2011 11th IEEE-RAS International Conference on Humanoid Robots
Bled, Slovenia, October 26-28, 2011

AutoIncSFA and Vision-based Developmental Learning for Humanoid Robots

Varun Raj Kompella, Leo Pape, Jonathan Masci, Mikhail Frank and Jürgen Schmidhuber
IDSIA, Galleria 2
Manno-Lugano 6928, Switzerland
Email: {varun, pape, jonathan, kail and juergen}@idsia.ch

***Abstract*—Humanoids have to deal with novel, unsupervised high-dimensional visual input streams. Our new method AutoIncSFA learns to compactly represent such complex sensory input sequences by very few meaningful features corresponding to high-level spatio-temporal abstractions, such as: a person is approaching me, or: an object was toppled. We explain the advantages of AutoIncSFA over previous related methods, and show that the compact codes greatly facilitate the task of a reinforcement learner driving the humanoid to actively explore its world like a playing baby, maximizing intrinsic curiosity reward signals for reaching states corresponding to previously unpredicted AutoIncSFA features.**

I. Introduction

Human beings are able to acquire many skills based on interaction with the environment even without the intervention of a teacher. On humanoid robots, reinforcement learning (RL) [1], [2] could be used to learn skill repertoires, especially if there are self-generated intrinsic curiosity rewards [3]–[7] for action sequences leading to the discovery of new regularities in the observations. Most RL algorithms, however, tend to work only if the dimensionality of the state space is small, or its structure is very simple. To deal with the complex, massive streams of raw sensory information obtained through vision as primary sensor modality, it is essential to reduce the input dimensionality, building low-dimensional but informative representations of the environment.

Here we propose an unsupervised learning system that greatly reduces the dimensionality of a robot's vision data, called AutoIncSFA, which is a novel combination of an autoencoder (AE) [8] and Incremental Slow Feature Analysis (IncSFA) [9], designed to extract few abstract spatio-temporal features that can feed an RL robot with inputs. The AE performs spatial compression while IncSFA extracts spatio-temporal features that change slowly over time. Legenstein et al. [10] have shown a similar two stage learning system composed of a hierarchical slow feature analysis (H-SFA) network [11] for preprocessing and a simple reward-trained neural network on top. This batch technique is not well-suited to developmental learning though. Our method, however, is completely incremental, and helps to make an intrinsically motivated [3], [6], [7] robot learn interesting behaviors from scratch, based on raw pixel input streams.

The rest of this paper is organized as follows. Section I-A reviews SFA. Sections I-B and I-C introduce IncSFA and Autoencoders, respectively. Sec. II presents AutoIncSFA. Sec. III describes several experiments with an iCub humanoid robot learning several skills, by training RL machines to achieve states leading to novel (initially unpredictable, but learnable) AutoIncSFA features.

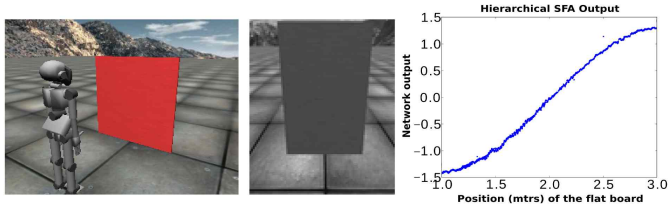

Fig. 1. (a) Simulated iCub watching a flat board move back and forth; (b) sample image from the input dataset; (c) slowest Hierarchical SFA output, coding for the board position.

A. Slow Feature Analysis (SFA)

Slow Feature Analysis (SFA) [12] is an unsupervised learning technique guided by the *slowness* principle. In many settings, the best functions mapping the input stream to the most *slowly changing* output signals are representative of some fundamental invariant agent-world property [13], abstracting away irrelevant details picked up by the sensors that often change at a much faster timescale. Consider for example a mobile agent with high-dimensional video input exploring an otherwise static room. The input is caused by the agent's position and orientation, and the emerging slow features compactly encode this information [11].

Formally, SFA is concerned with the following optimization problem:

Given an I-dimensional input signal $\mathbf{x}(t) = [x_1(t), ..., x_I(t)]^T$, find a set of J instantaneous real-valued functions $\mathbf{g}(x) = [g_1(\mathbf{x}), ..., g_J(\mathbf{x})]^T$, which together generate a J-dimensional output signal $\mathbf{y}(t) = [y_1(t), ..., y_J(t)]^T$ with $y_j(t) := g_j(\mathbf{x}(t))$, such that for each $j \in \{1, ..., J\}$

$$\Delta_j := \Delta(y_j) := \langle \dot{y}_j^2 \rangle \quad \text{is minimal} \tag{1}$$

under the constraints

$$\langle y_j \rangle = 0 \quad \text{(zero mean)}, \tag{2}$$

$$\langle y_j^2 \rangle = 1 \quad \text{(unit variance)}, \tag{3}$$

$$\forall i < j : \langle y_i y_j \rangle = 0 \quad \text{(decorrelation and order)}, \tag{4}$$

with $\langle\cdot\rangle$ and $\dot{y}$ indicating temporal averaging and the derivative of y, respectively.

The problem is to find instantaneous functions g_j generating different output signals that are as *slowly varying* as possible. The decorrelation constraint (4) ensures that different functions g_j do not code for the same features. The other constraints (2) and (3) avoid trivial constant output solutions. The above optimization problem is solved by computing the principal components (with smallest eigenvalues) of a whitened difference signal, where whitening produces decorrelated input dimensions with unit variance.

Given an input signal with two components that vary quickly over time (e.g., $\mathbf{x(t)}$ given by $x_1(t) = \sin(t) + \cos(11\ t)^2$, $x_2(t) = \cos(11\ t)$, $t \in [0, 2\pi]$), SFA will find the slowest feature hidden in the signal (here: $y_1(t) = x_1(t) - x_2(t)^2 = \sin(t)$). Sometimes, however, the slowest component is not the most intuitive one; for example when observing an object that moves in front of a camera and occasionally leaves the field of view, the slowest feature is the presence/absence of the object, not its position.

Figure 1 [9] illustrates the behavior of a Hierarchical-SFA network on a simple simulated interactor, modeled as a flat rectangular board that moves toward and away from the observer (camera of the simulated robot). Hierarchical SFA finds a slow feature that codes instantaneously for the position of the interactor (Figure 1(b)), like place cells in the hippocampus.

A straightforward implementation of the equations 1-4 is batch-wise SFA [12]. However, this approach has several shortcomings:

1) Batch-wise SFA techniques estimate or store covariance matrices from input data, which is expensive for open-ended learning.
2) SFA units are driven by the input signal's derivative [12] approximated by the difference in the signal between successive time instants. For constant signals, however, computing the principal components of the difference signal's covariance matrix will result in singularity errors, since the matrix won't have full rank. This is an important problem in humanoid robot applications where often only a small part of the input image changes. Also, since the covariance matrix of the difference signal is used, this does not extend well to episodic learning.
3) Environments of real robots typically contain uncontrolled external factors that are spatially insignificant but may change more slowly than the object of interest, greatly affecting SFA outputs.
4) Generalization properties of Hierarchical SFA are limited. For example, training H-SFA on one human interactor, but testing on another, may yield erroneous output.

Shortcomings (1,2) are overcome by using incremental slow feature analysis (IncSFA) [9].

B. Incremental Slow Feature Analysis (IncSFA)

SFA uses principal component analysis (PCA) [14] twice. In the first stage, PCA whitens the signal to decorrelate it with unit variance along each PC direction. In the second stage, PCA on the derivative of the whitened signal yields slow features. IncSFA replaces the batch PCA by incremental alternates. In the pre-whitening stage IncSFA uses the state-of-the-art incremental PCA method, Candid Covariance-Free Incremental Principal Component Analysis (CCIPCA) [15]. Since CCIPCA is not feasible for the second stage as the slow features correspond to the *least* significant components, Minor Components Analysis (MCA) [16]–[18] is used. It incrementally extracts the principal component with the smallest eigenvalue (the slowest feature). To extract multiple minor components in parallel, it uses MCA with sequential addition [17].

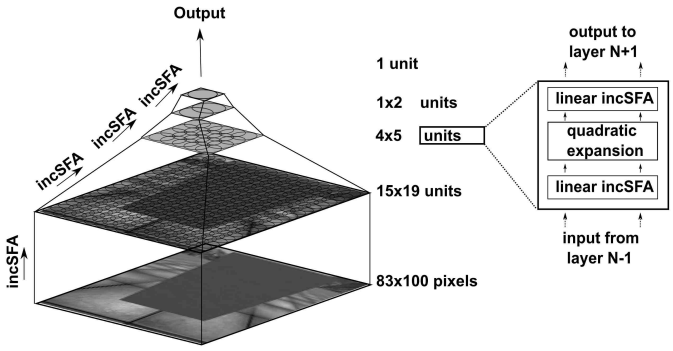

Fig. 2. Hierarchical Incremental Slow Feature Analysis (H-IncSFA) Network

Kompella et al. [9] also discuss an implementation of hierarchical IncSFA (H-IncSFA) to handle high-dimensional image data (Figure 2). The hierarchical network has 616 units spread over three layers, each layer trained sequentially from bottom to top. H-IncSFA does not need to store covariance matrix or input data and is therefore suitable to open-ended developmental learning.

However, H-IncSFA still does not overcome the issues concerning the effect of spatially insignificant and slowly varying external factors. In addition, the higher layers of the H-IncSFA network need the lower layers to converge first. Hence more samples are required before the network is fully functional. A combination with incremental spatial compression techniques such as AutoEncoders (AE) can potentially overcome these issues. An AE with a reduced hidden representation codes only for the dominant spatial information in the dataset, therefore the final output is not severely affected by insignificant yet slowly changing environmental elements. Spatial compression also helps to eliminate much of the redundant static information in the data, resulting in a reduction of the number of IncSFA units.

C. Spatial Compression: Autoencoders

Autoencoders (AEs) are widely used to extract robust features from the data, following the unsupervised encoder-decoder paradigm: A non-linear input transformation yields a compact code sufficient to reconstruct the data. Typically a neural network is trained to implement this identity function under constraints such as: (a) the hidden code layer is much

smaller than the input layer, e.g., [13], (b) the latent representation across the code layer should be sparse, (c) the input is viewed as being noisy (denoising AE [19]), (d) the mapping should have low information-theoretic complexity [20], [21]. AEs tend to generate interesting and useful feature detectors representing only basic constituent features of the data in a way that is robust to noise and perturbations. Applied to image patches [22], AEs learn biologically plausible Gabor-like filters resembling the responses of the striate mammalian cortex [23]. AEs are often used to initialize parameters of deep architectures [19], to perform non-linear PCA [24], to find sparse and/or low-complexity codes [20], or to reduce input dimensionality for RL [25]. Here we briefly describe the basic concepts.

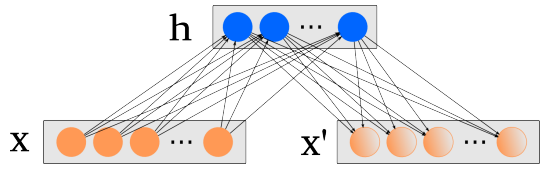

Fig. 3. Schematic representation of an AE. The input x is mapped onto the latent code h (of smaller dimension than x), from which x' is reconstructed.

An auto-encoder (AE) takes an input $\mathbf{x} \in \mathcal{R}^d$ and maps it to the latent representation $\mathbf{h} \in \mathcal{R}^{d'}$ using a deterministic mapping function of the type

$$\mathbf{h} = f_\theta = \sigma(W\mathbf{x} + b) \tag{5}$$

where $\sigma(\cdot)$ is a non-linear function and the parameters are $\theta = \{W, b\}$. This code is then used to reconstruct the input into the vector $\mathbf{x}'$ by reverse mapping of f:

$$\mathbf{x}' = f_{\theta'}(h) = \sigma(W'\mathbf{h} + b') \tag{6}$$

with $\theta' = \{W', b'\}$. In the most widely used variant, the two parameter sets are constrained to be of the form $W' = W^T$, using the same weights for encoding and decoding; the AE is said to have tied weights. Each training pattern x_i is then mapped onto its code h_i and its reconstruction x'_i. A schematic representation is shown in Figure 3. The parameters are optimized via minimization of an appropriate cost function, usually MSE,

$$E(\theta^*, \theta'^*) = \arg\min_{\theta^*, \theta'^*} \frac{1}{2n} \sum_{i=1}^{n} ||x_i - x'_i||_2^2 \tag{7}$$

over the training set $\mathcal{D}_n = \{(x_0, t_0), ..., (x_n, t_n)\}$.

Variations of this model obtain salient features [19], [20] and deal with overcomplete representations [20], [22], [26]. Here we do not need such constraints as the hidden representation is not overcomplete.

II. Method (AutoIncSFA)

Since AEs are able to find compact representations of the relevant components of visual input, and IncSFA can extract relevant variation in time, we propose the following approach to dimensionality reduction of a robot's visual input in both space and time:

1) **Input Signal:** Acquire the current raw I-dimensional input as vector $\check{\mathbf{x}}(t)$.
2) **Normalization:** Normalize the input signal to obtain

$$\mathbf{x(t)} := [x_1(t), ..., x_I(t)] \tag{8}$$

$$\text{with } x_i(t) := \frac{\check{x}_i(t) - \langle \check{x}_i \rangle}{F} \tag{9}$$

where, F is an upper bound of $\mathbf{x}$

$$\text{so that } \langle x_i \rangle = 0, \tag{10}$$

$$\text{and } 0 \leq x_i < 1. \tag{11}$$

3) **AE Update:** For each input pattern **x(t)**, infer the reconstruction $\mathbf{x}'$ and update the weights of the model using gradient descent. The weights are used to get the code $\mathbf{h(t)}$.
4) **IncSFA Update:**
   a) **Whitening by CCIPCA:** The hidden unit activations $\mathbf{h(t)}$ are normalized to generate $\mathbf{z(t)}$ with zero mean and identity covariance matrix $\mathbf{I}$. This so-called *whitening* can be done incrementally with the help of Candid Covariance-free Incremental Principal Component Analysis (CCIPCA) on $\mathbf{h(t)}$.
   b) **Derivative signal:** To capture the variation of the signal over time, $\mathbf{z(t)}$ is differentiated with respect to t to produce $\dot{\mathbf{z}}(\mathbf{t})$. We use the difference over a single time step as a fast approximation of the derivative.
   c) **Slow Features:** By applying incremental minor component analysis to the matrix $\langle \dot{\mathbf{z}}\dot{\mathbf{z}}^\mathbf{T} \rangle$, J eigenvectors with the lowest eigenvalues λ_j are extracted. These are the current estimates of the slow features; $\mathbf{W}(t)$.
5) **Output:** Then, $\mathbf{y}(t) = \mathbf{z}^T(t)\mathbf{W}(t)$ is the AutoIncSFA output.

III. Results

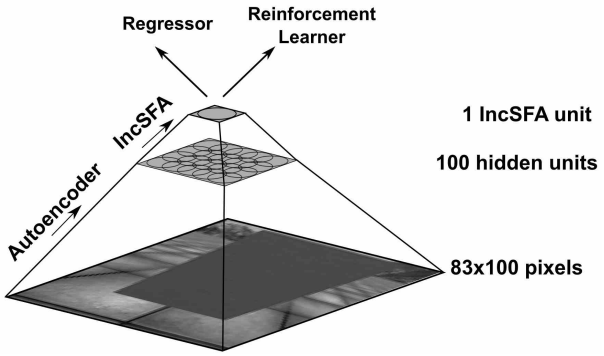

Fig. 4. The network architecture of AutoIncSFA. It contains a single layer of AEs with a single IncSFA unit. The output is fed to either a regressor or a reinforcement learner.

We evaluate the capacity of our algorithm by testing it on an iCub humanoid robot [27]. The robot receives a

high-dimensional video stream, converted to grayscale and downsampled to 83×100 pixels (i.e., an input dimension of $I = 8,300$). Figure 4 illustrates the AutoIncSFA network architecture consisting of an AE with 100 hidden units and a single IncSFA unit on top.

A. Human Interaction Experiment

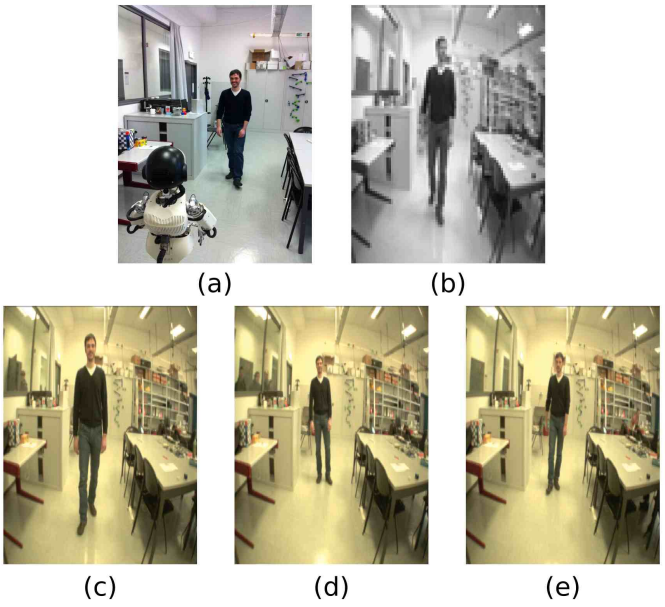

Fig. 5. (a) Experimental Setup: person moving toward and away from the robot. (b-e) Sample images from the dataset, some of which show external elements: (c) Opening/closing of the door in the corridor; (d) people passing by in the corridor; (e) Appearance of leg and shadow of a person sitting at the table.

TABLE I
MOVING ELEMENTS IN THE SCENE

Moving Element	Spatial Significance	Occurrence
Door opening/closing	∼1.5%	once
People passing by	∼2.1%	∼10 times
Person's leg appearance	∼4.5%	∼50 times
Interactor's motion	∼30%	∼200 times b&forth

To evaluate the performance of AutoIncSFA and as a proof of concept, we compare to results of the state-of-the-art batch H-SFA network. A human interactor is walking freely back and forth over the range $[0.6, 3]$ meters in front of a real iCub robot as shown in Figure 5(a). Figure 5(b) shows a sample image from the dataset of 3000 images collected from the robot's left eye. To test the robustness of our model, we infused several spatially insignificant but quite natural external elements in the dataset: a door that opens and closes in the corridor (see top-left corner in Figure 5(c)), people passing by in the corridor (Figure 5(d)), and a sitting man's visible legs and shadow (see bottom-right corner of Figure 5(e)). Table 1 summarizes the moving elements in the dataset along with their spatial significance and their temporal occurrence. A video of the dataset can be found at: http://www.youtube.com/watch?v=T8jZjN4IZ14

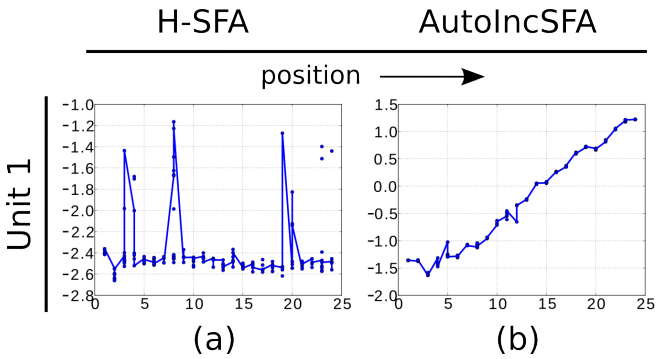

Fig. 6. Response of unit 1 with respect to position, (a) in the H-SFA network; (b) in the AutoIncSFA network. (position (x-axis) scaled in dm)

Both Hierarchical SFA and our model AutoIncSFA are trained on the video footage above. A test set contains 24 positions within the range of $[0.6, 3]$ meters, i.e., each position is separated by 10 cm. Figures 6(a)-(b) show the first output unit of each network plotted with respect to the position of the interactor. We see that H-SFA completely fails, while our method replicates the simulated result (Figure 1(c)), coding for the position of the human interactor. H-SFA is highly sensitive to slowly varying elements despite their spatial insignificance.

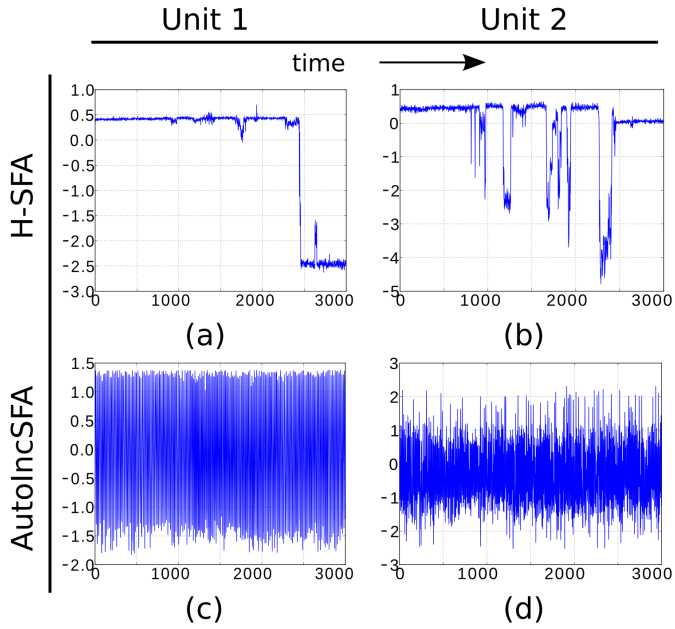

Fig. 7. Response of the first 2 units as a function of time. **H-SFA**: (a) Unit 1 codes for the opening/closing of the door in the corridor; (b) unit 2 detects people passing by. **AutoIncSFA**: (c) Unit 1 encodes the interactor's movement in front of the robot; (d) Unit 2 encodes the second harmonic of the interactor's movement. (time (x-axis) is represented by image samples)

Analyzing the output features of H-SFA, we found that most features code for slowly occurring noisy elements, such as the door opening or closing and people passing by. Only subsequent units encode the interactor position, which usually gets mixed with higher frequencies of the first units. Figure

7(a) shows the activation of H-SFA unit 1 as a function of time. It encodes the door opened at about t=2400. The second unit (Figure 7(b)) detects people moving in the corridor. AutoIncSFA, however, gets rid of spatially insignificant image variations, coding only for dominant ones. Its first unit encodes the interactor's movement (a half-sine wave over the interactor's positions), the second its 2^{nd} harmonic (a full sine wave over the interactor's positions) [11]. We expect our method also to be robust to other small variations of this type.

AutoIncSFA generalizes well to unseen data. We trained it with an interactor shown in Figure 8(a) but tested with a different one shown in Figure 8(b). Figure 8(c) shows the first AutoIncSFA unit's output, encoding the position of the second interactor. A video of the test set for the second interactor can be found at: http://www.youtube.com/watch?v=syUJWCphBOs.

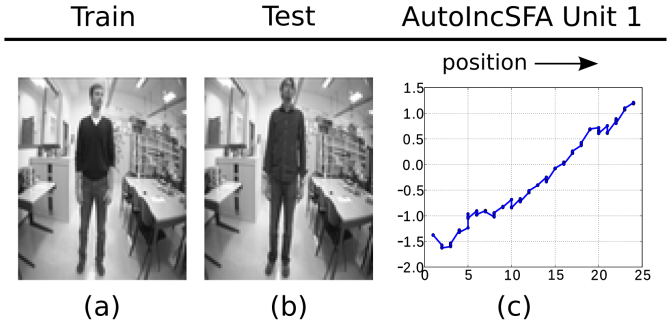

Fig. 8. (a) Sample image from the training set. (b) Sample image from a test set with a different interactor. (c) AutoIncSFA's output response with respect to the new interactor's position. (position (x-axis) scaled in dm)

B. Objects Interaction Experiment

One important application area of AutoIncSFA is episodic learning. Much of developmental learning happens in series of several episodes of interactions with the environment. With a minor modification, the algorithm can be readily extended to episodic tasks. The derivative signal, which is computed as a difference over a single time step, is not computed for the starting sample of each episode, and therefore only updating the whitening vector, not the slow feature vector. Here we present results obtained through the robot's interactions with objects in it's field of view.

1) Single Object Interaction: A plastic cup is placed in the iCub robot's field of view as shown in Figure 9(a). The robot performs motor babbling in one joint using a movement paradigm presented by Franzius et. al [11]. During the course of babbling, it happens to topple the cup on its way (Figures 9(c)-(e)); the episode ends after it. Since the cup being toppled or upright is the "slowest" event in the scene (ignoring the trivial case of static background), AutoIncSFA builds a step response for the object's state (toppled or upright). Figure 9(b) shows the first output unit at the end of 70th episode ($\sim$ 7000 images). Such a clear step response invariant to the robot's arm position is a highly useful feature, greatly facilitating training of a subsequent regressor or a reinforcement learner. A video

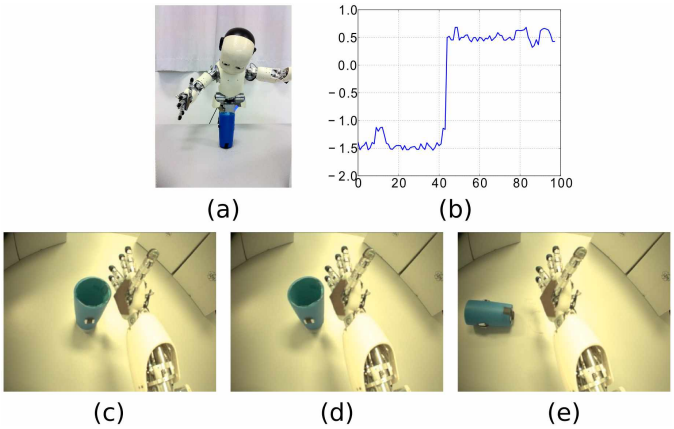

Fig. 9. (a) Experimental Setup (b) AutoIncSFA network output unit-1 at the end of 70^{th} episode. It codes for the toppling of the plastic cup. It is highly active when the cup is toppled, and nearly inactive otherwise (x-axis unit time is represented by image samples in an episode). (c)-(e) Sequence of images when the robot topples the object during motor babbling.

of the experiment can be found at http://www.youtube.com/watch?v=1piHHIvRWe0.

2) Multiple Object Interaction: Here we conduct an experiment similar to the one above, but with two objects in the robot's field of view (a cup and a bottle). The robot performs motor babbling to topple both the objects—see Figures 10(c)-(e). Toppling events of the objects are statistically independent, hence AutoIncSFA learns to code individually for each of the object. Figure 10(a) shows the first output feature at the end of the 145th episode($\sim$ 15,000 images), which encodes the state of the bottle independently from the state of the cup. Figure 10(b) shows the second slow feature which encodes the state of the cup independently from the state of the bottle. The coding order of objects depends on the relative frequency of their toppling events. A video of the experiment can be found at http://www.youtube.com/watch?v=WSGebK-wd2I.

C. Learning a Repertoire of Actions Through Regularity Discovery

The *Formal Theory of Fun and Creativity* [6], [7] mathematically formalizes driving forces and value functions behind all kinds of curious and creative behavior. Consider an agent living in an initially unknown environment. At any given time, it uses one of the many reinforcement learning (RL) methods [1] to maximize not only expected future external reward for achieving certain goals, such as avoiding hunger / empty batteries / obstacles etc, but also *intrinsic* reward for action sequences that improve an internal model of the environmental responses to its actions. Such an agent continually learns to better predict / explain / compress the growing history of observations influenced by its experiments, actively influencing the input stream such that it contains previously unknown but learnable algorithmic regularities which become known and boring once there is no additional subjective *compression progress* or *learning progress* any more [3]–[5], [7]. Schmidhuber et al. have argued that the particular

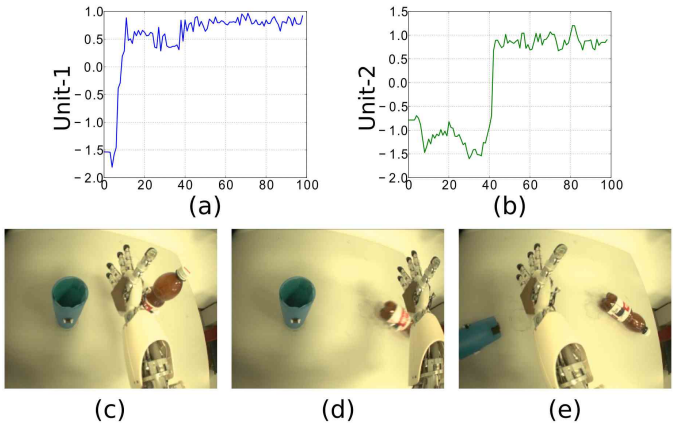

Fig. 10. (a) AutoIncSFA network output unit-1, which codes for the toppling of the bottle in the scene. The output is high when the bottle is toppled and low when it is not, and ignores the state of the second object in the scene (plastic cup). (b) AutoIncSFA network output unit-2, which codes for the toppling of the plastic cup, ignoring the state of the bottle (x-axis unit time is represented by image samples in an episode). **Repertoire of Actions:** (c) Explore (d) Topple the bottle (e) Topple the cup

compression progress-based utility functions associated with this theory explain essential aspects of intelligence including selective attention, curiosity, creativity, science, art, music, humor, e.g., [6], [7].

Essentially, curiosity-driven agents not only focus on potentially hard-to-solve externally posed tasks, but also creatively invent self-generated tasks that have the property of currently being still unsolvable but easily learnable, given the agent's present knowledge, such that the agent is continually motivated to improve its understanding of how the world works, and what can be done in it. Its growing skill repertoire may at some point help to achieve more external reward as well [3], [5], [7].

The permanent intrinsic goal of achieving additional compression progress / prediction progress on the observation history so far can be partially approximated through something our AutoIncSFA is good at, namely, the discovery of invariant spatio-temporal properties of the input stream. Note that any such invariance must reflect an environmental regularity that allows for better compressing the observed data. Hence we can implement a curious, playful robot by simply making it wish to learn to create additional, still unknown, AutoIncSFA-encodable invariances.

As a proof of concept, we build an unsupervised, curiosity-driven, three-stage system with an AutoIncSFA module for extracting spatio-temporal regularities from the visual input stream, a predictor module that learns to predict the slow features, and an RL machine motivated to learn policies to reach states that reduce the errors of the predictor modules, thus being interested in unknown, yet learnable invariances, that is, regularity discovery, or compression progress [6], [7], [28]. Since the focus of this paper is on the construction of useful invariant representations of the environment, not on learning complex actions skills, we use a straightforward tabular action-value-based reinforcement learner and a tabular least-squares predictor. The predictor takes a state-action pairs as an input and predicts the corresponding AutoIncSFA output at time *t*. The reward function is given by the decrease in prediction error, and thus rewards the agent for learning progress. The intrinsic curiosity reward signal allows the robot to focus on those parts of the environment that can easily be *learned* by its limited learning methods.

Learning Phase: The robot performs motor babbling in its (limited) joint space, affecting objects in the scene. With the video from the robot's eyes as an input sequence, AutoIncSFA automatically builds several abstract features of the robot's interaction with the environment as discussed in Section III-B. Once the features converge, a predictor is trained to predict the slow features, together with an RL algorithm that learns a policy for improving the accuracy of the predictor, in an episodic manner. An RL episode ends once the object is toppled or the episode counter reaches its upper-limit (25 steps). In experiments with multiple objects, multiple RL modules are coupled to the AutoIncSFA output features, such that several policies (one for invoking each feature) can be learned and stored for later use. Since the features developed by AutoIncSFA invariantly encode various events in the scene, the RL modules can learn a growing repertoire of robot behaviors.

1) Single Object Interaction: We use a playing RL agent with 10 discrete states (joint positions) and 2 actions (left or right) and a setup similar to the one of Section III-B1. It executes the learning procedure above, creating a policy to topple the plastic cup. The reward signal is derived from the reduction of prediction error [3] ($\dot{\epsilon_p}$) as:

$$R = \begin{cases} 0 & \dot{\epsilon_p} > 0 \\ |\dot{\epsilon_p}| & \dot{\epsilon_p} < 0 \end{cases}$$

Figures 11(a)-(c) show the progress of the RL agent over several episodes. Figure 11(a) shows the Q-table, state-value table, predictor and prediction error entries at the end of the 5th episode. In state 3 the robot is in contact with the cup and topples it by either moving left or right. This is reflected in the Q-table and the predictor, where the values peak at the 3^{rd} state. The prediction error is high since the predictor has not yet learnt the step response. Figure 11(b) and (c) correspond to the 10th and 15th episode. The predictor now has low error, and the Q-table and state-value table almost converged, reflecting a policy for toppling. Figure 11(d) shows a plot of the intrinsic curiosity reward received by the RL agent over several episodes, We see from Figure 11(d) how the reward decreases, indicating that the robot stops the toppling behavior once it has acquired that skill. In a setting with many learnable regularities, the robot could now focus its attention on learning the next skill.

2) Multiple Object Interaction: Next, we conducted another experiment with a setup similar to the one discussed in III-B2, along with two different predictors and reinforcement learners with 11 states (joint positions) and 2 actions (left and right). Since AutoIncSFA learns two features that invariantly encode the states of the objects, these feature outputs can now serve

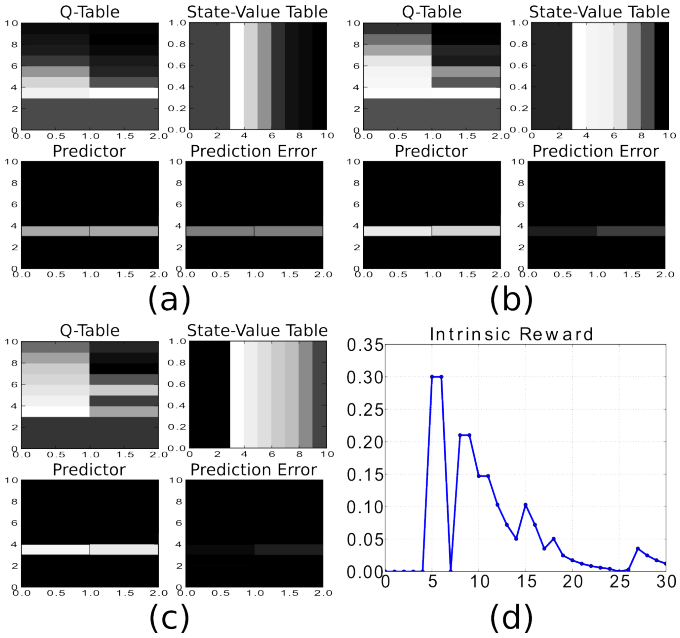

Fig. 11. (a)-(c) Progress of the RL agent at the end of the 5^{th}, 10^{th} and 15^{th} episode. Q-table and state-value table show the developing policy. The predictor predicts the step response in the AutoIncSFA output. The robot topples at state 3 either by moving left or right. The predictor learns to reduce prediction error over the episodes. (d) Plot of the intrinsic curiosity reward signal derived from the decrease in prediction error. (x-axis represents number of episodes)

as an input to the two predictors and reinforcement learners that independently learn policies for reducing the prediction errors. A RL module is selected for updating using an ϵ-greedy criterion based on the cumulative prediction progress of the previous episode. Figure 12(a) shows the Q-table and state-value table for both RL agents at the end of the 30^{th} episode. It can be seen that the Q-table and state-value table for the bottle have a maximum value at state 9, while for the cup, they have a maximum value at state 3. These two different policies now serve as two different skills for the iCub robot. Figure 12(b) shows the intrinsic reward received by both RL agents over several episodes.

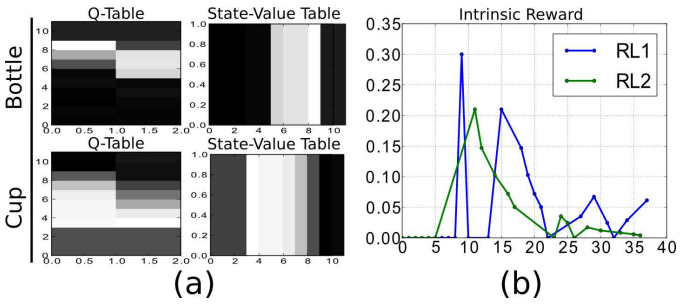

Fig. 12. (a) Policies developed for each of the objects at the end of 30^{th} episode. (b) Intrinsic curiosity reward plot for both RL agents. (x-axis represents number of episodes)

IV. Conclusion

Our novel unsupervised learning method AutoIncSFA derives meaningful low-dimensional spatio-temporal representations of the environment, given high-dimensional raw pixel visual input streams. Applying AutoIncSFA to video camera inputs of the iCub humanoid robot, we showed its robustness to distractions by noisy or less significant event sequences. AutoIncSFA generalizes well on unseen data. It can also feed a subsequent reinforcement learner with compact but informative inputs, greatly helping curious, exploring RL robots to build novel behaviors from scratch, motivating them to create novel, previously unknown AutoIncSFA-encodable invariances or regularities in their input stream. A preliminary proof-of-concept experiment for developmental learning showed how to use AutoIncSFA for building a meaningful action repertoire without any external reward signal. Although we used traditional RL on top of AutoIncSFA, we expect the latter to be even more useful for more complex, modular RL approaches [29]–[33]. We also look forward to carrying out experiments with a moving observer in the immediate future.

Acknowledgment

The experimental paradigm used for the Human Interaction Experiment was first developed by the first author under the supervision of Dr. Mathias Franzius, at the Honda Research Institute Europe. We would like to acknowledge Dr. Franzius for his contributions in this regard. We would also like to acknowledge Dr. Tom Schaul and other researchers from IDSIA, Lugano for their valuable inputs for the development of the ideas presented in this paper. This work was partially funded by the EU projects FP7-ICT-IP-231722 (IM-CLeVeR).

References

[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *Journal of AI research*, vol. 4, pp. 237–285, 1996.

[2] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. Cambridge, MA, MIT Press, 1998.

[3] J. Schmidhuber, "Curious model-building control systems," in *Proceedings of the International Joint Conference on Neural Networks, Singapore*, vol. 2, pp. 1458–1463, IEEE press, 1991.

[4] J. Storck, S. Hochreiter, and J. Schmidhuber, "Reinforcement driven information acquisition in non-deterministic environments," in *Proceedings of the International Conference on Artificial Neural Networks, Paris*, vol. 2, pp. 159–164, EC2 & Cie, 1995.

[5] J. Schmidhuber, "Artificial curiosity based on discovering novel algorithmic predictability through coevolution," in *Congress on Evolutionary Computation* (P. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and Z. Zalzala, eds.), pp. 1612–1618, IEEE Press, 1999.

[6] J. Schmidhuber, "Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts," *Connection Science*, vol. 18, no. 2, pp. 173–187, 2006.

[7] J. Schmidhuber, "Formal theory of creativity, fun, and intrinsic motivation (1990-2010)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 230 –247, 2010.

[8] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length, and helmholtz free energy," in *Advances in Neural Information Processing Systems 6*, pp. 3–10, Morgan Kaufmann, 1994.

[9] V. R. Kompella, M. D. Luciw, and J. Schmidhuber, "Incremental slow feature analysis," in *IJCAI*, pp. 1354–1359, 2011.

[10] R. Legenstein, N. Wilbert, and L. Wiskott, "Reinforcement learning on slow features of high-dimensional input streams," *PLoS Computational Biology*, vol. 6, no. 8, p. e1000894, 2010.

[11] M. Franzius, H. Sprekeler, and L. Wiskott, "Slowness and sparseness lead to place, head-direction, and spatial-view cells," *PLoS Computational Biology*, vol. 3, no. 8, p. e166, 2007.
[12] L. Wiskott and T. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Computation*, vol. 14, no. 4, pp. 715–770, 2002.
[13] J. Schmidhuber and D. Prelinger, "Discovering predictable classifications," *Neural Computation*, vol. 5, no. 4, pp. 625–635, 1993.
[14] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.
[15] J. Weng, Y. Zhang, and W. Hwang, "Candid covariance-free incremental principal component analysis," *Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1034–1040, 2003.
[16] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, no. 6, pp. 927–935, 1992.
[17] T. Chen, S. Amari, and N. Murata, "Sequential extraction of minor components," *Neural Processing Letters*, vol. 13, no. 3, pp. 195–201, 2001.
[18] D. Peng, Z. Yi, and W. Luo, "Convergence analysis of a simple minor component analysis algorithm," *Neural Networks*, vol. 20, no. 7, pp. 842–850, 2007.
[19] P. Vincent, L. Hugo, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, ICML '08, (New York, NY, USA), pp. 1096–1103, ACM, 2008.
[20] S. Hochreiter and J. Schmidhuber, "Feature extraction through LOCOCODE," *Neural Computation*, vol. 11, no. 3, pp. 679–714, 1999.
[21] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.
[22] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1 –8, june 2007.
[23] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision Research*, vol. 37, pp. 3311–3325, December 1997.
[24] M. Scholz and R. Vigário, "Nonlinear pca: a new hierarchical approach," in *ESANN*, pp. 439–444, 2002.
[25] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1 –8, july 2010.
[26] C. Ekanadham, "Sparse deep belief net model for visual area v2," in *Advances in Neural Information Processing Systems 20*, MIT Press, 2008.
[27] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, PerMIS '08, (New York, NY, USA), pp. 50–56, ACM, 2008.
[28] J. Schmidhuber, "Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes," in *Anticipatory Behavior in Adaptive Learning Systems. From Psychological Theories to Artificial Cognitive Systems* (G. Pezzulo, M. V. Butz, O. Sigaud, and G. Baldassarre, eds.), vol. 5499 of *LNCS*, pp. 48–76, Springer, 2009.
[29] J. Schmidhuber, "Learning to generate sub-goals for action sequences," in *Artificial Neural Networks* (T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, eds.), pp. 967–972, Elsevier Science Publishers B.V., North-Holland, 1991.
[30] M. B. Ring, "Incremental development of complex behaviors through automatic construction of sensory-motor hierarchies," in *Machine Learning: Proceedings of the Eighth International Workshop* (L. Birnbaum and G. Collins, eds.), pp. 343–347, Morgan Kaufmann, 1991.
[31] M. Wiering and J. Schmidhuber, "HQ-learning," *Adaptive Behavior*, vol. 6, no. 2, pp. 219–246, 1998.
[32] B. Bakker and J. Schmidhuber, "Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization," in *Proc. 8th Conference on Intelligent Autonomous Systems IAS-8* (F. G. et al., ed.), (Amsterdam, NL), pp. 438–445, IOS Press, 2004.
[33] H. R. Maei and R. S. Sutton, "Gq( ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces," *Proceedings of the 3d Conference on Artificial General Intelligence AGI10*, pp. 1–6, 2010.