SUPPORTING MATERIAL
SOM learning from binary vs. continuous input data

Matej Hoffmann, Zdenek Straka, Igor Farkas, Michal Vavrecka, and Giorgio Metta

For both simulated and real tactile stimuli that served as inputs to the SOM algorithm in the main article, binary taxel activations were used. This supporting document investigates the effects of this choice by comparing with continuous inputs.

First, we provide a theoretical account. The SOM tends to map similar inputs to neighboring neurons in the map by being forced to follow the principle known as topographic mapping. Hence what matters are the similarities between two input vectors (typically measured by their Euclidean distance or the dot product). Given this, the tactile inputs represented as binary vectors (with a cluster of ones for activated taxels, and zeros for inactivated taxels) show similarity (i.e. large dot product) if they correspond to two neighboring locations on the skin (because they share some activated taxels). However, different shapes of input vectors on this skin, such as Gaussian blobs, or pyramid-shaped patterns, will lead to the same organization because the distance matrix (of input vectors) will be very similar (due to similar overlaps between neighboring stimuli).

The second part of this document is devoted to an empirical assessment: we ran supplementary experiments with identical SOM settings but continuous input data in order to directly compare with the results reported in the main article. The MRF (Maximum Receptive Field size) modification was not considered (MRF=1 was used, which is equivalent to a standard dot product SOM algorithm). The comparison shows that the both binary and continuous data lead to a very similar organization of the learned maps.

Continuous input data on the simulated skin

Firstly we ran an experiment using continuous input data on the simulated skin, to be compared with binary inputs (see Section III.A.1 of the main article). The stimuli were randomly generated as Gaussian blobs, where activation of the taxel in a position (i, j) at time t was calculated as

$$S_t(i,j) = G_t \frac{1}{\sqrt{2\sigma_t^2 \pi}} e^{-\frac{\|(i,j)-sr_t\|^2}{2\sigma_t^2}}, \tag{1}$$

where $G_t \sim \mathcal{N}(1, 0.2)$ is a gain at time t, $\sigma_t \sim \mathcal{N}(1.2, 0.2)$ is a standard deviation at time t, $sr_t \sim \mathcal{U}([1, 20] \times [1, 20])$ is uniformly and randomly generated position of the stimulus on the skin at time t. An example is in Fig.1.

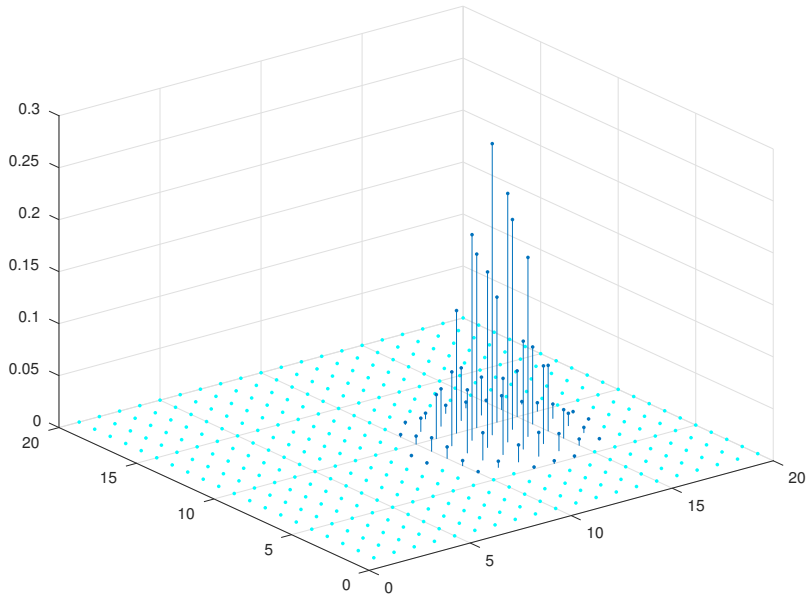

Figure 1: **Example of a Gaussian stimulus on the simulated skin.** Each dot represents a corresponding taxel activation. Dot location on the horizontal plane denotes the taxel position on the skin. The vertical component denotes the value of the taxel activation.

Other parameters of the SOMs and learning were the same as the binary counterpart. The learned SOMs had a very similar organization to those trained on binary input data—as illustrated in Fig. 2.

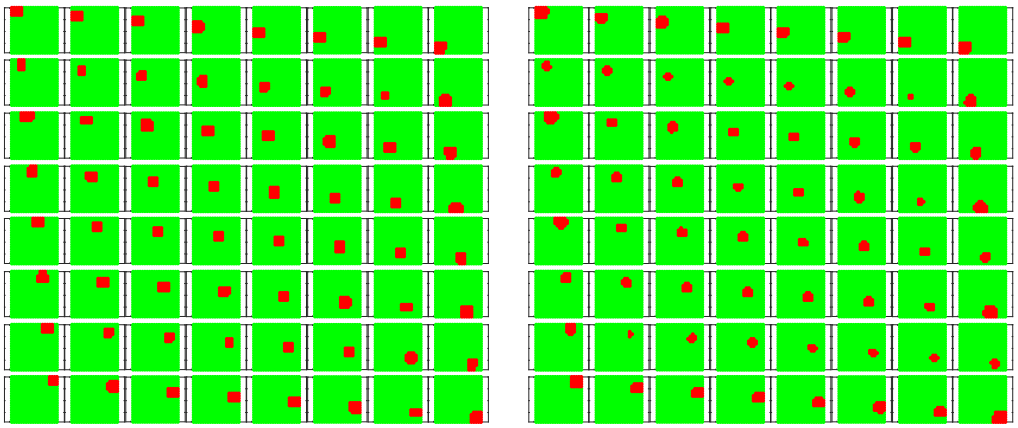

Figure 2: **Comparison of SOMs trained on binary (left) and Gaussian stimuli (right) – simulated skin.** The 8x8 matrix is the lattice of output neurons. Each element (subplot) then depicts a miniature version of the simulated skin, in which the set of red taxels represents the receptive field of the corresponding neuron. For visualization, we used a biologically inspired method of determining the RFs of individual neurons please see Section Biomimetic RFs from simulated skin stimulation under II-D (main article).

Continuous input data on the iCub torso

Secondly, we performed an experiment using continuous input data from the iCub torso, to be compared with binary inputs (see Section III.A.2 of the main article). The "continuous" training data comes from the same data set as the binary data—continuous data was logged from a different port, with values proportional to pressure and in the range $[0, 255]$. Based on the actual range in the data set, taxel activations were then divided by 168.47, scaling them to $[0, 1]$. An example stimulus is in Fig. 3.

The parameters of the SOMs and learning were same as in the binary case. As in the previous case, the SOMs after training had a very similar organization, as illustrated in Fig. 4.

All datasets, functions, etc. used in this analysis are available in "S4 Data and Code" at:
`https://github.com/matejhof/robotic-homunculus-supporting-materials`.

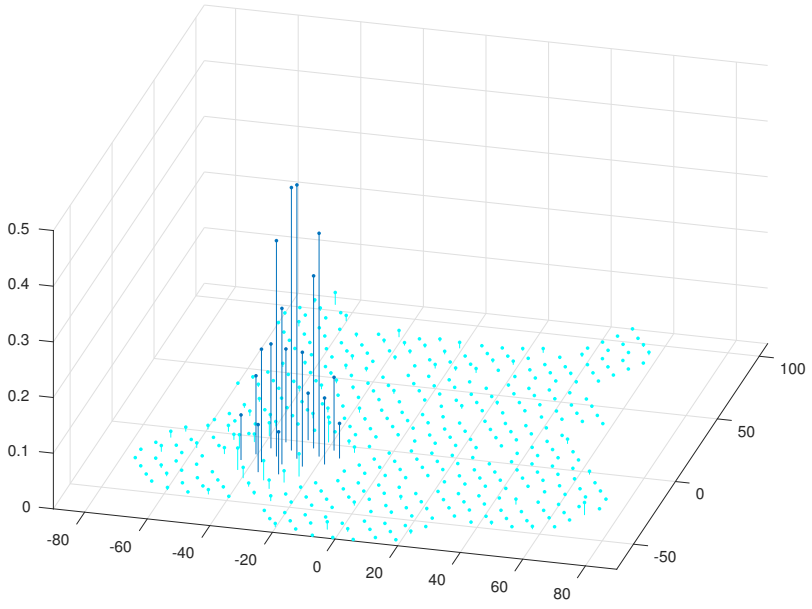

Figure 3: **Example stimulus on the iCub torso.** Each dot represents a taxel activation. Dot location on the horizontal plane denotes the taxel position on the skin. The vertical component denotes the value of the taxel activation.

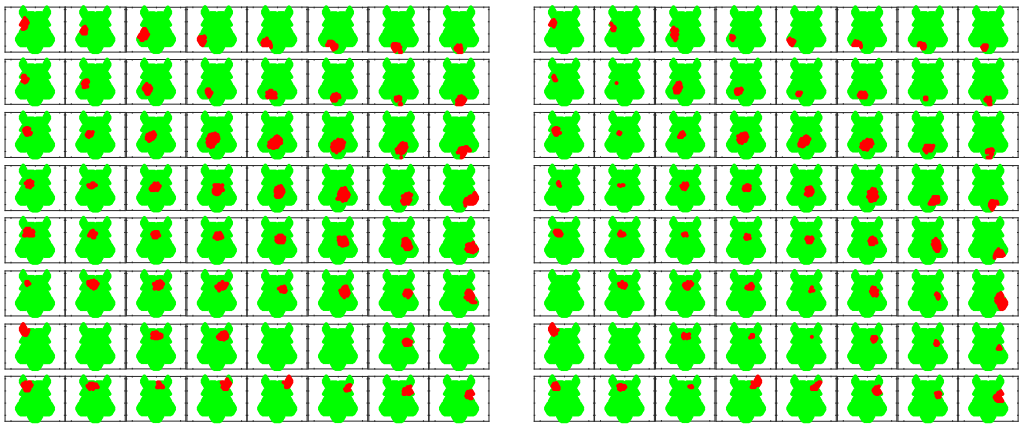

Figure 4: **Comparison of SOMs learned on binary (left) and continuous stimuli (right) – iCub torso.** The 8x8 matrix is the lattice of output neurons. Every element (subplot) then depicts a miniature version of iCub torso, in which the set of red taxels represents the receptive field of the corresponding neuron. For visualization, we used a biologically inspired method of determining the RFs of individual neurons (a threshold for the binary case was set to 10, for the continuous case was set to 6) – please see Section Biomimetic RFs from simulated skin stimulation under II-D of the main article.