Mapping with sparse local sensors and strong hierarchical priors

Charles W. Fox and Tony J. Prescott

Active Touch Laboratory at Sheffield
University of Sheffield, UK.
charles.fox@sheffield.ac.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 in environments where only sparse, local sensory information is available. To compensate for these weak likelihoods, we make use of strong hierarchical object priors. Hierarchical models were popular in classical blackboard systems but are here applied in a Bayesian setting and novelly deployed as a mapping algorithm. We give proof of concept results, intended to demonstrate the algorithm's applicability as a part of a tactile SLAM module for the whiskered SCRATCHbot mobile robot platform.

1 Introduction

The paradigm case for robotic mapping, as in Simultaneous Localisation and Mapping (SLAM) problems [27], considers a mobile robot with noisy odometry and laser scanners. 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 [27].

In contrast, we consider the mapping problem in environments where only sparse, local sensory information is available. For example, a fire-fighting robot building up a map in a smoke-filled house cannot rely on vision or laser scanners functioning at all times, and could instead operate by feeling its way around with touch sensors. Proof that this type of navigation is possible is found in biology: electric fish make use of highly localised electric field sensors [13] and rats navigate through dark underground tunnels using their whiskers [4, 2], both having ranges of a few centimetres. In robotics, touch sensors are relatively cheap in both material and computational processing terms, and their use has previously been considered to enhance navigation in cheap household robots [17, 6]. (Related work on research robot platforms includes [24, 23, 15, 14, 8]).

As an example of this type of mapping, we consider the case of a mobile robot having six whiskers, able to report the (noisy) locations and orientations of contacts with surfaces. Such mechanical sensors and computational classifiers have previously been demonstrated in [17, 6, 7], and are able to report locations, orientations and textures of contact points (note that textures are especially difficult to report using other sensor modalities). The present mapping algorithm is intended to form part of a future

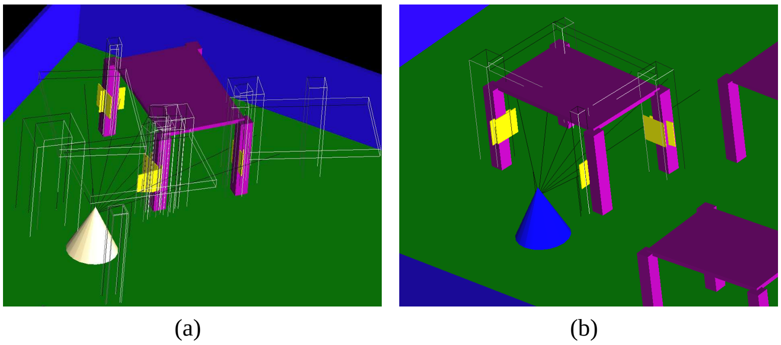

Fig. 1. (a) Simulation screen-shot at high annealing temperature. Many hypothesised (wire-frame) tables and legs are on the blackboard, primed by the shapelets (yellow rectangles) contacted by the robot (cone)'s whisker sensors, in an arena containing a physical table (pink). (b) Simulation screen-shot at low annealing temperature. A single table hypothesis remains, aligned correctly with the physical table.

SLAM navigation module for the whiskered SCRATCHbot hardware platform [21], but here we give a proof of concept mapping-only algorithm in a simulated and simplified microworld. It is the first work to begin fusing whisker contact reports to perform mapping.

To compensate for the sparseness of the sensory information available from short-range touch sensors, we make use of strong, hierarchical priors about objects in the world. Hierarchical object recognition models were popular in classical, symbolic AI in the guise of blackboard systems [5, 19, 3] but have recently been recast in terms of dynamically constructed Bayesian networks [10, 18, 16, 26]. Here we provide a novel application of Bayesian blackboards to the robotic mapping problem.

Object based mapping models have recently appeared [28, 11, 25, 22] which use laser sensors to recognise and learn complex spatial models. However in the sparse local sensor case, this level of detail is unavailable, and only a few contact points may be present. Thus we go beyond the use of individual movable objects, to use strong hierarchical model priors. For example, on recognising a table leg, we may then 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 blackboard systems. However, following [10], we treat such heuristics as approximations to inference in a dynamically-constructed, Monte Carlo Markov Chain (MCMC) sampling Bayesian network, endowing them with probabilistic semantics.

2 Methods

Consider the task of building a map of an arena populated by four-legged table-like objects as in figs. 1(a) and 1(b). (Such objects could include chairs and desks for example). A mobile whiskered agent moves along a predetermined trajectory of location-angle

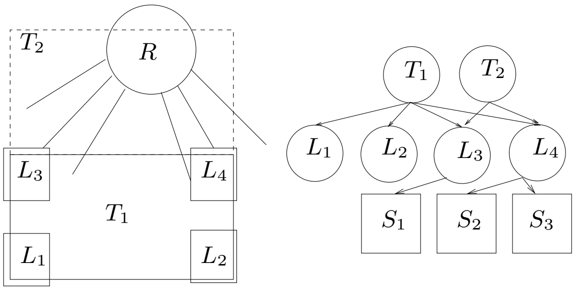

Fig. 2. Hierarchical object recognition. *Left:* Robot R (circle) with six whiskers (lines) makes tactile contact with legs L_j (squares) of a hypothesised table T_1 (rectangle). The two contact points ('shapelets') on the right are sufficient to infer the location of the corner of leg L_4. Coupled with 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.

poses, (x^t, y^t, θ^t), around the arena, over discrete time steps t. At each time step, its whiskers, $w \in 1:6$, report egocentric estimates of the radial distance r to, and surface normal ϕ and texture τ of, any contacts made,

$$\hat{r}^t_w = r^t_w + \varepsilon_r, \tag{1}$$

$$\hat{\phi}^t_w = \phi^t_w + \varepsilon_\phi, \tag{2}$$

$$\hat{\tau}^t_w = \tau^t_w + \varepsilon_\tau, \tag{3}$$

where ε are i.i.d. Gaussian noises having zero mean and standard deviations $\sigma^w_r, \sigma^w_\phi, \sigma^w_\tau$ respectively. (Such reports are currently available from the whiskered SCRATCHbot hardware platform, [21, 17, 6]). Assuming perfect robot localisation in the present study only, these estimates are converted into allocentric Cartesian coordinates to give *shapelet reports,* which are tuples $S(x_S, y_S, \phi_S, \tau_S)$.

2.1 Static structures: generative models

Tables, T, are parametrised by tuples, $T(x_T, y_T, \theta_T, w^x_T, w^y_T, w^L_T, \tau_T)$, where x, y, θ is the pose, w^x_T and w^y_T 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. We assume a flat prior probability *density* generating tables in the world,

$$p(T(x_T, y_T, \theta_T, w^x_T, w^y_T, w^L_T, \tau_T)|\emptyset) = c_T, \tag{4}$$

where c_T is a (non-normalising) constant.

If a table T exists, its presence causes (as in [20]) the presence of four leg objects,

$$L(x_L, y_L, \theta_L, w_L, \tau_L, T), \tag{5}$$

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

$$\begin{aligned} p(L(x_L, y_L, \theta_L, w_L, \tau_L, T) | T(x_T, y_T, \theta_T, w_T^L, \tau_T)) \\ = \alpha_L \exp\left(-\varDelta_{TL}\right), \end{aligned} \tag{6}$$

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

$$\begin{aligned} \varDelta_{TL} = \min_i \left(\frac{(x_T^i - x_L)^2 + (y_T^i - y_L)^2}{\sigma_r^2} \right) + \left(\frac{\theta_T - \theta_L}{\sigma_\theta} \right)^2 \\ + \left(\frac{w_T^L - w_L}{\sigma_w} \right)^2 + \left(\frac{\tau_T - \tau_L}{\sigma_\tau} \right)^2, \end{aligned} \tag{7}$$

where $0 \leq i \leq 3$, 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. 5) 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,

$$p(S(x_S, y_S, \theta_S, \tau_S) | L(x_L, y_L, \theta_L, w_L, \tau_L, T)) = \alpha_S \exp\left(-\varDelta_{LS}\right) \tag{8}$$

where

$$\varDelta_{LS} = \left(\frac{r}{\sigma_r^S} \right)^2 + \left(\frac{f(\theta_L) - \theta_S}{\sigma_\theta^S} \right)^2 + \left(\frac{\tau_L - \tau_S}{\sigma_\tau^S} \right)^2, \tag{9}$$

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),

$$p(L(x_L, y_L, \theta_L, w_L, \tau_L, \emptyset) | \emptyset) = c_L, \tag{10}$$

$$p(S(x_S, y_S, \theta_S, \tau_S) | \emptyset) = c_S, \tag{11}$$

with constants such that the marginalised densities,

$$p(S(x_L, y_L, \theta_L)) < p(L(x_L, y_L, \theta_L)) < p(T(x_L, y_L, \theta_L)), \tag{12}$$

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. 5, shapelets may be caused by *mixtures* of multiple leg hypotheses and by the null prior (eqn. 11). 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,

$$P(x_i | pa(x_i)) = 1 - \prod_{x_j \in pa(x_i)} \left(1 - P(x_i | x_j)\right). \tag{13}$$

where $pa(x_i)$ denotes the set of parents of generic node x_i. As we use probability *density* functions we require the continuous version of noisy-OR, proved in the Appendix:

$$p(x_i|pa(x_i)) = \sum_{x_j \in pa(x_i)} p(x_i|x_j). \tag{14}$$

We allow legs to be caused by a mixture of their *single* specified parent (i.e. the T parameter in eqn. 6) and null prior (eqn. 10), using a similar combination rule. Tables are caused by the null prior only (eqn. 4).

Taken together, the equations in this section define a Bayesian network for any given collection of tables, legs and shapelets as shown in fig. 2. 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

$$P(\{x_i\}_i) = \prod_i P(x_i|pa(x_i)), \tag{15}$$

To model these constrains, we extend the Bayesian network to the factor graph,

$$P(\{x_i\}_i) = \frac{1}{Z}\left(\prod_i P(x_i|pa(x_i))\right) \times \left(\prod_{ij} \phi_c(x_i,x_j)\phi_b(x_i,x_j)\right)\left(\prod_i \phi_a(x_i)\right), \tag{16}$$

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

$$\phi_a(x_i) = \epsilon_a^m, \tag{17}$$

$$\phi_b(x_i,x_j) = \epsilon_b^v, \tag{18}$$

$$\phi_c(x_i,x_j) = \epsilon_c^r, \tag{19}$$

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. 5).

2.2 Inference

For a given set of shapelet observations and a set of candidate hierarchical legs and tables, we may thus construct a factor graph. (We later describe how such a set of candidates is obtained automatically). Inference becomes highly complicated if the agent has 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,

$$\mathrm{MAP_t} = \arg_{\{T_j\}} \max P(\{T_j\}_j | \{S_k\}_k). \tag{20}$$

Thus we currently – naively – treat 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 the agent’s location will be considered.

Algorithm 1 Approximate Metropolis-hasting proposals generation.

```
for each time step $t$ do
  update shapelet queue $S$ by reading sensors
  for each annealing inverse temperature $\beta$ 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
```

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

$$\begin{aligned} P(H((x - \tfrac{\delta}{2}, x + \tfrac{\delta}{2}), (y - \tfrac{\delta}{2}, y + \tfrac{\delta}{2}), (\theta - \tfrac{\delta}{2}, \theta + \tfrac{\delta}{2}), \varTheta)) \\ = \delta^3 p(H(x, y, \theta, \varTheta)), \end{aligned} \tag{21}$$

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

We use 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 [5].

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 [12] 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[1] and r_i thresholds; and use the annealed original P distribution from the factor graph as a simple Gibbs [1] acceptance probability,

$$P(\text{accept } H_i) = P^{\beta}(H_i | mb(H_i)), \tag{22}$$

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

$$\begin{aligned} P(H_i | mb(H_i)) &= P(H_i | pa(H_i), cop(H_i), ch(H_i), riv(H_i)) \\ &= \frac{1}{Z} \frac{\varPhi_a \varPhi_b \varPhi_c P(H_i|pa) P(ch|H_i)}{P(ch|(H_i))P(H_i|pa) + P(ch|\neg H_i) P(\neg H_i|pa)} \\ &= \frac{1}{Z} \frac{\varPhi_a \varPhi_b \varPhi_c p(H_i|pa) p(ch|H_i)}{\delta^3 (p(ch|H_i) p(H_i|pa) + p(ch|\neg H_i) p(\neg H_i|pa))}, \end{aligned}$$

where Z normalizes the factors contribution $\varPhi_a \varPhi_b \varPhi_c$ only; δ is the constant of eqn. 21; $\varPhi_a = \phi_a(H_i) \prod_{j \in pa(i)} \phi_a(H_j)$ 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; $\varPhi_b = \prod_{j \in mb(i)} \phi_b(H_i, H_j)$ and $\varPhi_c = \prod_{j \in mb(i)} \phi_c(H_i, H_j)$. The update

[1] 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.

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.

$$pa(H_i) := \{H_j : P(H_i|H_j) \gg 0\}. \tag{23}$$

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.

3 Results

We have implemented a simple simulation of a whiskered robot in a world populated by six four-legged, table-like objects, in a simple mapping task. The simulation is coded in C++ using the ODE physics engine (`www.ode.org`) for collision detection. Source code is available on request, and contains handset values for all parameters described in this paper. The sensor noise levels are comparable to those found in strain template shapelet classifiers [9]. 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. 3. 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.

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. 4. Comparing against the ground truth in fig. 3, 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. 2. 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. 2. 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. 5(a) and 5(b) for comparison. In both cases, the approximate locations of inferred tables are similar, though the accuracy of inferred table poses depends on the noise.

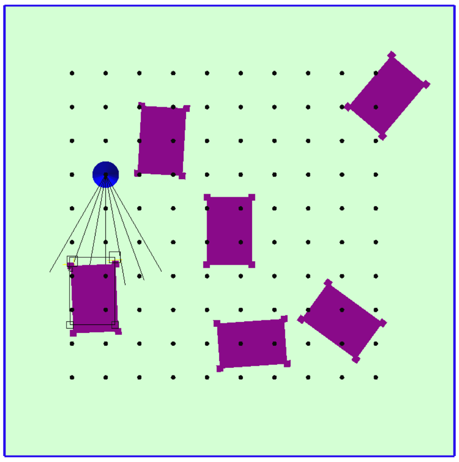

Fig. 3. Overhead view showing ground truth table configuration, and locations (black dots) of the discrete poses occupied by the robot. There are four angle poses at each location, facing in compass directions.

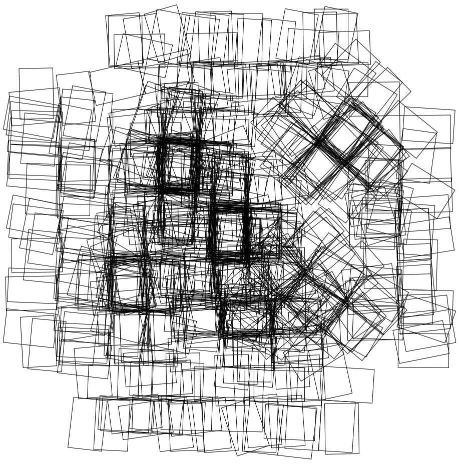

Fig. 4. Montage showing collection of inferred tables from each independent robot pose, for realistic [9] ($\sigma_r = 0.1, \sigma_\theta = \pi/32$) sensors.

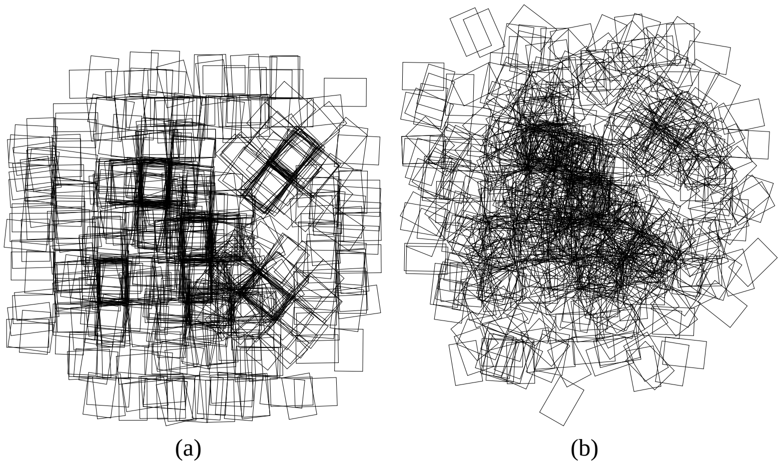

Fig. 5. (a) Montage showing the collection of inferred tables from each independent robot pose, for ideal, noiseless sensors. (b) Montage showing the collection of inferred tables from each independent robot pose, for very noisy ($\sigma_r = 0.5, \sigma_\theta = \pi/8$) sensors.

4 Discussion

We have presented a proof-of-concept implementation of a novel framework for hierarchical object-based mapping from sparse local sensors, such as whiskers and other types of touch; but also applicable to sensors such as low-power or covert short range scanners; or local field-based sensors as used by electric fish.

Many simplifications were made in this proof-of-concept, which future versions of the system should relax. The results presented here are simply the collation of many independent MAP_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. The present system makes no use of negative evidence, i.e. the observed absence of shapelets on non-contacting whiskers: this could 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).

Importantly, the proof-of-concept simulation 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 [10]. 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.) We plan to 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 Bayesian Blackboard. When combined with RJ-MCMC acceptance probabilities, this gives a way to speed up the proposals but retain the probabilistic semantics.

We next hope to extend our implementation to recognise several types of object of varying size, and move from simulation to the SCRATCHbot platform [21], which is currently able to report shapelets of the form used in simulation. SCRATCHbot includes noisy odometry, so will require our mapping system to function as part of a SLAM system. New forms of loop-closure in SLAM may become possible by recognising different parts of the same hierarchical object.

Acknowledgements This work was supported by EU Framework projects BIOTACT (ICT-215910) and ICEA (IST-027819).

References

1. Aarts, E., Korst, J.: Simulated Annealing and Boltzmann Machines. Wiley (1988)
2. Ahl, A.: The role of vibrissae in behavior: a status review. Veterinary Research Communications 10(1), 245–268 (1986)
3. Binford, T., Levitt, T.: Evidential reasoning for object recognition. IEEE Tranactions on Pattern Analysis and Machine Intelligence (2003)
4. Carvell, G., Simons, D.: Biometric analyses of vibrissal tactile discrimination in the rat. J. Neurosci. 10(8), 2638 (1990)
5. Erman, L., Hayes-Roth, F., Lesser, V., Reddy, R.: The Hearsay-II system. ACM Computing Surveys 12(2) (1980)
6. Evans, M., Fox, C., Pearson, M., Prescott, T.: Spectral Template Based Classification of Robotic Whisker Sensor Signals. Proc. TAROS2009 (2009)
7. Evans, M., Fox, C., Prescott, T.: Tactile discrimination using template classifiers. From Animals to Animats SAB2010. (2010)
8. Fend, M.: Whisker-based texture discrimination on a mobile robot. Advances in Artificial Life pp. 302–311 (2005)
9. Fox, C., Pearson, M., Mitchinson, B., Pipe, T., Prescott, T.: Simple features for texture classification. Somatosensory and Motor Research 24(3), 139–162 (2007)
10. Fox, C.: ThomCat: A Bayesian blackboard model of hierarchical temporal perception. In: Proc. FLAIRS (2008)
11. Gallagher, G., Srinivasa, S.S., Bagnell, J.A., Ferguson., D.: Gatmo: A generalized approach to tracking movable objects. In: ICRA (2009)
12. Green, P.: Reversible jump Markov chain Monte Carlo computation. Biometrika 82(4), 711–732 (1995)
13. Heiligenberg, W.: Neural nets in electric fish. MIT (1991)
14. Kaneko, M., Kanayama, N., Tsuji, T.: Active antenna for contact sensing. IEEE Transactions on robotics and automation 14(2), 278–291 (1998)
15. Kim, D., Moller, R.: Biomimetic whiskers for shape recognition. Robotics and Autonomous Systems 55(3), 229–243 (2007)
16. Laskey, K.B., da Costa, P.C.: Of starships and klingons: Bayesian inference for the 23rd century. In: Proc. UAI (2005)

17. Lepora, N., Evans, M., Fox, C., Diamond, M., Gurney, K., Prescott, T.: Naive Bayes texture classification applied to whisker data from a moving robot. Proc. IEEE WCCI (2010)
18. Milch, B.: Probabilisitic Models with Unknown Objects. Ph.D. thesis, UC Berkeley (2006)
19. Mitchell, M.: Analogy-Making as Perception. MIT (1993)
20. Pearl, J.: Causality. Cambridge University Press (2000)
21. Pearson, M.J., Mitchinson, B., Welsby, J., Pipe, A.G., Prescott, T.J.: Scratchbot: Active tactile sensing in a whiskered mobile robot. From Animals to Animats SAB2010. (2010)
22. Petrovskaya, A., Khatib, O., Thrun, S., Ng, A.Y.: Touch based perception for object manipulation. In: ICRA (2007)
23. Schultz, A., Solomon, J., Peshkin, M., Hartmann, M.: Multifunctional whisker arrays for distance detection, terrain mapping, and object feature extraction. Proc. ICRA2005 (2005)
24. Seth, A., McKinstry, J., Edelman, G., Krichmar, J.: Texture discrimination by an autonomous mobile brain-based device with whiskers. In: Proc. IEEE ICRA (2004)
25. Srinivasa, S., Ferguson, D., Helfrich, C., Berenson, D., Romea, A.C., Diankov, R., Gallagher, G., Hollinger, G., Kuffner, J., Vandeweghe, J.M.: Herb: a home exploring robotic butler. Autonomous Robots 28(1), 5–20 (January 2010)
26. Sutton, C., Burns, B., Morrison, C., Cohen, P.R.: Guided incremental construction of belief networks. In: Proc. Fifth Int. Symp. Intelligient Data Analysis (2003)
27. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT (2006)
28. Wang, C.C., Thorpe, C., Thrun, S., Herbert, M., Durrant-Whyte, H.: Simultaneous localization, mapping and moving object tracking. Int. J. Robotics Research 26, 889 (2007)

Appendix: Noisy-OR density combination

Let Y_i range over nodes $pa(X)$ in a continuous-valued Bayesian network with noisy-OR parent combinations,

$$P(X|\{Y_i\}_i) = 1 - \prod(1 - P_i). \tag{24}$$

with $P_i = P(X|Y_i)$. Consider the probability of a small range of hypotheses,

$$\delta^3 p(X|\{Y_i\}_i) = 1 - \prod(1 - \delta^3 p_i), \tag{25}$$

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

$$\delta^3 p(X|\{Y_i\}_i) = \delta^3 \sum p_i. \tag{26}$$

The δ^3 terms cancel to yield

$$p(X|\{Y_i\}_i) = \sum p_i. \tag{27}$$