Dear Mr./Mrs.,

with the present letter I wish to submit our contribution to IEEE Transaction on Robotics as REGULAR PAPER.

AUTHORS:

Emanuele Menegatti, Alberto Pretto, Alberto Scarpa and Enrico Pagello

CORRESPONDING AUTHOR:

Emanuele Menegatti
emg@dei.unipd.it
Dept. of Information Engineering
via Gradenigo 6/a
35131 Padova
ITALY
tel. +39 049 827 7722
fax. +39 049 827 7799

OTHER AUTHORS' ADDRESSES:

Enrico Pagello
epv@dei.unipd.it
Dept. of Information Engineering
via Gradenigo 6/a
35131 Padova
ITALY
tel. +39 049 827 7687
fax. +39 049 827 7799

Alberto Pretto
alberto.pretto@email.it
Dept. of Information Engineering
via Gradenigo 6/a
35131 Padova
ITALY

Alberto Scarpa
alscarpa@dei.unipd.it
Dept. of Information Engineering
via Gradenigo 6/a
35131 Padova
ITALY

TITLE:

Omnidirectional vision scan matching for robot localization in dynamic environments

ABSTRACT:

The localization problem for an autonomous robot moving in a known environment is a well studied problem that have had many elegant solutions. Nevertheless, the robot localization in a dynamic environment populated of several moving obstacles is still a challenge for research. In this paper, we used the omnidirectional camera mounted on a mobile robot to perform a sort of scan matching. The omnidirectional vision system finds the distances of the closest color transitions in the environment, mimicking the way laser range finders detect the closest obstacles. The similarity of our sensor with classical range finders allows the use of almost the same Monte-Carlo algorithms, with the additional advantage of being able to easily detect occlusions caused by moving obstacles. The proposed system was initially implemented in the RoboCup Middle-size domain, but the experiments we present in this paper prove it to be valid in a general indoor environment with natural color transitions. We present localization experiments both in the RoboCup environment and in an unmodified office environment. In addition, we proved the robustness of the sensor to occlusions caused by other robots moving in the environment. The localization system runs in real-time on a low-cost CPU.

INDEX TERMS:

omnidirectional vision, Monte-Carlo Localization, dynamic environments, robustness to sensor occlusion, mobile robot

DECLARATION:

The submitted material has not been published or submitted to any other journal. Some parts of this work was previously published in conferences as detailed in the bibliography.

Best regards,
Emanuele Menegatti

Omnidirectional vision scan matching for robot localization in dynamic environments

E. Menegatti, *Member, IEEE,* , A. Pretto, A. Scarpa and E. Pagello†, *Member, IEEE,*
Intelligent Autonomous Systems Laboratory
Department of Information Engineering, The University of Padua, Italy
†*also with:* Institute ISIB of CNR Padua, Italy
{emg, albe76, alscarpa, epv}@dei.unipd.it

***Abstract*— The localization problem for an autonomous robot moving in a known environment is a well studied problem that have had many elegant solutions. Nevertheless, the robot localization in a dynamic environment populated of several moving obstacles is still a challenge for research. In this paper, we used the omnidirectional camera mounted on a mobile robot to perform a sort of scan matching. The omnidirectional vision system finds the distances of the closest color transitions in the environment, mimicking the way laser range finders detect the closest obstacles. The similarity of our sensor with classical range finders allows the use of almost the same Monte-Carlo algorithms, with the additional advantage of being able to easily detect occlusions caused by moving obstacles. The proposed system was initially implemented in the RoboCup Middle-size domain, but the experiments we present in this paper prove it to be valid in a general indoor environment with natural color transitions. We present localization experiments both in the RoboCup environment and in an unmodified office environment. In addition, we proved the robustness of the sensor to occlusions caused by other robots moving in the environment. The localization system runs in real-time on a low-cost CPU.**

I. Introduction

Localization is the fundamental problem of estimating the pose of the robot inside the environment. Some of the most successful implementations of robust localization systems are based on the Monte-Carlo localization (MCL) approach [3].

The MCL approach has been implemented on robots fitted either with range finder sensors or with vision sensors. Lately, vision sensors have been preferred over range finders, because they are cheaper and provide a more reach information about the environment. Moreover, they are passive sensors, so they do not interfere with other sensors or do not pose safety concerns in populated environments.

In this work, we consider the problem of Monte-Carlo Localization using an omnidirectional camera. The vision system has been designed to extract from the omnidirectional images the distances of the closest color transitions of interest existing in the environment. In some way, our system uses an omnidirectional vision system to emulate and enhance the behavior of range-finder sensors. This results in a scan of the current location similar to the one obtained with a laser range finder, enabling the use of Monte-Carlo algorithms only slightly modified to account for this type of sensors.

The most significant advantages with respect to classical range finders are: (i) a conventional range-finder device senses the vertical obstacles in the environment, while our sensor is sensitive to the chromatic transitions in the environment gathering a more reach information, and (ii) our sensor can reject some measurements if an occlusion by an obstacle is detected.

Combining the omnidirectional vision scans with the Monte-Carlo algorithms provides a localization system robust to occlusions and to localization failures, and able to exploit as localization clues the natural color transitions existing in the environment.

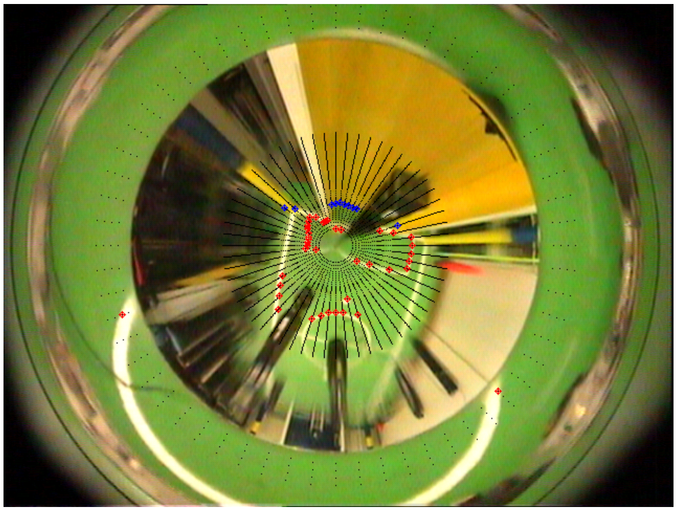


Fig. 1. A snapshot of the algorithm seeking the image for chromatic transitions. Green-white chromatic transitions are highlighted with red crosses, green-yellow transitions with blue crosses, black pixels represent the receptor pixels used for the scan that is performed in a discrete set of distances. Notice the crosses in the outer part of the mirror: this part is used for low distance measures. If a not expected color transition is detected (e.g. another robot is occluding the sensor, like the three black robots in the image) the scan is stopped and a typical value (*FAKE_RAY*) is stored in the distances vector.

The Middle-Size RoboCup field is a perfect testbed for studying the localization problem in a highly dynamic and densely populated environment. In a dynamic multi-agent world, a precise localization is necessary to effectively perform high level coordination behaviors. At the same time the presence of other robots makes localization harder to perform. In fact, if the density of moving obstacles in the environment is high, occlusion of the robot's sensors is very frequent. Moreover, if, like in RoboCup Middle-Size, collisions among robots are frequent, the localization system must be able to recover from localization errors after collisions.

In this paper, we explicitly discuss the robustness of our system with respect to the classical problems of global localization, position tracking, and robot kidnapping, as introduced in [13]. We provide a detailed discussion of the robustness against sensor's occlusion when the robot moves in a densely populated environment, as introduced in [14]. In addition, we present the experimental evidence the system we developed is not limited to the RoboCup domain, but works in a generic unmodified office-like environment (as shown by Fig. 15 and Fig. 16). The only assumption that must be satisfied is that a geometric map of the environment is available and the map reproduces the transitions of colors in the environment.

The paper is organized as follows. Section Iintroduced the main ideas presented in this paper. Section II discusses previous researches related to this topic. Section III describes how we process the omnidirectional image to obtain range information. Section IV summarizes the well known Monte-Carlo localization (MCL) algorithm and discusses the motion model and sensor model used in the experiments and the modifications to the classical MCL to adapt it to our sensor. In Section V, we present the experiments performed with the robot in the RoboCup Middle-Size field of play and in the corridors of the building of our department. A detailed analysis of the performances and the robustness of the localization system is presented with a particular attention to to occlusion caused by other robots. Eventually, in Section VI conclusions are drawn.

II. Related Work

The seminal works on Monte-Carlo localization for mobile robots used range finders as main sensors. The range finders were used to perform scans of the static obstacles around the robot and the localization is calculates matching those scans with a metric map of the environment [3], [19]. However, several times in dynamic environments, the static features that are detectable are not enough for a robust localization (as illustrated in Fig. 2) or they may be occluded by moving obstacles. One possibility is to design algorithms able to filter out the moving obstacles in the range scans, leaving only the static obstacles that can be used as landmarks. Example are the *distance filters* [7], but usually these algorithms are computationally expensive. Another possibility is to use a reacher sensor, like a color camera, able to detect the distances of much more static features, enabling a more reliable *"scan matching"*.

In the usual approaches to Monte-Carlo localization with vision sensors, the cameras have been used either to recognize characteristic landmarks subsequently matched within a map [5], [17] or to find the reference image most similar to the image currently grabbed by the robot [9], [15], [16], [21], [22]. However, when the robot has to match the current sensors' reading with a previous sensors' reading, moving obstacles like people or other robots can impair the localization process. Several solutions have been adopted. One possibility is to look at features that cannot be occluded by moving obstacles, like the ceiling of the museum hall in [2]. However these features are not always available or they do not carry enough information.

Our sensor is able to detect occlusions as non-expected color transition, so it can use only a subset of reliable distances in the scan, obtaining a more reliable localization. Color transitions are usually available in the environment and usually they carry reach information about the environment structure (e.g. change of color of the carpet, doors with a color different from the color of the walls, etc.).

In the RoboCup Middle-Size competitions, an approach based only on laser range finders was used, very effectively, by the CS Freiburgh Team [20]. They extracted the lines of the walls from the laser scans and matched them against a model of the field of play. However, when in 2002 the walls surrounding the field were removed, the reliability of this approach was impaired by the lack of static features detectable by a range-finder sensor.

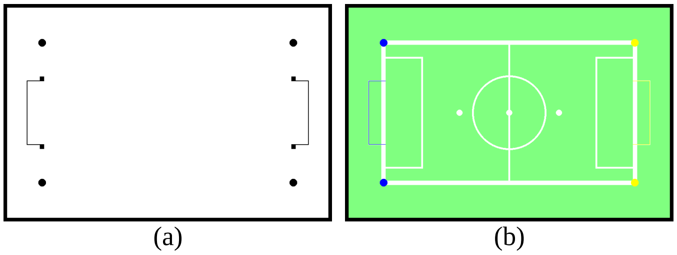


Fig. 2. The metric maps used for the computation of the expected scans: in (a) are represented the static obstacles (they are too sparse for an effective localization), in (b) are represented all the chromatic transitions of interest of the environment

In Fig. 2(a), are presented the static obstacles detectable by a range finder, in the Middle-Size field with the 2003 layout. The only detectable objects are the two goals and the four corner-posts. With a sensor sensitive to color transitions, one can detect not only the static objects in the field shown in Fig. 2(a), but also all color transitions existing in Fig. 2(b). Schulenburg *et al.* in [18] combined a laser range finder and an omnidirectional camera to extend the CS Freiburgh's approach by detecting lines both with the laser range finder and with the omnidirectional vision system. However, the integration of laser data with vision data do not improves much the localization with respect to vision data alone (due to the shortage of laser detectable features) and the image processing algorithms needed to extract the field's line are computationally demanding. Even if the algorithm we used to locate the color transitions in the image is very similar to the one of Schulenburg *et al.* , we do not need to post-process these data to extract the field's line. In fact also in [1] and especially in [10], a similar approach to find color transitions is adopted, but, again they used these data to find geometrical models of the lines in the field and matched them against a geometrical model of the world. We use the raw range scans and we will show in the remanider of the paper that a robust localization is achievable performing a scan matching between an enhanced color map of the environment and the scans of the distances of the pre-eminent color transitions in the image.

III. An omnidirectional camera as a range finder

As we said, the main sensor of our robot is an omnidirectional camera. This gives our "range finder" a 360 degrees field of view, much larger than the usual field of view of conventional range finders. The omnidirectional camera is composed by a perspective camera pointed upward to a multi-part mirror with a custom profile [12]. The custom profile of the omnidirectional mirror was designed to have good accuracy both for short and long range measurements. In fact, conic omnidirectional mirrors fail to obtain good accuracy for short distance measurements (because the area close to the robot is mapped in a very small image area), while hyperbolic mirrors fail to obtain good accuracy for long distance measurements (because of the low radial resolution far away from the sensor). With our mirror, the area surrounding the robot is imaged in the wide external ring of the mirror and the area far away from the robot is imaged in the inner part of the mirror [12]. The inner part of the mirror is used to measure objects farther than 1 m away from the robot, while the outer part is used to measure objects closer than 1 m from the robot, see Fig.1.

As we said, we search the omnidirectional image for, what we called, *chromatic transitions of interest*. In the RoboCup domain, we are interested in *green-white*, *green-blue* and *green-yellow* transitions. These transitions are related to the structure of the RoboCup fields, where the play-ground is green, lines are white, and goals and corner posts are blue or yellow. The image is scanned along radial line 6 degrees apart and with a sampling step corresponding to 4 centimeters in the world coordinate system, as shown in Fig.1. We first scan for chromatic transitions of interest close to the robot's body, in the outer mirror part, and then we scan the inner part of the image from the center of the image up to 4 meters away from the robot's body.

In RoboCup, usually, a color quantization is performed on the image before any image processing. Our system looks for the chromatic transitions of interest only along the receptors of the 60 rays depicted in Fig.1. Therefore, we do not need to color quantize the whole image, but only some of the pixels lying along the 60 rays need to be classified into one of the 8 RoboCup colors [1] plus a further class that include all colors not included in the former classes (called *unknown color*). At the setup stage, the RGB color space is quantized into the nine color classes. To achieve a real time color quantization, a look-up table is stored in the main memory of the robot. The look-up table associates every possible RGB triple to one of the 9 color classes.

The distances to the nearest *chromatic transitions of interest* are stored in three vectors [2], one for each color transition of interest. During the radial scan, we can distinguish three situations:

1) a chromatic transition of interest is found, then the real distance of that point is stored in the corresponding vector;
2) no transition of interest is detected, then a characteristic value called *INFINITY* is stored in the vector (this means no transition can be founded along this ray);
3) a non-expected transition is found, then a *FAKE_RAY* value is stored in the vector (this means something is occluding the vision sensor).

Moreover, we use the information about the static obstacles extracted from the map of Fig. 2(a) to improve the scanning process (e.g. if we find a yellow pixel, this is a goal or a corner-post, so it is not worth looking farther for a white line and so we stop the scanning process along this ray).

Algorithm 1 Omni-vision as enhanced range finder.
Function QUANT(x,y) returns quantized color of pixel x,y.
Function REAL_DIST(x,y) returns distance in real world of pixel x,y.
Pixel$(0,0)$ is located in the image center.

```
Ensure: $dist\_white[N\_RAY], dist\_blue[N\_RAY],$
  $dist\_yellow[N\_RAY]$
  for $i = 1 : N\_RAYS$ do
    $dist\_white[i] = dist\_blue[i] = dist\_yellow[i] = INFINITY$
    $search\_for\_white = true$
    $x = y = ray = 0$
    $lastColor = QUANT(x, y)$
    for $ray = 1 : MAX\_RAY$ do
      $x = ray * cos(\alpha_i)$, $y = ray * sin(\alpha_i)$
      $color = QUANT(x, y)$
      if $color$ isn't unknown or green then
        if $color$ is blue and $lastColor$ is green then
          $dist\_blue[i] = REAL\_DIST(x, y)$
          break
        else if $color$ is yellow and $lastColor$ is green then
          $dist\_yellow[i] = REAL\_DIST(x, y)$
          break
        else if $color$ is white then
          if $search\_for\_white$ then
            $search\_for\_white = false$
            if $lastColor$ is green then
              $dist\_white[i] = REAL\_DIST(x, y)$
            else
              $dist\_white[i] = FAKE\_RAY$
            end if
          end if
        else
          $dist\_blue[i] = dist\_yellow[i] = FAKE\_RAY$
          if $search\_for\_white$ then
            $dist\_white[i] = FAKE\_RAY$
          end if
          break
        end if
      end if
      $lastColor = color$
    end for
  end for
```

The algorithm to find the nearest chromatic transitions

[1] In RoboCup environment the ball is red, the lines are white, one goal is blue and the other is yellow, the robots are black, the robots' marker are cyan and magenta

[2] The three vectors are called "*scans*" in the reminder of the paper

of interest is presented in pseudo-code in Algorithm 1 (to simplify the comprehension only the scan in the inner section of the multi-mirror part omnidirectional mirror is presented).

The scan obtained from the image is compared with the scans extracted from the chromatic map of the environment, called *expected scans*. The map in Fig. 2(b) shows the chromatic characteristics of the environment. We use this map to compute the expected scan by ray-tracing, as we will explain in Section IV-B.

In summary, the advantages with respect to conventional range finders are: we have three scans for every pose of the robot (one for every chromatic transition of interest: green-white, green-blue and green-yellow), we immediately know which rays of the scan should be discarded because caused by occluding objects when a non-expected transition is detected (i.e. a chromatic transition that we are not looking for).

To manage the uncertainty in the scans' measurements, we slightly modified the classical Monte-Carlo Localization algorithm.

IV. Monte-Carlo Localization

Monte-Carlo localization (MCL) is a well-known probabilistic method, in which the current location of the robot is modelled as a posterior distribution conditioned on the sensors' data (Eq.1). The posterior probability distribution of the robot pose is called also the robot's **belief**. The belief about the robot's position is represented with a set of discrete points in the configuration space of the robot. These points are called **particles**. To update the belief over time, the particles are updated. Each particle is an hypothesis of the robot's pose, and it is weighted according to the posteriors. The belief about the robot's position is updated every time the robot makes a new measurement (i.e. it grabs a new image or a new odometry measure is available). It can be described by:

$$Bel(l_t) = \alpha p(o_t|l_t) \int p(l_t|l_{t-1}, a_{t-1}) Bel(l_{t-1}) dl_{t-1} \quad (1)$$

where $l_t = (x_t, y_t, \theta_t)$ is the robot pose at time t and a_t and o_t are respectively the sensor and the odometry readings at the time t. To calculate Eq. 1, the knowledge of two conditional densities, called *motion model* and *sensor model* is needed. The motion model expresses the probability the robot moved to a certain position given the odometry measures (*kinematics*), see Sec. IV-A. The sensor model describes the probability of having a certain sensor measurement in a certain pose, see Sec. IV-B. The motion model and the sensor model depend respectively on the particular robot platform and on the particular sensor. The localization algorithm is composed by 3 steps:

1) All particles are moved according to the motion model of the last kinematics measure;
2) The weights of the particles are determined according to the sensor model for the current sensor reading;
3) A re-sampling step is performed: high probability particles are replicated, low probability ones are discarded. The process repeats from the beginning.

The resampling step is performed with the Sampling Importance Resampling (SIR) algorithm [8] with the resampling technique of [11] The final estimation on the pose of the robot is obtained simply averaging the poses of all particles. For more details please refer to [3], [19].

A. Motion model

The motion model $p(l_t|l_{t-1}, a_{t-1})$ is a probabilistic representation of the robot kinematics, which describes a posterior density over possible successive robot poses. We implemented the MCL system on an holonomic robot, called Barney. The peculiarity of this robot is that it can move in any direction without the need of a previous rotation. A movement between two poses $l_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})$ and $l_t = (x_t, y_t, \theta_t)$ can thus be described with (α_u, T, θ_f), where α_u is the difference of heading between the two poses, T is the translation and θ_f is the motion direction. Updating the robot position according only to the kinematics does not take into account errors given by odometry inaccuracy and possible collisions of the robot with other obstacles. Therefore, a random noise term is added to the values given by the last odometry reading. Noise is modelled with Gaussian zero centered random variables $(\Delta_\alpha, \Delta_\mathrm{T}, \Delta_{rr}, \Delta_{r\mathrm{T}})$. They depend on both the amount of translation and of rotation. So, the motion model can be written as:

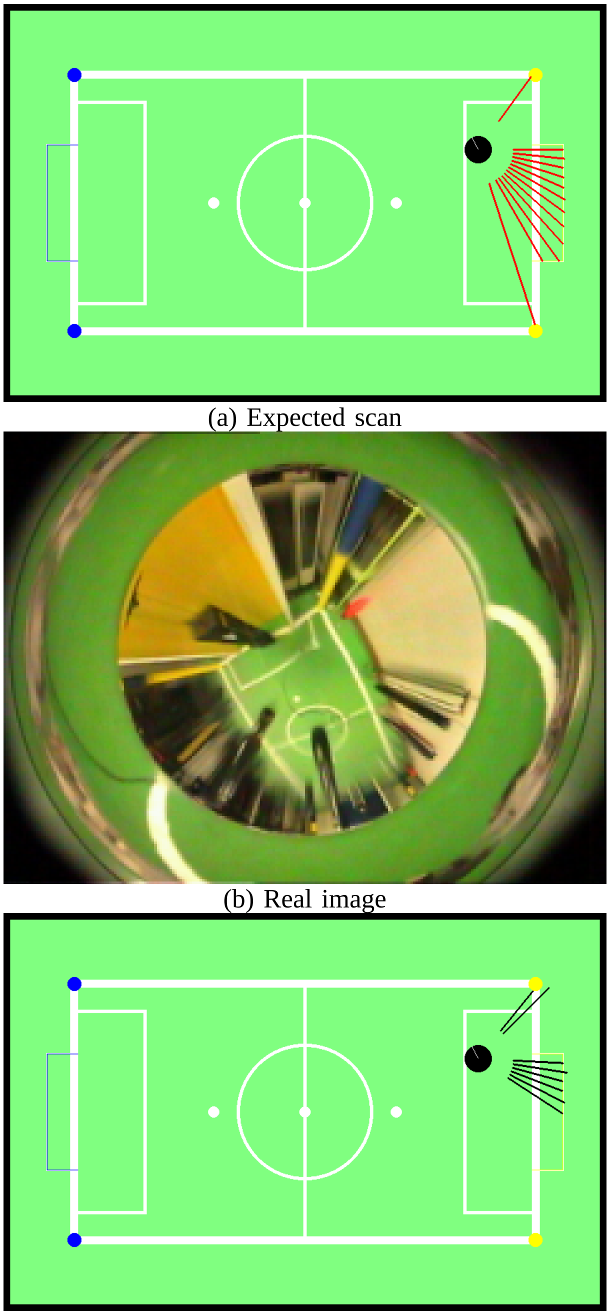
$$\begin{aligned} \alpha'_u &= \alpha_u + \Delta_\alpha(\alpha_u) \quad ; \\ T' &= T + \Delta_\mathrm{T}(T) \quad ; \\ \theta' &= \theta + \Delta_{rr}(\theta) + \Delta_{r\mathrm{T}}(T) \quad . \end{aligned}$$

For our holonomic platform, we found that good values for the standard deviations of the added noise contributions are $\sigma_\alpha = 30°/360°$, $\sigma_T = 200mm/m$, $\sigma_{rr} = 30°/360°$, $\sigma_{rT} = 30°/m$. We experimentally verified these values overestimate the actual errors and so provide good performances.

B. Sensor model

The sensor model $p(o_t|l_t)$ describes the likelihood to obtain a certain sensor reading given a robot pose. The sensor model is used to compute the weights of the particles. For each particle j, located in the pose l_t^j, the associated weight is proportional to $p(o_t|l_t^j)$ (i.e. to the likelihood of obtaining the sensor reading o_t when the robot has pose l_t^j). To calculate $p(o_t|l_t^j)$, we need to know the "*expected scan*" $o(l_t)$. The expected scan is the scan an ideal noise-free sensor would measure in that pose, if in the environment there were no obstacles, Fig. 3(a). Given l the robot pose, the expected scan $o(l)$ for one of the three chromatic transitions of interest is composed by a set of expected distances, one for each α_i, the rays of the scan (the black radial line in Fig. 1): $o(l) = \{g(l,i)|0 \leq i < N\_RAYS\}$. We can compute the expected distances $g(l,i)$ for an ideal noise-free sensor in a empty environment, using a ray tracing technique. The basic idea is: (i) to reproduce the pose of the robot in the metric maps of Fig. 2; (ii) to trace the rays exiting from the robot until they encounter the first chromatic transition of interest;

(iii) to store the length of these rays in the expected scan. The likelihood $p(o_t|l_t)$ can be calculated as $p(o_t|l_t) = p(o_t|o(l_t))$. In other words, the probability $p(o_t|o(l_t))$ models the noise in the scan by the expected scan [3], [19].


(a) Expected scan

(b) Real image

(c) Measured scan

Fig. 3. An example of expected and measured scans for the green-yellow transition. Given a pose, in (a) is represented the expected scan for an ideal noise-free sensor in a free environment. In (b) is shown the frame grabbed by the robot in that pose, in (c) is represented the corresponding measured scan. Only the rays with a right measurement are shown. Rays with $INFINITY$ values or with $FAKE\_RAY$ values are not displayed.

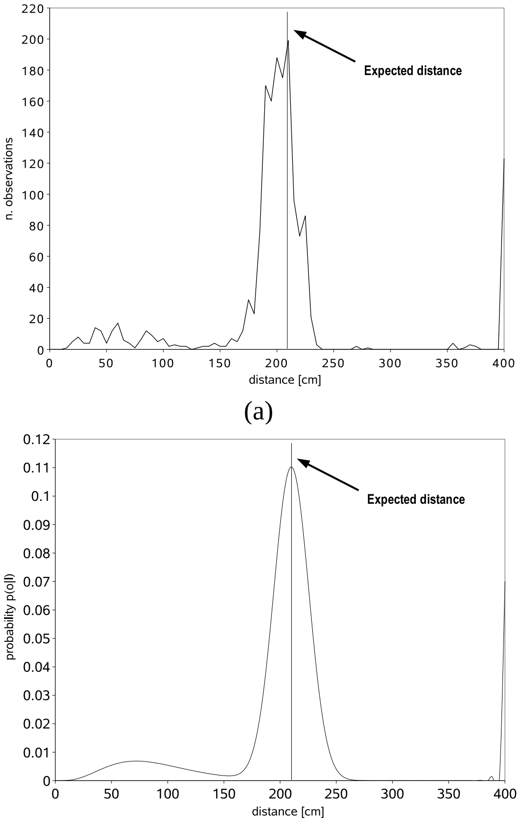
Fig. 3 compares the expected scan (top) and the real sensor scan (bottom). In the middle is the image grabbed by the robot. The scan is looking for the *green-yellow* chromatic transition of interest. So, the only rays with a right value in the distance vector are the rays painted in red in Fig. 3(a), i.e. the rays intercepting the yellow goal and the corner posts. Due to the noise in the image, it might happen that a color transition is not detected (like the rays striking the lower part of the yellow goal and the ray striking the lower corner post in Fig. 3(c)), or is detected at the wrong distance (like the fourth ray starting from the top in Fig. 3(c)), or it is falsely detected (like the second ray in Fig. 3(c)). It may happen also that a color transition is not detected because of occlusion (like in Fig. 3(b) were the goalkeeper is occluding part of the yellow goal), but we will discuss this in detail in Section V-B. So, we need to create a model of the sensor's noise.

1) Sensor noise: The probability $p(o|o(l))$ models the noise in the measured scan conditioned on the expected scan. For every frame grabbed by the sensor we obtain three scans (one for each chromatic transitions of interest), so we have to calculate three probability values. Since every scan is composed by a set of distances, one for each ray, we first model the probability that a single ray correctly detects the chromatic transition and then we combine the measurements of all rays. Eventually, we need to combine the three probability values given by the three chromatic transitions of interest. Let us describe these steps in more details.

The scan performed by the sensor is composed by a set of distances, one for each α_i: $o = \{o_i | 0 \leq i < N\_RAYS\}$. To compute $p(o_i|l)$ (i.e. the probability to obtain for a single ray a distance o_i given the pose l), we can consider directly the single expected distance $g(l,i)$, so we can write $p(o_i|l) = p(o_i|g(l,i))$. To create a statistical model of the distance measurement along the single ray of the scan, we collected a large number of omnidirectional images (about 2.000) in different known poses in the field of play. For every image we calculated the estimated distance of the chromatic transition of interest. The resulting measures are distributed along a different probability density one for each chromatic transition of interest. As an example, in Fig. 4(a) is plotted the probability density of the measured distance $p(o_i|l)$ for the green-white color transition. We described this density with the mixture of three probability density of Eq. 2. The three terms in Eq. 2 are respectively: an Erlang probability density, a Gaussian probability density and a discrete density. The numerical values of the parameters in Eq. 2 are calculated with a modified EM algorithm iteratively run on the 2000 images [4]. The resulting mixture, for the green-white transition, is plotted in Fig. 4(b). The Erlang variable models wrong readings in the scan caused by image noise and non-perfect color segmentation. The index n depends on the profile of the omnidirectional mirror used in the sensor. Our mirror (Sec. III) maps the area around the robot in the outer image ring where we have good accuracy and almost no noise, while in the inner part a certain amount of noise is present. We set the value of n, the Erlang variable, equal to the index of the first pixel scanned in the inner part of the image. So, the Erlang density will have a peak at the distance corresponding to the transition between the two mirror parts. The Gaussian density models the density around the maximum likelihood region (i.e. the region around the true value of the expected distance). The discrete density represents the probability to miss the detection of the chromatic transition, obtaining an *INFINITY* value in the scan vector, as described in Sec. III.

$$p(o_i|l) = \zeta_e(\frac{\beta^n o_i^{n-1} e^{-\beta o_i} 1(o_i)}{(n-1)!})+ \\ \zeta_g \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(o_i-g(l,\alpha_i))^2}{2\sigma^2}} + \zeta_d \delta(o_i - \infty) \quad (2)$$

$\zeta_e, \zeta_g, \zeta_d$ are the mixture coefficients, normalisation implies $\zeta_e + \zeta_g + \zeta_d = 1$. A different density mixture was computed for each one of the three chromatic transitions.



(a)

(b)

Fig. 4. In (a) the experimental distribution of measured distances for an expected known distance. The peak is at the expected distance. The measures before the expected one are due to the image noise. The last peak on the right of the plot means that due to image noise several time the chromatic transition has not been detected. In (b) the density $p(o|l)$ that represent our sensor model computed using EM-algorithm. The curve is the result of three contributions: (i) an Erlang variable with index n which depending on the geometry of the mirror, (ii)a Gaussian distribution centered at the expected distance and (iii) a discrete distribution representing the measurements resulting in the $INFINITY$ value.

Once the $p(o_i|l)$ is computed, it's possible to compute the probability of the whole scan given a pose l multiplying all the $p(o_i|l)$, Eq. 3.

$$p(o|l) = \prod_i p(o_i|l) = \prod_i p(o_i|g(l,i)) \quad (3)$$

2) Sensor occlusion: To cope with unexpected measures due to occlusion of the sensor by the moving objects in the environment (i.e. the other robots in the field or the ball), we filtered out all rays which distance o_i equal the *FAKE_RAY* value, see Sec. III (the $FAKE\_RAY$ value is represented by ϕ in Eq. 4). We called this process **ray discrimination**. The detection of occluding obstacles along the rays of a scan is very frequent in a densely crowded environment like the Middle-Size RoboCup field. In conventional range finders there is no ray discrimination system, so all measured distances contribute to the computation of $p(o|l)$. If a large number of distances are affected by the presence of other agents around the robot, the localization process might fail. Our ray discrimination technique enables to compute the sensor model only with a subset of reliable distances, obtaining a faster and more reliable localization.

$$p(o|l) = \prod_{\{i|o_i \neq \phi\}} p(o_i|l) = \prod_{\{i|o_i \neq \phi\}} p(o_i|g(l,i)) \quad (4)$$

From this equation follows that if the occlusion of the sensor increases, more and more rays will be discriminated and less information will be available for localization. Nevertheless, in our system all reliable information is exploited. We will see in Section V the ray discrimination technique enables to correctly localize the robot even in situations of severe occlusion.

C. Weights Calculation

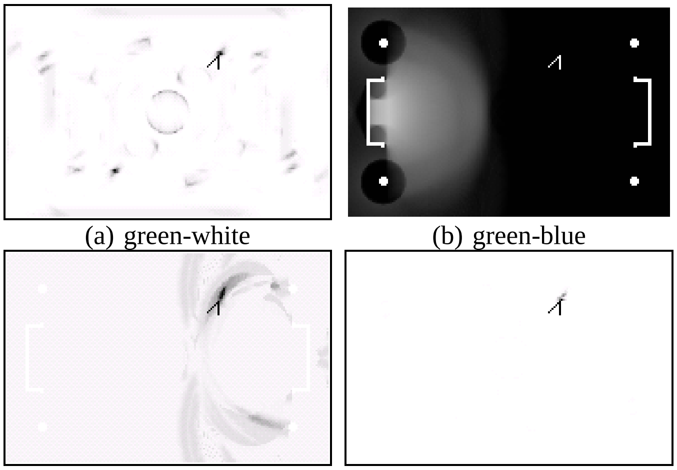
Returning to Monte-Carlo Localization, we can now compute, the weight $w^{(j)}$ associated to each particles j. We first calculate the quantity $\bar{w}^{(j)} = p(o|l_j)$ using Eq. 3), then all $\bar{w}^{(j)}$ are normalized such that $\sum_j \tilde{w}^{(j)} = 1$

$$\tilde{w} = \frac{\bar{w}^{(j)}}{\sum_j \bar{w}^{(j)}} \quad (5)$$

Since our system scans the acquired image for the three chromatic transitions of interest, this provides three scans for every frame, so three weight values are associated to every particles. To obtain a single weight value, we compute the product of the three weights (Eq. 6), and re-normalize all weights with Eq. 5 again.

$$w^{(j)} = \prod_{k=1}^{N} \tilde{w}_k^{(j)} \quad (6)$$

In Fig. 5, we give a pictorial visualization of the weights calculated by the three different scans of the three chromatic transitions of interest. The real pose of the robot is marked by the arrow. Higher weight values are depicted as darker points, lower weight values are depicted as lighter points. In Fig. 5 (a), are represented the weight contributions calculated by the scan looking for the green-white transition. One can notice that, due to the symmetry of the white lines in the field, two symmetric positions resulted to have high likelihood. In Fig. 5 (b), are depicted the weight contributions calculated by the scan looking for the green-blue transition. One can notice that all positions far away from the blue goal have a high likelihood, because no green-blue transition was found in the image scan. In Fig. 5 (c), are represented the weight contributions calculated by the scan looking for the green-yellow transition. One can notice there is an approximate symmetry around the yellow goal. All these contributions are


(a) green-white

(b) green-blue

(c) green-yellow

(d) overall

Fig. 5. Probability distributions $p(o_t|l_t)$ for all possible poses $l = (x, y, \theta)$ of the robot in the field given the scans of a single image. Darker points corresponds to high likelihood. The arrow represents the actual robot pose. In (a) is represented the probability given the scan for the green-white transition , in (b) for green-blue transition, in (c) for green-yellow transition, in (d) the three are combined.

combined with Eq. 6 to calculate the overall weights and depicted in Fig. 5 (d). Here, the weights with higher values are clustered only around the actual position of the robot.

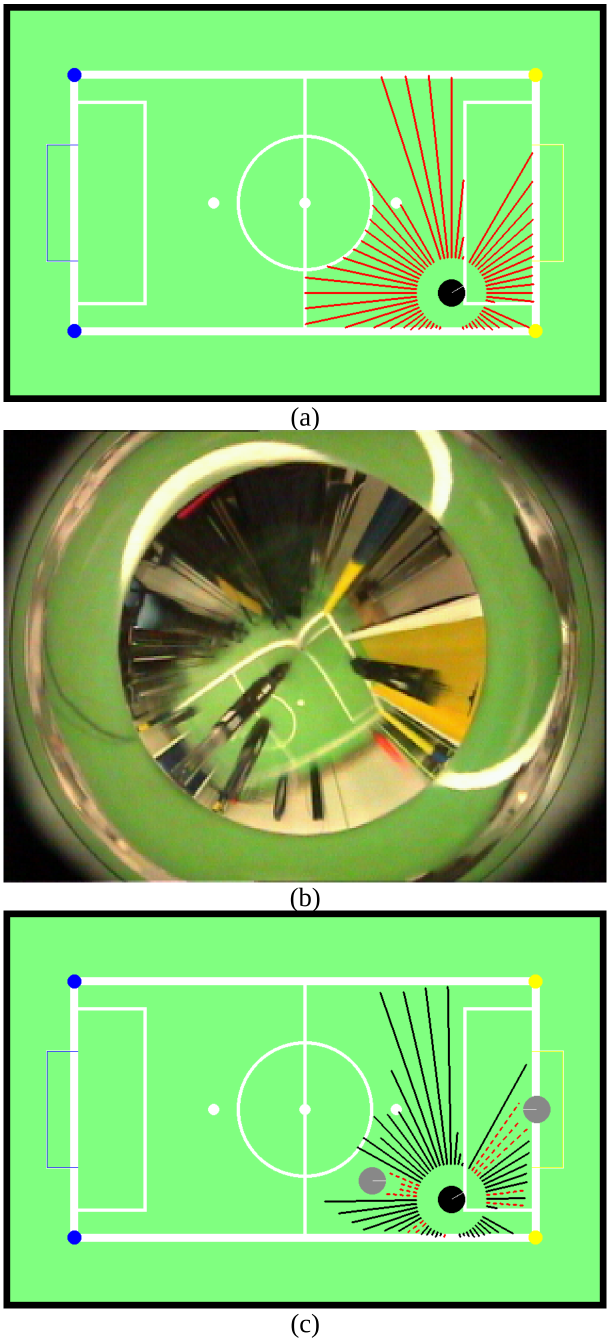
V. Experiments

The robot we used in the experiments is the holonomic custom-built platform, equipped with the omnidirectional sensor described in Sec. III. This paragraph is divided in three sections. In the first one, we evaluate the performance of the localization system depending on the number of particles used. In the second section, the robustness of the system to sensor occlusion is evaluated. In the third section, we present experiments in the corridors of our department to show the proposed system can be applied in any environment in which stable color transitions can be identified.

In order to improve the time performances of the system, the distances (see Sec. III) in the environment are divided in a grid of 5x5 cm cells, in a way similar to [7]. The expected distances for all poses and the probabilities $p(o_i|g(l, i))$ for all $g(l, i)$ can be pre-computed and stored in six look-up tables (two for each chromatic transition). Each look-up table takes about 13Mb. In this way the probability $p(o_i|l)$ can be quickly computed with two look-up operations, this enables our system to work in real-time at 10 Hz on a PC-104 Pentium III 700 MHz fitted with 128 Mb of RAM using 1000 particles.

A. Localization in the RoboCup field of play

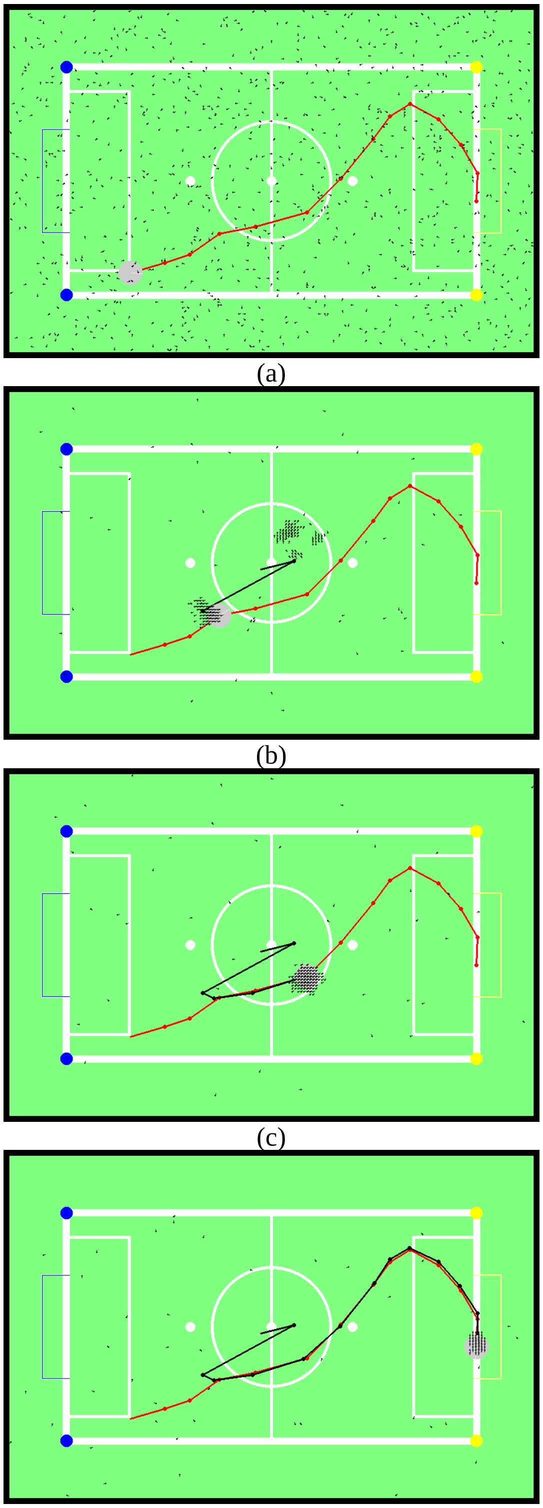
We tested the system on five different paths (an example path is shown Fig. 7). For each path we collected a sequence of omnidirectional images with the ground truth positions where those images were grabbed and with the odometry readings between two consecutive positions. In order to take into account the odometry errors, robot movements were performed by remote robot control. We tested our algorithms using different amounts of particles calculating the mean localization error for the three fundamental localization problems: (1) the global localization problem (the robot must be localized without any a priori knowledge on the actual position of the robot), (2) the


(a)

(b)

(c)

Fig. 6. An example of expected and measured scans for the green-white color transition. The robot's position is represented by the black circle. (a) represents the expected scan for an ideal noise-free sensor in a empty environment. (b) shows the frame grabbed by the robot in that pose. (c) represents the corresponding measured scan. In (c), the black line represents the measured distances while the dotted red line represents the rays in which a not expected transitions is detected (*FAKE_RAYS*). This can be caused by image noise or other robots (represented with grey circles). Look for example in the omnidirectional image (b) at the yellow goal: inside there is a robot (the goalkeeper) and three rays of the scan detect it (c) (along these rays is detected the black color that we are not searching for). For all these rays a *FAKE_RAYS* value is stored instead of the proper distance.



(a)

(b)

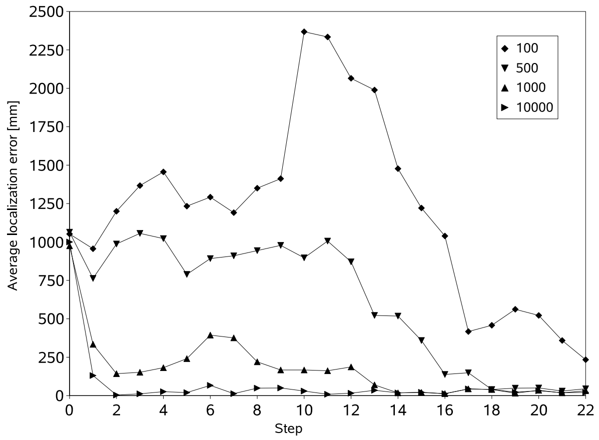
(c)

(d)

Fig. 7. A sequence of global localization and position tracking. The grey circle represents the actual robot pose, the red line represents the ground-truth path, the black line represents the estimated path of the robot, the black dots represent the particles (1000 particles are used).

position tracking problem (a well localized robot must maintain the localization) and (3) the kidnapped robot problem. In the kidnapped robot a well-localized robot is moved to a different pose without any odometry information: this problem can frequently occur in an high populated environment like RoboCup, where often robots push each other attempting to win the ball. To solve the kidnapped robot problem, we adopted the classical technique to reserve a certain percentage of the particles to this scope and to randomly scatter them in the environment to act as seeds for a re-localization process in case of localization failure [6].

One of the five test path is shown in Fig. 7. In Fig. 7(a), the particles are uniformly distributed (no knowledge is available on robot position). In Fig. 7(b), after moving 2 meters away and grabbing 4 images and getting 4 odometry readings, the particles are condensed around three possible poses. In Fig. 7(c), after 4 meters, 6 images and 6 odometry readings, uncertainty is solved and particles are condensed around the actual pose of the robot. In Fig. 7 (d) after 14 steps, one can see the position of the robot is well tracked along the test path (position tracking). The particles that are still dispersed in the environment are the particles scattered to solve the kidnapped robot problem.



(a)

Fig. 8. The average error in the global localization problem for a specific path with different amounts of particles.

The reactivity and the accuracy of the localization system increase with the number of particles, on the other hand also the computational load is increased. We tested the performances of the system with different numbers of particles. In Fig. 8 is shown the average localization error for global localization using respectively 100, 500, 1000, 10000 particles when the same path is repeated 100 times. A number of 1000 particles is compatible with real-time requirements and assures a robust and accurate localization. Also for the position tracking problem, 1000 is a good value for the number of particle. In Fig. 9 is shown the average and the maximum localization error in the position tracking phase for all test paths using different amount of particles. Already, with 1000 particles is possible to achieve good accuracy, acceptable average error (about 10 cm) and acceptable maximum error (about 30 cm) without burdening the CPU of the robot.

In Fig. 10 is shown the error for a kidnapped robot episode using 1000 particles and different rate of particles uniformly distributed in the environment. With a higher rate of particles

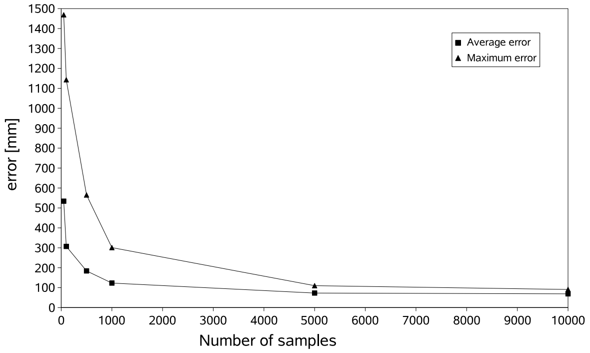


Fig. 9. The average error end maximum error in the position tracking problem over the five reference paths, calculated with different amount of particles, respectively 50, 100, 500, 1000, 5000, and 10000)

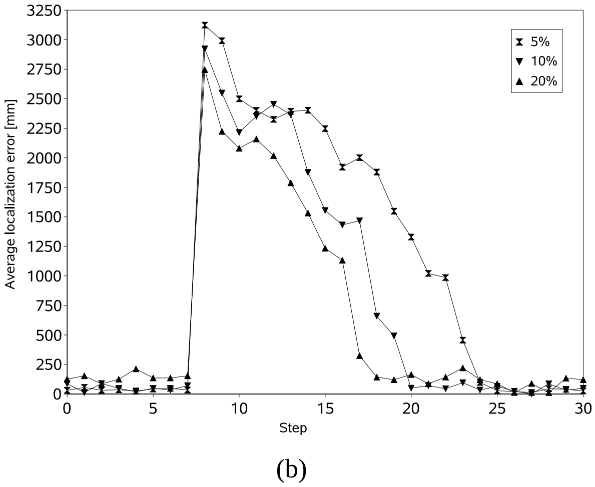


Fig. 10. The average error in the re-localization phase after a kidnap of the robot, varying the rate of uniformly distributed particles.

scattered in the environment the re-localization is faster (there is a higher likelihood that a particle is close to the actual position of the robot), but the average error is higher due to the lower number of particles clustered around the correct robot's pose. Notice that with 20% the re-localization is faster, but the average position tracking error is higher, because the number of the randomly distributed particles is so high that their contribution in the calculation of the center of gravity of the particles spoils the correct estimation of the robot pose. Therefore, we uniformly distributed in the environment only 10% of the 1000 particles. This ensures low contribution in the calculation of the center of gravity and acceptably fast recovery from the kidnap situation.

B. Robustness to sensor occlusion

In order to show the robustness of our approach in densely crowded environments, we tested the system on six different paths (like the one shown in Fig. 11). To understand how the occlusion of the omnidirectional camera affects the localization process consider the images on the left of Fig. 12. The plots on the right of Fig. 12 show the probability distributions of the robot's pose given by the image scans for increasing amounts of sensor occlusion, respectively 0%, 25%, and 50%. With increasing amount of occlusion the particles are more and more dispersed around the true position of the robot. Uncertainty increases, but most of the probability is still condensed around the right position. This is the result of the ray discrimination technique presented in Section IV-B.2. In Fig. 12, to obtain a measurable amount of occlusion, occlusion is obtained covering the sensor with black strips. Every strip cover 12.5% of the sensor and well simulate the presence of one robot close to the sensor. In real situations, like the one depicted in Fig. 3, is extremely hard to have more than two robot close to the sensor, while other robots are usually quite far and occlude a small fraction of the sensor.

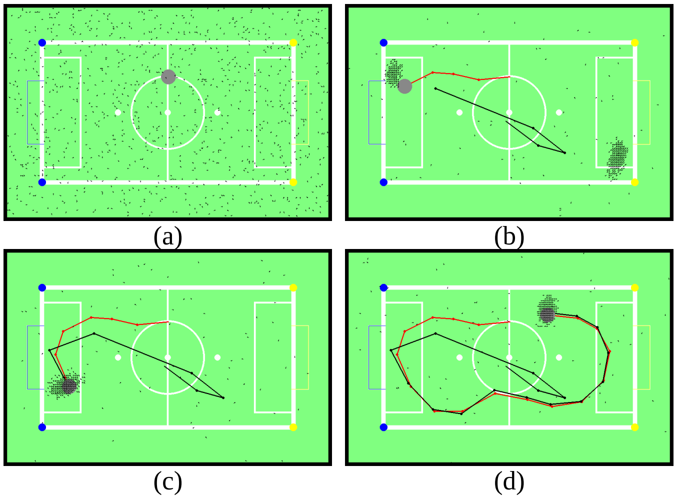


Fig. 11. A sequence of global localization in presence of 12.5% of sensor occlusion. Note that with respect to Fig. 7 the particles are more scattered around the true position of the robot.

For each path we collected five sequences of omnidirectional images respectively with 0%, 12.5%, 25%, 37.5%, and 50% occlusion. For every image we recorded the ground truth pose of the robot and the odometric readings between two consecutive positions. In order to take into account the odometric errors, robot movements were performed by remote control. We tested our algorithms for the three fundamental localization problems: global localization, in Fig. 11 (a) and (b); position tracking, in Fig. 11 (c) and (d); and kidnapped robot (not shown).

In Fig. 13 is shown the average error for a global localization experiment along the same reference path for three different amounts of sensor occlusion. Obviously without occlusion, localization is fast and accurate, but also in a densely crowded environment (sensor always covered for a rate of 50%) the robot is able to localize itself and to maintain localization with good accuracy. We obtained very good results also in the kidnapped robot problem. Recovery from a localization failure is obtained thanks to a small amount of particles (10% of the total number of particles) uniformly distributed in the environment. A few steps after a kidnapped robot episode most of the particles are again concentrated around the correct position and the situation is the same of global localization (due to lack of space we did not reported these experiment in

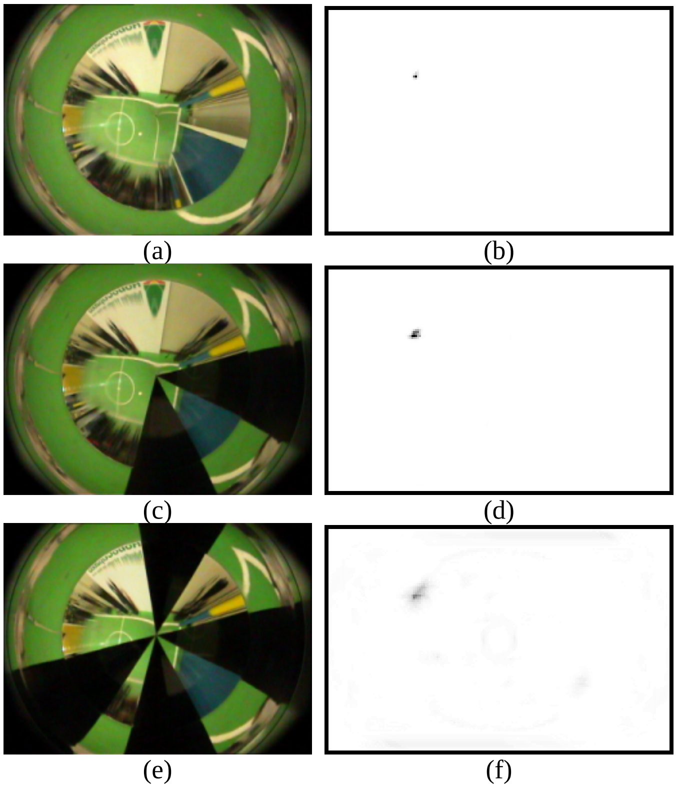

(a) (b) (c) (d) (e) (f)

Fig. 12. (On the left) The occlusion of the sensor is obtained with black stripes simulating the presence of other robots close to the sensor,. This was done in order to have a measurable amount of sensor's occlusion ((a) 0% of occlusion, (c) for 25% and (e) for 50%). (On the right) The probability distributions calculated for the corresponding amount of sensor's occlusion. Notice that in the situations of higher occlusion (d) (f), the particles are more scattered than in (b), but most of the probability is still condensed around the correct position.

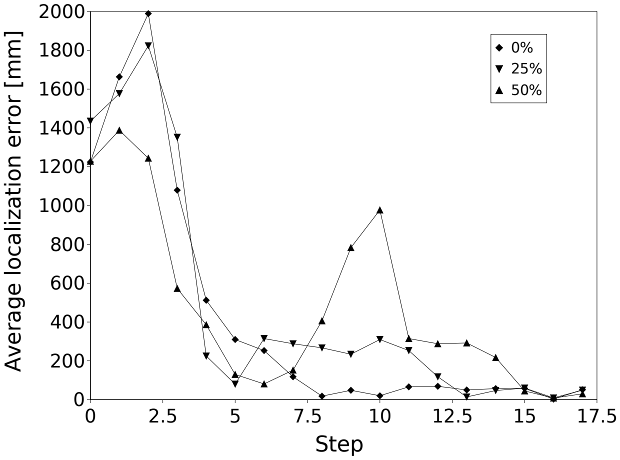


Fig. 13. The plots compares the global localization errors for a specific path with different amount of sensor's occlusion.

this paper).

Finally, we performed a statistical evaluation of our approach in the conventional situation of position tracking repeating 100 times all reference paths with different amount of occlusions. In Fig. 14 are reported the average error and the maximum error over all reference paths. Notice that both remain small also in a densely and constantly crowded environment.

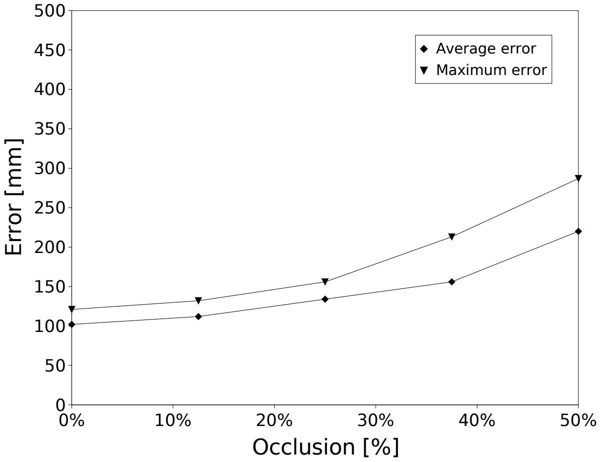


Fig. 14. Statistical evaluation of our system in the position tracking problem for all our reference paths. Accuracy (average error end maximum error) is represented for different amount of sensor's occlusion (0%, 12.5%, 25%, 37.5%, 50%).

C. *Localization in office environment*

Even if our system was developed for the RoboCup domain, from the beginning it was designed thinking to applications in a every-day environment. As a result, the localization system is not dependent from the peculiar chromatic transitions. In other words, it doesn't matter if the chromatic transitions of interest are three or a different number. It doesn't matter if they are green-white, green-blue, and green-yellow or transitions between different colors. As we said the only requirements are (i) some stable chromatic transitions can be identified in the environment and (ii) these chromatic transitions are reported in a metric map available to the system.

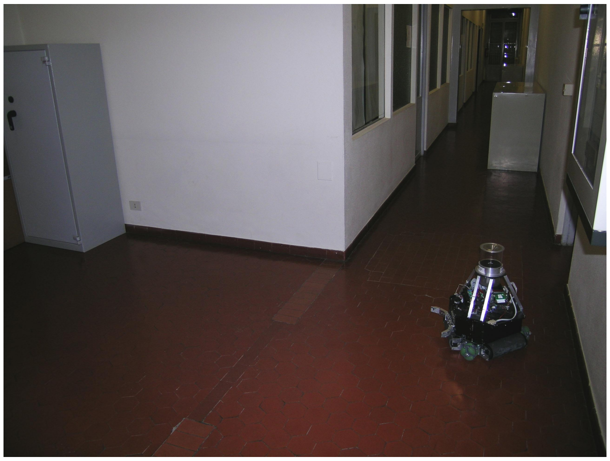


Fig. 15. The office-like environment in which our localization system was tested to prove its portability to real-world environments.

The environment in which we tested the generality of our system are the corridors of our department shown in Fig. 15.

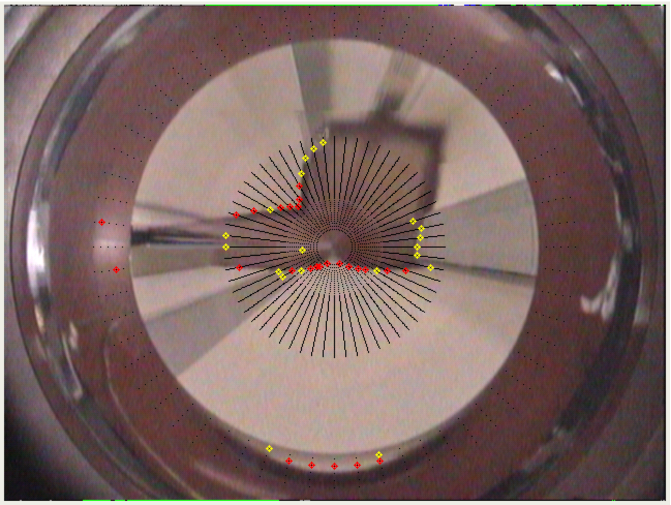


Fig. 16. The scanning algorithm at work on an image grabbed in a corridor of an office-like environment. The colored crosses highlight the color transitions of interest of the environment detected along the dotted radial lines in the omnidirectional image.

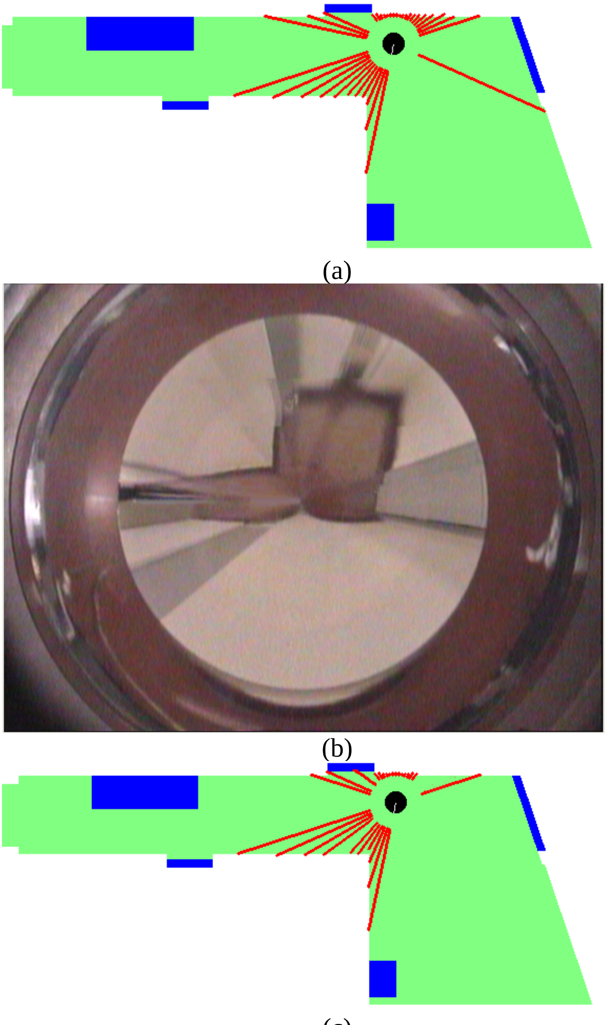
The floor of the corridor is of composed of red tiles, the walls are painted white, doors and furniture are painted grey. The corridor is 26 meters long and its width ranges from a minimum of 2 meters to a maximum of 4 meters. The trapezoidal room is about 4×5 meters. Along the corridors there are two grey lockers 2 meters wide.

This environment is much more challenging than the RoboCup environment due to uneven illumination and to the low contrast between existing colors. Nevertheless, even if sometimes the transition detection software misses or mistakes the type of chromatic transition, the system is abe to successfully locate the robot in the environment, as we will see.

A typical input image for the robot is the image shown in Fig. 16. The chromatic transition of interest we selected in this environment are *red-white* and *red-grey*. The red-white transitions in Fig. 16 are marked with a red cross and the red-grey transitions are marked with a yellow cross.

In Fig. 17 is shown a comparison between the expected and the real scan in the office-like environment for the red-white transition. In the color-metric maps, the red floor is represented in green, the grey objects are drawn in blue, and the rays of the scan are painted in red.

To test the robustness of the system in a general indoor environment without any lighting control, we performed the test in a dim day. Due to low ambient brightness, the noise in the image is high and the contrast between the white and the grey is low. In this situation, some chromatic transitions of interest are not detected or are erroneously detected. For instance, several transition are not detected in Fig. 17(c), especially in the top-right of the scan and the long ray at south-est of the scan. An example of wrong transition detection is the north-west ray in the scan in Fig. 17(c), that is erroneously detecting a red-white transition where there is a red-grey transition caused by the grey door. The probability distribution calculated from the red-white transition from the image of Fig. 17 is depicted in Fig. 18. In the picture, the darker regions have high likelihood to contain the robot, lighter regions have lower likelihood. One can see the probability distribution is quite sparse. Nevertheless combining also the information from the second chromatic transition of interest and combining the information coming from different measurements thanks to the MCL algorithm a robust localization can be achieved, as shown in Fig. 19 (the reliability of the system could be further improved if a more robust color transition detection algorithm is used (like the ones proposed in [10], [17]), but this is out of the scope of the present work). Starting without any knowledge on the robot position (Fig. 19(a)) after four-five steps most of the particle condense around the actual robot position (Fig. 19(d)). Again, we spread a 10% of particles randomly in the environment to solve the kidnapped robot problem.



(a)

(b)

(c)

Fig. 17. (Top) A close-up view of the color-metric map of the environment with the expected scan for the red-white transition. (Middle) The actual image grabbed by the robot. (Bottom) The real scan extracted from the image. Note that this contains several wrong detection due to the noise in the image.

Fig. 18. The probability distribution calculated from the red-white transition.

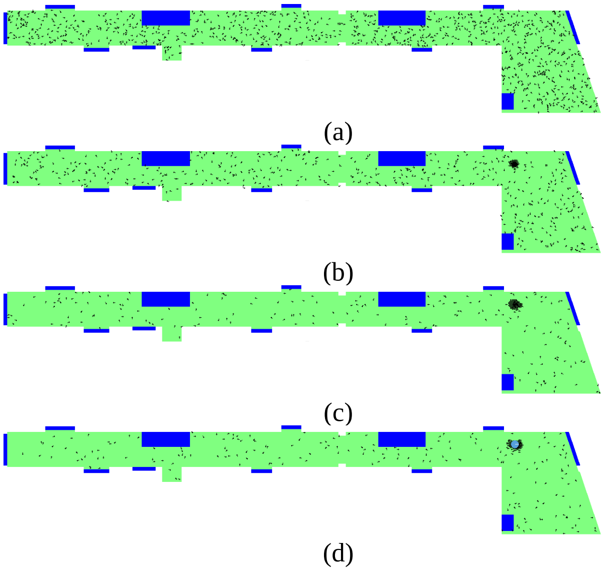


Fig. 19. An example of global localization in the office-like environment in our department. 10% of the particles are randomly distributed in the environment to recover in case of wrong localization (kidnapped robot).

VI. Conclusions

In this paper we propose a vision-based Monte-Carlo localization system particularly suitable for densely crowded environments thanks to the discussed ray discrimination technique. The omnidirectional vision sensor is used to emulate the behavior of range-finder devices and, thanks to the ability to distinguish different color transitions it can detect and reject wrong measurements caused by occlusions the sensor. We developed our system in the Middle-Size RoboCup domain, but we proved it can be used to localize the robot in any environments in which meaningful chromatic transition exists. The only requirement is that a map with the metric and chromatic characteristic of the environment is available. This means a map that depicts the static obstacles and the chromatic transitions of interest in the environment. This map can be as simple as a drawing stored in a image file, representing the plan of the environment augmented with the information on the color transitions. From these maps, the system will automatically recalculate all look-up tables used in the localization process. We presented successful experiments of global localization, position tracking and kidnapped robot both in the RoboCup environment and in a the corridors of our department. We experimentally showed the robustness of the localization system to sensor's occlusion and to chromatic transitions poorly contrasted.

References

[1] G. Adorni, S. Cagnoni, S. Enderle, G.K. Kraetzschmar, M. Mordonini, M. Plagge, M. Ritter, S. Sablatnög, and A. Zell. Vision-based localization for mobile robots. *Robotics and Autonomous Systems*, 36:103–119, 2001.

[2] Frank Dellaert, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, June 1999.

[3] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte Carlo Localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA99)*, May 1999.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[5] S. Enderle, M. Ritter, D. Fox, S. Sablatnög, G. Kraetzschmar, and G. Palm. Soccer-robot locatization using sporadic visual features. In E. Pagello, F. Groen, T. Arai, R. Dillman, and A. Stentz, editors, *Proceedings of the 6th International Conference on Intelligent Autonomous Systems (IAS-6)*. IOS Press, 2000.

[6] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. In *Proceedings of National Conference on Artificial intelligence (AAAI'99)*, pages 343–349, July 1999.

[7] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.

[8] N. Gordon, D. Salmond, and A. F. M. Smith. Novel approach to nonlinear and non-gaussian bayesian state estimation. *IEEE Proceedings-F*, 140:107–113, 1993.

[9] H.-M. Gross, A. Koenig, Ch. Schroeter, and H.-J. Boehme. Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS 2003)*, pages pp. 1505–1511, October 2003, Las Vegas USA.

[10] F. Hundelschausen, S. Behnke, and R. Rojas. An omnidirectional vision system that finds and tracks color edges and blobs. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup-2001: Robot Soccer World Cup V.*, L. N. on A. I. 2377, pages 374–379. Springer, 2002.

[11] G. Kitagawa. Monte Carlo filter and smoother for non-gaussian nonlinear state space models. *Journal Of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[12] Emanuele Menegatti, Francesco Nori, Enrico Pagello, Carlo Pellizzari, and Davide Spagnoli. Designing an omnidirectional vision system for a goalkeeper robot. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup-2001: Robot Soccer World Cup V.*, L. N. on A. I, pages 78–87. Springer, 2002.

[13] Emanuele Menegatti, Alberto Pretto, and Enrico Pagello. A new omnidirectional vision sensor for Monte-Carlo localisation. In *RoboCup Symposium 2004*, pages CD–ROM (13 pages), 2004.

[14] Emanuele Menegatti, Alberto Pretto, and Enrico Pagello. Testing omnidirectional vision-based Monte-Carlo localization under occlusion. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS04)*, pages pp. 2487–2494, September 2004.

[15] Emanuele Menegatti, Mauro Zoccarato, Enrico Pagello, and Hiroshi Ishiguro. Hierarchical image-based localisation for mobile robots with Monte-Carlo localisation. In *Proc. of European Conference on Mobile Robots (ECMR'03)*, pages 13–20, September 2003.

[16] Emanuele Menegatti, Mauro Zoccarato, Enrico Pagello, and Hiroshi Ishiguro. Image-based Monte-Carlo localisation with omnidirectional images. *Robotics and Autonomous Systems, Elsevier*, 48(1):17–30, August 2004.

[17] T. Röfer and M. Jüngel. Vision-based fast and reactive Monte-Carlo localization. In *Proc. of International Conference on Robotics and Automation (ICRA-2003)*, 2003.

[18] E. Schulenburg, T. Weigel, and A. Kleiner. Self-localization in dynamic environments based on laser and vision data. pages –, Las Vegas, NV, 2003.

[19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[20] T. Weigel, J.-S. Gutmann, M. Dietl, A. Kleiner, and B. Nebel. Cs freiburg: Coordinating robots for successful soccer playing. *IEEE Transactions on Robotics and Automation*, 18(5):685–699, 2002.

[21] T. Wilhelm, H.-J. Böhme, and H.-M. Gross. A multi-modal system for tracking and analyzing faces on a mobile robot. *Robotics and Autonomous Systems*, 48:pp. 31–40, August 2004.

[22] J. Wolf, W. Burgard, and H. Burkhardt. Using an image retrieval system for vision-based mobile robot localization. In *Proc. of the International Conference on Image and Video Retrieval (CIVR)*, 2002.