Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Outdoor self-localization of a mobile robot using SFA

Benjamin Metka, Mathias Franzius, Ute Bauer-Wersing

2013

Preprint:

This is an accepted article published in Int. Conf. on Neural Information Processing (ICONIP). The final authenticated version is available online at: https://doi.org/[DOI not available]

Outdoor self-localization of a mobile robot using slow feature analysis

Benjamin Metka[1], Mathias Franzius[2], and Ute Bauer-Wersing[1]

[1] University of Applied Sciences Frankfurt,
Nibelungenplatz 1, 60318 Frankfurt am Main, Germany
bmetka@fb2.fh-frankfurt.de
[2] Honda Research Institute Europe GmbH,
Carl-Legien-Straße 30, 63073 Offenbach, Germany

Abstract. We apply *slow feature analysis* (SFA) to the problem of self-localization with a mobile robot. A similar unsupervised hierarchical model has earlier been shown to extract a virtual rat's position as slowly varying features by directly processing the raw, high dimensional views captured during a training run. The learned representations encode the robot's position, are orientation invariant and similar to cells in a rodent's hippocampus.
Here, we apply the model to virtual reality data and, for the first time, to data captured by a mobile outdoor robot. We extend the model by using an omnidirectional mirror, which allows to change the perceived image statistics for improved orientation invariance. The resulting representations are used for the notoriously difficult task of outdoor localization with mean absolute localization errors below 6%.

Keywords: Self-Localization, SFA, Mobile Robot, Biomorphic System, Omnidirectional Vision, Outdoor Environment

1 Introduction

Self-localization is a crucial ability for animals. In rats, hippocampal place cells fire when the animal is in a certain location and these cells are strongly driven by visual input [13]. How does the brain extract position information from the raw visual data it receives from the retina? While the sensory signals of single receptors may change very rapidly, e.g., by slight eye movement, the brain's high level representations of the environment (where am I, what objects do I see?) typically change on a much lower timescale. This observation has lead to the concept of slowness learning ([1–4]).

It has been shown earlier that slowly varying features extracted from the visual input of a virtual rat can model place cells and head-direction cells [5, 10]. Recordings from rats' place cells in open field experiments typically show that cells encode the animal's own position, invariant to head direction. Theoretical analysis of the biomorphic model in [10] has shown that in slowness learning, the resulting representation strongly depends on the movement statistics of the

animal. To achieve position encoding with invariance to head direction, for example, a relatively large amount of head rotation around the yaw axis compared to translational movement is required during mapping of the environment.

Here, we extend the results from [10] by applying the model to a mobile robot in an outdoor environment for the first time. Furthermore, we extend the system by using an uncalibrated omnidirectional mirror[3], which allows to easily add simulated rotation of the camera system. Thus the system finds orientation-invariant representations of its own position without having to rotate the camera or the robot much[4]. In the next section, we briefly describe the model as introduced in [10], and explain our extensions to the model.

2 Model for learning self-localization

Slow Feature Analysis SFA solves the following objective [3]: given a multi-dimensional input signal $\boldsymbol{x}(t)$, find instantaneous scalar input-output functions $g_j(\boldsymbol{x})$ such that the output signals $y_j(t) := g_j(\boldsymbol{x}(t))$ minimize $\Delta(y_j) := \langle \dot{y}_j^2 \rangle_t$ under the constraints $\langle y_j \rangle_t = 0$ (zero mean), $\langle y_j^2 \rangle_t = 1$ (unit variance), $\forall i < j : \langle y_i y_j \rangle_t = 0$ (decorrelation and order) with $\langle \cdot \rangle_t$ and $\dot{y}$ indicating temporal averaging and the derivative of y, respectively.

The Δ-value is a measure of the temporal slowness of the signal $y_j(t)$. It is given by the mean square of the signal's temporal derivative, so small Δ-values indicate slowly varying signals. The constraints avoid the trivial constant solution and ensure that different functions g code for different aspects of the input. We use the MDP [8] implementation of SFA, which is based on solving a generalized eigenvalue problem.

Orientation Invariance The goal for our self-localizing robot is to extract the robot's position on the x- and z-axis as slowly varying features and become invariant to orientation. As stated above, learned slow features strongly depend on the movement pattern of the mobile robot during training. In order to achieve orientation invariance, the orientation of the robot has to change on a faster timescale than its position during training. A constantly rotating robot with a fixed camera is inconvenient to drive, and a robot with a rotating camera is undesirable for mechanical stability and simplicity. As an alternative, we simulate additional robot rotation, which is illustrated in Fig. 1.

Network architecture and training As input image dimensionality is too high to learn slow features in a single step, we employ a hierarchical converging network. The network consists of several convergent layers, each consisting of multiple identical nodes arranged on a regular grid. The numbers of nodes and layers are depicted in Fig. 2. Each node performs a sequence of steps: linear SFA for dimensionality reduction, quadratic expansion of the reduced signals, and another SFA step for slow feature extraction. The nodes in the lowest layer process patches of 10x10 RGB image pixels and are positioned every five pixels.

[3] The omnidirectional mirror we used is actually a chrome-colored plastic egg warmer.
[4] Note that also for 360° field of view, orientation invariance is nontrivial.

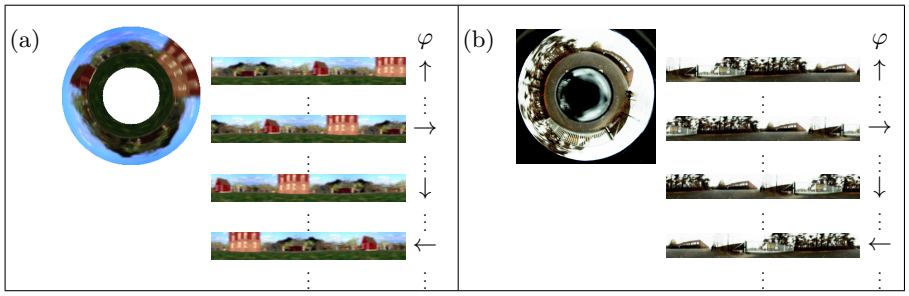


Fig. 1: **Simulated rotation** for (a) simulator and (b) real world experiments. The circular image of the surrounding is transformed to a panorama view with periodic boundaries. Rotation is simulated for every view from one location by sliding a window over the panorama image with increments of 5 pixels. Thus the variable φ denotes the relative orientation w.r.t. the robot's global orientation. Arrows indicate a relative orientation of 0°, 90°, 180° and 270°.

In the lower layers the number of nodes and their dimensionality depends on the concrete setting, but dimensionality is chosen to be a maximum of 300 due to numerical stability. The highest layer contains a single node, whose first (i.e., slowest) 8 outputs $y_j(t)$ we use for all experiments and which we call SFA-output units. The layers are trained subsequently with all training images. Instead of training each node individually, a single node per layer is trained with stimuli from all node locations in its layer and replicated throughout the layer after training. This technique is similar to weight sharing in Neural Networks[5].

Analysis of learned representations How well does a learned output encode position, how much orientation dependency does it have? According to [10], the sensitivity of a SFA-output function $f_j, j = 1...8$ to the spatial position $\boldsymbol{r} = (x, z)$ is characterized by its mean positional variance $\eta_{\boldsymbol{r}}$ over all orientations φ: $\eta_{\boldsymbol{r}} = \langle \mathrm{var}_{\boldsymbol{r}}(f(\boldsymbol{r}, \varphi)) \rangle_{\varphi}$. Similarly, the sensitivity to the orientation φ is characterized by its mean orientation variance η_{φ} over all positions $\boldsymbol{r}$: $\eta_{\varphi} = \langle \mathrm{var}_{\varphi}(f(\boldsymbol{r}, \varphi)) \rangle_{\boldsymbol{r}}$. In the ideal case $\eta_{\boldsymbol{r}} = 1$ and $\eta_{\varphi} = 0$, if a function only codes for the robot's position on the x- and z-axis and is completely orientation invariant. The spatial information encoded by an output will be visualized by two dimensional *spatial firing maps* (see Fig. 2c, 3a, 5a). They illustrate the unit's output value color-coded for every position $\boldsymbol{r} = (x, z)$ for a fixed orientation, which is indicated by an arrow. A unit which codes for the position on a certain axis produces a map that shows a color gradient along this axis. If the SFA-units are perfectly orientation invariant these maps should look the same regardless of the specific orientation.

[5] Note that this design is chosen only for its computational efficiency and that network performance *increases* for individually learned nodes.

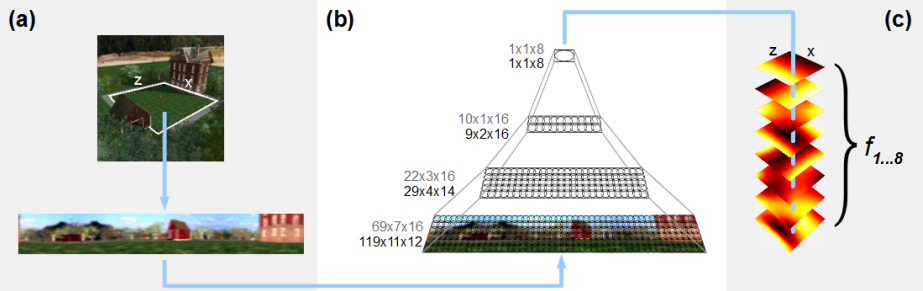


Fig. 2: **Model architecture**. (a) The robot's view associated with a certain position $r = (x, z)$ is steadily captured and transformed to a panorama view. (b) The view is processed by the four layer network. Numbers of nodes in each layer are given for the simulator (gray) and real world (black) experiments, respectively. Each node performs linear SFA for dimensionality reduction followed by SFA on the expanded outputs for slow feature extraction. (c) Eight slowest SFA-outputs $f_{1...8}$ over all positions r. The color coded outputs, so-called *spatial firing maps*, ideally show characteristic gradients along the coordinate axes and look the same independent of the specific orientation. Thus SFA outputs $f_{1...8}$ at position r are the orientation invariant encoding of location.

3 Experiments

The procedure is to record the views and corresponding metric coordinates of the robot from every position during training- and test-runs. After the training step, we need to quantify and visualize the encoded spatial information of the SFA-outputs in a metric way. Therefore we compute a regression function from the SFA-outputs to the metric ground truth positions and subsequently apply it to SFA-outputs.

3.1 Simulated environment

The model was first applied in a virtual reality simulator to validate the extended model under fully controllable settings. The virtual robot was placed on discrete positions forming a regular 30x30 grid. We recorded 624 omnidirectional RGB images for the training set and 196 for test set and transformed them to panorama views with a resolution of 350x40 pixel (Fig. 1a).

Results All resulting SFA-units have a high spatial structure and are almost completely orientation invariant as their outputs for the training views have a mean positional variance $\eta_r \approx 1$ and the mean directional variance η_φ ranges from 0.00 (f_1) to 0.17 (f_8). This is also reflected by the *spatial firing maps* in Fig. 3a which show an obvious encoding for the position on the coordinate axes and look nearly identical under different orientations.

Since the views of the training- and test-run are identical for the same location we only use the test data for the regression analysis. Random 50/50 splits

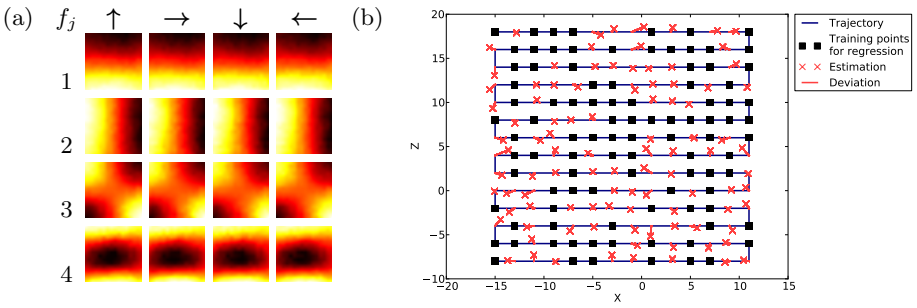


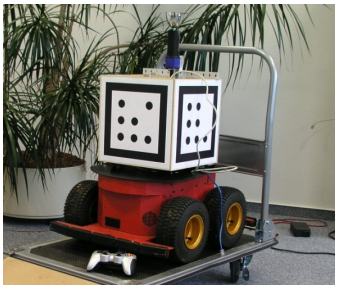
Fig. 3: **Simulated Environment.** (a) *Spatial firing maps* of the four slowest SFA-Outputs $f_{1...4}$ for relative orientations $0°, 90°, 180°$ and $270°$. Obviously the first and second outputs are spatially orthogonal, coding for z- and x-position, respectively. Output values are monotonically increasing from north to south and east to west. The third unit is a mixture of the first two units and unit four is a higher oscillating representation of the first unit. (b) Ground truth and estimated coordinates computed by the regression. Estimations are averaged over the windows of the simulated rotation for one location.

are used to train the regression and evaluate the coordinate prediction. Repeating it 100 times results in an overall mean absolute error (MAE) for the x- and z-coordinate estimation of 1.83% and 1.68%, relative to the coordinate range of the test run (Fig. 3b). Thus the experiment has shown the capability of our extended model to replicate results from [10].

3.2 Real world environment

The experiment was transferred to an outdoor scenario to examine how the model copes with real-world conditions like a non-static environment, changing light conditions and noisy sensor readings. We used a suitable mobile robot (*Pioneer 3AT*) equipped on top with an omnidirectional vision system (Fig. 4a). Outdoor experiments were done within an area of approximately 5x7 meters on asphalted ground. Test data was recorded directly after the training data. The training and test set consists of 5900 and 2800 RGB panaromic images with a resolution of 600x60 pixel. During training and test phase the robot was moved with a wireless joystick at a maximum velocity of 40 cm/s in a grid like trajectory so that the translations along the x- and z-axis were fairly equal distributed with respect to the traveled distance (Fig. 4b).

Unlike in the simulator framework the true position of the robot has to be acquired independently through an external monitoring system. For indoor applications several approaches based on sensors mounted on the room ceiling have been proposed (e.g. [12]), but said approaches turned out to be unfeasible for outdoor applications. To keep ground truth acquisition flexible and robust we mounted a 30cm cube on the robot with optical, binary markers attached to its

(a) 

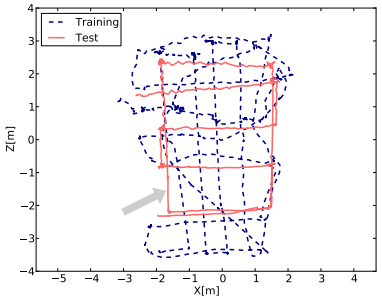
(b) 

Fig. 4: (a) Pioneer 3AT equipped with omnidirectional vision system and marker-box. (b) Captured trajectories of the training- and test-run on an area of approximatly 5x7 meters. The arrow indicates a region in the south-west which has been passed during the test-run but was not part of the training trajectory.

facets (Fig. 4a). A stationary camera was installed to capture images of the whole area throughout the training- and test-runs. 3d-pose was computed, based on the features of the detected markers, by solving the *Perspective-n-Point problem*. Implementation is based on the *OpenCV*-library [11]. In an experimental setup with a HD-webcam the method provided a detection up to a distance of nine meters with a MAE of about 3cm (0.3%), as verified by laser distance meter.
Results All SFA-units of the network have a mean positional variance $\eta_r \approx 1$ and their mean directional variance η_φ ranges from 0.00 (f_1) to 0.05 (f_8) and thus are almost only coding for spatial position while being orientation invariant. Note that the lower magnitude of η_φ, compared to the simulation results, is caused by the faster changing orientation due to the robot's additional real rotation.

As expected the *spatial firing maps* in Fig. 5a do not encode position as clearly as in the simulation due to the non-static environment and the inhomogeneous distribution of position and velocity. *Spatial firing maps* of the first unit encode the position on the z-axis, while x-position is less obvious encoded in the maps of units three and four.

In contrast to the simulation we compute the regression from the SFA-outputs to the metric ground truth positions for the training data and apply it to SFA outputs on the test set. The resulting MAE is 0.23m (5.3%) for the x-coordinate and 0.175m (3.7%) for the z-coordinate and the standard deviation amounts to 0.20 and 0.13. Higher errors can be noticed in a small area in the west that was not passed in the training-run (see Fig. 4b) and an area in the south west, which could also be noticed in the *spatial firing map* with the highest SFA-outputs. Another prominent area with higher errors is located in the north west, where the maps of units two and three show discontinuities. Minor deviations can be observed at turning points in the trajectory, where vibrations of the vision system caused distortions in the unwarped panorama images. Even though the coding for the x-position is less obvious compared to the simulation, it is apparently sufficient for self-localization.

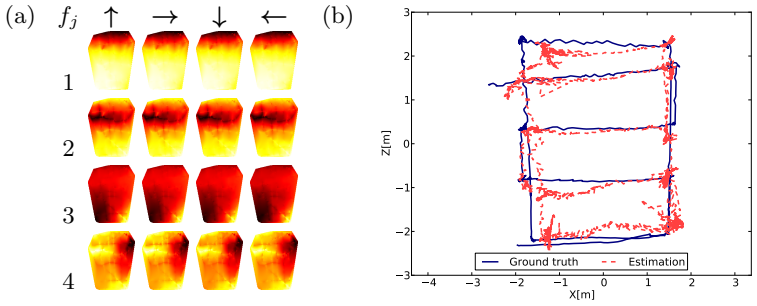


Fig. 5: **Real World Environment.** (a) *Spatial firing maps* of the four slowest SFA-outputs $f_{1...4}$ for relative orientations $0°, 90°, 180°$ and $270°$. First SFA output encodes the position on the z-axis with low values in the north and high values in the south. Notice the area in the south-west showing the highest values. This region has been passed at different timepoints, so that environmental changes led to variations. The second unit is a higher oscillating representation of the first one, which indicates that other varying components of the configuration space changed at least twice as fast as the z-position. Units three and four suggest weak encoding of the x- and z-position. (b) Ground truth and estimated coordinates for the test run. Estimations are averaged over the simulated rotation for one location.

4 Summary and conclusion

We systematically transferred the biologically motivated concept of SFA step-by step into a self-localization task of a mobile robot and successfully showed its application in an outdoor environment. Explorations in the simulated environment have shown that SFA combined with simulated rotation of an omni-directional view allows for self-localization with errors of under 2% relative to the coordinate range. Experiments in the outdoor environment showed an average self-localization accuracy of 0.23m (5.3%) for the x-coordinate and 0.175m (3.7%) for the z-coordinate, which is significantly smaller than the robot's own size (approx. 50x50cm).

The problem of visual self-localization in unknown environments has been investigated in great detail as an inherent part of the Simultaneous Localization and Mapping (SLAM) algorithms (e.g., [9]). Visual SLAM approaches typically require highly calibrated optics and extract local image descriptors, like SIFT or SURF, at regular time intervals to characterize a scene. Typical errors given in the SLAM literature are about 1% to 5% with respect to travelled route. Although localization accuracies are hard to compare in this context, relative errors of our approach are within the same order of magnitude. Our core system, as described in Section 2, however, focuses on simplicity and biological plausibility as it is derived from a model of rat navigation. It repeats the same unsupervised learning in a converging hierarchy which yields location-specific and orientation-invariant slow feature representations by itself and is based on

cheap uncalibrated hardware (but note [7]). It is important to emphasize that unlike in SLAM approaches our aim is not to simultaneously map and locate. Instead the approach, presented here, learns a map of orientation invariant slow feature representations. These are projected to metric space using a supervised regression step. Please note that an autonomous robot does not necessarily need metric coordinates to navigate, but instead it can follow gradients directly in slow feature space.

We have proven the concept of SFA self-localization in real world environments, but nevertheless the experimental results suggest issues that need further investigation: (i) Achieving the orientation invariant representation based on smaller window sizes of the simulated rotation is desirable since it speeds up computation and extends the capabilities of the model to identify objects that were not present during the training phase. (ii) The apparently weak representation of the x-position in the outdoor environment may be due to global changes in the environment, which vary on an equal time-scale as the robot's translation or are not decorrelated (orthonormal) to it. In this respect, choosing another feature representations than the raw pixel values could help to exclude known, changing variables from the configuration space and furthermore improve model performance, if applied to data sets captured at different daytimes or even seasons for the same training area.

References

1. Földiak, P.: Learning invariance from transformation sequences. Neural Comput 3: 194–200 (1991)
2. Stone J.V. and Bray A.: A learning rule for extracting spatio-temporal invariances. Network-Comp Neural 6: 429–436 (1995)
3. Wiskott L. and Sejnowski T.: Slow feature analysis: Unsupervised learning of invariances. Neural Comput, 14(4):715-770 (2002)
4. Körding K., Kayser C., Einhäuser W. and König P.: How are complex cell properties adapted to the statistics of natural scenes? J Neurophysiol, 91(1):206–212 (2004)
5. Wyss R., König P. and Verschure P.: A model of the ventral visual system based on temporal stability and local memory. PLoS Biol 4:1–8 (2006)
6. Berkes P. and Wiskott L.: Slow feature analysis yields a rich repertoire of complex cell properties. J Vision, 5(6) (2005)
7. Milford M., Schulz R., Prasser D., Wyeth G., and Wiles J.: Learning spatial concepts from RatSLAM representations. Robot Auton Syst 55(5):403–410 (2007)
8. Zito T., Wilbert N., Wiskott L., and Berkes P.: Modular toolkit for data processing (mdp): a python data processing framework. Front Neuroinform 2(8) (2009)
9. Davison A., Reid I., Molton N., and Stasse D.: MonoSLAM: Real-time single camera SLAM. IEEE Trans Pattern Anal Mach Int 29(6): 1052–1067 (2007)
10. Franzius M, Sprekeler H., and Wiskott L. Slowness and sparseness lead to place, head-direction, and spatial-view cells. PLoS Comp Biol 3(8) (2007)
11. Bradski G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
12. Smith A., Balakrishnan H., Goraczko M., and Priyantha N.: Tracking Moving Devices with the Cricket Location System. MobiSys (2004)
13. Jeffery K. and OKeefe J.: Learned interaction of visual and idiothetic cues in the control of place field orientation. Exp Brain Res 127(2): 151–161 (1999)