CRANFIELD UNIVERSITY

VICTOR JOURNE

Multiple UAVs SENSOR OPTIMISATION

SCHOOL OF AEROSPACE, TRANSPORT
SYSTEMS AND MANUFACTURING

MSc THESIS

CRANFIELD UNIVERSITY

School of Aerospace, Transport Systems and Manufacturing

MSc THESIS

Academic Year 2014-2015

Victor Journé

Multiple UAVs, sensor optimization

Supervisor: Dr H.S Shin

August 2015

This thesis is submitted in partial (50%) fulfillment of the requirements for the degree of Master of Science

Abstract

Targets detection with several UAVs turns out to be a significant issue for a lot of reasons, especially if UAVs are equipped with quite low-cost cameras. Indeed, contrary to radar detection where errors rates can be known in advance and vary hardly, detection based-vision processes are subject to many unknown factors of uncertainty, as illumination variations. Moreover, since UAVs or targets are moving, viewpoints changing may have to be taken in account. The scope of this Master thesis settles on high level considerations, where each asset scans the environment and sets a binary map of target presence. As uncertainty in the process detection cannot be ignored, an obvious rapprochement can be done with Image Registration. This is a very general problem in computer vision and naturally in automatic target recognition. The main goal is to unveil the spatial transformation between two frames. To achieve it, two sets of interest points belonging to the two frames are extracted, which leads to correspondence issues.
Although the spatial transformation itself was solved in this paper only for simple cases, this Master thesis offers to determine false alarm probabilities of the detection process as well as the prior targets presence with only two overlapped scans of the environment. A kind of Bayesian filter was implemented to recursively recover a fixed pattern on the ground and simultaneously estimate the UAV position flying over it. Finally, a multiple targets tracking problem was considered by fusing local detection of several cameras.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my principal supervisor Dr. Shin for the continuous support, for his availability and for his kindness during the last five months. His guidance helped me in all the time of research and writing of this thesis.

This Master Thesis was partially supported by the company Selex ES, on the behalf of Richard Theobald, who provided insight and expertise that certainly assisted the research.

The other support I received comes from the Course Director of my Master, Dr. Tsourdos, present and attentive in case of need.

Finally, A great thanks to the British weather that has prevented me from leaving the Library for days and nights.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Detection theory has applications in many fields such as quality control or target tracking. With several assets focus on the same phenomena, the immediate question that arises is how to fuse their local decisions. The framework of this paper turns round to ground targets detection by using standard or infrared cameras, eventually embedded in UAVs. Complications can be introduced in layers; fixed targets on the ground with only one UAV, several targets moving independently with fixed cameras and finally several moving targets with moving cameras.

The foundation of this Master thesis is to find out how far each asset can be trust and how many targets are lying on a map, by merging the prospection maps of each asset. A prospection map is the map explored by an UAV or by the scene viewed by a camera, composed in our case of binary decision; 0 if the target is supposed absent and 1 conversely. A such map registers bits in a spatial representation of the environment. Binary decisions are taken by detection algorithms not studied in this paper, but mentioned in the literature review.

Those decisions are subject to errors; false alarms or miss detections due to the imperfection of the process and to physical noise. By comparing only two binary maps representing the same environment, we will show how to know accurately these probabilities and the prior presence probability to have a target. The mathematical justification and the main assumptions are given in the chapter 2.

This approach can found applications in Feature-based Registration problem, where arises the determination of an optimum spatial transformation between two set of features belonging to two consecutive frames. A simple problem of template matching was considered by using a Bayesian filter. The goal is twofold; estimate the position of a platform onto a unknown fixed pattern and recursively recover it. The difficulty is to overcome the high and unknown false alarm probability of the detection algorithm, with no prior information about the drawn ground pattern.

The third undertaken problem encompasses a targets tracking issue. Several independent targets are moving with a simple motion model onto a map in a environment very cluttered, with occlusions. Targets can be destroyed or appear. Three camera viewpoints are available using the same detection process, with again false alarm rates unknown and no prior information about the current number of targets. The goal is to merge those information to get the targets paths.

Each problem will be dealt with separately in their dedicated chapters (4, 5), and are mathematically supported by the third one (3). Further works and discussions are made directly inside every chapter.

Chapter 2

Literature Review

2.1 Detection probability

The more general problem regarding the theory of the decision is the issue of choosing between two hypothesis H_0 and H_1, which enables to explain or interpret observed data x. In our case the question is whether or not, in view of data, which hypothesis is true : either there is no target (H_0) or the target is here (H_1). Let us assume now that we have a sensor outputting 0 in the case of it chooses H_0 and 1 for H_1; this is its decision rule. We assume a parametric measurement model $p(x|\theta)$ and the goal is to estimate the parameter $\theta \in \Theta$. The parameter space Θ is made up of Θ_0 and Θ_1 forming a partition of Θ. In detection theory, we wish to know which hypothesis is true to take the appropriate decision.
The hypothesis that has to be considered is:

$$\begin{cases} H_0 : & \theta \in \Theta_0 \\ H_1 : & \theta \in \Theta_1 \end{cases}$$

The error taken by the detector can be of two types:

The detector chooses the hypothesis H_1 whereas the signal obeys to the parameter θ_0. This is a *false alarm* P_{FA}. Conversely, the detector chooses H_0 whereas the signal's type is θ_1This is a *miss detection* P_M.

If the probability to observe H_i is known, the Bayes theorem can be applied to determine, in view of a new set of observation, whether or not the target is present in our case. The Bayes detection rule states that the good hypothesis is the one that maximize the posterior. This method drives to minimize the error of classification, i.e. correct classification between the two hypothesis as it can be shown [1]. This provides a satisfying answer as long as the prior probabilities $p(H_i)$ are known. The Neyman Pearson approach relaxes this condition by finding a threshold setting the probability of detection maximal while keeping P_{FA} under

a chosen value.

A contiguous issue remains, to which my effort was lead: the estimation of the false alarm probability, the detection probability and the probability to have H_i in the case of simple detection, i.e. binary decision.

A good illustration of this approach may be highlighted with the problem of populations census that zoologists have tackled for a long time. For instance, the counting of nest of a specific bird specie over a given area is not an straightforward issue since they can be confused with other species and the observer might miss right ones. The idea related by article [2], which upon it all zoologists are based, suggests to have two observers relating the location of the nests spots. These strategy is called double observer point-count. Several estimates more or less empirical could be built depending on some assumptions. The more general problem relies on unknown detection probabilities of the both observers. The ultimate estimator aims at tending towards the true number of nests.

2.2 Template matching

An important preamble to the next subsections is the template matching matter. The goal is to find the best location of a template as a small part of a whole image. This issue supports naturally both problems; mapping and target tracking.

Several approaches can be considered depending on the context. If the template image has a lot of strong features, a feature-based approach is suitable. It is computationally efficient. Without strong features, more basic methods using the template as a convolution mask enable to find the best transformation bounding the two frames. Since convolution in the spatial domain is a multiplication in the Fourier domain, we have to compute once the Fourier transform function of the mask and the whole map before multiplying together and then compute the inverse transform of the product. It yields all possible translations in one step. To find out the rotation and scale effects between two images, Fourier analysis and cross-correlation can be very time consuming since every configuration for a particular rotation and scale effect have to be tested. This is no conceivable for real-time applications such targets tracking or simultaneous localization and mapping (SLAM) issues, where a featured-matching approach is generally used. Some algorithms as [3] can significantly reduces this computational time by pruning in the search loop impossible solutions. This algorithm has also the advantage to be contrast and brightness invariant.

These other solutions, which discard grayscale information, can drastically reduce the computational time, even with few features. Those strategies focus

on edges or interest points or more difficult pre-proccessing transformations. We could quote the Scale-invariant feature transform (SIFT), which is an algorithm detecting and describing local features in an image, like corner. This is also the case of Hough transform. The standard version is able to find lines in an image. A binary image with only the edges are commonly used as input of this particular transformation. In the case of line detection, every edge pixels map to a sinusoidal in a 2D parameter space representing all possible lines that could pass through that image point. Each lines corresponding to one particular pixel (x, y) at one in the input space can be express as the sinusoid curve $r = xcos(\theta) + ysin(\theta)$. Thus, the resulting Hough space (r, θ) is filled up these curves and their intersections correspond to a line of the input space. The generalized Hough transform [4] extends the principle to identifying positions of arbitrary shapes.This strategy is not really sensitive to noise and offers a relevant solution to template matching in the frame of vision-based navigation. Furthermore, variations in the shape such as rotations, scale changes correspond to straightforward transformations inside the Hough space .

That is why scale invariant features transforms offer interesting outlets in the field of object recognition or video tracking. The idea is to extract features from an image, which are invariant by scaling effect rotation and translation. The motivation is to find features good enough to describe an object seen under different viewpoint. It could turn out very useful to find the motion of an object from two consecutive views. There is two main parts for this approach: key features detection and pairs association between two consecutive frames. Even though the method is very robust, the computational time is substantial.

2.3 Ego-motion

2.3.1 Probability Grids

The self localization of a robot is often divided into two problems, position tracking, which uses the initial point and odometer for instance, and global localization. The latter faces the issue of relocalization under global uncertainties after being lost for instance. The question of the environment representation is crucial and two main approaches can be undertaken, either a metric or topological map. Both can produce good results depending on the context. Metric map are based on spatial or plan representation of the objects position and are sensitive to noise. Whereas topological map only considers places and relations between them by using graph structure. Uncertainty is dealt with probability representation of the map. Occupancy grid is the discrete version of the metric representation, mapping the space with the probability to have an object in a cell. When the environment is rigid and fixed, several approaches as [5] are used and have been successfully

adopted.

The paper [6] implements a successful fusion between those two sub problems by using occupancy grids and multiple hypothesis tracking. The developed solution deals with the global localization problem by tracking multiple Gaussian hypothesis over the space of possible location of the robot's map. Two sensor types are used: ultrasonic range finder sensors keeping a constant alignment thanks to a compass and odometer measure. The initialization constructs a set of hypotheses of possible robot position. The odometer predicts the next position for each hypotheses. When a new scan measurement is available, a current local occupancy map made up with 3 states (occupied, empty or unknown) is created. The strength of the match between it and the stored former map is used to provide a likelihood for every assumed past position. Then an updated prior probability for each assumed position is computed and the process is repeated.

2.3.2 SLAM

The simultaneous localization and mapping is a key issue in robotics field and a number of research teams have developed robotic system for mapping indoor or outdoor environments. Visual SLAM (VSLAM) [7] offers a new strategy to the traditional SLAM based on range finders, and proposes to deal with rich information from low-cost cameras. The methodology lies on the tracking of interests points or landmarks to localization purposes. Nevertheless, VSLAM argues to overcome data association issues and scalability matters. So as to find correspondence between to consecutive frames, several image association methods are possible and are quoted in previous subsection .The standard mono camera VSLAM proceeds in this following manner. A Extended Kalman filter is used, the state vector includes the position and orientation of the camera enhanced over the navigation with the position of the objects detected in the current image. The predicting step is based on the odometry measurement. The updating state derives from the template matching based on the appearance of predefined object template. Indeed, to avoid large image association computation a prior data base of target or landmark objects is created. Thus, this high level environment representation of objects enables real time applications. The central problem of object range finding is solved with Inverse Depth Parameterization [8]. This paper gives a measurement model to find roughly the coordinate of the object relative to the camera location. However, this method is suitable only for tracking objects whose appearance doesn't vary too much during motion.

2.4 Target tracking

Targets vision-based tracking can be seen as a sub problem of the template matching one; consecutive frames association cannot be ignored. A suitable approach

for real time surveillance applications is Feature Point Tracking. In an image sequence, moving objects are reduced to their more elementary representation; a point or a set of points attached to a feature. The point are treated indistinguishable and kinetic constraints are solely used to get the correspondences. Feature points may temporarily disappear, enter and leave the view field. The only hypothesis made is about the motion of those points, supposed smooth and with a bounded speed. The goal of algorithms proposed in [9, 10] is to find the best point association forming trajectories so as to minimize a global cost function, which takes in account the change of direction and the speed. The limitations come up when there is too much feature points to associate (more than 50 over 250×250 pixels). Yet, for small targets, those resolutions offer good results but encounter trouble for large one with non-rigid assumption. In the contrary case, for large deformable targets, other information as color or shape can be treated for both association and deformation solving.

Chapter 3

Mathematical Approach

This part aims to tackle the problem of several observers who report the target presence. The fixed targets laid on a map and some UAVs fly around this area to detect it. For several reasons, the presence information given by each asset for each space unit on the map may be distorted; the binary assessments may be reversed with a unknown probability. It could be due to a deliberate or accidental hidden data set. At this stage, data are provided by two assets and collected in a ground station.

3.1 Mathematical problem

3.1.1 Assumptions

Let us suppose that we have two set of measurements m_1 and m_2 , correlated but independent.

Indeed, those sets are created following this way:

$$\begin{aligned} m_1 &= b \oplus e_1 = b.\bar{e_1} + \bar{b}.e_1 \\ m_2 &= b \oplus e_2 = b.\bar{e_2} + \bar{b}.e_2 \end{aligned} \tag{3.1}$$

b The sequence of the phenomena to measure, equals to 1 when it occurs and 0 elsewhere.

e_i The error source, its effect is to reverse the true information embodied by b.

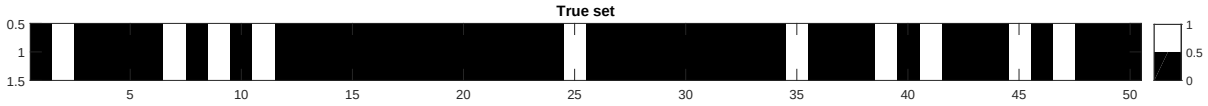

Figure 3.1: True set

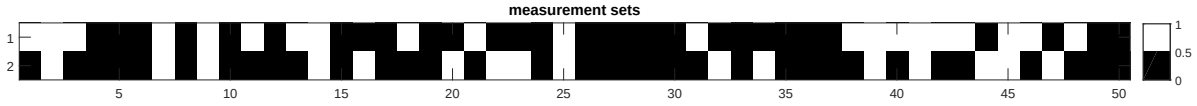

Figure 3.2: Measurement sets

We do assume that e_1 and e_2 are independent and do not depend of b as well. The last assertion is not really realistic and not consider any differences between a false alarm and a non detection, its probabilities are equals. The probability to tell lies is so indifferently the false alarm or non detection probability.

The probabilities of occurrence of apparition of b, e_1 and e_2 are unknown and the following part proposes to find out those three variables. As an illustration, let us assume that ten targets lie on a space of 50 spots, depicted in figure 3.1. We have at our disposal two noisy sets created as in the previous paragraph given by figure 3.2.

The only information we have is the two following set of measurements of figure 3.2. Here the probabilities to say the truth (i.e. to not reverse the true bit) were chosen at 0.7 and 0.8. These values are obviously unknown for our problem.

3.1.2 Resolution

At this stage, we know the number of 1 of each set, let us call $M_1 = \sum m_1$ and $M_2 = \sum m_2$.

By comparing for every spot the booleans produced by the two assets, we could determine the occurrence of each of the four outcomes. Both assets agree; (0,0) or (1,1), or disagree: (0,1) or (1,0). This is summarized in the following table 3.1. Let call P_0, P_1 , P_2 and P_3 those occurrences.

For instance, P_0 is the number of time where m_1 and m_2 are simultaneously equal to 1.

To switch the problem towards probability formulation, we have to normalize those values by dividing it by n, the length of a sequence. In probability terms, the probability p_0 can be expressed as, using the formula of the conditional probability:

	P_0	P_1	P_2	P_3
m_1	1	1	0	0
m_2	1	0	1	0

Table 3.1: The possible outcomes and their number of occurrence

$$\begin{aligned} p_0 &= p(m_1 = 1, m_2 = 1) \\ &= p(b = 1).p(e_1 = 0 \cap e_2 = 0) + p(b = 0).p(e_1 = 1 \cap e_2 = 1) \end{aligned}$$

That is to say with shorter notations:

$$p(m_1 m_2) = p(b)(1 - p(e_1))(1 - p(e_2)) + (1 - p(b))p(e_1)p(e_2) = p_0 = \frac{P_0}{n}$$

Let us note $(x, y, z) \in [0, 1]^3$ the triplets such as:

$$\begin{cases} x = p(b) & \text{presence probability} \\ y = (1 - p(e_1)) & \text{detection probabily of the first sensor} \\ z = (1 - p(e_2)) & \text{detection probabily of the first sensor} \end{cases}$$

We have easily this system after writing the probabilities of the four outcomes

$$\begin{cases} p(m_1 m_2) = xyz + (1 - x)(1 - y)(1 - z) & = p_0 \\ p(m_1 \overline{m}_2) = xy(1 - z) + (1 - x)(1 - y).z & = p_1 \\ p(\overline{m}_1 m_2) = x.(1 - y).z + (1 - x)y(1 - z) & = p_2 \\ p(\overline{m}_1 \overline{m}_2) = x(1 - y)(1 - z) + (1 - x)yz & = p_3 \end{cases}$$

This system is highly non-linear and its resolution could appear quite hard. To fire up this issue, we can rearrange it as follow, in order to delete the term in xyz.

By the variable change $X \leftarrow x - \frac{1}{2}$, $Y \leftarrow y - \frac{1}{2}$ and $Z \leftarrow z - \frac{1}{2}$ and some developments, we get:

$$\begin{cases} p_0 = +XY + XZ + YZ + \frac{1}{4} \\ p_1 = +XY - XZ - YZ + \frac{1}{4} \\ p_2 = -XY + XZ - YZ + \frac{1}{4} \\ p_3 = -XY - XZ + YZ + \frac{1}{4} \end{cases}$$

$$\Leftrightarrow \begin{cases} XY = (p_0 + p_1 - \frac{1}{2})/2 & L_0 \\ XZ = (p_0 + p_2 - \frac{1}{2})/2 & L_1 \\ YZ = (p_0 + p_3 - \frac{1}{2})/2 & L_2 \end{cases}$$

In the case of at least one of the variable X, Y or Z is equal to 0, that is to say a sensor lies one over two times or the phenomena to observe occurs the half of the time, we cannot solve the previous system. We cannot deduce anything, only that one of them at least is null. This situation is observed whenever $p_0 \backsimeq p_1 \backsimeq p_2 \backsimeq p_3$. In the other case, If $X \neq 0$, $Y \neq 0$ and $Z \neq 0$, we could get the next expressions by achieving the operations on the lines L_1 , L_2 and L_3;

$$\Rightarrow \boxed{\begin{cases} X^2 = \frac{(p_0+p_1-\frac{1}{2})(p_0+p_2-\frac{1}{2})}{2(p_0+p_3-\frac{1}{2})} & \leftarrow \frac{L_0.L_1}{L_2} \\ Y^2 = \frac{(p_0+p_1-\frac{1}{2})(p_0+p_3-\frac{1}{2})}{2(p_0+p_2-\frac{1}{2})} & \leftarrow \frac{L_0.L_2}{L_1} \\ Z^2 = \frac{(p_0+p_2-\frac{1}{2})(p_0+p_3-\frac{1}{2})}{2(p_0+p_1-\frac{1}{2})} & \leftarrow \frac{L_1.L_2}{L_0} \end{cases}}$$

To take the square of those equality, we have to know the sign of X, Y, Z. As we could assume without loss of generality that the sensors are correctly designed to sense the phenomena, the probability to reverse a bit is less than 0.5. That is why Y an Z could be assumed to be greater than 0.

Regarding X, we can a priori guess nothing about its sign. If the phenomena occurs more than one over two times ($x > 0.5$), then X is positive and conversely. However, its sign could be deduced by counting the number of 1 in m_1 for instance and compare it to $n/2$. Indeed, if the sample is large enough and the number of m_1 is far enough from $n/2$, greater for instance, then X is likely to be worth more than 0.5. Without satisfactory justification, and as we focus on target detection in a large space, we will assume in the whole next paper that we have less targets than space spots, and so that X is negative. The next equation comes after these assumptions.

$$\begin{cases} x = sign(p_0 + p1 - 1/2).\sqrt{\frac{(p_0+p_1-\frac{1}{2})(p_0+p_2-\frac{1}{2})}{2(p_0+p_3-\frac{1}{2})}} + \frac{1}{2} & \text{presence probabiliy} \\ y = \sqrt{\frac{(p_0+p_1-\frac{1}{2})(p_0+p_3-\frac{1}{2})}{2(p_0+p_2-\frac{1}{2})}} + \frac{1}{2} & \text{detection probability of the first asset} \\ z = \sqrt{\frac{(p_0+p_2-\frac{1}{2})(p_0+p_3-\frac{1}{2})}{2(p_0+p_1-\frac{1}{2})}} + \frac{1}{2} & \text{detection probability of the second asset} \end{cases} \tag{3.2}$$

3.1.3 Fast implementation.

The goal at this stage is to compute as fast as possible p_0, p_1, p_2 and p_3.

We could naively create a loop over the length of the measurements to count the occurrences reminded by table 3.1. This solution was implemented and quite time consuming since the sequences can be very long. To alleviate this problem, we can notice that the terms present in the formula giving the three probabilities can be expressed as:

$$\begin{cases} p_0 + p_1 = \frac{1}{n}\sum m_1 = \frac{M_1}{n} \\ p_0 + p_2 = \frac{1}{n}\sum m_2 = \frac{M_2}{n} \\ p_0 + p_3 = \frac{1}{n}\sum m_1 \odot m_2 = \frac{1}{n}\sum m_1.m_2 + \frac{1}{n}\sum \bar{m}_1.\bar{m}_2 \end{cases}$$

With m_i the sequences of measurements and M_i its number of ones. $\cdot\odot\cdot$ is the logical operation called XNOR, the complement of the exclusive OR (XOR) operation. It is worth one as soon as m_1 and m_2 are in the same logical state and zero else.

This relations are indeed convenient because both Matlab implementation or embedded implementation are easy to achieve by this way.

3.1.4 Confidence Interval

This subsection tackles the issue of the confidence intervals attached to the three probabilities. Uncertainties come from the variables p_0, p_1, p_2 and p_3. The goal turns out to find the probability laws for the observed numbers P_0, P_1 , P_2 and P_3, which are the number of the four outcomes. The probability distribution function of the variable vector $x = (x_0, x_1, x_2, x_3)$ related to these observations after the realization of n independent trials follows a Dirichlet distribution of parameter $P = (P_0, P_1, P_2, P_3)$ and n , such as $P_0 + P_1 + P_2 + P_3 = n$.

Thus, $P(x_0,\, x_1,\, x_2,\, x_3|P_0,\, P_1,\, P_2,\, P_3) = \frac{1}{Z(P_0,\, P_1,\, P_2,\, P_3)} \prod_{k=0}^{3} x_i^{P_i-1}$

With $Z(P_0,\, P_1,\, P_2,\, P_3)$ a constant of normalization

The important point given by the theory is that the random variable x_i is a beta law, such as:

$$x_i \sim Beta(P_i, n - P_i).$$

It induces that:

$$E(x_i) = \frac{\alpha_i}{n} = p_i \text{ and } Var(x_i) = \frac{\alpha_i(n-\alpha_i)}{n^2(n+1)} = p_i(1-p_i)/(n+1)$$

The bias of each outcome p_i is inversely proportional to the square root of the length of the measurements n. To get the lower and upper bound of x, y, z we have to frame $p_i + p_j$ accordingly to the expression of the probabilities written in formula 3.2.

Moreover, $x_0 + x_1$ is a Beta law again as the sum of two Beta laws;

$$x_0 + x_1 \sim Beta(P_0 + P_1, n - P_0 + P_1)$$

. For a lot of measurements meaning n large, the Beta distribution could be approximated quite well by a normal distribution such as

$$\begin{aligned} x_0 + x_1 &\sim \mathcal{N}(p_0 + p_1, \sqrt{\frac{(n - P_0 + P_1)(P_0 + P_1)}{n^2(1+n)}}). \\ &= \mathcal{N}(p_0 + p_1, \sqrt{\frac{(1 - p_0 + p_1)(p_0 + p_1)}{(1+n)}}) \end{aligned}$$

This approximation enables to get quickly confidence intervals since the normal distribution provides easy formulas depending only to the mean and bias.
In this manner it could be written, with a confidence of 95%;

$$x_0 + x_1 \in p_0 + p_1 \pm 1.96.\sigma_{01} = p_0 + p_1 \pm 1.96.\sqrt{\frac{(p_2 + p_3)(p_0 + p_1)}{n+1}}$$

Identically, $x_0 + x_2$ and $x_0 + x_3$ can be bounded in order to obtain the lower and upper limits of the probabilities x, y and z. It can be stated a square root convergence of the boundary intervals towards the computed probabilities denoted with the subscript c.
Indeed, a limited development of the limits of the presence probability, $p(b)$ (noted x at the beginning), gives when n is large enough:

$$x \in x_c \pm \Delta_{x,n}$$

Where:

x_c the value computed directly from equation 3.2.

$\Delta_{x,n}$ the boundary such as :

$$\begin{aligned} \Delta_{x,n} &= \frac{1.96}{2}(-x_c + \frac{1}{2})(\sqrt{\frac{1}{4a^2} - 1} + \sqrt{\frac{1}{4b^2} - 1} + \sqrt{\frac{1}{4c^2} - 1}).\frac{1}{\sqrt{n}}) \\ &= \frac{1.96}{2}C(a,b,c).(-x_c + \frac{1}{2}).\frac{1}{\sqrt{n}} \end{aligned}$$

$a = p_0 + p_1 - \frac{1}{2}$

$b = p_0 + p_2 - \frac{1}{2}$

$c = p_0 + p_3 - \frac{1}{2}$

$C(a,b,c) = \sqrt{\frac{1}{4a^2} - 1} + \sqrt{\frac{1}{4b^2} - 1} + \sqrt{\frac{1}{4c^2} - 1}$

With a similar computation, we can get likewise the boundaries on the probabilities to say the truth for each sensor:

$$\Delta_{y,n} = \frac{1.96}{2}C(a,b,c)(y_c - \frac{1}{2}).\frac{1}{\sqrt{n}} \tag{3.3}$$

$$\Delta_{z,n} = \frac{1.96}{2}C(a,b,c)(z_c - \frac{1}{2}).\frac{1}{\sqrt{n}}$$

It is interesting to observe that those limits increase with the value of the computed probabilities. More a sensor tells lies, more its probability to lie is precisely known. Finally, it is very easy to get those boundaries, only the direct

computation of the constant C is required for all of them.

For example, with $x = 0.2$, $y = 0.7$ and $z = 0.8$, figure 3.3 depicted the convergence speed towards those hidden probabilities.

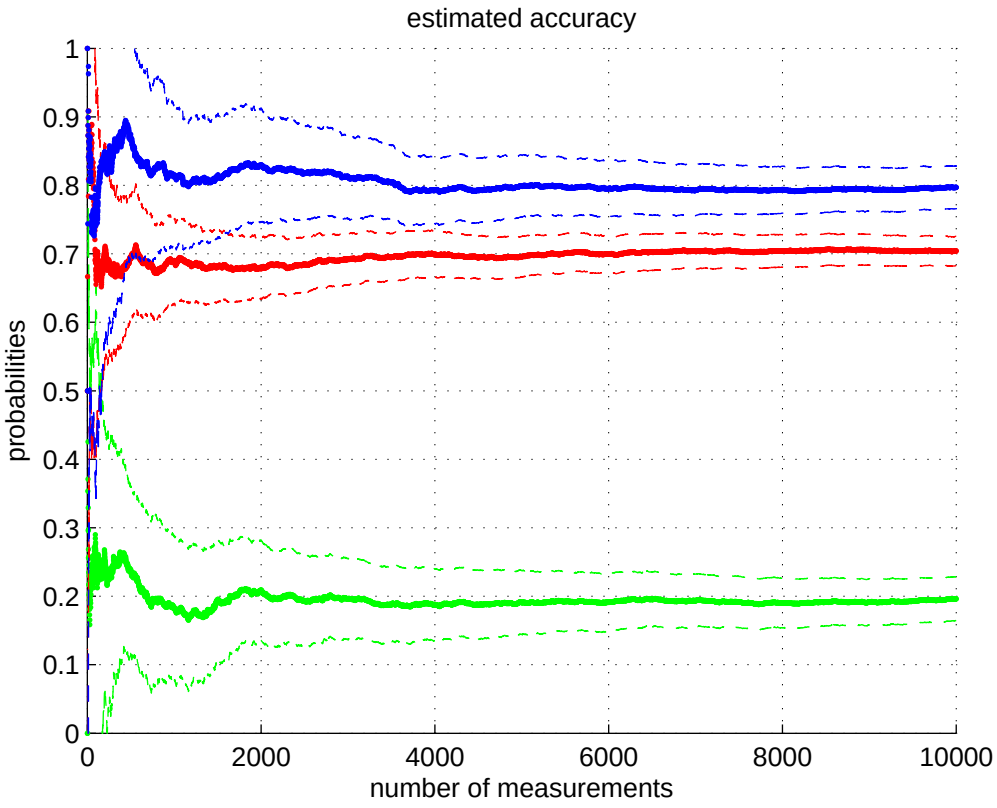

Figure 3.3: Convergence

After 2500 measurements (a 50×50 frame or occupancy grid with the same size), the results are equal to the computed solution given by 3.2 bounded at 95% of confidence by roughly ± 0.05.

3.1.5 Recovering of the true set

After the discovering those probabilities, the Bayes theorem can be applied to get the posterior probability to have a target. Let us call by:

B the event: the target is present on the current spot ; $b = 1$.

M the three current measurements $m_1 = \delta_1, m_2 = \delta_2$ with δ_i the Kronecker symbol , which is worth one or zero.

The Bayes theorem provides:

$$P(B|M) = \frac{P(M|B).P(B)}{P(M|B).P(B) + P(M|\bar{B}).P(\bar{B})}$$

With , according to equation 3.1.

$$\begin{aligned} P(M|B) &= P(m_1 = \delta_1, m_2 = \delta_2|b = 1) \\ &= P(e_1 = 1 - \delta_1, e_2 = 1 - \delta_2) \end{aligned}$$

As the errors are independent one another:

$P(M|B) = P(e_1 = 1 - \delta_1).P(e = 1 - \delta_2)$

For instance, to compute the probability to have a target on the current spot or on a pixel in 2 dimensions, with $m_1 = 1$ for the first sensor and $m_2 = 0$ for the second one, we would write:

$$P(B|M) = \frac{(1-y)z.x}{(1-y)z.x + y(1-z).(1-x)}$$

With x, y, z the probabilities already computed
The recover set is so made up with four different values which correspond to the four outcomes possible of table 3.1.

3.2 Results

Let us apply those results to the problem of nest counting. Two observers plan in advance to visit 500 point counts -a point count is a tree for instance- and report their results independently. To simplify the problem, we assume that there is at the most one nest by tree. The goal is twofold; get both the detection probabilities, the number of nests and their locations. The number of nest found by the two assets are summed up in 3.2.

	observer 1	observer 2
number of nest counted	178	120

Table 3.2: number of nest counted

By using the previous analysis, it is possible to obtain the estimate number of nests, and the detection probabilities P_D of the two assets, recapped in table 3.3.

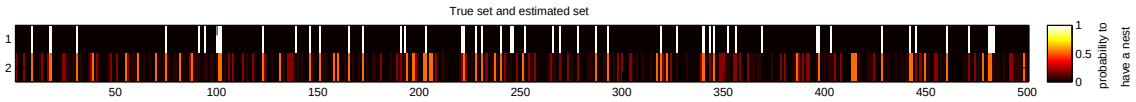

Figure 3.4: Map recovery by two sets of noisy measurement

The result are given with a confidence of 68% this time. We have to replace in formula 3.3 1.96 by 1.

	number of nests	P_{D1}	P_{D2}
estimate	52 ± 38	0.81 ± 0.06	0.7 ± 0.04
true unknown value	50	0.8	0.7

Table 3.3: Results given with 68% of confidence

Figure 3.4 shows the result in the case of 1D measurements, after performing the Bayes theorem and computing the three probabilities.

3.3 Further work

Two more general problems could be considered. The first one is the case with two different detectors focus on the same set, made up with objects with two features. Let us imagine a set with filled or empty triangles and circles as figure 3.5 shows it. The sensor 1 detects the circles and the sensor 2 the filled shapes. We have their measurements m_1 and m_2 . The goal is to know with the measurements the number of filled circles. With absence of errors, i.e. if the sensors were perfect, the number of filled circles would be the number of time where m_1 and m_2 are worth 1 simultaneously. With noise, the problem appears more difficult.

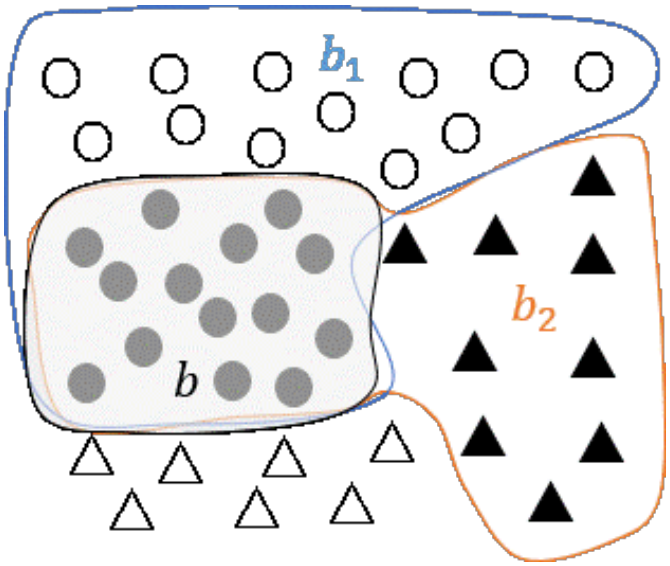

Figure 3.5: True set

What we can do is to use the previous work.

The following notations are both booleans and sets for simplicity, but should be distinct in strict logic

Let us note as on figure 3.5 , b_1 the true set of circle and b_2 the true set of filled shape. For instance, b_1 is equal to 1 for a circle and 0 on the other shapes. We could decompose b_1 as $b_1 = b + d_1$, d_1 being the empty circles set and b the targets that we want to estimate their numbers. We could always write with the same assumption for the errors:

$$\begin{aligned} m_1 &= b_1 \oplus e_1 = b_1.\bar{e_1} + \bar{b_1}.e_1 \\ m_2 &= b_2 \oplus e_2 \end{aligned}$$

Let us focus on the first line:

$$\begin{aligned} m_1 &= b_1 \oplus e_1 = b_1.\bar{e_1} + \bar{b_1}.e_1 \\ &= (b + d_1).\bar{e_1} + \overline{(b + d_1)}.e_1 \\ &= b.\bar{e_1} + (\bar{d_1}.e_1 + d_1.\bar{e_1}).\bar{b} \quad \text{because} \quad d_1 = d_1.\bar{b} \end{aligned} \tag{3.4}$$

The goal is to get something looking like $m_1 = b.\bar{\epsilon_1} + \bar{b}.\epsilon_1$ so as to find a similar expression than 3.1. By noticing that $b.\bar{e_1} = b.\bar{e_1}.\bar{d_1}$ as $b.\bar{d_1} = b$ and that the term $d_1.e_1.b = 0$ since $d_1 \cap b = \emptyset$, we could rearrange 3.4 such as:

$$\begin{aligned} m_1 &= (d_1.e_1 + \bar{d_1}.\bar{e_1}).\bar{e_1} + (\bar{d_1}.e_1 + d_1.\bar{e_1}).\bar{b} \\ &= \overline{d_1 \oplus e_1}.\, b + d_1 \oplus e_1.\, \bar{b} \text{ with} \quad \epsilon_1 = d_1 \oplus e_1 \end{aligned}$$

The similar process can be applied for m_2 and theoretically we find the same wanted expression that in 3.1. The previous statistical approach can be applied to find out $p(b)$, $p(\epsilon_1)$ and $p(\epsilon_2)$.

Unfortunately, as the errors terms ϵ_1 and ϵ_2 depend of b , the results to find $p(b)$ are biased. However, it could be very interesting to solve this problem.

The second problem tackles the issue of the original one by separating the false alarm probability and the miss detection one as two distinct probabilities. For one sensor, it could be expressed as:

$$m_1 = b.\bar{d} + \bar{b}.f$$

d being the miss detection term, and f the false alarm one.

It could be theoretically solvable with three sensors since 8 outcomes are available for 7 unknown variables. Unfortunately, I found no relevant developments . A further work is required.

Chapter 4

SLAM Problem

4.1 Bayesian Filter

4.1.1 Problem formulation

Let us assume that we have a fixed 50×50 pixels window of observation where a pattern is moving in a rigid motion behind it. Its motion is unknown and the observation is noisy accordingly to the previous model 3.1. We collect for example 10 versions shifted and noisy given by figure 4.1 of the pattern depicted in figure 4.2. From only those measurements, the aim is to find out the true unknown pattern and if possible an estimation of its shifting. We assume in addition to the model of the noise, that the shifting is only translations along the two main axis. The unknown probabilities of false alarm, p_{fa} equal in our model with the non detection probabilities p_{nd}, vary between 0.25 and 0.45 randomly in the next simulation.

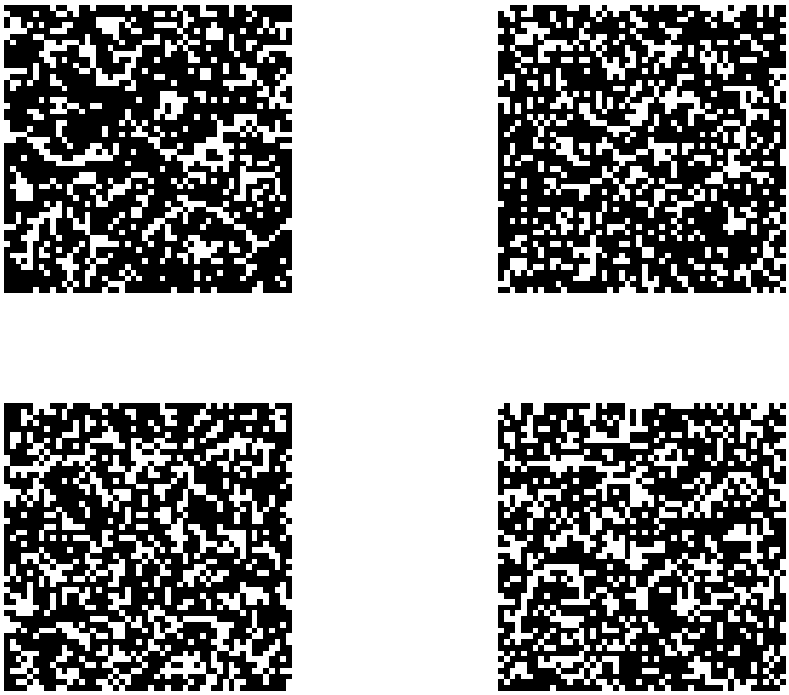

Figure 4.1: 4 observations of 50×50 pixels frames

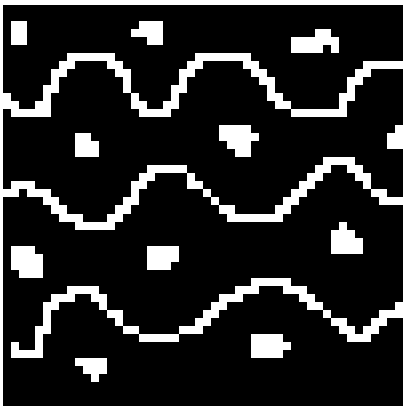

Figure 4.2: True unknown pattern of 50×50 pixels

4.1.2 Resolution

A kind of Bayesian filter was implemented and this section presents its working. The pseudo algorithm is given in 4.1.

Algorithm 4.1 Recursive filter pseudo-algorithm

```
%initialization
create from the true pattern the measurement frames noisy and shifted,
in randomly way
frame_posterior(1)=measurement_frame(1)
for t in 2:time_end
   % with the cross-correlation tool, get the best translation vector
 - T(t)=get_translation(frame_posterior(t),measurement_frame(t))
  % shift the current measurement frame to the reference
 - measurement_frame(t)=calibrate(measurement_frame(t),T(t))

  % with the tool developed in the previous part
 - P=get_probabilities( measurement_frame(t-1),measurement_frame(t)))
  % Build by the Bayes theorem the posterior map
 - frame_posterior(t)=recover_pattern(measurement_frame(1:t),P)
end loop
```

This filter is recursive, takes as input the current measurement frame and outputs a posterior map of probability estimating the pattern's shape. At each time step, the new measurement frame is tested at several positions by comparing it with the posterior map. the best fitting is got whenever the cross-correlation is maximum between those latter frames. The relative translation is saved and the current measurement frame is used to update the posterior frame. At this stage, there is two possibilities compared after. To fuse the new measurement with the posterior map, we could naively sum pixels by pixels these two maps, or use the previous mathematical part. Indeed, it enables to get the probability of the false alarm of the current measurement compared to another measurement, the previous one for instance. The probably to have a bit at one, denoted x in the previous work is determined as well. The Bayes theorem provides then the updated posterior map.
This process is recursive and ends at the last measurement frame. By this approach, we are able to build up an estimation of the searched pattern. A false fitting between the posterior frame and the current measurement one gives a false approximation of the pattern's shifting but not affect so much the next iterations. Although the posterior map is a bit deteriorated, the translation found with a new measurement frame is relative to the posterior map. Thus, at the end, the estimated pattern could be well defined but with a residual displacement.

4.1.3 Results

The two methods are compared in this subsection. They vary only from the update of the posterior and give different results. Only the last posterior maps are compared and we focus on the shifting estimation later. It could be observed that the

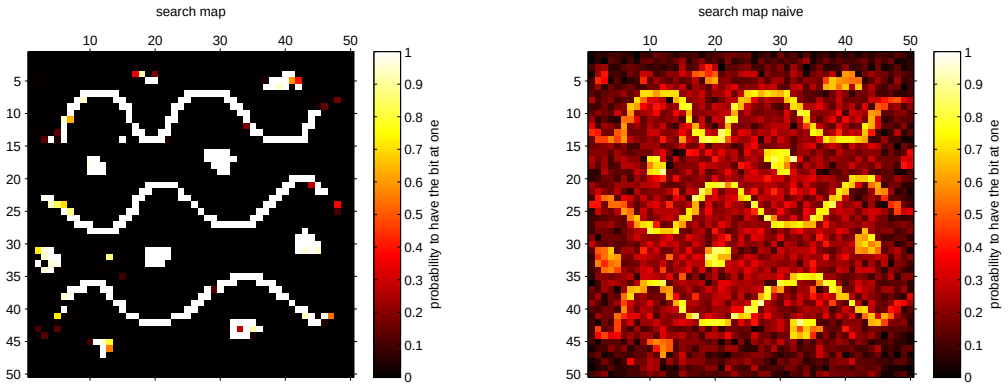

Figure 4.3: Posterior maps

naive solution has a poor probability contrast compared to the novel one. Moreover, simulations preformed with false alarm rate, randomly distributed between 0.25 and 0.4, give badly pattern restitution. However, just one measurement very accurate ($p_{fa} < 0.1$ for instance) among a lot of poor ones could be detected as a good one and its contribution would improve drastically the posterior map. The same case would not occur with the naive approach and this good measurement would be drowned by the others.

4.1.4 Limits

Firstly, with a high false alarm rate, greater than 0.4, the pattern is poorly defined and sometimes totally false. Moreover, the edges of the estimated pattern is removed. Indeed, the filter stacks the posterior map with a new measurement frame, corresponding to the shifted pattern with two black borders, as figure 4.4 depicts it. Over the recovery step by the Bayes Theorem, this area in black distorts the updated posterior map. By this way, the information on the edges of the final posterior map is lost. This problem leads to think about simultaneous localization and mapping.issues that next section deals with.

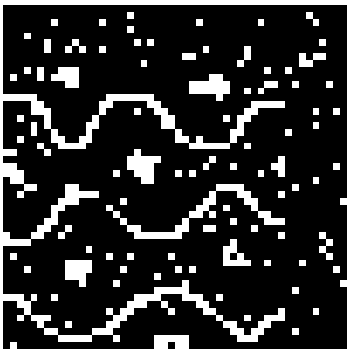

Figure 4.4: Loss of information at the borders

4.2 Simultaneous localization and mapping filter

A similar filter was implemented in this part but this time with a new recursive estimate; the position. Let us consider the problem in a different way, a platform moves randomly onto a large map where a binary pattern is drawn. The only way that the platform owns to localize itself is a low-cost camera. After processing, the platform has to be able to estimate both its position and the pattern. The two main assumptions rest on the motion of the platform. The first one is about the translation movement into a plan parallel to the ground. The second hypothesis assumes that between two iterations, the shifting of the platform is bounded between$[-T_{max}, T_{max}]$ for the two space directions.

4.2.1 Principle

The principle of this filter will be explained by taking a simple example with a very little source of uncertainty regarding the vision of the platform. The probability to reverse bits for each measurements is fixed between 0 and 0.01 in this subsection, so as to explain better how the posterior map of the platform's world is built and how the positions could be found. Let us take in a standard example with a 150×150 pixels world map representing the alphabet, depicted in figure 4.5. The platform starts from the top corner (above the letter A) and gets the first measurement frame M_1 shown in figure 4.6.

Figure 4.6: First and second measurement M_1 *and* M_2

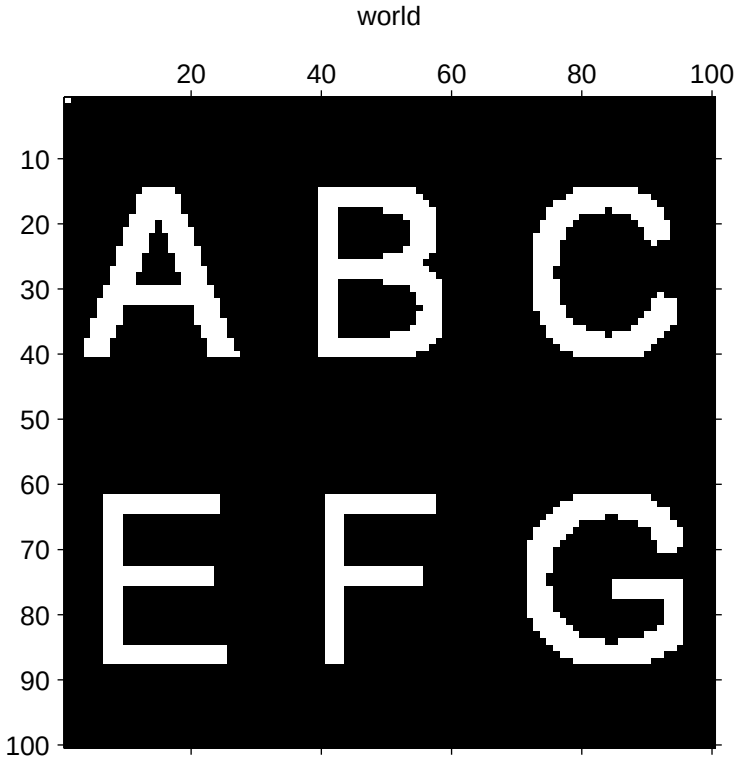

Figure 4.5: True map

At this stage, the posterior map P_1 is made up only with this first measurement, but integrated in a larger map with 0 as figure 4.7 shows us. Indeed the posterior map is a map of probability to have pixels at 1, and 0 indicates a total incertitude, i.e. no measurement. For convenience, The initial point X^0_{est} is used

to create the posterior map since it is difficult to expand dynamically a map on Matlab. With a new measurement frame M_2 represented in figure 4.6, the objective is to find the translation between the posterior map P_1 and M_2 thanks to the cross-correlation tool. As these two images do not have the same size, we have to either expand the small one (M_2) or crop the large one (P_1). To get the process quicker, the posterior map P_1 is cropped around X^0_{est} inside the cross-correlation step so as to find the best fitting. Thus, $(2T_{max})^2$ configurations are tested to find the best relative translation T^1_{est} between those two frames. In our example it is $[1, 3]$. At this stage, the position can be updated; $X^1_{est} = X^0_{est} + T^1_{est}$.

Next, let us applied the function outputting the probabilities of false detection and to have a a pixel at 1 between two consecutive measurements frames M_1 and the current one M_2. As these two frames do not share exactly the same pattern background, both of them have to be cropped accordingly to the translation T^1_{est} linking them each other. After this step, we have p^1_{fa}, p^2_{fa} and p^{12}_b the probability to have a pixel at one in he common part of M_1 and M_2.

The last step is the Bayes theorem application to assess the probability to have a one for each pixel knowing the measurements. This step provides us an update P_2 of the posterior map P_1. This step is semi-iterative as the next section shows it.

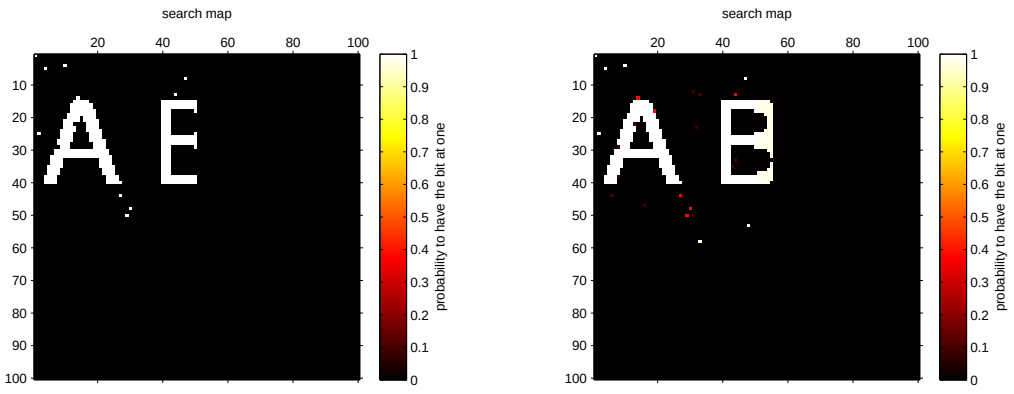

Figure 4.7: Posterior map 1 and 2

The principle of this filter may be as well summarized by graph 4.8.

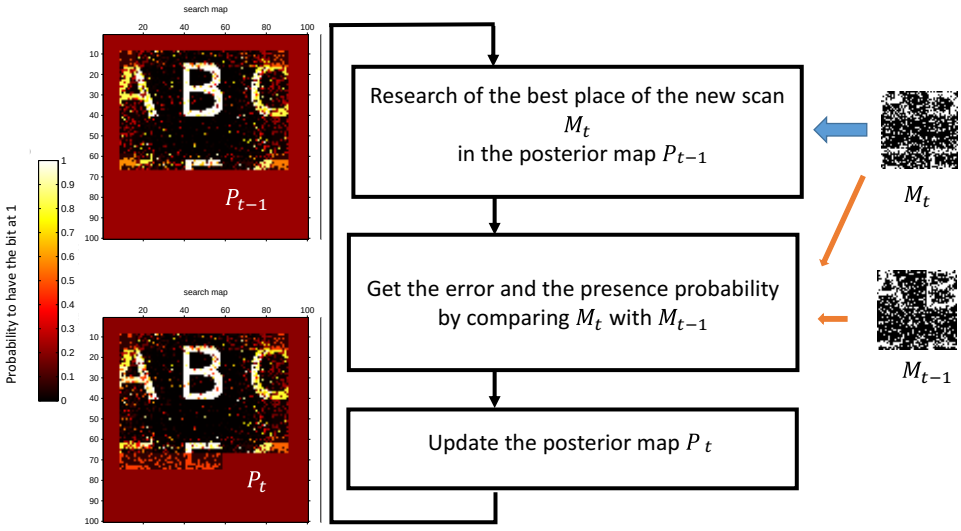

Figure 4.8: Algorithm explanation

4.2.2 Formal description

Let us denote the useful objects to explain formally how the filter works

P The matrix of the entire world map where the pattern is drawn. $P \in \mathcal{M}_{L,W}(\mathbb{R})$
L and W are the length and the width of the world

$\mathbf{P^t_{est}}$ The matrix representing the posterior probability at the time t to have a pixel at one. This is the estimate of the world of the platform

$\mathbf{M^t}$ The binary measurement frame at the time t. $M^t \in \mathcal{M}_{l,w}(\{0,1\})$. This is version of P, cropped and noisy.
l and w are the size of the measurement window

$\mathbf{X^t_{est}}$ The estimate position of the the top left corner of the measurement frame M_t at the time t. It is important to note that its coordinates are integers, varying between $[1, L] \times [1, W]$. X^0_{est} is the reference point, equal to the true one X^0

$\mathbf{T^t_{est}}$ The estimate translation performed by the platform between the time $t-1$ and t. A strong assumption is that $||T_{est}|| < ||T_{max}||$. With T_{max} is the maximum actual shifting between two consecutive time steps. Indeed, two consecutive measurement frames have to be at least 1000 pixels in common to apply the following method.

$\mathbf{p^t_{fa}}$ The false alarm probability of the measurement frame M_t

$\mathbf{p^t_b}$ The probability to have a pixel at one of the measurement frame

- *Initialization*

P^0_{est} is the matrix M_o completed with zeros elsewhere. The submatrix M_0 is positioned inside P^0_{est} by its first element (Top left corner) at the index $X^0_{est} = X^0$.

- *New measurement frame, search of the best translation*

A new scan M^t arrives. The goal is to find the best fit with P^{t-1}_{est} by using the cross-correlation tool. We are dealing with well-known problem in template matching. Several submatrix $P_{sub} \in \mathcal{M}_{l,w}(\mathbb{R})$ of P^{t-1}_{est} are tested, whose its first elements (the top left corner) are chosen around the index in the area of $X^{t-1}_{est} + T$ of the whole matrix P^{t-1}_{est},with $||T|| < ||T_{max}||$.
We have to precise that the area in the posterior map not explored yet, i.e. no measurement available so far, was set to p^{t-1}_b for the cross correlation step. Keeping the non explored area of the posterior map to 0 would have biased the cross-correlation result and would have driven it towards favorable matchings inside the explored area.

We find T^t_{est} such as $T^t_{est} = \underset{||T||<||T_{max}||}{\arg\max}\ (M^t(T).P_{sub})$

With $M^t.P_{sub}$ the matrix dot product multiplying elements by elements before summing. This is cross correlation between the images M^t shifted of T and P_{sub}

We could at this stage update the position estimate of the platform, referencing the position of M^t in P^{t-1}_{est}. $X^t_{est} = X^t_{est} + T^t_{est}$.

- *Getting the probabilities p^t_{fa} and p^t_b of the scan M^t*

The previous work in the part 3.1 is used to get those three probabilities. The previous scan M^{t-1} serves to compare with the new one. As those scans don't measure the same same area (there is a translation between them) the estimate T^t_{est} enables to crop them to stack them onto their common parts.
If these found probabilities are greater than $1 + \Delta$ or lower than $0 - \Delta$, Δ being their joint incertitude, T^t_{est} is rejected and the cross correlation step is again done.

- *Update of the posterior map P^t_{est}*

Only the submatrix of P^t_{est} referenced by the index of its first element (the top left corner) ,which is X^t_{est}, will be updated. Indeed, as no new measurements have been done elsewhere over the time t, the posterior probability remains the same

out of the span of M^t.

Let us take a measured pixel $m \in \{0,1\}$ in P_{est}^{t-1}, in the domain of M^t, i.e. where M^t has an influence within the computation of the posterior probability map P_{est}^t. Let us denote by m^{t-1} the array containing all the measured pixels already saved at this spot and m^t the same one with m at the end of the array. If the spot has never been measured, the array is just m and enriches P_{est}^t with a new value. Let us call by $b \in \{0,1\}$ the real value of that spot. The prior probably $P(b=1)$ has been computed in the previous step and is worth p_b^t.

The Bayes Theorem gives:

$$\begin{aligned} P_{est}^t(b &= 1) = \frac{P(m^t|b=1).P(b=1)}{P(m^t|b=1).P(b=1) + P(\bar{m}^t).P(b=0)} \\ &= \frac{P(m^t|b).p_b^t.}{P(m^t|b).p_b^t. + P(\bar{m}^t|b).(1-p_b^t).} \end{aligned}$$

$P_{est}^t(b=1)$ is not recursive but the likelihoods $P(m^t|b)$ and $P(\bar{m}^t|b)$ are, as well as p_b^t. Indeed:

As m is independent from the other ones in m^{t-1}

$$\begin{aligned} P(m^t|b) &= P(m^{t-1}|b).P(m|b) \\ &= \begin{cases} P(m^t|b) = P(m^{t-1}|b).P(m|b) = P(m^{t-1}|b).p_{fa}^t & \text{if } m=0 \\ P(m^t|b) = P(m^{t-1}|b).P(m|b) = P(m^{t-1}|b).(1-p_{fa}^t) & \text{if } m=1 \end{cases} \end{aligned}$$

and likewise;

$$\begin{aligned} P(\bar{m}^t|b) &= P(\bar{m}^{t-1}|b).P(\overline{m}|b) \\ &= \begin{cases} P(m|b) = P(\bar{m}^{t-1}|b).P(\bar{m}|b) = P(\bar{m}^{t-1}|b).p_{fa}^t & \text{if } m=1 \\ P(m|b) = P(\bar{m}^{t-1}|b).P(\bar{m}|b) = P(\bar{m}^{t-1}|b).(1-p_{fa}^t) & \text{if } m=0 \end{cases} \end{aligned}$$

For p_b^t, the average all of the p_b computed is taken pixel by pixel as well.

4.2.3 applications

4.2.3.1 standard example

Let us take again an alphabet pattern similar to figure 4.2 but larger. For the next simulation the unknown probabilities of false alarm p_{fa}^t are randomly distributed

Figure 4.9: The first four measurement scans

between 0.1 and 0.3 and the measurement window is a 50×50 pixels frame. An example of the measurement frames received by the platform are depicted by figure 4.9. As it could be noticed, it is difficult to follow the trajectory of the platform, even if it was supposed to be pure translations. The posterior map got after 75 time increments is shown with figure 4.10. Figure 4.11 sums up the estimation of the platform's positions. At the edges there are only few measurements scanning the pattern so the posterior map is less precise. The probability contrast is nevertheless very good.

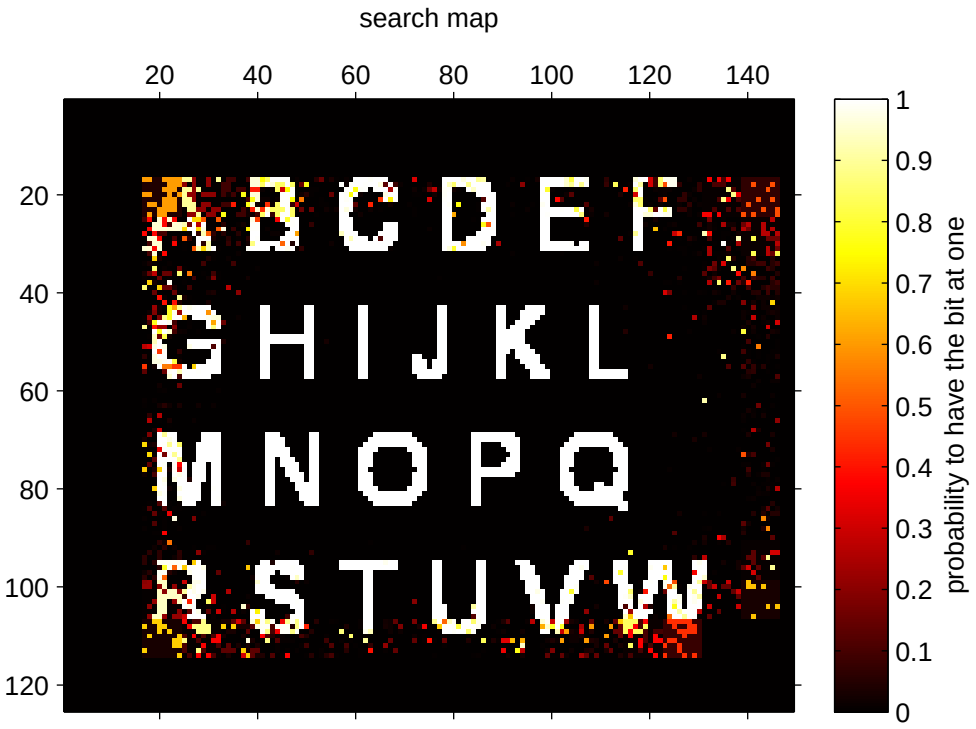

Figure 4.10: Posterior map of the standard example

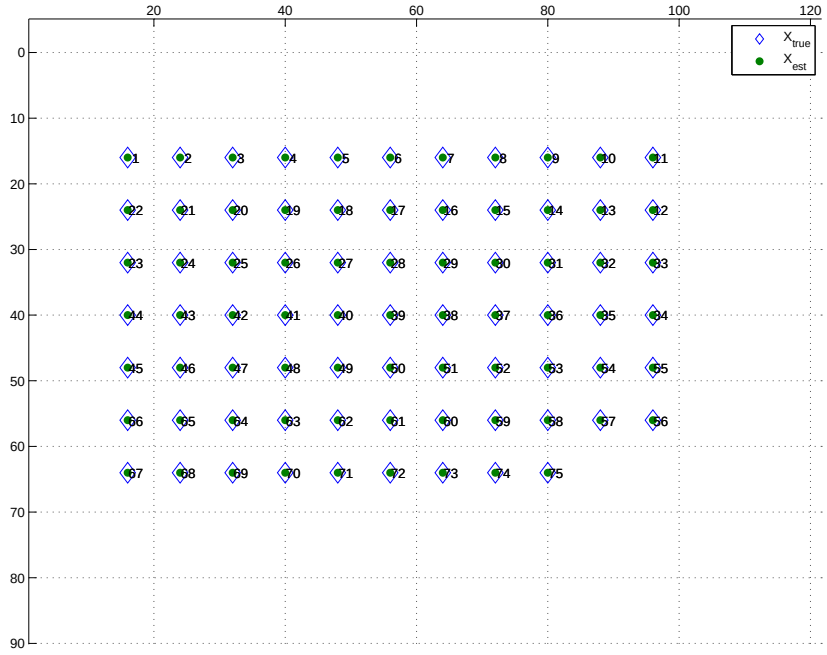

Figure 4.11: Estimated positions

4.2.3.2 Sparse, disorganized environment

So far organized and compact patterns were used. Let us try now with a sparse and desorganized background. For instance, the platform captures bright points as stars or light spots coming from a city seen far from the ground. Figure 4.12 illustrates this class of pattern. As previously, we get only small size scans noisy and shifted in a random way. The unknown path of the platform is now a spiral from the exterior to the center. For space applications whose the luminous spots are stars, assumed to be at an infinite distance from the platform, the translation movement seen between two measurement scans can be interpreted as self rotation of the platform around the translations axes.

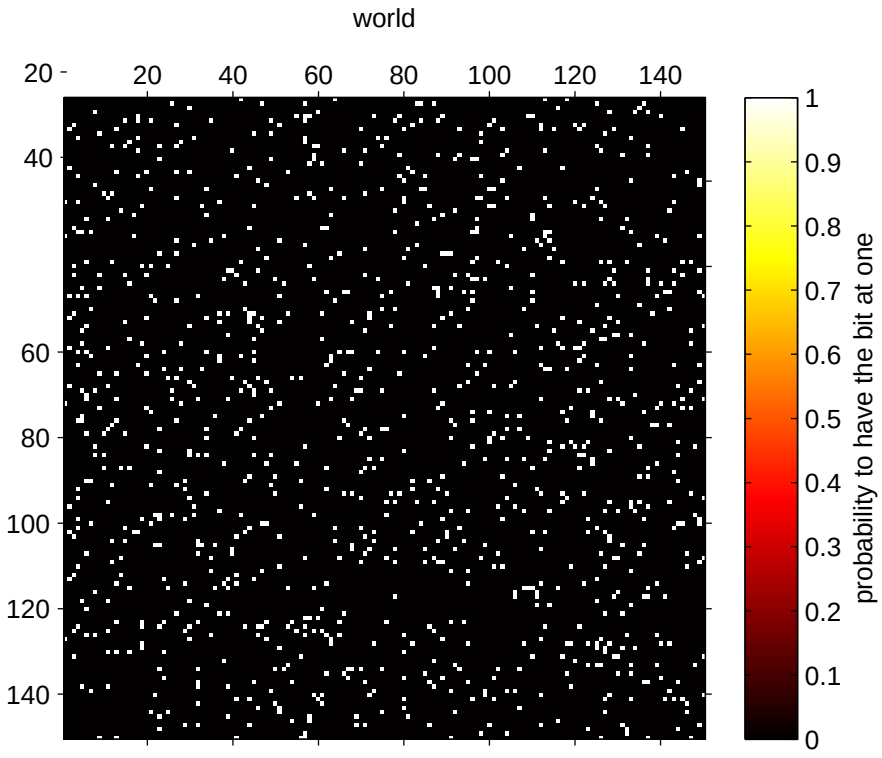

Figure 4.12: Disorganized pattern

For example, we have those sequence of measurements depicted in 4.13, with unknown false alarm rates between 0.2 and 0.4. The estimated position is given in 4.14. Finally, the estimated pattern of the sky or of ground luminous spots is presented in showing the last posterior probability to have a pixel at one.

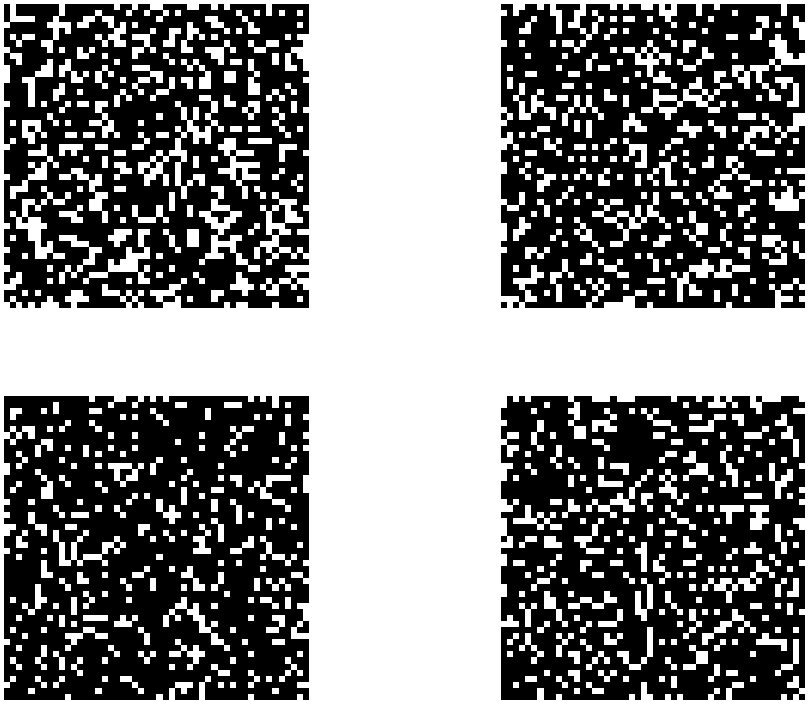

Figure 4.13: Luminous spots

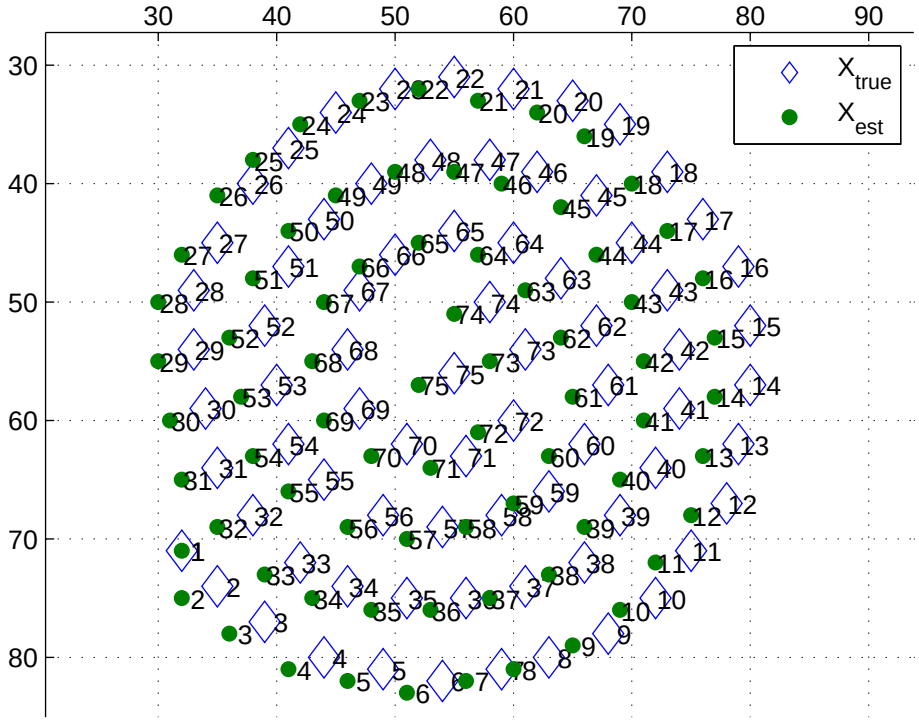

Figure 4.14: Estimated positions

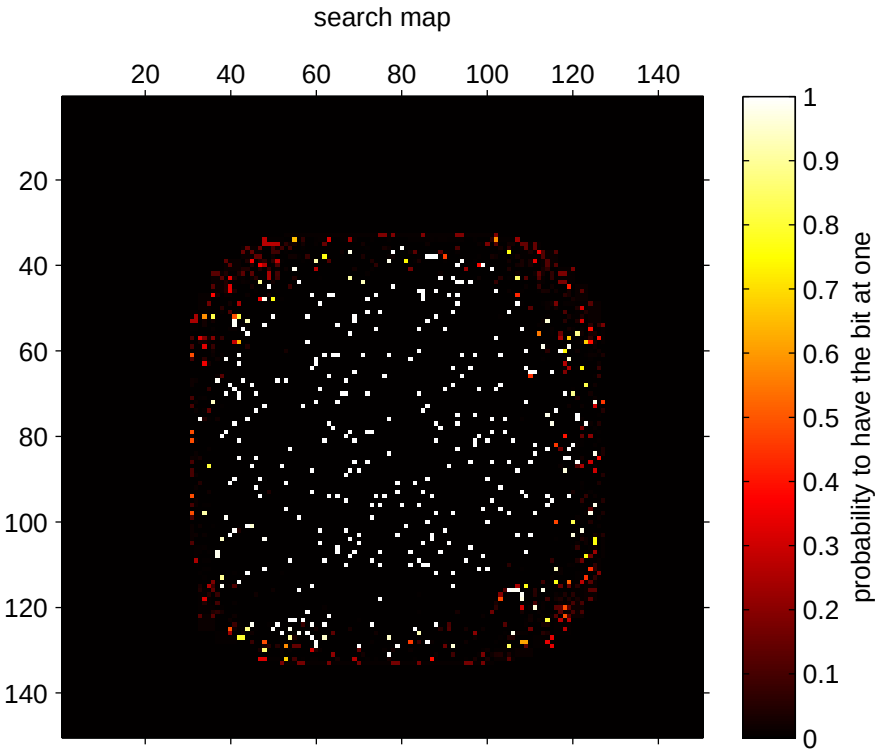

Figure 4.15: Pattern recovering

The correlation between the true and the estimated pattern is 0.88 and the offset, i.e the residual shift is $[-3; 1]$ pixels.

4.2.3.3 Edge imaging

In some applications, it may be interesting to use skeleton images rather than the original one to decrease the number of data to process. Indeed, this quick image processing routine emphasizes geometrical and topological properties of the object shape. Let us apply this example trough the following scenario. A drone flies around a village presented in 4.16. A current scan is shown with 4.17 and with its edge version. The drone flies at a constant altitude, let us say 20 meters with a low-cost camera gimbal pointing towards the ground. There is no GPS but there is a magnetometer to calibrate all camera scans along the same direction. Each scan is a planar representation of the ground with a size of 60×60 pixels, without necessarily an auto focus . The goal is both estimate the path of the drone and construct a map of the environment.

The edge version was performed upon the grayscale version of the color one. Even if the Canny edge detector is supposed to reduce the false positive or false

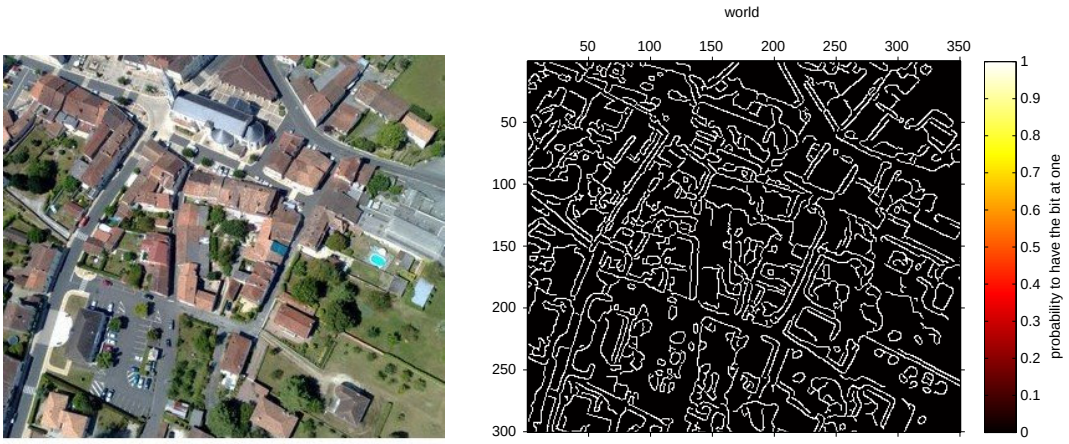

Figure 4.16: True map

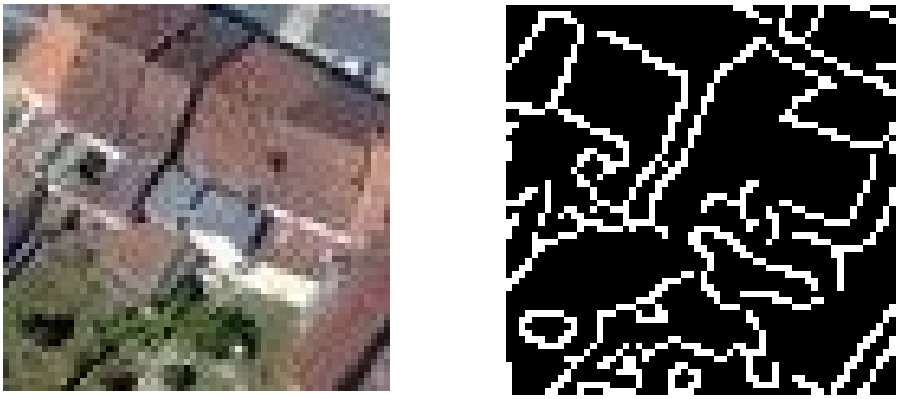

Figure 4.17: A current scan, without noise and its skeleton version

negative edge pixel, two consecutive measurements in our physical scenario will not produce the same edge version frame. It could come from intensity variations, camera and signal processing noise. So for our simulations, as no real measurement frame was at my disposal, the global frame resulting to the edge detection processing in figure 4.16 was used to generate measurement scans. Then, random and unknown noise has been added for each of them. An example of consecutive measurement frame are presented in figure 4.18.
Result regarding estimate position and map recovering are shown with figure 4.19 and 4.20. The unknown path is roughly straight and its estimation has been well recovered. The cross-correlation between the actual pattern end the estimate one is equal to 0.91.

The main advantage of this Bayesian filter is the speed of convergence towards

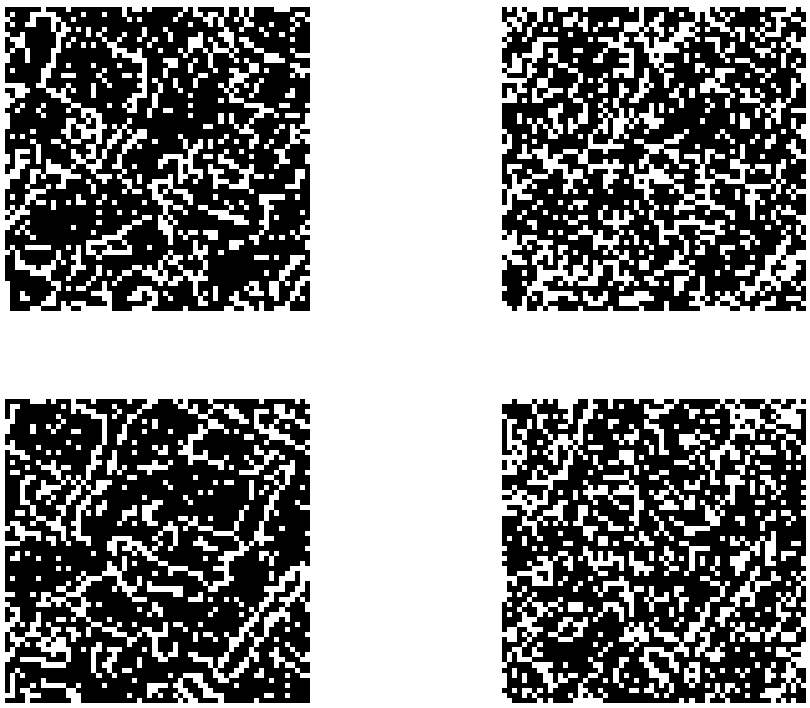

Figure 4.18: Consecutive measurement frames

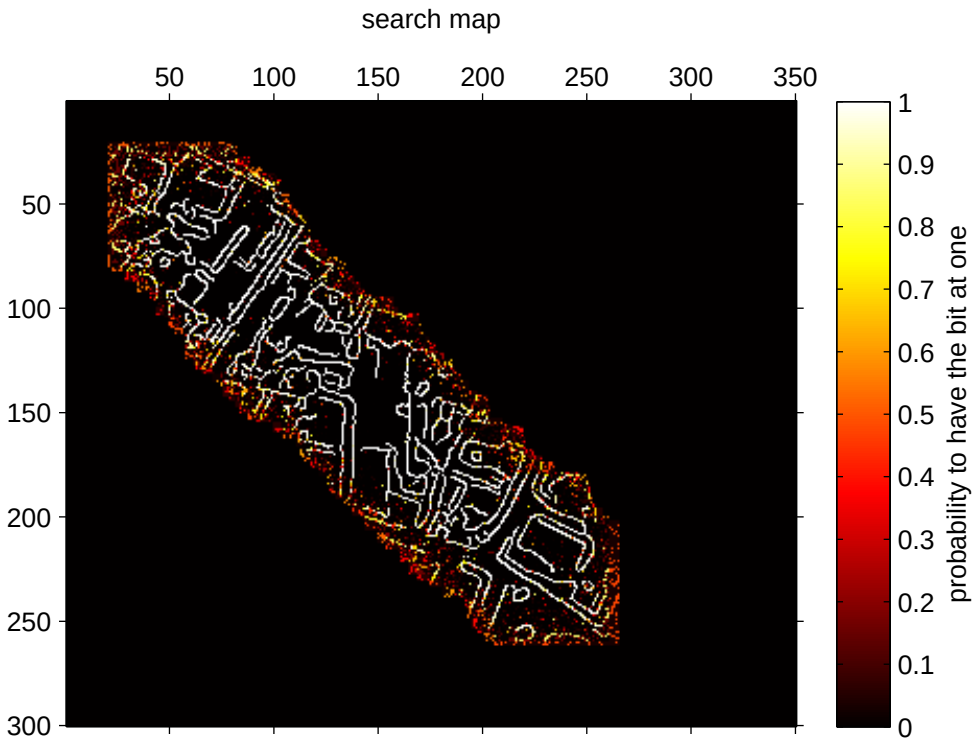

Figure 4.19: Recover map

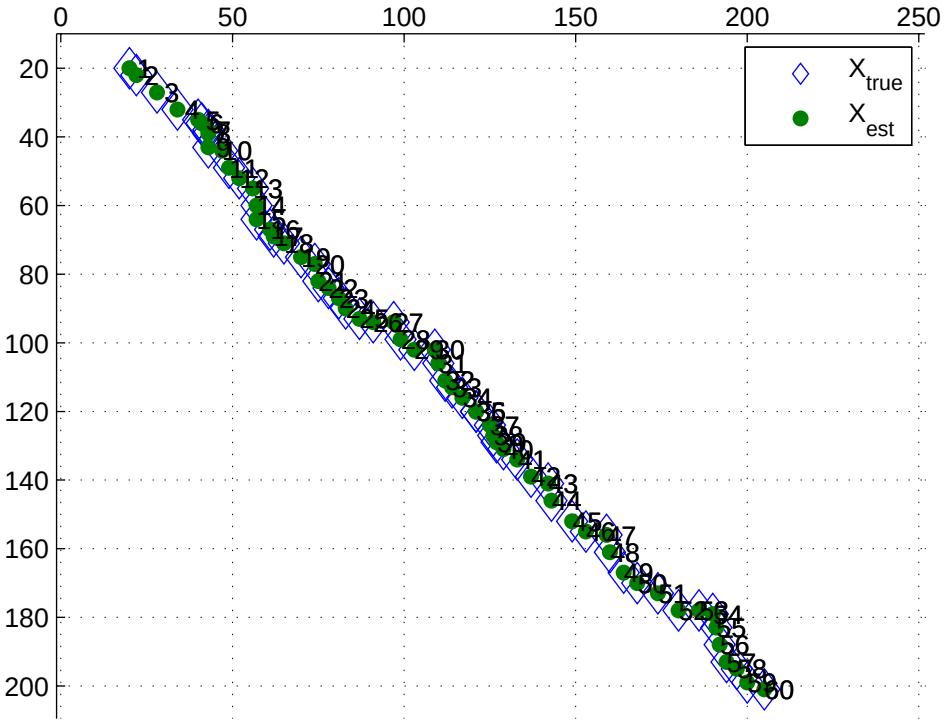

Figure 4.20: Estimate positions

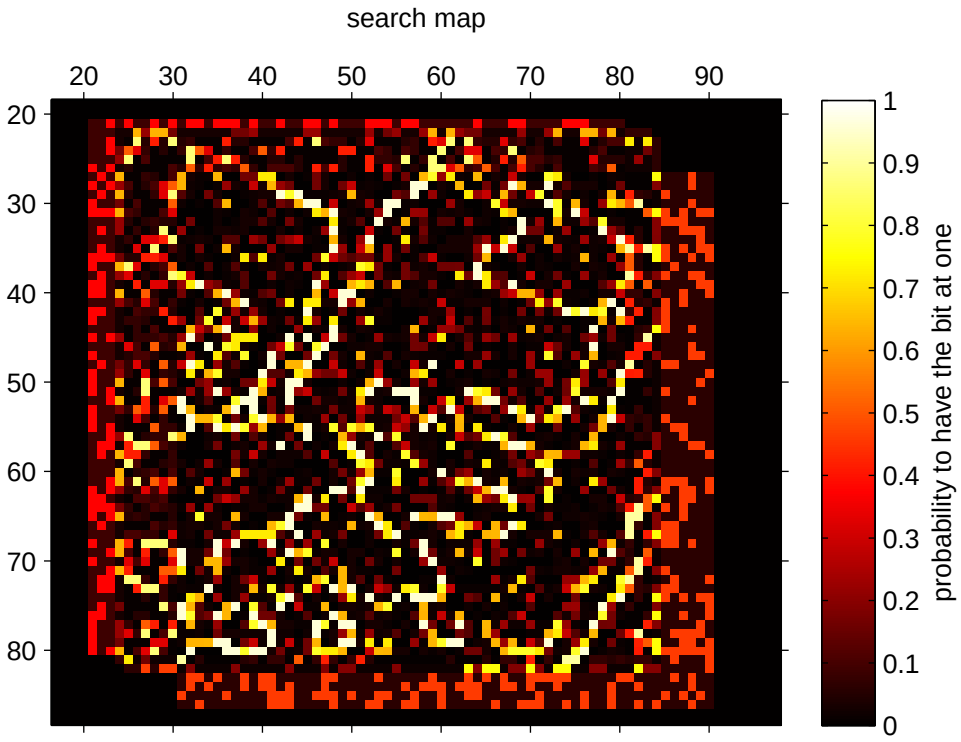

Figure 4.21: Recover map after 4 scans

the true pattern. After the fourth first scans of figure 4.18, the posterior map depicted in figure 4.21 has a good contrast with only the first four measurements. It leads chiefly to a good match with the next scan.

.

4.2.4 Comparisons

This kind of Bayes Filter will be compared with a naive filter for different maps and different false alarm probabilities intervals. The naive filter behaves exactly as the previous one but not compute at each time step the probabilities. After getting the best fitting between a new measurement scan M^t and the posterior map $P^t_{naive\,est}$, the naive filter adds at the supposed right place this scan inside the posterior map. Basically, it is the superposition of all the scans at their estimated place. It means that $P^t_{naive\,est}$ is the sum of all the measured pixel at one. Let us describe the step of the posterior update:

$P^t_{naive\,est}$=$(P^{t-1}_{naive\,est} + \tilde{M}^t)$ with $\tilde{M}^t$ being a matrix of the size of $P^{t-1}_{naive\,est}$ positioned at its estimate place with 0.5 elsewhere.

Actually, everywhere out of the span of M^t, no measurement is provided, and hence fill $\tilde{M}^t$ with 0 would have been a mistake. Indeed, it would have meant that the area not yet explored does not contain pixel at one, perverting the computation of the estimate translation.

Those two filters ran in parallel with two distinct posterior map and position estimate. Figure 4.22 shows the two posterior maps with the aerial edge pattern. We have to note that out of the explored zone, the posterior maps were set at 0 to highlight the area already explored.

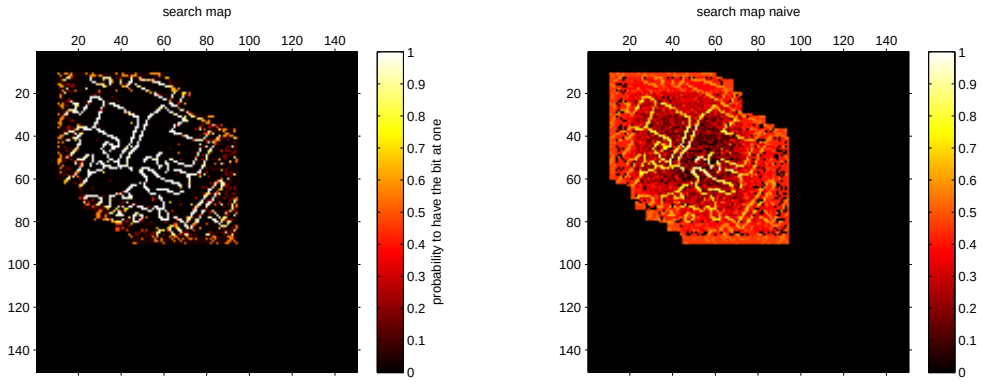

Figure 4.22: comparison of the posterior maps

The correlation between the posteriors map and the true one are recaped in table 4.1. For each column, the interval of the unknown probability of false alarm varies, several intervals were tested. The first approach is much more accurate than the naive one as it can be shown. The naive approach gives satisfying result as long as the false alarm probability is lower than 0.2. Below, the platform is lost and the posterior map is not at all well defined. The limit of the first approach arises for $P_{FA} \in [0.25 - 0.35]$, it provides a posterior map hardly correlated but the position estimate is still well . Below, the platform is lost and neither the posterior map nor the position are correct.
The intersting point regarding the elaborate approach lies on the robustness against a poor measurement. Indeed, as the last column of table 4.1 underlines it, a false alarm rate includes between $[0, 0.4]$ gives accurate results. Basically, a poor measurement scan is detected and does not contribute to the current posterior map computation.

4.2.5 further improvements and limitation

In this precise framework, as at each time step the false alarm of two consecutive measurement scans are computed, we have for each scan -except for the first and

P_{FA}	0	0-0.1	0.1-0.9	0.2-0.3	0.25-0.35	0.3-0.4	0-0.4
$Corr(P, P_{est})$	1	0.99	0.89	0.8	0.65	0.24	0.83
$Corr(P, P_{naive\,est})$	0.84	0.8	0.6	0.4	0.4	0.4	0.43

Table 4.1: Correlations between the true and estimate map

the last one- two false alarm rates. For the scan M^t, we have thus p_{2fa}^{t-1} and p_{1fa}^{t} associated on it. We could get information from their differences. As it was been shown in the previous chapter dedicated to the mathematical approach, after 2500 pixels compared, the false alarm probability is bound between its computed value and $\pm\Delta$. Δ depends on the computed value but is always lower than 0.05. This knowledge can detect a misplacement got by the cross-correlation tool. In fact, if the translation estimated by the cross correlation between M^{t+1} and P_{est}^{t} is not correct, the second value p_{1fa}^{t} related to M^t , computed right after, will be out of the confident interval around p_{2fa}^{t-1}. It means necessarily that the pose of M^{t+1} does not fit with M^t. These overlaid areas of those scans undoubtedly do not reflect the same true pattern. We could then reject the cross correlation measure and find another fitting between $M^{t+1}$and P_{est}^{t}. Whenever that condition is respected, the filter routine can keep on running normally until the next time step.

Similarly, the computation of the probability to have pixel at one p_c^{t+1} can warn about a misplacement of M^{t+1}on P_{est}^{t}. The criterion can be the comparison of $p_c^{t+1}$with p_c^{t} since both of these probabilities have been computed with the common part of M^t and M^{t+1} , i.e the overlaid area.

More generally, as it was underlined several times, this filter works for simple platform movements; translations only. The cross correlation tool could solve rotation and scale effects separtly, but with a computational time much greater. Scale-invariant feature transforms overcome this problem by finding correspondance between two images and so the spatial transformation linking them. In the case of SLAM framework, objects of the environment are sensed by different viewpoints and building a posterior map of the environment may appear useless. Nevertheless, we could imagine update a 3D global posterior map taking in account some strong features of the environment, based on this Bayesian filter. It wi would enable to refine poor measurement detections inside the features extraction process.

Chapter 5

Targets with motion

In this part we tackle the issue of tracking targets with a simple motion supposed to be Brownian. That is to say targets can move either on the left or on the right or stay on its spot with equal probabilities. A one dimension representation has been used rather than a plan representation to make the plots more visual. Indeed, the time is represented and will be used for the third dimension. The only difference between a 1D and 2D problem is that the target can also step back or front. Thus the problem is tracking the targets in a time range with all the posterior distributions given in the mathematical approach part. Even if the problem is not really realistic since the UAVs are not able to get the whole map at each time step, it could be interesting in a first approach to achieve it.

By way of an example, let us imagine three surveillance cameras focus on the same map. A post processing strategy is adopted to detect targets, as features based-extraction for instance. Targets often may be misdetected by the recognition system of each camera since its accuracy is poor or the environment is cluttered. Moreover, this practical problem assumes that each camera post-processing system sends information to a center in charge of fuse it. We focus only on the high level problem, where a global occupancy grid represents the position of potential targets, assumed to be a local pixel.

5.1 Resolution

At each time step, targets move independently one another in a Brownian motion described above. As three measurement scans are available, an adaptation of the model with only two have to be done. Let us note $(x, y, z) \in [0, 1]^3$ the triplets probability of detection, m_i the sequences of measurement and $p(b)$ the probability to have a target. A similar approach than part 3.1 leads:

$$2^3 \text{ equations;} \begin{cases} p(m_1 m_2 m_3) = p(b).xyz + (1-p(b)).(1-x)(1-y)(1-z) & L_1 \\ p(m_1 m_2 \bar{m}_3) = p(b).xy.(1-z) + (1-p(b)).(1-x)(1-y).z & L_2 \\ ... & L_i \\ p(\bar{m}_1 \bar{m}_2 \bar{m}_3) = p(b).(1-x)(1-y)(1-z) + (1-p(b)).xyz & L_8 \end{cases}$$

The left terms can be found by counting the number of occurrences of the 8 outcomes of the sequences (111, 110, 101,....,000).

It may be rearranged by summing L_1 and L_8, L_2 and L_7 and so on. In this way, we get:

$$\begin{cases} p_0 = xyz + (1-x)(1-y)(1-z) \\ p_1 = xy(1-z) + (1-x)(1-y).z \\ p_2 = x.(y-1).z + (1-x)y(1-z) \\ p_3 = (x-1).yz + x(1-y)(1-z) \end{cases}$$

We find the same equations that the case with two sensors and so the resolution does not change. the triplet (x, y, z) may be expressed with p_0, p_1, $p_2 \, and \, p_3$. $p(b)$ can be got by taking the average value in the system of equation 5.1. The posterior probability is got with the Bayes theorem.

At this stage, from the real targets movements, (standby, move on the right or left) given in the next figure 5.1, the posterior distribution, given with figure 5.2. can be extracted at each new time step.

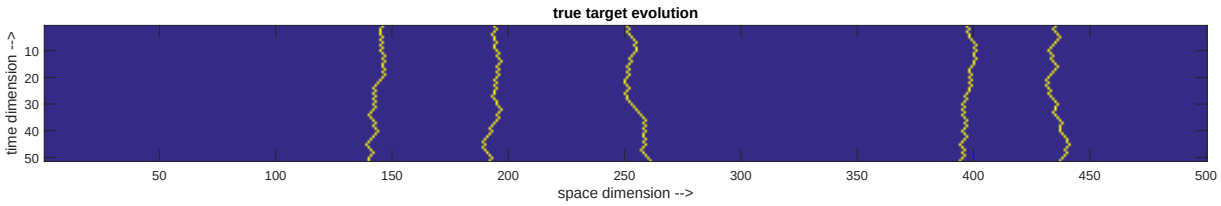

Figure 5.1: true target evolution

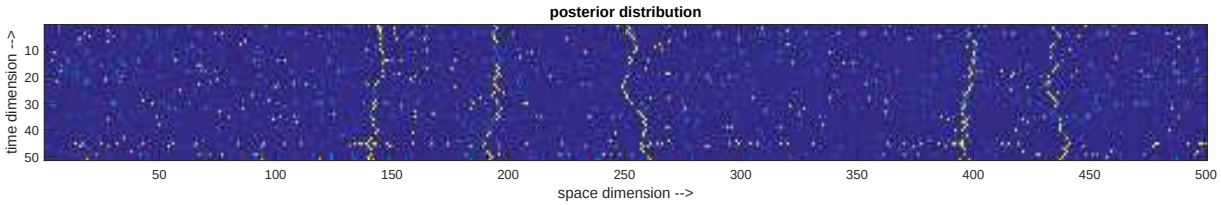

Figure 5.2: posterior distribution

5.2 Particle filter

The tracking could be improved by including the motion model of targets into the likelihood distribution in order to get in real-time the best estimation as possible of target positions. The best way found to achieve it is to use a particle filter.
Its working is simple, at the first time increment, ten particles are laid on each measurements space and its likelihoods are worth the first likelihood distribution corresponding to the running measurement space. The bootstrap method to get the posterior particles distribution was used. The update is got by passing the particles through the motion model (roughly the third of the particles moves on the left, the other on the right and the last one remains on the spot).
The second iteration takes this set of particles position and the same process is applied ; a new likelihood is computed and so on.

Formally, let us denote by:

L The length of the 1D world. This is the number of space spot of the world

n The number of particles

$\mathbf{x_t}$ The particle positions on the x axis , at the time t. $x_t \in [1, L]^n$

$\mathbf{m_t}^\mathbf{i}$ The three measurements over the x axis, $i \in \{1, 2, 3\}$.

$\mathbf{p(m_t}^\mathbf{i}|\ \mathrm{x_t})$ The likelihood at the time t.

$\mathbf{p_t}$ The probability to have a target $p_t(k) = \frac{card(x_t == k)}{card(x_t)}$, $k \in [1, L]$

- Initialization

$\frac{n}{L}$ particles are dropped at each space spot initially so as to spread it uniformly.

- Update

There is no measurement model but the likelihood depends on the measurements such as:

$$\begin{cases} p(m_t^i|x_t) = xyz & \text{if } m_1 = m_2 = m_3 = 1 \\ p(m_t^i|x_t) = xy(1-z) & \text{if } m_1 = m_2 = 1 \text{ and } m_3 = 0 \\ ... & ... \\ p(m_t^i|x_t) = (1-x)(1-y)(1-z) & \text{if } m_1 = m_2 = m_3 = 0 \end{cases}$$

Every particle is associated with a likelihood distribution. There is 8 different possible values.

The classical bootstrap method [11] is used to resample the particles according to the weight of their likelihoods. At this stage we have a new set of particles x_t.

- Prediction

The dynamic model leads to move each particle according to the Brownian movement. Each particle can either stay or move on both sides randomly.
At this stage the probability to have a target for each space spot can be determined by computing p_t.

5.3 Results

In this simulation, 50 time steps were considered with five targets moving in accordance with the previous assumption onto a 500 space locations.

The result is presented on figure 5.3. The probability distribution of having a target is as described before the number of particles on the running spot divided by the total number of particles.

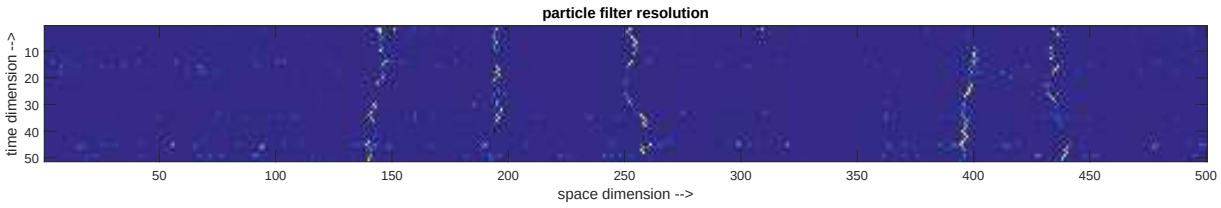

Figure 5.3: Distribution given by the particle filter

The advantage of this filter is to eliminate the poor measurements of the good one.

We can add the assumption that the targets are sometimes destroyed and other may appear and the particle filter is quite robust. Results are presented in next figure 5.4.

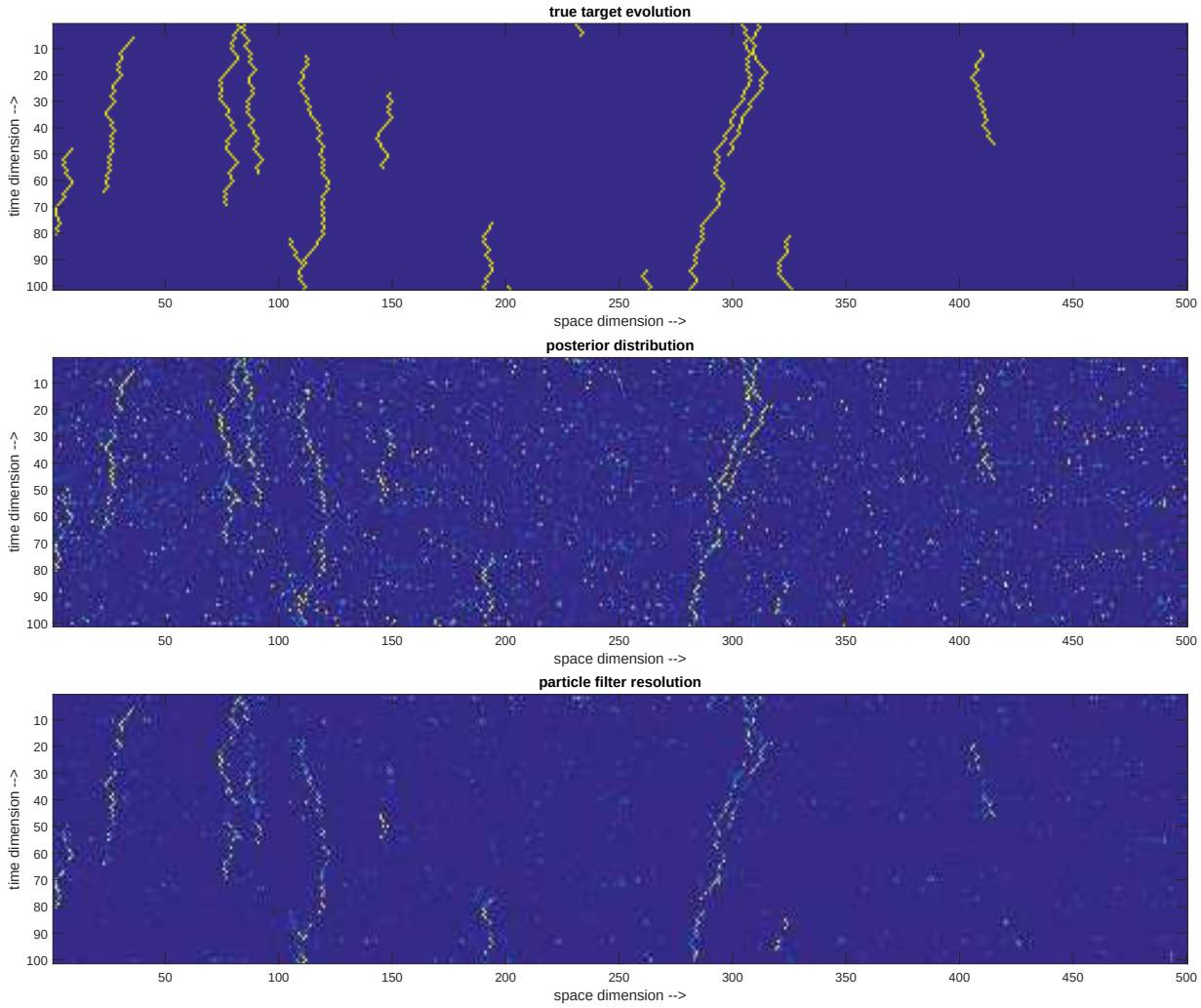

Figure 5.4: Performance of the particle filter

5.4 Discussion

Occlusions are taken in account as being a misdetection. The noise model assumes to have the same rate of false alarm than misdetection for a camera. It means that for 5 targets over 500 free space spots, with a false alarm rate of 20%, there is 100 bits reversed over 500. Statistically, there is 99 false alarms and one occlusion, or misdetection. If all targets are occluded for one camera, it does not matter both for the noise model and for the general tracking process since there are few targets. For more targets, let us say 100, if 50 targets are occluded for one camera and normally visible for the others, the probabilities computation of formula 3.2 gives biased results. Indeed, errors would be constraint to reverse bits at one rather than be spread uniformly, in accordance with the noise model.

The second point no considered is the tracking of two passing each other. Two crossing paths are in this model impossible to track.
It could be interesting in a further work to implement in parallel a multiple hypothesis filter (MHT) [12]. The goal of a such filter is to rule out bad data association in case of conflict, typically when two targets are getting closer. MHT filter needs to have false alarm probabilities and target presence probability, that is provided by 3. After, some hypothesis are build and discarded by pruning non-admissible associations.

Chapter 6

Conclusion

Throughout this Master thesis, several applications in the field of detection theory were proposed. The central point of those applications is the use of an novel way to count the number of errors in two binary measurement sequences sensing the same phenomena. The prior number of one in the true sequence, i.e. the occurrence of the phenomena, is got as well. The binary measurement sequence may come from detection algorithms processing, as locally corner or edge detection, or even global target detection. The natural application of the mathematical chapter 3 in detection theory is vision-based detection since the error made by classifiers depends greatly on the unknown environment and cannot be accurately known in advance. By comparing two simultaneous reports if targets move, or two consecutive reports if the targets are fixed, it is possible to refine the measurement as the two last chapters shown it.

Indeed, the chapter dealing with the SLAM issue offers a method to recover a fixed drawn pattern on the ground with a platform embedding a low-cost camera. Even if the problem was simplified by assuming elementary spatial platform's movements, a successful use of a Bayesian filter enabled to recover both the pattern and the platform position for very high measurement false error rates.

The other issue solved lies in multiple targets tracking. An occupancy grid of potential targets maps a cluttered environment and detection based-vision algorithms provide simultaneous occupancy reports, with time-varying false alarm rates. The question arose was how to fuse these local decisions. The previously presented mathematical approach fires up this matter by giving a likelihood presence used as input of a particle filter. Even though the method is robust for appearing and disappearing targets, it makes any tracking distinctions for crossed targets paths.

In fact, these two sub problems lie within a more general one in computer vision theory, namely the image registration problem. The determination of an optimum

spatial transformation between two set of features belonging to two consecutive frames remains still an open problem, especially under the real time constraint. In Feature-based Registration problem, correspondence between two sets of points is made and the statistical basis proposed in this thesis could improve it.

Last but not least, the mathematical approach proposed in chapter 3.1 suffers from a too strong assumption regarding the error. Indeed, the false alarm probability was equal to the miss detection one. Further works so as to relax this assumption could be very interesting.

References

[1] J. O. Berger, *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[2] J. D. Nichols, J. E. Hines, J. R. Sauer, F. W. Fallon, J. E. Fallon, and P. J. Heglund, "A double-observer approach for estimating detection probability and abundance from point counts," *The Auk*, vol. 117, no. 2, pp. 393–408, 2000.

[3] H. Y. Kim and S. A. De Araújo, "Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast," in *Advances in Image and Video Technology*, pp. 100–113, Springer, 2007.

[4] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[5] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial intelligence*, vol. 128, no. 1, pp. 99–141, 2001.

[6] T. Duckett and U. Nehmzow, "Mobile robot self-localisation using occupancy histograms and a mixture of gaussian location hypotheses," *Robotics and Autonomous Systems*, vol. 34, no. 2, pp. 117–129, 2001.

[7] M. Hasan and M. Abdellatif, "Fast template matching of objects for visual slam," in *Intelligent Robotics and Applications*, pp. 484–493, Springer, 2012.

[8] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular slam," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 932–945, 2008.

[9] V. Salari and I. K. Sethi, "Feature point correspondence in the presence of occlusion," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 1, pp. 87–91, 1990.

[10] D. Chetverikov and J. Verestoy, "Tracking feature points: a new algorithm," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 2, pp. 1436–1438, IEEE, 1998.

[11] J. V. Candy, *Bayesian signal processing: Classical, modern and particle filtering methods*, vol. 54. John Wiley & Sons, 2011.

[12] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, 2004.