

# Learning peripersonal space representation through artificial skin for avoidance and reaching with whole body surface

Alessandro Roncone, Matej Hoffmann, Ugo Pattacini, and Giorgio Metta

**Abstract**—With robots leaving factory environments and entering less controlled domains, possibly sharing living space with humans, safety needs to be guaranteed. To this end, some form of awareness of their body surface and the space surrounding it is desirable. In this work, we present a unique method that lets a robot learn a distributed representation of space around its body (or peripersonal space) by exploiting a whole-body artificial skin and through physical contact with the environment. Every taxel (tactile element) has a visual receptive field anchored to it. Starting from an initially blank state, the distance of every object entering this receptive field is visually perceived and recorded, together with information whether the object has eventually contacted the particular skin area or not. This gives rise to a set of probabilities that are updated incrementally and that carry information about the likelihood of particular events in the environment contacting a particular set of taxels. The learned representation naturally serves the purpose of predicting contacts with the whole body of the robot, which is of clear behavioral relevance. Furthermore, we devised a simple avoidance controller that is triggered by this representation, thus endowing a robot with a “margin of safety” around its body. Finally, simply reversing the sign in the controller we used gives rise to simple “reaching” for objects in the robot’s vicinity, which automatically proceeds with the most activated (closest) body part.

## I. INTRODUCTION

Pushed by societal needs and economic opportunities, robots are leaving controlled factory environments and enter domains that are far less structured, possibly even sharing living space with humans. As a consequence, they need to dynamically adapt to unpredictable interactions and guarantee their own as well as others’ safety at every moment. However, robotic technologies used in industry typically rely on preprogrammed models and “blindly” executed end-effector trajectories. The rest of the body is typically represented as a kinematic chain; the volume and surface of the body itself being “numb” and rarely taken into account. Technologies that endow robots with whole-body tactile sensing—or artificial skin—open up new possibilities to address the above-mentioned shortcomings [?], [?], [?]. There are several directions in which tactile arrays covering extensive parts of robot bodies can be exploited. One major area deals with detection and appropriate handling of physical contacts with

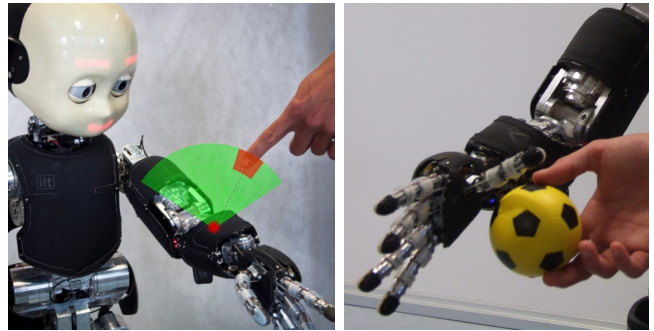


Fig. 1: *Illustration of the setup. (left) Receptive field above one of the left forearm taxels. (right) An object approaching the right palm.*

the environment or humans (e.g., [?]). Traditionally, the interaction forces were controlled using a variety of techniques (such as impedance/admittance control, hybrid position-force control, or parallel control), relying primarily on force/torque measurements and often assuming that contact was occurring at the end-effector. If this assumption is not valid and in the presence of multiple contacts, localization of the contact points becomes indispensable for correct response. Artificial skin can supply the necessary information [?], [?], [?].

Another area where sensing on the body surface can be utilised is to allow the robot to autonomously learn about the properties of its body, such as its spatial extent. Hoffmann et al. [?] provide a survey of the literature that deals with “learning a body schema” and note that the majority of the studies deal with “visuo-proprioceptive calibration”. Artificial skin provides another valuable sensory modality that can complement vision and proprioception. Alternatively, it can even replace vision on the self-calibration task by relying on closing the loop through self-touch configurations [?].

The work presented here in a sense combines the two domains described above and aims at learning not only a representation of the extent of the body itself, but also about the space immediately surrounding it—*peripersonal space*. This can then enhance the robot’s interactions with the environment, allowing it to anticipate contacts (see Fig. 1). We will combine visual, proprioceptive (joint encoder), and, importantly, tactile information to achieve this.

Peripersonal space is also of special relevance for every animal. In this space, objects can be reached for, grasped, and, at the same time, they may pose threats and may evoke appropriate avoidance response. In particular, we want to mimic the visual body-part-centered receptive fields (RFs) of neurons that were observed in monkeys [?]. Robotic models in this direction were developed by Asada and

This work was supported by the European Union Seventh Framework Programme, projects WYSIWYD (FP7-ICT-612139) and Xperience (FP7-ICT-270273). M.H. was supported by a Marie Curie Intra-European Fellowship (iCub Body Schema 625727).

A. Roncone, M. Hoffmann, U. Pattacini, and G. Metta are with iCub Facility, Istituto Italiano di Tecnologia, Via Morego 30, 16123 Genova, Italy. {alessandro.roncone, matej.hoffmann, ugo.pattacini, giorgio.metta}@iit.it



Fig. 2: *Pressure-sensitive skin of the iCub.* (left) iCub forearm with exposed skin patches. (right) Four triangular modules with 10 taxels each.

colleagues, employing biologically motivated learning architectures (self-organizing maps, hebbian learning, attention modules) [?], [?]. Compared to these studies, the architecture presented here is less motivated by the putative brain circuitry, but builds on top of existing engineering solutions and targets practical functionality in the real robot and full 3D space around it.

One of the important behaviors are defensive responses. The respective neurons in primate brains fire as soon as a potentially harmful object enters their RFs, which “grow” out of individual body parts. This gives rise to a “margin of safety” around the body, such as the flight zone of grazing animals or the multimodal attentional space that surrounds the skin in humans [?]. Analogous behavior is desirable in robots. In this work, we propose an architecture that achieves this functionality thanks to a visual RF anchored to each taxel (tactile element) on the robot skin. Starting from an initially blank state, the distance of every object entering this RF is recorded with respect to the taxel’s frame of reference (FoR), together with information whether the object has eventually contacted the particular skin area or not. This gives rise to a set of probabilities that are updated incrementally and that carry information about the likelihood of particular events in the environment physically contacting a particular set of taxels. In order to achieve the desired coordinate transformations to convert visual inputs to the respective taxel FoR, existing kinematic representations of the robot, including head and eye, are used. However, the representation regarding the likelihood of contact is learned on top of this and based on actual physical contact with the skin, thus automatically incorporating / compensating for any inaccuracies that the existing kinematic mappings contain.

We demonstrate the utility of this representation in showing avoidance responses to objects in the environment as they come close to the skin and trigger activation in corresponding RFs around the “endangered” taxels. Note that this differs from self-protective behaviors that occur only after contact (e.g. [?] in the iCub). Crucially, the avoidance is triggered in an anticipatory fashion prior to contact. A similar approach was used in [?], but relying on local proximity sensors embedded in a multimodal skin. In this work, it is to our knowledge for the first time that a “margin of safety” connecting visual, proprioceptive, and tactile information is realized in

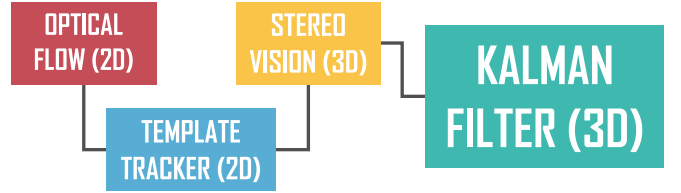


Fig. 3: *Object tracking schematics.* See text for details.

a robot. Finally, a simple reversal of the direction of the desired movement vector gives rise to simple approaching or reaching behaviors—the body parts most strongly activated by an object in the vicinity are automatically pulled towards it.

This article is structured as follows. In Section II, we describe the real robot and its relevant software modules, introduce the proposed representation, the data collection procedure, and specifications of a simulation environment. Results will be presented in Section III, followed by a discussion and conclusion section.

## II. MATERIAL AND METHODS

### A. iCub humanoid robot and key modules

The iCub is an open-source platform for research in cognitive robotics [?]. In the following, we describe the key components relevant for this work.

1) *Artificial skin:* The iCub was recently equipped with an artificial pressure-sensitive skin covering most body parts [?]. In the experiments performed in this work, we restrict ourselves to the forearms and palms. The skin covering body parts consists of patches with triangular modules of 10 taxels each (Fig. 2 right). There are in total 23 modules on the forearm in two patches and hence 230 taxels (Fig. 2 left). However, for the purposes of this study—spatial RFs around body parts—it would be an unnecessarily high resolution to consider every taxel independently. Therefore, in what follows, every triangular module acts as single virtual taxel. That is, the taxel in the center of the module acts as a representative of the whole module and an activation of any of the module’s taxels is represented as a signal coming from this virtual (representative) taxel.

A spatial calibration of the skin of the forearm with respect to the iCub kinematic model has been performed in [?]. Using data from the CAD model, we have added calibration of the palm. Therefore, the poses of all taxels (position and orientation) as well as the virtual taxels in local reference frames of the robot are known.

2) *Joint angle sensing:* Proprioceptive inputs in the iCub simply consist in angular position measurements in every joint. For most joints, they are provided by absolute 12bit angular encoders.

3) *Head and eyes:* Vision of the iCub is provided by two cameras mounted in the robot’s eyes. The head of the robot has 3 degrees of freedom (DOFs) at the neck and 3 in the eyes allowing for tracking and vergence behaviors. The movement of the eyes is coupled, following an anthropomorphic arrangement. With appropriate calibration, depth information can be extracted from binocular disparity.

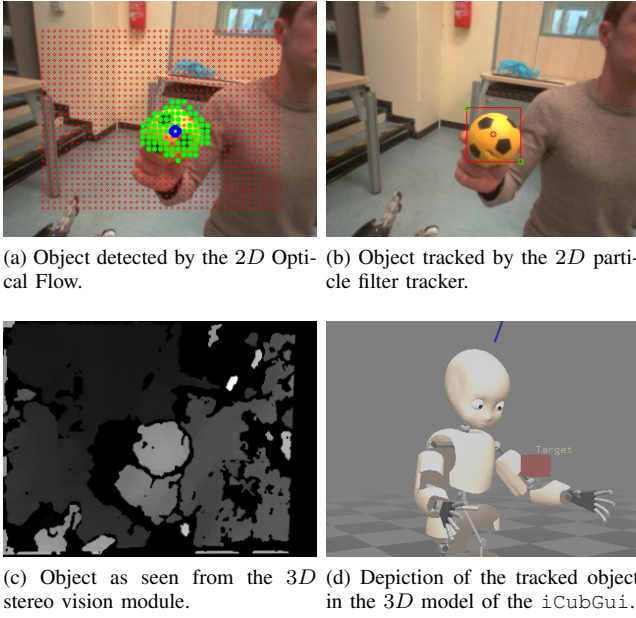


Fig. 4: *Object tracking pipeline*. Each figure shows the output of one of the four modules involved.

4) *Visual processing and gaze control*: To track external objects in 3D space as they approached the robot’s body, a module featuring a visual processing pipeline was necessary: moving objects need to be detected, segmented out of the background, their position retrieved and followed by the robot’s gaze. We developed a pipeline that allowed us to track general objects under some assumptions on the availability of visual features and limits on their velocity and size. The software architecture is composed of several interconnected modules, schematically depicted in Fig. 3, whereas the outcome of each step is presented in Fig. 4. The first module uses 2D Optical Flow [?] to detect motion in the image plane. If the computed motion is consistent with the presence of a moving object in the nearby space of the robot, it triggers a pipeline composed of three interconnected modules:

- a 2D particle filter [?] able to track the object in the image plane based on its color properties;
- a 3D stereo disparity module [?], able to convert the 2D planar information related to the incoming event (the centroid of the object and an estimation of its size) into 3D coordinates;
- a Kalman filter that receives 3D coordinates from the stereo vision module and improves the robustness of the estimation. It employs a fourth order dynamic model of the object motion, and estimates both the 3D position and the 3D velocity of the incoming object with respect to the robot’s Root FoR (located in the waist of the robot).

Finally, a gaze controller was employed in order for the head to smoothly follow the tracked object in space. The details of the gaze controller can be found in [?].

5) *Kinematic model and coordinate transformations*: The 3D position of the object in the Root FoR obtained as described in the previous section needs to be further transformed into the FoR of individual taxels. Learning these

transformations was not the goal of this work; therefore, we have employed the existing kinematic model of the iCub that is based on the Denavit-Hartenberg (DH) convention and embedded in a software library (*iKin* [?]). The final transformation from the last link of the DH chain to individual taxels comes from the skin calibration.

However, these composite transformations are subject to numerous errors that include (i) mismatch between the robot model based on the mechanical design specifications (CAD model) and the actual physical robot; (ii) inaccuracies in joint sensor calibration and measurements; (iii) unobserved variables from the sensed configuration coming from joint backlash or mechanical elasticity; (iv) inaccuracies in taxel pose calibration; (v) additional errors in visual perception coming from inaccurate camera calibration etc. As a whole, these errors can amount to several centimeters. For example, it can happen that oncoming objects will seemingly penetrate the robot’s skin—based on our model and measurements, not in reality—and will thus have negative distance w.r.t. the taxel normal. However, in the approach adopted here, the systematic component of the errors will be automatically compensated for by the representations that every taxel will learn regarding the space surrounding it.

## B. Representation of “Space Around the Body”

We have chosen a distributed representation in which every taxel is learning a collection of probabilities regarding the likelihood of objects from the environment contacting it.

1) *Data collection for learning*: A volume was chosen to demarcate a “visual receptive field” around every taxel. It is cone-shaped and grows out of every taxel along the normal to the local surface and extends to maximum 20cm away from the taxel (green region in Fig. 1 left and Fig. 5). External objects were then approaching individual skin parts (Fig. 1 right). Learning was realized in a local, distributed, event-driven manner. Once an object entered the RF, it marked the onset of a potentially interesting event. From this time on, the position of the object w.r.t the taxel was recorded such that the distance  $D$  could be computed:

$$D = \text{sgn}(\vec{d} \cdot \vec{z}) \|\vec{d}\| \quad (1)$$

where  $\vec{d}$  is the displacement vector pointing from the taxel to the event (center of the oncoming object) as measured by the visual processing pipeline described in Section II-A.4, and  $\vec{z}$  is the z-axis of the reference frame centered on the taxel and pointing outward (coincident with the normal to the skin surface at the taxel position) – see Fig. 5. The sign of their dot product is thus positive if the angle between them is lower than  $180^\circ$ , i.e. if the object is belonging to the positive hemisphere extending from the taxel. Hence, according to our definition, the distance,  $D$ , will preserve the information about the relationship of the event w.r.t. taxel normal. In this way, objects apparently “beneath” the skin surface will acquire negative distances, distinguishing them clearly from their counterparts in the positive hemisphere (green area in Fig. 5).



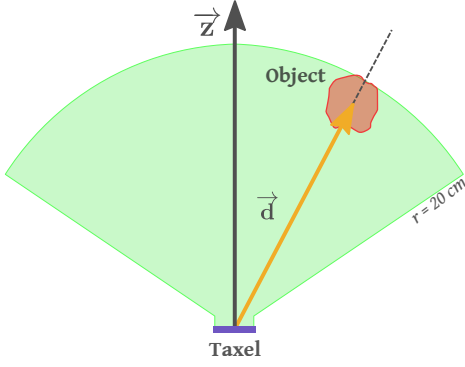


Fig. 5: *Receptive field of a taxel and approaching object (event).* See text for details.

This procedure—logging of  $D$  of approaching objects—proceeded in parallel for every taxel whose RF has been penetrated. Data was buffered for 3 seconds and it was used for learning only if the object eventually contacted the skin and was perceived by at least one taxel. In this case, a learning iteration was triggered that proceeded as follows: (i) For all the taxels that experienced contact, the buffer of object positions in their local FoR was traversed back in time with time steps of 50 ms. While the object was still in their respective RFs, the distance at every time step was recorded as positive examples in every taxel’s “memory”. (ii) For all the other taxels on the same body part, the procedure was analogous, but negative examples were appended to the respective memories.

2) *Internal representation:* We defined the range of  $D$  as  $[-10, 20]$ cm. This space was discretized into equally sized bins (different resolutions were tested, cf. Section III-A). Every taxel stored and continuously updated a record of the count of positive and negative examples it has encountered for every bin. The main advantage of this representation is its simplicity and ease of incremental updating. Most relevant for the agent is an estimation of the probability of an object hitting a particular part of the skin. Thus, for every oncoming object, its distance w.r.t. every taxel can be binned in the manner described above and a frequentist probability estimate obtained simply as:

$$P(D) \approx f(D) = \frac{n_{\text{positive}}(D_i)}{n_{\text{positive}}(D_i) + n_{\text{negative}}(D_i)} \quad (2)$$

where  $D_i$  is the bin in which the distance fell. Such an approach—discretized representation and querying—would constitute the simplest solution. However, it may give rise to unstable performance, in particular in the case when the state space is undersampled. Therefore, it is desirable to obtain a continuous function  $f$  that can be sampled at any real values of  $D$  and that is capable of smoothing out the discretized space<sup>1</sup>. This has been achieved by adapting

<sup>1</sup>It is worth noting that only the bins corresponding to discretized  $D$  have estimates of a probability function associated with them, each bin  $D_i$  independently from others. However,  $f(D)$  cannot be interpreted as a probability mass function (in discrete case) or probability density function (in continuous case), because the overall probability for the whole range of  $D$  can take any value and does not sum up to 1.

the Parzen-Window density estimation algorithm [?] to our situation—employing it as a data interpolation technique. In a 1-dimensional case, the interpolated value  $p(x)$  for any  $x$  is given by:

$$p(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^2} \Phi\left(\frac{x_i - x}{h}\right) \quad (3)$$

where  $x_i$  are the data points in the discrete input space,  $\Phi$  is the window function or kernel and  $h$  is the bandwidth parameter, which is responsible for weighting the contributions of the neighbors of the point  $x$ . We used a Gaussian function, hence we have:

$$p(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - x)^2}{2\sigma^2}\right) \quad (4)$$

For the standard deviation  $\sigma$ , different settings were explored, as will be presented in the Results Section. In summary, for any value of  $D = d$ , the final interpolated value,  $p(d)$ , represents an estimate of the probability of an object at distance  $d$  hitting the specific taxel under consideration.

#### C. Monte Carlo simulation of a single taxel model

In order to investigate the behavior of the representation proposed in Section II-B and to find suitable values for its parameters, a Monte Carlo simulation was carried out in the Matlab environment. To this end, a 3-dimensional model of a single taxel and its surroundings with oncoming simulated objects was set up. The model parameters were chosen to mimic the real robot setup as close as possible. The simulated taxel itself had a radius of 0.235 cm, which mimics the radius of the real iCub taxels. However, objects landing within 2 cm from the taxel’s center were still considered positive, resembling the size of a triangular module that is composed of 10 taxels—the basic building block of the iCub skin (see Fig. 2). These “virtual taxels” were used also in the real setup. In addition, the oncoming objects were also simulated. Since the nature of our data collection and learning method requires positive examples (objects contacting the virtual taxel) as well as negative examples (objects contacting neighbouring taxels), we simulated additionally 3 neighbouring virtual taxels. We implemented a stochastic “shower” of objects with their starting points uniformly distributed in a “starting zone” and their landing points following a Gaussian distribution centered on the simulated taxel ( $\mu = 0$ ;  $\sigma = 5$  cm). The velocity of the object was a vector directed from the starting point to the landing point, with speed uniformly distributed between 5 cm/s and 15 cm/s (but constant over the time). With the object’s trajectory thus defined, its position was then sampled at 50 ms.

#### D. Avoidance and reaching controller

The representations learned were finally utilised in an avoidance/reaching scenario. The robot was able to exploit the learned representation in order to either avoid or catch an approaching object with any of the skin parts that have been trained. The experiments were conducted by presenting the robot with a series of objects detected through the optical

flow tracker (Section II-A.4), similarly to the learning stage. Any oncoming object thus triggered an activation in each taxel given by the taxel's previous experience with such events (in terms of  $D$ ). Consequently, this gave rise to a distribution of activations throughout the skin. In order to achieve the desired behavior, we implemented a velocity controller able to move any point of either the left or the right kinematic chain in a desired direction. During an avoidance task, the motion should be directed away from the point of maximum activation, along the normal to the local surface in that point. For catching, the desired movement vector is the same, only in opposite direction. For this reason, we computed a weighted average for both the position of the avoidance/catching behavior and its direction of motion:

$$\begin{aligned} \mathbf{P}(t) &= \frac{1}{k} \sum_{i=1}^k [a_i(t) \cdot \mathbf{p}_i(t)] \\ \mathbf{N}(t) &= \frac{1}{k} \sum_{i=1}^k [a_i(t) \cdot \mathbf{n}_i(t)] \end{aligned} \quad (5)$$

where  $\mathbf{P}(t)$  and  $\mathbf{N}(t)$  are the desired position and direction of motion in the robot's Root FoR respectively,  $\mathbf{p}_i(t)$  and  $\mathbf{n}_i(t)$  are the individual taxels' positions and normals. These are weighted by the activations,  $a_i(t)$ , of the corresponding taxels. The weighted average is computed by cycling through all the taxels whose activation is bigger than a predefined threshold at any given time. Therefore, the resultant position and the direction of motion of the avoidance/catching behavior were proportional to the activation of the taxels' representations and changed dynamically as the activation levels of different taxels varied. The velocity control loop employed a cartesian controller [?] whose reference speed was fixed to 10cm/s.

### III. RESULTS

#### A. Learning in a single taxel model

The parameters of the proposed representation (Section II-B) and its behavior in different circumstances were investigated in a single taxel model (Section II-C).

Firstly, we studied the effect of two key parameters: the number of bins (nbins) used for discretizing the input space and the standard deviation  $\sigma$  used in the Parzen Window representation (cf. Equation 4). For all the experiments, the number of input events was 100, i.e. 100 objects were fired toward the taxel and sampled at 20 Hz (50ms). The representations learned are shown in Fig. 6, depicting 4 rather extreme combinations: nbins = [8, 48] and  $\sigma = [4 * \text{bin\_size}, 0.25 * \text{bin\_size}]$  respectively (where bin\_size is the width of the single bin and varies with nbins). Whereas a smaller number of bins (top plots) might be useful in the case of few training samples (more data points per bin), it has the obvious consequence of losing resolution of the representation. Conversely, a high number of bins (bottom plots) is prone to give rise to very "jagged" profiles, in particular if the input space is not sufficiently sampled. To counter this effect, the bandwidth parameter  $\sigma$  of the Parzen

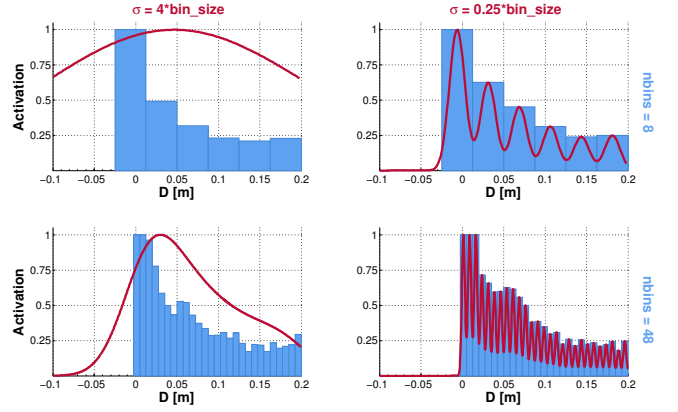


Fig. 6: Different parameter settings of proposed representation. Y-axis, "activation", corresponds to the learned likelihood of the object at distance  $D$  contacting the taxel. Blue rectangles represent the value of  $f(D)$  (cf. Eq. 2) according to the discretization of the input space into bins. The red line represents the Parzen Window interpolation of  $f(D)$  (Eq. 4).

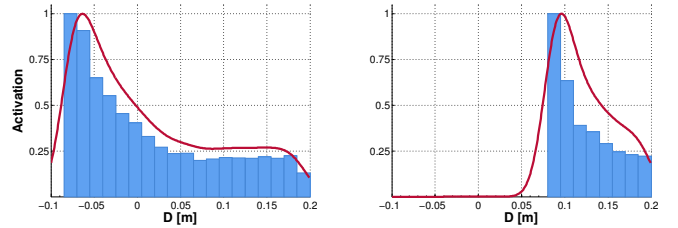
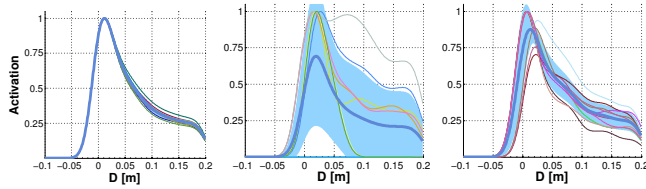


Fig. 7: Effect of systematic errors / offsets on learned representation. (left)  $-8\text{cm}$  error; (right)  $+8\text{cm}$  error. See text for details and Fig. 6 for a description of the plots.

window interpolation can be utilised. This is demonstrated in the left plots, where  $\sigma = 4 * \text{bin\_size}$  demonstrates a pronounced smoothing effect, while sacrificing some of the details of the representation. Therefore, our final choice was a compromise between these extremes: nbins = 20,  $\sigma = \text{bin\_size}$ . From now on, this set of parameters will be used.

In a second step, the proposed representation was tested under different environmental conditions. In particular, we validated it with different amounts of systematic error, noise in the measurements, and number of training samples. First, in order to come closer to the situation in the real robot, we accounted for the fact that the object position is subject to systematic errors / offsets as detailed in Section II-A.5. An offset of  $-8\text{cm}$  ( $+8\text{cm}$ ) was thus introduced on the distance measurements in the simulation – see Fig. 7. The representation compensates for the offset: it is most strongly activated at the distance corresponding to the actual contact, rather than at 0 distance that comes from the inaccurate model (or measurement).

Subsequently, we performed a set of simulations in which the taxel was subject to a variable number of input events and different amounts of noise. The results of 10 trials with every parameter setting are depicted in Fig. 8. From now on, only Parzen window interpolation will be shown (no bins). Fig. 8a



(a) 1000 events per trial, (b) 10 events per trial, (c) 1000 events per trial, no noise (ideal case) no noise Gaussian noise

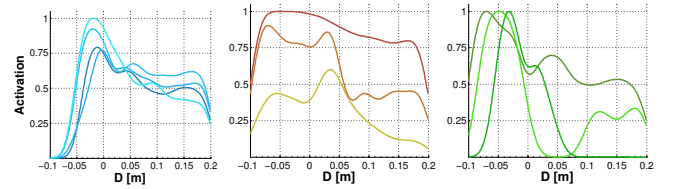
Fig. 8: *Effect of number of samples and noise on representation.* Representation learned from single trials (thin lines in plots); average of 10 trials (thick blue lines); standard deviation (blue area). (c) Gaussian noise added to every velocity ( $\mu = 0, \sigma = 5\text{cm/s}$ ) and position ( $\mu = 0, \sigma = 2\text{cm}$ ) component of the simulated oncoming object.

represents the ideal case, with 1000 input events per trial and no noise in the measurements: with a sufficient amount of data, the responses converge to a stable representation. Fig. 8b depicts an extreme situation with only 10 input events per trial (10 objects coming onto the taxel). In this case, there is high variability in the learned representations, indicating that more samples would be necessary. Finally, Fig. 8c is addressing the presence of noise in the acquisitions (with 1000 events per trial). The results are only slightly drifting from the ideal case of 8a.

### B. Learning in the real robot

The proposed framework was then tested in a real-world scenario in which external stimuli were nearing the robot’s body—real objects were coming onto the iCub’s skin. The objects were detected, tracked and their trajectory prior to contact was recorded and later used for learning the representation of nearby space in corresponding taxels. The tracking was performed with the visual processing pipeline described in Section II-A.4. This setup was validated using two differently shaped objects—a cube and a small soccer ball—approaching the taxels on the robot’s body. Training was applied to 4 virtual taxels (cf. Section II-A.1) placed in the inner part of the left forearm, 4 belonging to the outer part of the left forearm, and 4 virtual taxels of the palm of the right hand. Please refer to the accompanying video for an overview of the experiments (full resolution available at <http://youtu.be/scaFDZZnIZs>).

With both objects taken together, we conducted a total of 138 trials for the right hand, 126 trials for the inner part of the left forearm, and 76 trials for the outer part of the left forearm. Note that each of the trials/events of objects nearing the robot were sampled for 3s before contact and resulted in up to 60 samples entering the representation of individual taxels. The results are shown in Fig. 9 with right hand on the left (all 4 taxels shown; 1916 samples per taxel on avg.), inner part of the left forearm in the middle (3 taxels out of 4 received sufficient amount of data, 864 samples per taxel on avg.), the outer part on the right (3 taxels out of 4 are depicted; 246 samples on avg.). Unlike in Fig. 8, here every line in the plot corresponds to the aggregate representation learned by a particular taxel (in its FoR) from all events taken



(a) Right hand (palm). (b) Left forearm (internal taxels). (c) Left forearm (external taxels).

Fig. 9: *Representation of nearby space learned by individual taxels on the iCub.* Every line in the plot corresponds to the representation learned by a particular taxel (in its FoR).

together.

The representations learned for the taxels on inner part of the right hand (right palm) are very smooth, with little variance between the neighbouring taxels and matching the predictions obtained from simulation (cf. Fig. 8c). The forearm taxel representations learned (Fig. 9b, Fig. 9c) rely on comparatively fewer samples than the palm and are thus less smooth and display bigger variance (but some variance is correct due to the different physical placement of the taxels and hence different experience they were subject to). Interestingly, the forearm taxels’ representation demonstrate that there was a systematic error (as we studied in simulation, cf. Fig. 7) that can be possibly attributed to the coordinate transformation pipeline (Section II-A.5). The effect is less clear in the inner part of the forearm (middle figure), but the peaks of two taxels’ representations are suggestive of a positive offset of about 3 cm. For the outer part of the forearm (right figure), the effect is clearly visible with maximum activation crossing the distance axis at a negative offset of around  $-5\text{ cm}$ . Since the taxels lie on opposite sides of the forearm, the opposite offset sign is plausible. Importantly, the representations automatically compensate for this error.

### C. Exploitation of learned representations of nearby space

The utility of the learned representations was validated during additional experimental sessions. The setup was identical to the training sessions, but we used a new object (a pink octopus toy) to approach the robot’s body parts. First, the performance of the representation was qualitatively verified by relaying the activations pertaining to the the taxels’ nearby space to the visualization that normally receives activation from the iCub skin (with green color for the nearby space activations; red for tactile inputs – see accompanying video). Indeed, objects approaching the body parts trained previously elicited predictive, prior-to-contact, activations.

Then, to demonstrate the behavioral utility of this capacity, we connected these activations that anticipate contact to avoidance/reaching controllers as described in Section II-D above. Only activations above a certain threshold contributed to the resultant movement vector that was executed by the controller. This threshold was empirically set to 0.4, corresponding to a 40% chance of that taxel being contacted by the oncoming object (according to the representation it learned).

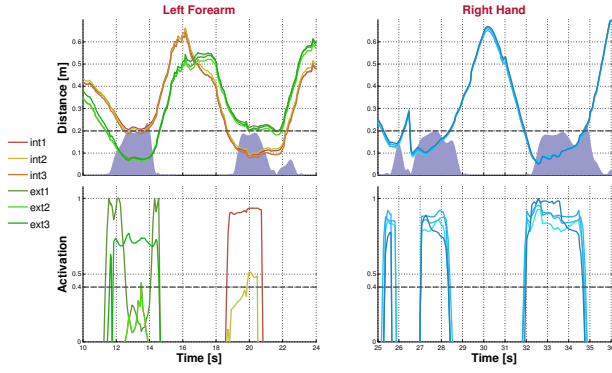


Fig. 10: *Avoidance behavior.* (left) Left forearm. First approaching behavior was directed to the external part of the forearm (taxels in tones of green); second approach toward the internal part (taxels in tones of red) (right) Right hand. See text for details and consult accompanying video.

1) *Avoidance with whole body surface:* An experimental session of roughly 20 min. duration in which the experimenter performed a series of approaching movements, alternating between the body parts and varying the approaching direction, was conducted. Here we restrict ourselves to a qualitative assessment. In short, the avoidance behavior was successfully triggered prior to contact in all cases. A snapshot illustrating typical behavior in a 14 s window for the left forearm (Fig. 10 left) and a 11 s window for the right palm (Fig. 10 right) is shown—with two approaching events in each plot. Representations pertaining to the same taxels of the left forearm and the right hand shown in Fig. 9 were considered. The top plots depict the distance of the approaching object from the individual taxels (in their respective FoR). The bottom plots show the activations of the learned representations for each taxel. As the object comes closer, there is an onset of activation in the representations of the “most threatened” taxels. Once the activation level exceeds a predefined threshold (0.4 in this case – horizontal line in bottom plots), the avoidance behavior is triggered. This is illustrated in the top plots with the shaded violet area that marks the velocity of the body part as commanded by the controller. The upper plots clearly demonstrate that the avoidance behavior was effective—a safety margin was always preserved as the object never touched the robot.

2) *“Reaching” with arbitrary body parts:* In a similar fashion, we probed the “reaching” controller, which was identical to the avoidance one but with the opposite direction of movement in a roughly 10-min. session. A snapshot illustrating the performance while approaching both the inner part of left forearm and the right hand is shown in Fig. 11. The graphical illustration is the same as in the avoidance case. The spatial representations pertaining to the taxels get activated (bottom plot) and trigger the movement, which is approaching the object this time. In addition, the bottom plot illustrates also the physical skin activation (red shaded area). Importantly, contact is generated in both cases, as the skin activation testifies.

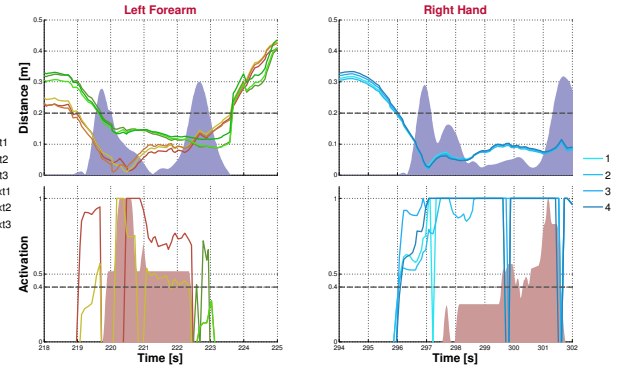


Fig. 11: *“Reaching” with arbitrary body parts.* (left) Left forearm. Internal taxels in tones of red; external taxels in tones of green. (right) Right hand. See text for details and consult accompanying video.

#### IV. DISCUSSION AND CONCLUSION

In this work, we presented to our knowledge the first robot that learns a distributed representation of space around its body by exploiting a whole-body artificial skin and through physical contact with the environment. More specifically, every taxel has a spatial receptive field extending into 20 cm along the normal to the skin surface. In this space, “visual events” triggered by objects coming close to the robot are recorded. If they eventually result in physical contact with the skin, the activated taxels update their representation tracing back the oncoming object and increasing the stored probability that such an event is likely to contact the particular taxel. Other taxels on the body part that were not physically contacted also update their representations with negative examples. The spatial RF around every taxel is mediated by an initial kinematic model of the robot; however, it is adapted from experience, thus automatically compensating for errors in the model as well as incorporating the statistical properties of the oncoming objects. This representation naturally serves the purpose of predicting contacts with the whole body of the robot, which is of clear behavioral relevance. Furthermore, we devised a simple avoidance controller that is triggered by this representation, thus endowing a robot with a “margin of safety” around its body. Finally, simply reversing the sign in the controller we used gives rise to simple “reaching” for objects in the robot’s vicinity, which automatically proceeds with the most activated (closest) body part.

An important asset of the proposed architecture is that learning is fast, proceeds in parallel for the whole body, and is incremental. That is, minutes of experience with objects coming toward a body part give already rise to a reasonable representation in the corresponding taxels that is manifested in the predictive activations—prior to contact—as well as avoidance behavior. The smoothing approach used (Parzen windows applied to the discrete representation) specifically contributes to this effect in the case of undersampled spaces.

One possible practical limitation of the presented architecture could be its computational and memory requirements. The distributed and parallel nature of the representation has

many advantages. At the same time, however, the complexity grows linearly with the number of taxels—each of them monitoring its spatial RF and possibly updating the representation. Nonetheless, this can be mitigated by adapting the resolution of the learned representation—in two ways. First, the spatial resolution regarding the taxels we have chosen can be easily adapted by redefining the “virtual taxel” concept (currently, we worked with “virtual taxels” of around 2 cm in diameter on the skin surface, corresponding to the size of the triangular modules). This would directly affect both the computational requirements—fewer taxels (threads) will monitor the space around them—as well as memory requirements, since fewer representations will be stored and updated. Second, as we have shown in the simulation part, the resolution of the space around the taxels can also be adapted by choosing the number of bins. This directly impacts the storage needed, but also affects the computation time during the application of the Parzen window algorithm.

The demonstrators—avoidance and “reaching”—are simply exploiting the Cartesian Controller to generate movements of a virtual point that is a result of voting of taxels activated by an object near the robot. The direction of the movement is also a weighted average of the normals of the activated taxels. Avoidance differs from “reaching” in the direction of this movement vector only. The response is thus local in a sense that there is only one averaged locus for the response. At the same time, the response is executed globally, since the Cartesian controller is employed, recruiting multiple joints in a coordinated fashion (unlike local reflexes that involve single joints only [?], [?]). However, this approach will not automatically scale to multiple skin parts activated at the same time (the averaging may produce counterintuitive locus and movement vectors in some configurations) or to the presence of multiple objects near the robot. The “reaching” behavior is in fact rather a local “magnet-like” response that will pull the skin parts close to an object toward it. No response will be elicited if the object leaves the 20cm zone surrounding the body. Therefore, integrating the proposed representation with proper reaching in the robot’s workspace in the presence of clutter, while utilising the safety margin or the “magnetizing margin” on the way, remains the topic of future work. In addition, the proposed representation could also be expanded by incorporating an additional variable next to the distance, namely the velocity or time to contact of the oncoming objects. Finally, the framework proposed is applicable also to other robots that are equipped with the key sensory modalities: vision (could be easily replaced by other sensors such as Microsoft Kinect or laser range finders), proprioception, and touch (see [?], [?], [?]).