PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERÍA

ACTIVE VISUAL LOCALIZATION FOR MOBILE ROBOTS

JAVIER CORREA VILLANUEVA

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
ALVARO SOTO

Santiago de Chile, August 2007

© MMVIII, JAVIER CORREA

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERÍA

ACTIVE VISUAL LOCALIZATION FOR MOBILE ROBOTS

JAVIER CORREA VILLANUEVA

Members of the Committee:
ALVARO SOTO
MIGUEL TORRES
ANA MARÍA ARANEDA
MARCELO GUARINI

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, August 2007

To my family

ACKNOWLEDGEMENTS

I want to acknowledge my advisor, Álvaro Soto, for guiding me thru this experience. And my family for always been there for me.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

LIST OF FIGURES vi

LIST OF TABLES vii

ABSTRACT viii

RESUMEN ix

1. INTRODUCTION 1
   1.1. Introduction 1
   1.2. Previous work 3

2. THEORY AND DEVELOPMENT 6
   2.1. Particle filter 6
   2.2. Motion model 7
   2.3. Perception model 9
   2.4. Scoring of perceptual actions 9
   2.5. Implementation 11

3. SIMULATIONS AND TESTING 13
   3.1. Simulation 13
   3.2. Real robot 17

4. CONCLUSION AND FUTURE RESEARCH 23

REFERENCES 25

APPENDIX A. Information Theory 27

APPENDIX B. Equations 28

APPENDIX C. Implementation details 29

LIST OF FIGURES

1.1 Steps of the localization process. a) Right diagram, main steps of robot Localization using Bayesian techniques. b) Left diagram, proposed framework 2

2.1 Motion model. From initial position $s_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})$, the robot rotates an angle α_t, then moves a distance Δ_t, and finally rotates an angle ω_t to reach its final state $s_t = (x_t, y_t, \theta_t)$. 8

3.1 In the localization process with action selection on simple map a) Top-left graph, expected information of landmarks. b) Top-right graph, polar graph of expected information on the action space. c) Bottom graph, robot position estimate. 14

3.2 Step on the simulation. a) Left graph, localization without action selection. b) Right graph, localization using our strategy as action selection 15

3.3 Mean square error along a simulated path 15

3.4 Example of estimated positions using different approaches. Also the looking direction, shown as the FOV of our approach 17

3.5 Pictures of the robot on the testing environment 18

3.6 Position estimation of the robot movements 20

3.7 Variance of the real robot position estimation 21

LIST OF TABLES

3.1 Average MSE and deviation, measured in pixels, of 50 simulations. 17

3.2 Average waiting time in seconds 20

3.3 Calculations per seconds 22

ABSTRACT

Localization is a key issue a mobile robot must solve. Achieve a robust and efficient localization method is not a trivial task, as an increase in robustness usually leads to an impoverishment in efficiency and viceversa. One way to improve the localization process is the introduction of an active localization strategy. Several of these strategies have been developed, but in general these active localization strategies make use of the robot movement in order to achieve the desire behavior.

In this work, we propose a localization strategy using visual active sensors and information theory. Our strategy separates the movement action from the observation action. To achieve this separation, we need to introduce a new step, in which the robot is able to select the best possible observation action to perform. To carry out this selection, we make use of information gain, in other words, we search for the most informative observation actions to the localization process. Another important point is that we use the facts that our visual sensor detect landmarks, and the we use of a particle filter (a Monte Carlo method) for localizing the robot, to carry out an efficient selection of the observation action to perform.

After performed several test on both, simulated and real environments, we show that the introduction of our strategy of active localization, yields improvements on the localization process of the robot, getting better estimates of the its real position.

Keywords: mobile robot, robot localization, active localization, active perception.

RESUMEN

La localización es un problema clave para la robótica móvil. Lograr una localización eficiente y robusta no es una tarea fácil, dado que un incremento en la robustez, implica una caída en la eficiencia y viceversa. Una forma de mejorar el proceso de localización es mediante la introducción de una estrategia de localización activa, en donde varios de los trabajos desarrollado se basan en la utilización del mismo movimiento del robot para lograr el comportamiento deseado.

En este trabajo proponemos una estrategia de localización con sensores visuales activos. En nuestra propuesta separamos la acción de desplazamiento, de la acción de observación del robot. Para lograr esta separación, necesitamos la introducción de una etapa donde el robot sea capaz de seleccionar la mejor acción de observación posible, recurriendo a la ganancia de información para realizar la búsqueda de las posibles acciones. Así, la selección se plantea como la búsqueda de la acción más informativa para el proceso de localización. Otro punto importante es que aprovechamos el hecho que nuestro modelo de percepción visual se basa en la detección de hitos (*landmarks*), y a la utilización un filtro de partículas (un método de Monte Carlo) para llevar a cabo la localización del robot, para realizar una selección eficiente de la acción de observación a ejecutar.

Después de llevadas a cabo varias pruebas tanto en ambiente simulados como reales, vemos que la introducción de nuestra estrategia para lograr una localización activa, genera una mejora en el proceso de localización del robot, obteniendo estimaciones más acertadas sobre la posición real del éste.

Palabras Claves: robots móbiles, localization de robots, localización activa, percepción activa.

1. INTRODUCTION

1.1. Introduction

The problem of obtaining an accurate estimation of the position of a mobile robot within its environment, task known as localization, is one of the key issues a mobile robot must solve. Achieving a robust and efficient localization method is not a trivial task, as an increase in robustness usually leads to an impoverishment in efficiency and viceversa. This stresses the need to focus the perceptual resources on the most informative parts of the environment.

At present, the state of the art solutions to indoor robot localization are mainly based on Bayesian techniques, such as the Extended Kalman Filter (EKF) or the Particle Filter (PF) (Thrun, Burgard, & Fox, 2006). These techniques have three main basic steps: i) A prediction step, where the robot uses its last position and self-motion information to predict its new position, ii) An observation step, where the robot senses its environment, and finally, iii) A position refinement step, where the robot updates the estimation of its new position using the new observations from the environment. Figure 1.1.a shows a schematic view of these 3 steps.

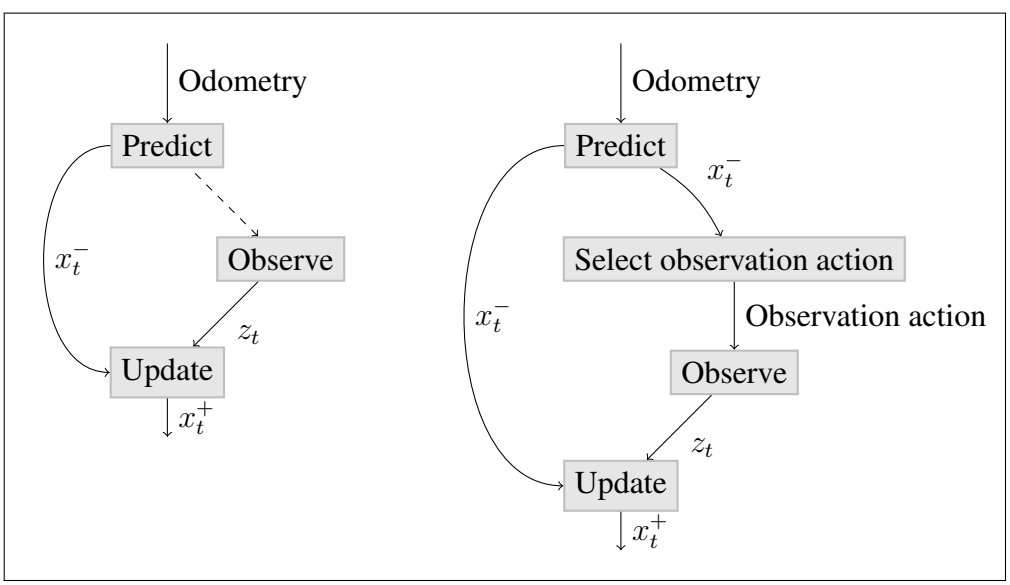

FIGURE 1.1. Steps of the localization process. a) Right diagram, main steps of robot Localization using Bayesian techniques. b) Left diagram, proposed framework

A particular feature of the previous Bayesian scheme is that the observation step does not consider the possibility of actively controlling the sensors in order to direct them to the most informative parts of the environment. In effect, most current localization techniques consider fix sensors usually mounted on the robot. These sensors provide observations about the part of the environment that is local to the current robot position. In this work, we use concepts from information theory and planning under uncertainty to develop a sound strategy for actively controlling the robot sensors.

As a test bed, we use a differential drive wheeled robot, provided with an odometer on each wheel and a color camera mounted on a pan-tilt system. By using the odometer, the robot is able to track its motions. By using the video camera and suitable computer vision algorithms, the robot is able to distinguish a set of visual landmarks that represent the map of the environment. As our focus is the localization problem, we assume that the map of the environment is known in advance, see (Thrun et al., 2006) for a description of relevant mapping techniques.

The novelty of our localization approach resides in actively using the pan-tilt mechanism to direct the visual sensor to the parts of the environment where the robot expects to obtain the most useful information. We understand the most useful information, as the one that helps the robot to achieve the best localization. The idea is that by selecting the most suitable perceptual action, the robot can see the most informative landmarks in its local surrounding, being able to achieve a better estimation of its current location.

We accomplish this by adding a new step between the prediction and observation steps of Figure 1.1.a. This new step selects the best observation-action in terms of the expected value of the information the robot anticipates to receive from the environment. To estimate this best perceptual action the robot uses the prediction of its current position and the map of the environment. Figure 1.1.b shows the new scheme.

We propose an approach where the robot assigns a score to each possible perceptual action based on how much information we expect they provide to the robot localization process. We use these scores to perform an action selection step, where the robot searches through its

set of possible perceptual actions looking for the best action to perform. This allows the robot to bias its perception and maximize the expected information of the observation step. Our experiment indicates that using this approach; we are able to substantially improve the localization estimation of our robot, obtaining more accurate and less uncertain estimation.

The previous idea is closely related to recent applications of computational models of visual attentional mechanisms than combine top-down information with bottom-up data (Navalpakkam & Itti, 2006) (Espinace, Langdon, & Soto, 2008). In our case, the map of the environment and the current estimation of the robot position give the top-down information. The bottom-up information is given by the images acquired by the video camera according to its current field of view. The idea is to use the top-down information to bias the perception of the robot. This generates adaptive behaviors in the perceptual system of the robot that increase the efficiency and accuracy of the robot position estimation.

This document is organized as follows. Section 1.2 describes related previous work. In Chapter 2 presents background theory that we later use in Section 2.4, which presents the proposed method and relevant implementation details. Chapter 3 presents our empirical results using simulations and a real case. Finally, Chapter 4 presents the main conclusions of this work and future avenues of research.

1.2. Previous work

In this section we review some of the work done on using information theory or entropy on action selection for different robot task. We also review work done on visual attentional systems that work as foundation to our work. The use of entropy, or information gain, as criteria to select actions by a robot has already been used in several works with different problems. Also biasing the perception of visual information has been done in some very interesting works on perception.

In (Mitsunaga & Asada, 2006), they develop a framework to chose the best robot action based directly on perception and on information criteria, without requiring first localize the robot. They manually train the robot, so it can later select the best action to perform based

on the expected information of actions, how useful the robot expect the action to be. In (Stachniss, Grisetti, & Burgard, 2005) they use the information gain criteria to create an exploration plan for the SLAM process that balance exploring new areas and improving the estimation on the map and the robot localization. The implementation is done using a Rao-Blackwellized particle filter. In our work we use a similar approach to estimate the values of information gain of actions to be performed by the robot, taking in consideration that we are on a localization context.

One of the first works related to active localization was (Roy & Thrun, 1999), where they implement a partially observable Markov decision process (POMDP) to find a route between two points in a map. Finding a solution to a POMDPs is in *PSPACE-complete*, therefore to have a real time system, first they augmented the state representation adding explicitly the uncertainty, as entropy of the system, and use some pre-calculated information to speed-up the process. The behavior that was generated with the use of the POMDP was a coastal navigation: the robot first moves near a wall or objects it could use to localize itself and then it moves along the walls to reach the target position, minimizing the probability of getting lost. In (Burgard, Fox, & Thrun, 1997), they use entropy minimization to create an active localization strategy. This active localization strategy has two parts: where to move and where to look. The first part allows the robot to move to different locations where, by reaching these locations, the robot would improve its localization. The second part select from the array of sonars, which one use to sense the environment. They define general strategy for minimizing the entropy using occupancy grids as representation of the robots current position. But the main focus is on active navigation instead of active perception. The biggest different of the previous mentioned works with our approach, is that we focus solely on active perception; we assume that the navigation is controlled by an external system, that guides the robot to the completion of a goal.

Another similar work is (Davison & Murray, 2002), where they use Extended Kalman Filter SLAM (EKF-SLAM) for calculating the robots position estimation, and define a way to search where the robot must fixate its camera to have better readings of its environment. To perform this active perception, they manually created a policy on how to look for landmarks:

they look for landmark that is the hardest to predict. Because they already have defined what action to do, they need to rate landmarks using the information provided by the Kalman Filter. They do not create a navigation plan; they assume that an external goal-directed module controls the navigation. Asides from the fact that they attack the problem of active visual perception over SLAM and no localization, the main difference is that model the active perception as a landmark rating an action search, and we do it as an action search.

2. THEORY AND DEVELOPMENT

In this section we provide the main details of our approach. First, we briefly describe the main features of the particle filter, the technique that we use to keep a probabilistic estimation of the robot position. Then, we describe the main details of the motion and perception models used by our implementation of the particle filter. Afterwards, we describe our criterion to search the space of possible perceptual actions. Finally, we show how we implement this criterion using a particle filter.

2.1. Particle filter

The particle filter is a useful non-parametric technique to perform sequential state estimation via Bayesian inference. It provides great efficiency and extreme flexibility to approximate any functional non-linearity. The key idea is to use samples, also called particles, to represent the posterior distribution of the state given a sequence of sensor measurements. As new information arrives, these particles are constantly re-allocated to update the estimation of the state of the system.

In Bayesian terms, the posterior distribution of the state can be expressed by:

$$p(s_t|o_{1:t}) = \beta\, p(o_t|s_t)\, p(s_t|o_{1:t-1}) \tag{2.1}$$

Where β is a normalization factor; s_t represents the state of the system at time t; and $o_{1:t}$ represents all the information collected until time t. Equation (2.1) assumes that s_t totally explains the current observation o_t.

The particle filter provides an estimation of the posterior in Equation (2.1) in 3 main steps: sampling, weighting, and re-sampling. The sampling step consists of taking samples (particles) from the so-called dynamic prior $p(s_t|o_{1:t-1})$. Next, in the weighting step, the resulting particles are weighted by the likelihood term $p(o_t|s_t)$. Finally, a re-sampling step is usually applied to avoid the degeneracy of the particle set. The key point that explains the efficiency of the filter comes from using a Markovian assumption and expressing the dynamic

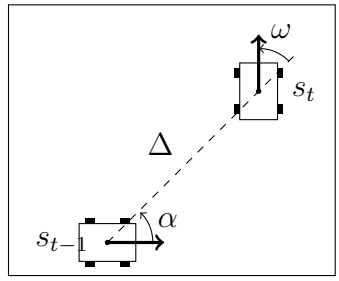

FIGURE 2.1. Motion model. From initial position $s_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})$, the robot rotates an angle α_t, then moves a distance Δ_t, and finally rotates an angle ω_t to reach its final state $s_t = (x_t, y_t, \theta_t)$.

prior by:

$$p(s_t|o_{1:t-1}) = \int p(s_t|s_{t-1})\, p(s_{t-1}|o_{1:t-1}) ds_{t-1}. \tag{2.2}$$

This provides a recursive implementation that allows the filter to use the last estimation $p(s_{t-1}|o_{1:t-1})$ to select the particles s^i_{t-1} for the next iteration. These particles are then propagated by the dynamics of the process $p(s_t|s^i_{t-1})$ to obtain the new set of particles that estimate the state at time t. In this way, at all time the state of the system is estimated by the set of weighted samples:

$$S_t = \{\langle s^{(i)}_t, \omega^{(i)}_t \rangle | i = 1, \ldots, n\}, \tag{2.3}$$

Where each weight $\omega^{(i)}_t$ is a factor called importance weight, and altogether they sum up to one.

In the area of robot localization, the particle filter was first used in (Dellaert, Fox, Burgard, & Thrun, 1999). Since then, it has become one of the favorite tools for robot localization and also for the related problem of constructing maps of the environment (Thrun et al., 2006). According to the generic scheme of Figure 1.1, the key steps of the particle filter are the prediction step, given by $p(s_t|s^i_{t-1})$, and the observation step, given by $p(o_t|s_t)$. In terms of robot localization, these steps are given by the so-called motion and perception models, respectively. We discuss these probabilistic models next.

2.2. Motion model

As usual in mobile robotics, we assume a point model for the robot. Also, to simplify our analysis, we consider only the case of planar motions, although, the extension to the 3D case is straightforward. Accordingly, the position of the robot at time t is given by the state vector: $s_t = (x_t, y_t, \theta_t)$. Following (Araneda, Fienberg, & Soto, 2007), our motion model assumes that to move from s_{t-1} to s_t, the robot performs three independent actions, as represented in Figure 2.1. First, the robot rotates an angle α_t to face the direction of the translation. Afterwards, it translates a distance Δ_t to reach (x_t, y_t). And finally, the robot rotates again, an angle ω_t, in order to face its final orientation θ_t.

The set of actions that moves the robot from s_{t-1} to s_t are $\delta_t = (\alpha_t, \Delta_t, \omega_t)$, and are estimated from the robot's odometers, in our case using the kinematics of a differential drive robot. Given the values of the movement action δ_t, our motion model is given by:

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} + \Delta_t \cos(\theta_{t-1} + \alpha_t) \\ y_{t-1} + \Delta_t \sin(\theta_{t-1} + \alpha_t) \\ \theta_{t-1} + \alpha_t + \omega_t \end{pmatrix} \tag{2.4}$$

In the implementation we add some white noise to the movement values. Let $\hat{\delta}_t = (\hat{\alpha}_t, \hat{\Delta}_t, \hat{\omega}_t)$ be the noisy values of the movement δ_t, we assume that the uncertainty in this values corresponds to Gaussian models with zero mean and known variance, as given by:

$$\begin{aligned} \hat{\alpha}_t &\sim N(\alpha_t; \phi_\alpha^2) \\ \hat{\Delta}_t &\sim N(\Delta_t; \phi_\Delta^2) \\ \hat{\omega}_t &\sim N(\omega_t; \phi_\omega^2) \end{aligned} \tag{2.5}$$

2.3. Perception model

Our perception model is based on the detection of visual landmarks that are modeled as 2D points. For each landmark, our visual detection system can sense range and bearing with respect to the robot's local frame of coordinates. In this planar context, we limit our analysis to control only the panning motion of the robot camera. The extension to the 3D case, considering an azimuth angle, is straightforward, although, as we discuss below, there is an increase in the computational complexity.

If we denote, respectively, as r_t^l and β_t^l, the range and bearing measurements that we expect to read from our sensor for a given landmark l a time t, and we denote by (x_t, y_t, θ_t) the robot position, the perception model can be expressed by:

$$\begin{pmatrix} r_t^l \\ \beta_t^l \end{pmatrix} = \begin{pmatrix} \sqrt{(x_t - l_x)^2 - (y_t - l_y)^2} \\ \tan^{-1}(l_y - y_t, l_x - x_t) - \rho_t - \theta_t \end{pmatrix} \tag{2.6}$$

Where (l_x, l_y) is the position of landmark l and ρ_t is the panning angle of the camera.

Similar as the motion model, we use a Gaussian models to quantify the uncertainty in the sensor measurements. Let $o_t^l = (\hat{r}_t^l, \hat{\beta}_t^l)$ be the noise values detected of the landmark l from the camera, then:

$$\begin{aligned} \hat{r}_t^l &\sim N(r_t^l; \phi_r^2) \\ \hat{\beta}_t^l &\sim N(\beta_t^l; \phi_\beta^2) \end{aligned} \tag{2.7}$$

2.4. Scoring of perceptual actions

To evaluate the expected benefit of each possible perceptual action, we use the concept of information gain. We use mutual information, $I(X;Y)$, that measure how much information of random variable X we get by knowing the random variable Y, and entropy, $H(X)$, that

measure the uncertainty around random variable X. For more details on definitions and properties see Appendix A.

Our intuition is to use a score that allow us to evaluate the level of uncertainty between the estimation of the robot position with and without considering the information provided by the target perceptual action. We can express this intuition in terms of information gain by:

$$\Theta(o_t, a) = I(s_t|o_{1:t-1}, m; s_t|o_{1:t-1}, m, o_t, a) \tag{2.8}$$

Which can also be written as

$$\Theta(o_t, a) = H(s_t|o_{1:t-1}, m) - H(s_t|o_{1:t-1}, m, o_t, a) \tag{2.9}$$

Where s_t is the estimation of robot position, o_t is the observation at time t (what landmark is seen at time t), a is the observation action, and m is the map. From now on, we will omit m and $o_{1:t-1}$ in the equations because they are constant to all terms.

In Equation (2.9), only the right term of the right hand side depends on the perceptual action. As we show in Appendix B by using the definition of entropy and Bayes theorem, we can express this term as a function of $p(s_t|a)$ and $p(o_t|s_t, a)$.

In terms of $p(s_t|a)$, each possible perceptual action a does not affect the knowledge that we have about the position s_t, therefore, $p(s_t|a) = p(s_t)$. In terms of $p(o_t|s_t, a)$, we know the map and the field of view (FOV) of the robot's camera, therefore, at each position s_t we know deterministically the landmarks that are visible. To estimate $p(o_t|s_t, a)$, we consider a constant probability $P_z > 0$ of sensing each of the visible landmarks and a null probability of sensing landmarks that are out of the FOV. We also assume that landmarks are observed independently. In this way:

$$\Theta(o_t, a) = H(s_t) - \log p(o_t|a) + \frac{P_z}{p(o_t|a)} \int_{FOV} p(s_t) \log P_z p(s_t) ds_t \tag{2.10}$$

Where:

$$p(o_t|a) = \int p(o_t|a, s_t) p(s_t|a) ds_t = P_z \int_{FOV} p(s_t) ds_t \tag{2.11}$$

Now that we have an approximation to the value of (2.9), we have to search for the optimum action to perform. We achieve this by searching over the space of possible actions, the action that provides the highest expected Θ. As a further refinement, we also consider in our model the cost associated with executing different perceptual actions. This cost can be related to the difficulty to perform the action or the time needed to complete it. Considering this type of costs, our function to select perceptual actions is given by :

$$\begin{aligned} a &= \arg\max_a \{E[\Theta(o_t, a)]_{o_t|a} - \alpha \cdot cost(a)\} \\ &= \arg\max_a \sum_i p(o_t = i|a)\Theta(o_t, a) - \alpha \cdot cost(a) \end{aligned}$$

Where α is a weighting factor that trades-off the expected Θ and cost of each possible perceptual action. This factor also takes care of differences in the units, nats or bits for Θ and seconds for the cost.

In our case, we use a pan-tilt unit to execute perceptual actions. We defined our cost function proportional to the angle that the pan-tilt has to move to reach the target position. Empirically, we set the weighting factor $\alpha = 0.1$, as a good compromise between the degree of panning and the time to wait for the action to be completed.

2.5. Implementation

Given that there is not a close solution to Equation 2.9, we use a Monte Carlo approximation, and employ the samples provided by the particle filter. First, we approximate the term $H(s_t)$(as in (Stachniss et al., 2005)) by

$$H(s_t) \approx -\sum_k \omega_k \log \omega_k \tag{2.12}$$

Similarly, we approximate the entropy $H(s_t|o_t, a)$ of equation 2.9, as follow:

$$H(s_t|o_t, a) \approx \log p(o_t|a) - \frac{P_z}{p(o_t|a)} \sum_j \omega_j \log P_z \omega_j \tag{2.13}$$

with $p(o_t|a) = P_z \sum_j \omega_j$, and the j index, in this equation as well in equation 2.13, goes over all particles, that can detect the seen landmark. One of the advantages of our approach is that it is easy to perform the approximation, it is done in a single iteration over the set of particles, performing sums and log-sums of particles weight. More details on the implementation can be seen on Appendix C.

In our implementation, the estimation of equation 2.9, is performed using all the samples provided by the particle filter. This approach yield good results, but at the cost of a decrease on the complete process performance. In section 3.2 we show that the total overhead introduced by having the ability to select landmarks, in the real robot, is mainly produced by waiting the observation action to conclude, in our case, wait the pan-tilt unit to reach the desired pan angle. If the main reason for lost of performance was produced by the calculations made, we could increase the performance not using all the available particles, and sampling a small fixed number of particles from the total pool, making the estimation less exact but reducing the time needed to perform the calculations.

From the complexity point of view, the computation needed to estimate equation 2.8 using the O notation, with a naive implementation, is $O(|A|ln)$, with n being the number of particles used in the estimation, l the number of landmarks in the map and $|A|$ the cardinality of the actions space (or action space partition as in our case). The complexity shown here comes form a very naive implementation, we use all available particles to perform the estimation of equation 2.8.

3. SIMULATIONS AND TESTING

In this section we show tests results of the proposed strategy on two different environments. First a simulated robot using MATLAB® that uses our approach to select where to look. Later, using CARMEN (Montemerlo, Roy, & Thrun, 2003) we implemented our approach to select where to point a camera on a real robot. Expected results are to get an improvement on the robot position estimation while maintaining a rather low overhead.

3.1. Simulation

In order to test the performance of the proposed strategy, we simulated a robot using MATLAB, which uses our approach to select where to point a simple simulated sensor. This sensor always detects landmarks wherever they are placed in the robot FOV. The outputs of the sensor are: detected landmark with its range and bearing both with some white noise (as the model presented in equation 2.5). The simulated robot provides us with movement information composed by: initial rotation, traveled distance and final rotation, all values with white noise (as the model presented in equation 2.7). The values of used deviation are 0.17 for the rotations and 0.4 for the traveled distance.

To compare the effect of introducing a perceptual action selection to the localization process, we carry out simulations of the robot with different observation action selection strategies. First a static sensor always looking in the same direction (in our case looking straight). Then an active sensor that looks at random places (uniform angles over the circle), and finally selecting the optimum angle given by our strategy.

A simulation consists on taking a map and a path within the map for the robot to follow, then odometry and perception data are generated. To simulate odometry we took the relative movement between two consecutive points on the real path, adding white noise to obtained values. In order to simulate a perception measure, at every position of the robot, we search for detected landmarks, then we calculate range and bearing of each one of them using: the known map, robot position and observation action selected by the robot (as equation 2.6). In a similar manner as we simulate odometry, we add withe noise to detection measurement.

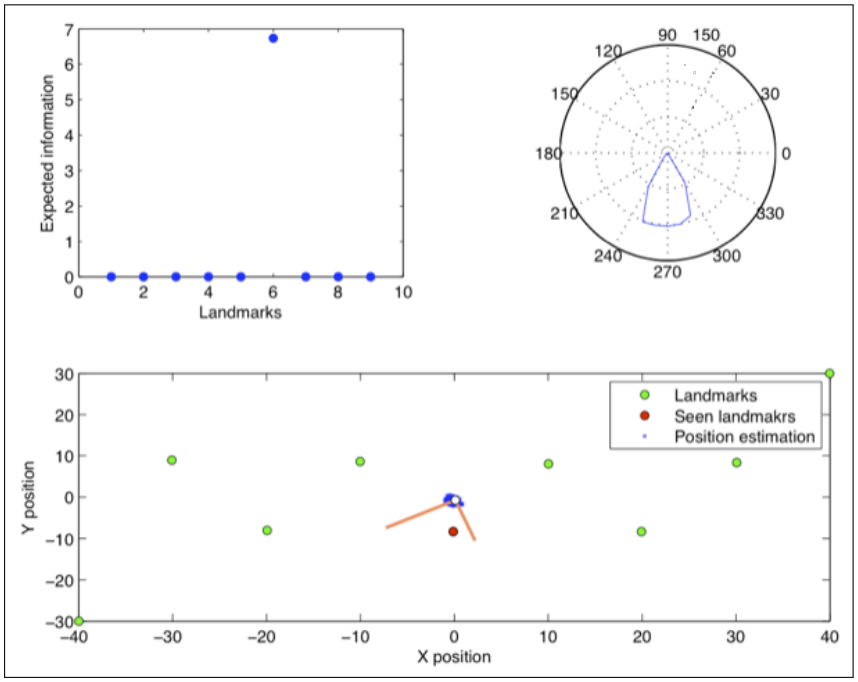

FIGURE 3.1. In the localization process with action selection on simple map a) Top-left graph, expected information of landmarks. b) Top-right graph, polar graph of expected information on the action space. c) Bottom graph, robot position estimate.

Figure 3.1 shows a complete status of the simulated system on a simple environment. In this step, the robot is located next to the fourth landmark looking downwards. This is due that the maximum information gain is reached with an observation action of rotating approximately $-90°$ (which gives a real angle of $270°$ approximately). Within a normal localization process, the robot does not have the ability to detect that particular landmark, the robot pose and the sensor configuration make that particular landmark fall out the robot FOV.

Figure 3.2 shows another example. In this case it is possible to compare the difference in the robot simulation status with and without the ability to select an observation action. In the figures the blue clouds of points represent the estimation that the particle filter has on the robot position. Looking at the different estimates of the robot position, we see that in figure 3.2.b the cloud is smaller and it encloses better the real robot position than the estimate at figure 3.2.a. By comparing the graphs we can see the improvement introduced by having the ability of active sensing.

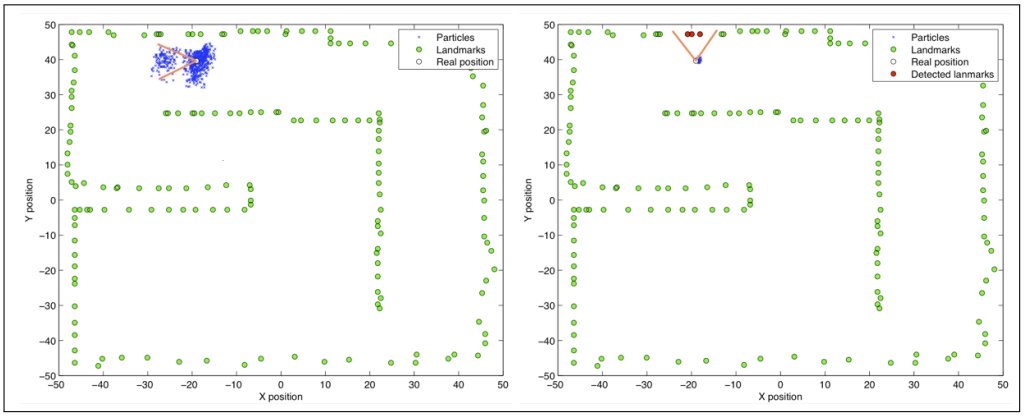

FIGURE 3.2. Step on the simulation. a) Left graph, localization without action selection. b) Right graph, localization using our strategy as action selection

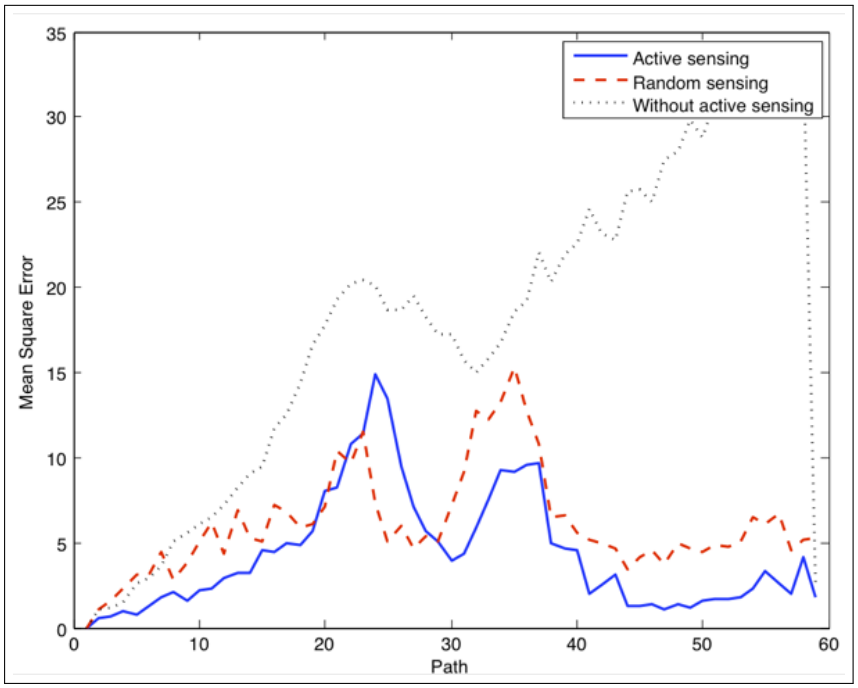

FIGURE 3.3. Mean square error along a simulated path

To have a more rigorous analysis of the robot behavior and the robot position estimation, we run the simulation several times on one of our maps (map from figure 3.2), and calculate the mean square error on the position estimate along the path. As mentioned before, we simulated different action selection strategies: active sensing with our optimum action selection, random action selection, and no action selection (look straight). Figure 3.3 shows the mean

square error along the simulated path, and that the lower error, in general, is obtained with our approach. The worst results are obtained when the simulated robot has a fixed observation action.

In figure 3.4 we see different results of the position estimation along the path. We also show the selected observation action (angle to move the pan-tilt unit) of our approach. We only show random action selection and our strategy because both are a lot superior to a fixed action strategy (looking straight in our case). In this simulation, the robot starts approximately at position $(-20, -40)$, and move counter-clockwise along the path. As we see, both approaches have a decent estimation on the firsts steps, but the random action approach tends to be less accurate than ours. For example, on the sub paths from $(40, -40)$ to $(40, 30)$, or $(10, 40)$ to $(-40, 40)$ our approach tends to maintain the robot localized, while random looking tends to get lost from time to time. Or when the robot arrives to the line between $(-40, 10)$ an $(10, 10)$, our approach maintains a better estimate longer, but eventually get a little lost (mainly because no landmark can be seen for some steps after that line). After that, both approaches get back on track. Now by looking at the actions selected by our approach (the different FOV on the graph) we can see that it select the correct looking angle in all but one place, when it arrives to $(-40, -10)$.

Now that we have seen that along the path, our strategy get better estimates than random and fixed action selection, we want to compare the average error on the complete path estimation. Table 3.1 shows the average mean square error (MSE) of the complete path estimate after performing 50 simulations. We see that in average, the active localization strategy proposed in this work, get lower error of the estimates than the other two approaches. This improvement is generated when the robots detect a landmark that under normal conditions, robot position and static sensor configuration or random looking, should not be able to detect. But with an active localization strategy the restriction of a detector with a fixed configuration is overcome, resulting on a improvement of the estimates of the robot position respect to a normal localization.

	Averaged MSE	Deviation
With selection	4.23	3.47
Without selection and straight looking	17.54	9.62
Without selection and random looking	6.09	3.06

TABLE 3.1. Average MSE and deviation, measured in pixels, of 50 simulations.

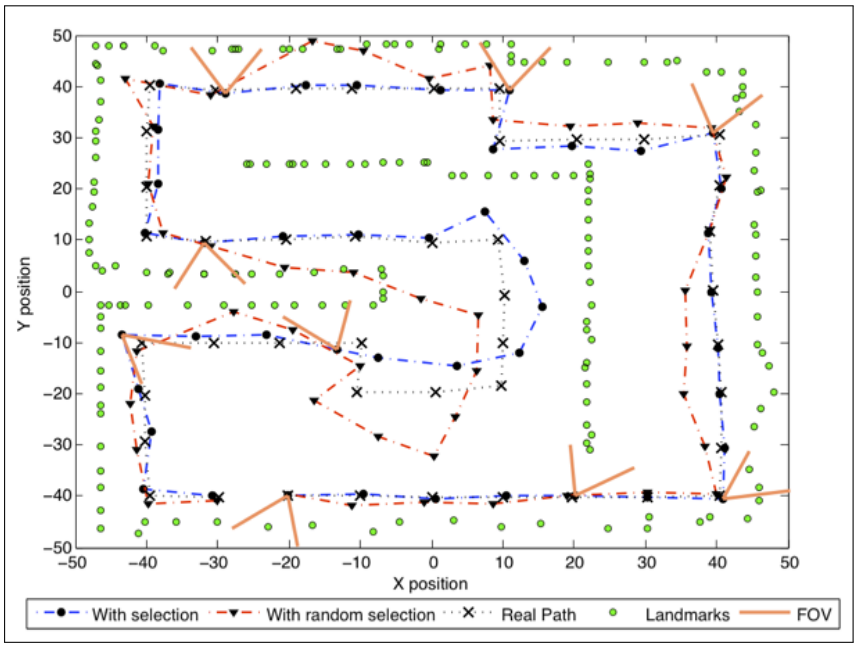

FIGURE 3.4. Example of estimated positions using different approaches. Also the looking direction, shown as the FOV of our approach

3.2. Real robot

In the previous section we have seen that under simulated conditions the proposed strategy produce an improvement on the robot position estimate. Now we show results of testing our proposed strategy on real environment. We used a Pioneer-3 AT robot, equipped with a Directed Perception PTU-C46 pan-tilt unit and a single Point Grey Dragonfly2 camera. We only use the pan movement of the pan-tilt unit, moving the camera over the horizontal plane only. We use CARMEN as an interface with the robot and the pan-tilt unit. To interact with the camera we used the open-source library libdc1394 that give us direct access to the camera. The environment where we ran the tests was an outdoor environment with synthetic landmarks, composed of cylinders of different colors that are easy to detectable by the camera

FIGURE 3.5. Pictures of the robot on the testing environment

(figure 3.5). In our implementation we use 2000 particles, and our action space is composed of 60 different pan angles (the complete 360 degrees divided in 6 degrees steps).

Our perception system was composed by a single camera, that detects well-known landmarks. These landmarks are built on black paper cylinders covered with two smaller patches

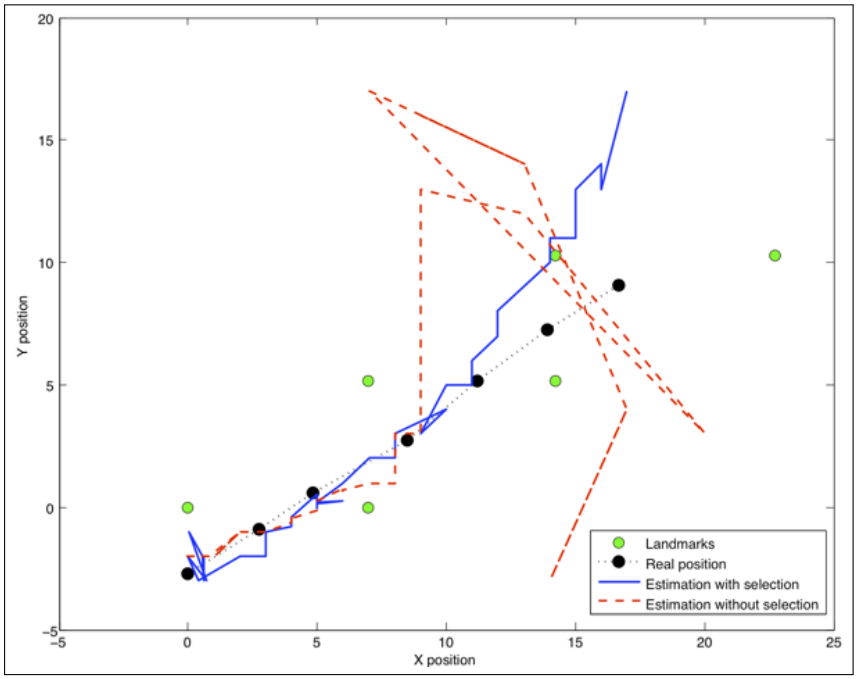

FIGURE 3.6. Position estimation of the robot movements

of distinctive and easily detectable colors. We know the size and shape of each landmark, allowing us to measure range, based on the detected landmark area, and bearing of detected landmarks based on its horizontal position on the image. We use a detection probability ($P(z_t = i|x_t, a)$) of 70% because our landmark detector misses some landmarks, changes on lightning conditions and shadows (among others) made our detection engine miss a few landmarks. This probability was estimated empirically by counting the times landmarks were detected under different conditions.

The odometry model is the same as in the simulation. CARMEN provides us with a raw estimation of the robot position; using them we calculate the initial rotation, traveled distance and final rotation of each step. We do not add noise to the odometry or detector data, because these values come with noise from the natural environment.

The test was carried out moving the robot in an open space where we placed six landmarks, we also marked on the ground the path the robot should follow, these marked points on the ground are our waypoints. On the real robot we tested two strategies: fixed pan angle and the our active observation strategy. Measuring the accuracy of the position estimate

is a lot more complex in this case. The robot performs a lot of computations while moves between waypoints, this make it hard to match the robot position with a specific position in the real path. Also there is a difference between using or not the active localization in terms of computation per second (CPS), and therefore number of points on the estimated path. That's why for the real robot, we measure the uncertainty of the estimation, as variance on the position estimate, while the robot moves around the map.

Figure 3.6 shows the result on one of the tests on the real robot. The robot moves from left to right in the figure. In this test, none of the robot path estimates are perfect, but we can see an improvement with the active localization. In the last steps both approaches tested fail because the robot fails detect some landmarks. This makes our strategy, that had a better estimate before, work better than the fixed angle strategy.

Figure 3.7 shows the variance on the robot position estimate of the different tested strategies. We see a lower variance using the proposed strategy than not using the active perception at all. The decrease in the variance can be explained because the robot have access to detect landmarks from poses were normally could not do it without the active perception, it not only had an improvement on lower average mean square error than the non-active version, but also decreased the uncertainty (measured as the variance) of his estimated position.

Test	Mean	Deviation
1	1.35	1.12
2	1.78	1.10
3	1.49	1.06
4	1.73	1.00

TABLE 3.2. Average waiting time in seconds

Besides the impact that our strategy have on the robot position estimates, we analyze the introduced overhead of the needed calculations. Using the logs we generated from the real robot experiment, we have the total time we ran the experiment, the number of calculation steps performed and the waiting times for the pant-tilt unit to reach the desired position, among others. With these data first we estimate the calculations per seconds, CPS, the robot perform with and without the introduction of the observation action selection strategy,

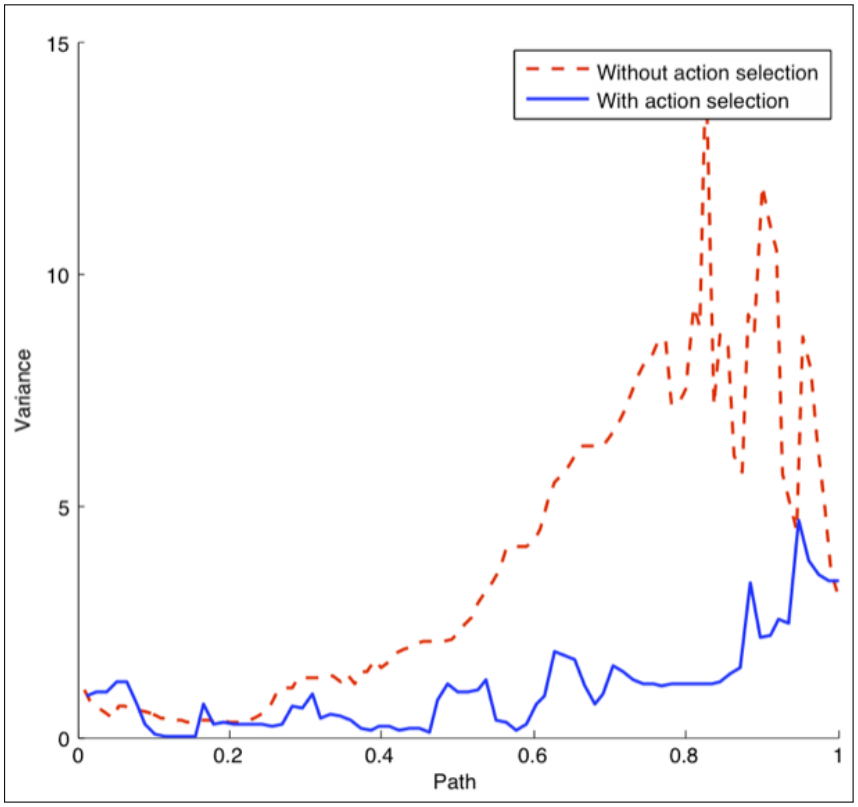

FIGURE 3.7. Variance of the real robot position estimation

results are shown in table 3.3. The introduction of perceptual action selection, under the same conditions, creates a decrease of 71% on the performance compared to localizing without selecting an observation action. This diminution has two main reasons: the introduction of more calculations to the process, and the waiting time on the observation action to finish. Table 3.2 shows the average waiting times of the pan-tilt unit to reach the desired position with its deviations. We see that the waiting times are relative high values taking into account that our localization implementation, without selecting observations actions, perform around $1cps$. To have a better estimation of the overhead introduced by the calculations itself, we perform an average CPS assuming there is no waiting time for the observation action to conclude. This assumption gives us an expected CPS of $1.07cps$, only 21% decrease on performance. We show the grouped results in table 3.3. With this results in mind, we conclude that much of the performance lost is due the waiting time on the pan-tilt unit to conclude the observation action.

Without landmark selection	With landmark selection	
	Base-case	Waiting time to 0
1.37	0.39	1.07

TABLE 3.3. Calculations per seconds

4. CONCLUSION AND FUTURE RESEARCH

Several strategies for active localization of mobile robots have been developed. In general terms these strategies make use of the robot movements in order to achieve the desired behavior. In this work, we have proposed, implemented and tested, an active localization strategy, using visual active sensors as well as a particle filter, with the assumption that the observation action is independent from the robot movements. In this strategy, we introduce an additional step into the normal localization process, in which we select the observation action. In order to select this action, we use the information gain to rate the possible actions also taking into account the cost of performing such actions. Since we use a particle filter for localization, the approximations used reduce the equations involved to sums and log-sums of particles weight, and can be calculated in an efficient way.

To test our proposed strategy, we performed simulations and carry out an implementation phase on a real robot. After the analysis of the results obtained from the different tests performed on both environments, we confirm that our proposed strategy for active visual localization yields better estimates of the current robot position.

Concerning the overhead introduced by the proposed strategy on the real robot, we identify that the main source of overhead is due to the waiting time arising from the selected observation action to be concluded. This waiting time appears from the use of a pan-tilt unit that moves the camera, this movement constituting our observation action. Estimation of overhead, assuming no waiting time, i.e. overhead is only due to calculation process, results in a rather low overhead, only 21% compared to the 71% from the base-case analysis considering waiting time. From a complexity point of view, implementation of the proposed strategy can be done in $O(|A|ln)$, with $|A|$ the cardinality of the action space, l the number of landmarks and n the number of samples, or particles, used to do the estimates.

Improvement to the present work can be achieved in several ways. For instance, expanding the active observation to SLAM, considering not only the uncertainty on the localization, but also the uncertainty on the map estimation in a way that is still simple to calculate. Also,

the use of a topological map in conjunction with the position estimate of the robot could result in a simplification and fastening of the calculations, further reading on topological maps on (Remolina & Kuipers, 2004).

Other possibility could be the changing of the observation action to a bias on the image perception, as in (Frintrop, Backer, & Rome, 2005) or (Navalpakkam & Itti, 2006). This type of observation action would allow the mixing of top-down with bottom-up knowledge, we could mix the top-down information as in this proposed strategy, with the bottom-up, data driven information arising from the image, similar to work at (Rasolzadeh, Björkman, & Eklundh, 2006). Once the observation action is changed, the waiting time for the action to conclude could be neglected; this implies the need to find ways for improving the overall strategy performance. One possibility to explore could be the use of a fixed sampling set to estimate the value of equation 2.9, as discussed in section 2.5.

REFERENCES

Araneda, A., Fienberg, S., & Soto, A. (2007). A statistical approach to simultaneous mapping and localization for mobile robots. *Annals of Applied Statistics*, *1*(1), 66-84.

Burgard, W., Fox, D., & Thrun, S. (1997). Active mobile robot localization. In *Proc. of the international joint conference on artificial intelligence.*

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory 2nd edition (wiley series in telecommunications and signal processing)*. Wiley-Interscience.

Davison, A. J., & Murray, D. W. (2002). Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 865-880.

Dellaert, F., Fox, D., Burgard, W., & Thrun, S. (1999). Monte carlo localization for mobile robots. In *International conference on robotics and automation, icra.*

Espinace, P., Langdon, D., & Soto, A. (2008). Unsupervised identification of useful visual landmarks using multiple segmentations and top-down feedback. *Robotics and Autonomous Systems*, *56*(6), 538-548.

Frintrop, S., Backer, G., & Rome, E. (2005, August). Goal-directed search with a top-down modulated computational attention system. In *Proceedings of the annual meeting of the german association for pattern recognition (dagm '05).* Wien, Austria.

Mitsunaga, N., & Asada, M. (2006). How a mobile robot selects landmarks to make a decision based on an information criterion. *Auton. Robots*, *21*(1), 3–14.

Montemerlo, M., Roy, N., & Thrun, S. (2003, October). Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (carmen) toolkit. In *Proceedings of the ieee/rsj international conference on intelligent robots and systems (iros 2003)* (Vol. 3, p. 2436-2441). Las Vegas, NV.

Navalpakkam, V., & Itti, L. (2006). An integrated model of top-down and bottom-up attention for optimal object detection. In (p. 2049-2056).

Rasolzadeh, B., Björkman, M., & Eklundh, J.-O. (2006, May). An attentional system combining top-down and bottom-up influences. In *International cognitive vision workshop (icvw06) at eccv2006.* Graz, Austria.

Remolina, E., & Kuipers, B. (2004). Towards a general theory of topological maps. *Artif. Intell.*, *152*(1), 47–104.

Roy, N., & Thrun, S. (1999). Coastal navigation with mobile robots. In *Advances in neural processing systems* (Vol. 12, p. 1043-1049).

Stachniss, C., Grisetti, G., & Burgard, W. (2005). Information gain-based exploration using rao-blackwellized particle filters. In *Proc. of robotics: Science and systems (rss).* Cambridge, MA, USA.

Thrun, S., Burgard, W., & Fox, D. (2006). *Probabilistic robotics*. New York: Cambridge University Press.

APPENDIX A. INFORMATION THEORY

We use tools from information theory to be able to work under uncertainty. This tools we use are two, entropy and mutual information.

The entropy of a random variable X with probability mass function $p(X)$ is defined as follow:

$$H(X) = \int p(X) \log p(X) dX \tag{A.1}$$

Entropy measure an average uncertainty in the random variable X. Using $\log$ with base 2, we get a measure of the number of bits average needed to describe the random variable X.

The mutual information of two random variables X and Y and joint mass function $p(X, Y)$ is defined as follows:

$$I(X;Y) = H(X) - H(X|Y) = \int \int p(x, y) \log \frac{p(x, y)}{p(x), p(y)} dxdy \tag{A.2}$$

The mutual information is a measure of dependence of two random variables. The mutual information is non-negative, symmetric on both X and Y, and equal to 0 if and only if X and Y are independent random variables.

More details on information theory and properties of entropy and mutual information can be found at (Cover & Thomas, 2006).

APPENDIX B. EQUATIONS

Here we show the detail of how we got equation 2.10. First, by the definition of entropy and applying Bayer theorem we get:

$$
\begin{aligned}
&H(s_t|o_t,a) = \\
&\quad -\int P(s_t|o_t,a)\log P(s_t|o_t,a)ds_t \\
&= -\frac{1}{P(o_t|a)}\Bigg(\int P(o_t|a,s_t)P(s_t|a)\log P(o_t|a,s_t)P(s_t|a)ds_t \\
&\quad - \log P(o_t|a)\int P(o_t|a,s_t)P(s_t|a)ds_t\Bigg)
\end{aligned}
$$

First, the value of $P(o_t|a,s_t)$ is 0 if the the landmark is outside the FOV of s_t (because then it simply can not be seen), so every integral only need to be evaluated over the places where the landmark can be seen.

By the law of total probabilities $P(o_t|a) = \int P(o_t|a,s_t)P(s_t|a)ds_t$. We also assumed $P(o_t|a,s_t)$ as a constant, P_z. And as we mentioned, the distribution of the position is independent on the action performed, so $P(s_t|a)$ ca be changed to $P(s_t)$in the equations.

$$
\begin{aligned}
&H(s_t|o_t,a) = \\
&\approx -\frac{1}{P(o_t|a)}\left(\int_{FOV} P_z P(s_t)\log P_z P(s_t)ds_t - P(o_t|a)\log P(o_t|a)\right) \\
&= \log P(o_t|a) - \frac{P_z}{P(o_t|a)}\int_{FOV} P(s_t)\log P_z P(s_t)ds_t
\end{aligned}
$$

Where FOV are the places s from which the landmarks can be observed.

Finally, the missing value, $P(o_t|a)$ can be calculated as follow:

$$
P(o_t|a) = \int P(o_t|a,s_t)P(s_t|a)ds_t = P_z\int_{FOV} P(s_t)ds_t \tag{B.1}
$$

APPENDIX C. IMPLEMENTATION DETAILS

We introduce 2 new operations to the normal loop of the filter. First, we search within the action space the best action to perform with he known information finally we tell the robot to perform the action. This is shown in the code below.

FILTER-WITH-OBSERVATION-ACTION(x_{t-1}, O, M) :

$x_t^- \leftarrow$ PREDICT(x_{t-1}, O)

$a \leftarrow$ GET-BEST-ACTION(x_t^-, M)

PERFORM-ACTION(a)

$S \leftarrow$ SENSE()

$x_t \leftarrow$ UPDATE(x_t^-, S)

To search for the best action we do an exhaustive search over the action space.

GET-BEST-ACTION(x_t^-, M) :

$max_inf \leftarrow -\infty$

for $a \in Actions$

do $info \leftarrow$ EXPECTED-INFORMATION(x_t^-, a, M)

if $info > max_inf$

then $max_inf \leftarrow info$

$best_action \leftarrow a$

return $best_action$

The Monte-Carlo method for EXPECTED-INFORMATION(x_t^-, a, M) is shown next.

EXPECTED-INFORMATION(x_t^-, a, M) :

$total_inf \leftarrow 0$

$h_x \leftarrow 0$

$entropy \leftarrow$ FALSE

for $m \in M$

do $p_m \leftarrow 0$

$plog_m^{FOV} \leftarrow 0$

$plog_m^{\neg FOV} \leftarrow 0$

for $\rho \in x_t^-$

do $\rho' \leftarrow$ SIMULATE-ACTION(ρ, a)

if $\rho' \in FOV_m$

then $p_m \leftarrow p_m + \omega_\rho$

$plog_m^{FOV} \leftarrow plog_m^{FOV} + \omega_\rho \log \omega_\rho$

else

$plog_m^{\neg FOV} \leftarrow plog_m^{\neg FOV} + \omega_\rho \log \omega_\rho$

if $\neg entropy$

then $h_x = -(plog_m^{FOV} + plog_m^{\neg FOV})$

$entropy \leftarrow$ TRUE

$I \leftarrow 0$

if $0 < p_m \leq 1$

then $I \leftarrow h_x - (\log p_m - \frac{plog_m^{FOV}}{p_m})$

elseif $p_m = 0$

then $I \leftarrow$ R

else $I \leftarrow 0$

$total_inf \leftarrow total_inf + p_m \cdot I$

return $total_inf$

Wtih R the constant entropy when resampling is done.

Finally SIMULATE-ACTION(a, ρ) just change the θ component of the particle ρ, simulating action performed by the pan-tilt.