Where wall-following works: case study of simple heuristics vs. optimal exploratory behaviour

Charles Fox

Sheffield Centre for Robotics, University of Sheffield, UK

Abstract. Behaviours in autonomous agents – animals and robots – can be categorised according to whether they follow predetermined stimulus-response rules or make decisions based on explicit models of the world. Probability theory shows that optimal utility actions can in general be made only by considering all possible future states of world models. What is the relationship between rule-following and optimal utility modelling agents? We consider a particular case of an active mapping agent using two strategies: a simple wall-following rule and full Bayesian utility maximisation via entropy-based exploration. We show that for a class of environments generated by Ising models which include parameters modelling typical robotic mazes, the rule-following strategy tends to approximate optimal action selection but requires far less computation.

1 Introduction

Autonomous behaviours can be categorised according to whether they follow predetermined rules or make decisions. Examples of rule-following robots include mobile robots programmed to follow walls, solve mazes by turning left at every junction, or operate factory equipment using machine vision and if-then rules to generate actions based on the percepts. Decision making agents do not follow such pre-programmed behaviours, but internally simulate the effects of various possible actions in order to choose the best one. For example, classic AI game-playing agents search trees of board games for the best move; Monte Carlo supply simulations search for best actions to operate business supply chains [17] and financial trades [8]; and active SLAM [2] simulates mobile robot explorations to find best movements for building maps. 'Best' is always specified by some utility function, and the agent's sole objective is to maximise this utility function. For example, an active SLAM robot could assign additive utilities to a world state based both on how much map-building is achieved (positive utilities) and how much time and power has been expended on motion (negative utilities).

During our experiments on Active SLAM [3], [6], we noticed a tendency for wall-following behaviour to emerge from the full Bayesian decision computations. This was surprising, because wall following can be obtained so easily from simple rules as well as by these large, expensive computations. This suggested that in at least some environments, full Bayesian computation may be unnecessary and well-approximated by the simple wall-following rule. In the present paper we

perform controlled experiments in a simplified class of environments to examine how general this approximation might be.

In purely theoretical decision making, the utility maximisation approach is always the 'correct' one, as it is by definition mathematically optimal. However, practical deployed systems tend to use simple rules instead. Such simple-rule based systems were popular in the Artificial Life research of the 1990s, before the advent of Bayesian machine learning, which has shifted the focus to utility maximisation. In Artificial Life, inspiration was drawn from natural systems such as insects following simple rules such as 'turn left if about to hit something' in order to form apparently complex behaviours [12] which were formalised into robotic systems such as Braitenberg vehicles [1]. Rats are known to perform much of their exploration by wall-following behaviour [10], the implementation of which has been modelled as emerging simply from maximum contact minimum impingement (MCMI) behaviour in conjunction with other biomimetic control systems [14]. State of the art robotic vacuum cleaners still use similar simple rules, such as 'follow walls' or 'if hit something, turn a random angle and move in a straight line' [16].

What is the relationship between these rules and the ideal utility maximisations? From a Bayesian perspective, we consider that an agent's objective when in some state s_0 is to maximise some utility function,

$$U(\{a_t\}_{\forall t}, s_0) = \sum_{t=1}^{\infty} \sum_{s_t} p(s_t|s_0, a_{0:t}) R(s_t), \tag{1}$$

where s are states of the world (which may include the agent's knowledge state), t is discretised time, a are actions and $R(s_t)$ are (discounted present value) rewards for being in states s_t. In general full planning, the agent must consider and optimise over all of its future action sequences to select a plan or sequence of actions $\{a_t\}_{\forall t}$,

This optimisation is generally computationally infeasible, especially for real-time systems, being exponential compute time in the number of time steps when the infinite sum is truncated. Even if a myopic approach is assumed as an extreme approximation to full planning, and we assume that all utility contribution occurs at the next greedy step,

$$U(a_t, s_t) \approx \sum_{s_{t+1}} (s_{t+1}|s_t, a_t) R(s_{t+1}), \tag{2}$$

or if a Q-learning temporal difference approach is taken (and ignoring the issue of how to compute Q),

$$U(a_t, s_t) \approx \sum_{s_{t+1}} p(s_{t+1}|s_t, a_t) Q(s_{t+1}), \tag{3}$$

with

$$Q(s_\tau) \approx R_\tau + \max_{\{a_t\}_{t=\tau:\infty}} \sum_{t=\tau+1}^{\infty} \sum_{s_t} p(s_t|s_\tau, a_{\tau:t}) R(s_t), \tag{4}$$

we can still be left with an exponentially hard problem to compute this and select an action because the state s_t can itself be a joint variable $s = (s^1, s^2, \cdots s^S)$ having a state space with size exponential in S which must be summed over (e.g. the joint states of many cells in a map). Furthermore, it is sometimes the case that for *each* s, $R(s)$ may be similarly exponentially hard to compute (as will be the case in the present study). Therefore it is imperative that a real-time[1] agent uses some fast approximation to this computation instead of sitting still doing computation. The psychological literature has identified many 'heuristics and biases' [11] and 'simple heuristics' [7] showing cases where humans behave as if following simple rules rather than computing utilities. Daw and Dayan [4] have shown similar heuristic behaviour in rats, and further shown the existence of complexity boundaries in families of tasks which cause the rats to switch from full utility computation to the use of heuristics.

As a case study of this general problem of understanding how simple rules relate to full Bayesian utility computation, we will study in this paper a simple form of greedy active mapping – a component of Active SLAM [2] – problem, and its relationship to the popular wall-following heuristic used in many exploring robots. In what environments does wall-following tend to work so well in practice?

1.1 Active mapping

Active SLAM is the Bayesian formalisation of the problem of how a robot should act to explore and localise itself to build a map of an environment and its location in it. Active SLAM is defined in terms of maximising the expected reduction in the entropy of an agent's belief about its map of the world including its own location in it, or informally, 'where to look next'. Active *mapping* is a simplified case of this problem which considers only the map of the world, and assumes the agent has a perfect location sensor at all times.

We assume the physical world w is made of discrete cell locations w^i which are either occupied or not, $w^i \in \{0, 1\}$. A map of the world at time t, m_t, is similarly made of discrete cells m_t^i, and an agent's belief about what the true map is, is represented by a distribution $P(m_t|o_{1:t})$ given all available observations up to time t, $o_{1:t}$.

While this belief may be an inseparable joint distribution over all the cells, its entropy can be (crudely but standardly [2]) approximated by assuming independent cells and summing the marginal entropies,

$$H[P(m_t|o_{1:t})] \approx \sum_i H[P(m_t^i|o_{1:t})], \tag{5}$$

with

$$H[P(m_t^i|o_{1:t})] = P(m_t^i|o_{1:t}) \log P(m_t^i|o_{1:t}) + P(\neg m_t^i|o_{1:t}) \log P(\neg m_t^i|o_{1:t}), \tag{6}$$

[1] Where 'real-time' here means having a large negative utility for slow decision making.

where $P(m_t^i|o_{1:t})$ is the occupancy probability for a cell, and $P(\neg m_t^i|o_{1:t}) = 1 - P(m_t^i|o_{1:t})$ is the probability of non-occupancy.

At each time t, greedy active mapping seeks to choose the best action $\hat{a_t}$ which reduces the entropy of the map belief as much as possible,

$$\hat{a_t}|o_{1:t} = \arg_{a_t} \max U(a_t, o_{1:t}) \tag{7}$$

$$= \arg_{a_t} \max \left(H[P(m_t|o_{1:t})] - H[P(m_{t+1}|a_t, o_{1:t+1})]\right) \tag{8}$$

$$= \arg_{a_t} \min H[P(m_{t+1}|a_t, o_{1:t+1})]. \tag{9}$$

At decision time we do not know what the future observations o_{t+1} will be, so greedy active mapping marginalises the decision over their expectation,

$$\hat{a_t} \approx \arg_{a_t} \min \langle H[P(m_{t+1}|a_t, o_{1:t})]\rangle_{P(o_{t+1}|m_t,a_t)}. \tag{10}$$

2 Methods

To isolate the phenomenon of interest in our experiments, we made as many simplifying assumptions as possible to produce a micro-world model of more general robotic exploration. We ignore the localisation part of the Active SLAM task and assume perfect (GPS-like) agent knowledge of its location at all times. Rather than model directional distal sensors such as vision or laser range finders, we assume 360 degree proximal sensors (such as touch sensors all around its body). We discretise the world into cells, and use a hexagonal grid as shown in fig. 1. (This has the modelling advantage of each cell having six well-defined, equally important neighbours rather than having to handle diagonal contacts in a square grid).

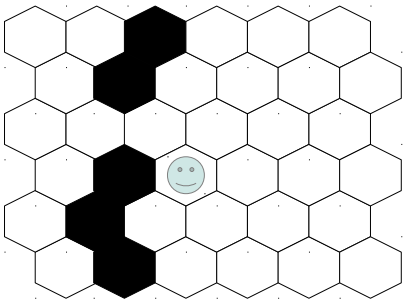

Fig. 1. A physical map. Black=occupied, white=unoccupied. The world is discretised into hexagonal cell locations. The agent position is shown by the face.

In order to test for advantageous behaviour from wall-following, relative to the Bayesian optimal solution, we wish to compute approximate average entropy changes over all possible physical worlds. To approximate this expectation we draw large numbers of samples of worlds. A simple way to model natural environments with the hex grid is to use the Ising [13] model,

$$P(\{w^i\}_{\forall i}) = \frac{1}{Z} \exp\left(\sum_i \pi(w^i) + \sum_{i,j:neigh(i,j)} \phi(w^i, w^j) \right), \tag{11}$$

with factors

$$\pi(w^i) = (\pi)^{w^i}(1-\pi)^{1-w^i} \tag{12}$$

$$\phi(w^i, w^j) = (\phi)^{w^i w^j}, \tag{13}$$

where $neigh(i,j)$ means that cells i and j are neighbours in the hexagon grid topology and Z is a normaliser. Fig. 2 shows sample worlds drawn from this model with various parameters. The model assigns probabilistic factors π to individual cells and ϕ to neighbouring pairs of cells. The effect of π is to increase the overall number of occupied cells. The effect of ϕ is to increase the contiguity of occupation, i.e. make it more probable that a cell is on if its neighbours are on. Intuitively this models the fact that physical 'stuff' in the world tends to actively clump together into large objects such as walls and furniture, but empty space does not, except as a result of the stuff clustering. By manual parameter search we found that $\pi = 0.05, \phi = 2$ gives similar environmental characteristics to our real robot mazes such as in [3]. Note that we use all positive factors rather than negative energy terms as in statistical physics[13]; the Boolean exponents in the factors act as on-off switches; and the pairwise factors equal one if it is not the case that both neighbours are occupied. To draw samples from eqn. 11 we use standard Gibbs sampling [13], with a burn-in of 100 samples initialised with independent cell draws from the π factors only.

To simplify the simulation further, we sample each decision instance in a new environment, rather than a previous partially mapped one. In a typical exploration of a *completely new* part of an environment, an agent will always have arrived from some direction, so have no (greedy) need to consider moving back in that direction. With proximal sensors, it will sense which cells around it are occupied. Fig 3 shows an example of the agent's mental world corresponding to the physical world of fig. 1, including its previous location that it has moved from. So for each decision, we sample a new world from the Ising occupancy model, place the agent at a random location, and consider the five possible moves it could take, excluding a randomly chosen simulated arrival direction. As we only consider one movement decision per world, we can use very small, 36-cell worlds as in the figures, and allow their edges to wrap around to simplify the mathematics.[2]

[2] This does however have an effect on the characteristics of sampled worlds such as island sizes, however we have chosen parameters to give realistic characteristics after the wrap around was included.

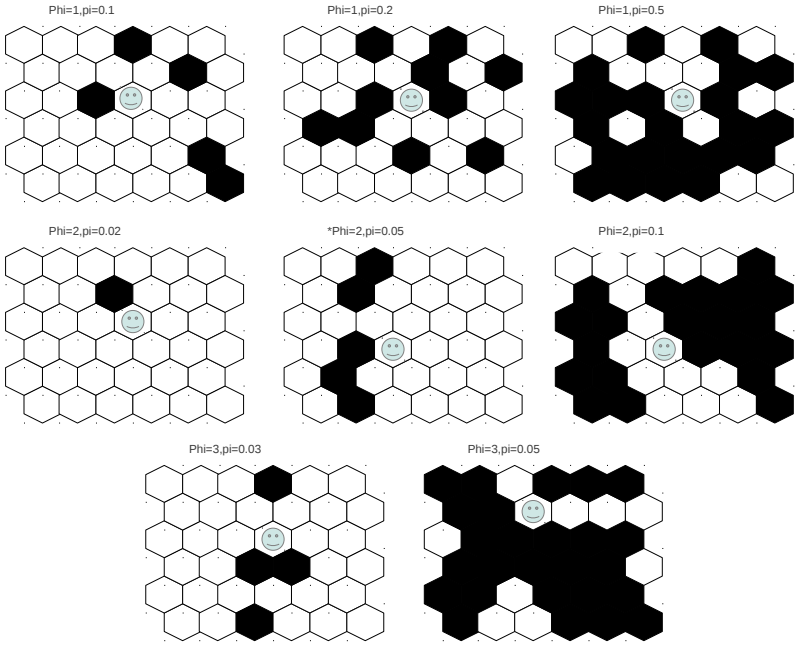

Fig. 2. Physical world samples with various parameters. Parameters around $\phi = 2, \pi = 0.05$ tend to generate worlds most like typical man-made environments, being quite sparse but locally correlated so as to produce structures resembling walls.

As we are investigating only the question of whether wall following is useful, the decisions that interest us are only those that occur when the agent begins next to at least one wall cell. If our sample world and random robot location do not meet this criterion then we discard the simulation and draw a new sample world.

At each movement decision, we consider moving to each neighbouring unoccupied cell. We compute (our best approximation to) the Bayesian expected entropy reduction given that move. We also test whether or not the move is a wall-following move (defined as a move in which the pre-move and post-move states share a neighbouring occupied cell). We store records of expected entropy reductions for every wall-following and non-wall-following move considered so that we may compare their averages to determine whether wall-following is significantly more useful than non-wall-following.

For each possible action, the ideal correct sensor values were obtained by consulting the physical world model. This is a further approximation to the expectation in eqn. 10, ideally we would use further samples from the agent's mental model here to integrate over possible physical worlds, but this approximation again speeds up the computation.

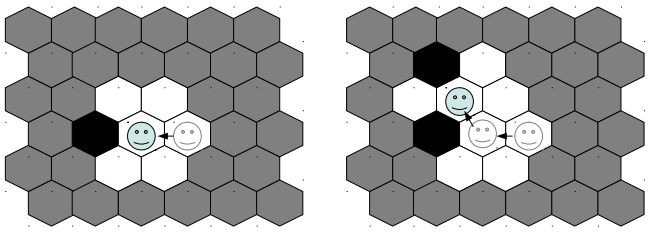

Fig. 3. Mental maps before and after a movement. Grey=uncertain. Outline robots show previously visited locations.

2.1 Approximation of entropies

For both the pre-move and post-move states we must compute eqn. 5. We make use of the standard independence assumption in the entropy sum, but we do want to retain the joint dependencies,

$$P(m_t|o_{1:t}) \neq \prod_i P(m_t^i|o_{1:t}). \tag{14}$$

This is because we are interested in how the correlations in the world induced by occupancy clumping in natural environments might affect the utility of the wall-following heuristics.

In order to compute eqn. 5 under these assumptions, we must therefore draw samples from the *joint* distribution of map cells, i.e. from the space of possible mental worlds given the observations. This can be done using a 'wake' [9] version of the Gibbs sampler used to generate physical worlds from eqn. 11, i.e. by clamping the observed cells to their observed values and Gibbs sampling from all remaining cells.

The pre-move and post-move mental map joint distributions were each approximated by drawing 100 wake samples from the unobserved cells, using 100 random Gibbs updates between each sample to approximate burn-in. The marginal entropy terms in eqn. 6 were then estimated from Good-Turing ('add-one') adjusted occupancy frequencies [13]. (It has been found empirically that correlated, short approximate burn-ins often work surprisingly well, e.g.. the use of single updates in contrastive divergence learning [9]. A full analysis of this assumption is beyond the scope of the present paper.)

3 Results

Setting $\phi = 2, \pi = 0.05$ which gives typical robot-arena-like environments, we ran the simulation for 100,000 world samples with about 3.5 possible actions per sample. The total computation time using these approximations was around 10 hours on a 3GHz Pentium Duo PC.

We obtained a sample mean expected entropy per cell reduction of 0.001933 for wall-following moves, and -0.001687 for non-wall-following moves.[3] The sample standard deviations were 0.0127 and 0.0122, and the numbers of observations were 49,241 and 64,039. To avoid any possible bias from simulating multiple actions from the same world sample, we will conservatively use N=10,000 rather than these counts. Assuming both populations standard deviations to be 0.015, a classical t-test gives a significant difference in favour of the wall-following moves. (t=46.7, N=10,000, df=10,000, $p \leq 0.001$).

Having demonstrated that this number of world samples lead to a significant result for our manual-parametrised world, we then ran a further 70 repeats of the experiments (700 hours computation time) for a range of (ϕ, π) parameters, to test how general the wall-following rule's usefulness is. The differences in expected entropy reduction means are plotted in fig. 4.

4 Discussion

Where do animals [10] and practical robots [6] benefit from using simple wall-following behaviour in place of full Bayesian utility maximation? The results show that for proximal sensing agents exploring previously unvisited regions in

[3] The negative entropy may seem strange but these are *per cell* statistics. The number of uncertain cells changes after a move when new cells are observed which may explain the apparent increase in uncertainty *per cell* rather than total map uncertainty. It is only the difference between the two means that is of interest.

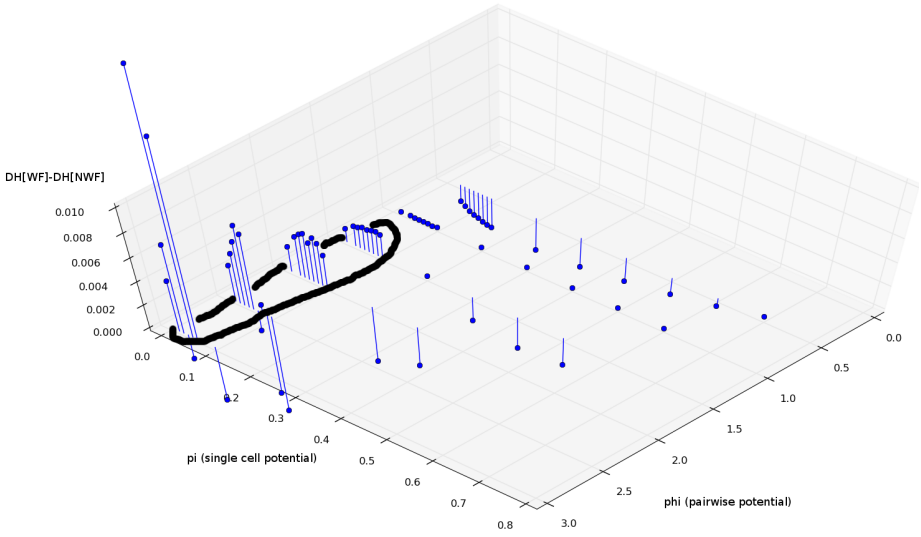

Fig. 4. Results. x and y axes are Ising parameter values. z axis is the sample mean difference between entropy change (DH) given wall following (WF) moves, and entropy change given non-wall-following (NWF) moves. The lasso shows the region of parameter space where wall following gives a significant advantage over non-wall-following.

a realistic environment class having sparse occupancy but strong local correlation, under simplifying assumptions, that wall-following actions do have significantly better outcomes than non-wall-following actions. This suggests that wall-following provides a fast approximation to optimal entropy exploratory behaviour in this type of environment.

Fig. 4 suggests that there is a 'sweet spot' in the parameter space where wall following works better than non-wall-following. This has been indicated by the hand-drawn lasso. It is interesting to see that the centre of this lasso corresponds roughly to the parameters $(\pi = 0.05, \phi = 2)$ that were originally chosen by hand to give real-environment-like characteristics. Fig. 2 shows samples of physical worlds with various parameters, corresponding to parameter coordinates inside and outside the lasso. The $(\pi = 0.05, \phi = 2)$ sample in the centre perhaps looks more like a typical room environment than any of the others in terms of density and connectivity. (Note that the hex grid wraps around, so correlations around the edges count like any others.)

Our simulations used the assumption that all exploration was in some 'new' area, i.e. that only one previously-visited cell is nearby, that it is a neighbour, and that we have just moved to the present location from it. This assumption is valid for agents just beginning to explore a new region of a map, but not for agents building up a detailed map over a longer time and revisiting areas. A pure wall-follower starting at the wall would never discover an object in the centre of a room for example. Revisiting also often occurs as an optimal action

in full Active SLAM systems, where the agent must re-localise from time to time as well as explore. Our simulations do not include tasks where particular locations must be visited (e.g. vacuuming a room, finding a power charger), rather their utility is only in mapping itself. Further work would be needed to test the utility of wall following in these more complex situations. An intriguing idea would be to test for similar effectiveness of other simple rules in these tasks, against full utilities, for example attempting to compare the Roomba's default combination of wall-following and random-straight-line wall-bouncing against an Active-SLAM approach to room coverage. The greedy active mapping utility used in our Bayesian utility maximisation is itself an approximation to non-greedy full planning, the latter would be expected to behave differently from the former in long-term exploration, but is so computationally hard as to be infeasible to simulate to significance even in our micro-world.

The hex-grid worlds used have the nice features that all actions are cleanly classified into wall-following or non-wall-following, and there are typically similar numbers of each that get sampled by our simulation. (49,241 and 64,039 in the main example above.) It is possible that other simple strategies could exist that are more selective that both wall-following and non-wall-following, and provide an even better approximation to optimal behaviour, though such strategies are beyond the scope of the present study.

In using the same (π, ϕ) parameters in both world generation and mental map sampling, we have implicitly assumed that the agent knows the Ising statistics of the class of environments it will be placed in. Such parameters could be learned from experience, but it would be interesting to investigate the effect of mismatches between the physical and mental parameters.

We have made use of large compute power sampling approximations in our results, and it is possible that for some simple worlds such as the Ising hex worlds, there exist analytic solutions for the expected entropy reduction. Further theoretical work could analyse the models used here to search for closed form solutions which may provide further insights into why the rule works. The 700 hour compute time used was necessary to obtain significance in the results (a 6×6 hex world has 2^{36} physical states, and far more mental states, to sample from), and gives some indication of how computationally hard optimal exploration can be, even for very simple worlds (cf. [15], which uses similarly tiny micro-worlds and approximations to compute entropy based explorations.)

While we have demonstrated a particular simple rule approximating optimal utility-based action selection, this work is also intended as an example of the more general case [5]. Can we analyse other well-known heuristics in this way? It is becoming clear that even with large supercomputers many decisions are intractable, and therefore a relevant research agenda for psychology and robotics is to seek out and try to understand useful rules that approximate such decisions. In what classes of environment are they valid and when do they break down? We could try to invent new rules from intuition or data-mining, or look at rules traditionally used by humans to find test candidates.

Bibliography

[1] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* Bradford Bks. MIT Press, 1986.

[2] L. Carlone, Jingjing Du, M.K. Ng, B. Bona, and M. Indri. An application of Kullback-Leibler divergence to active SLAM and exploration with particle filters. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 287 –293, oct. 2010.

[3] C.Fox, M.Evans, M.J.Pearson, and T.J.Prescott. Towards hierarchical blackboard mapping on a whiskered robot. *Robotics and Autonomous Systems*, 60:1356–1366, 2012.

[4] N.D. Daw, Y. Niv, and P. Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatum systems for behavioral control. *Nature Neuroscience*, 8:1704–1711, 2005.

[5] C. Fox. Formalising robotic ethical reasoning as decision heuristics. In *Proc. First UK Workshop on Robot Ethics, Sheffield*, 2013.

[6] C. W. Fox, M. H. Evans, M. J. Pearson, and T. J. Prescott. Tactile SLAM with a biomimetic whiskered robot. In *ICRA*, 2012.

[7] G. Gigerenzer, P.M. Todd, and ABC Group. *Simple Heuristics That Make Us Smart.* Oxford University Press, 2000.

[8] P. Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer, 2003.

[9] G. E. Hinton, S. Osinderoi, and Y-W Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[10] G. Hosterrer and G. Thomas. Evaluation of enhanced thigmotaxis as a condition of impaired maze learning by rats with hippocampal lesions. *Journal of Comparative and Physiological Psychology*, 63(1):105–110, 1967.

[11] D. Kahneman. *Thinking Fast and Slow.* Macmillan, 2011.

[12] C. G. Langton. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22(1–3):120–149, 1986.

[13] D.J.C. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2003.

[14] B. Mitchinson, M. Pearson, A. Pipe, and T. Prescott. The emergence of action sequences from spatial attention: Insight from rodent-like robots. In *Proc. Biomimetic and Biohybrid Systems*, 2012.

[15] Z. Saigol. *Automated Planning for Hydrothermal Vent Prospeting using AUVS.* PhD thesis, University of Birmingham, UK, 2010.

[16] P. Strauss. Roombas make dazzling time-lapse light paintings. *http://technabob.com/blog/2009/09/29/roomba-time-lapse-art/*, URL.

[17] M. Taylor and C. Fox. Inventory management with dynamic bayesian network software systems. In *Proc. Int. Conf. Business Information Systems, Springer LNBIP*, 2011.