Tactile SLAM with a biomimetic whiskered robot

Charles Fox, Mat Evans, Martin Pearson and Tony Prescott

***Abstract*— Future robots may need to navigate where visual sensors fail. Touch sensors provide an alternative modality, largely unexplored in the context of robotic map building. We present the first results in grid based simultaneous localisation and mapping (SLAM) with biomimetic whisker sensors, and show how multi-whisker features coupled with priors about straight edges in the world can boost its performance. Our results are from a simple, small environment but are intended as a first baseline to measure future algorithms against.**

I. Introduction

Tactile sensing allows rodents to excel in environments where many other sensory modalities are impaired. Rodents often operate in the dark or in complex underground tunnels, many are diurnal or nocturnal, and consequently cannot rely on their eyes to navigate, hunt and explore. Instead they are excellent at tactile sensing and use their whiskers (known anatomically as 'vibrissae') to gather information about the world. Unlike distal sensors such as lasers and vision, whiskers make direct contact with the world around the location of the agent only. There are environments where distal sensors are inappropriate in robotics, for example smoke-filled search and rescue operations, or adversarial environments where covert (emission-less) sensing is required. Biomimetic touch sensors such as whiskers have been proposed as useful in such environments [14].

The task of determining the nature of a surface a whisker has made contact with is a difficult one. Both the location in space, and the identity of an object must be determined while taking the agent's location uncertainty into account. A number of systems have been constructed for whisker-based tactile discrimination: using the geometry of whisker-object contact [22]; measurement of bending moments at the whisker base [18], [4]; or through the extraction of features from the whisker deflection signal for texture [15],[12] and radial distance estimation [7]. Integrating sensory systems onto an autonomous mobile robot presents additional challenges. Though a number of whiskered mobile robots have been built, [21] few if any perform mapping and navigation, beyond simple reactive behaviours such as wall-following and obstacle avoidance.

On first consideration, it may appear that touch-based SLAM is doomed to failure due to the sparsity of the likelihood functions. A previous study [10] gave examples of whisker-based likelihood functions, and showed them to have high spatial variances as well as many discontinuities and non-Gaussian shapes. The high spatial variances occur because many locations in the world look the same to touch sensors. In related – but different – modalities, short range infrared [2], [1] and sonar sensors [27] have enabled SLAM through multiscans, landmark constraints and inferences over pose histories; and infrared sensors with the (very) strong prior assumption that the world is made of rectilinear structures only [26]. (Rectilinear assumptions have also been made for covering tasks [5].)

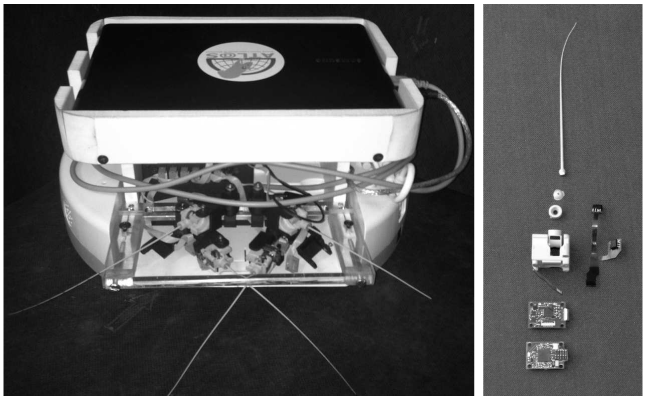

(a) (b)

Fig. 1. (a) The robot platform. (b) A single whisker unit. See Section II-A for a description of the components

This paper presents an alternative approach to strengthen mapping, using properties of whisker sensors without multiscan or rectilinear assumptions. Unlike infra-red or sonar sensors, which collect only *distance* information from a nearby *point*, whiskers can recover information about the local *orientation* of the surface from contacts [9]. We present three methods for whiskered tactile SLAM. First, using contact locations only; second, using geometric contact information from groups of whiskers to recover surface orientation (these two methods would also be applicable to the other types of short range sensors mentioned above) and third, using whisker-specific strain time series to recover surface orientation. In all methods we also exploit prior knowledge about the world structure: all three methods assume that neighbouring points are correlated; the two surface orientation methods assume that the environment is made mostly of straight edges, but of any orientation, unlike the rectilinear assumption.

II. Methods

A. Whiskers.

Experiments were performed using four artificial whiskers of 140mm in length, 1.45mm diameter at the base tapering

Supported by FP7 BIOTACT (ICT-215910) and EFAA (ICT-270490).

Fox, Evans and Prescott are with the Sheffield Center for Robotics, UK. Pearson is with Bristol Robotics Laboratory, UK.

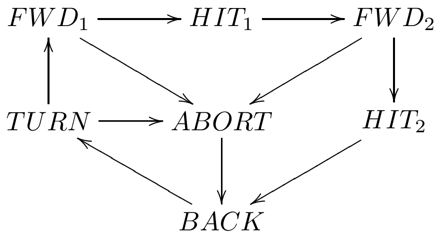
linearly to 0.3mm at the tip, and designed to mimic properties of rodent whiskers, at a scale of ≈5:1. They are built from nanocure25 using an Evisiontec rapid prototyping machine. Like rat whiskers, sensing takes place only at the base (or ‘follicle’), and measures the local strain there [6]. Magnets are bonded to the bases of the whiskers and held in place by plugs of polyurethane approximately 0.75 mm above Melexix 90333 tri-axis Hall effect sensor ICs [19]. These sensors each generate two outputs representing the magnetic field direction (in two axes) with respect to its calibrated resting angle. These two 16-bit values are sampled by a local dsPIC33f802 micro-controller which 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 a software driver and API (Cesys GmbH), users can request horizontal and vertical strain data from all whiskers at a sample rate of 2kHz.

B. Robot platform.

Four whiskers are mounted in the cargo bay of an iRobot Create base (`www.irobot.com`), positioned on a rapid prototyped ball joint mountings which allow adjustment of the whiskers. We have also extended the cargo bay mounting to accommodate a netbook PC, which is used for local control of the robot and 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`) providing high-level, networked API interfacing to the Create’s serial port commands. Low-level processes such as texture and shape recognition and basic motor control can run on the netbook, reading the raw data from the circular buffer. These processes send their results to a desktop machine which handles mapping. Communication is via the C++ Thrift RPC protocol. Differential and absolute odometry data from the Create is also sent to the mapping server. Preliminary experiments showed that the odometry of the Create, once loaded with the sensing and control hardware, is accurate to $< 5\%$ of any straight line or turn on the spot movements. It was useful to cache commonly used trigonometric quantities describing the whisker geometry to enable fast lookup during navigation.

C. Environment and behaviour

The robot is placed in the 1.25m × 1.25m square arena shown in Fig. 2, containing several square objects. Its movements are controlled by a finite state machine (FSM), which moves the robot forward in a straight line (FWD_1) *repeatedly* until its whiskers hit something ($HIT1$). It then moves the robot forward again at a slower speed (FWD_2) until either a second whisker makes contact (HIT_2), or the strain in any whisker exceeds a safety threshold ($ABORT$). The robot then reverses ($BACK$) and turns on the spot ($TURN$). Turns angles are draw from a mixture of two Gaussians, one of which has a small mean and variance (0.14π, 0) to encourage wall following and the other a large mean and variance (0.3π,0.25π) to encourage movement away from walls to explore other parts of the arena. Each FWD_1, FWD_2 and $BACK$ step of the FSM lasts for 0.5s, the FWD_1 and $BACK$ state moves at 0.05m/s and the FWD_2 state moves at 0.02m/s for safety. Turning is at 0.3rad/s. Within each 0.5s FSM step, the whisker strains are monitored regularly for strains exceeding the safety threshold – if this occurs, the motion is terminated early and the FSM switches to the $ABORT$ state then $BACK$ to escape. Under these behaviours, the robot tends to move anticlockwise overall around the arena, interspersed with periods of wall following and exploration, and typically makes around 3 or 4 circuits (and hopefully loop closures) in a 6 minute run.



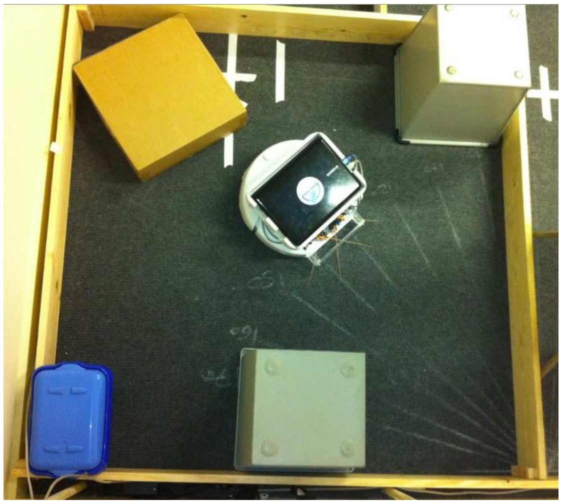


Fig. 2. Arena used in map building experiments

Combined odometry and *radial distance* (i.e. the distance along each whisker shaft to any contact, or lack of any contact) reports are sent after every FSM step with the exception of $BACK$ states, which revisit recently visited poses and would double-count recent observations there if their likelihoods were fused into localisation and mapping.

D. Localisation

Localisation is performed using the standard [25] particle filter of algorithm 1. 100 particles $\{s^i\}_{i=1:100}$ are maintained, each of which carries a continuous-valued pose (2D location and orientation) and a grid map, $m[x, y]$, of the environment. We use a 50 × 50 grid map covering a 2.5m×2.5m space (double the dimensions of the arena to allow for overspill; 50mm cells). Updates occur at each FSM step.

Algorithm 1 Sequential importance particle filter

```
for each time step $t$ do
    for $s = 1 : N$ do
        sample $s^I[t] \sim P(s^i|s^i[t-1], \delta_{trans}, \delta_{rot})$
    end for
    for $s = 1 : N$ do
        $\lambda^i[t] \leftarrow \frac{1}{Z} P(o[t]|s^i[t])$
    end for
    resample $s^i[t] \leftarrow s^j[t-1]$ with $P(j) = \lambda^j[t]$
end for
```

The observation vector o consists of radial distance estimates $\{r_i\}_{i=1:4}$ from the four whiskers, along with translational (δ_{trans}) and rotational (δ_{rot}) odometry estimates from the Create platform (and wall surface angle estimates ϕ, not used for localisation). The resampling step draws samples according to the likelihood (importance) weights of the previous set.

Radial distances from whisker sensors may be estimated using methods such as inverse beam theory [4], [18] and feature extraction [7] which could later be plugged in here – but in the present localisation (not mapping) module we simply set a strain threshold and declare a contact at a standard, fixed radius (130mm) when that threshold is exceeded (and the FSM moves to HIT_1 at this point).

Each particle's grid map contains occupancy probabilities which are used to compute the likelihood function λ as the product of the individual whisker likelihoods,

$$\lambda = \prod_{w=1}^{4} \lambda^w. \tag{1}$$

The likelihood of the ith whisker is a function of its radial contact distance, r_i, and the particle pose and particle map m. We discretise the whisker shaft into 5 segments, and assume the segments are represented by equally spaced points, $G \in 1:5$ along the length of the shaft. The observed radial distance r is discretised to the nearest segment number R. The location L of each segment point in the grid map is found, $[x, y] = L(G)$. The whisker likelihood is then given by the product of the map's probability of contact at the reported location, $m(L(R))$ and all the probabilities of no contacts at the segments between the base of the shaft and the observed contact,

$$\lambda^w = m(L(R)) \prod_{G=0}^{R} (1 - m(L(G))). \tag{2}$$

Software speed is important in particle filtering, so we preallocate memory for two populations of particles, then copy and update values between them in the manner of double-buffering, to avoid computational overheads of construction and destruction of objects. The two population buffers are held in shared memory, which allows monitors such as GUI displays to run as separate processes on the dual-core desktop SLAM machine with minimal communications overhead.

E. Blob-based mapping

The simplest whiskered mapping method would treat each contact as an observation of a single cellof the grid map at the contact location, and assume independence between cells. Preliminary experiments showed this is impractical, as there are many cells and only a small number of contacts (e.g. 30) during a run (of 6 minutes). A simple extension of this idea is to assume a local correlation between cells, as in a Markov Random Field. Under this assumption, a single contact observation gives rise to a small local Gaussian Δm likelihood to be fused into the grid map m,

$$\Delta m[x, y] = \Delta[x_c, y_c] \exp\{-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\}, \tag{3}$$

where (x_c, y_c) are the coordinates of the contact cell, σ is set to make the resulting blob affect a radius of about two pixels, and $\Delta m[x_c, y_c]$ is the likelihood of the original contact cell occupancy given the current particle and observation (set to a constant > 0.5).

Whiskers that do not make contact also carry likelihoods that cells along their lengths are empty. We approximate this by a single Gaussian as in eqn. 3, but with $\Delta m[x_c, y_c] < 0.5$, and (x_x, y_c) in the center of the whisker shaft. Furthermore, we know that the region occupied by the robot's current body position cannot be occupied by another object, so we can also fuse a similar negative evidence Gaussian centred on the robot body location.

F. Angle-based maps with multi-whisker contact geometry

A more sophisticated mapping strategy is to exploit prior knowledge about the structure of the world, coupled with using features from multiple whiskers together. Previous work [26] made the strong assumption that all objects in the environment have straight edges aligned along Cartesian axes, exploiting the fact that many man-made environments are based on square grids. Strong hierarchical object priors were used by [13] to constrain the interpretation of contacts as known 3D object forms. Here we use a prior whose strength lies somewhere between these strong prior approaches and the weak prior blob method of sec. II-E. We assume that the environment is made up mostly of long, straight edges, but do not impose a Cartesian grid on their poses or make assumptions about the 3D forms of objects. So rather than placing Gaussian blobs at contact points, we place a blur of long, oriented edges.

We and others have previously investigated the recovery of surface normal information from *individual* whisker strains [18],[9]. A new simple approach for *multi*-whiskered robots is to locate two contact points on the same surface with two different whiskers, then compute the angle between them. Assuming that edges in the world are locally straight at this scale, then this angle gives the angle of the surface. The two contacts can be read during the FSM states HIT_1 and HIT_2 as described in section II-C. (In some cases the FWD_2 state terminates to $ABORT$ without a second contact due to a strain safety threshold being exceeded. In these cases, we revert to mapping a single Gaussian blob at the first contact point only as in sec. II-E.)

When an oriented surface is found in this way, we fuse a blur of long oriented edges into the map,

$$\Delta m[x,y] = \Delta[x_c,y_c]\exp\{-\frac{R^2}{2\sigma_R^2} - frac(\theta-\theta_c)^2 2\sigma_\theta^2\}, \tag{4}$$

where (R,θ) are radial coordinates centered on the contact midpoint (x_c,y_c) and estimated surface angle θ_c. We use $\sigma_R = 0.25$ and $\sigma_\theta = \pi/12$, values which were tuned by hand to produce a *long* (0.25m) blurred edge in the map around the contact point, giving the best maps for this environment.

G. Angle-based mapping with multi-whisker templates

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 [11], and stationary robot hardware [8]. In the present study we have access to four whiskers together, so we can train templates corresponding to contact *angle* classes from the 8-dimensional time series from the whole *multi-whisker* set (four whiskers, each with vertical and horizontal strain channels). The rationale for this approach is that the geometric multi-whisker method of sec. II-F must assume that the estimated contact locations are accurate – which is not necessarily true – and is restricted to utilising data from two contact whisker locations only. In contrast, a template method can utilities bulk data from all whiskers to find similar surface normals, and without any geometric assumptions. It is a purely data-driven method. Oriented edges as in eqn. 4 may again be added to the map once surface normals are found using templates.

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 (at HIT_1 occurrence), 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 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 online SLAM, strain time-series data was logged from immediately after each HIT_1 to the following HIT_2 then sent to the classifier at HIT_2. The average squared error, e for each template, T_i is computed over the N logged data points,

$$e(T_i) = \frac{1}{N}\sum_{t=1}^{n}(I(t) - T_i(t))^2. \tag{5}$$

The template with the lowest sum of squared errors was determined the winner, and its surface normal used in eqn. 4 to fuse an oriented long edge into the map.

III. Results

Results are presented for mapping using the three methods: blobs, geometric multi-whisker and template multi-whisker. In each instance the robot was programmed to run for twenty trials of six minutes (6 hours' data in total). The resulting grid maps are compared to a ground truth grid map (Fig. 3; as built by a human observer – which is then smoothed with a 5 × 5 cell Gaussian filter, standard deviation 2.5). Occupancy in the grid map is represented by a 1 (object present) or a 0 (object not present). Unexplored areas are marked with 0.3 as this is approximately the mean occupancy of the arena. Grid maps, $m[x,y]$, are compared by an element-wise sum of absolute errors calculation to the ground truth map, $gt[x,y]$,

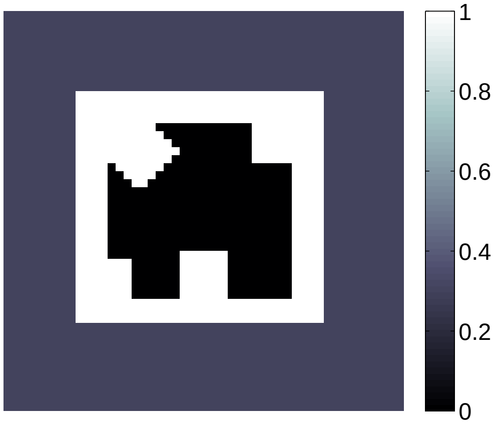


Fig. 3. Ground truth grid map based on the arena used for comparison with generated maps. Brightness indicates occupancy. (cf. Fig. 2.)

$$\frac{1}{N}\sum_{t=1}^{n}\frac{1}{50}\sum_{x=1}^{50}\frac{1}{50}\sum_{y=1}^{50}|m[x,y] - gt[x,y]|. \tag{6}$$

The mean error per cell is reported for each map. For baseline comparison, error for a random map populated from a Bernoulli distribution ($p = 0.5$) is 0.47 (unitless difference of probability).

Fig. 4(a) shows the average map generated from running the robot with the blob based mapping system. Mean occupancy error was 0.40. Fig. 4(b) shows the average map generated from running with the geometric multi-whisker based mapping system. Mean occupancy error was 0.39. Fig. 4 shows the average map generated from running with the template based mapping system. Mean occupancy error was 0.37.

So get some idea of failure cases, we also tested a larger 2.5m square arena in simulation only, using Player/Stage and the geometric angle method (physical strains for templates being unavailable in simulation), which allowed us to explore different odometry noise levels. For 2% noise in δ_{trans} and δ_{rot}, the inferred map and ground truth arena are shown in Fig. 6. Correct location tracking was maintained during the 200 steps used to build this map. We then ran a simulation with 5% odometry noise, but loop closures failed in this case.

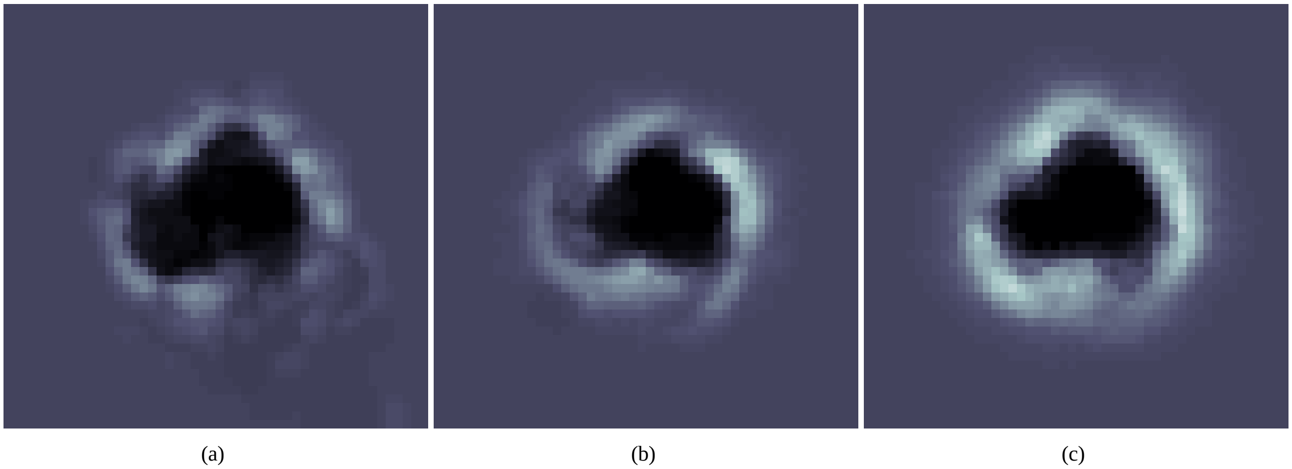


Fig. 4. (a) Average grid map generated when using blob based mapping over twenty trials. Note grid occupancy outside of the area of the arena and low mean grid occupancy (dark occupied regions). (b) Average grid map generated when using multi-whisker angle based mapping over twenty trials. Note grid occupancy restricted to the area of the arena and high mean grid occupancy (brighter occupied regions). (c) Average grid map generated when using template based mapping over twenty trials. Note grid occupancy restricted to the area of the arena and high mean grid occupancy (brightest occupied regions). Brightness indicates occupancy, all maps are drawn on the same occupancy scale (0:1)

IV. Discussion

All three mapping methods perform well. The template method is the best and blob-based is the worst under the metric of eqn. 6. This seems to be because errors in localisation meant the robot became lost and occupied the map with objects outside of the area of the arena (as can be seen when comparing the lower right quadrants of Fig.3 and Fig. 4(a). Geometric multi-whisker angle mapping was an improvement on blob based mapping, with grid occupation being restricted to the area of the arena, and surface contours are recovered partially (white patches in Fig.4(b)).

The best performance comes from the template method. Mapping is restricted to the area of the arena and large sections of surface contours are recovered (prominent white patches in Fig.4(c)). Templates are especially useful as they provide strong predictions of surface angle even when only single whisker contacts are made – unlike the geometric method. Templates therefore can extract more information from impoverished whisker data, informing stronger predictions of object contours, leading to a greater occupancy in the grid maps.

Differences between the mapping performance of the three methods can be seen more clearly in the maps generated on individual trials. Fig.5 shows typical maps generated in each of the three conditions. Blob based mapping (Fig.5 (a) and (b)) results in sparse object location reports, and unreliable localisation resulting in mapping outside of the area of the arena. Geometric multi-whisker based mapping (Fig.5 (c) and (d)) generates predictions of object contours, and these improve localisation to restrict mapping to the bounds of the arena. Template based mapping (Fig.5 (e) and (f)) generates more, better predictions of object contours (white areas in the grid maps), improving localisation. Object features such as sharp corners can also be seen in the template based grid maps (lower region of Fig.5 (a) and (b)).

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 has been done in the field of haptic touch [24],[23] and is commonly used in vision [17], and audition [3]. It has been proposed that biological cells in the thalamus and cortex of the rat are encoding features [20],[16] in this way. In our own lab we are developing features for whisker based tactile sensing of contact geometry [7] and texture of different materials [12]. In future we hope to be able to combine such features for diverse tactile properties in rich environments into a coherent system onboard a mobile robot.

V. Conclusion

We previously [10] showed the Create platform performing basic localisation from whiskers and a given map. This study has addressed the other aspect of SLAM – mapping – showing it is feasible to map *small* arenas using whisker sensors by exploiting timing information between contacts and prior knowledge about edges in the environment.

The results presented here are from a simple small arena only, and are of course not competitive with current SLAM systems using visual or laser sensors, which can run in large outdoor environments. However we have presented the first results of whiskered grid mapping, which may serve as a baseline for improved whiskered SLAM systems. For example, [13] presents more complex whiskered object recognition using even stronger priors, which could in future form part of a landmark-SLAM method for larger environments, and it would be useful to compare its results – and

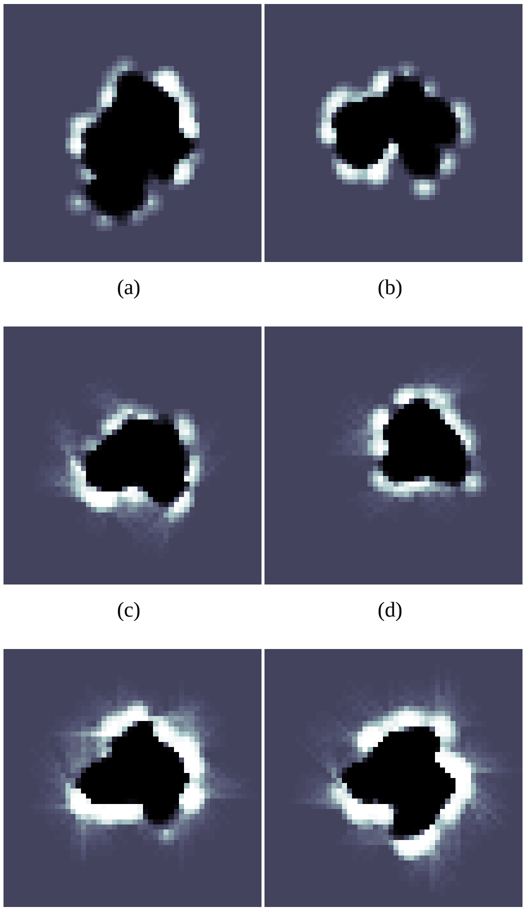


Fig. 5. Grid maps generated on individual trials for blog based mapping ((a) and (b)), multi whisker geometry based mapping ((d) and (e)), and template based mapping ((e) and (f)). Brightness indicates occupancy.

other algorithms such as multiscans and landmark constraints – with the grid-SLAM baseline presented here.

References

[1] K. R. Beevers and W. H. Huang. Inferring and enforcing relative constraints in slam. In *WAFR*, 2006.

[2] K.R. Beevers and W.H. Huang. SLAM with sparse sensing. In *ICRA*, pages 2285–2290. IEEE, 2006.

[3] J. B. Bello and M. Sandler. Techniques for automated music transcription. *ISMIR*, 2000.

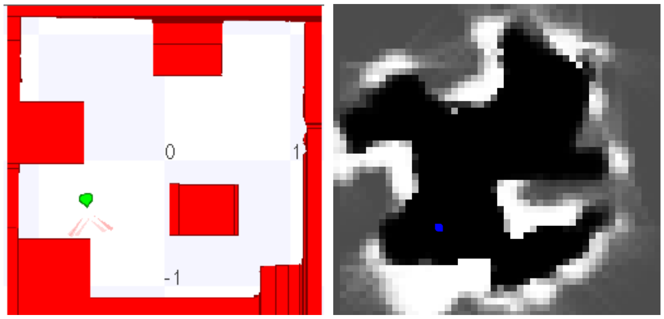
[4] J.A. Birdwell, J.H. Solomon, M. Thajchayapong, M.A. Taylor, M. Cheely, R.B. Towal, J. Conradt, and M.J.Z. Hartmann. Biomechanical models for radial distance determination by the rat vibrissal system. *J Neurophysiol*, 98(4):2439–2455, 2007.

[5] Z. J. Butler, A. A. Rizzi, and R. L. Hollis. Contact sensor-based coverage of rectilinear environments. In *Proc. of IEEE Int'l Symposium on Intelligent Control*, 1999.

[6] Mathew E Diamond, Moritz von Heimendahl, Per Magne Knutsen, David Kleinfeld, and Ehud Ahissar. 'where' and 'what' in the whisker sensorimotor system. *Nat Rev Neurosci*, 9(8):601–612, 2008 Aug.

[7] M. H.. Evans, C. W. Fox, N.F. Lepora, M. J. Pearson, and T. J. Prescott. Whisker-object contact speed affects radial distance estimation. In *Proceedings IEEE ROBIO*, 2010.

[8] M. H. Evans, C. W. Fox, M. J. Pearson, and T. J Prescott. Tactile discrimination using template classifiers: Towards a model of feature extraction in mammalian vibrissal systems. In *SAB*, 2010.



(a) (b)

Fig. 6. (a) Stage simulation of a larger arena, running geometric multi-whisker mapping. (b) Grid map generated in Stage simulation, with 2% odometry noise.

[9] M. H. Evans, C.W. Fox, M. J. Pearson, and T. J. Prescott. Object location, orientation, and velocity extraction from artificial vibrissal signals. In *SFN Abs. 174.8/Z12*, 2009.

[10] C. W. Fox, M. H. Evans, N. F. Lepora, M. J. Pearson, A. Ham, and T. J. Prescott. A mobile whiskered robot platform. In *TAROS*, 2011.

[11] C. W. Fox, M. H. Evans, M. J. Pearson, and T. J. Prescott. Temporal inference for shape recognition from whiskers. In *TAROS*, 2008.

[12] C. W. Fox, B. Mitchinson, M. J. Pearson, A. G. Pipe, and T. J. Prescott. Contact type dependency of texture classification in a whiskered mobile robot. *Autonomous Robots*, 2009.

[13] C. W. Fox and T. J. Prescott. Mapping with sparse local sensors and strong hierarchical priors. In *TAROS*, 2011.

[14] V.V. Hafner. Cognitive maps in rats and robots. *Adaptive Behavior*, 13(2):87, 2005.

[15] J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M. E. Diamond, and P. Konig. Texture signals in whisker vibrations. *J Neurophysiol*, 95(3):1792–1799, 2006.

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

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

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

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

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

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

[22] R.A. Russell and J.A. Wijaya. Object location and recognition using whisker sensors. In *ACRA*, 2003.

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

[24] 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.

[25] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT, 2006.

[26] Y. Zhang, J. Liu, G. Hoffmann, M. Quilling, K. Payne, P. Bose, and A. Zimdars. Real-time indoor mapping for mobile robots with limited sensing. In *IEEE MASS*, 2010.

[27] G. Zunino and H.I. Christensen. Navigation in realistic environments. In *9th Int. Symp. on Intelligent Robotic Systems, SIRS*. Citeseer, 2001.