the absence of the person sitting on a chair (at $\sim 300^\circ$). In particular, the arrow on the graph indicates a point where the estimated position inside the panorama was completely wrong. The four points A, B, C, D are relative instead to the instants when the person, walking around the robot, was occluding the camera's view.

The last result about the *IMA* applied to panoramic images is perhaps the most important. As its main purpose is distinguishing different locations, we wanted to compare the result obtained in the last case (old panoramic image and occluding person) with the output obtained from another location, in the same room but one meter far from the original position. The resulting output is represented by the solid line in Fig. 11 and compared to the previous case, which is the dashed line. Although the output in the new location is not very low, in general it is well distinguishable from that one obtained from the original position. Failure cases, like the overlap indicated by an arrow in Fig. 11 (at $\sim 315^\circ$), are situations of perceptual aliasing. Here it is clear the need of additional information for resolving the ambiguity, i.e. integrating odometry and previous states with a Markov-like approach, as explained in Section 4.

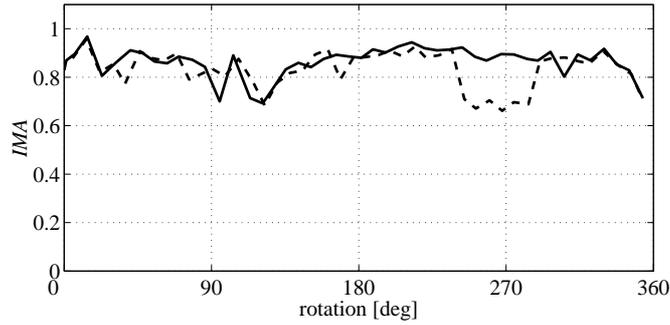


Fig. 9 *IMA*'s outputs for panoramic image

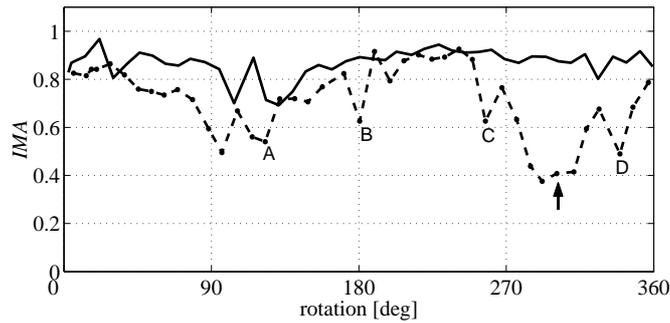


Fig. 10 *IMA*'s output the day after with occlusions

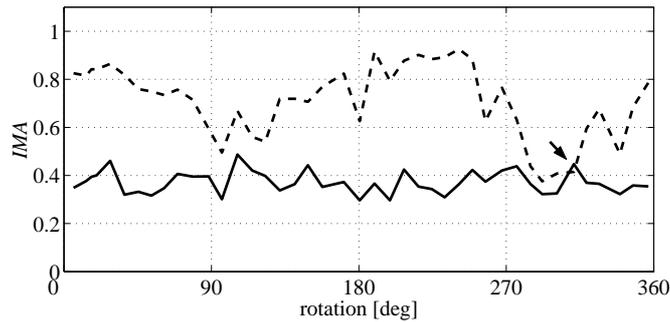


Fig. 11 *IMA*'s output from a different location

5.2 Correction of the visual compass

This section illustrates some results regarding the heading angle extraction using the *IMA* and equation (1). In a first experiment, the robot rotated around a position where a panoramic image was previously reconstructed. Data was collected for the heading angle given by odometry and by the vision during 10 consecutive rotations, measuring at intervals of 45° . Fig. 12 shows the final results. The real angle is on the abscissa and the heading angle measured by the robot is on the ordinate; the dashed line refers to the odometry and the solid one is the angle extracted using the *IMA* as visual compass. It is clear that the angle given by odometry becomes unreliable after a few rotations due to the internal cumulative error. At the seventh rotation, the odometry's error already reached -45° with respect to the real direction. Instead, the error of the heading angle given by the visual compass is always between $\pm 10^\circ$, without suffering of any cumulative error.

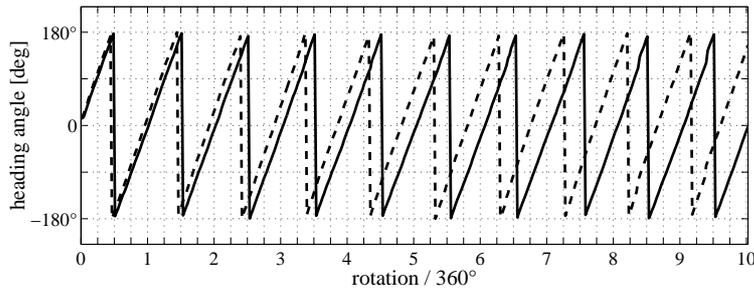


Fig. 12 Comparison of heading angle from odometry and visual compass

Although the precision of the visual compass is not sufficient to give a perfect measure of the robot’s orientation, it is still good enough to correct from time to time the odometry and help with the localization. This is demonstrated for example with the following experiment. The robot performed 10 rounds following the path of Fig. 13a, which shows a schematic of the laboratory and eight topological nodes on a grid of 1m×1m per square. The coordinates given by the odometry during the robot’s motion are illustrated in Fig. 13b. Because of the cumulative error affecting the odometry, the points are spread in the room instead of being concentrated in proximity of the path. Fig. 13c instead shows the same points after the correction of the visual compass as part of the localization process. It can be noted that the new distribution is much closer to the real path followed by the robot, despite some outliers due to odometry reset or localization errors.

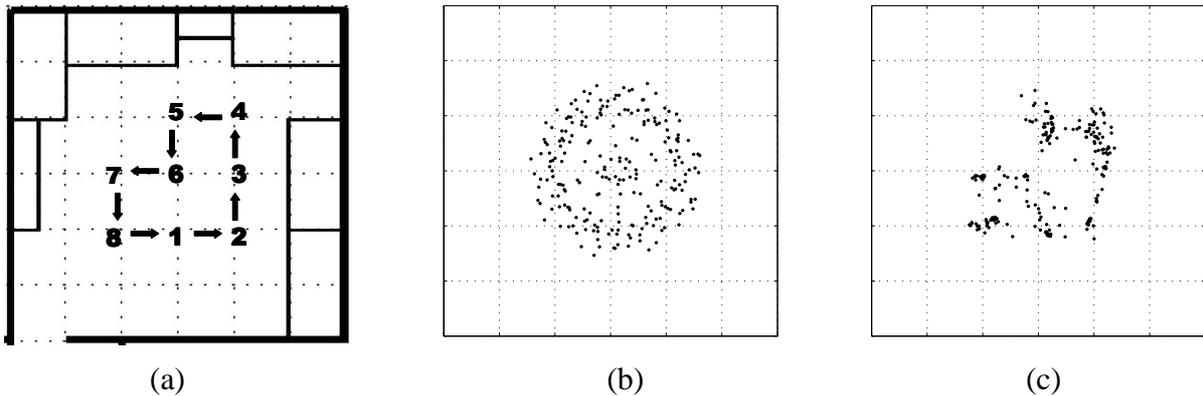


Fig. 13 Reference path and odometry correction with visual compass

5.3 Effect of the digital zoom

This section demonstrates that the use of digital zoom increases the place recognition capability, enabling the robot to identify not just an exact point in the environment, but the whole of the surrounding area. The following results are relative to a normal single image of reference, rather than a panoramic one, in order to reduce the noise and avoid wrong matches. The same principles however are still valid when using panoramic images.

In Fig. 14, the observed scene and relevant graphs for three different zoom factors are illustrated. The distance of the robot from the wall in the middle of the scene was about 4m; the robot moved from -1m to +1m with respect to the original position. The variation of the *IMA*’s output is shown on the graphs, where the dashed line is the reference, without any zoom, and the solid line is relative to the current zoom factor.

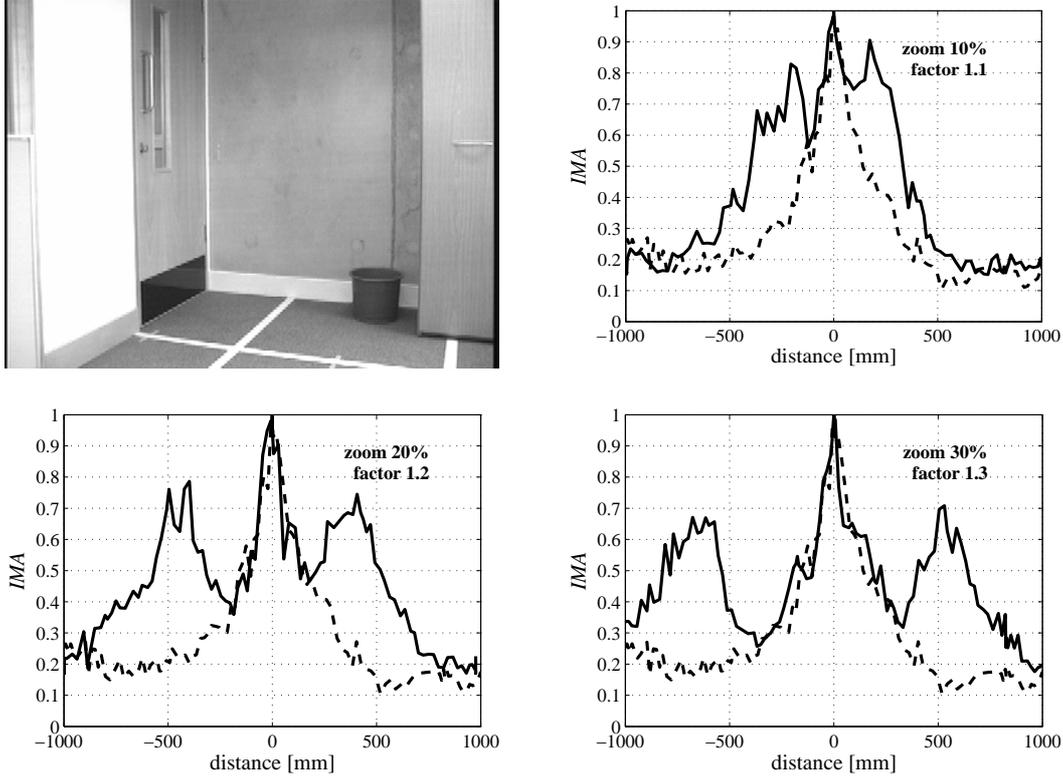
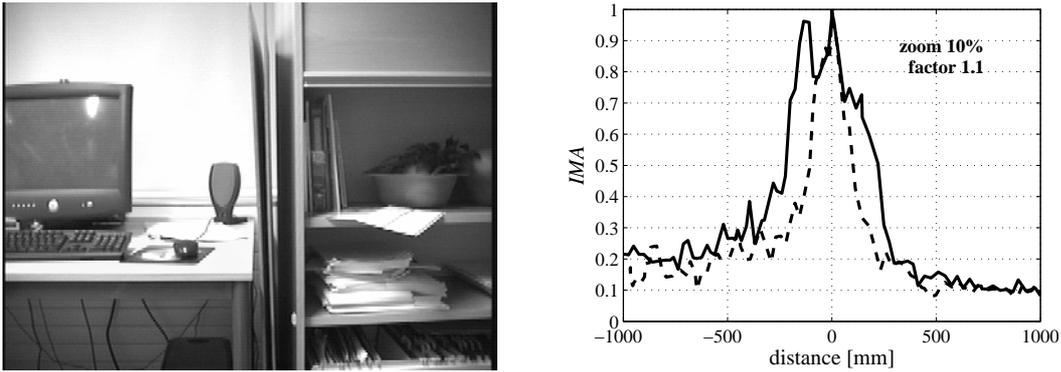


Fig. 14 *IMA*'s performances varying the digital zoom

Two important considerations arise from observation of the graphs. First, the output is similar to the combination of three different peaked distributions, as anticipated. Unfortunately, the amplitude of the external peaks decreases considerably when the zoom factor augments. Despite the fact that, when translating, it is not easy to keep the robot constantly on the same direction, the main reason for this decrease is the loss of resolution implicit in the zoom process. The second concerns the gaps between the external picks and that one in the middle. From these it can be seen that the internal (local) minimum goes quickly below 0.5 already with a zoom of 20%. This is because the peaked curves are not very wide and the distance from the observed scene is quite long – recall from (3) that the virtual displacement obtained with the digital zoom is directly proportional to this distance. Note also that the formula given in (3) was an approximation to an ideal case, but in the real world the virtual displacement is influenced by several other factors. For example, in case of a zoom of 20% ($\rho_{in} = 1.2$) the hypothetical displacement Δx for a distance $D = 4\text{m}$ should be 0.67m; in practice, the graph shows two external peaked curves not further than 0.5m from the origin. Again, the higher the zoom factor, the bigger the error.

With the next result, we would like to demonstrate also how the distance of the observed scene influences the effect of the digital zoom on the recognition area. In Section 3.4 we stated that the shape of the recognized region depends on the environment because of the linear relation (3) between virtual shift and distance of the scene. According to that, we would expect a reduction on the width of the recognition's curve when observing a closer scene. Therefore we repeated the same test illustrated above, but this time placing the robot just 1m far from the closest obstacles. The result for a zoom of 10% is illustrated in Fig. 15. Comparing this new graph with the previous one in Fig. 14 (for zoom 10%), it is evident that the recognition interval becomes smaller, and this is due to the decrease of the relative virtual displacement for the current observation. In general then, the region where the zoom-based recognition holds becomes thinner (wider) in the direction of a closer (farther) scene.

Fig. 15 *IMA* and digital zoom on a closer scene

Apart from these limitations, if compared to the output without any zoom, even a value of 10% is a big improvement of the place recognition. For the next localization experiment, this was the value used. First, we disabled the digital zoom procedure on the localization system and moved the robot along the path of Fig. 13a, carefully passing through the centers of each of the eight topological nodes. After one turn and 19 updates of the localization, we had an error due to the estimation of a wrong place, which was however an adjacent node of the correct one (at less than 1.5m, about the length of a square's diagonal on the grid of Fig. 13a). Still without digital zoom, the robot performed a new turn following the same path, but this time avoiding the centers of the nodes. As expected, the number of localization updates decreased to 14 because the *IMA* output was often below the threshold ε_M . Furthermore, 4 of those updates were wrong, with a couple of estimations involving nodes more than 1.5m far from the correct locations. Finally, we moved the robot on the same path, avoiding again the nodes of the centers but making use of the digital zoom procedure. As reported in Table 4, in this case the localization succeeded, generating 22 correct estimations without any error. These results show that the digital zoom was an essential part of the localization system; therefore it was always used in the following experiments.

Table 4 Errors with and without digital zoom

Case	no zoom & center	no zoom & no center	zoom & no center
Update steps	19	14	22
Errors due to adjacent nodes:	1	2	0
Errors due to distant nodes:	0	2	0
Total number of errors:	1	4	0

5.4 Localization performances in a dynamic environment

To test the localization system in a dynamic environment, updated panoramas were used, all reconstructed the same day. The robot performed 10 rounds following the same path with eight nodes in Fig. 13a. During the experiments, the robot was always avoiding the exact centres of the topological position, so to force the use of the digital zoom for place recognition. In the meanwhile, two people were continuously moving around the robot, sometimes walking or standing in front of the camera and sometimes simply sitting on chairs. Examples of such situations are shown in the robot's snapshots of Fig. 16.

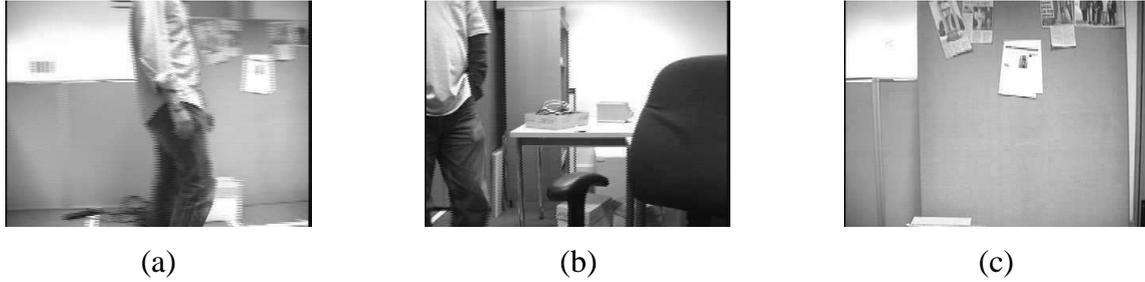


Fig. 16 Snapshots of error cases

The results presented in Table 5 are encouraging; indeed there were only 3 incorrect localizations out of a total of 253 update steps. Two of them happened because of people obstructing the scene, so the robot computed that it was at node 3 instead of the correct positions of nodes 5 and 6. The reduced video information obtained from the real positions has not been sufficient to resolve the perceptual aliasing, even with the odometry's help. The relative robot's snapshots are shown in Fig. 16a and 16b. In the third error case, the robot was at node 7, but the localization estimated node 1, although nobody was obstructing the view at that moment. The reason is probably the poor quantity of features in that particular scene, which can be seen in Fig. 16c. However, the three errors, in terms of distance from the correct node, were all less than 1.5m; that is, in the worst case the estimation was a topological node adjacent to the correct one.

For comparison, in Table 5 are reported also the error cases for the same localization experiment using only place recognition, without the Markov updating process. As we can see, the performance decrease is considerable, with a total number of 37 localization errors. Among these, 34 were assigned to adjacent nodes, less than 1.5m far from the correct position, instead 3 errors involved more distant nodes. From these results, it is obvious the benefit given by the Markov update to the localization system.

Table 5 Errors with and without Markov update

Case	with Markov update	without Markov update
Update steps	253	253
Errors due to adjacent nodes:	3	34
Errors due to distant nodes:	0	3
Total number of errors:	3	37

5.5 Localization in a bigger environment

To show the performance of the localization system in a bigger environment, we used an adjacent corridor, connected to the laboratory through a small entry and both already mapped several days before. The new locations are poor of features and quite narrow, just $2 \times 2m^2$ for the entry and about $2 \times 10m^2$ for the corridor, both illuminated by artificial light. Fig. 17 shows a panoramic image taken from the corridor and Fig. 18 illustrates the new map.



Fig. 17 Panoramic image of the corridor (from node 13 of the map)

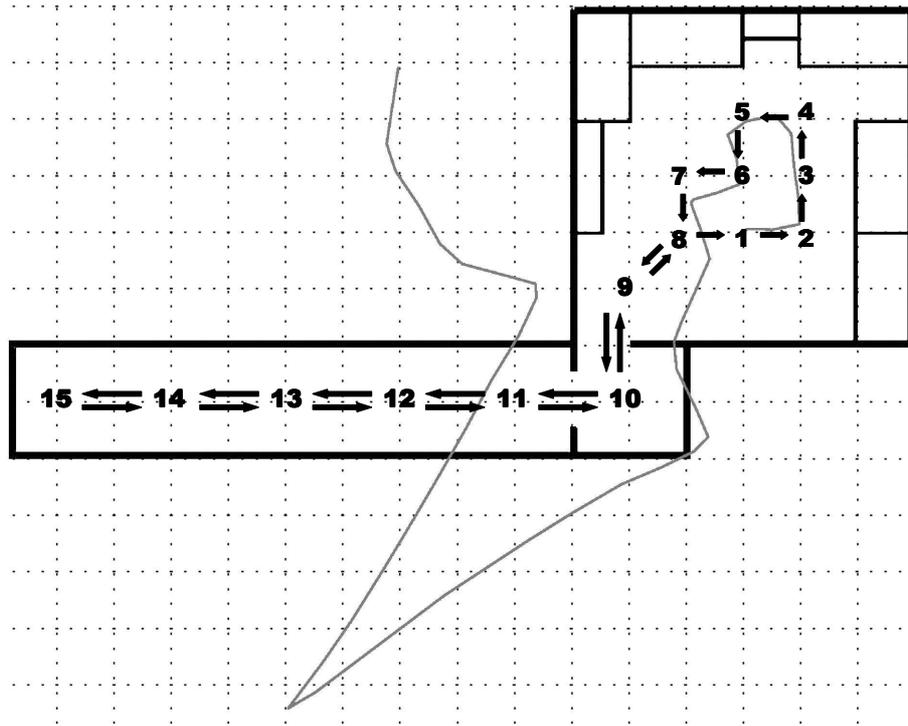


Fig. 18 Map of laboratory and corridor with reference path. The gray line is the odometry.

The robot followed the new path drawn in Fig. 18, starting from node 1 until node 15, at the end of the corridor, and then back to node 1. In the figure is also illustrated the path given by the odometry when performing such a trip, showing clearly the effects of its cumulative error. Like for the previous case, the robot always avoided the exact center of the topological places; also, all the parameters of the localization algorithm (slots, zoom, thresholds, etc.) were the same adopted for the laboratory. We repeated the trial 3 times, collecting data for a total number of 202 update steps, and the results are reported in Table 6. We can notice an increment of the error cases, which still represent however a small percentage of the total steps' number. The localization's failures were distributed on the whole environment, with a relatively high concentration (4 errors) inside the entry room, at node 10. The reason is that the panoramic image of this room was taken with two doors closed, while during the experiment the same doors were open. In such a small place, these doors took almost half of the panoramic image and their state influenced much the recognition performance. However, the localization in general was very reliable and the few errors were always limited to adjacent nodes, less than 1.5m far from the correct ones.

Table 6 Errors in a bigger environment

Update steps	202
Errors due to adjacent nodes:	9
Errors due to distant nodes:	0
Total number of errors:	9

6. Conclusions

An appearance-based localization system for indoor environments has been developed, making use of a simple unidirectional camera and odometry information. The approach is strongly based on a novel place recognition algorithm (*IMA*) enhanced by digital zoom. The same algorithm also permits the generation of panoramic images used for mapping the environment and, from these, to estimate the absolute robot orientation with a visual compass. The latter in particular gave promising results, considering also the hardware limitations that had to be dealt with. Finally, within a probabilistic framework, odometry is integrated with the visual information to resolve cases of ambiguity. The experiments presented show the robustness of the approach, even in case of dynamic environments, making the localization suitable for service-robot applications.

There are two main topics that should be explored in the future: (i) the automatic update of panoramic images, and (ii) the use of incremental digital zoom. The first would boost the place recognition and would be a natural step towards a complete system of self-localization and map-learning (SLAM). The second is an innovative technique that has just been introduced, but which shows great potential to improve the effectiveness of the localization. By adding incremental digital zoom to the frame captured by the camera, “off node” map locations could be identified. It would be worthwhile extending this technique to support interpolation of location between map nodes.

References

- [1] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, D. Schulz, MINERVA: A Second-Generation Museum Tour-Guide Robot, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'99), 1999, v. 3, pp. 1999-2005.
- [2] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, Experiences with an interactive museum tour-guide robot, *Artificial Intelligence* 114 (1-2) (1999), pp. 3-55.
- [3] B.A. Maxwell, L.A. Meeden, N. Addo, L. Brown, P. Dickson, J. Ng, S. Olshfski, E. Silk, J. Wales, Alfred: The Robot Waiter Who Remembers You, in: Proc. of the AAAI Workshop on Robotics, Florida, USA, 1999.
- [4] G. Gini, A. Marchi, Indoor Robot Navigation with Single Camera Vision, in: Proc. of Pattern Recognition in Information Systems (PRIS 2002), 2002, pp. 67-76.
- [5] N.X. Dao, B.J. You, S.R. Oh, M. Hwangbo, Visual Self-Localization for Indoor Mobile Robots Using Natural Lines, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'03), Las Vegas, USA, 2003, pp.1252-1257.
- [6] S. Enderle, H. Folkerts, M. Ritter, S. Sablatnög, G. Kraetzshmar, G. Palm, Vision-based Robot Localization using Sporadic Features, in: Proc. of the Int. Workshop on Robot Vision, Auckland, New Zealand, 2001, pp. 35-42.
- [7] E. Menegatti, A. Pretto, E. Pagello, A New Omnidirectional Vision Sensor for Monte-Carlo Localization, in: Proc. of RoboCup Symposium, 2004, pp. 97-109.
- [8] I. Ulrich, I. Nourbakhsh, Appearance-Based Place Recognition for Topological Localization, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'00), San Francisco, USA, 2000, pp. 1023-1029.

- [9] H. Andreasson, T. Duckett, Topological Localization for Mobile Robots using Omni-directional Vision and Local Features, in Proc. of the 5th Symposium on Intelligent Autonomous Vehicles (IAV), Portugal, 2004.
- [10] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110.
- [11] A. Torralba, K. P. Murphy, W. T. Freeman, M. A. Rubin, Context-based vision system for place and object recognition, *IEEE Int. Conf. on Computer Vision*, France, 2003.
- [12] J.S. Gutmann, W. Burgard, D. Fox, K. Konolige, An Experimental Comparison of Localization Methods, in: Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS'98), 1998, pp. 736-743.
- [13] J.S. Gutmann, D. Fox, An experimental Comparison of Localization Methods Continued, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'02), 2002, pp. 454-459.
- [14] S. Kristensen, P. Jensfelt, An Experimental Comparison of Localization Methods, the MHL Sessions, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robot and Systems (IROS'03), 2003, pp. 992-997.
- [15] A.C. Bovik, C.W. Chen, D.B. Goldgof, T.S. Huang, *Advances in Image Processing and Understanding*, World Scientific, Singapore, 2002.
- [16] K. Zuiderveld, *Graphics gems IV*, Academic Press Professional, Inc., San Diego, 1994.
- [17] D. H. Ballard, C. M. Brown, *Computer Vision*, Prentice Hall, New York, 1982.
- [18] D. Fox, Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation. Doctoral Thesis. Institute of Computer Science III, University of Bonn, Germany, 1998.
- [19] D. Filliat, J. A. Meyer, Global localization and topological map learning for robot navigation, in Proc. of the 7th Int. Conf. on Simulation of Adaptive Behavior (SAB'02), 2002, pp. 131-140.
- [20] P. Jensfelt, S. Kristensen, Active global localization for a mobile robot using multiple hypothesis tracking, *IEEE Trans. on Robotics and Automation*, 17(5) (2001), pp. 748-760.
- [21] W. Burgard, D. Fox, D. Hennig, T. Schmidt, Estimating the absolute position of a mobile robot using position probability grids, in: Proc. of the 13th National Conf. on Artificial Intelligence (AAAI'96), Portland, Oregon, USA, 1996, pp. 896-901.
- [22] H.P. Moravec, Sensor fusion in certainty grids for mobile robots, *AI Magazine* (1988) pp. 61-74.
- [23] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, vol. 39 B (1977), pp. 1-38.
- [24] S. Thrun, Finding landmarks for mobile robot navigation, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'98), 1998, pp. 958-963.
- [25] S. Thrun, Bayesian Landmark Learning for Mobile Robot Localization, *Machine Learning* 33 (1998), pp. 41-76.
- [26] S. Oore, G. E. Hinton, G. Dudek, A mobile robot that learns its place, *Neural Computation* 9(3) (1997), pp. 683-699.
- [27] T. Duckett, U. Nehmzow, Mobile robot self-localization using occupancy histograms and a mixture of Gaussian location hypotheses, *Robotics and Autonomous Systems* 34 (2001), pp. 117-129.
- [28] T. Duckett, S. Marsland, J. Shapiro, Fast, On-Line Learning of Globally Consistent Maps, *Autonomous Robots* 12 (2002), pp. 287-300.
- [29] F. Labrosse, Visual compass, in Proc. of Towards Autonomous Robotic Systems (TAROS'04), 2004, University of Essex, Colchester, UK.



Nicola Bellotto received his Laurea degree in Electronic Engineering from the University of Padua in Italy. He is currently a PhD candidate in robotics at the University of Essex in UK. His doctoral thesis focuses on Multisensor Data Fusion for Human-Robot Interaction. Other research interests include robot localization, computer vision and embedded systems programming. Before joining the Human Centred Robotics Group in Essex, he has been an active member of the Intelligent Autonomous System Laboratory in Padua and of the Centre for Hybrid Intelligent Systems at the University of Sunderland. He gained also several years of professional experience in Italy and UK as embedded systems programmer and software developer for entertainment robotics.



Kevin Burn received a BSc degree in Mechanical Engineering from Newcastle University in 1984 and an MEng from Durham University in 1986. After working in the power industry for several years he returned to Newcastle University as a Research Associate and was awarded his PhD in 1994 for research into robotics and teleoperator systems. He is now a Senior Lecturer at Sunderland University in the School of Computing and Technology, where his research interests are in robotics and intelligent systems.



Professor Eric Fletcher graduated from the University of Hull with an honours degree in Physics and Pure Mathematics in 1965. He obtained a Ph.D. in Geophysics from the University of Newcastle upon Tyne in 1975. Subsequently he did research in Solid State Physics specialising in Auger Spectroscopy of Glass surfaces. In 1986 he transferred to Computer Science initially specialising in Mathematical Modelling Simulation and Decision Support Systems. In recent years he has concentrated on image analysis applied to biological systems and traffic flow monitoring. He has supervised 14 Ph.D. students and has over 80 publications in Computer Science and Solid State Physics. In December 2003 he retired from the full time post of Professor of Applied Computing at the University of Sunderland and is now Emeritus Professor of Computing and is continuing his interests in image analysis.



Professor Stefan Wermter holds the Chair in Intelligent Systems at the University of Sunderland, UK and is the Director of the Centre for Hybrid Intelligent Systems. His research interests are in Intelligent Systems, Neural Networks, Cognitive Neuroscience, Hybrid Systems, Language Processing, and Learning Robots. He has an MSc from the University of Massachusetts, USA and a PhD and Habilitation from the University of Hamburg, Germany, both in Computer Science and was a Research Scientist at Berkeley, USA before joining the University of Sunderland in 1998. Professor Wermter has written or edited five books and published about 130 articles on this research area, including books like "Hybrid Connectionist Natural Language Processing" or "Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing", "Hybrid Neural Systems", "Emergent Neural Computational Architectures based on Neuroscience" and "Biomimetic Neural Learning for Intelligent Robots".