Markov Based Localization Device for a Mobile Robot

GECHTER Franck  
Equipe MAIA  
LORIA UHP NANCY I  
BP 239 54506 Vandœuvre Lès Nancy  
gechterf@loria.fr

CHARPILLET François  
Equipe MAIA  
LORIA INRIA  
BP 239 54506 Vandœuvre Lès Nancy  
charp@loria.fr

**Keywords:** Robot Vision, Localization, Mobile Robot, Markov Decision Process, Stochastic, PCA.

**Abstract**

This paper presents a vision based localization system which can be used on a wide range of mobile vehicles. The principle of the algorithm developed is to merge the information given by a vision system with the data extracted from the moves of the vehicle. The image processing level is performed by using principal components analysis that allows low cost position estimation by using a representative set of images obtained during an initial exploration of the environment. This set of images makes it possible to represent the environment as a partially observable Markov decision process. The originality of this approach is the resulting data fusion process that uses both image matching and decision made by the robot in order to estimate the set of plausible positions of the robot and the associated probabilities. Furthermore, this stochastic localization device shows better results compared with the classical methods such as static neighborhoods. The main characteristics of this localization device are its robustness, its accuracy and its low cost compared with usual localization methods.

**1. Introduction**

A localization device for a mobile vehicle has to respect a lot of constraints. Mobile vehicles and robotic applications are prone to drifts on account of the inaccuracy of their sensors and their actuators. Such systems need to take those uncertainties into account to produce an autonomous and reliable robot. The localization device we have developed answers this issue: providing an effective and relative low-cost system which ensures a robust navigation. The method presented in this paper is based on a stochastic estimator of the effective position of the vehicle (according to the previous moves) coupled with a vision system.

Most of the vision modules currently used in robotics are based on feature extraction and interpretation. These extracted features allow the mobile platform to determine some key characteristics of the environment. Those algorithms are very efficient in structured environment such as offices and corridors. However, they have a major drawback: they all have a high computational cost and thus require a DSP in order to allow real-time processing. Moreover, some of the vision based localization algorithms, like stereoscopic reconstruction [3,4], require a precise calibration to be accurate enough. However reliable they can be, they are still restrictive because they require a lot of manipulation to obtain a useful vision module since the moves of the robot prevent the camera from staying calibrated. In order to overcome these drawbacks, we propose, in this paper, an approach based on a Principal Component Analysis (PCA) which is a very low cost algorithm initially developed for classification tasks. The PCA focuses on the picture obtained by cameras in its entirety, doesn't attempt to extract features and, in this, is not sensitive to the accuracy of the calibration. The Karhunen-Loeve transform presents other assets. First, the main part of the processing can be carried out on a non-mobile computer, thus, reducing considerably the computational cost during recognition task. Moreover, since the recognition task doesn't require a high frame rate to produce good results, parallel image processing can be added. On account of all its advantages, the use of PCA in vision processes has been already well-developed for a wide range of applications: for example, face recognition by taking into account the influence of light exposition [1,6] and robotic vision processes both on global and partial images [7,8]. Unfortunately, even if the learning of the environment could be considered in the future, the system initially requires a certain knowledge of the environment, in other words, a set
of images meant to represent the world in which the vehicle will evolve.

Besides, single PCA methods are not accurate enough to produce a robust localization system. Therefore, the results must be improved by incorporating new information such as the previous moves of the vehicle. The system could henceforth rely on estimations of its real pose and anticipation to reduce the space of search. Our two first localization systems were based on neighborhoods. It consists in constructing at each step a neighborhood which must contain the real state of the robot. The first system developed uses static neighborhood centered on the supposed pose of the vehicle (see part 3). The second uses growing neighborhood whose size is controlled by an estimation of the status of the robot; which can be lost or not (lien). The last system overcomes the limitations of neighborhoods method: it estimates more precisely the possible poses of the robot by using stochastic models (lien).

This paper will, first of all, shortly present the principle of the PCA Analysis and its results without any enhancement. Then, the next parts will describe the evolution steps of the algorithm from the use of the static neighborhoods to the use of the POMDP. At last, some results will be presented.

2. Principal Component Analysis and position estimation.

The main goal of Principal Component Analysis (or Karhunen-Loeve Transform), applied to the recognition task, is to sort out multidimensional and homogeneous data [11]. This sort is performed by determining the discrimination axis of the data base by computing the covariance matrix of the system. The classification between the elements of the data base is performed by interpreting their weight vectors which are the projections of the elements on the discrimination axes. Consequently, the comparisons aimed at classifying the data are performed in the eigenspace of the system by using only the weight vectors. The extension of this method to a set of images requires an adaptation. It has to be assumed that an image can be represented as a point in a n-dimensional space where n is the number of pixels in the image. Considering the principle of the algorithm, the set has to be composed of homogenous images in term of size and gray level range. (We can also applied the Principal Component analysis directly on color pictures) This method has the great advantage to perform the comparison between the pictures by taking into account only general properties instead of the classical methods, such as stereovision algorithms [9,10], that compare images by using key elements of the images. Furthermore, the PCA based algorithm has a very low computational cost, since the comparisons are performed with the weight vectors. This technique is very competitive compared with the standard ones, and thus can be used on a large panel of mobile platforms in order to obtain a near real time processing even on standard computers.

The algorithm is split into two parts. The first one is the construction of the space model that requires the computation of the covariance matrix of the system and its eigen-elements. Then, the weight vectors of the images are computed by using the eigenvectors matrix. This information (eigenvectors, weight vectors, mean image of the base) is saved as a structured space model. As explained before, the recognition task is performed by using the weight vectors of the images. As a result, the first stage of the recognition is a projection of the unknown image in the eigenspace. Then the unknown weight vector is compared with the weight vectors of the database. It is assumed that the more probable position of the robot correspond to the nearest image of the initial set.

Figure 1: Pictures used for the initial test in an indoor environment

The results obtained with this method without any enhancement are acceptable provided the lighting conditions are well controlled. In such a case, the recognition rate is near 36% in a slightly ambiguous environment (see figure 1). By contrast, when the environment is semi opened (i.e. when the light is both natural and artificial), the results sink up to 10%. (see figure 2)

Figure 2: Pictures used for the initial test in a semi opened environment

The main problem encountered with the PCA based localization, is the teleportation effect. Indeed, even if the position of the mobile platform is quiet the
same (i.e. less than 1 meter for instance) the results of the localization device can be very different. The use of PCA is consequently not accurate enough to be sole basis of localization system since successive recognition tasks can lead to very different estimations of the position. Because of this, the vision algorithm has been enhanced with interpretation and anticipation on the moves of the robot. The search in the database can be reduced by using some constraints such as taking into account the move of the robot in the position estimation. Thus, several methods were developed: the neighborhoods methods (static and growing neighborhoods first) and the Markov based methods (pseudo-MDP and POMDP). Those enhancement methods are explained in the next few sections.

3. Neighborhood methods

Two localization systems based on neighborhoods have been developed. One is based on static neighborhoods: the neighborhood of a position is defined the same way for all positions of the environment(in particular, they have the same size for each position). The second algorithm is based on dynamic neighborhoods (or growing neighborhoods) which are controlled by an estimation of the status of the robot: lost or not. If the robot is lost, the current neighborhood is extended in order to have a better chance to find a good visual estimator during the next recognition task.

3.1. Static Neighborhoods

The use of a selective search during the recognition phase allows both to reduce the number of comparisons between the weight vectors and to avoid teleportation problems during the position estimation. To this way, each element of the base is tied with a neighborhood of seven images corresponding to the most probable positions after the next robot move. This neighborhood is centered on the most probable position of the robot if it is assumed that the order given to the robot has been well performed. The robot can deviate from the expected move by having a small rotation or translation. So, the search can be restricted to the neighborhood represented on the figure 3 that takes into account the possible errors of the robot.

Considering this neighborhood, a selective recognition is performed by trying to find an equivalent to the acquired image among the seven images. Thus, we can consider that the recognition is forced, since we focus the search only among the seven images of the neighborhood.

![Figure 3: The neighborhood used to force the recognition of the position of the robot. Dark: the most probable position of the robot. Light: other probable positions considering the inaccuracy of the moves.](image)

The major advantage of this planning-like method is to reduce considerably the recognition time, since the use of the neighborhood allows to anticipate the future acquisition at almost every step. The results are pretty good (the recognition rate is over than 90%) while the real position of the robot can be found among the images of the neighborhood.

The main drawback of this method is the deviation obtained after a small number of moves (4 or 5 moves on average). This deviation is due to the small size of the neighborhood. Indeed, when the real position is not well recognized, we have no way to detect that the robot is lost or to recover the real position. Another problem is the fact that we have to label accurately the image base in order to easily construct the neighborhood.

In order to limit those problems, we can consider larger neighborhoods. The choice of the size of the neighborhoods must be done by balancing the precision required and the extent of the search in the database.

3.2. Growing Neighborhoods

Considering the limitations of the previous method, the use of the neighborhood has been enhanced by introducing a new parameter. It will determine if the distance between the image taken by the robot and the one recognized is small enough to conclude about the real position of the vehicle. In other words, this parameter permits to determine if the robot is lost. Thus, the robot is considered to be
lost when the distance between the recognized image and the current one is too big. In that case, the robot is allowed to make another action and the neighborhood is increased. The growth of the neighborhood stops when a satisfying image is found in the neighborhood. Figure 4 shows an example of a growing neighborhood when the robot is lost. On this figure each big square correspond to a position of the robot and each small square correspond to an orientation of the robot. The black dot represents the most probable position of the robot and the gray dots represent the growing neighborhood used to recognize the position.

On stage A, the robot moves in order to have the position showed on stage B. The recognition on stage B is well performed, the robot recognizes its real position. On stage C the robot decides to turn from a 45-degree angle. The recognition fails on C and the robot decides to turn from another 45-degree angle. The recognition fails on D either. Thus, the robot decides to move forward. The recognition succeeds on E and the robot updates its position in the environment. On stage F the robot virtual position is calibrated to the real position.

This type of neighborhoods presents greater advantages than the previous. Firstly, it allows to estimate when the robot is lost by using the distance criteria. This distance criteria can easily be linked to the policy of the robot. Indeed, if it is decided to have an exploration policy, the move of the robot through the area can be favored when the distance criteria is not respected. By contrast, if it is decided to get back to a given point in the environment with as much efficiency as possible the position of the robot which are recognized by the PCA algorithm can be favored. Second hand, the calibration of the position is more accurate than with the previous method since the recognized position is not forced among seven chosen by a slightly arbitrary way. Eventually, this method is more flexible because it makes it possible for the robot to recognize a wrong position and to recalibrate its position afterwards thanks to the growing neighborhoods. Nevertheless, this algorithm has some drawbacks. In particular, the use of the distance criteria can become a problem because it is simply characterized by a threshold that is empirically tuned. We have not found yet a manner to accurately determine this threshold by using mathematical considerations. This threshold is the real weakness of the growing neighborhood method because it requires a lot of manipulation to be well tuned and adapted to the environment where the robot has to move into. So another method has to be found which will be able to give as good result as the growing neighborhood algorithm while avoiding the requirement of parameters that have to be empirically tuned.

4. Markov Decision Processes and diffusion of the most probable position.

Because of the limitations of the neighborhood methods, a better method has been designed by using Markov decision processes (MDP’s) instead of the neighborhood in order to estimate the probable future position of the vehicle. This method is based on the same principle as the two previous ones. Indeed, its aim is to focus the search in the database by evaluating the probable future positions of the vehicle considering its effective move. While in the neighborhood methods all positions have the same probability, the MDP method diffuses the probability among the positions of the entire environment by using the start position and the actions that have already been made. In order to estimate the probability for the vehicle to be in a specific state (each state is characterized by an orientation and a position), the algorithm developed in this paper is inspired by the Simmons and Koenig method [2]. This model provides estimations, called “belief states”, of the real pose of the robot. Those belief states B(t,s) contain the probability at each step t, for each state s, to be the effective state of the vehicle. They are evaluated by taking into account the action made and the observation detected.

A Markov Decision Process (MDP) is a probabilistic automaton: a start state and a given action could lead to different states according to a probability distribution. A Markov model is then defined by

- S, a finite set of states (in our case, those states are the different poses of the robot (position and orientation))
- A, a finite set of actions (for our robot, they are: “go forward”, “turn left”, “turn right”)
- T[s’s,a], a probability distribution.

![Figure 4: The different stages of a growing neighborhood.](image-url)
The result of an action is determined by the Matrix T which gives the probability for the system, making a known action a, from a known state s, to arrive in the state s’. In MDP, the system is constantly aware of its real state. Unfortunately, such models don’t fit our goal because we want to localize a robot which doesn’t know precisely where it might be, or, in other words, what is its state. So, our issue requires a generalization of Markov Decision Processes: POMDP (Partially Observable Markov Decision Process). Henceforth, the real state of the system is not known but new information is available: the robot could make observations and gathers hints of its real pose. A POMDP consists of:

- A MDP (a set of states S, a set of actions A and a probability distribution T)
- O, a set of observations.
- P(o|s), the probability distribution of observing o in state s.

This model provides estimations, called “belief states”, of the real pose of the robot. Those belief states B(t,s) contain the probability at each step t, for each state s, to be the effective state of the vehicle. They are evaluated by taking into account the action made and the observation detected, this is the purpose of the following law:

\[
B(t,s) = K \times \left( \sum_{s' \in S} T(s,s',a) \times B(t-1,s') \times P(o,s) \right)
\]

Equation 1

This evolution law could be divided in 2 parts:

- The first part \( \left( \sum_{s' \in S} T(s',s,a) \times B(t-1,s') \right) \) represents the evolution of the belief states according to the action undertaken by the vehicle. It consists to estimate the probability to follow a transition from an unknown state to the state s. The probability of effecting the transition from state s’ to state s equals: probability of being in s’ in time t-1 * probability of taking transition to s (known that the system is in state s’). It turns out that the sought probability is the sum of the previous probabilities for all states s’.
- The second part \( P(o,s) \) updates belief states according to the observation made. If the vehicle senses a wall in front of him, it is more likely to be in a state near a wall (but it can’t be sure because of sensor’s imperfection). The factor K ensures that the sum of probabilities of presence B(t,s) equals 1.

### 4.1. First Approach: Pseudo MDP

Two possible approaches of Markov models can be used. In the first approach, Markov models can be considered as a passive help for the recognition system. To that way, a « pseudo Markov Decision Process » (pseudo-MDP) is used to evaluate iteratively the possible positions of the robot in the entire environment by using the knowledge of the start position and the succession of transitions that have been followed. In fact, the pseudo-MDP are Partially Observable Markov Decision Processes (POMDP) where the probability of observation is set to 1 for all observations (see equation 2). It means that the robot doesn’t take the information it can perceive into account to update its belief states.

\[
B(t,s) = K \times \sum_{s' \in S} T(s,s',a) \times B(t-1,s')
\]

Equation 2

The robot starts from a perfectly known position: the belief state is equal to 0 everywhere except in one state. Then the system evaluates step by step the new belief states and assumes that the real pose of the robot is among the states whose belief state is high (in other words, a threshold has been set, under which the state is not taken into account for the recognition task). The results obtained after the recognition task with this method are very encouraging in the beginning of execution but the belief states tend to be uniform since there is no recalibration during their evolution. As a result, after a few step the robot is in a configuration where only the PCA can give an estimation of the current position.

### 4.2. Second Approach: Real POMDP

In the second approach, a real POMDP can be used to improve the recognition system. In fact, this method is nearly the same as developed previously, but, feedback has been added to the system by using the observation given by the recognition task and by recalibrating the state of the robot. The key point of this method is using the results of the PCA in order to update the belief states of the system (see equation 3). The information furnished by the PCA is an evaluation of the distance between a given image of the database and the one perceived by the robot. In order to use this distance, the belief state update has to be modified to take into account that the shorter the distance between the image obtained by the camera and the image of database corresponding to the state s, the higher the probability for the robot to be
effectively in state $s$. Like in the previous method, the extent of the search in the database is reduced by using a probability threshold.

$$B_t(s)=K\times\left(\sum_{t<s} T(s',a)\times B_{t-1}(s')\right)\times\frac{1}{d(u,s)}$$

Equation 3

The results obtained with this method are promising. If the recognized image corresponds to the position that has the best belief state, this one becomes even more predominant since the distance between the observed scene and the corresponding image of the data base is small. By contrast, if the recognized image does not correspond to the real position, the belief state will not be strongly reduced, since, in that case, the distance between the images is large. Besides, this method avoids “sensor aliasing” problems since the information contained in the history of the moves made is kept throughout the belief state. The main advantage of this method is its robustness. Indeed, even if the robot is lost in the environment, the real position can be easily recovered in a few steps. In addition, recognition is not required at each step to accurately localize the vehicle. Thus, the vision can be used only when the robot is lost; in the other cases the POMDP is accurate enough to allow precise guidance through the environment.

5. Experimental results

5.1. The environment

In order to obtain the most representative base as possible, the environment is divided into small 1-meter by 1-meter squared area. Each of them is tied with eight images that correspond to the eight orientations N, N-W, W, S-W, S, S-E, E and N-E.

The environment is a 18-squared meter area where the lighting conditions are not well controlled. The area is opened to the sunlight through wide windows. This test area can be considered as a semi-opened environment. Moreover, the walls and the floor are high reflective and thus make it difficult to obtain a very accurate vision-based localization system. (figure 2)

![Figure 5: Some examples of the pictures used for the space model](image)

5.2. Comparison of the three methods developed

In order to perform the tests, a simulation program that takes into account the constraints of a real robot has been developed. In particular, it performs the actions of the vehicle by considering the hazard of the propulsion system. The structured data base of the environment is composed with 144 pictures. The test base is composed with 144 other pictures taken with different lighting conditions. The positions represented in the test base are shifted compared with the one represented in the initial structured base. We performed the same sequence for the three methods by counting the number of failures in the position estimation and the number of good recalibration of the robot after the loose of the real position at the previous step. The figure 7 shows the rate of bad estimations obtained by the neighborhoods methods (static and growing neighborhoods) and by the Markov based method.

![Figure 6: Rate of bad estimations](image)

The previous results point out that the POMDP based method has a very low rate of bad estimation of the position. The figure 8 shows an evaluation of the effectiveness of each methods developed in this paper. The effectiveness is compute by making a comparison between the rate of bad estimation and the number of recalibration success. Moreover, the results are set to 100% for the POMDP methods in order to compare more easily the different approaches.
The POMDP based localization method developed in this paper appears to be very reliable compared with the classical neighborhoods algorithms. Considering its low cost both at a software and hardware level, this algorithm can be used on a wide range of mobile platforms even on low powered ones. Its effectiveness even when there are bad estimation levels give it a decisive advantage compared to the standard localization algorithms. Moreover, this method has a greater flexibility that allows its use in many different environments and under highly variable lighting conditions. It doesn’t need to be tuned at the hardware level, since the camera doesn’t need to be calibrated, nor at the software level, as the method has no empirical threshold as opposed to the neighborhood algorithms. Furthermore, the recognition task by itself is no longer critical in this method, since the real position of the robot can be found even if the recognition task has failed.

6. Conclusion

Even if the growing neighborhoods method and the POMDP based method are similar as for the results presented in this section, the second method is more reliable. Indeed, the Markov-based method is not tied to empirical threshold, thus allowing a better adaptation skill in various environments.

Bibliography


