

# Non-parametric Bayesian State Space Estimator for Negative Information

Guillaume de Chambrier<sup>1,\*</sup>, Aude Billard<sup>1</sup>

<sup>1</sup>*École Polytechnique Fédérale de Lausanne (EPFL), Route Cantonale, 1015 Lausanne, Switzerland*

Correspondence\*:

Guillaume de Chambrier  
guillaume.dechambrier@epfl.ch

2 WORD COUNT: 11'000

## 3 ABSTRACT

4 Simultaneous Localisation and Mapping (SLAM) is concerned with the development of filters  
5 to accurately and efficiently infer the state parameters (position, orientation,...) of an agent and  
6 aspects of its environment, commonly referred to as the map. A mapping system is necessary  
7 for the agent to achieve situatedness which is a precondition for planning and reasoning. In  
8 this work we consider an agent who is given the task of finding a set of objects. The agent has  
9 limited perception and can only sense the presences of objects if a direct contact is made, as  
10 a result most of the sensing is negative information. In the absence of recurrent sightings or  
11 direct measurements of objects there are no correlations from the measurement errors which  
12 can be exploited. This renders SLAM estimators, for which this fact is their backbone such as  
13 EKF-SLAM, ineffective. In addition for our setting, no assumptions are taken with respect to the  
14 marginals (beliefs) of both the agent and objects (map).

15 From the loose assumptions we stipulate regarding the marginals and measurements, we adopt  
16 a histogram parametrisation. We introduce a Bayesian State Space Estimator (BSSE), which we  
17 name Measurement Likelihood Memory Filter (MLMF), in which the values of the joint distribution  
18 are not parametrised but instead we directly apply changes from the measurement integration  
19 step to the marginals. This is achieved by keeping track of the history of likelihood functions'  
20 parameters.

21 We demonstrate that the MLMF gives the same filtered marginals as a histogram filter and show  
22 two implementations: MLMF and scalable-MLMF which both have a linear space complexity. The  
23 original MLMF retains an exponential time complexity (although an order of magnitude smaller  
24 than the histogram filter) whilst the scalable-MLMF introduced independence assumption such  
25 to have a linear time complexity. We further quantitatively demonstrate the scalability of our  
26 algorithm with 25 beliefs having up to 10'000'000 states each. In an Active-SLAM setting we  
27 evaluate the impact that the size of the memory's history has on the decision theoretic process in  
28 a search scenario for a one step look ahead information gain planner. We report on both 1D and  
29 2D experiments.

30 **Keywords:** Negative Information, SLAM, Bayesian State Space Estimator, Histogram-SLAM, Active-exploration

## 1 INTRODUCTION

31 Estimating the location of a mobile agent whilst simultaneously building a map of the environment has  
32 been regarded as one of the most important problems to be solved for agents to achieve autonomy. It is a  
33 necessary precondition for any agent to have an estimation of the world at its disposal which accurately  
34 maintains all acquired knowledge. There has been a vast amount of research surrounding the field of  
35 Simultaneous Localisation And Mapping (SLAM) which branches out into a wide variety of sub-fields  
36 dealing with problems from building accurate noise models of the agent sensors Plagemann et al. (2007),  
37 to determining which environmental feature caused a particular measurement, also known as the data  
38 association problem Montemerlo and Thrun (2003) and many more.

39 Although the amount of research might seem overwhelming at first, all current SLAM methodologies  
40 are founded on a single principle; the uncertainty of the map's features are correlated through successive  
41 measurements of agent obtained at different locations Durrant-Whyte and Bailey (2006a). As the agent  
42 localises itself by reducing its position uncertainty, the maps' uncertainty also decreases as the landmarks  
43 are correlated. The predominant mathematical formulation of a SLAM problems is to parameterise the  
44 joint distribution of the agent's position and maps' features and to recursively integrate applied actions and  
45 senses measurements, which is referred as Probabilistic-SLAM Cadena et al. (2016); Durrant-Whyte and  
46 Bailey (2006a).

47 There are broadley three main paradigms to parameterising the joint distribution and integrating the  
48 sensed information:

49 The first paradigms assumes the joint distribution to be a Multivariate Gaussian distribution and is  
50 parameterised by a mean and covariance. The mean is a state vector composed of the agent and map  
51 feature positions whilst the elements of the covariance encode the position uncertainty of the state vector.  
52 If the motion and measurement system models are linear the Kalman Filter recursion is the best estimator  
53 of the joint's parameters. In case the system models are non-linear they can be linearised via a Taylor  
54 approximation and in the SLAM context this is known as EKF-SLAM (Extendend-Kalman Filter) Durrant-  
55 Whyte and Bailey (2006a) an early popular method.

56 Unscented Kalman Filter Wan and Van Der Merwe (2000) Huang et al. (2013),

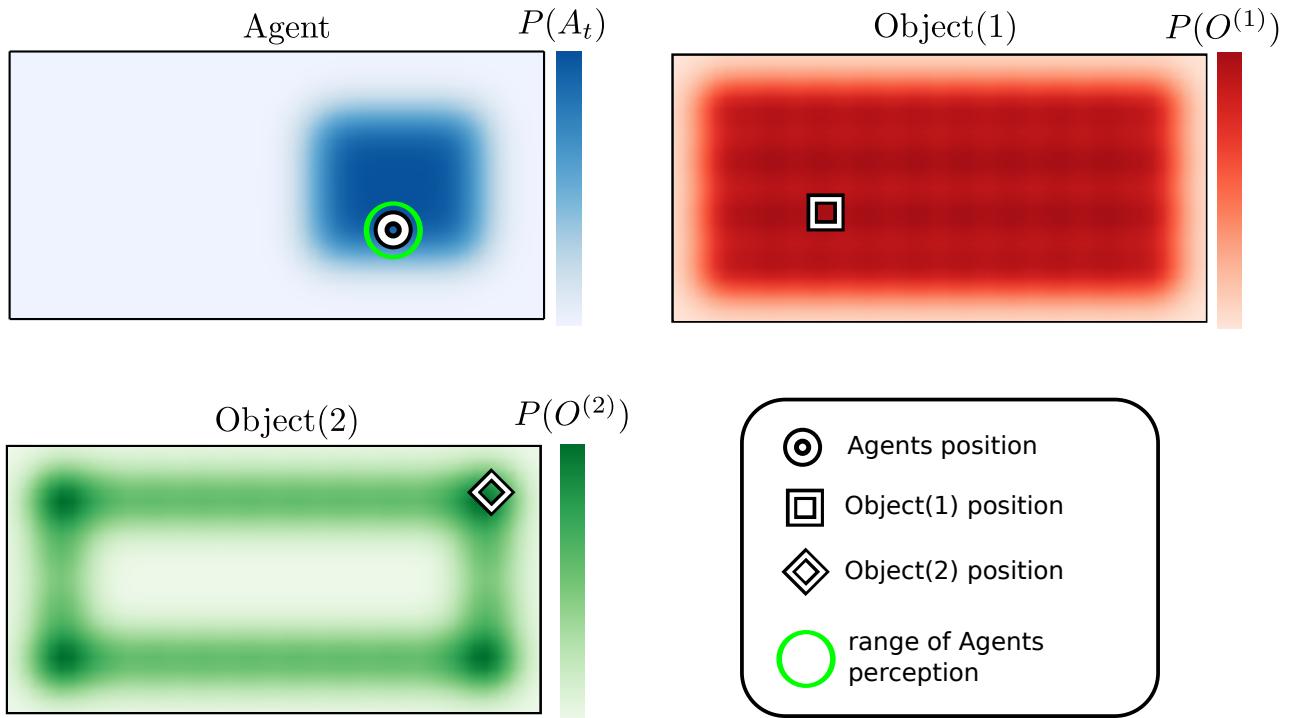
57 Extended information Filter Thrun et al. (2004), Torres-Gonzlez et al. (2014).

58 Hesch et al. (2014) reviewed the inconsistencies in vision-aided navigation system (VINS) in which data  
59 from camera and inertial measurement units are fused to estimate the position and orientation of a sensor.  
60 Main source of inconsistency is due to the linearisation of the measurement function which falsely reduces  
61 the uncertainty in pose estimation which results an increase of the estimation error. The authors introduce a  
62 constraint on the Jacobian of both motion and measurement functions which explicitly enforces that no  
63 unwanted information is gained. The authors demonstrated their EKF-SLAM with additional constraints  
64 on 1.5km outdoor navigation and mapping of the University of Minnesota campus.

65 Li and Mourikis (2013) multi-state-constraint Kalman filter MSCK F2.0 is a real-time EKF-SLAM  
66 for visual-inertial odometry (VIO) sensory information. Linearisation of system models (motion and  
67 measurement models) lead to the uncertainty in the state being underestimated.

68 MSCK Mourikis and Roumeliotis (2007) algorithm outperforms EKF-SLAM methods by a wide margin.  
69 As oppose to EKF-SLAM MSCKF maintains a sliding window of poses in its state vector and uses feature  
70 measurements to impose constraints on these poses. MSCKF is linear in the number of features. Performs  
71 also online camera calibration.

- 72 Hidalgo and Brunl (2015)
- 73 FastSLAM: one drawback is that information is lost by during resampling process.
- 74 Paull et al. (2014)
- 75 Mallios et al. (2014)
- 76 The second approach is Graph-SLAM Grisetti et al. (2010). Graph-SLAM estimates the full path of  
77 the agent and considers every measurement to be a constraint on the agent's path. It is parameterised by  
78 the canonical Multivariate Gaussian. At each time step a new row and column is added to the precision  
79 matrix which encodes landmarks which have been observed as constraints on the robot's position. At  
80 predetermined times, a nonlinear sparse optimisation is solved to minimise all the accumulated constraints  
81 on the robot's path.
- 82 The third method is FastSLAM Montemerlo et al. (2003). FastSLAM exploits the fact that if we know  
83 the agent's position with certainty all landmarks become independent. It models the distribution of the  
84 agent's position by a particle filter. Each particle has its own copy of the map and updates all landmarks  
85 independently which is the strength of this method. However, if many particles are required each must have  
86 its own copy of the map. It is beyond the scope of this chapter to provide a detailed review of these three  
87 paradigms and the reader is referred to Thrun et al. (2005), Thrun and Leonard (2008).
- 88 **1.1 Active-SLAM & Exploration**
- 89 Active-SLAM refers to a decision theoretic process of choosing control actions so as to actively increase  
90 the convergence of the map. It is used in conjunction with exploration of an unknown environment in  
91 a SLAM setting. The two steps of this process are: (i) generate a set of candidate destination positions,  
92 (ii) evaluate these positions based on a utility function. The utility is a trade off between reducing the  
93 uncertainty of the map or reducing the uncertainty of the agent's position.
- 94 Most approaches use a two-level representation of the map in an exploration setting. At the lower level  
95 there is the chosen (landmark-based) SLAM filter and at the higher level a coarser representation of the  
96 world. Such representations can be occupancy grids Thrun and Bü (1996) which encode either occupied  
97 and free space or a topological representation Kollar and Roy (2008).
- 98 Early and current approaches to selecting candidate exploratory locations are based on evaluating Next-  
99 best-view González-Baños and Latombe (2002) locations. Next-best-view points are sampled around *free*  
100 *edges* which are at the horizon of the known map (*frontier* regions). In such a setting only target points are  
101 generated, not the full trajectory. Probabilistic Road Map (PRM) Kavraki et al. (1996) based methods have  
102 been used as planners to reach desired target locations, such as in Huang and Gupta (2008), where a Rapidly  
103 Exploring Random Trees (RRT) is combined with FastSLAM. In Carrillo et al. (2012a), paths to *frontier*  
104 regions are computed via PRM on a occupancy grid map and at the lower level they use Pose-SLAM  
105 (synonym for Graph-SLAM).
- 106 An alternative approach taken to generating candidate locations is the specification of high level macro  
107 actions, they being either *exploratory* or *revisiting* actions as is the case in Stachniss et al. (2005). Macro  
108 actions reduce the costly evaluation of actions, especially in the case of FastSLAM, which requires  
109 propagating the filter forward in time so as to infer the information gain of each action.
- 110 The last approach is to solve the planning problem through formulating it as Partially Observable Markov  
111 Decision Process (POMDP) Ross et al. (2008). However all methods take an approximation of the POMDP  
112 and consider a one time step planning horizon (Lidoris, 2011, p.37).



**Figure 1. Table World** There are three different probability density functions present on the table. The blue represents the believed location of the agent, the red and green probability distributions are associated with object 1 and 2. The white shapes in each figure represent the true location of each associated object or agent.

113 There are many ways of generating actions or paths, however their utility is nearly all exclusively based  
 114 on the *information gain*, which is the estimated reduction of entropy a particular action or path would  
 115 achieve. A few utilities use f-measures such as the Kullback-Leibler divergence. Evaluation of different  
 116 utility metrics are presented in Carrillo et al. (2012b); Carlone et al. (2010).

## 117 1.2 Problem Statement

118 We consider an agent searching for a set of objects in a partially-known environment, in which  
 119 exteroceptive feedback is extremely limited. In the case of our agent, we can think of it as having a  
 120 range sensor which only provides a response when in direct contact with an object. Our agent lives in a  
 121 *Table Top* world (see Figure 1) in which is located a set of objects. The agent's uncertainty of its location  
 122 and that of the objects is encoded by probability distributions  $P(\cdot)$ , which at initialisation are known as the  
 123 agent's prior beliefs.

124 As the agent explores the world, it integrates all sensing information at each time step and updates its  
 125 prior beliefs to posteriors (the result of the prior belief after integrating motion and sensory information).  
 126 All current SLAM methods are limited in that they consider only uncertainty induced by sensing inaccuracy  
 127 inherent in the sensor and motion models. The reason for this wide spread limitation is that during  
 128 theoretical foundation of SLAM, a period which is referred to as the *classic age* (1986-2004) Cadena et al.  
 129 (2016) (this period gave the three paradigms discussed above), the application problems considered the  
 130 signal noise in range sensor, cameras and odometry measurements. The features were always considered to

131 be measurable except when occluded. In our setting as the sensory information is strictly haptic, we can  
 132 confidently assume no measurement noise.

133 In the search task illustrated in Figure 1, the beliefs and sparse measurement information available to  
 134 the agent are the source of the uncertainty which is, the absence of positive object measurements. This is  
 135 known as **negative information** (Thrun et al., 2005, p.313) Thrun (2002); Hoffman et al. (2005). Thus  
 136 SLAM methodologies which use the **Gaussian error** between the predicted and estimated position of  
 137 features, such as in the case of EKF-SLAM and Graph-SLAM, will not perform well in this setting as no  
 138 measurements of the features positions available until our agent “bumps” into a feature.

139 In addition to the negative sensing information, the original beliefs depicted in Figure 1 are **non-Gaussian**  
 140 and **multi-modal**. We make **no assumption** regarding the form of the beliefs. This implies that the joint  
 141 distribution can no longer be parameterised by a Multivariate Gaussian. This is an assumption made in  
 142 many SLAM algorithms, notably EKF-SLAM, and allows for a closed form solution to the state estimation  
 143 problem. Without the Gaussian assumption no closed form solution to the filtering problem is feasible.  
 144 Using standard non-parametric methods (Kernel Density, Gaussian Process, Histogram,...) to represent  
 145 or estimate the joint distribution becomes unrealistic after a few dimensions or additional map features.  
 146 FastSLAM could be a potential candidate, however as it parameterises the position uncertainty of the agent  
 147 by a particle filter and each particle has its own copy of the map, the memory demands become quickly  
 148 significant. For planning purposes we would also want to have a single representation of the marginals.  
 149 The box below summarises the desirable attributes and assumptions for our filter.

#### Attributes & Assumptions

- Non-Gaussian joint distribution, no assumptions are made with respect to its form.
- Mostly negative information available (absence of positive sightings of the landmarks).
- Joint distribution volume grows exponentially with respect to the number of objects and states.
- Joint distribution volume is dense, there is high uncertainty.

150

### 151 1.3 The main contribution to the field of Artificial Intelligence

152 In a wide range of Artificial Intelligence (AI) applications the agent’s beliefs are discrete. This non-  
 153 parametric representation is the most unconstraining but comes at a cost. The parameterisation of the belief’s  
 154 joint distribution grows at the rate of a double exponential. We propose a Bayesian State Space Estimator  
 155 (BSSE) which delivers the same filtered beliefs as a traditional filter but without explicitly parametrising  
 156 the joint distribution. We refer to our novel filter as the Measurement Likelihood Memory Filter (MLMF).  
 157 It keeps track of the history of measurement likelihood functions, referred to as the memory, which have  
 158 been applied on the joint distribution. The MLMF filter efficiently processes negative information. To the  
 159 author’s knowledge there has been little research on the integration of negative information in a SLAM  
 160 setting. Previous work considered the case of active localisation Hoffmann et al. (2006). The incorporation  
 161 of negative information is useful in many contexts and in particular in Bayesian Theory of Mind, Bake et al.  
 162 (2011), where the reasoning process of a human is inferred from a Bayesian Network and in our own work  
 163 de Chambrier and Billard (2013) where we model the search behaviour of a intentionally blinded human.  
 164 In such a setting much negative information is present and an efficient belief filter is required. Our MLMF

165 is thus applicable to the SLAM & AI community in general and to the Cognitive Science community which  
 166 models human or agent behaviours through the usage of Bayesian state estimators.

167 By using this new representation we implement a set of passive search trajectories through the state space  
 168 and demonstrate, for a discretised state space, that our novel filter is optimal with respect to the Bayesian  
 169 criteria (the successive filtered posteriors are exact and not an approximation with respect to Bayes rule).  
 170 We provide an analysis of the space and time complexity of our algorithm and prove that it is always more  
 171 efficient even when considering worst case scenarios. Lastly we consider an Active-SLAM setting and  
 172 evaluate how constraining the size of the number of memorised likelihood functions impacts the decision  
 173 making process of a greedy one-step look-ahead planner.

174 The rest of the document is structured as follows: in section 2, we overview the Bayes filter recursion  
 175 and apply it to a simple 1D search scenario for both a discrete and Gaussian parametrisation of the beliefs.  
 176 We demonstrate that discrete parametrisation gives the correct filtered beliefs but at a very high cost and  
 177 that the EKF-SLAM fails to provide the adequate solution. Section 3 we introduce the Measurement  
 178 Likelihood Memory Filter and overview its parametrisation. Section 4 we derive the computational time  
 179 and space complexity of the MLMF. Section 5 describes additional assumptions made with respect to the  
 180 MLMF to make it scalable (scalable-MLMF). In section 6 we numerically evaluate the time complexity of  
 181 the scalable-MLMF and check the assumption we made for it to be scalable. We investigate the filter's  
 182 sensitivity with respect to its parameters in an Active-SLAM setting.

## 2 BAYESIAN STATE SPACE ESTIMATION

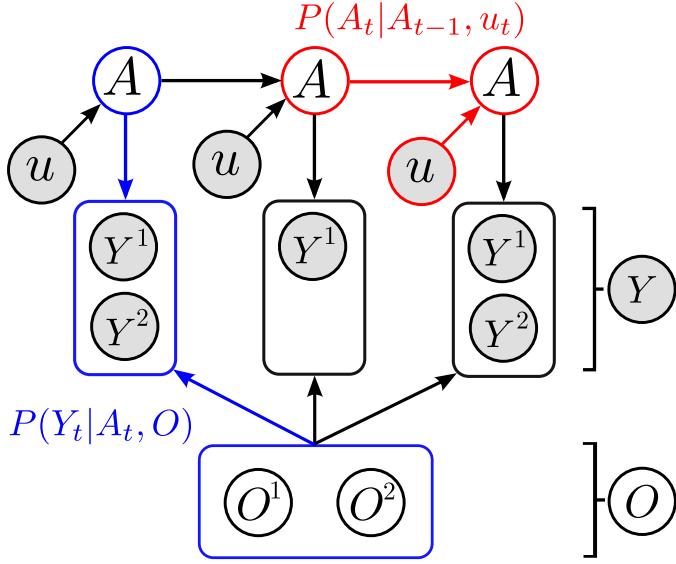
183 Bayesian State Space Estimation (BSSE) focuses on incorporating observations to update a prior distribution  
 184 to a posterior distribution over the state space through the application of Bayes probability rules. The  
 185 agent's random variable,  $A$ , is associated with the uncertainty of its location in the world, the same holds for  
 186 the object(s') random variable(s),  $O$ . Given a sequence of actions and observations,  $\{u_{1:t}, y_{0:t}\}$  (subscript  
 187  $0 : t$  is all the indexed variables from  $t = 0$  to the current time  $t = t$ ), algorithms of the BSSE family  
 188 incorporate this information to provide an estimate  $P(A_t, O|Y_{0:t}, u_{1:t})$ . This is known as the filtering  
 189 problem where all past information is incorporated to estimate the current state.

190 In Figure 2 we depict the general Bayesian Network (BN) of a BSSE. The BN conveys two types of  
 191 information, the dependence and independence relation between the random variables in the graph which  
 192 can be established through *d-separation* Shachter (1998). The **conditional dependence**  $A \perp\!\!\!\perp O|Y$  is key to  
 193 all BSSE and SLAM algorithms. The strength of the dependence between the agent and object random  
 194 variable is governed by the measurement likelihood  $P(Y_t|A_t, O)$ . If the measurement likelihood does not  
 195 change the joint distribution, then the agent and object random variables will be independent,  $A \perp\!\!\!\perp O$ . If  
 196 they are independent, then no information acquired by the agent can be used to infer changes in the object  
 197 estimates.

198 We next demonstrate the behaviour of the BN joint distribution, Figure 2, for two different  
 199 parameterisations in the case of the absence of direct sighting of the object by the agent.

### 200 2.1 EKF-SLAM

201 In EKF-SLAM the joint density  $p(A_t, O|Y_{0:t}, u_{1:t}) = g([A_t, O]^T; \mu_t, \Sigma_t)$  is parametrised by a single  
 202 Gaussian function  $g$  with mean,  $\mu_t = [\mu_{A_t}, \mu_{O^{(1)}}, \dots, \mu_{O^{(M-1)}}]^T \in \mathbb{R}^{3+2 \cdot (M-1)}$  where the object random  
 203 variables are in  $\mathbb{R}^2$ , and covariance,  $\Sigma_t$ . The mean value of the agent  $\mu_a = [x, y, \phi]^T \in \mathbb{R}^3$  and those of the  
 204 objects are  $\mu_{O^{(i)}} = [x, y]^T \in \mathbb{R}^2$ .



**Figure 2.** Directed graphical model of dependencies between the agent(A) and object(O)'s estimated location. Each object,  $O^{(i)}$  is associated with one sensing random variable  $Y^{(i)}$ . The overall sensing random variable is  $Y = [Y^{(1)}, \dots, Y^{(M-1)}]^T$ , where  $M$  is the total number of agent and object random variables in the filter. For readability we have left out the time index  $t$  from  $A$  and  $Y$ . Since the objects are static, they have no temporal process associated with them thus they will never have a time subscript. The two models necessary for filtering are the motion model  $P(A_t | A_{t-1}, u_t)$  (red) and measurement model  $P(Y_t | A_t, O)$  (blue).

$$\Sigma_t = \begin{bmatrix} \Sigma_a & \Sigma_{ao} \\ \Sigma_{oa} & \Sigma_o \end{bmatrix} \in \mathbb{R}^{(3+2\cdot(M-1)) \times (3+2\cdot(M-1))} \quad (1)$$

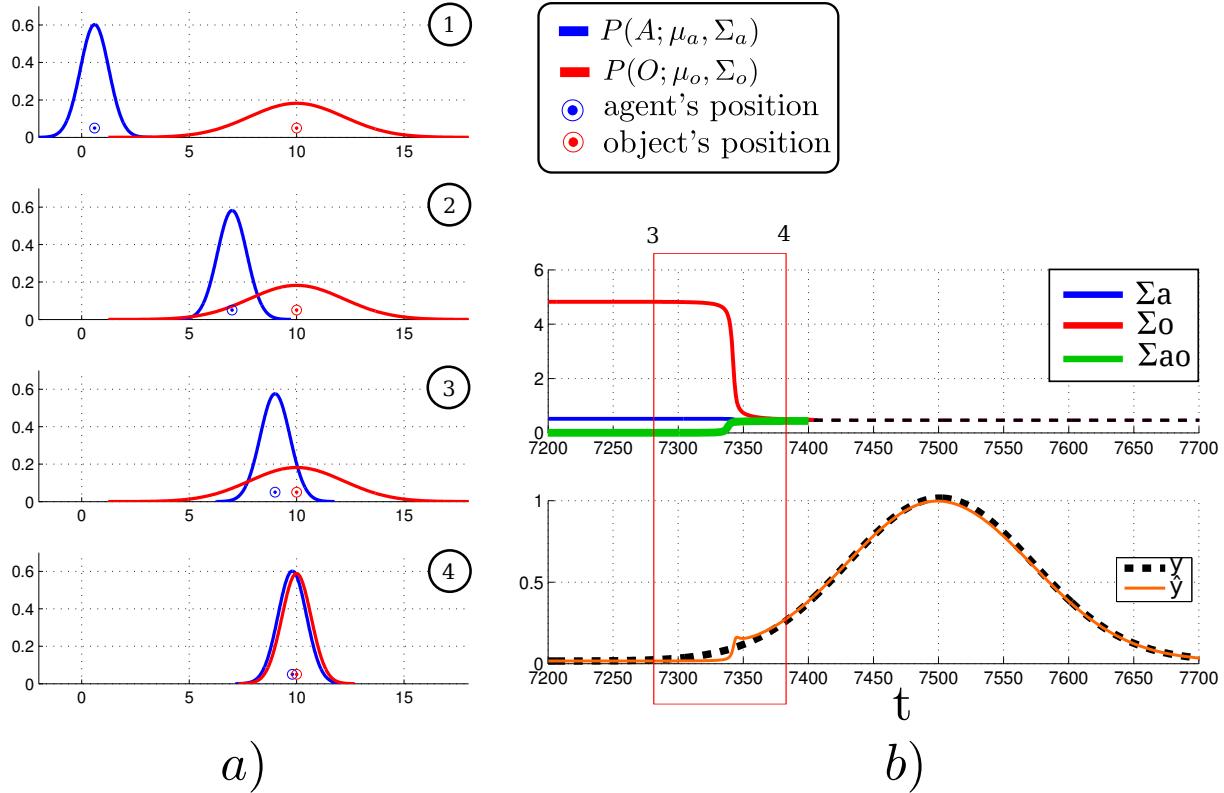
The  $j$ 'th object measurement is described by range and bearing  $Y_t^{(j)} = [r, \phi]$  in the frame of reference of the agent. EKF-SLAM assumes that the measurement is corrupted by Gaussian noise,  $\epsilon \sim \mathcal{N}(0, R)$ , resulting in the likelihood function:

$$p(Y_t | A_t, O_t) = \frac{1}{|2\pi R|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (Y_t - \hat{Y}_t)^T R^{-1} (Y_t - \hat{Y}_t) \right) \quad (2)$$

$$\hat{Y}_t = \exp \left( -\frac{1}{2\sigma^2} \|A_t - O\|^2 \right) \quad (3)$$

where the covariance,  $R$ , encompasses the uncertainty in the measurement and Equation 3 is the measurement function. The elements of the covariance matrix capture the measurement error between the true  $Y$  and expected  $\hat{Y}$  range and bearing of the object. As the joint distribution is parametrised by a single Multivariate Gaussian, a closed form solution to the filtering Equations exists, called the Kalman Filter Durrant-Whyte and Bailey (2006b).

The error between the true and expected measurement  $e = (Y_t - \hat{Y}_t)$  is an important part of the application of EKF-SLAM. In our scenario the agent can only perceive the objects once he enters in direct contact with them. This means that the variance of the observation  $Y_t$  will always be equal to  $\hat{Y}$  until a contact



**Figure 3.** **a)** EKF-SLAM time slice evolutions of the pdfs. The temporal ordering is given by the numbers in the top right corner of each plot. The blue pdf represents the agent's believed location and the circle is the agent's true location. The same holds for the red distribution which represents the agent's belief of the location of an object. **b)** Evolution of the covariance components of  $\Sigma$  over time and true  $Y_t$  and expected measurements,  $\hat{Y}_t$ .  $\Sigma_a$  and  $\Sigma_o$  are the variances of the agent and object positions and  $\Sigma_{ao}$  is the cross-covariance term.

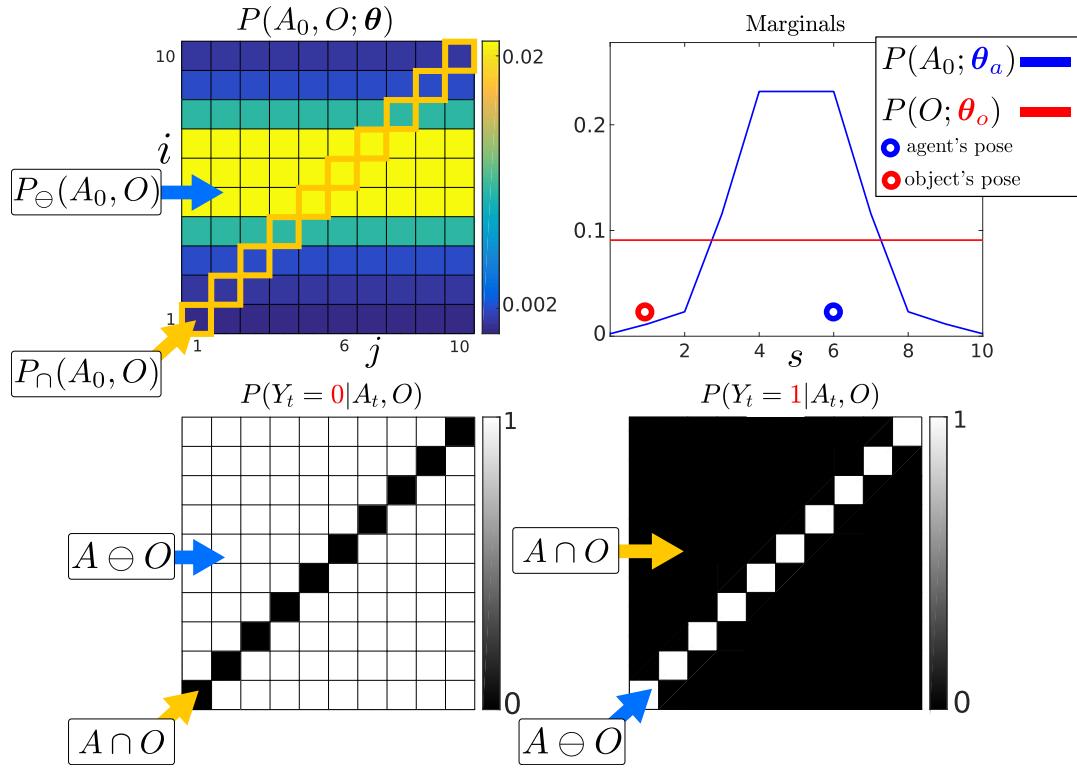
213 occurs. To illustrate the problems which this gives rise to, we give an illustration of a 1D search. Figure 3  
214 shows the resulting updates of the beliefs for 4 chosen time segments.

215 As expected we do not get the desired behaviour, that the beliefs start updating as soon as they are  
216 overlapping, see 2nd-3rd temporal snapshot in the Figure. Even when most of the belief mass of the  
217 agent's location pdf overlaps that of the object pdf, no belief update occurs. The multivariate Gaussian  
218 parameterisation only guarantees a dependency between the agent and object random variables when there  
219 is a positive sighting of the landmarks. This can be seen in Figure 3 (b), where the component  $\Sigma_{ao}$  is 0  
220 most of the time which implies that  $A \perp\!\!\!\perp O|Y$  which is undesirable.

## 221 2.2 Histogram-SLAM

222 In Histogram-SLAM, the joint distribution is discretized and each bin has a parameter,  
223  $P(A_t = i, O = j|Y_{0:t}, u_{1:t}; \boldsymbol{\theta}) = \boldsymbol{\theta}^{(ij)}$ , which sums to one,  $\sum_{ij} \boldsymbol{\theta}^{(ij)} = 1$ . For shorthand notation we  
224 will write  $P(A_t, O|Y_{0:t}, u_{1:t})$  instead of  $P(A_t = i, O = j|Y_{0:t}, u_{1:t}; \boldsymbol{\theta})$ . The probability distribution of the  
225 agent's position is given by marginalising the object random variable:

$$P(A_t|Y_{0:t}, u_{1:t}; \boldsymbol{\theta}_a) = \sum_{j=1}^{|O|} P(A_t, O = j|Y_{0:t}, u_{1:t}; \boldsymbol{\theta}) \quad (4)$$



**Figure 4.** **Top:** *Left:* Initialisation of the agent and object joint distribution. *Right:* Marginals of the agent and object parameterised by  $\theta_a$  and  $\theta_o$ , giving the probability of their location. The marginal of each random variable is obtained from Equation 4. The probability of the agent and object being in state  $s = 6$  is given by summing the blue and red highlighted parameters in the joint distribution. **Bottom:** 1D world Likelihood  $P(Y_t | A_t, O)$ , the white regions  $A \cap O$  will leave the joint distribution unchanged whilst the black regions will evaluate the joint distribution to zero. *Left:* No contact detected with the object, the current measurement is  $Y_t = 0$ , both the agent and object cannot be in the same state. *Right:* The agent entered into contact with the object and received a haptic feedback  $Y_t = 1$ . The agent receives only two measurement possibilities, contact or no contact.

226 The converse holds true for the object's marginal, that is the summation would be over the agents variable.  
 227 Figure 4 (*Top*) illustrates the joint distribution of both the agent and the object random variable. The 1D  
 228 world of the agent and object is discretised to 10 states, producing a joint distribution with 100 parameters!  
 229 For a state space of  $N$  bins,  $s = 1 \dots N$ , and there is a total of  $M$  random variables (one agent and  $M - 1$   
 230 objects) and the joint distribution has  $N^M$  parameters. This exponential increase renders Histogram-SLAM  
 231 intractable with this parameterisation.

232 In the tasks we consider, an observation occurs only if the agent enters in contact with the object, which  
 233 implies that both occupy the same discrete state. The likelihood function  $P(Y_t | A_t, O)$  is:

$$P(Y_t = 1 | A_t, O) = \begin{cases} 1 & \text{if } A_t = O \\ 0 & \text{if } A_t \neq O \end{cases} \quad (5)$$

234 Figure 4 (*Bottom left*), illustrates the likelihood function, Equation 5, in the case when a no contact  
 235 measurement  $Y_t = 0$ . When there is no measurement all the parameters of the joint distribution which are  
 236 in the black regions become zero, which we refer to as the **dependent states**  $A \cap O$  of the joint distribution.  
 237 The white states are the **independent states**  $A \ominus O$ , they are not changed by the likelihood function and

238 the values of the joint distribution in those states,  $P_{\cap}(A_t, O|Y_{0:t}, u_{1:t})$ , will be unchanged by the likelihood  
 239 function  $P_{\ominus}(A_t, O|Y_{0:t}, u_{1:t}) \propto P_{\ominus}(A_t, O|Y_{0:t-1}, u_{1:t})$ . When the object is detected (*Bottom right*) the  
 240 likelihood constrains all non-zero values of the joint distribution to be in states  $i = j$ , which in the case  
 241 of a 2-dimensional joint distribution is a line. The **sparsity** of the likelihood function will be key to the  
 242 development of the MLMF filter. Two models are needed to perform the recursion, namely the motion  
 243 model  $P(A_t|A_{t-1}, u_t)$  and the measurement model  $P(Y_t|A_t, O)$ , which we already detailed. Both models  
 244 applied consecutively to the initial joint distribution results in a posterior distribution. Both Equation 7-8  
 245 are part of the Histogram Bayesian filter update:

Histogram Bayesian recursion

**initialisation**

$$P(A_0, O; \boldsymbol{\theta}) = P(A_0; \boldsymbol{\theta}_a) P(O; \boldsymbol{\theta}_o) = \boldsymbol{\theta}_a \times \boldsymbol{\theta}_o \quad (6)$$

**motion**

$$P(A_t, O|Y_{0:t-1}, u_{1:t}) = \sum_{A_{t-1}} P(A_t|A_{t-1}, u_t) P(A_{t-1}, O_{t-1}|Y_{0:t-1}, u_{1:t-1}) \quad (7)$$

**measurement**

$$P(A_t, O|Y_{0:t}, u_{1:t}) = \frac{P(Y_t|A_t, O) P(A_t, O|Y_{0:t-1}, u_{1:t})}{P(Y_t|Y_{0:t-1}, u_{1:t})} \quad (8)$$

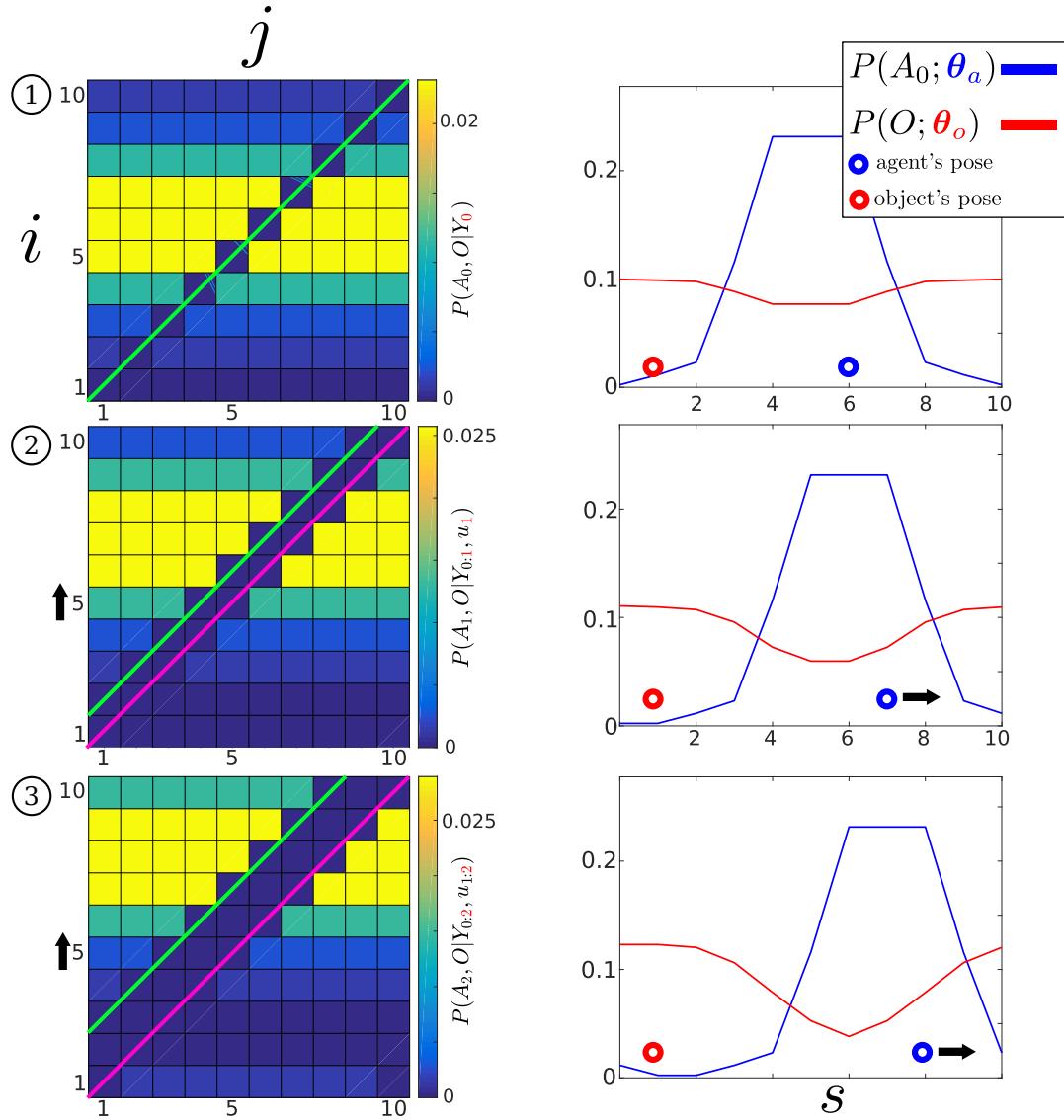
246

247 Figure 5 illustrates the evolution of the joint distribution in a 1D example. The agent and object's true  
 248 positions (unobservable) are in state 6 and 1. The agent moves three steps towards state 10. At each time  
 249 step, as the agent fails to sense the object, the likelihood function  $P(Y_t = 0|A_t, O)$  (Figure 4, *Bottom left*)  
 250 is applied. As the agent moves towards the right, the motion model shifts the joint distribution towards  
 251 state 10 along the agent's dimension,  $i$  (note that state 1 and 10 are wrapped).

252 As the agent moves to the right more joint distribution parameters become zero. The re-normalisation by  
 253 the **evidence** ( $P(Y_t|Y_{0:t-1}, u_{1:t})$ , denominator of Equation 8), which increases the value of the remaining  
 254 parameters, is equal to the sum of the probability mass which was set to zero by the likelihood function.  
 255 Thus the values of the parameters of the joint distribution which fall on the pink line in Figure 5 (green line  
 256 also, but only for first time slice) become zero and their values are redistributed to the remaining non-zero  
 257 parameters.

258 The **inconvenience** with Histogram-SLAM is that its time and space complexity is exponential as the  
 259 joint distribution is discretised and parametrised by  $\boldsymbol{\theta}^{(ij)}$ . Instead we propose a new filter, MLMF, which  
 260 we formally introduce in the next section. This filter achieves the same result as the Histogram filter but  
 261 without having to parameterise the values of the joint distribution, thus avoiding the exponential growth  
 262 cost.

263 The **key idea** behind the mechanism of the MLMF filter is to evaluate only the joint distribution  
 264  $P_{\cap}(A_t, O|Y_{0:t}, u_{1:t})$  in dependent states and updates directly to the marginals without parameterising the  
 265 values of the joint distribution. The MLMF filter parametrises **explicitly** the marginals  $P(A_t|Y_{0:t}, u_{1:t}; \boldsymbol{\theta}_a)$ ,  
 266  $P(O|Y_{0:t}; \boldsymbol{\theta}_o)$ . This contrasts the Histogram filter where the marginals are derived from the joint distribution  
 267 by marginalisation over the entire joint state space.



**Figure 5.** Histogram-SLAM, 4 time steps. **1** Application of likelihood  $P(Y_0 = 0|A_0, O)$  and the agent remains stationary, all states along the green line become zero. **2** The agent moves to the right  $u_1 = 1$ , the motion  $P(A_1|A_0, u_1)$ , and likelihood models are applied consecutively. The right motion results in a shift (black arrow on the left) in the joint probability distribution towards the state  $i = 10$ . All parameters on the pink line are zero. **3** Same as two. At each time step a new likelihood function (pink line) is applied to the joint distribution.

### 3 MEASUREMENT LIKELIHOOD MEMORY FILTER

MLMF keeps a **function parameterisation** of the joint distribution instead of a **value parameterisation** as it is the case for Histogram-SLAM. At initialisation the joint distribution is represented by the product of marginals, Equation 9, which would result in the joint distribution illustrated in Figure 4, if it were to be evaluated at all states  $(i, j)$  as it is done for Histogram-SLAM. MLMF will only evaluate this product, when necessary, at specific states. At each time step the motion and measurement update are applied, Equation 10-11. An important distinction is that these updates are performed on the **un-normalised** joint distribution  $P(A_t, O, Y_{0:t}|u_{1:t})$ , which is not the case in Histogram-SLAM where the updates are done on the conditional  $P(A_t, O|Y_{0:t}, u_{1:t})$ . After applying multiple motion and measurement updates the resulting joint distribution is given by Equation 12, see Appendix 8.3 for a step-by-step derivation.

## MLMF Bayesian filter

## joint marginals (initial)

$$P(A_0, O) = P(A_0; \theta_a^*) P(O; \theta_o^*) \quad (9)$$

## motion

$$P(A_t, O, Y_{0:t-1}|u_{1:t}) = \sum_{A_{t-1}} P(A_t|A_{t-1}, u_t) P(A_{t-1}, O, Y_{0:t-1}|u_{1:t-1}) \quad (10)$$

## measurement

$$\begin{aligned} P(A_t, O, Y_{0:t}|u_{1:t}; \theta_o^*, \theta_a^*, \Psi_{0:t}) &= \\ P(Y_t|A_t, O) P(O; \theta_o^*) P(A_t|u_{1:t}; \theta_a^*) P(Y_{0:t}|A_t, O, u_{1:t}; \bar{\Psi}_{0:t}) \end{aligned} \quad (11)$$

## joint

$$P(A_t, O|Y_{0:t}, u_{1:t}; \theta_o^*, \theta_a^*, \Psi_{0:t}, \alpha_{0:t}) = \frac{P(A_t, O, Y_{0:t}|u_{1:t}; \theta_o^*, \theta_a^*, \Psi_{0:t})}{P(Y_{0:t}|u_{1:t}; \alpha_{0:t})} \quad (12)$$

## filtered marginal

$$P(A_t|Y_{0:t}; \theta_a) = P(A_t|Y_{0:t-1}; \theta_a) - \left( P_{\cap}(A_t|Y_{0:t-1}) - P_{\cap}(A_t|Y_{0:t}) \right) \quad (13)$$

$$P(O|Y_{0:t}; \theta_o) = P(O|Y_{0:t-1}; \theta_o) - \left( P_{\cap}(A_t|Y_{0:t-1}) - P_{\cap}(A_t|Y_{0:t}) \right) \quad (14)$$

277

278 The MLMF filter is parameterised by the agent and object **joint marginals**  $P(A_t|u_{1:t}; \theta_a^*)$ ,  $P(O; \theta_o^*)$ ,  
 279 the **filtered marginals**  $P(A_t|Y_{0:t}, u_{1:t}; \theta_a)$  ( $u_{1:t}$  not shown in the above box),  $P(O|Y_{0:t}; \theta_o)$ , the evidence  
 280  $P(Y_{0:t}|u_{1:t}; \alpha_{0:t})$  and the history of likelihood functions,  $P(Y_{0:t}|A_t, O, u_{1:t}; \Psi_{0:t})$  Equation 15, which  
 281 is the product of all the likelihood functions since  $t = 0$  until  $t$  and we will refer to it as the **memory**  
 282 **likelihood function**:

$$P(Y_{0:t}|A_t, O, u_{1:t}; \Psi_{0:t}) := \prod_{i=0}^t P(\mathbf{Y}_i|A_t, O, u_{i+1:t}; l_i) \quad (15)$$

$$P(\mathbf{Y}_i = 0|A_t, O, u_{i+1:t}; l_i) := \begin{cases} 0 & \text{if } A_t + l_i = O \\ 1 & \text{else} \end{cases} \quad (16)$$

283

$$l_i := \sum_{j=i+1}^t u_j \quad (17)$$

284 The memory likelihood function's parameters  $\Psi_{0:t} = \{(Y_i, l_i)\}_{i=0:t}$  consist of a set of measurements  
 285  $Y_{0:t}$  and offsets  $l_{0:t}$  depicted in greed. The measurements  $Y_i \in \{0, 1\}$  are always binary, whilst the offsets  $l_i$ ,  
 286 actions  $u_t$ , agent  $A_t$  and object  $O$  variables' size are equal to the dimension of the state space. The subscript  
 287  $i$  of an offset  $l_i$  indicates which likelihood function it belongs to. The offset of a likelihood function is  
 288 given by the summation of all the applied actions from the time the likelihood was added until the current  
 289 time  $t$ , Equation 17, which can be computed recursively. The motion update, Equation 10, when applied to

290 the joint distribution results in the initial marginal  $P(A_0; \theta_a^*)$  and the likelihood functions being moved  
 291 along the agent's axis. In Algorithm 1, we detail how an action  $u_t$  and measurement  $Y_t$ , result in the update  
 292 of the memory likelihood's parameters from  $\Psi_{0:t-t}$  to  $\Psi_{0:t}$ ; this is an implementation of Equations 10-11.

---

**Algorithm 1:** Memory Likelihood update
 

---

**input :**  $\Psi_{0:t-1}, Y_t, u_t$   
**output :**  $\Psi_{0:t}$

---

293     **motion update**  $\bar{\Psi}_{0:t} \leftarrow \Psi_{0:t-1}$   
 1     **for**  $l_i \in \Psi_{0:t-1}$  **do**  
 2          $l_i = l_i + u_t$   


---

     **measurement update**  
 3      $\Psi_{0:t} \leftarrow \{\bar{\Psi}_{0:t}, (Y_t, l_t := 0)\}$

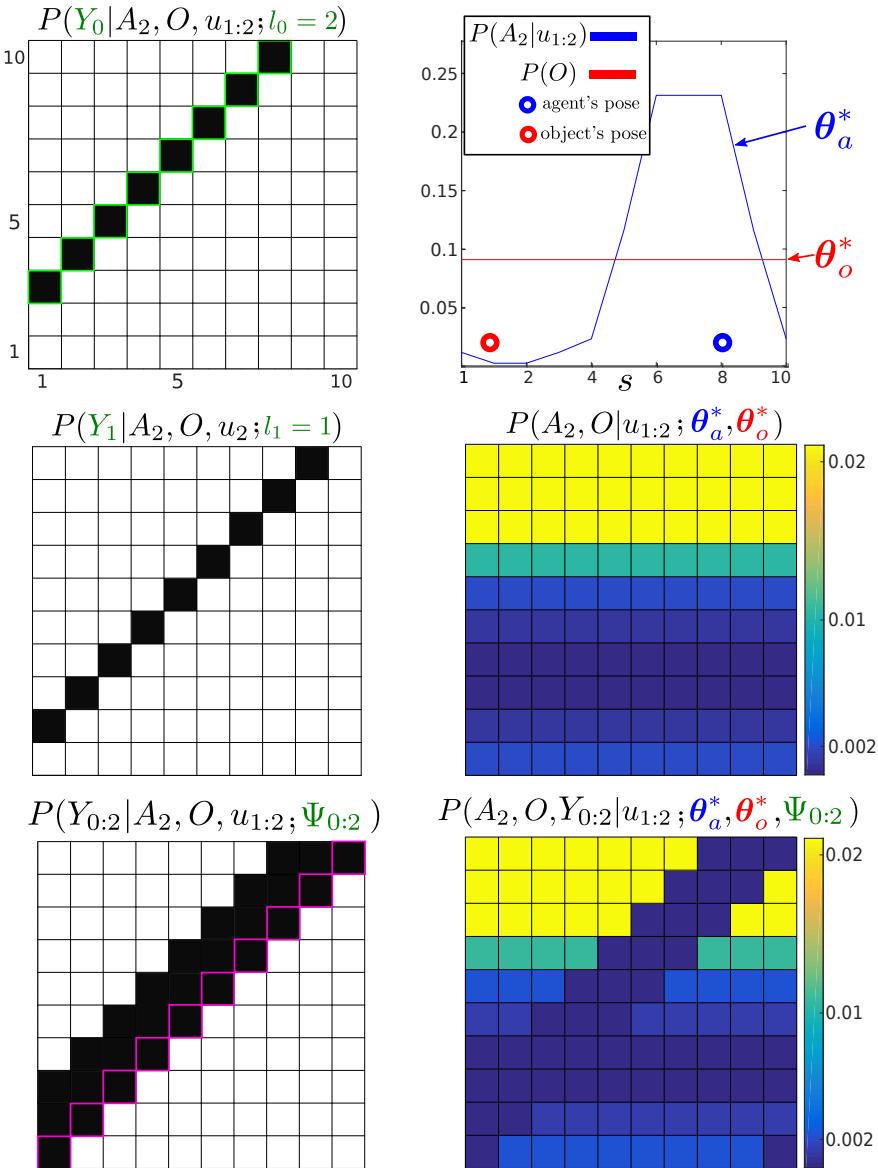
---

294     Figure 6 illustrates the evolution of the **un-normalised** MLMF joint distribution  $P(A_t, O, Y_{0:t}|u_{1:t})$ ,  
 295 Equation 12. For ease of notation we will omit at times the parameters of the probability functions. Both  
 296  $P(A_0; \theta_a^*)$  and  $P(O; \theta_o^*)$  were initialised as for the Histogram-SLAM example in Figure 5 on page 11.  
 297 Two actions  $u_{1:2} = 1$  are applied and three measurements  $Y_{0:2} = 0$  received which are then integrated into  
 298 the filter. Since initialisation of the joint distribution at  $t = 0$  until  $t = 2$  the object's marginal  $P(O; \theta_o^*)$   
 299 remains unchanged and the agent's marginal  $P(A_2|u_{1:2}; \theta_a^*)$  is updated by the two actions according to  
 300 the motion update, see Figure 6 (*Top-right*). The product of these two marginals (terms of Equation 12  
 301 before the memory likelihood product) results in the joint probability distribution  $P(A_2, O|u_{1:2}; \theta_a^*, \theta_o^*)$   
 302 illustrated in Figure 6 (*Middle-right*).

303     In the left column of Figure 6 we illustrate how the memory likelihood term, Equation 15, is updated  
 304 according to Algorithm 1. In the *Top-left*, the first likelihood function  $P(Y_0|A_2, O, u_{1:2}; l_0)$  is illustrated.  
 305 As two actions have been applied, Algorithm 1 is applied twice which results in a  $l_0 = 2$  parameter for the  
 306 first likelihood function. In the figure we highlighted the likelihood in light-green to indicate that it was the  
 307 first added to the memory term making it convenient to compare to Figure 5 on page 11. As for the second  
 308 likelihood function  $P(Y_1|A_2, O, u_2; l_1)$  only one action has been applied and the third likelihood function  
 309  $P(Y_2|A_2, O; l_2 = 0)$  has not yet been updated by the next action. The parameters of the memory likelihood  
 310 function, Equation 15, are:  $\Psi_{0:2} = \{(0, 2)_{i=0}, (0, 1)_{i=1}, (0, 0)_{i=2}\}$  and its evaluation is illustrated in the  
 311 *Bottom-left* of Figure 6.

312     The reader may have noticed that the amplitude of the values of the filtered joint distribution illustrated in  
 313 Figure 6 have changed when compared with Figure 5, but not the structure. This is because we have not  
 314 re-normalised the joint distribution by the evidence  $P(Y_{0:t}|u_{1:t}; \alpha_{0:t})$ . We will show in the next section how  
 315 we can **recursively** compute the evidence without having to integrate the whole joint distribution which  
 316 would be expensive.

317     Our goal is to be able to compute the marginals  $P(A_t|Y_{0:t}, u_{1:t}; \theta_a)$ ,  $P(O|Y_{0:t}; \theta_o)$  of the agent and object  
 318 random variables and evidence  $P(Y_{0:t}|u_{1:t}; \alpha_{0:t})$  **without** having to perform an **expensive marginalisation**  
 319 over the entire space of the joint distribution as was the case for Histogram-SLAM. The next section  
 320 describes how to efficiently compute the evidence and the marginals. For ease of notation we will not  
 321 always show the conditioned actions  $u_{1:t}$ , so  $P(A_t, O|Y_{0:t}, u_{1:t})$  will be  $P(A_t, O|Y_{0:t})$ .



**Figure 6.** Un-normalised MLMF joint distribution, numerator of Equation 12, at time  $t = 3$ . Three measurements (all  $Y = 0$ ) and two actions (both  $u = 1$ ) have been integrated into the joint distribution, for simplicity we do not consider any motion noise. *Left column:* The first plot illustrates the likelihood of the first measurement  $Y_0$ . We highlight the contour in light-green to indicate that it was the first applied likelihood function (see the correspondence with Figure 5). The first likelihood function has been moved by the 2 actions, the second likelihood function has been moved by one action (the last one,  $u_2 = 1$ ) and the third likelihood has had no action applied to it yet. The last applied likelihood function is highlighted in pink and the product of all the likelihoods since  $t = 0$  until  $t = 3$  is depicted at the bottom of the figure which is  $P(Y_{0:2}|A_2, O, u_{1:2})$ . *Right column:* the top figure illustrates the original marginal of the object  $P(O; \theta_o^*)$ , which remains unchanged, and the agent's marginal  $P(A_2|u_{1:2}; \theta_a^*)$  which has moved in accordance to the motion update function. Their product would result in the joint distribution  $P(A_2, O|u_{1:2}; \theta_a^*, \theta_o^*)$  illustrated in the middle figure if evaluated at each state  $(i, j)$ . The bottom figure is the result of multiplying  $P(A_2, O|u_{1:2}; \theta_a^*, \theta_o^*)$  with  $P(Y_{0:2}|A_2, O, u_{1:2}; \Psi_{0:2})$  giving the filtered joint distribution, Equation 12.

### 322 3.1 Evidence and marginals

323 In order to compute efficiently the marginal likelihood (also known as evidence)  $P(Y_{0:t}|u_{1:t}; \alpha_{0:t})$  and the  
 324 filtered marginals  $P(A_t|Y_{0:t}, u_{1:t}; \theta_a), P(O|Y_{0:t}; \theta_o)$  we take advantage of the fact that only a very small

area in the joint distribution space will be affected by the measurement likelihood function at each time step. Without lost of generality the likelihood function will only make a difference to dependent  $A \cap O$  states in the joint distribution, states where the likelihood function is less than one. The states inside  $A \ominus O$  will not be affected, where the likelihood function is equal to one.

$$P(A_t, O|Y_{0:t}) = P_{\cap}(A_t, O|Y_{0:t}) + P_{\ominus}(A_t, O|Y_{0:t}) \quad (18)$$

This formulation will lead to large computational gain as the independent term is not influenced by the measurement function:  $P_{\ominus}(A_t, O, \mathbf{Y}_{0:t}) = P_{\ominus}(A_t, O, \mathbf{Y}_{0:t-1})$  and  $P_{\ominus}(A_t, O|Y_{0:t}) \propto P_{\ominus}(A_t, O|Y_{0:t-1})$ .

### 3.1.1 Evidence

The evidence of the measurement  $P(Y_{0:t}|u_{1:t}; \alpha_{0:t})$  is the normalisation coefficient of the joint distribution Equation 12. It is the amount of probability mass re-normalised to the other parameters as a result of the consecutive application of the likelihood function. At time step  $t$ , the normalising factor to be added to the evidence is the difference between the probability mass located inside  $A \cap O$  before and after the application of the measurement function  $P(Y_t|A_t, O)$ , see Equation 19-20 (see Appendix 8.4 for the full derivation).

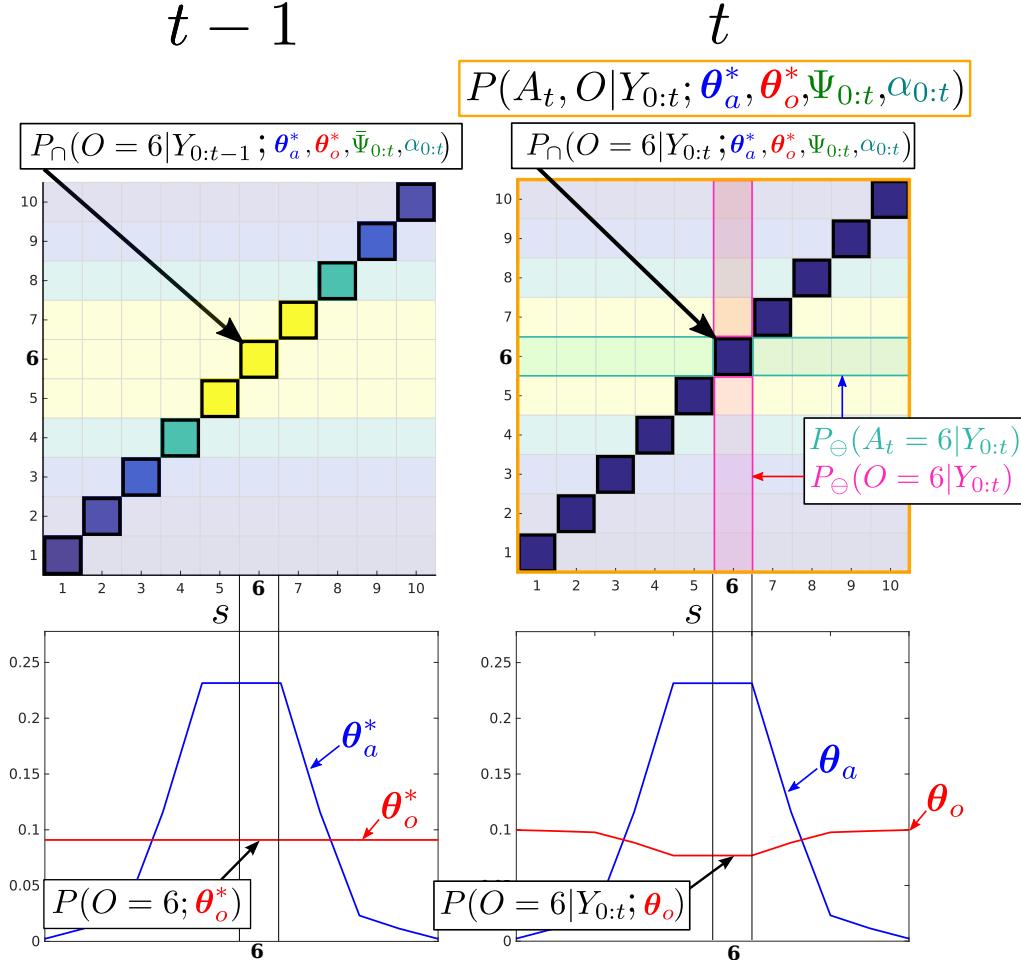
$$\alpha_t = \sum_{A_t} \sum_O \left( P(Y_t|A_t, O) - 1 \right) P_{\cap}(A_t, O, Y_{0:t-1}|u_{1:t}) \quad (19)$$

$$P(Y_{0:t}|u_{1:t}; \alpha_{0:t}) = 1 + \underbrace{\alpha_{0:t-1} + \alpha_t}_{\alpha_{0:t}} \quad (20)$$

The advantage of Equation 19 is that the summation is only over the states which are in the dependent area  $\cap$  of the joint distribution. Until an object is sensed, the likelihood will always be zero  $P(Y_t|A_t, O) = 0$  and  $\alpha_t$  will correspond to the probability mass which falls within the region of the joint distribution in which the likelihood function is zero. The point of interest is that as we perform the filtering process we will never re-normalise the whole joint distribution, but only keep track of how much it should have been normalised. To this end the marginals  $P(A_t|u_{1:t}; \theta_a^*)$  and  $P(O; \theta_o^*)$  are never re-normalised but are used at each step to compute how much of the probability mass  $\alpha_t$  should go to the normalisation factor  $P(Y_{0:t}|u_{1:t}; \alpha_{0:t})$ . The normalisation factor in question will never be negative, as the joint distribution sums to one and each  $\alpha_t$  represents some of the mass removed from the joint distribution. Since we keep track of the history of applied measurement likelihood functions the same amount of probability mass is never removed twice from the joint distribution.

### 3.1.2 Marginals

There are two different sets of marginals used in the MLMF filter. The first set are the **joint marginals** of the joint distribution, Equation 12 parameterised by  $\theta_a^*$  and  $\theta_o^*$ . The second set of marginals are the **filtered marginals** which are updated by evaluating the joint distribution in dependent states and are parameterised by  $\theta_a$  and  $\theta_o$ . At initialisation before the the first action or observation is made the parameters of the filtered marginal are set equal to those of the joint distribution. MLMF takes advantage of the sparsity of the likelihood function which results in only the dependent elements of the marginal being affected, Equation 21 (see Appendix 8.5 for the full derivation of Equation 21).



**Figure 7.** Filtered marginals. Illustration of the agent and object marginal update, Equation 21. The joint distribution parameters which are independent  $A \ominus O$  are pale and the dependent areas  $A \cap O$ , where  $P(Y_t < 1 | A_t, O)$ , are bright. MLMF only evaluates the joint distribution in dependent states. For each state  $s$  of the marginals  $1, \dots, 10$  the difference of the marginals inside the dependent area, before and after the measurement likelihood is applied, is evaluated and removed from the marginals  $P(A_t | Y_{0:t-1}, u_{1:t}; \theta_a)$ ,  $P(O | Y_{0:t-1}; \theta_o)$  leading to  $P(A_t | Y_{0:t}, u_{1:t}; \theta_a)$ ,  $P(O | Y_{0:t}; \theta_o)$  (we did not show  $u_{1:t}$  in the figure for ease of notation). *Bottom-left:* joint marginals  $P(A_t | u_{1:t}; \theta_a^*)$  and  $P(O | \theta_o^*)$  remain unchanged by measurements.

$$P(O | Y_{0:t}; \theta_o) = P(O | Y_{0:t-1}; \theta_o) - \left( P_{\cap}(O | Y_{0:t-1}) - P_{\cap}(O | Y_{0:t}) \right) \quad (21)$$

$$P_{\cap}(\mathbf{O} | \mathbf{Y}_{0:t}; \theta_a^*, \theta_o^*, \Psi_{0:t}, \alpha_{0:t}) = \sum_{A_t} P_{\cap}(\mathbf{A}_t, \mathbf{O} | \mathbf{Y}_{0:t}, \mathbf{u}_{1:t}; \theta_a^*, \theta_o^*, \Psi_{0:t}, \alpha_{0:t})$$

$$= \frac{\sum_{A_t} P_{\cap}(O; \theta_o^*) P_{\cap}(A_t | u_{1:t}; \theta_a^*) P(Y_{0:t} | A_t, O, u_{1:t}; \Psi_{0:t})}{P(Y_{0:t} | u_{1:t}; \alpha_{0:t})} \quad (22)$$

357 Equation 21 is recursive,  $P(O|Y_{0:t}; \theta_o)$  is computed in terms of  $P(O|Y_{0:t-1}; \theta_o)$ . Figure 7 illustrates  
 358 a measurement update of the MLMF. The illustrated marginals (*Bottom row*) are (on the **left**) the  
 359 **joint marginals**  $P(A_t|u_{1:t}; \theta_a^*)$ ,  $P(O; \theta_o^*)$  and (on the **right**) the **filtered marginals**  $P(A_t|Y_{0:t}, u_{1:t}; \theta_a)$ ,  
 360  $P(O|Y_{0:t}; \theta_o)$ .

361 The shape of the **joint marginals** remain unchanged by measurements during the filtering process, they  
 362 are the parameters of the joint distribution used to update the filtered marginals. Table 1 summarises the  
 363 functions and parameters of the MLMF for two random variables, an agent and object.

functions	parameters	description
$P(A_t Y_{0:t}, u_{1:t})$	: $\theta_a$	filtered marginals
$P(O Y_{0:t})$	: $\theta_o$	
$P(A_t u_{1:t})$	: $\theta_a^*$	joint marginals
$P(O)$	: $\theta_o^*$	
$P(Y_{0:t} u_{1:t})$	: $\alpha_{0:t} \in \mathbb{R}$	evidence
$P(Y_{0:t} A_t, O, u_{1:t})$	: $\Psi_{0:t} = \{(Y_i, l_i)\}_{i=0:t}$	likelihood history

**Table 1.** MLMF functions with associated parameters. The marginal parameters are the discretisation of the state space  $\theta \in \mathbb{R}^N$ ,  $\theta^{(s)}$  correspond to the probability being in state  $s$ .

364 We evaluated the MLMF with Histogram-SLAM in the case of the 1D filtering scenario illustrated in  
 365 Figure 5 on page 11 and we found them to be identical. Having respected the formulation of Bayes rule,  
 366 we assert that the MLMF filtering steps (see Algorithm 2, Appendix 8.1 for a more detailed application of  
 367 motion-measurement update steps) are Bayesian Optimal Filter<sup>1</sup>. Next we evaluate both space and time  
 368 complexity of the MLMF filter.

## 4 SPACE & TIME COMPLEXITY

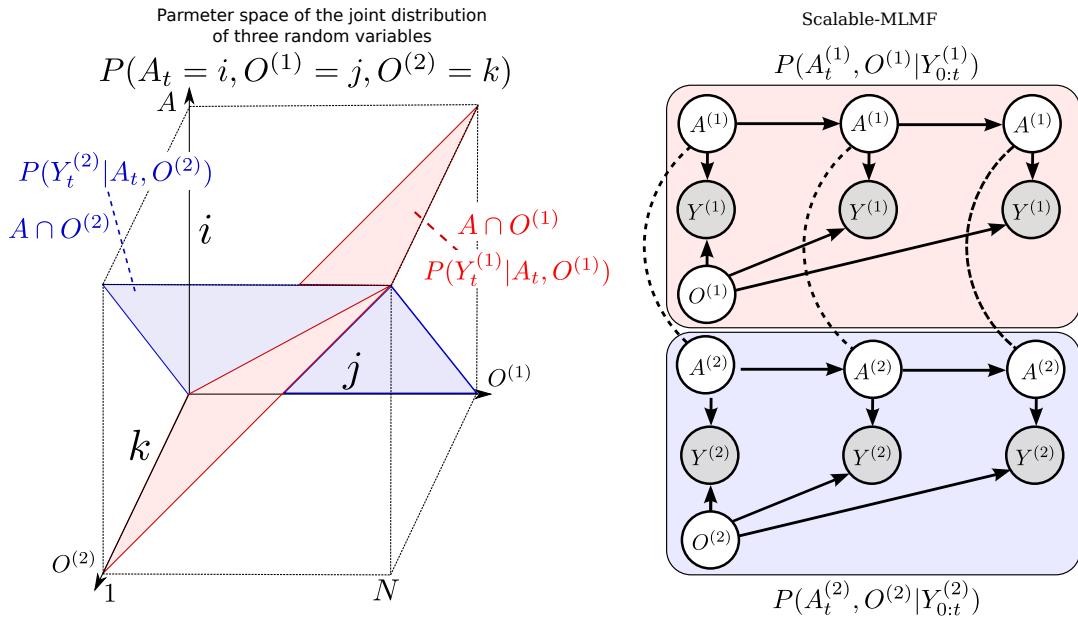
369 For discussion purposes we consider the case of three beliefs, namely that of the agent and two other  
 370 objects  $O^{(1)}$  and  $O^{(2)}$ , we subsequently generalise. As stated previously  $M$  stands for the number of filtered  
 371 random variables including the agent and  $N$  is the number of discrete states in the world. In the following  
 372 section, we compare the space and time complexity of MLMF-SLAM with Histogram-SLAM.

### 4.1 Space complexity

374 Figure 8 *Left* illustrates the volume occupied by the joint distribution for a space with  $N$  states. Histogram-  
 375 SLAM would require  $N^3$  parameters for the joint distribution  $P(A, O^{(1)}, O^{(2)}; \theta)$  and  $3N$  parameters to  
 376 store the marginals. In general for  $M$  random variables  $N^M + M N$  parameters are necessary, give a space  
 377 complexity of  $\mathcal{O}(N^M)$ .

378 For MLMF-SLAM, each random variable requires two sets of parameters,  $\theta$  and  $\theta^*$  (see Table 1). Given  
 379  $M$  random variables, the initial number of parameters is  $M(2N)$ . At every time step the likelihood memory  
 380 function increments by one measurement and offset,  $(Y_t, l = 0)$  (Algorithm 1). Given a state space of size  
 381  $N$ , there can be no more than  $N$  different measurement functions (one for each state). In the worst case  
 382 scenario the number of memory likelihood function parameters  $\Psi_{0:t}$ , Equation 15, will be  $N$ . The total  
 383 number of parameters is  $M(2N) + N$  which gives a final worst case space complexity linear in the number  
 384 of random variables,  $\mathcal{O}(MN)$ .

<sup>1</sup> An optimal Bayesian solution is an exact solution to the recursive problem of calculating the exact posterior density Arulampalam et al. (2002)



**Figure 8.** *Left:* Joint distribution  $P(A, O^{(1)}, O^{(2)})$  of the agent and two objects ( $Y_{0:t}$  and  $u_{1:t}$  omitted). Each likelihood function,  $P(Y|A, O^{(1)})$ ,  $P(Y|A, O^{(2)})$  corresponds to a hyperplane in the joint distribution. The state space is discretised to  $N$  bins giving a potential total of  $N^3$  parameters for the joint distribution (Histogram case). *Right:* **Scalable-MLMF** Each agent-object joint distribution pair is modelled independently. For clarity we have left out the action random variable  $u$  which is linked to every agent node. Two joint distributions  $P(A^{(1)}, O^{(1)}|Y_{0:t}^{(1)})$  and  $P(A^{(2)}, O^{(2)}|Y_{0:t}^{(2)})$  parametrise the graphical model. The dashed undirected lines represent a wanted dependency, if present  $O^{(1)}$  and  $O^{(2)}$  are to be dependent through  $A$ . In the standard setting there will be no exchange of information between the individual joint distributions. However we demonstrate later on how we perform a one time transfer of information when one of the objects is sensed.

## 385 4.2 Time complexity

386 For Histogram-SLAM, the computational cost is equivalent to that of the space complexity,  $\mathcal{O}(N^M)$ ,  
387 since every state in the joint distribution has to be summed to obtain all the marginals.

388 For MLMF-SLAM, every state in the joint distribution's state space which has been changed by the  
389 likelihood function has to be summed, see Figure 7 on page 16. As a result the computational complexity  
390 is directly related to the number of dependent states  $|A \cap O|$ . In Figure 7, this corresponds to states where  
391  $i = j$  and there are  $N$  out of a total  $N^2$  states for that joint distribution. Figure 8 (Left) illustrates a joint  
392 distribution with  $N^3$  states. The dependent states  $|A \cap O^{(1)} \cap O^{(2)}|$  are those which are within the blue and  
393 red planes (where the likelihood evaluates to zero) and comprise  $N^2$  states each, giving a total of  $2N^2 - N$   
394 dependent states (negative is to remove the states we count twice at the intersection of the blue and red  
395 plane).

396 The likelihood term  $P(Y_t|A_t, O^{(1)})$  evaluates states to zero which satisfy  $(i = j, \forall k)$ , as the  
397 measurement of object  $O^{(1)}$  is independent of object  $O^{(2)}$ . With 3 objects, the joint distribution  
398 would be  $P(A_t = i, O^{(1)} = j, O^{(2)} = k, O^{(l)} = l)$  then the likelihood  $P(Y_t|A_t, O^{(1)})$  evaluated to zero  
399 for  $(i = j, \forall k, \forall l)$  which would mean  $N^3$  dependent states. In general, for  $M$  random variables the  
400 computational cost is  $(M - 1)N^{M-1}$  which gives  $\mathcal{O}(N^{M-1})$  as opposed to the Histogram-SLAM's  
401  $\mathcal{O}(N^M)$ . The computation complexity in this setup is still exponential but to the order  $M - 1$  as opposed  
402 to  $M$  which nevertheless quickly limits the scalability as more objects are added.

403 Computing the value of a dependent state  $(i, j, k)$  in the joint distribution required evaluating Equation  
 404 12 which contains a product of  $N$  likelihood functions, in the worst case scenario. However the likelihood  
 405 functions are not overlapping and binary. As a result the complete product does not have to be evaluated  
 406 since only one likelihood function will effect the state  $(i, j, k)$ . Thus evaluating Equation 12 yields a cost  
 407 of  $\mathcal{O}(1)$  and **not**  $\mathcal{O}(N)$ .

## 5 SCALABLE EXTENSION TO MULTIPLE OBJECTS

408 To make the MLMF filter scalable we introduce an **independence assumption** between the objects and  
 409 model the joint distribution (Equation 23) as a product of agent-object joint distributions:

$$P(A_t, O^{(1)}, \dots, O^{(M-1)} | Y_{0:t}, u_{1:t}) = \prod_{i=1}^{M-1} P(A_t^{(i)}, O^{(i)} | Y_{0:t}^{(i)}, u_{1:t}) \quad (23)$$

410 The measurement variable  $Y_t$ , is the vector of all agent-object measurements,  $Y_t = [Y_t^{(1)}, \dots, Y_t^{(M-1)}]^T$ .  
 411 Each agent-object joint distribution has its own parametrisation of the agent's marginal,  $A_t^{(1)}, \dots, A_t^{(M-1)}$   
 412 which combine to give the overall marginal of the agent  $A_t$ . The computation of each object marginal  
 413  $P(O^{(i)} | Y_{0:t}^{(i)})$  is independent of the other objects. This is evident from the marginalisation see Equation  
 414 24-25.

$$P(O^{(i)} | Y_{0:t}^{(i)}, u_{1:t}) = \sum_{A_t^{(i)}} P(A_t^{(i)}, O^{(i)} | Y_{0:t}^{(i)}, u_{1:t}) \quad (24)$$

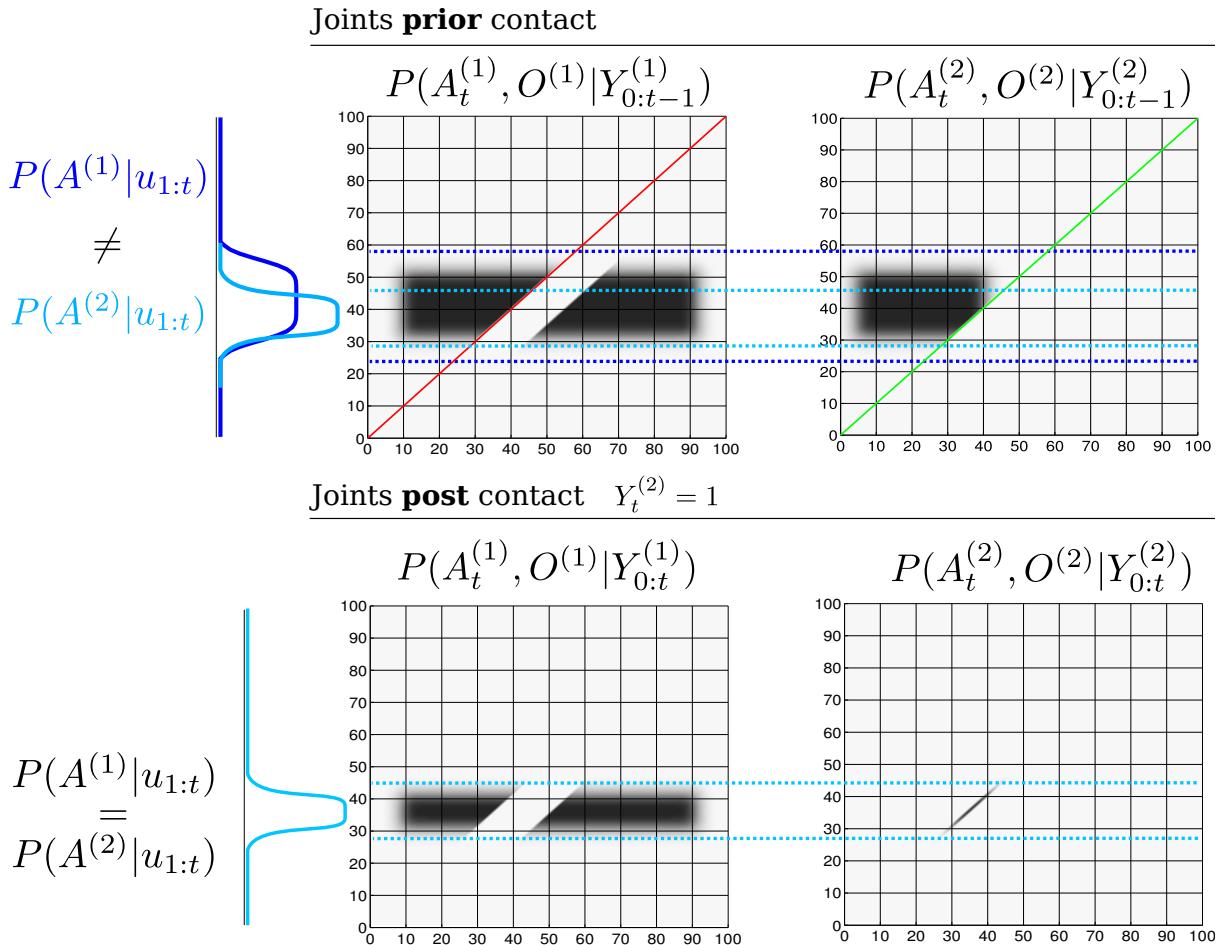
$$P(A_t | Y_{0:t}, u_{1:t}) = \prod_{i=1}^{M-1} P(A_t^{(i)} | Y_{0:t}^{(i)}, u_{1:t}) \quad (25)$$

415 The independence assumption will create an unwanted effect with respect to agent's marginal  
 416  $P(A_t | Y_{0:t}, u_{1:t})$ . At initialisation the agent marginals should be equal,  $P(A_0 | Y_0) = P(A_0^{(i)} | Y_0^{(i)}) \forall_i$ ,  
 417 however this is not the case because of Equation 25. To overcome this we define the marginal,  
 418  $P(A_t | Y_{0:t}, u_{1:t})$ , of the agent as being the average of all the individual pairs  $P(A_t^{(i)} | Y_{0:t}^{(i)}, u_{1:t})$ .

$$P(A_t | Y_{0:t}, u_{1:t}) := \frac{1}{M-1} \sum_{i=1}^{M-1} P(A_t^{(i)} | Y_{0:t}^{(i)}, u_{1:t}) \quad (26)$$

419 Figure 8 (*Right*), depicts the graphical model of the scalable formulation. As each joint distribution pair  
 420 has its own parametrisation of the agent's marginal and these do not subsequently get updated by one  
 421 another, the information gained by one joint distribution pair is **not transferred**. A solution is to transfer  
 422 information between the marginals  $A^{(i)}$  at specific intervals namely when one of the objects is sensed by  
 423 the agent.

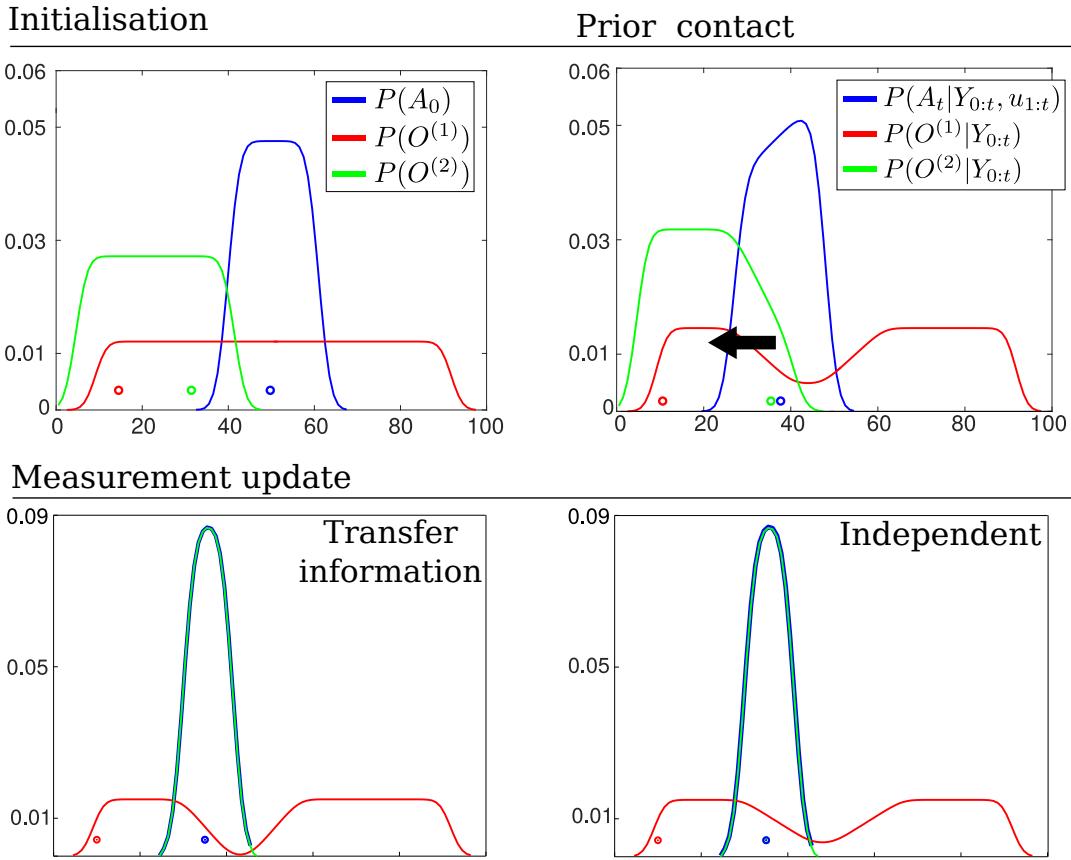
424 The exchange of information of one joint distribution to another is achieved through the agent's marginals  
 425  $A^{(i)}$  according to Algorithm 3, Appendix 8.2. The measurement update is the same as previously described



**Figure 9. Transfer of information (joint distributions)** *Top:* Joint distributions of  $P(A_t^{(1)}, O^{(1)}|Y^{(1)})$  and  $P(A_t^{(2)}, O^{(2)}|Y^{(2)})$  prior sensing,  $Y_t^{(2)} = 1$ , see Figure 10 (*Top right*) for the corresponding marginals. The red and green lines across the joint distributions correspond to the region in which the likelihood functions  $P(Y_t^{(1)}|A_t^{(1)}, O^{(1)})$  and  $P(Y_t^{(2)}|A_t^{(2)}, O^{(2)})$  will change the joint distributions. The dotted blue lines are to ease the comparison of the joint distributions prior and post sensing. *Bottom right:* After the agent has sensed  $O^{(2)}$ , all the probability mass which was not overlapping the green line becomes an infeasible solution to the agent and object locations. At this point the marginals  $P(A_t^{(1)}|u_{1:t}) \neq P(A_t^{(2)}|u_{1:t})$  are no longer equal (see the blue marginals *Top*). *Bottom left:* The constraint imposed by the likelihood function of the second object (green line) is transferred to the joint distribution of the first object according to Algorithm 3. This results in a change in the joint distribution  $P(A_t^{(1)}, O^{(1)}|Y^{(1)})$ , which satisfies the constraints imposed by the agent's marginal from the joint distribution  $P(A_t^{(2)}, O^{(2)}|Y^{(2)})$ .

426 in Algorithm 2 in the case of no positive measurements of the objects. If the agent senses an object, all of  
 427 the agent marginals of the remaining joint distributions are set to the marginal of the joint distribution pair  
 428 belonging to the positive measurement  $Y_t^{(i)}$ .

429 Figure 9, depicts the process of information exchange between object  $O^{(1)}$  and  $O^{(2)}$  in the event  
 430 that the agent senses  $O^{(2)}$ . Prior to the positive detection, both marginals  $P(A_t^{(1)}|Y_{0:t-1}^{(1)}, u_{1:t})$  and  
 431  $P(A_t^{(2)}|Y_{0:t-1}^{(2)}, u_{1:t})$  occupy the same region and are identical. When the agent senses  $O^{(2)}$  the line defined  
 432 by the measurement likelihood function  $P(Y_t^{(2)}|A_t^{(2)}, O^{(2)})$  becomes a hard constraint implying that both



**Figure 10. Transfer of information (marginals)** *Top left:* Initial beliefs of the agent and object’s location. The agent moves to the left until it senses object  $O^{(2)}$ . *Top right:* Marginals prior the agent entering in contact with the green object, see Figure 9 (Top) for an illustrate of the joint distributions. *Bottom left:* resulting marginals after setting the agent marginals of each joint distribution equal  $A_t^{(1)} = A_t^{(2)}$  according to Algorithm 3. The object marginal  $P(O^{(2)}|Y_{0:t})$  is recomputed. *Bottom Right:* resulting marginals in which the objects have no influence on one another. Note that a transfer of information has caused a change in the marginal  $O^{(1)}$ .

433 the agent and  $O^{(2)}$  have to satisfy this constraint. Figure 10 shows marginals at initialisation, prior contact  
 434 between the agent and object and the after the measurement (post contact) has been integrated into the  
 435 marginals (resulting from the joint distributions in Figure 9).

436 The result of introducing a dependency between the objects through the agent’s marginals in the event of  
 437 a sensing and treating them independently gives the same solution as the histogram filter in this particular  
 438 case. However as each individual marginal  $A_t^{(i)}$  diverges from the other marginals, the filtered solution  
 439 will diverge from the histogram’s solution. We assume however that the objects are weakly dependent  
 440 and sharing information during positive sensing events is sufficient. In section 6.2 we will evaluate the  
 441 independence assumption with respect to the histogram filter.

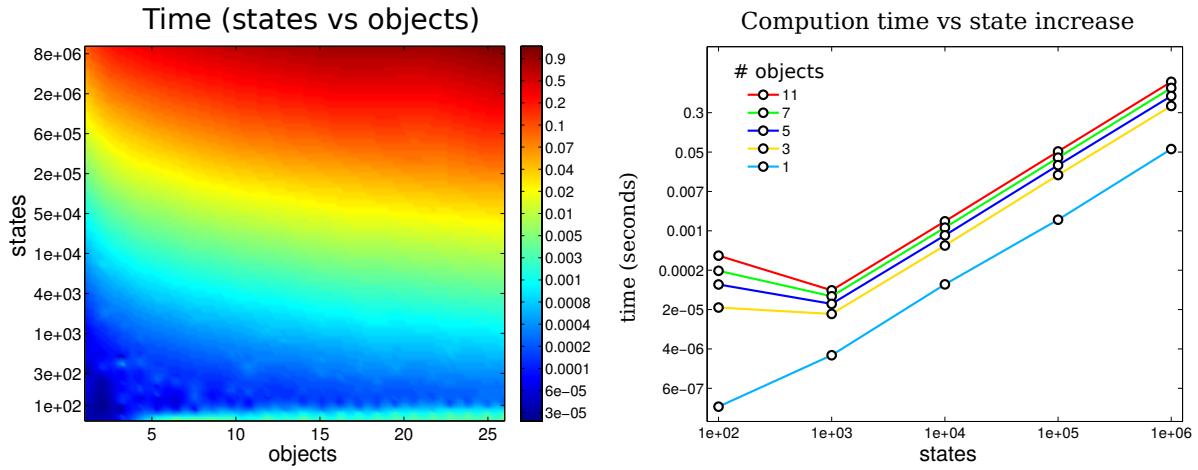
442 Table 2 summarises the time and space complexity for the three filters.

## 6 EVALUATION

443 We conduct three different types of evaluation to quantify the scalability and correctness of the scalable-  
 444 MLMF filter. The first experiment tests the scalability of our filter in terms of processing time taken per

	space	time
Histogram	$\mathcal{O}(N^M)$	$\mathcal{O}(N^M)$
MLMF	$\mathcal{O}(MN)$	$\mathcal{O}(N^{(M-1)})$
scalable-MLMF	$\mathcal{O}(MN)$	$\mathcal{O}(MN)$

**Table 2. Time and space complexity summary** For both MLMF and scalable-MLMF the worst case scenario is reported for the space complexity.



**Figure 11. Time complexity:** *left*: mean time taken for a loop update (motion and measurement) as a function of the number of states in a marginal and the number of objects present. *right*: time taken for a loop update with respect to the number of states in the marginal. The colour coded lines are associated with the number of objects present. The computational cost is plotted on a log scale. As the number of states increases exponentially the computational cost matches it.

445 motion-measurement update cycle. The second experiment evaluates the independence assumption made  
 446 in the scalable-MLMF filter between the objects. The third and final experiment determines the effect of  
 447 the memory size on a search policy to locate all the objects in the *Table* world.

## 448 6.1 Evaluation of time complexity

449 We measured the time taken by the motion-measurement update loop, as a function of the number  
 450 beliefs and number of states per belief. We started with a 100 states per belief and gradually increase it  
 451 to 10'000'000 over 50 steps. Each of the 50 steps treated 2 to 25 objects. Figure 11 *left* illustrates the  
 452 computational cost as a function of number of states and objects. For each state-object pair 100 motion-  
 453 measurement updates were performed. Most of the trials returned time updates below 1 Hz. Figure 11 *right*  
 454 shows the computational cost as a function of the number of states plotted for 6 different filter runs with a  
 455 different number of objects. As the number of states increases exponentially so does the computational  
 456 cost. Note the cost increases at the same rate as the number of states meaning that the computational  
 457 complexity is linear with respect to the number of states. This result is in agreement with the asymptotic  
 458 time complexity.

## 459 6.2 Evaluation of the independence assumption

460 In section 5 we made the assumption (for scalability reasons) that the objects' beliefs are independent  
 461 of one another. This assumption is validated by comparing the MLMF filter on three random variables,  
 462 an agent and two objects, with the ground truth which we obtain from the standard histogram filter. For

463 each of the three beliefs (the agent and two objects), 100 different marginals were generated and the true  
 464 locations (actual position of the agent and objects) were sampled. Figure 12 *Top-left* illustrates one instance  
 465 of the initialisation of the agent and object marginals with their associated sampled true position. The agent  
 466 carries out a sweep of the state space for each of the marginals and the policy is saved and run with the  
 467 scalable-MLMF filter. In the first experiment we assumed that the objects are completely independent and  
 468 that there was no transfer of information between the pair-wise joint distributions. In the second and third  
 469 experiments there is an exchange of information as described in Algorithm 3. Here we compare the effect  
 470 of using the product of the agent’s marginals, Equation 25, with the average of the marginals, Equation 26.  
 471 We expect the average of the the agent’s marginal to yield a result closer to the ground truth as the marginal  
 472 of the agent  $P(A_t|Y_{0:t}, u_{1:t})$  at initialisation is the same as the ground truth (the Histogram-SLAM’s). As  
 473 for the marginal of the objects  $P(O^{(i)}|Y_{0:t})$  we expect the difference between them to be independent of  
 474 whether the product or average of the agent’s marginal is used. This results from Algorithm 3. When an  
 475 object  $i$  is sensed all the corresponding agent marginals  $P(A^{(j)}|u_{1:t})$  are set equal to  $P(A^{(i)}|u_{1:t})$  and not  
 476 to  $P(A_t|Y_{0:t}, u_{1:t})$ . This is a design decision of our information transfer heuristic. There are many other  
 477 possibilities but this is one of the simplest. For each of the 100 sweeps the ground truth is compared with  
 478 the scalabe-MLMF using the Hellinger distance (Equation 27)

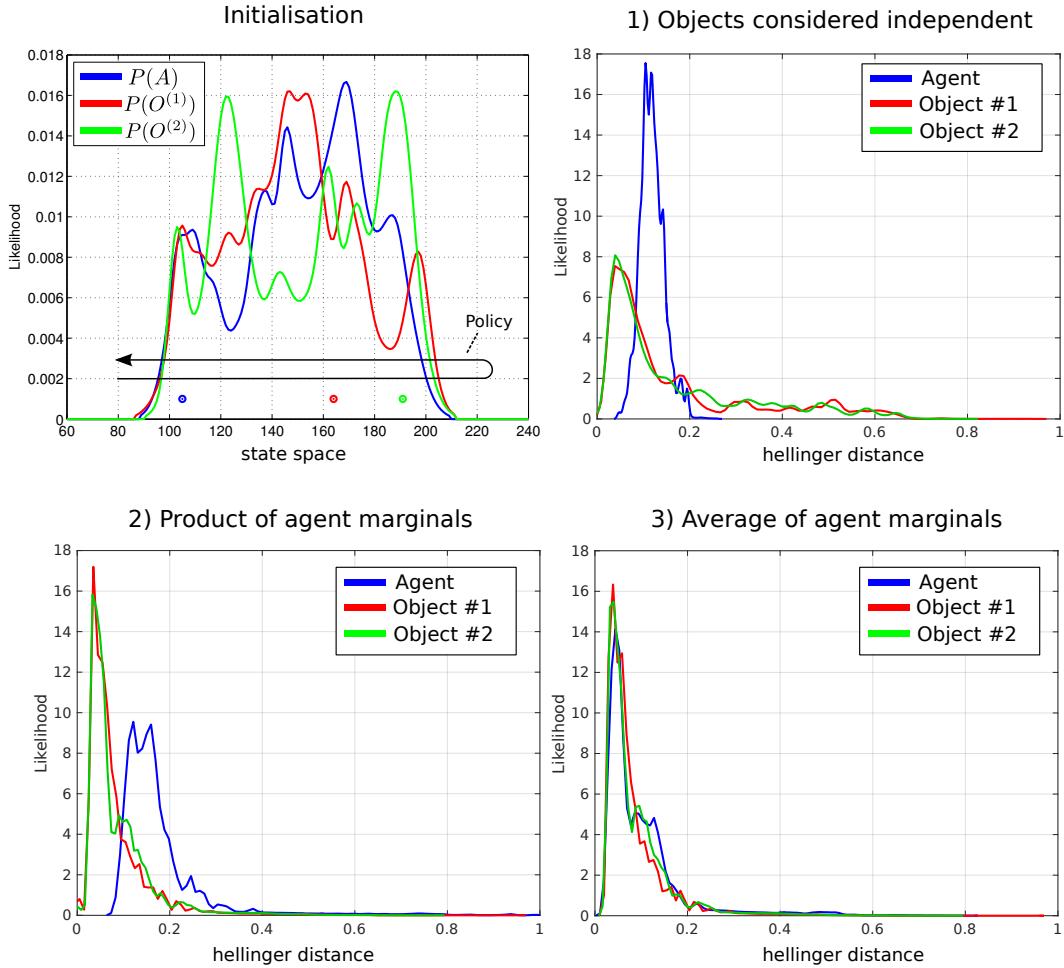
$$H(P, Q) = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2 \quad (27)$$

479 which is a metric which measures the distance between two probability distributions. Its value lies strictly  
 480 between 0 (the two distributions are identical) and 1 (no overlap between them). Figure 12 shows the  
 481 kernel density distribution of the Hellinger distances taken at each time step for all 100 sweeps. In the  
 482 *Top-left* of the figure, for the case when no transfer of information is applied, all the marginals are far  
 483 from the ground truth. This results from the introduction of the independence assumption, necessary to  
 484 scale the MLMF. Figure 12 *Bottom* shows the results for difference between the product and average of  
 485 the agents marginals. As expected there is no difference between the objects’ marginals when considering  
 486 both methods (product and average) with respect to the ground truth. The predominant difference occurs in  
 487 the agent’s marginal  $P(A_t|Y_{0:t}, u_{1:t})$ . This is also expected and prompted the introduction of the average  
 488 method instead of the product.

489 The scalable-MLMF information exchange heuristic will not lead to any of the objects marginals  
 490 probability mass being falsely removed during the information transfer, which is close to a winner-take-all  
 491 approach in terms of beliefs. When object  $i$  is sensed its associated agent marginal is set to all other  
 492 agent-object joint pairs, which results in the information accumulated in the  $j$ th agent marginals being  
 493 replaced by the  $i$ th.

### 494 6.3 Evaluation of memory

495 The memory measurement likelihood function  $P(Y_{0:t}|A_t, O, u_{1:t}; \Psi_{0:t})$  is parameterised by the history  
 496 of all the measurement likelihood functions which have been applied on the joint distribution since  
 497 initialisation. As detailed previously there can be no more than  $|\Psi_{0:t}| \leq N$  different measurement  
 498 likelihood functions added to memory. In the case of a very large state space this might be cumbersome.  
 499 We investigate how restricting the memory size, the number of parameters  $|\Psi_{0:t}|$ , can impact on the  
 500 decision process in an Active-SLAM setting. Given our set up a breadth-first search in the action space is  
 501 chosen with a one time step horizon, making it a greedy algorithm. The objective function utilised is the  
 502 information gain of the beliefs after applying an action, Equation 28.

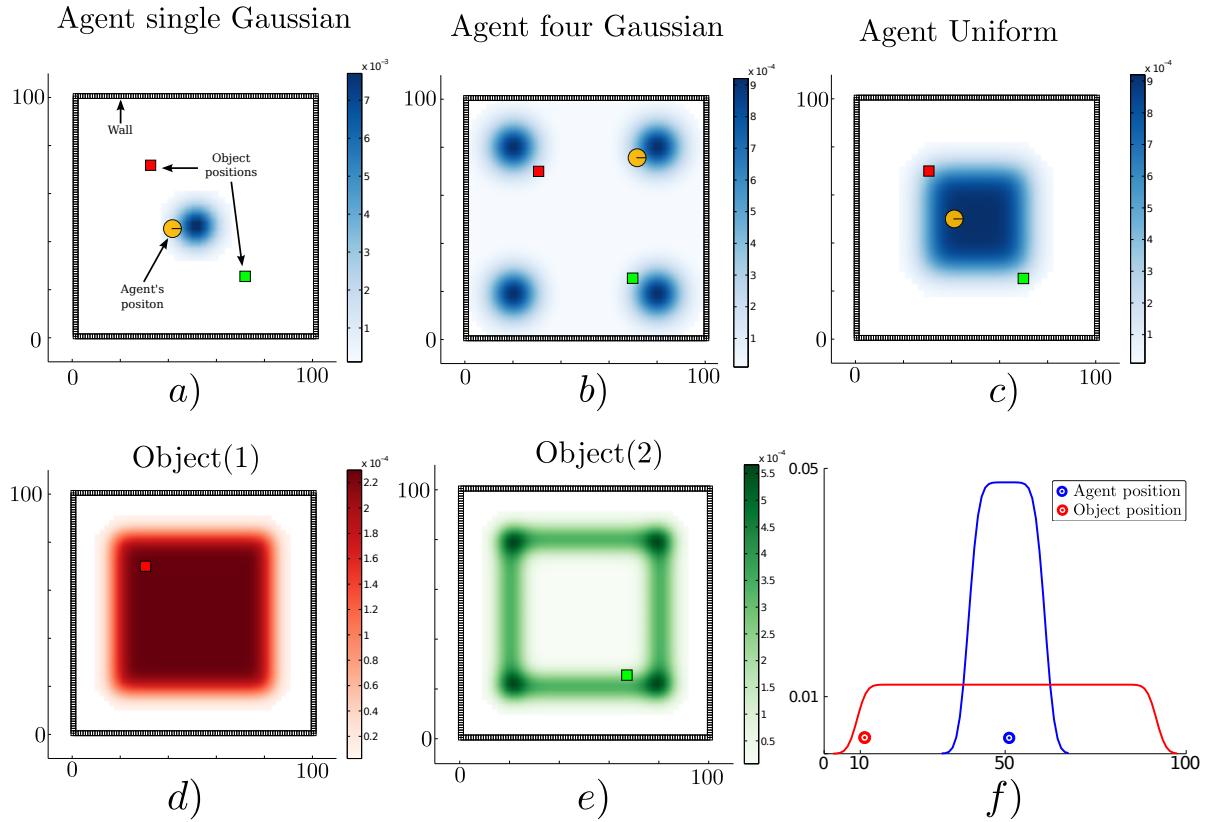


**Figure 12. Comparison of scalable-MLMF and the histogram filter** A deterministic sweep policy was carried out for 100 different initialisations of the agent and object beliefs. **Top left:** One particular Initialisation of the agent and object random variables. The true position of the agent and objects were sampled at random. The black arrow indicates the general policy which was followed for each of the 100 sweeps. These were performed for **1)** scalable-MLMF with objects considered to be independent at all times (no Algorithm 3). **2)** Agent marginal  $P(A_t|Y_{0:t}, u_{1:t})$  is the product of marginals  $P(A_t^{(i)}|Y_{0:t}^{(i)}, u_{1:t})$ , Equation 25. **3)** marginal  $P(A_t|Y_{0:t}, u_{1:t})$  is taken to be the average of all marginals  $P(A_t^{(i)}|Y_{0:t}^{(i)}, u_{1:t})$ , Equation 26. For each of these three experiment we report the kernel density estimation over the Hellinger distances taken at every time step between ground truth (from histogram filter) and scalable-MLMF.

$$u_t = \arg \max_{u_t} H\{P(A_{t-1}, O|Y_{0:t-1}, u_{1:t-1})\} - \mathbb{E}_{Y_t} [H\{P(A_t, O|Y_{0:t}, u_{1:t})\}] \quad (28)$$

503 For each action the filter is run forward in time and all future measurements since we cannot know ahead  
 504 of time the actual measurement. The information gain is the difference between the current entropy (defined  
 505 by  $H\{\cdot\}$ ) and the future entropy after the simulated motion and measurement update. The action with the  
 506 highest information gain is subsequently selected. This is repeated at each time step. Figure 13 illustrates  
 507 the environment setup for a 1D and 2D case. The agent's task is to find the objects in the environment.

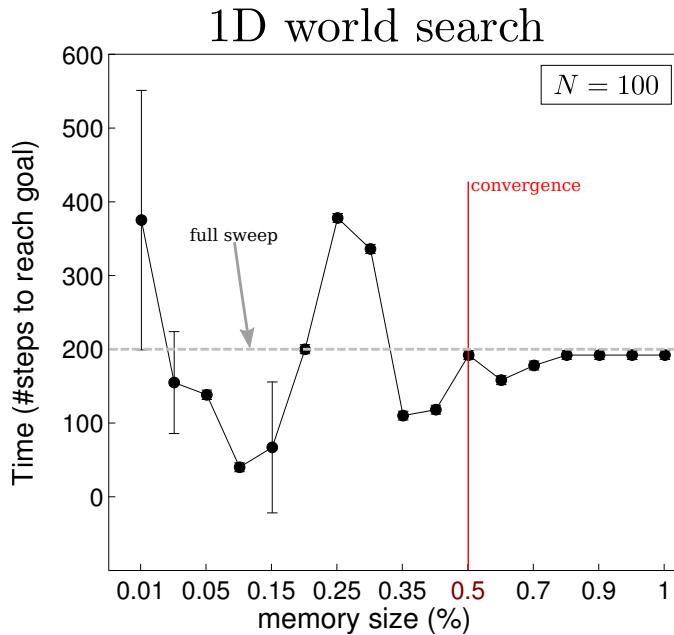
508 For the 2D search we consider three different initialisations (single-Gaussian, four-Gaussian, Uniform)  
 509 for the agent's belief where there are two objects to be found. Ten searches are carried out for each of the



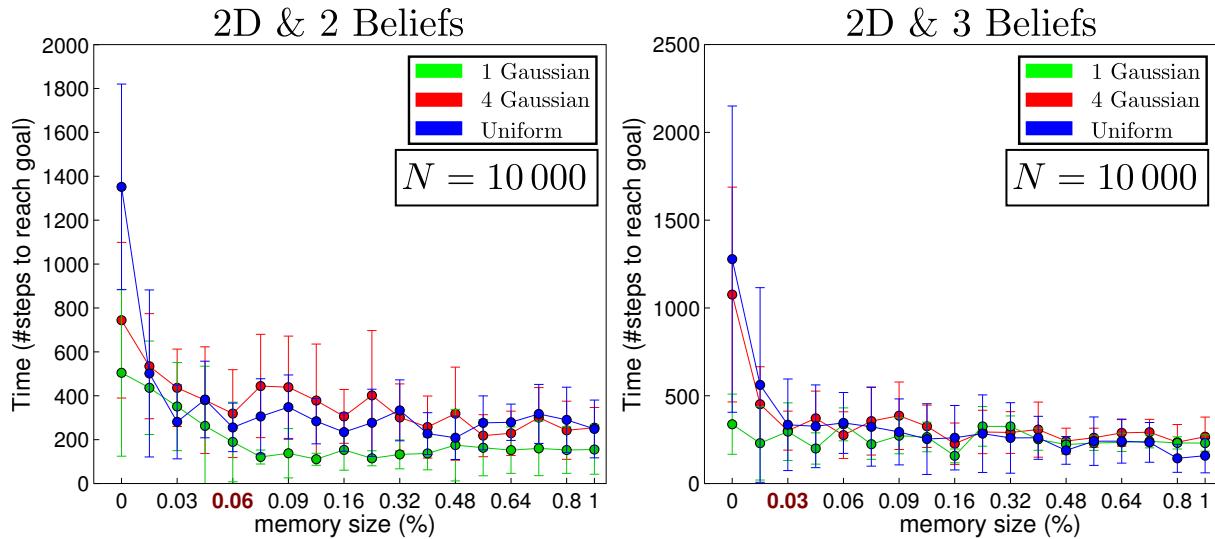
**Figure 13. Agent's prior beliefs.** Two types of environment, the first is a 2D world where the agent lives in a square surrounded by a wall whilst the second is a 1D world. In the 2D figures the agent is illustrated by a circle with a bar to indicate its heading. The true location of the objects are represented by colour coded squares. *Top row* three different initialisations of the agent's location. *Bottom row* d) the agent's prior beliefs with respect to the location of the first object and e) belief of the second object's location. *bottom row* f) 1D world with one object.

510 three initialisations of the agent's beliefs. The agent's true location, for each search, is sampled from its  
 511 initial belief, and the objects' locations (red and green squares in Figure 13) are kept fixed throughout all  
 512 searches. Each search is repeated for 18 different memory sizes ranging from 1 to  $N$  (the number of states).  
 513 For the 1D search case one object is considered since adding more objects makes the search easier and the  
 514 interest lies in the memory effects of the search and not the search itself. In Figures 14-15 we report on the  
 515 time taken to find all objects with respect to a given memory size which is shown as the percentage of the  
 516 total number of states. In the 1D search case the time variability taken to find the object converges when  
 517 the memory size is at 60% of the original state space. As for the 2D search with 2 beliefs (agent & 1 object)  
 518 the convergence depends on the agent's initial belief. For the 1-Gaussian (green line) all searches take  
 519 approximately the same amount of time after a memory size of 9%. As for the remaining two initialisations  
 520 convergence is achieved at 48%. The same holds true for the case of 3 beliefs (agent & 2 objects).

521 In the 2D searches, the memory size has a less impact on the time taken to find the objects than in the 1D  
 522 (which is a special search case). Only when the memory size is less than 6% is there a significant change.  
 523 We conclude that at least in the case of the greedy one step-look ahead planner which is frequently used  
 524 in the literature, the size of the memory seems not to be a limiting factor in terms of the time taken to  
 525 accomplish the search.



**Figure 14. Memory size vs time to find object in 1D** Results of the effect of the memory size on the decision process for the 1D search illustrated in Figure 13*f*). The memory size is reported as the percentage of total number of states present in the marginal space. At 100% the size of the memory is equal to that of the state space,  $N = 100$  in this case. A total sweep of the entire state space would result in a total of 200 steps, the dotted grey line in the above figure. When no restrictions are placed on the memory size the policy following the greedy approach takes around 180 steps. This result converges when the number of parameters  $|\Psi_{0:t}|$  of the memory likelihood function is greater than 50% of the original state space.



**Figure 15. Memory size vs time to find objects in 2D.** The initial beliefs correspond to those of Figure 13, a) for Gaussian (green line), b) 4 Gaussians (red line) and c) Uniform (blue line), both objects are initialised according to d) and e).

## 7 CONCLUSION

This work addresses the Active-SLAM filtering problem for scenarios in which sensory information relating to the map is very limited. Current SLAM algorithms filter the errors originating from sensory measurements and not prior uncertainty. By making the assumption that the joint distribution of all the

529 random variables is a multivariate Gaussian, inference is tractable. Since the origin of the uncertainty does  
530 not originate from the measurement noise, no assumption can be made about the structure of the joint  
531 distribution. In this case a suitable filter would be the histogram which makes no assumption about the  
532 shape or form taken by the joint distribution. However, the space and time complexity are exponential with  
533 respect to the number random variables and this is a major limiting factor for scalability.

534 The main contribution of this work is a formulation of a histogram Bayesian state space estimator in  
535 which the computational complexity is both linear in time and space. A different approach to other SLAM  
536 formulations as been taken in the sense that the joint distribution is not explicitly parameterised avoiding the  
537 exponential increase in parameter space which would otherwise have been the case. The MLMF parameters  
538 consist of the marginals and the history of measurement functions which have been applied. By solely  
539 evaluating the joint distribution at the states which are affected by the current measurement function whilst  
540 taking into account the memory, the MLMF filter obtains the same filtered marginals as the histogram  
541 filter. Further, the worst case space complexity is linear rather than exponential and the time complexity  
542 remains exponential but increases at lower rate than in the histogram filter. In striving to make the filter  
543 scalable we make the assumption that the objects are independent. An individual MLMF is used for each  
544 agent-object pair. We evaluate the difference between the scalable-MLMF with a ground truth provided by  
545 the histogram filter for 100 different searches with respect to the Hellinger distance. We conclude that the  
546 divergence is relatively small and thus the scalable-MLMF filter provides a good approximation to the true  
547 filtered marginals. We evaluate the time taken to perform a motion-update loop for different discretisations  
548 of the state space (100 to 10'000'000 states) and number of objects (2 to 25). In most of the cases we  
549 achieve an update cycle rate below 1Hz. We evaluate how the increase of the number of states effects the  
550 computational cost and find the relationship to be linear and thus in agreement with our analysis of the  
551 asymptotic growth rate. We analyse the effect of the memory size (the remembered number of measurement  
552 likelihood functions) on the decision theoretic process of reducing the uncertainty of the map and agent  
553 during a search task. We conclude that in the 2D case the memory size has much less effect than in the 1D  
554 case and that it is unnecessary to remember every single measurement function.

555 This implies that the MLMF and scalable-MLMF that we have are a computationally tractable means  
556 of performing SLAM in a case scenario in which mostly negative information is present and the joint  
557 distribution cannot be assumed to have any specific structure. Furthermore, the filter can be used at a higher  
558 cognitive level than the processing of raw sensory information as is often the case in Active-SLAM. MLMF  
559 would be well suited for reasoning tasks where the robot's field of view is limited.

560 An interesting future extension could be to make the original MLMF filter scalable without introducing  
561 assumptions. One possibility could to be to consider Monte Carlo integration methods for inference. These  
562 can scale well to high dimensional spaces whilst still providing reliable estimates. A second possibility  
563 could be to investigate the use of Gaussian Mixtures as a form of parameterisation of the marginals to blend  
564 our filter with EKF-SLAM. This would allow the parameters to grow quadratically with respect to the  
565 dimension of the marginal space as opposed to exponentially as is the case with the histogram and MLMF  
566 filters.

## 8 APPENDIX

### 567 8.1 MLMF Algorithm

---

#### Algorithm 2: MLMF-SLAM

---

**input :**

**measurements**

$\mathbf{Y}_t, \mathbf{u}_t$

**joint distribution parameters:**

$P(A_{t-1}|u_{1:t-1}) P(O), \Psi_{0:t-1}, \alpha_{0:t-1}$

**filtered marginals:**

$P(A_{t-1}|Y_{0:t-1}, u_{1:t-1}), P(O|Y_{0:t-1})$

**output :**

**joint parameters:**

$P(A_t|u_{1:t}), \Psi_{0:t}, \alpha_{0:t}$

**filtered marginals:**

$P(A_t|Y_{0:t}, u_{1:t}), P(O|Y_{0:t})$

---

#### initialisation

$$P(A_0; \theta_a) := P(A_0; \theta_a^*)$$

$$P(O; \theta_o) := P(O; \theta_a^*)$$

$$\Psi_0 := \{\}$$

$$\alpha_0 := 0$$

568

---

#### motion update

$$P(A_t|u_{1:t}) = \sum_{A_{t-1}} P(A_t|A_{t-1}, \mathbf{u}_t) P(A_{t-1}|u_{1:t-1})$$

$$P(A_t|Y_{0:t-1}, u_{1:t}) = \sum_{A_{t-1}} P(A_t|A_{t-1}, \mathbf{u}_t) P(A_{t-1}|Y_{0:t-1}, u_{1:t-1})$$

$$\bar{\Psi}_{0:t} \leftarrow \Psi_{0:t-1} : \text{Algorithm 1 (motion update)}$$

---

#### measurement update

$$\alpha_{0:t} = \alpha_{0:t-1} + \sum_{A_t} \sum_O \left( P(\mathbf{Y}_t|A_t, O) - 1 \right) P_{\cap}(A_t, O, Y_{0:t-1}|u_{1:t})$$

$$P(Y_{0:t}|u_{1:t}) = 1 + \alpha_{0:t}$$

$$P(A_t|Y_{0:t}) = P(A_t|Y_{0:t-1}) - \left( P_{\cap}(A_t|Y_{0:t-1}) - P_{\cap}(A_t|Y_{0:t}) \right)$$

$$P(O|Y_{0:t}) = P(O|Y_{0:t-1}) - \left( P_{\cap}(O_t|Y_{0:t-1}) - P_{\cap}(O_t|Y_{0:t}) \right)$$

$$\Psi_{0:t} \leftarrow \bar{\Psi}_{0:t} : \text{Algorithm 1 (measurement update)}$$

569 **8.2 Scalabe-MLMF Algorithm**

---

**Algorithm 3:** Scalable-MLMF: Measurement Update

---

```

input :  $P(A_t^{(i)}|u_{1:t}), P(A_t^{(i)}|Y_{0:t-1}^{(i)}, u_{1:t})$   

         $P(O^{(i)}), P(O^{(i)}|Y_{0:t-1}^{(i)}, u_{1:t})$   

         $Y_t^{(i)}$   

 $i = 1, \dots, M$ 

▷ If object  $i$  has been sensed by the agent

1 if  $Y_t^{(i)} == 1$  then
2    $P(O^{(i)}|Y_{0:t}^{(i)}) \leftarrow P(O^{(i)}|Y_{0:t-1}^{(i)})$ ;           ▷ measurement update Algo. 2
570 3    $P(A_t^{(i)}|Y_{0:t}^{(i)}, u_{1:t}) \leftarrow P(A_t^{(i)}|Y_{0:t-1}^{(i)}, u_{1:t})$ 
4   forall  $j \in (1, \dots, M-1) \setminus i$  do
5      $P(A_t^{(j)}|Y_{0:t}, u_{1:t}) = P(A_t^{(i)}|Y_{0:t}, u_{1:t})$ 
6      $P(A_t^{(j)}|u_{1:t}) = P(A_t^{(i)}|u_{1:t})$ 
7      $P(O^{(j)}|Y_{0:t}^{(i)}) \leftarrow \sum_{A^{(j)}} P(A_t^{(j)}, O^{(j)}|Y_{0:t}^{(i)})$ 
8 else
9   forall  $i \in (1, \dots, M)$  do
10    measurement update Algo. 2

```

---

571 **8.3 Recursion example**

572 Derivation of the filtered joint distribution,  $P(A_t, O, Y_t|Y_{0:t}, u_{1:t})$ , for two updates. At initialisation when  
 573 no action has yet been taken the filtered joint distribution is the product of the initial marginals and first  
 574 likelihood function:

$$P(A_0, O, Y_0) = P(O)P(A_0)P(Y_0|A_0, O) \quad (29)$$

The a first action,  $u_1$  is applied, which to get the filtered joint distribution is marginalised:

$$P(A_1, O, Y_0|u_1) = P(O) \sum_{A_0} P(A_1|A_0, u_1)P(A_0)P(Y_0|A_0, O) \quad (30)$$

$$= P(O) \sum_{A_0} P(A_1, A_0, Y_0|u_1, O) \quad (31)$$

$$= P(O)P(A_1, Y_0|u_1, O) \quad (32)$$

$$= P(O)P(Y_0|A_1, O, u_1)P(A_1|u_1, \cancel{O}) \quad (33)$$

$$= P(O)P(Y_0|A_1, O, u_1)P(A_1|u_1) \quad (34)$$

575 From Equation 32 to 33 we used the Chain rule and the cancellation in Equation 33 arise from the  
 576 factorisation of the joint distribution, see Figure 2 on page 7,  $A$ 's marginal does not depend on  $O$ . After the  
 577 application of the first action, the filtered joint has the following form:

$$P(A_1, O, Y_0|u_1) = P(O)P(A_1|u_1)P(Y_0|A_1, O, u_1) \quad (35)$$

A second measurement  $Y_1$  and action  $u_2$  are integrated into the filtered joint distribution:

$$\begin{aligned} P(A_2, O, Y_{0:1}|u_{1:2}) &= P(O) \sum_{A_1} P(A_2|A_1, u_2) P(A_1|u_1) P(Y_0|A_1, O, u_1) P(Y_1|A_1, O) \\ &= P(O) \sum_{A_1} P(A_2, A_1|u_{1:2}) P(Y_{0:1}|A_1, O, u_1) \\ &= P(O) \sum_{A_1} P(A_2, A_1, Y_{0:1}|O, u_{1:2}) \\ &= P(O) P(A_2, Y_{0:1}|O, u_{1:2}) \end{aligned} \quad (36)$$

$$= P(O) P(Y_{0:1}|A_2, O, u_{1:2}) P(A_2|\cancel{O}, u_{1:2}) \quad (37)$$

578 We expand the function  $P(Y_{0:1}|A_2, O, u_{1:2})$  to give a sense of how the likelihood function's positions get  
 579 as illustrated in Figure 5 on page 11.

$$P(Y_0, Y_1|A_2, O, u_1, u_2) = P(Y_0|\cancel{Y_1}, A_2, O, u_1, u_2) P(Y_1|A_2, O, \cancel{u_1}, u_2) \quad (38)$$

$$= P(Y_0|A_2, O, u_{1:2}) P(Y_1|A_2, O, u_2) \quad (39)$$

580 The first likelihood of measurement  $Y_0$  is dependent on the last two applied actions whilst the likelihood of  
 581  $Y_1$  is dependent on the last action.

582 Repeating the above for  $Y_{2:t}$  and  $u_{3:t}$  results in:

$$P(A_t, O, Y_{0:t}|u_{1:t}) = P(O) P(A_t|u_{1:t}) \prod_{i=0}^t P(Y_i|A_t, O, u_{i+1:t}) \quad (40)$$

If  $t = 3$ ,  $(Y_{0:3}$  and  $u_{1:3})$  according to the above equation we would get:

$$\begin{aligned} P(A_3, O, Y_{0:3}|u_{1:3}) &= P(O) P(A_3|u_{1:3}) P(Y_0|A_3, O, u_{1:3}) \\ &\quad P(Y_1|A_3, O, u_{2:3}) \\ &\quad P(Y_2|A_3, O, u_{3:3}) \\ &\quad P(Y_3|A_3, O, \cancel{u_{4:3}}) \end{aligned} \quad (41)$$

583 We introduce some notation rules, first if  $(i + 1) > t$  for  $u_{(i+1):t}$  then it cancels out since the current  
 584 measurement  $Y_t$  cannot depend on a future action  $u_{(i+1)}$ .

#### 585 8.4 Derivation of the evidence

586 The evidence, also known as the marginal likelihood, is the marginalisation of all non measurement  
 587 random variables from the filtered joint distribution  $P(A_t, O, Y_{0:t}|u_{1:t})$ . We detail below how we compute  
 588 the evidence in a recursive manner whilst only considering dependent regions of the joint distribution.

589 We start with the **standard** definition of the evidence:

$$P(Y_{0:t}|u_{1:t}) = \sum_{A_t} \sum_O P(A_t, O, Y_{0:t}|u_{1:t}) \in \mathbb{R} \quad (42)$$

If both  $A_t$  and  $O$  are random variables defined over a discretised state space of  $N$  states, the above double integral will sum a total of  $N^2$  states which is the complete state space of the joint distribution  $P(A_t, O, Y_{0:t}|u_{1:t}) \propto P(A_t, O|Y_{0:t}, u_{1:t})$ , see Figure 6 on page 14 for an illustrate of such a joint distribution. As we are interested in a recursive computation of the evidence, we consider the gradient:

$$\alpha_t = \nabla_{Y_t} P(Y_{0:t}|u_{1:t}) = P(Y_{0:t}|u_{1:t}) - P(Y_{0:t-1}|u_{1:t}) \quad (43)$$

$$\alpha_t = \sum_{A_t} \sum_O P(A_t, O, Y_{0:t}|u_{1:t}) - P(A_t, O, Y_{0:t-1}|u_{1:t}) \quad (44)$$

$$= \sum_{A_t} \sum_O P(Y_t|A_t, O)P(A_t, O, Y_{0:t-1}|u_{1:t}) - P(A_t, O, Y_{0:t-1}|u_{1:t}) \quad (45)$$

$$= \sum_{A_t} \sum_O (P(Y_t|A_t, O) - 1)P(A_t, O, Y_{0:t-1}|u_{1:t}) \quad (46)$$

590 The gradient  $\alpha_t$  is the difference in mass before and after the application the likelihood function,  
 591  $P(Y_t|A_t, O)$ . The above summation, Equation 46, is over the entire joint distribution state space. We  
 592 can take advantage of the fact that the likelihood function is sparse and will only affect a small region of  
 593 the joint distribution, which we called the dependent states,  $\cap$ . The states which are not affected by the  
 594 joint distribution will result in a contribution of zero to Equation 46. We rewrite the gradient update in  
 595 terms of only the dependent regions:

$$\alpha_t = \sum_{A_t} \sum_O (P(Y_t|A_t, O) - 1)P_{\cap}(A_t, O, Y_{0:t-1}|u_{1:t}) \quad (47)$$

596 Consider the first update of the evidence at time  $t = 0$ :

$$\alpha_0 = \sum_{A_t} \sum_O (P(Y_0|A_0, O) - 1)P(A_0, O) \quad (48)$$

597 The one in Equation 49 is the original value of the normalisation denominator before any observation is  
 598 made and as the initial joint distribution  $P(A_0, O)$  is normalised the value of the denominator is one.

$$P(Y_0) = 1 + \alpha_0 \quad (49)$$

599 For the evidence  $P(Y_{0:t}|u_{1:t})$  we consider the summation of all the derivatives  $\alpha_t$  from time  $t = 0$  until  $t$ :

$$P(Y_{0:t}|u_{1:t}) = 1 + \sum_{t=0}^T \alpha_t \quad (50)$$

## 600 8.5 Derivation of the marginal

601 The marginal of a random variable is the marginalisation or integration over all other random variables,  
 602  $P(A_t, | Y_{0:t}) = \sum_O P(A_t, O | Y_{0:t})$ . Below we give a form of this integration which exploits the independent  
 603 regions in the joint distribution.

$$P(A_t, | Y_{0:t}) = \mathbf{P}(\mathbf{A}_t | \mathbf{Y}_{0:t-1}) - \left( \mathbf{P}(\mathbf{A}_t | \mathbf{Y}_{0:t-1}) - P(A_t | Y_{0:t}) \right) \quad (51)$$

604 In Equation 51 we add and subtract  $P(A_t | Y_{0:t-1})$  and we further split  $P(A_t | Y_{0:t-1})$  into independent  
 605 and dependent components:

$$\begin{aligned} P(A_t, | Y_{0:t}) &= P(A_t | Y_{0:t-1}) - \\ &\left( \underbrace{P_{\cap}(A_t | Y_{0:t-1}) + P_{\ominus}(A_t | Y_{0:t-1})}_{P(A_t | Y_{0:t-1})} - \underbrace{P_{\cap}(A_t | Y_{0:t}) + P_{\ominus}(A_t | Y_{0:t})}_{P(A_t | Y_{0:t})} \right) \end{aligned} \quad (52)$$

606 From equation 52 to 53 we used the fact that independent regions of the marginal distributions will remain  
 607 unchanged after an observation,  $P_{\ominus}(A_t | Y_{0:t-1}) = P_{\ominus}(A_t | Y_{0:t})$ , and before re-normalisation. This results  
 608 in the final recursive update:

$$P(A_t, | Y_{0:\textcolor{red}{t}}) = P(A_t | Y_{0:\textcolor{red}{t-1}}) - \left( P_{\cap}(A_t | Y_{0:t-1}) - P_{\cap}(A_t | Y_{0:t}) \right) \quad (53)$$

609 Equation 53 states that only elements of the marginals which are dependent will change by the difference  
 610 before and after a measurement update.

## 611 8.6 Figures

612 Frontiers requires figures to be submitted individually, in the same order as they are referred to in the  
 613 manuscript. Figures will then be automatically embedded at the bottom of the submitted manuscript. Kindly  
 614 ensure that each table and figure is mentioned in the text and in numerical order. Figures must be of  
 615 sufficient resolution for publication see here for examples and minimum requirements. Figures which are  
 616 not according to the guidelines will cause substantial delay during the production process. Please see here  
 617 for full Figure guidelines

## 618 8.7 Tables

619 Tables should be inserted at the end of the manuscript. Please build your table directly in LaTeX. Tables  
 620 provided as jpeg/tiff files will not be accepted. Please note that very large tables (covering several pages)  
 621 cannot be included in the final PDF for reasons of space. These tables will be published as Supplementary  
 622 Material on the online article page at the time of acceptance. The author will be notified during the  
 623 typesetting of the final article if this is the case.

## REFERENCES

624 Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for  
 625 online nonlinear/non-gaussian bayesian tracking. *Transactions on Signal Processing* 50, 174–188.

- 626 doi:10.1109/78.978374
- 627 Bake, C., Tenenbaum, J., and Saxe, R. (2011). Bayesian theory of mind: Modeling joint belief-desire  
628 attribution. *Journal of Cognitive Science* doi:doi=10.1.1.365.1528
- 629 Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Past, present, and  
630 future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Transactions*  
631 *on Robotics (TRO)* 32, 13091332
- 632 Carlone, L., Du, J., Ng, M., Bona, B., and Indri, M. (2010). An application of kullback-leibler divergence  
633 to active slam and exploration with particle filters. In *International Conference on Intelligent Robots*  
634 *and Systems (IROS)*. 287–293. doi:<http://dx.doi.org/10.1109/IROS.2010.5652164>
- 635 Carrillo, H., Reid, I., and Castellanos, J. (2012a). On the comparison of uncertainty criteria for active slam.  
636 In *International Conference on Robotics and Automation (ICRA)*. 2080–2087. doi:<http://dx.doi.org/10.1109/ICRA.2012.6224890>
- 638 Carrillo, H., Reid, I., and Castellanos, J. (2012b). On the comparison of uncertainty criteria for active slam.  
639 In *International Conference on Robotics and Automation (ICRA)*. 2080–2087. doi:<http://dx.doi.org/10.1109/ICRA.2012.6224890>
- 641 de Chambrier, G. and Billard, A. (2013). Learning search behaviour from humans. In *International*  
642 *Conference on Robotics and Biomimetics (ROBIO)*. 573–580. doi:<http://dx.doi.org/10.1109/ROBIO.2013.6739521>
- 644 Durrant-Whyte, H. and Bailey, T. (2006a). Simultaneous Localisation and Mapping (SLAM): Part I The  
645 Essential Algorithms. *Robotics & Automation Magazine* 13, 99–110. doi:10.1109/mra.2006.1638022
- 646 Durrant-Whyte, H. and Bailey, T. (2006b). Simultaneous localization and mapping: part i. *Robotics*  
647 *Automation Magazine* 13, 99–110. doi:10.1109/MRA.2006.1638022
- 648 González-Baños, H. H. and Latombe, J. (2002). Navigation strategies for exploring indoor  
649 environments. *Internationl Journal of Robotic Research* 21, 829–848. doi:<http://dx.doi.org/10.1177/0278364902021010834>
- 651 Grisetti, G., Kummerle, R., Stachniss, C., and Burgard, W. (2010). A tutorial on graph-based slam.  
652 *Intelligent Transportation Systems Magazine* 2, 31–43. doi:10.1109/MITS.2010.939925
- 653 Hesch, J., Kottas, D., Bowman, S., and Roumeliotis, S. (2014). Consistency analysis and improvement of  
654 vision-aided inertial navigation. *Transactions on Robotics (TRO)* 30, 158–176. doi:10.1109/TRO.2013.2277549
- 656 Hidalgo, F. and Brunl, T. (2015). Review of underwater slam techniques. In *2015 6th International*  
657 *Conference on Automation, Robotics and Applications (ICARA)*. 306–311. doi:10.1109/ICARA.2015.7081165
- 659 Hoffman, J., Spranger, M., Gohring, D., and Jungel, M. (2005). Making use of what you don't see: negative  
660 information in markov localization. In *International Conference on Intelligent Robots and Systems*  
661 *(IROS)*. 2947–2952. doi:<http://dx.doi.org/10.1109/IROS.2005.1545087>
- 662 Hoffmann, J., Spranger, M., Gohring, D., Jungel, M., and Burkhard, H.-D. (2006). Further studies on the  
663 use of negative information in mobile robot localization. In *International Conference on Robotics and*  
664 *Automation (ICRA)*. 62–67. doi:<http://dx.doi.org/10.1109/ROBOT.2006.1641162>
- 665 Huang, G., Mourikis, A., and Roumeliotis, S. (2013). A quadratic-complexity observability-constrained  
666 unscented kalman filter for slam. *IEEE Transactions on Robotics* 29, 1226–1243. doi:10.1109/TRO.2013.2267991
- 668 Huang, Y. and Gupta, K. (2008). Rrt-slam for motion planning with motion and map uncertainty for robot  
669 exploration. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference*  
670 *on*. 1077–1082. doi:<http://dx.doi.org/10.1109/IROS.2008.4651183>

- 671 Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path  
672 planning in high-dimensional configuration spaces. *Transactions on Robotics and Automation* 12,  
673 566–580. doi:<http://dx.doi.org/10.1109/70.508439>
- 674 Kollar, T. and Roy, N. (2008). Efficient optimization of information-theoretic exploration in slam. In  
675 *National Conference on Artificial Intelligence (AAAI)*. vol. 3, 1369–1375
- 676 Li, M. and Mourikis, A. I. (2013). High-precision, consistent ekf-based visual-inertial odometry. *The*  
677 *International Journal of Robotics Research* 32, 690–711. doi:[10.1177/0278364913481251](https://doi.org/10.1177/0278364913481251)
- 678 Lidoris, G. (2011). *State Estimation, Planning, and Behavior Selection Under Uncertainty for Autonomous*  
679 *Robotic Exploration in Dynamic Environments* (kassel university press GmbH)
- 680 Mallios, A., Rida, P., Ribas, D., and Hernández, E. (2014). Scan matching slam in underwater  
681 environments. *Autonomous Robots* 36, 181–198. doi:[10.1007/s10514-013-9345-0](https://doi.org/10.1007/s10514-013-9345-0)
- 682 Montemerlo, M. and Thrun, S. (2003). Simultaneous localization and mapping with unknown data  
683 association using fastslam. In *International Conference on Robotics and Automation (ICRA)*. vol. 2,  
684 1985–1991 vol.2. doi:[10.1109/ROBOT.2003.1241885](https://doi.org/10.1109/ROBOT.2003.1241885)
- 685 Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). Fastslam 2.0: An improved particle  
686 filtering algorithm for simultaneous localization and mapping that provably converges. In *International*  
687 *Conference on Artificial Intelligence (IJCAI)*. 1151–1156
- 688 Mourikis, A. and Roumeliotis, S. (2007). A multi-state constraint kalman filter for vision-aided inertial  
689 navigation. In *International Conference on Robotics and Automation (ICRA)*. 3565–3572
- 690 Paull, L., Saeedi, S., Seto, M., and Li, H. (2014). Auv navigation and localization: A review. *Journal of*  
691 *Oceanic Engineering* 39, 131–149
- 692 Plagemann, C., Kersting, K., Pfaff, P., and Burgard, W. (2007). Gaussian beam processes: A nonparametric  
693 bayesian measurement model for range finders. In *In Proc. of Robotics: Science and Systems (RSS)*
- 694 Ross, S., Pineau, J., Paquet, S., and Chaib-draa, B. (2008). Online planning algorithms for pomdps. *Journal*  
695 *Artificial Intelligence Research* 32, 663–704
- 696 Shachter, R. D. (1998). Bayes-ball: Rational pastime (for determining irrelevance and requisite information  
697 in belief networks and influence diagrams). In *Conference on Uncertainty in Artificial Intelligence (UAI)*.  
698 480–487
- 699 Stachniss, C., Grisetti, G., and Burgard, W. (2005). Information gain-based exploration using rao-  
700 blackwellized particle filters. In *Robotics Science and Systems (RSS)*. doi:[10.1.1.134.3647](https://doi.org/10.1.1.134.3647)
- 701 Thrun, S. (2002). Particle filters in robotics. In *Annual Conference on Uncertainty in AI (UAI)*. vol. 17.  
702 doi:[doi=10.1.1.20.567](https://doi.org/10.1.1.20.567)
- 703 Thrun, S. and Bü, A. (1996). Integrating grid-based and topological maps for mobile robot navigation. In  
704 *National Conference on Artificial Intelligence (AAAI)*. vol. 2, 944–950
- 705 Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous*  
706 *Agents)* (The MIT Press)
- 707 Thrun, S. and Leonard, J. J. (2008). Simultaneous localization and mapping. In *Springer Handbook of*  
708 *Robotics*. 871–889. doi:[http://dx.doi.org/10.1007/978-3-540-30301-5\\_38](http://dx.doi.org/10.1007/978-3-540-30301-5_38)
- 709 Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., and Durrant-Whyte, H. (2004). Simultaneous  
710 localization and mapping with sparse extended information filters. *International Journal of Robotics*  
711 *Research* 23
- 712 Torres-Gonzlez, A., d. Dios, J. R. M., and Ollero, A. (2014). Efficient robot-sensor network distributed  
713 seif range-only slam. In *International Conference on Robotics and Automation (ICRA)*. 1319–1326.  
714 doi:[10.1109/ICRA.2014.6907023](https://doi.org/10.1109/ICRA.2014.6907023)

715 Wan, E. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. 153–158.  
716 doi:10.1109/ASSPCC.2000.882463