Towards hierarchical blackboard mapping on a whiskered robot

C.W. Fox[a,*], M. H. Evans[a,*], M. J. Pearson[b], T.J. Prescott[a]

[a]*Sheffield Centre for Robotics, University of Sheffield, Western Bank, Sheffield, S10 2TF, UK*
[b]*Bristol Robotics Laboratory, Bristol, UK*

Abstract

The paradigm case for robotic mapping assumes large quantities of sensory information which allow the use of relatively weak priors. In contrast, the present study considers the mapping problem for a mobile robot, CrunchBot, where only sparse, local tactile information from whisker sensors is available. To compensate for such weak likelihood information, we make use of low-level signal processing and strong hierarchical object priors. Hierarchical models were popular in classical blackboard systems but are here applied in a Bayesian setting as a mapping algorithm. The hierarchical models require reports of whisker distance to contact and of surface orientation at contact, and we demonstrate that this information can be retrieved by classifiers from strain data collected by CrunchBot's physical whiskers. We then provide a demonstration in simulation of how this information can be used to build maps (but not yet full SLAM) in an zero-odometry-noise environment containing walls and table-like hierarchical objects.

1. Introduction

Touch-based mapping has two principal applications. Firstly, as a sole sensory system in environments where other types of sensors fail, such as smoky or dusty search-and-rescue sites, especially where covert (no signal emission) operation is required. Secondly, as a complement to other sensors such as vision, with which it can be fused or used as a 'last resort' during adverse conditions as in the sole sensor case.

However, the paradigm case for robotic mapping, as in Simultaneous Localisation and Mapping (SLAM) problems [53], instead considers a mobile robot with noisy odometry and vision or laser scanners. Vision and laser scanners provide large amounts of sensory information, and have effectively unlimited range in indoor environments. Such large quantities of input information allow the use of relatively weak priors, such as independent grid cell occupancy or flat priors over the belief of small feature sets [53].

This study considers the touch-based mapping problem in which only sparse, local sensory information is available. Proof that navigation from such sensors is possible is readily found in biology: rats navigate through dark underground tunnels using their whiskers [7, 2], having ranges of only a few centimetres. In robotics, whisker sensors are relatively cheap in both material and computational processing terms, and their use has previously been considered in constrained tasks [47, 46, 31, 30, 15]. The previous robotic attempts at mapping from sparse local sensors have either used the extremely strong generic prior that the whole world is made entirely of north-south and east-west straight edges [57] or have used relatively long range but sparse ray sensors integrated over multiscans [3].

We will demonstrate touch-based mapping using a mobile robot, CrunchBot, having six whisker touch sensors only. First, it is shown that CrunchBot's whiskers are able to recover approximate position and orientation reports about contacts with surfaces. Then it is shown how these reports can be fused with strong priors to recognise hierarchical objects such as tables and chairs, as a step in building a map of the environment.

Fig. 1 gives an overview of the general framework for perception and navigation with whiskers within which this study operates. When biomimetically inspired by rodents, whisker sensors have strain sensors at their base only. When a rat investigates an object it palpates the surface in a back and forth oscillatory sweeping behaviour known as 'whisking' [55],[9]. It is thought that whisking is important for gathering the most reliable signals from whisker contacts [38]. Straight whiskers can make two distinct types of contact with an object,

*These authors contributed equally and are joint first authors
Email addresses: charles.fox@sheffield.ac.uk (C.W. Fox), mat.evans.sheffield.ac.uk (M. H. Evans), martin.pearson@brl.ac.uk (M. J. Pearson), t.j.prescott@sheffield.ac.uk (T.J. Prescott)

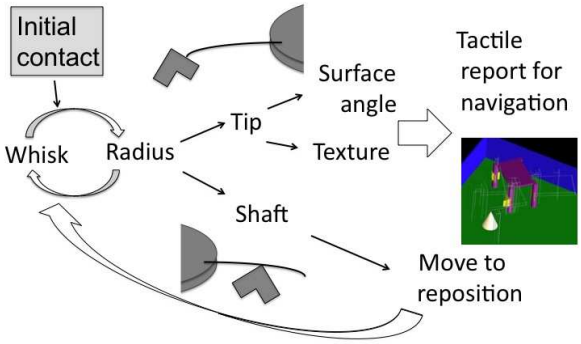

Figure 1: A new framework for extracting contact parameters. After initial contact a whisk behaviour allows the discrimination of object location. If contact is made along the whisker shaft the agent must move to reposition the whisker for subsequent contacts. If contact is made at the whisker tip a robust discrimination of surface properties can be made. Reports of surface properties can be used by other systems, such as for navigation or to construct complex object models as in the present study.

contacting it either at their tip or their shaft. Tip contacts are generally the most useful, because they provide a standardised, constrained setting (i.e. with the contact point at a known location at precisely the end of the whisker) from which surface properties such as orientation and texture can be identified [31],[15]. In contrast, shaft contacts are less informative. For example, an unknown distance to an object along the shaft can confuse attempts to classify surface orientation and texture [18]. Shaft contacts are rare in practice in both rodents and mobile robots, occurring only when small objects enter the field of multi-whisker arrays between the whisker tip points. In the scheme used here, a feature based radial distance estimator [13] is first used to make a decision of whether the contact is at the tip or the shaft. If it is a shaft contact, then the robot should use the radial distance information to move to another location that is likely to yield a more useful tip contact. Following a tip contact, we can read surface orientation and texture information (and possibly speed of object when there are moving objects in the world) and pass them as an observation to a navigation or mapping system.

This study provide an implementation of the distance and angle stages of this framework on CrunchBot (Fig.2(a)). Individual components of such a system have previously been investigated in isolation, including whiskered texture recognition [15],[27],[11], [18],[34], surface shape recognition [31],[25],[22],[13], and object recognition [20]. These components have previously been tested under ideal laboratory conditions or in individual mobile settings [44]; here we present steps integrating them into a single platform for hierarchical object recognition, along with results and observations on their performance 'in the wild' in a common arena environment.

To compensate for the sparseness of the sensory information available from these distance-orientation reports, we fuse them with strong hierarchical priors about objects in the world. Hierarchical object recognition models were popular in classical AI in the guise of 'blackboard systems' [10, 37, 5] but have recently been recast in terms of dynamically constructed Bayesian networks [16, 36, 33, 52]. Here we provide an application of Bayesian blackboards to robotic mapping. We do not consider the full SLAM problem here, but instead work in a simulation of CrunchBot having zero odometry noise to avoid the localisation problem and focus on mapping only. Related object-based mapping models have recently appeared [54, 24, 49, 43] using laser sensors to recognise and learn complex but non-hierarchical spatial models. However as data available through whiskers to CrunchBot is much sparser than that from laser scanners, the required level of sensor detail is unavailable, therefore we compensate with the new mapping technique of fusing contact reports into *hierarchical* models. For example, on recognising a single table leg, we may infer the probable presence the rest of the table, including other leg objects, and edges and corners making up these legs, without ever sensing them directly. To construct hierarchical objects, we use hypothesis priming and pruning heuristics as in classical blackboard systems. However, following [16], we treat such heuristics as approximations to inference in a dynamically-constructed, Monte Carlo Markov Chain (MCMC) sampling Bayesian network, whose observations are the distance-orientation reports from the whiskers.

2. Methods

2.1. Whiskers.

CrunchBot's six whiskers measure 160mm in length, 1.45mm diameter at the base tapering linearly to 0.3mm at the tip. They are built from nanocure25 using an Envisiontec rapid prototyping machine. A magnet is bonded to the base of the whisker and held in place by a plug of polyurethane approximately 0.75 mm above a Melexis 90333 tri-axis Hall effect sensor IC [35]. This sensor generates two outputs representing the direction of the magnetic field (in two axes) with respect to its calibrated resting angle. These two 16-bit values are sampled by

Figure 2: CrunchBot, a whiskered mobile robot platform.

a local dsPIC33f802 micro-controller which, in turn, is collected using an FPGA configured as a bridge to a USB 2.0 interface. Up to 28 whiskers can be connected to this FPGA bridge at one time. Using the vendor provided software driver and API (Cesys GmbH), a user can request the data from all whiskers at minimum intervals of 500μs (a maximum sample rate of 2kHz).

2.2. Robot platform.

CrunchBot is based on the iRobot Create base (`www.irobot.com`) platform, with the whiskers mounted in the cargo bay, being positioned on an adjustable metal bar and rapid prototyped ball joint mountings. These mountings allow adjustment of the whiskers. For data collection experiments in the present study, only four whiskers are used, configured in the horizontal plane to detect objects in an arena (the other two whiskers scrape along the floor and are used in other experiments, such as for texture discrimination in our previous study, [21]). We have also extended the cargo bay mounting to accommodate a netbook PC, which is used for local control of the robot. The netbook runs Ubuntu 10.10 on a single-core Intel Atom processor. A circular buffer in shared memory is used to make data from the Cesys driver available to other processes. The netbook hosts a Player server (`playerstage.sourceforge.net`) which provides high-level, networked API interfacing to the Create's serial port commands. Processes such as texture and shape recognition and basic motor control run on the netbook, reading the raw data from the fast circular shared memory buffer, and writing their results every 0.1s to a Python Pyro server (`pyro.sourceforge.net`) on the remote desktop which runs hierarchical object recognition and mapping.

2.3. Robot movement

Previous work has shown that accurate object localisation with a whisker requires some measure of contact speed [13], or of the applied forces and bending moments at the base of the whisker [25],[31], values that are not always available in the mobile case as agent movement will affect these contact properties. To address these points a 'body whisk' behaviour was included in the robot program. As the whiskers were not actuated the whole robot must rotate in a systematic way. Upon initial contact with an object the robot first reverses away a short distance before rotating at $\pi/24$ radians per second towards the object, then rotating at $\pi/24$ radians per second away from the object. This allows this whiskers to move over the surface of the contact object, collecting data about its location and orientation. After the whisk the robot reverses again to clear the object, then rotates in a random direction and moves forward again.

2.4. Radial distance reporting

To determine whether an object has made contact with any of CrunchBot's whiskers at the tip or the shaft, and to discriminate between contacts with the surfaces or corners of objects, object localisation was implemented. Previous work [13] has shown that peak deflection magnitude could be used as a feature for radial distance discrimination at a given speed. Whisker data was recorded during the 'body whisk' contact, and the maximum whisker deflection was measured. Deflection magnitude was taken as the Hall effect sensor output voltage at peak deflection, which is proportional to the bending moment. This feature f_1 can be defined as,

$$f_1 = \max_t \theta(t), \tag{1}$$

where $\theta(t)$ is the time-dependent deflection magnitude measured by the Hall effect sensor.

During the training phase a dataset was collected for each whisker, consisting of 5 contacts at each point along the whisker at 10mm intervals over a 50mm range from the tip of the whisker. Though the whisker is 160mm long, only 140mm is external to the 'follicle'. A model was then generated of the relationship between the deflection magnitude and the corresponding radial distance to contact by fitting a linear equation to the training data in MATLAB. To find an estimate of radial distance r,

$$r = a_1 f_1 + a_0, \tag{2}$$

was fitted to the data with a linear-in-the-parameters regression on the line, giving a least-squares fit for (a_0, a_1) for each whisker.

2.5. Surface orientation reporting

A complementary data-driven approach to feature extraction methods is to store instances of time series as a set of templates for comparison to novel data. It has been shown that simple k-means style templates on strain time series from *individual* whiskers can be used for discriminating contact *distance* classes in physical simulation [22], and stationary robot hardware [12]. In the present study we have access to four whiskers together, so we can train templates corresponding to surface orientation classes from the 8-dimensional time series from the whole *multi-whisker* set (four whiskers, each with vertical and horizontal strain channels). Orientation reports could be used to inform complex object models as in sec. 2.6. The rationale for this particular approach is that a template method can utilise bulk data from all whiskers to find surface orientations, without any of the geometric assumptions required when splining individual radial distance reports [31].

Offline training data was collected by programming the robot to drive into a wall at fifteen different angles (20°:160° in 10° intervals) four times. Data was aligned to initial contacts, low pass filtered (17Hz) to remove oscillations caused by robot body movement, recorded for 2s, and smoothed with a five-point moving average. Templates were generated by averaging across three of the four sets for each angle. Templates for each angle comprised data of all eight channels from the four whiskers to allow multi-whisker information to inform classification.

During testing the fourth data set was used compared to the averaged template using a sum of squared error measure. The average squared error, e for each template, T_i is computed over the N logged data points,

$$e(T_i) = \frac{1}{N} \sum_{t=1}^{n} (I(t) - T_i(t))^2. \qquad (3)$$

The template with the lowest sum of squared errors was determined the winner, and the orientation of this template recorded. This process was repeated four times, each time using a different data set as the test set, and a template constructed from the remaining three sets. In this manner it is possible to generate a robust estimate of mean classifier performance, while preserving individual trial differences for inspection in the results.

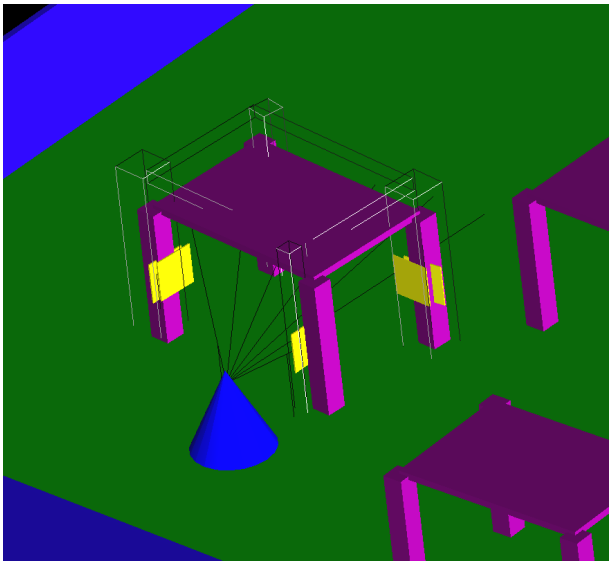

Figure 3: Simulation screen-shot at low annealing temperature. A single table hypothesis remains, aligned correctly with the physical table.

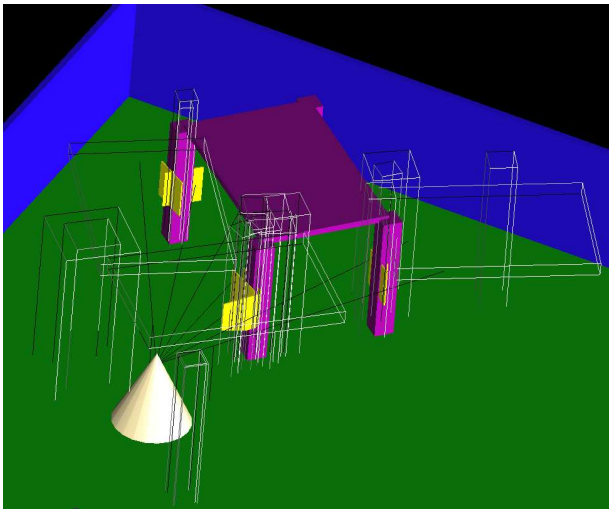

Figure 4: Simulation screen-shot at high annealing temperature. Many hypothesised (wire-frame) tables and legs are on the blackboard, primed by the shapelets (yellow rectangles) contacted by the robot (cone)'s whisker sensors, in an arena containing a physical table (pink).

2.6. Hierarchical Object Models

CrunchBot's task is to build a map of an arena populated by four-legged table-like objects as in figs. 4 and 3. Such objects could include chairs and desks in a home or office environment for example. In our object recognition experiment, CrunchBot moves along a *predetermined* trajectory of location-angle poses, (x^t, y^t, θ^t) around the arena (though we discuss possibilities for autonomous exploration in sec. 2.9), over discrete time steps t. We assume that at each discrete time step t, CrunchBot's whiskers, $w \in 1:6$, each report egocentric estimates of the radial distance r to, and surface orientation ϕ, using the methods of previous sections, and of contact texture[1] τ of, any contacts made,

$$\hat{r}_w^t = r_w^t + \varepsilon_r, \tag{4}$$

$$\hat{\phi}_w^t = \phi_w^t + \varepsilon_\phi, \tag{5}$$

$$\hat{\tau}_w^t = \tau_w^t + \varepsilon_\tau, \tag{6}$$

where ε are i.i.d. Gaussian noises having zero mean and standard deviations $\sigma_r^w, \sigma_\phi^w, \sigma_\tau^w$ respectively. Assuming perfect odometric localisation in the present study, these estimates may be converted into allocentric Cartesian coordinates to give tuples $S(x_S, y_S, \phi_S, \tau_S)$ which we call *shapelets* and which will be treated as observations in graphical models.

Tables, T, are parametrised by tuples, $T(x_T, y_T, \theta_T, w_T^x, w_T^y, w_T^L, \tau_T)$, where x, y, θ is the pose, w_T^x and w_T^y are width and breadth, w^L is the width of the (square) legs, and $\tau \in (0, 1)$ is a texture parameter describing roughness or smoothness of the material. A generative model of tables is used. CrunchBot assumes a flat prior probability *density* generating tables in the world,

$$p(T(x_T, y_T, \theta_T, w_T^x, w_T^y, w_T^L, \tau_T)|\emptyset) = c_T, \tag{7}$$

where c_T is a (non-normalising) constant.

If a table T exists, its presence causes (in the sense of [41]) the presence of four leg objects,

$$L(x_L, y_L, \theta_L, w_L, \tau_L, T), \tag{8}$$

where w_L is the width of the square table leg; x_L, y_L, θ_L are its location and rotation, and τ_L is its texture, with probability density

$$p(L(x_L, y_L, \theta_L, w_L, \tau_L, T)|T(x_T, y_T, \theta_T, w_T^L, \tau_T))$$

[1]Texture reports are not yet implemented on the physical CrunchBot, but can already be handled by our hierarchical framework in simulation so are included for completeness. In the present simulations we assume all tables and reports have the same texture, $\tau = 1$.

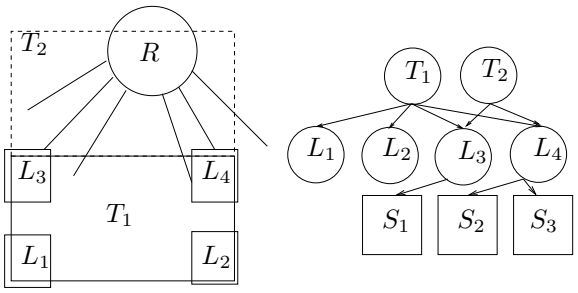

Figure 5: Hierarchical object recognition. *Left:* Robot R (circle) with six whiskers (lines) makes tactile contact with legs L_j (squares) of a hypothesised table T_1 (rectangle). The two contact points ('shapelets') on the right are sufficient to infer the location of the corner of leg L_4. Coupled with prior knowledge about the shape and size of tables, and the third shapelet, this can be used to infer that there is a table either in the ground truth location or in a second configuration T_2 (dashed rectangle). *Right:* Bayesian network constructed to represent the same scenario. Square nodes are the shapelet observations.

$$= \alpha_L \exp\left(-\Delta_{TL}\right), \tag{9}$$

where α is a (non-normalising) constant, and the distance measure is

$$\Delta_{TL} = \min_i \left(\frac{(x_T^i - x_L)^2 + (y_T^i - y_L)^2}{\sigma_r^2} \right) + \left(\frac{\theta_T - \theta_L}{\sigma_\theta} \right)^2 + \left(\frac{w_T^L - w_L}{\sigma_w} \right)^2 + \left(\frac{\tau_T - \tau_L}{\sigma_\tau} \right)^2, \tag{10}$$

where $0 \le i \le 3$, and (x_T^i, y_T^i) are the coordinates of the table's four corners, and σ_w, σ_τ are parameters specifing standard deviations of the leg's w_L, τ_L values conditioned on the table's corresponding w_T^L, τ_T values. The inclusion of T in the parametrisation of L (eqn. 8) means that L is the hypothesis that the leg was caused *only* by table T rather than any other table or cause.

Shapelets are assumed to be generated by nearby legs,

$$p(S(x_S, y_S, \theta_S, \tau_S)|L(x_L, y_L, \theta_L, w_L, \tau_L, T)) = \alpha_S \exp\left(-\Delta_{LS}\right) \tag{11}$$

where

$$\Delta_{LS} = \left(\frac{r}{\sigma_r^S} \right)^2 + \left(\frac{f(\theta_L) - \theta_S}{\sigma_\theta^S} \right)^2 + \left(\frac{\tau_L - \tau_S}{\sigma_\tau^S} \right)^2, \tag{12}$$

and r is the shortest radial distance from the perimeter of the leg to (x_S, y_S), computed by basic geometry, $f(\theta_L) = \theta_L + m\pi/2$ picks the angle of the corresponding side m of the leg at this shortest-distance contact point, and $\sigma_r^S, \sigma_\theta^S, \sigma_\tau^S$ model sensor noise.

We also provide small *null priors* to allow legs and shapelets to exist in the absence of any generative parents. (These are required later, during construction on

the blackboard, so that these objects can survive before their parents are constructed),

$$p(L(x_L, y_L, \theta_L, w_L, \tau_L, \emptyset)|\emptyset) = c_L, \quad (13)$$

$$p(S(x_S, y_S, \theta_S, \tau_S)|\emptyset) = c_S, \quad (14)$$

with constants such that the marginalised densities,

$$p(S(x_L, y_L, \theta_L)) < p(L(x_L, y_L, \theta_L)) < p(T(x_L, y_L, \theta_L)), \quad (15)$$

i.e. larger objects are more probable to exist without high-level causes than smaller objects are.

Unlike the parametrisation of L on T in eqn. 8, we assume that shapelets may be caused by *mixtures* of multiple leg hypotheses and by the null prior (eqn. 14). For example if there are two legs very close together then the density for observing shapelets in the area increases. We assume that multiple causal sources combine using noisy-OR semantics,

$$P(x|pa(x)) = 1 - \prod_{x_j \in pa(x)} (1 - P(x|x_j)). \quad (16)$$

where $pa(x)$ denotes the set of causing ('parents' in Bayesian network terminology) objects of an object ('node' in Bayesian network terminology) x such as a leg, table or shapelet. As we use probability *density* functions we require the continuous version of noisy-OR, proved below,

Theorem

$$p(x|pa(x)) = \sum_{x_j \in pa(x)} p(x|x_j). \quad (17)$$

Proof

$$P(x|pa(x)) = 1 - \prod_j (1 - P_j). \quad (18)$$

with $P_j = P(x|pa(x)_j)$, the probability of x given the jth possible cause. Consider the probability of a small range of hypotheses,

$$\delta^3 p(x|pa(x)) = 1 - \prod_j (1 - \delta^3 p_j), \quad (19)$$

where p are probability densities and P are corresponding probabilities. Expansion terms with powers of δ that are > 3 vanish, so

$$\delta^3 p(x|pa(x)) = \delta^3 \sum p_j. \quad (20)$$

The δ^3 terms cancel to yield

$$p(x|pa(x)) = \sum p_j, \quad (21)$$

as required.

We allow legs to be caused by a mixture of their *single* specified parent (i.e. the T parameter in eqn. 9) and null prior (eqn. 13), using a similar combination rule. Tables are caused by the null prior only (eqn. 7).

Taken together, the equations in this section may be viewed as a Bayesian network [40] for any given collection of tables, legs and shapelets as shown in fig. 5. However, in addition to the previous causal probabilities, we need to model the following constraints: (a) tables always have four legs; (b) each table leg is at a different corner of the table (we should not see two legs attached to the same corner); (c) two objects of the same type (table or leg) cannot overlap in physical space. Standard Bayesian networks cannot model such relations, as they are limited to joint distributions of the form

$$P(\{x_i\}_i) = \prod_i P(x_i|pa(x_i)). \quad (22)$$

To model these additional constrains, we extend the Bayesian network using undirected penalty factors, to form the factor graph [32],

$$P(\{x_i\}_i) = \frac{1}{Z} \left(\prod_i P(x_i|pa(x_i)) \right) \times \left(\prod_{ij} \phi_c(x_i, x_j) \phi_b(x_i, x_j) \right) \left(\prod_i \phi_a(x_i) \right), \quad (23)$$

where Z is a normalising constant, and ϕ_a, ϕ_b, ϕ_c are unnormalised penalty factors corresponding to the three new constraints. Using superscripts for exponentiation, these are

$$\phi_a(x_i) = \epsilon_a^m, \quad (24)$$

$$\phi_b(x_i, x_j) = \epsilon_b^v, \quad (25)$$

$$\phi_c(x_i, x_j) = \epsilon_c^r, \quad (26)$$

where m is the number of missing legs iff x_i is a table, and $m = 0$ otherwise; v is a Boolean (0,1) value, true if hypotheses x_i and x_j are of the same type and overlap in physical space; and r is a Boolean, true if hypotheses x_i and x_j are legs and share the same parent (modelling this parent-sharing is why we parametrise L by T in eqn. 8).

Algorithm 1 Blackboard-inspired approximate Metropolis-hasting proposals generation.

for each time step t **do**
  update shapelet queue S by reading sensors
  for each annealing inverse temperature β **do**
    for each shapelet $S_i \in S$ **do**
      propose and test parent H_i from $Q(pa(S_i))$
      if accepted, add H_i to hypothesis set B
    end for
    for each hypothesis $H_i \in B$ **do**
      $r \leftarrow rand(0, 1)$
      if $r < r_1$ **then**
        propose death of H_i
        if accepted, remove H_i from B
      else
        if $r < r_2$ **then**
          propose parent change for H_i
          if accepted, replace H_i's parent parameter
        else
          if $r < r_3$ **then**
            propose child H_j from $Q(ch(H_i)|H_i)$
            if accepted, add H_j to hypothesis set B
          else
            propose parent H_j from $Q(pa(H_i)|H_i)$
            if accepted, add H_j to hypothesis set B
          end if
        end if
      end if
    end for
    prune all hypotheses not linked to any shapelet directly or via a common ancestor.
  end for
end for

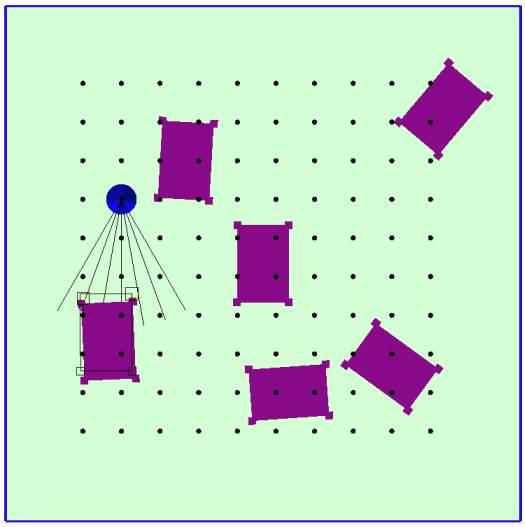

Figure 6: Overhead view showing ground truth table configuration, and locations (black dots) of the discrete poses occupied by the robot. There are four angle poses at each location, facing in compass directions.

2.7. Hierarchical Object Inference

For a given set of shapelet observations and a set of candidate hierarchical legs and tables, CrunchBot may thus construct a factor graph. (We later describe how such a set of candidates is obtained automatically). Inference would become highly complicated if CrunchBot had an infinite memory for shapelets, so in the present study we use a working memory (queue) of the seven most recent shapelets, and discard all others. At each t, new shapelets are read from the sensors, and inference is performed with the aim of obtaining the Maximum A Posterior (MAP) interpretation of their table causes, before the next time step begins,

$$\mathrm{MAP_t} = \arg_{\{T_j\}} \max P(\{T_j\}_j|\{S_k\}_k). \qquad (27)$$

Thus CrunchBot currently – naively – treats each time step as an independent inference problem. Limiting inference to the most recent shapelets also has the effect of working within a local 'fovea' of attention: if no recent shapelets are from distant areas, then only hypotheses around CrunchBot's location will be considered.

There is some subtlety in defining the meaning of MAP states in continuous parameter spaces. In the present study, we assume that discrete hypotheses $H_i(x, y, \theta, \Theta)$ (where $H \in \{S, L, T\}$) represent small but non-infinitesimal collections of possible (x, y, θ) poses, with probability

$$P(H((x-\tfrac{\delta}{2}, x+\tfrac{\delta}{2}), (y-\tfrac{\delta}{2}, y+\tfrac{\delta}{2}), (\theta-\tfrac{\delta}{2}, \theta+\tfrac{\delta}{2}), \Theta))$$

$$= \delta^3 p(H(x, y, \theta, \Theta)), \quad (28)$$

where δ is a small but nonzero constant, Θ are the remaining parameters, and p is the density.

CrunchBot uses the annealed [1] approximate Metropolis-Hastings sampler of algorithm 1 to perform inference. Unlike standard inference problems, object-based mapping is a form of scene analysis task, i.e. the number of objects in the world – and therefore the number and type of nodes in the network – is unknown in advance. Algorithm 1 uses blackboard-like priming and pruning heuristics integrated with the sampling, to control the size of the network. Each hypothesis in the current 'blackboard' set B maintains (amongst other parameters), pose parameters x, y, θ and a current parent. The current parent may be another hypothesis, or may be null. Importantly, hypotheses that are not currently 'true' (according to the sampler) are never stored in B. The set B acts as a factor graph as detailed in the previous section, and may be thought of as the contents of a blackboard [10].

To obtain unbiased samples from the true joint distribution, Metropolis-Hastings sampling requires detailed technical conditions to be met, which are complicated by the jumps between factor graphs of different structures and sizes. Reversible jump methods [26] provide a rigorous theoretical basis from which to define acceptance probabilities based on reweighting proposals. Future work should incorporate such theory, for now we heuristically choose the Q distributions[2] and r_i thresholds; and use the annealed original P distribution from the factor graph as a simple Gibbs [1] acceptance probability,

$$P(\text{accept } H_i) = P^\beta(H_i|mb(H_i)), \quad (29)$$

where $mb(H_i)$ is the Markov blanket of H_i containing its parents, rivals $riv(H_i)$, and children $ch(H_i)$, β is inverse temperature. The Markov blanket conditional is

$$P(H_i|mb(H_i)) = P(H_i|pa(H_i), cop(H_i), ch(H_i), riv(H_i))$$
$$= \frac{1}{Z} \frac{\Phi_a \Phi_b \Phi_c P(H_i|pa) P(ch|H_i)}{P(ch|(H_i))P(H_i|pa) + P(ch|\neg H_i)P(\neg H_i|pa)}$$
$$= \frac{1}{Z} \frac{\Phi_a \Phi_b \Phi_c p(H_i|pa) p(ch|H_i)}{\delta^3 (p(ch|H_i)p(H_i|pa) + p(ch|\neg H_i)p(\neg H_i|pa))},$$

where Z normalizes the factors contribution $\Phi_a \Phi_b \Phi_c$ only; δ is the constant of eqn. 28; $\Phi_a = \phi_a(H_i) \prod_{j \in pa(i)} \phi_a(H_j)$ includes missing children of H_i and also the missing child penalty for each parent of H_i which would have a missing child in the case where H_i is false; $\Phi_b = \prod_{j \in mb(i)} \phi_b(H_i, H_j)$ and $\Phi_c = \prod_{j \in mb(i)} \phi_c(H_i, H_j)$. The update allows computation to proceed using density functions rather than probabilities, but depends on the choice of the small constant, δ.

Newly proposed nodes must be linked to existing ones, so it is necessary to locate all potential parents $pa(H_i)$. A threshold radius in pose space is used, which limits this set to candidates which are close enough to have non-negligible generating probabilities, i.e.

$$pa(H_i) := \{H_j : P(H_i|H_j) \gg 0\}. \quad (30)$$

For computational efficiency it is useful to implement a spatial hash-table to look up these nearby hypotheses. This hash-table may also be reused to look up overlapping hypotheses in the computation of ϕ_c.

2.8. Mapping task

To remove the complexities of noisy odometry localisation during mapping, a noiseless-odometry simulation of CrunchBot in a world populated by six four-legged, table-like objects was implemented. The simulation is coded in C++ using the ODE physics engine (`www.ode.org`) for whisker contact detection. Source code is available in the supplemental material. The sensor noise levels are comparable to those found in the physical classifiers. The agent follows a fixed sequences of poses around the world and runs algorithm 1 once at each pose. There are $10 \times 10 \times 4$ poses, from 10 discrete x and y positions and four compass θ angles, as shown in fig. 6. To further simplify the present simulation, tables and table hypotheses all have fixed identical w_T^x, w_T^y and τ_T parameters; and physical (but not hypothesis) tables and have fixed identical w_T^L parameters.

2.9. Entropy based exploration

While the mapping experiment uses a fixed sequence of CrunchBot locations, we performed a further experiment to investigate a potential method for autonomous exploration. Preliminary experiments suggested that a common scenario which could enable exploration is the presence of ambiguity between rival high level table percepts, as illustrated in fig. 3.4. In these cases, CrunchBot's whiskers have seen a set of shapelets which enable the presence of one or two legs to be inferred, but the configuration of the rival tables remains ambiguous. In fig. 3.4, the Gibbs sampler is switching

[2] details can be found in the source code, however note that MH sampling can operate on *any* proposal Q so its precise form is unimportant. Better results are obtained as the approximate Q becomes close to the true P.

between tables $T1$ and $T2$ which are equally valid explanations of legs $L1$ and $L2$. We note that the presence of $T1$ and $T2$ are strongly correlated with the presence of legs $(L3, L4)$ and $(L5, L6)$ respectively. Therefore, observing the presence or absence of shapelets caused by any of these legs could resolve the high-level ambiguity about the tables.

This idea can be made precise by using a novel hierarchical version of well-known entropy-based exploration methods (reviewed in [45]), integrated into the Bayesian blackboard architecture. During mapping, CrunchBot's goal is to maximise knowledge about table locations (i.e. construct a map showing the tables.) We can write this goal as maximising entropy of the distribution over sets (maps) of tables $\{T_i\}_i$,

$$\max H(\{T_i\}_i) = \max_{\{T_i\}_i} (-\langle \log P(\{T_i\}_i) \rangle_{T_i}) \quad (31)$$

Future work could explore ways to compute this entropy exactly within the Bayesian blackboard framework; in the present study we simplify the computation by quantising a local region of space around CrunchBot into a square occupancy grid, and working with the probabilities that each grid cell is occupied by a leg or a table instead of the full distribution over sets of tables. As is common in occupancy grid methods, we assume (strongly, but falsely) that cell probabilities are mutually independent. Under these assumptions the local map probability is,

$$\prod_{x,y} P(T(x,y)) \quad (32)$$

where $P(T(x,y))$ is the table occupancy probability of the cell at location (x, y). From this we further approximate the entropy of the distribution over sets of tables, by the sum of the grid cell entropies,

$$H(\{T_i\}_i) \approx \sum_{x,y} H(T(x,y)), \quad (33)$$

which is the new goal to minimise. We quantise CrunchBot's next possible (greedy) actions as movements to the same set of grid cells, and assume that visiting a cell will always find any legs in that cell (as a result of the body-whisks, and radial distance and orientation reports). We ignore any evidence that may be collected during the path to reach the action cell in our approximation. Let the action of moving to and observing the cell (x_a, y_a) be $a(x_a, y_a)$.

Performing action $a(x_a, y_a)$ will (by assumption) answer with certainty the Boolean question of whether or not there is a leg $L(x_a, y_a)$ in cell (x_a, y_a). It may also reduce the entropy of the tables, by restricting possible percepts to those matching the presence or absence of this leg, as in fig. 3.4. We can compute this potential entropy change in advance, from the *current* location, before the action is performed, by computing the distribution of tables conditioned on the possible Boolean leg state b,

$$\{P(T(x,y)|L(x_a,y_a) = b)\}_{x,y}, \quad (34)$$

then computing $\Delta H =$

$$H(\{T(x,y)\}_{x,y}|L(x_a,y_a)) - H(\{T(x,y)\}_{x,y}|\neg L(x_a,y_a)) \quad (35)$$

$$= \sum_{x,y} H(T(x,y)|L(x_a,y_a)) - H(T(x,y)|\neg L(x_a,y_a)) \quad (36)$$

for each $a(x_a, y_a)$. The action with the largest difference in entropy is the most informative about the table distribution and is thus could be a useful candidate to explore next.

In practice, we need a way to approximate equation 34. In the present study, we experimented by using samples of table sets $\{T_i\}_i$ drawn from the existing Metropolis-Hastings sampler. For each sample of tables, the Boolean cell occupancies are computed (by drawing the tables onto a grid using a graphics library, www.cairographics.org), then normalised occupancy frequencies summed over samples used as approximations to occupancy probabilities $P(T(x,y))$. To avoid estimation bias due to the changing annealing temperature, we extended the annealing cycle with a fixed, high temperature phase at $\beta = 1/7.5$ for $N = 20$ steps before beginning to reduce the temperate for the MAP optimisation annealing phase. The high temperature was used to allow the sampler to jump often between minima, as at $\beta = 1$ there is little probability of the rivals and missing children – which are necessary to transition to alternative ambiguous table percepts – ever occurring.

3. Results

3.1. Radial distance reports

Peak deflection magnitude for each contact is shown in Fig. 3.1. Standard deviation of error for radial distance estimation is shown in Table 1.

Peak deflection magnitude for contacts along the shaft of the whisker. Standard error for the regression is 4.98mm

Standard classification error is very low, typically less than 5mm over the 60mm range tested. For some whiskers classification error is even lower, below 2mm.

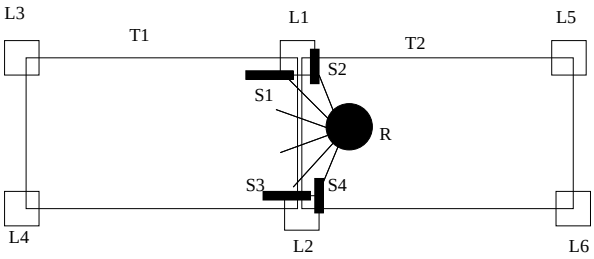

Figure 7: Typical ambiguous table scenario. Here the robot R has reported shapelets $S1-4$ and inferred legs $L1,2$ unambiguously. But these legs are equally compatible with tables $T1,2$ having legs $L3,4$ and $L5,6$. Exploring any of these four legs would remove the ambiguity about the tables.

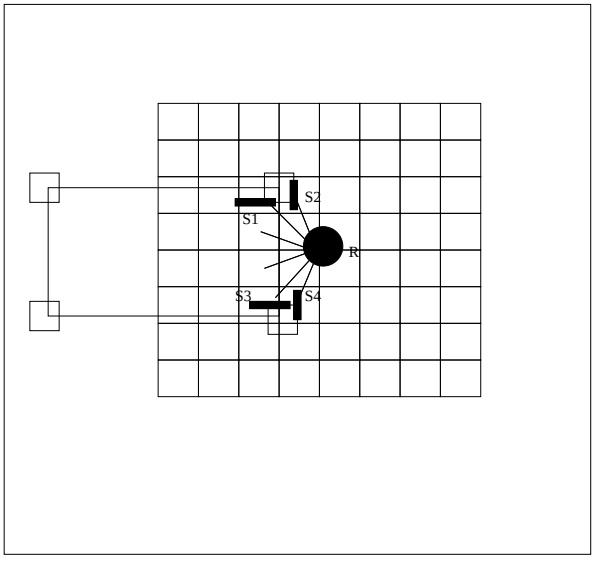

Figure 8: Illustration of the foveal grid used in entropy mapping. The grid covers a small local region around the robot, within the arena. Table and leg occupancies are recorded in each cell and used to find regions of interest, such as the ambiguity-resolving legs of fig. 3.4.

W	1	2	3	4	Combo
σ	2.78mm	1.82mm	4.37mm	5.68mm	4.98mm

Table 1: Standard classification error for radial distance estimation on the CrunchBot mobile robot. Results are given for each whisker in turn, and or all the whiskers together. W = whisker, σ = standard error of classification

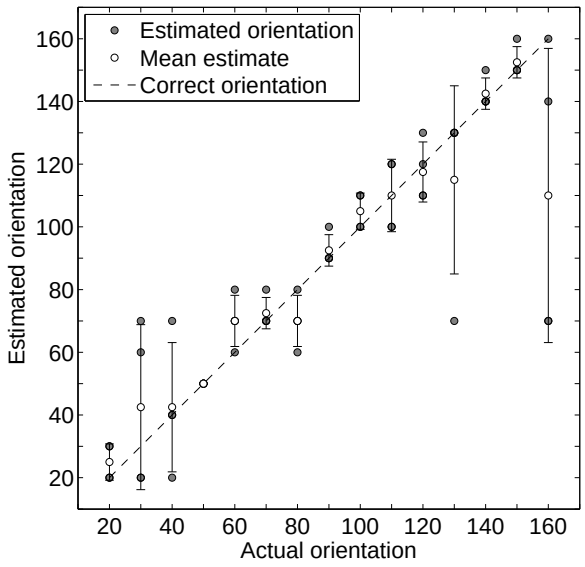

Figure 9: Surface orientation discrimination error with a template based classifier. Mean classification error is -2.2°, standard error is 21.2°

These results compare favourably with previous work under highly controlled conditions where speed was variable. This indicates that the noise in the odometry is low enough to ensure a consistent contact force and speed during the short periods of contact.

3.2. Surface orientation reports

Fig. 9 displays surface orientation estimation performance for the template based classifier. Mean classification error is -2.2°, standard error is 21.2°. Classification performance is best for orientations near 90°, with larger errors being made for large and small orientations. This may be due to fewer whiskers making contact with the surface at the extreme orientations, providing less information to the classifier with which to base a classification.

3.3. Hierarchical object mapping

The parameter values used were: $c_T = 0.4, c_L = 0.1, c_S = 0.05, \epsilon_a = 0.3\epsilon_b = 0.1, \epsilon_c = 0.1, \sigma_r = 0.2, \sigma_\theta = \pi/32, \delta^3 = 0.4$. Here, σ_r and σ_θ were

chosen to be similar to the noise levels found in the physical experiments, while the other parameters were set by hand trial-and-error to give good results in the particular arena used in the simulation. The annealing schedule was $\beta_i = \exp\{7.5 - 0.5i\}$. Steps in the inference are illustrated in the supplemental video material. The MAP hypothesis sets from all poses are collated and plotted onto a map of the arena in fig. 10. Comparing against the ground truth in fig. 6, the collated plot shows that table hypotheses are usually found in the correct locations, corresponding to the real tables. The average number of whiskers contacting tables at each pose having at least one table contact is 4.2 ± 1.7. As we would expect from such a sparse amount of data, there are thus many incorrect hypotheses found in MAPs of the form shown in fig. 5. These are created from poses which do not provide enough information about the tables to resolve ambiguities, for example when the robot is close enough to touch two legs but no third leg as in fig. 5. Also of interest in the results are the many table hypotheses perceived around the edge of the arena. These are due to the agent observing shapelets from contact with the walls around the arena. The system does not (yet) have perceptual models of walls, so the best available explanations for such shapelets are those which postulate tables with legs at these shapelet locations. (This is a form of perceptual relativism: lacking a WALL concept, the system explains the data using its best available TABLE theories.) Similar plots for noiseless and highly noisy sensor cases are shown in figs. 11 and 12 for comparison. In both cases, the approximate locations of inferred tables are similar, though the accuracy of inferred table poses depends on the noise.

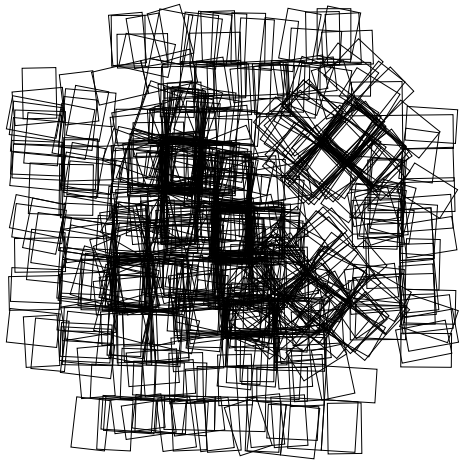

Figure 10: Montage showing collection of inferred tables from each independent robot pose, for realistic [19] ($\sigma_r = 0.1, \sigma_\theta = \pi/32$) sensors.

3.4. Entropy based exploration

While CrunchBot's annealed Bayesian Blackboard was sucessful in finding tables in the arena, it showed less success in finding good exploration locations. We have showed mathematically how to set up exploration with hierarchical objects as a conditional entropy minimisation task, however we noted that computation of eqn. 34 requires approximation and chose to use occupancy frequencies from the high-temperature sampler to approximate the occupancy probabilities. The resulting behaviour of the simulated CrunchBot was indistinguishable from random movements, and CrunchBot did not appear to explore disambiguating legs as in fig. 7. . Analysis of the entropy maps gives some idea of the failure of the approximation of eqn. 34, and examples are shown in fig. 13. The problem is that the entropy differences are dominated by the probabilities of the empty space around the tables. Beginning with a flat prior on

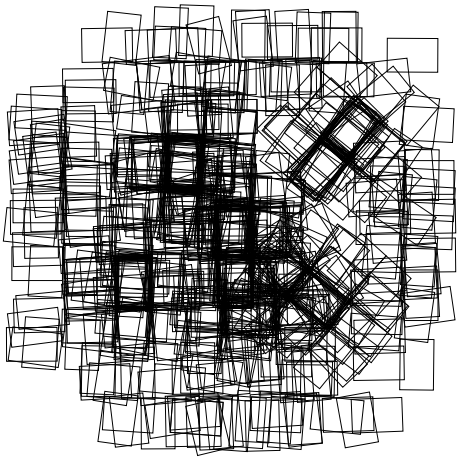

Figure 11: Montage showing the collection of inferred tables from each independent robot pose, for ideal, noiseless sensors.

Figure 12: Montage showing the collection of inferred tables from each independent robot pose, for very noisy ($\sigma_r = 0.5, \sigma_\theta = \pi/8$) sensors.

occupancy, and fusing in table percepts conditioned on leg states, the problem is that some leg states are visited more times than others. So as well as adding to the table distributions, the observations of empty space also deepen the probability of non-occupancy there. As we do not have access to an infinite number of observations, we do not reach the true occupancy distributions, but instead move towards them in proportion the number of observations. But the number of observations differs according to the leg state, meaning that the probability of non-occupancy for the background is higher for leg states that are visited more by the sampler. Thus the differences in background probability between common and uncommon leg states can become large, and contribute more to ΔH than the actual table distribution. This experiment shows that while the mathematical model and initial approximations may be sound, the frequency based approximation to the conditional occupancy probabilities is poor in this setting, and further work should be done to find better approximations. (This is potentially a large area of research, for example a recent entire PhD thesis was devoted to similar problems arising from a much more simplified grid environment, [45]).

4. Discussion

Map building with only CrunchBot's whisker sensors is a difficult task, and our previous paper [21] gave some indication of the problems faced by conventional particle filtering and EKF SLAM style approaches to

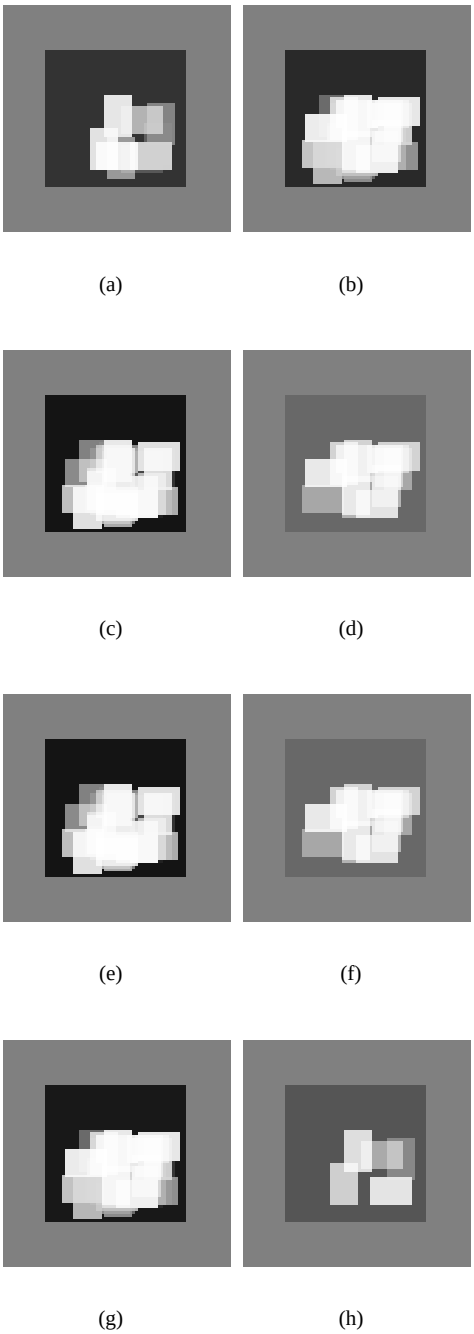

Figure 13: Examples of pairs (rows) of entropy maps. Each pair shows the table grid cell occpancy probabilities, conditioned on the presence (left) or absence (right) of a leg at some location in the arena. It can be seen that the entropy change is dominated by the shading of the background rather than the table distribution proper.

the mapping and navigation problems. In contrast, the present paper has shown how to combine signal processing for extraction of information about distance and surface orientation from physical whiskers with strong hierarchical priors about objects to compensate for the poverty and locality of the initial touch information.

After demonstrating that practical extraction of distance and orientation are possible on the real world, mobile CrunchBot platform (unlike previous work which has performed similar demonstrations in highly constrained, fixed-base environments), we then showed how these reports can be fused together using a Bayesian Blackboard to perceive high level objects such as tables that caused the reports.

Many simplifications were made in the present blackboard implementation, which future versions of the system should relax. The 'maps' presented here are simply the collation of many independent $\mathrm{MAP}_{\mathrm{t}}$ inferences made from the different poses, and no information is shared between poses. Storing longer-term memories of shapelets and fusing them into the inferences would obviously allow a more refined map of the arena to be constructed: at present each table shown in the results has been inferred from typically 4.2 ± 1.7 shapelets only, which is extremely sparse. To avoid combinatorial explosion from handling many historical shapelets, one approach would be to discard very old shapelets memories but preserve only the locations of recognised tables and other high level objects, similar to the approach used in [16]. Such an approach raise interesting questions and analogies about the biological split between perception of the immediate local present (thought to occur in cortex), and perception of distinct locations and the past (thought to occur in hippocampus [8]). The present system makes no use of negative evidence, i.e. the observed absence of shapelets on non-contacting whiskers: future 'null-shapelet' observations could report that a shapelet-sized region has been swept out by whiskers and found to be empty; these could then be used to remove some of the ambiguous percepts. The heuristic threshold constants in the proposal distribution should be replaced with Reversible Jump MCMC reweightings to remove bias in the sampling distribution (although in practice the heuristic thresholds can work well, as ultimately only the annealed MAP is sought, rather than an approximation to the whole distribution).

The template classifier was able to discriminate the orientation of a surface but was not trained to discriminate other sorts of contacts, for example with the corners of objects. In principle it is possible to train a template classifier on every possible contact in the arena. However collecting such a data set would be impractical, and the computations involved in comparing incoming data to templates for every possible contact could be cumbersome. An alternative approach is to extract features from the tactile data, as was done radial distance estimation in this paper, has been done in the field of haptic touch [51],[48] and is commonly used in vision [29], and audition [4]. It has been proposed that cells in the thalamus and cortex of the rat are encoding features [42],[28] in this way. Though reliable features can be extracted for radial distance estimation in this paper, and contact speed on a stationary robot [13], it is unclear what other features can be extracted from whisker deflection signals for discriminating different kinds of object properties. In our own lab we are developing features for whisker based tactile sensing of contact geometry [14] and texture [18]. In future we hope to be able to combine features for diverse tactile properties in rich environments into a coherent system onboard a mobile robot, which in turn would provide reports that could be used as inputs to hierarchical object models as presented in this study.

We showed how to frame the exploration question for hierarchical objects in terms of entropy, in a related but novel approach from standard entropy grid based mapping (known as Active SLAM, eg. [6][3]), but found an initial computational approximation to entropy to be lacking for the implementation purpose. It is possible that links to work of [45] may be useful to produce better approximations here in future work.

Importantly, the present system operates in a world having only one size and texture of table (though tables may have different leg sizes). Enlarging the parameter space to range over tables sizes and textures will allow inference of more realistic four-legged objects such as different kinds of chairs and desks. Other types of objects could also be introduced, such as walls, kitchen units and radiators. The Bayesian blackboard architecture is able to automatically select between rival object models, treating them as rival hypotheses [16]. However, as the number of models and parameters grows, sampling of course becomes less efficient. For example, it becomes less probable that a perfectly-fitting table will ever be proposed. (Even though once proposed, it will tend to remain accepted for having such a good fit.) Future work should investigate the use of 'smart proposals' which are classical heuristic object detectors (e.g. Hough transforms to find edges and corners) but re-purposed as Metropolis-Hastings proposals in the

[3]This paper, together with [23, 50], also gives ideas for how future CrunchBot versions could recover from getting lost during failed inference, by monitoring uncertainty about location.

Bayesian Blackboard. When combined with RJ-MCMC acceptance probabilities, this gives a way to speed up the proposals but retain the probabilistic semantics. Further research should also extend the parameter space to multiple textures, and incorporate our previous research on CrunchBot's texture recognising abilities [21]. The hierarchical objects used on the blackboard have similarities to those of classic inductive learning [56] and modern versions of these algorithms could perhaps form the basis for the automatic learning of new object types.

Future integration work should port CrunchBot's hierarchical mapping components from simulation to its physical platform using a standard API such as Player/Stage. The latter will involve handling the full SLAM problem rather than just mapping – recall the the reason simulation was used in the mapping part of this article was that it allows us to assume zero odometry noise and ignore the localisation part of SLAM. We have recently begun to demonstrate simple whiskered SLAM in very small environments using grid maps [17] which now provides a baseline to compare future hierarchical objects against. Using hierarchical objects maps should improve performance, for example by enabling CrunchBot to localise when encountering the back of a table which was previously touched only from its front. If localisation is achieved in this way, then it should be possible to treat observations of tables and other objects as landmarks in a standard EKF type approach. New forms of loop-closure in SLAM may become possible by again recognising different parts of the same hierarchical object, for example CrunchBot may be able to close a loop by recognising a previously unseen leg of a previously seen table. Tracking of moving hierarchical objects may become possible by fusing CrunchBot's hierarchical models with the SLAP algorithm [39].

References

[1] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, 1988.

[2] A. Ahl. The role of vibrissae in behavior: a status review. *Veterinary Research Communications*, 10(1):245–268, 1986.

[3] K. Beevers and W. Huang. Slam with sparse sensing. In *ICRA*, 2006.

[4] J. B. Bello and M. Sandler. Techniques for automated music transcription. *International Symposium on Music Information Retrieval*, 2000.

[5] T. Binford and T. Levitt. Evidential reasoning for object recognition. *IEEE Tranactions on Pattern Analysis and Machine Intelligence*, 2003.

[6] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri. An application of kullback leiber divergence to active slam and exploration with particle filters. In *IROS*, 2010.

[7] G. Carvell and D. Simons. Biometric analyses of vibrissal tactile discrimination in the rat. *J. Neurosci.*, 10(8):2638, 1990.

[8] C.Fox and T.Prescott. Hippocampus as unitary coherent particle filter. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2010.

[9] M. E. Diamond, M. von Heimendahl, P. M. Knutsen, D. Kleinfeld, and E. Ahissar. 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, 9(8):601–612, 2008 Aug.

[10] L. Erman, F. Hayes-Roth, V. Lesser, and R. Reddy. The Hearsay-II system. *ACM Computing Surveys*, 12(2), 1980.

[11] M. Evans, C. Fox, M. Pearson, and T. Prescott. Spectral Template Based Classification of Robotic Whisker Sensor Signals in a Floor Texture Discrimination Task. *Proceedings TAROS2009*, pages 19–24, 2009.

[12] M. Evans, C. Fox, and T. Prescott. Tactile discrimination using template classifiers. *From Animals to Animats SAB2010.*, 2010.

[13] M. Evans, C. W. Fox, M. J. Pearson, and T. J. Prescott. Whisker-object contact speed affects radial distance estimation. In *Proc IEEE ROBIO*, 2010.

[14] M. H. Evans, C. Fox, M. J. Pearson, and T. J. Prescott. Object location, orientation, and velocity extraction from artificial vibrissal signals. In *Society for Neuroscience Abstracts. Society for Neuroscience (Program No. 174.8/ Z12)*, 2009.

[15] M. Fend. Whisker-based texture discrimination on a mobile robot. *Advances in Artificial Life*, pages 302–311, 2005.

[16] C. Fox. ThomCat: A Bayesian blackboard model of hierarchical temporal perception. In *Proc. FLAIRS*, 2008.

[17] C. Fox, M. Evans, M. Pearson, and T. Prescott. Tactile slam with a biomimetic whiskered robot. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012, forthcoming, preprint at 5m.org.uk.

[18] C. Fox, B. Mitchinson, M. Pearson, A. Pipe, and T. Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 26(4):223–239, 2009.

[19] C. Fox, M. Pearson, B. Mitchinson, T. Pipe, and T. Prescott. Simple features for texture classification from robot whisker strains. *Somatosensory and Motor Research*, 24(3):139–162, 2007.

[20] C. Fox and T. J. Prescott. Mapping with sparse local sensors and strong hierarchical priors. TAROS 2011 (this volume).

[21] C. W. Fox, M. Evans, N. Lepora, A. Ham, and T. J. Prescott. Crunchbot: a mobile whiskered robot platform. In *Proceedings of Towards Autonomous Robots (TAROS)*, 2011.

[22] C. W. Fox, M. Evans, M. J. Pearson, and T. J. Prescott. Towards temporal inference for shape recognition from whiskers. In *Proc TAROS*, 2008.

[23] D. Fox. Adapting the sample size in particle filters through kld-sampling. *International Jounral of Robotics Researc*, 12, 2003.

[24] G. Gallagher, S. S. Srinivasa, J. A. Bagnell, and D. Ferguson. Gatmo: A generalized approach to tracking movable objects. In *ICRA*, 2009.

[25] V. Gopal and M. J. Z. Hartmann. Using hardware models to quantify sensory data acquisition across the rat vibrissal array. *Bioinspir Biomim*, 2(4):S135–45, 2007.

[26] P. Green. Reversible jump Markov chain Monte Carlo computation. *Biometrika*, 82(4):711–732, 1995.

[27] J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. Diamond, and P. Konig. Texture signals in whisker vibrations. *Journal of neurophysiology*, 95(3):1792, 2006.

[28] S. Jadhav, J. Wolfe, and D. Feldman. Sparse temporal coding of elementary tactile features during active whisker sensation. *Nat Neurosci*, 12:792–800, 2009.

[29] L. Juan and O. Gwun. A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(5), 2010.

[30] M. Kaneko, N. Kanayama, and T. Tsuji. Active antenna for contact sensing. *IEEE Transactions on robotics and automation*,

14(2):278–291, 1998.

[31] D. Kim and R. Moller. Biomimetic whiskers for shape recognition. *Robotics and Autonomous Systems*, 55(3):229–243, 2007.

[32] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[33] K. B. Laskey and P. C. da Costa. Of starships and klingons: Bayesian inference for the 23rd century. In *Proc. UAI*, 2005.

[34] N. Lepora, M. Evans, C. Fox, M. Diamond, K. Gurney, and T. Prescott. Naive Bayes texture classification applied to whisker data from a moving robot. *Proc. IEEE World Congress on Comp. Int. WCCI2010*, 2010.

[35] Melexis. www.melexis.com/assets/mlx90333_datasheet_5276.aspx.

[36] B. Milch. *Probabilisitic Models with Unknown Objects*. PhD thesis, UC Berkeley, 2006.

[37] M. Mitchell. *Analogy-Making as Perception*. MIT, 1993.

[38] B. Mitchinson, C. J. Martin, R. A. Grant, and T. J. Prescott. Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proc Biol Sci*, 274(1613):1035–1041, 2007.

[39] M. Montemerlo, W. Whittaker, and S. Thrun. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, 2002. ICRA.

[40] J. Pearl. *Intelligient Reasoning with Probabalistic Networks*. Morgan Kaufmann, 1988.

[41] J. Pearl. *Causality*. Cambridge University Press, 2000.

[42] R. Petersen, M. Brambilla, M. Bale, A. Alenda, S. Panzeri, M. Montemurro, and M. Maravall. Diverse and temporally precise kinetic feature selectivity in the VPm thalamic nucleus. *Neuron*, 60(5):890–903, 2008.

[43] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng. Touch based perception for object manipulation. In *ICRA*, 2007.

[44] T. Prescott, M. Pearson, B. Mitchinson, J. Sullivan, and A. Pipe. Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, 16(3):42–50, 2009.

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

[46] A. Schultz, J. Solomon, M. Peshkin, and M. Hartmann. Multifunctional whisker arrays for distance detection, terrain mapping, and object feature extraction. *Proc. ICRA2005*, pages 2588–2593, 2005.

[47] A. Seth, J. McKinstry, G. Edelman, and J. Krichmar. Texture discrimination by an autonomous mobile brain-based device with whiskers. In *Proc. IEEE Int. Conf. Robot. Autom. ICRA2004*, pages 4925–4930, 2004.

[48] J. Sinclair, J. Kuo, and H. Burton. Effects on discrimination performance of selective attention to tactile features. *Somatosensory & Motor Research*, 17(2):145–157, 2000.

[49] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. C. Romea, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and J. M. Vandeweghe. Herb: a home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, January 2010.

[50] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *In RSS*, pages 65–72, 2005.

[51] S. Stansfield. Primitives, features, and exploratory procedures: Building a robot tactile perception system. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1274–1279. IEEE, 1986.

[52] C. Sutton, B. Burns, C. Morrison, and P. R. Cohen. Guided incremental construction of belief networks. In *Proceedings of the Fifth International Symposium on Intelligient Data Analysis*, 2003.

[53] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT, 2006.

[54] C.-C. Wang, C. Thorpe, S. Thrun, M. Herbert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *International Jounral of Robotics Research*, 26:889, 2007.

[55] W. Welker. Analysis of sniffing of the albino rat. *Behaviour*, pages 223–244, 1964.

[56] P. Winston. Learning structural descriptions from exam- ples. In *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

[57] Y. Zhang, J. Liu, G. Hoffmann, M. Quilling, K. Payne, P. Bose, and A. Andrews Zimdars. Real-time indoor mapping for mobile robots with limited sensing. In *Mobile Adhoc and Sensor Systems (MASS), IEEE 7th International Conference on*, 2010.