

# Negative Landmark Information Influence in FastSLAM

(Bachelor thesis)

Jarno Kuipers (s1664646, j.t.kuipers@ai.rug.nl)  
Sjoerd de Jong\* and Herman Kloosterman†

September 1, 2010

## Abstract

The solution to Simultaneous Localisation And Mapping (SLAM) has been the Extended Kalman Filter (EKF) approach for a long time. A more recent and better performing solution based on a particle filter is called FastSLAM. There are several shortcomings to this solution to operate in the real world. The shortcoming we aim at is the assumption that all landmarks are certain. The possibility of wrong observations or moving landmarks are omitted. We will introduce two extentional methods in FastSLAM to take this landmark uncertainty in account. Both methods will clean up the map by deleting very uncertain landmarks. We will compare both methods with the original FastSLAM method, our results are promising but also show that a solution to real world SLAM needs more research.

## 1 Introduction

In autonomous systems research, the Simultaneous Localization and Mapping problem (SLAM) is a well-known issue (Leonard, Durrant-Whyte, and Princeton, 1991). The combination of mapping an environment with landmarks and simultaneous localization of a robot in that environment is needed for a well functioning autonomous robot. In the past Extended Kalman Filters (EKF) were used for this particular problem (Thrun, Burgard, and

Fox, 2005), later on Montemerlo et al created a better solution called FastSLAM 1.0 (Montemerlo, Thrun, Koller, and Wegbreit, 2002). FastSLAMs advantages are the possibility to cope with large numbers of landmarks and continuing a long SLAM session without solely building on previously made errors. FastSLAM uses a particle filter for the robot location combined with a EKF per particle for landmark estimation. There is a 2.0 version of FastSLAM which is modified by also including actual measurements from the sensors (Calonder, 2006). This modification makes the algorithm more complicated, but upgrades the performance of the model. In this research we will use FastSLAM 1.0.

Previous research introduced other expansions to FastSLAM (Montemerlo and Thrun, 2003). Montemerlo et al introduced the information of unseen, but expected, landmarks to FastSLAM. Based on the internal map, landmarks can be expected when they are within sensor range. When a landmark is expected but not observed (negative information) it should be considered that the robot is not at the expected place or the previous observation of a landmark was not correct. The extension of negative information is part of the expanded FastSLAM solution of Montemerlo et al. They concluded that the use of this information has a positive effect on the performance of FastSLAM. This conclusion is verified by Guo et al, they also included negative information in a series of extensions to FastSLAM (Guo, Wang, Meng, and Qiu, 2007). Hoffman et al researched that negative information also has a positive effect on navigating with a Sony Aibo, without FastSLAM. (Hoffmann, Spranger, Gohring, Jungel, and Burkhard, 2006).

---

\*University of Groningen, Department of Artificial Intelligence

†Philips Consumer Robotics Drachten

Basically there are two possible methods to incorporate negative information in a FastSLAM model. The first method counts for every landmark how often it was seen and how often it was missed; for example while according to the models map, the landmark should be seen.

$$ratio = \frac{\textit{times a landmark was seen}}{\textit{times a landmark was missed}}$$

The ratio is the certainty of the presence of the landmark. When the certainty is below a threshold, then the landmark should be deleted in the internal map.

The second method also contains a position uncertainty of the landmarks, this is reflected in the size of the ellipse representing the possible position of the landmark. The size of the ellipse is a measure of the certainty of the landmark, a bigger ellipse represents a more uncertain landmark. When the size of the ellipse is over a certain threshold, the landmark is deleted.

These methods will be discussed further in the article. This article aims at the second method, Marko Doornbos will investigate the first method (Doornbos, 2010). The conclusion of this article shows the performance changes by the different methods used in FastSLAM.

In contrary to the real world the used implementation has no problem with identifying landmarks. It is implemented with knowledge about which landmark the model is observing. Real world robots cope with the problem that they have to determine which landmark is observed, the robot even have to determine whether the landmark was seen before, or that it is seen for the first time. The knowledge of knowing the observed landmark is called (known) data association, the uncertain opposite is unknown data association. Previous research (Guo et al., 2007) combined unknown data association with negative information.

This research instead expands FastSLAM with negative information, but with a variation to data association. This variation is certainty about the knowledge which landmark is observed, but the uncertainty of presence of the landmark. I embrace the hypothesis that a FastSLAM model will better perform with the use of negative information than without the use of this information. Montemerlo et al showed in previous research that using negative information in combination with other Fast-

SLAM extensions will let the model perform better, this model is based on unknown data association. (Montemerlo and Thrun, 2003) The difference between this research and previous named articles is the point that I will only look at the effect of using negative information in contrary to the other articles in which combinations of additional extensions were used.

## 2 Method

### 2.1 Previous research

As mentioned earlier a constantly right observation of a landmark is difficult. Another problem is identifying the landmark correctly, in the real world this could be ambiguous. This issues are not taken in account by the original FastSLAM. Guo et al. (2007) did research incorporating unknown data association and also using negative information. The ambiguity in identifying landmarks was coped by per-particle maximum likelihood data association, this is the simplest way to the data association problem. Per particle it samples the maximum likelihood of the data associations. Particles with correct measurement explanations will receive high probabilities, disconfirming measurement explanations by particles will receive low probabilities and will be removed in further sampling steps. The particle with the highest probability will account for the identification of the landmark.

Negative information is abstracting information from the absence of an expected sensor reading. Guo et al. (2007) solution is a binary bayesian filter, this filter estimates which landmarks should be observed. An expected and confirmed observation adds to the ratio used for the presence of the landmark, an unseen landmark while it is expected decreases this ratio. When the ratio falls below a given threshold, the landmark is removed from the map. The experiments combining per-particle maximum likelihood and the use of negative information with a binary bayesian filter resulted in a better performing FastSLAM algorithm. (Hoffmann et al., 2006)

## 2.2 FastSLAM

FastSLAM is mainly based on two different algorithms. A rao-blackwellized particle filter estimates the position of the robot and an Extended Kalman Filter (EKF) estimates the position of a landmark calculated per particle (Montemerlo et al., 2002). A particle filter produces a distribution of possible locations, every particle has a weight which yields the certainty of localization. The particle with the highest weight is the solution of the filter. The filter used in FastSLAM is rao-blackwellization, this reduces the complexity by sampling a subset of the particle distribution instead of sampling the whole distribution (Doucet, de Freitas, Murphy, and Russel). The advantage of using a particle filter is that it not immediate rules out possible robot positions, EKF-Slam chooses one possible robot position. When a false robot position is selected in the beginning, it is nearly impossible to regain the correct position. In a particle filter the decision is postponed, so that the possibility that the correct solution is deleted will be much smaller.

The FastSLAM algorithm is calculated recursively in three consecutive steps.

1. Prediction: particles positions updated with odometry information
2. Correction: probability of each particle is recalculated based on observations
3. Resampling: Building new particles based on previous particles with highest probabilities

### 2.2.1 Prediction Step

A temporary particle set is created, often called proposal distribution. This distribution yields the estimated new robot position based on the position on the passed time step and the actual motion of the robot, measured by the odometrics. In the new robot position calculations the odometrics are multiplied by a Gaussian random value, this gives a distribution of new positions.

### 2.2.2 Correction Step

In this step the map will be corrected according to the current observations through the sensors. For

each particle the EKF estimates the landmark positions based on the current observations. The updated landmark position is calculated per particle, the new position is the mean of all estimations by EKF for this landmark position.

### 2.2.3 Resampling Step

When all particles are processed in the prediction and correction steps, a new set of particles is sampled from the current particle set. For every particle a weight is calculated by

$$particle\ weight = \frac{target\ distribution}{proposal\ distribution}$$

The new particles are based on previous particles with a probability proportional to their weight. There is a discussion when resampling should take place. In our particular case we choose to resample when the particle weight is below a threshold. There will be no resampling in the first lap, so resampling only takes place when landmarks are seen for multiple times.

## 2.3 Extensions to FastSLAM

The FastSLAM algorithm was a significant improvement in solving the SLAM problem, but it consists of some assumptions which are not certain in the real world. As mentioned in the introduction FastSLAM assumes known data association. For every observed landmark, there is complete certainty that the landmark exists and it can be correctly identified. The certainty of presence could be doubted by noisy perception, or the landmark not being stationary. It is possible that a landmark first is perceived, but missed at the next expected perception. We have many situations in which to describe the difficulty to constantly successfully observe a landmark, following examples

- A person who walks in the vision of the robot
- An object obviously identified as a landmark from the front side, but imperceptible at the backside
- Noise in the sensor data stream causes the robot thinking to see a landmark

## 2.4 Pruning

It is possible that a robot observes a landmark once, but after a single observation the object is gone or unsuitable as a landmark. In that case it is eligible to be able to remove the landmark from the map. With use of negative information the model is able to calculate if a landmark is useful. Pruning landmarks gives the advantage of creating a map which only yields landmarks with a high existence likelihood. A map with only certain landmarks should result in a better navigating performance than a map with uncertain landmarks. To prune landmarks we use a threshold to set a minimum likelihood that landmarks should have. We researched two methods to prune landmarks with the landmark likelihood threshold, these methods are described in sections Counter and Ellipse. When a landmark should be pruned, it is removed from the list of landmarks per particle. So it is possible that a landmark is present in particle  $i$ , but absent in particle  $j$ .

## 2.5 Negative information Methods

The used implementation is based on the FastSLAM 1.0 matlab code from Bailey and Nieto (2004) The first step for both methods is to check which landmarks are expected to be observed from the actual robot position. For every time step a per particle bounding area in front of the robot is calculated, based on sensor range (set to 5 meters) and the robots field of view. When a landmark is within this area it should be visible. Then the observed landmarks are compared with the landmarks that should be visible. The determination whether a landmark is visible is a binary operator we added to override the observation method. Landmarks that are unseen, but expected are residual. These residual landmarks are common input for to the negative information methods, described below.

### 2.5.1 Method 1 - Counter

Per particle a counter is kept for every landmark. This counter holds the number of time a landmark is observed and the number of times a landmark is missed. As described in the introduction, the ratio between these counters could be seen as the cer-

tainty of presence of the landmark. Pruning landmarks is done by setting a threshold, if the ratio is below the threshold, the landmark will be pruned.

### 2.5.2 Method 2 - Ellipse

In the standard FastSLAM algorithm there is an ellipse per landmark per particle that describes the uncertainty of the location of the landmark. A greater ellipse means a greater uncertainty and vice versa. This ellipse is calculated by the covariance-matrix which is per particle recalculated in the correction step. The size of an ellipse is based on the diagonal values of the covariance-matrix, a multiplication of these diagonal values gives one value that represents the relative size of the ellipse. A greater diagonal value means a more uncertain landmark.

Pruning landmarks is also done by setting a threshold, whenever the diagonal value passes the threshold, the landmarks will be pruned. The threshold is set to 0.5 after testing different values. The code to implement this method is

```
if (ellipse is missed)
    covariance matrix = 2 * covariance matrix
end

if (diagonal covariance matrix value > 0.5)
    prune landmark from particles map
end
```

## 2.6 Experimental Setup

The performance of the model is the most important factor that should be measured. The performance is explained as the error in the robot position expected by the model. The error is the Euclidian distance measured in meters between the expected and the actual robot position. A smaller error means a better performing model. The robot will drive three laps around a series of waypoints in different environments. The performance of the original FastSLAM 1.0 will be measured with and without missing observations of landmarks. This difference shows the importance of creating a solution to cope with missed landmarks. Then the models with negative information will run the same loops in the same environments. We will use the

mean error of the last lap the robot drives for comparison between the different models. The results will show the effectiveness of the use of negative information and which method performs best. By testing in various environments we ensure that the models will not be tested on specific features in the environment. We created six different environments (Figure 1) with equally lengths, in those environments we varied the amount of landmarks between about 25, 50 and 75 landmarks (Figure 2). The maximal visual observation range is set to 5 meters, 25 landmarks in the environment result in roughly one landmark observation at the time. It follows that a landmark density of 50 and 75 result in two and three land simultaneous observations. The environments are created in such a way, that the landmarks are observed only during one pass per lap, so landmarks are not observed by accident when the robot is at another position at the map.

The extensions to FastSLAM are intended to cope with missed landmarks, so the amount of missed landmarks observations will be also varied. In the first lap the robot will observe all landmarks, the second and third laps there is a possibility that landmarks will be missed. For each environment with a specific landmark count the model will be tested with a landmark miss rate of 10%, 25% and 50%.

According to the landmark likelihood threshold in method 1, a landmark should be pruned if a landmark is more often missed than it is observed. The landmark likelihood threshold for method 2 is set as 0.5, this value is chosen by testing. A

threshold value of 0.5 does not immediate result in landmark pruning when a landmark is missed, but for most landmarks two consecutive complete passes while missing the landmark result in pruning. Taken into account that the robot drives three laps, these threshold settings are suitable for this experiment. Most landmarks are removed after two passes without seeing the landmark. These misses are in the second and third lap of the complete run, so it can be expected that missed landmarks are pruned. The threshold is also not too low, so that every landmark immediate is being pruned after a miss. To make a fair comparison between the different methods, the random variable seed is fixed so that the same variables are used. Particle sampling uses random variables, without a fixed seed there can not be made a fair comparison.

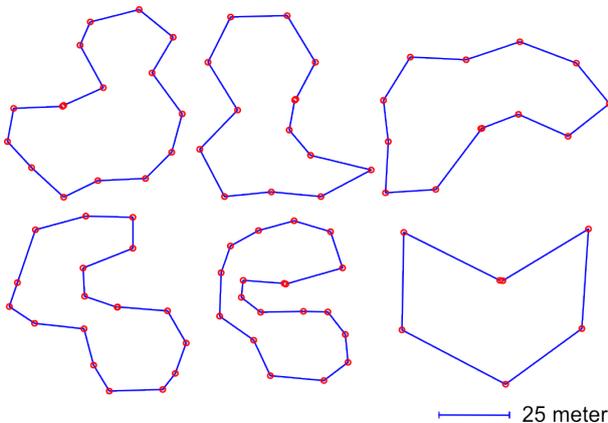


Figure 1: different environments

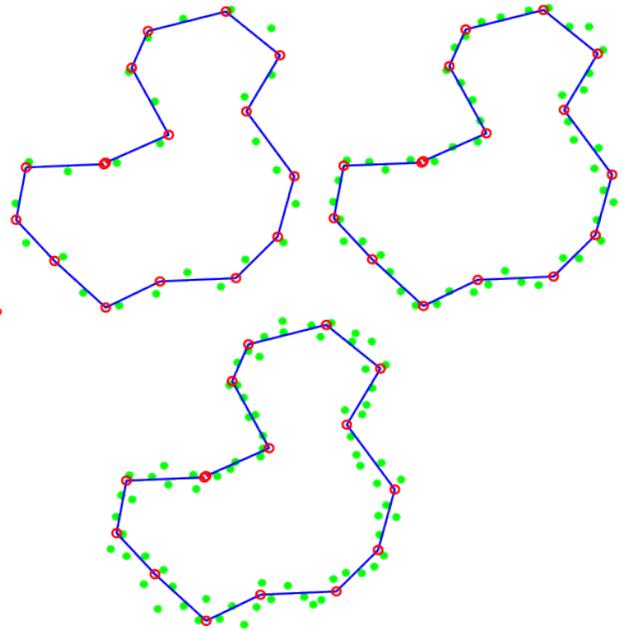


Figure 2: different landmark density

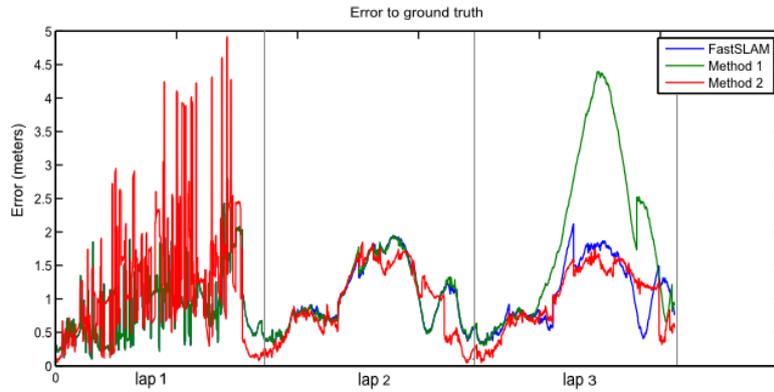


Figure 3: The performance expressed as the Euclidian distance between the model and the ground truth. The original FastSLAM, method 1 and method 2 are tested on environment number 5, with around 50 landmarks and a landmark miss percentage of 25%

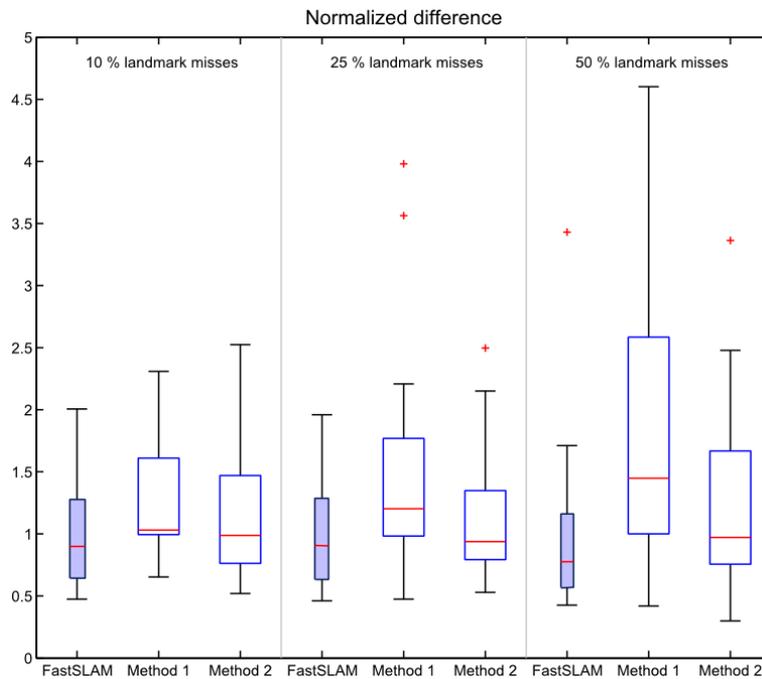


Figure 4: The mean error for the last lap per method, normalized to the original FastSLAM method with missing landmarks. The first blue shaded box per percentage section presents the mean error of the original FastSLAM for all environments. The boxplot is divided in three sections, each with a different percentage of missed landmarks in the second and third lap. For the box of method 1 in the 50 % landmark misses, there is one outlier not shown in the plot, this outlier has an error of 15.6 meters.

### 3 Results

Figure 3 is an example of complete run in a particular environment, this run is prototypical to the other environments. A difference between the methods is visible. The performance of the original FastSLAM and method 1, is for lap one and a part of lap two exactly the same, the difference starts when landmarks are being pruned. For every method the beginning and the end of lap is the least erroneous, figure 3 gives a clear distinction between the laps. The error changes most in the first lap for all three methods. After one complete lap the error remains more equal. The third lap of method 2 is remarkable compared to the other methods, the error is significant higher. This is common for the most environments.

In figure 4 all research data is incorporated, the difference in mean error in the third lap of all six environments is shown in a boxplot. Method 1 has a worse performance than the original method, the median of all mean errors is higher than the mean error of original FastSLAM. The median of method two is close to the original mean error. A higher percentage of missed landmarks result in more outliers in the boxplot.

P-Values	Original FastSLAM
Method 1	0.002
Method 2	0.422

Table 1: Significant difference in a pair wise T-Test

The results of a pair wise T-Test are shown in table 1. A value less than 0.05 means there is a significant difference, so the performance of method 1 is significant different than original FastSLAM, more precisely the performance is worse. Method 2 compared to FastSLAM is not significant different in performance.

### 4 Discussion

The question if the algorithm performs better with the introduced methods, can be answered negatively if we consider performance only in the Euclidian error. It could be expected that the performance not significantly becomes better, there is no addition of information in the model. Landmarks with very little, or no information are being

removed from the internal map. Performance increase only could be expected if the original method has ambiguous information and a better performing algorithm removes this ambiguity. Ambiguous information causes interference in decision making, in this case interference in navigating which is expressed in declining performance. In our case there is not a solution for ambiguity, but a solution for extracting useful information. The number of data associations calculations done by the algorithm is reduced by reducing the number of landmarks, so the introduced methods are upgrading the performance expressed in number of calculations done. When the Euclidian error does not change significantly, but the number of calculations reduces, we could say that the performance is better. The answer to the research question is that both introduced methods does not perform better, but with this new vision to performance method 2 will perform better.

An other suggestion for measuring performance could be incorporating the robots angle error. We only used the Euclidian distance to measure the error, because this is done by most researchers. The error measuring could be extended with calculating an error factor related to the angle error. When a robot travels a long distance without any observations, the error of the angle becomes important. The error increases for every time step if the angle is not correct.

In the results we noticed some minutiae in the figures. The algorithm resamples after the first lap, so there are multiple particles widely distributed. Sometimes there are multiple particles with the same highest weight, a small parameter change in any variable of the model may cause rapid changes of the particle with the highest weight. This could result in a quick change of estimated robot position, this explains the error hopping in the first lap. After one complete lap the new particles are based on the same parent particles with the highest weights.

#### 4.1 Data Association

Our research is based on an implementation with known data association. As mentioned the performance especially increases in data association calculating time. These calculations are done extensively in unknown data association, so the effect of decreasing calculation time will be more significant.

It could be expected that the decline of calculation time will have a significance positive influence to the performance of the model. Our second introduced method, the expanding of the landmark ellipse, has the potential to improve the performance of the original FastSLAM and therefore asks for more research in environments with unknown data associations.

## 4.2 Conclusion

The difference between the introduced methods is whether it changes the original algorithm in a certain way. The first method only holds a counter on which landmark pruning is based. The second method in contrary changes existing values in the algorithm. This explains why method 1 and original FastSLAM have the same error values before pruning takes place, method 2 changes the algorithm internally so that the model performance differs from the start. As mentioned method 2 performs better than method 1, the tested environments and parameters could have a large share in producing this conclusion. Further research is needed to explain the difference completely. Based on this result we advise to aim further research at the method 2, this method is relatively less complex to implement.

## References

T. Bailey and J. Nieto. Fastslam 1.0 and 2.0. <http://www-personal.acfr.usyd.edu.au/tbailey/>, 2004.

Michael Calonder. A brief comparison of two approaches to simultaneous localization and mapping (slam): Kalman filtering vs particle filtering, 2006.

M. Doornbos. The effect of using negative information on the accuracy of fastslam. Bachelor thesis, Department of Artificial Intelligence, University of Groningen, The Netherlands, 2010.

A. Doucet, N. de Freitas, K.P. Murphy, and S.J. Russel. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 176 – 183.

Lijin Guo, Huaxiang Wang, Qinghao Meng, and Yanan Qiu. Fastslam and its data association approaches for mobile robots. In *Automation and Logistics, 2007 IEEE International Conference on*, pages 1952 –1957, 18-21 2007.

Jan Hoffmann, Michael Spranger, Daniel Gohring, Matthias Jungel, and Burkhard. Further studies on the use of negative information in mobile robot localization. In *Proceedings of the 2006 IEEE international Conference on Robotics and Automation*. IEEE, 2006.

J. Leonard, H. Durrant-Whyte, and N.J. Princeton. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems '91. Intelligence for Mechanical Systems, Proceedings IROS '91*, pages 1442 – 1447, 1991.

Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *Robotics and Automation, 2003. Proceedings. ICRA 2003. IEEE International Conference on*, pages 1985–1991, 2003.

Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.

S. Thrun, W. Burgard, and D. Fox. Probabilistic robotics. In *Cambridge: The MIT Press. ISBN 0262201623*, pages 48 – 53, 2005.