Connection Science

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ccos20

Discriminating and simulating actions with the associative self-organising map

Miriam Buonamente[a], Haris Dindo[a] & Magnus Johnsson[b]

[a] Robotics Lab, DICGIM, University of Palermo, Viale delle Scienze, Ed. 6, 90128 Palermo, Italy

[b] Department of Philosophy, Lund University, Helgonavägen 3, Box 192, 221 00 Lund, Sweden

Published online: 08 Apr 2015.

Connection Science, 2015
http://dx.doi.org/10.1080/09540091.2015.1025571

Taylor & Francis
Taylor & Francis Group

Discriminating and simulating actions with the associative self-organising map

Miriam Buonamente[a]*, Haris Dindo[a] and Magnus Johnsson[b]

[a] *Robotics Lab, DICGIM, University of Palermo, Viale delle Scienze, Ed. 6, 90128 Palermo, Italy;*
[b] *Department of Philosophy, Lund University, Helgonavägen 3, Box 192, 221 00 Lund, Sweden*

(*Received 28 March 2014; accepted 26 February 2015*)

We propose a system able to represent others' actions as well as to internally simulate their likely continuation from a partial observation. The approach presented here is the first step towards a more ambitious goal of endowing an artificial agent with the ability to recognise and predict others' intentions. Our approach is based on the associative self-organising map, a variant of the self-organising map capable of learning to associate its activity with different inputs over time, where inputs are processed observations of others' actions. We have evaluated our system in two different experimental scenarios obtaining promising results: the system demonstrated an ability to learn discriminable representations of actions, to recognise novel input, and to simulate the likely continuation of partially seen actions.

Keywords: associative self-organising map; neural network; action recognition; internal simulation; intention understanding

1. Introduction

The "Holy Grail" of a successful human–robot interaction lies in the development of robust techniques to enable a robot to recognise and predict goals and intentions of other agents. Indeed, we see others' actions not only as mere movements, but rather as goal-directed behaviours driven by unobservable internal mental states such as the above-mentioned intentions, but also desires and beliefs.

The set of computational mechanisms and neural pathways that enables humans to go beyond the surface behaviour is usually studied under the umbrella of "theory of mind" or "mindreading" (Goldman, 2006; Premack & Woodruff, 1978). As an example, when we see someone walking to the front door and then ransacking in her pocket, we assume that she is looking for her keys; that is, we *recognise* the proximal action (i.e. that of ransacking the pocket) and *simulate* its likely continuation in the given context in order to predict the most likely distal goal(s) (i.e. that of looking for the key). In other words, we are ultimately interested in inferring the *intention* of the observed act and we use the recognition of the intermediate actions as a proxy that confirms or disconfirms our beliefs, where the intention can be defined as "a plan of action the organism chooses and commits itself to the pursuit of a goal, thus including both a means (action plan) as well as a goal" (Tomasello, Carpenter, Call, Behne, & Moll, 2005).

*Corresponding author. Email: miriam.buonamente@unipa.it

Over the last years there has been an increasing interest in discovering theoretical and computational mechanisms involved in the process of intention understanding both in artificial and natural agents (Demiris, 2007). Developing such an ability would lead to an emergence of socially intelligent agents able to interact with, and learn from their peers (Breazeal, 2004). However, endowing robots with a theory of mind still remains an ambitious goal despite decades of research in the fields of human–robot interaction, learning and cooperation (Argall, Chernova, Veloso, & Browning, 2009; Chella, Dindo, & Infantino, 2007; Dindo & Schillaci, 2010; Dindo, Zambuto, & Pezzulo, 2011). For a thorough review of computational and neuronal mechanisms underpinning the processes of intention inference, see Pezzulo, Candidi, Dindo, & Barca (2013).

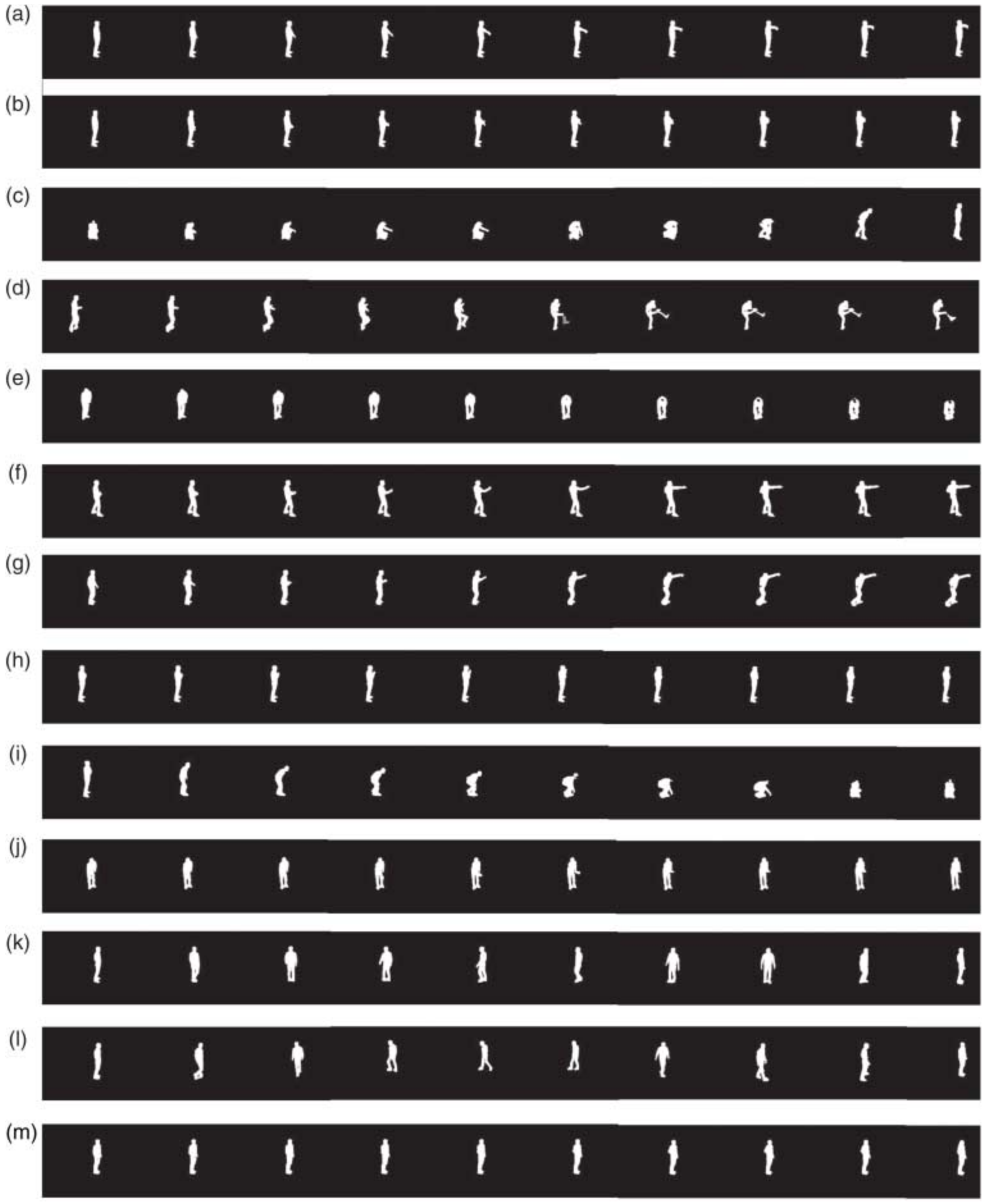


Figure 1. Prototypical postures of 13 different actions in our dataset: check watch, cross arms, get up, kick, pick up, point, punch, scratch head, sit down, throw, turn around, walk, and wave hand.

In this paper, we propose a *biologically inspired* model for the problems of action recognition and action simulation, seen as necessary prerequisites towards a fully fledged intention recognition abilities in particular, and mindreading abilities in general. Our goal is not to compete with state-of-the-art techniques for vision-based action recognition (for an updated overview of related techniques, see Poppe, 2010; Weinland, Ronfard, & Boyer, 2011), but rather to investigate self-organising principles leading to the emergence of sophisticated social abilities, and the work presented here is a step in that direction. We adopt the associative self-organising map (A-SOM) (Johnsson, Balkenius, & Hesslow, 2009), a variation of self-organising map (SOM) (Kohonen, 1988), to internally represent motion patterns and to use these patterns to recognise others' behaviour. In addition, once trained, the model is able to internally simulate the likely continuation of a partially seen action. In brief, A-SOMs remember perceptual sequences by associating current network activity with their own earlier activity. It is this ability that gives an A-SOM the ability to carry out sequence completion of perceptual activity over time.

We have tested the ability of our approach to recognise and simulate observed actions on two different datasets. The first dataset is constructed by subtracting consecutive raw black and white images depicting people performing their every day activities. In this way, we focused the attention on the dynamic part of the scene. These images were taken from the "INRIA 4D repository[1]", a publicly available dataset of movies representing 13 common actions: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up, and throw (see Figure 1). The second dataset is composed of images depicting human skeleton models captured via the Kinect sensor. The images chosen for the experiments were taken from the benchmark "MSRAction3D dataset" (Li, Zhang, & Liu, 2010).[2] The repository contains 20 action types: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up, and throw. We compared the results of two topologically different A-SOMs on the same dataset.

This paper is organised as follows: a short presentation of the A-SOM network is given in Section 2; Section 3 presents the experiments for evaluating the model; conclusions are outlined in Section 4.

2. Associative self-organising map

An A-SOM is an extension of the well-known SOM (Kohonen, 1988) that learns to associate its activity with the activity of other neural networks, or with its own delayed activity. It can be imagined as a SOM with additional (possibly delayed) ancillary inputs (see Figure 2).

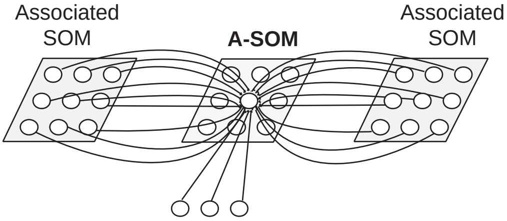


Figure 2. An A-SOM network connected with two other SOM networks. They provide the ancillary input to the main A-SOM (see the main text for more details).

By recurrently connecting the A-SOM to itself it becomes able to complete perceptual sequences over time. Many simulations prove that the A-SOM, given an initial input, can continue to elicit the likely following activity in the nearest future even though no further input is received (Johnsson, Gil, Balkenius, & Hesslow, 2010; Johnsson, Mendez, Hesslow, & Balkenius, 2011).

The A-SOM consists of an $I \times J$ grid of neurons with a fixed number of neurons and a fixed topology. Each neuron n_{ij} is associated with $r+1$ weight vectors $w^a_{ij} \in R^n$ and $w^1_{ij} \in R^{m_1}$, $w^2_{ij} \in R^{m_2}, \ldots, w^r_{ij} \in R^{m_r}$. All the elements of all the weight vectors are initialised by real numbers randomly selected from a uniform distribution between 0 and 1, after which all the weight vectors are normalised, i.e. turned into unit vectors.

At time t each neuron n_{ij} receives $r+1$ input vectors $x^a(t) \in R^n$ and $x^1(t-d_1) \in R^{m_1}$, $x^2(t-d_2) \in R^{m_2}, \ldots, x^r(t-d_r) \in R^{m_r}$ where d_p is the time delay for input vector x^p, $p = 1, 2, \ldots, r$.

The main net input s_{ij} is calculated using the standard cosine metric

$$s_{ij}(t) = \frac{x^a(t) \cdot w^a_{ij}(t)}{\|x^a(t)\| \|w^a_{ij}(t)\|}. \tag{1}$$

The activity in the neuron n_{ij} is given by

$$y_{ij} = [y^a_{ij}(t) + y^1_{ij}(t) + y^2_{ij}(t) + \cdots + y^r_{ij}(t)]/(r+1), \tag{2}$$

where the main activity y^a_{ij} is calculated by using the softmax function

$$y^a_{ij}(t) = \frac{(s_{ij}(t))^m}{\max_{ij}(s_{ij}(t))^m}, \tag{3}$$

where m is the softmax exponent and $0 \leq i \leq I, 0 \leq j \leq J, i, j \in N$. The softmax function acts as a normalisation factor and increases the contrast between highly activated areas and less activated areas (Bishop, 1995).

The ancillary activity $y^p_{ij}(t)$, $p = 1, 2, \ldots, r$, is calculated by again using the standard cosine metric

$$y^p_{ij}(t) = \frac{x^p(t-d_p) \cdot w^p_{ij}(t)}{\|x^p(t-d_p)\| \|w^p_{ij}(t)\|}. \tag{4}$$

The neuron c with the strongest main activity is selected

$$c = \text{argmax}_{ij} y_{ij}(t). \tag{5}$$

The weights w^a_{ijk} are adapted by

$$w^a_{ijk}(t+1) = w^a_{ijk}(t) + \alpha(t) G_{ijc}(t)[x^a_k(t) - w^a_{ijk}(t)], \tag{6}$$

where $0 \leq \alpha(t) \leq 1$ is the adaptation strength with $\alpha(t) \to 0$ when $t \to \infty$. The neighbourhood function $G_{ijc}(t) = e^{-\|r_c - r_{ij}\|/2\sigma^2(t)}$ is a Gaussian function decreasing with time, and $r_c \in R^2$ and $r_{ij} \in R^2$ are location vectors of neurons c and n_{ij}, respectively.

The weights w^p_{ijl}, $p = 1, 2, \ldots, r$, are adapted by

$$w^p_{ijl}(t+1) = w^p_{ijl}(t) + \beta x^p_l(t-d_p)[y^a_{ij}(t) - y^p_{ij}(t)], \tag{7}$$

where β is the adaptation strength.

All weights $w^a_{ijk}(t)$ and $w^p_{ijl}(t)$ are normalised after each adaptation.

3. Experiments

To validate our approach, we performed experiments using two different A-SOM configurations (either with one or two sets of recurrently connected time delayed ancillary connections) and two different datasets (images belonging to the INRIA 4D repository and human skeleton models captured via the Kinect sensor; Li et al., 2010).

The aim of the experiments was to verify whether the bio-inspired model was able to discriminate and simulate actions. We performed two kinds of experiments, by using two different A-SOM configurations and the two kinds of input datasets. The first experiment aimed at testing the representational capabilities of the A-SOM in an action discrimination task and the latter aimed at testing the A-SOM's ability to simulate the continuation of the initial part of an action.

Neurobiological studies argue that the human brain can perceive actions by observing only the human body poses, called postures, during action execution (Giese & Poggio, 2003). Thus, actions can be described as sequences of consecutive human body poses, in terms of human body silhouettes (Gkalelis, Tefas, & Pitas, 2008; Gorelick, Blank, Shechtman, Irani, & Basri, 2007; Iosifidis, Tefas, & Pitas, 2012).

The implementation of all code for the experiments presented in this paper was done in C++ using the neural modelling framework "Ikaros" (Balkenius, Morén, Johansson, & Johnsson, 2010). Next sections provide a detailed description of the experiments performed.

3.1. *Action discrimination*

In the action discrimination tests we used two different datasets. The first dataset was derived from the INRIA 4D repository, an image repository consisting of 959 postural images reproducing 13 different actions. Since we wanted agent to be able to discriminate one action at a time, we split the original movie into 13 different movies: one movie for each action (see Figure 1).

To reduce the computational load, the original postural images were reduced in number and in size. We reduced the numbers of images for each movie to 10 and as Figure 3 shows, the reduction does not affect the quality of the action reproduction. Different operations of image size reduction were done according to the experiment executed. For the action discrimination experiment, we needed images depicting the person who was performing actions. To this end, these images were centred at the person's centre of mass and bounding boxes of a size equal to the maximum bounding box enclosing the person's body were extracted. We cut the images by using the identified boundary box including only the person performing the action.

To simulate an attentive process in which the human eye observes and follows only salient parts of a scene, we subtracted consecutive images from the dataset focusing on the dynamic part of the action only. Each action was thus represented by a sequence of nine frames representing the parts of the postures that change, i.e. only the part of the human body involved in the action, see Figure 4. The reduction of the number of frames for each movie did not affect the overall quality of the video (see, for instance, the "check watch" action in Figure 5).

To further improve system performances, each frame was resized to a $N_H \times N_W$ matrix. Binary posture images were represented as matrices and these matrices were vectorised to produce posture vectors $p \in R^D$, where $D = N_H \times N_W$. Thus each posture image was represented by a

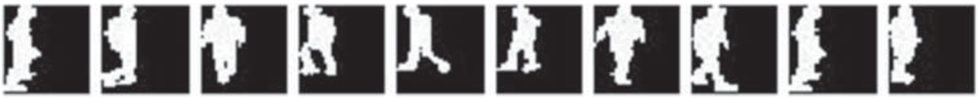


Figure 3. The walk action movie created with a reduced number of images.

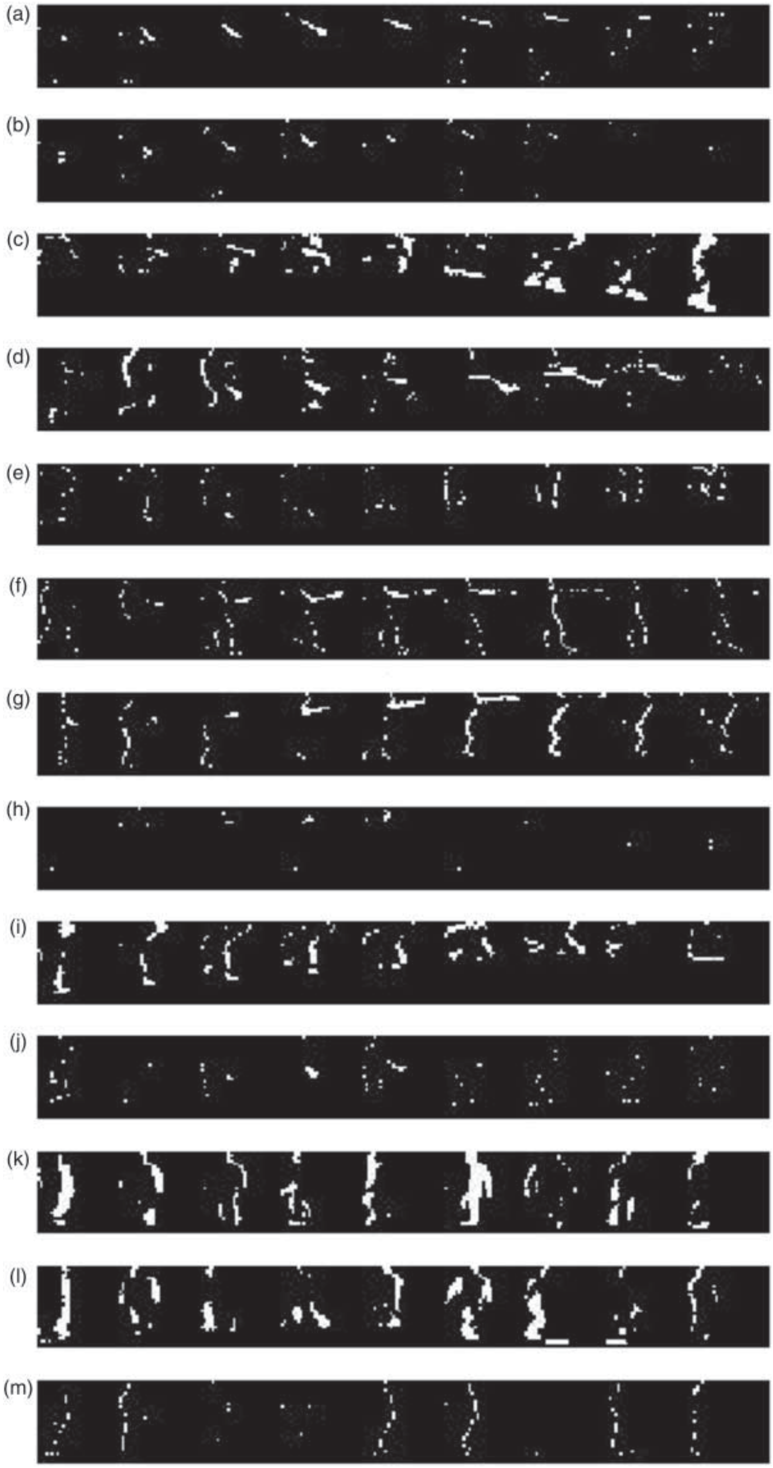


Figure 4. The parts of the human body involved in the movement of each action. Each sequence was obtained by subtracting consecutive images in each action sequence. The actions are (a) check watch, (b) cross arm, (c) get up, (d) kick, (e) pick up, (f) point, (g) punch, (h) scratch head, (i) sit down, (j) throw, (k) turn around, (l) walk, and (m) wave hand.

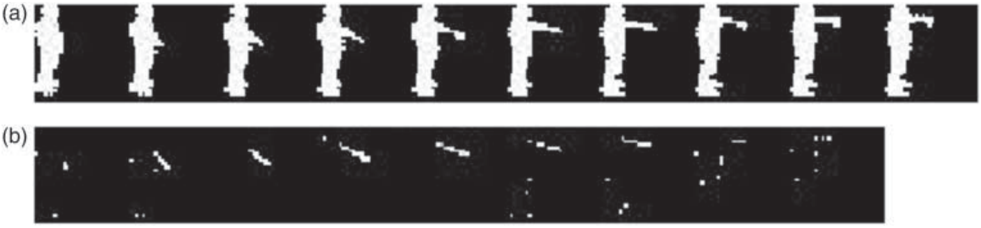


Figure 5. (a) The sequence of images depicting the check watch action and (b) the sequence of images obtained by subtracting consecutive images of the check watch action.

posture vector p. Finally, each action, consisting of nine frames, was represented by a sequence of posture vectors $p_i \in R^D$, $i = 1 \dots 9$, used as main inputs to the A-SOM network.[3]

The posture vectors were used to create the training set required to train the A-SOM. Our final training set for the action discrimination experiment was composed of 117 vectors representing the 13 different actions. The A-SOM network was trained by feeding it with sequences of vectors corresponding to the actions in a random way for 20,124 iterations, which corresponds to receive 2236 randomly selected actions from the training set during the training phase.

The goal of the action discrimination experiment was to verify if the A-SOM was able to discriminate actions. We evaluated the A-SOM by setting up a system consisting of one A-SOM connected to itself, see the leftmost part of Figure 6. To this end, 13 sequences, each containing 9 posture vectors (representing the binary images that form the videos) were constructed as explained above.

We tested the fully trained A-SOM by providing it with one action sequence at a time while recording the most activated neuron for each postural vector. The coordinates of these neurons were plotted in a diagram that shows how they are located in the neural network, as shown in Figure 7. Each picture shows the grid of neurons forming the A-SOM and illustrates the sequence of the most activated neurons (from now on called the centre of activity and represented by dots) during the corresponding action. The centres of activity are connected to each other through arrows describing the temporal evolution of the represented action, starting and final points are represented by dots with bigger dimensions. These diagrams depict the motion pattern for each action and allow us to evaluate if the A-SOM can discriminate the binary images that form each action. What we expected was to have a different activity pattern in the A-SOM during the action for each action, and that the centre of activity should be different for different posture vectors. The classification procedure for these activity patterns is easily implemented through stochastic state machines (e.g. Markov processes or similar machine learning methods; Murphy, 2012).

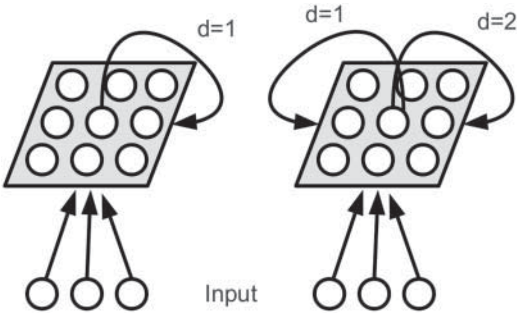


Figure 6. Two A-SOM configurations: the leftmost represents the configuration with one set of ancillary connections with a time delay of 1; the rightmost represents the configuration with two sets of time delayed ancillary connections, the time delay is equal to 1 for the first set of connections and equal to 2 for the second set.

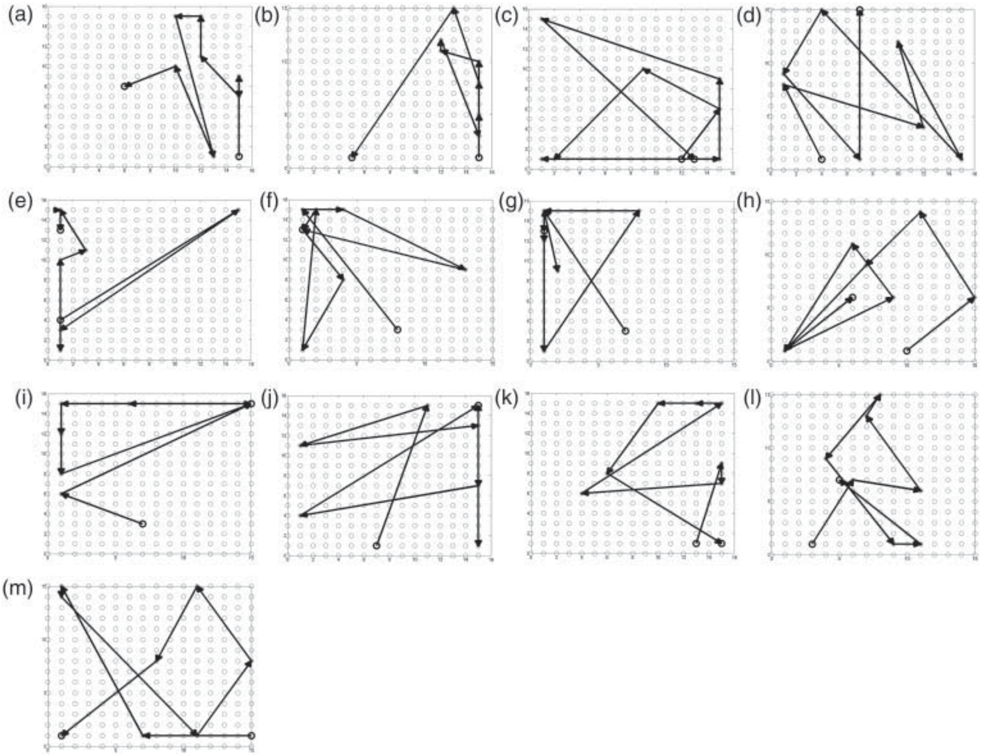


Figure 7. Activity patterns for the 13 actions: (a) check watch, (b) cross arms, (c) scratch head, (d) sit down, (e) get up, (f) turn around, (g) walk, (h) wave hand, (i) punch, (j) kick, (k) point, (l) pick up, and (m) throw. The points in the diagram represent the actions centre of activity and the lines indicate the action evolution over time. Actions made of similar posture vectors present few centres of activity; whereas movies made of posture vectors with different characteristics present several centres of activity. The diagrams give an indication about the ability of the A-SOM to create topology preserving maps in which similar postures are represented close to each other.

In the presented experiment, the A-SOM elicited different centres of activity for different posture vectors, thus creating different activity patterns during different actions, demonstrating its ability to discriminate actions properly. The activity patterns elicited by individual posture vectors, approximated by their centres of activity, yield information about the ability of the A-SOM to discriminate between individual postures that compose each action. Since the A-SOM elicited the same centres of activity for similar posture vectors, action movies made of similar images had few centres of activity. Actions composed of images with different characteristics, presented several centres of activity, one for each different image. It is possible to see, for example, in the "punch" movie, Figure 1(g), that the pattern presents only seven centres of activity, see Figure 7(i), whereas in the "check watch" movie, Figure 1(a), the pattern presents nine different centres of activity, see Figure 7(a). The plotted diagrams, furthermore, show that the A-SOM can create topology-preserving maps in which similar binary postures are located close to each other. For example, the motion pattern in Figure 7(b) of the action "cross arm" in Figure 1(b), the binary images depicting the person moving the arms are represented close each other. We can easily argue that the A-SOM is able to discriminate body postures representing actions. This is also proven by the results of the generalisation experiment, see Figure 1, where the recognition percentage is equal to 62%.

Through the introduction of depth cameras, it is possible to simplify the capturing technique of 3D human motion, improving at the same time the quality of the captured movement. Further improvement has been obtained through techniques for extraction of 3D joint positions of

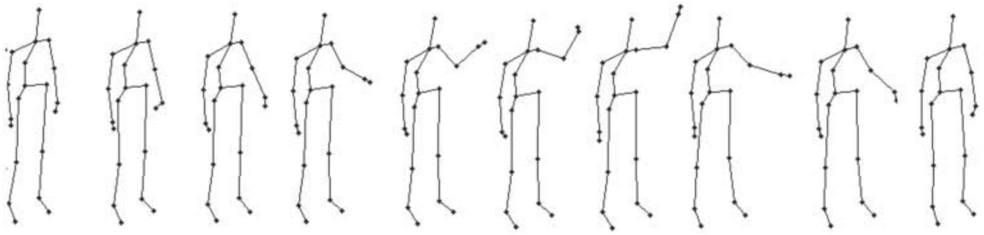


Figure 8. Skeleton models made of 20 joints depicting “high arm wave” action.

the human skeleton from the captured depth sequences, see Figure 8 where a skeleton reproducing “high arm wave” action is depicted. Sequences of joint positions are employed to represent motion of human body.

To verify whether the A-SOM was able to discriminate actions based on input sequences of skeleton models representing those actions, we carried out another action discrimination experiment. To this end, a dataset made of images depicting skeletons was chosen. This dataset was used in a discrimination experiment with the A-SOM with two sets of time delayed recurrent ancillary connections, see the rightmost part of Figure 6.

The dataset was created using a Kinect sensor and was composed of joint positions representing human skeleton models. It is made of sequences of skeleton models representing 20 types of actions extracted from the captured corresponding depth sequences by using the real-time skeleton tracking algorithm proposed in Shotton et al. (2013). Each skeleton model is made of 20 joint positions in 3D space.

By using the same procedure used for the experiment discussed above, we tested the fully trained A-SOM with one sequence a time and the centres of activity generated for each skeleton of the sequence were recorded and plotted in a diagram, see Figure 9. The expectation was that the A-SOM would generate different evolving activity patterns for different actions, i.e. the A-SOM would discriminate between different actions.

As can be seen in Figure 9, the A-SOM elicited different evolving patterns of activity for different actions. It is evident that it is possible to discriminate different evolving patterns of activity in the A-SOM. We observe that actions made of similar skeletons present few centres of activity, whereas actions made of more differing skeletons present more centres of activity. As in the previous experiment, described above, we observe that the A-SOM maintains the ability to create topological maps in which similar skeletons are represented close to each other. Also in this case the A-SOM gave evidence of its ability to learn discriminable representations of actions.

The results obtained through these experiments allowed us to look into the generalisation ability of the A-SOM. The generalisation is the network’s ability to recognise inputs it has never seen before. The idea is that if the A-SOM is able to recognise images as similar by generating close or equal centres of activity, then it will also be able to recognise an image it has never encountered before if this is similar to a known image. We checked if similar images in the dataset based on the INRIA 4D repository had the same centres of activity and if similar centres of activity corresponded to similar images. The results show that the A-SOM generated very close or equal values for very similar images, see Figure 10. Actions like “turn around”, “walk”, and “get up” present some frames very similar to each other and for such frames the A-SOM generates the same centres of activity. This ability is validated through the selection of some centres of activity and the verification that they correspond to similar images. “Check watch”, “get up”, “point”, and “kick” actions include in their sequences frames depicting the movement of the arm that can be attributed to all of them. For these frames the A-SOM elicits the same

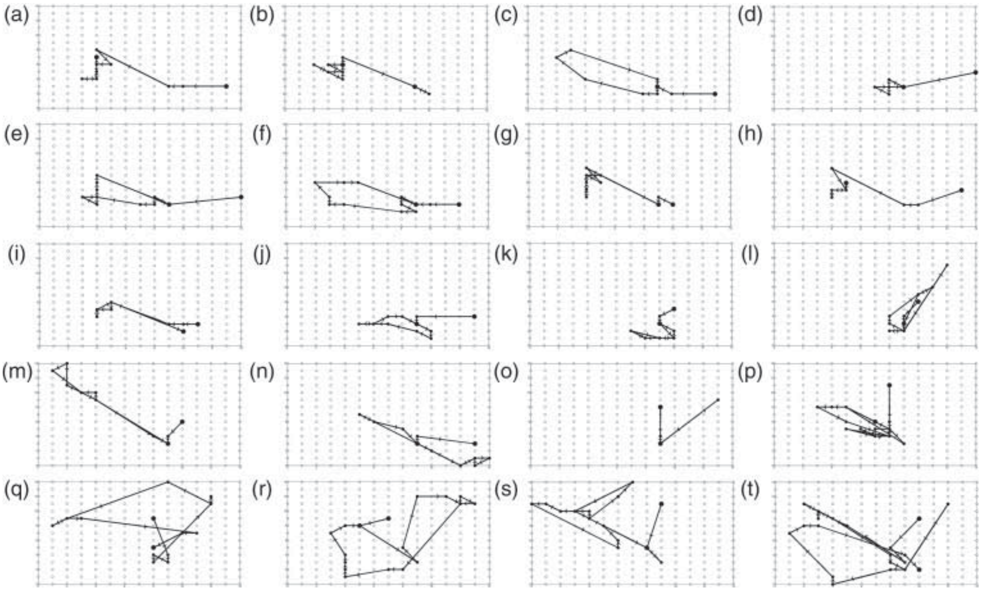
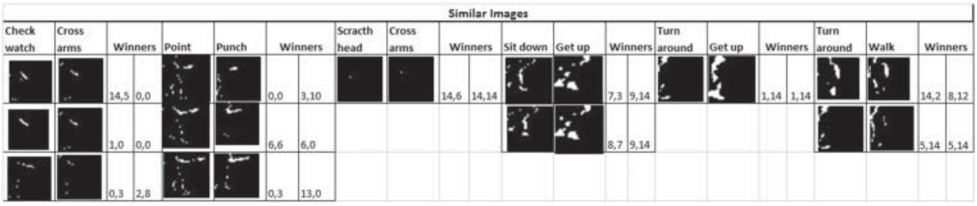


Figure 9. Motion patterns for 20 actions, clockwise from the top left corner: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, sideboxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up, and throw. The points in the diagram represent the actions centre of activity and the lines indicate the action evolution over time. Actions made of similar skeletons present few centre of activity; whereas actions made of different skeletons present several centres of activity. The diagrams give indication about the ability of A-SOM to create topological maps in which similar skeletons are located close to each other.

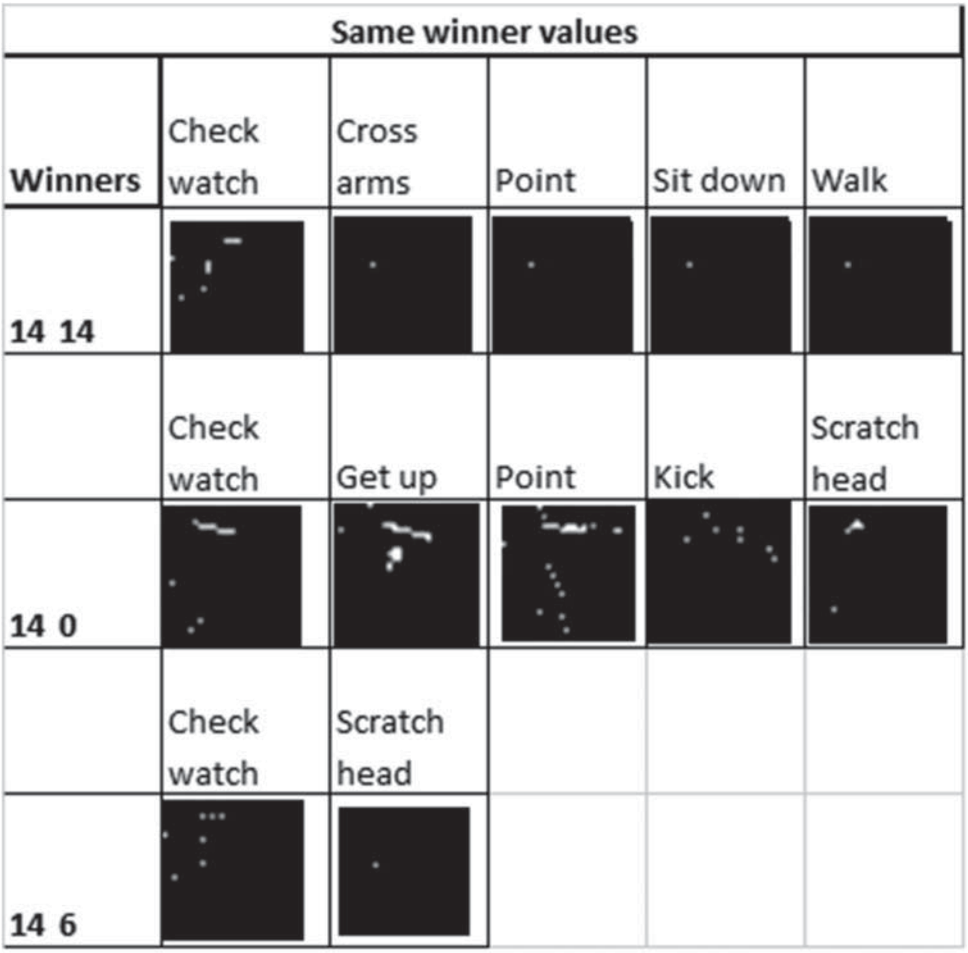


Similar Images																							
Check watch	Cross arms	Winners		Point	Punch	Winners		Scracth head	Cross arms	Winners		Sit down	Get up	Winners		Turn around	Get up	Winners		Turn around	Walk	Winners	
		14,5	0,0			0,0	3,10			14,6	14,14			7,3	9,14			1,14	1,14			14,2	8,12
		1,0	0,0			6,6	6,0							8,7	9,14							5,14	5,14
		0,3	2,8			0,3	13,0																

Figure 10. Similar images have similar centres of activity. The A-SOM elicits similar or equal centres of activity for images that are similar.

centre of activity, see Figure 11. The results presented here support the belief that our system is also able to generalise.

In order to validate our hypotheses, we gave the network an input never encountered before. What we expected was that the A-SOM generated equal or at least similar motion patterns for the same actions, even if those were performed by different actors. We selected a new video depicting another actor performing the same 13 actions. By using the same procedure used for the experiment discussed above, we tested the fully trained A-SOM with one sequence at a time and the centres of activity generated for each frame of the sequence were recorded and compared with those elicited during the previous experiment. The actors used in this paper were “Andreas” to train the A-SOM and to test its recognition ability, and “Hedlena” to test the A-SOM generalisation ability. The comparison between centres of activity elicited by Hedlena and those elicited by Andreas was performed and the results were collected and shown in Table 1. The results show an average 62% recognition rate on unknown sequences. We expect this percentage



Same winner values

Winners	Check watch	Cross arms	Point	Sit down	Walk
14 14					
	Check watch	Get up	Point	Kick	Scratch head
14 0					
	Check watch	Scratch head			
14 6					

Figure 11. Images with the same centres of activity. The frames present the same features which elicits the same centres of activity in the A-SOM.

to be higher by adopting more statistically robust classifiers trained on different patterns in the topological space (e.g. hidden Markov model, support vector machine, and alike).

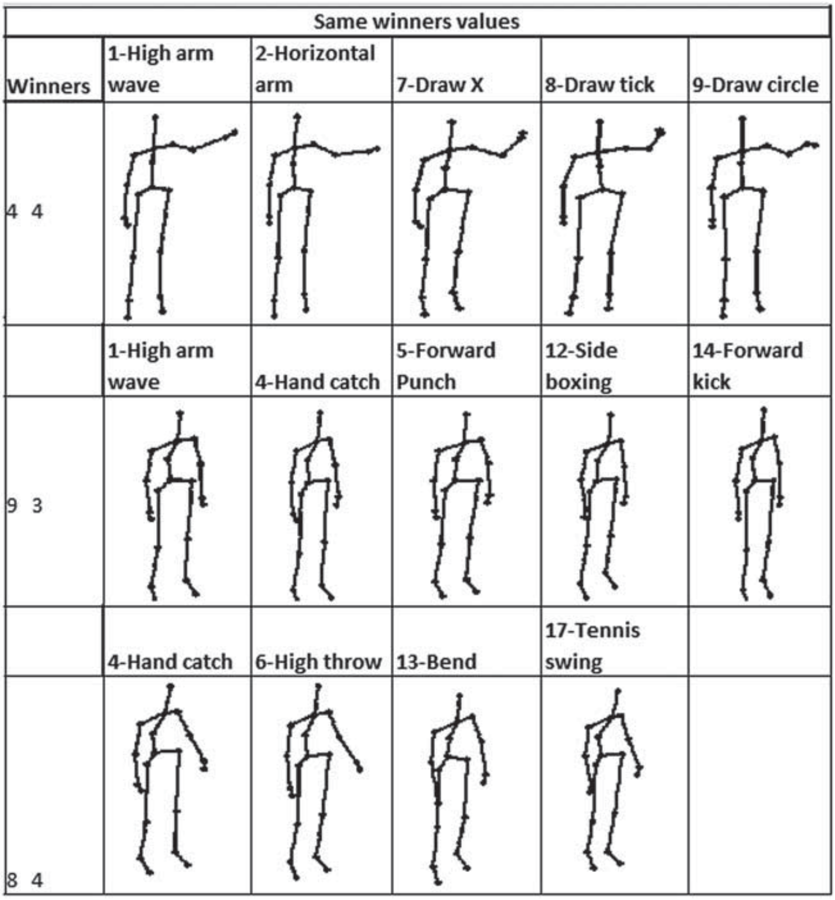
As shown in the figure, however, we still face the problem of ambiguity. “Check watch” action, for example, can be confused with the “cross arm” or “wave hand” actions; indeed, the A-SOM recognises all the three actions with the same recognition rate (33%). This happens because of the similarity in the original movement pattern. In particular, the initial part of the rising arm movement is similar for all the three actions. Only “cross arm” action differs because its movement involves both arms and its recognition rate is higher than in other samples.

Other factors, like scaling in image size, changes in the speed of the action, change in orientation of the point of view, changes in time of action execution and others can contribute to wrong results and to the ambiguity of recognition. Our system is based on posture vectors and it recognises vectors that are similar to each other. It can happen that same actions – captured by different cameras or performed by different actors – can produce radically different posture vectors once vectorised. The use of more advanced local descriptors in the preprocessing stage – e.g. scale invariant feature transform (Lowe, 1999) – would yield to better recognition rates and is currently under investigation.

Table 1. Recognition and generalisation results.

	Andreas													
Hedlena	Check watch	Cross arm	Scracth head	Sit down	Get up	Turn around	Walk	Wave hand	Punch	Kick	Point	Pick up	Throw	%Correctenss
Check watch	0.33	0.33		0.11				0.33		0.11	0.22		0.11	correct
Cross arm	0.11	0.56	0.11		0.11	0.11		0.11			0.11		0.11	correct
Scracth head		0.22	0.33	0.11	0.11	0.11	0.11	0.22						correct
Sit down		0.22		0.22		0.11	0.11		0.11		0.11		0.11	correct
Get up		0.11		0.11	0.44	0.11	0.11					0.11		correct
Turn around					0.56	0.11	0.33	0.11	0.11		0.11		0.11	
Walk						0.56	0.33		0.11				0.11	
Wave hand	0.11	0.33	0.11		0.11	0.11	0.22	0.44		0.11				correct
Punch				0.11		0.11	0.11	0.44		0.11		0.11		
Kick		0.11		0.22		0.33		0.11	0.11		0.11		0.11	
Point	0.22	0.22	0.11			0.11	0.11			0.22	0.33	0.22	0.11	correct
Pick up	0.22		0.11	0.11	0.11	0.11	0.11		0.11	0.11	0.11	0.22	0.11	correct
Throw		0.11		0.22	0.22	0.11	0.11	0.11				0.33	0.11	
%Correctness														0.62

Notes: The table was constructed by calculating the Euclidean distance between one frame of one action performed by Hedlena and the same frame of all the actions performed by Andreas, the minimum distance indicates that the two compared frames activate the same centres of activity. Then the percentage of correct frames by the total number of frames per action was calculated and reported in the table.



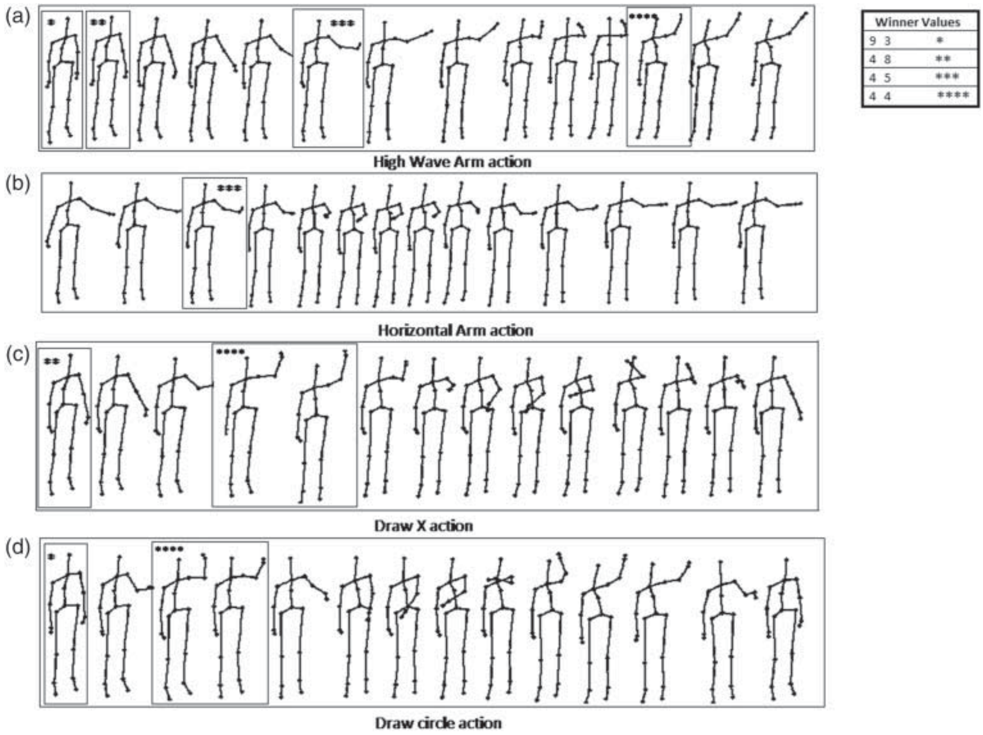
Same winners values					
Winners	1-High arm wave	2-Horizontal arm	7-Draw X	8-Draw tick	9-Draw circle
4 4					
	1-High arm wave	4-Hand catch	5-Forward Punch	12-Side boxing	14-Forward kick
9 3					
	4-Hand catch	6-High throw	13-Bend	17-Tennis swing	
8 4					

Figure 12. Skeleton models with the same centres of activity. The frames present the same features which lead the A-SOM to elicit the same centre of activity.

We also wanted to demonstrate if the A-SOM was able to generalise when it was fed by skeleton inputs. Its ability to recognise skeletons that are similar to each other and consequently its ability to recognise a new input if this is similar to another already known. To this end, we checked if the A-SOM was able to elicit similar centres of activity for similar skeletons and if similar skeletons had the same centres of activity.

As Figure 12 shows, the A-SOM generated the same centres of activity for similar skeletons. Actions like “high arm wave”, “horizontal arm”, “draw x”, “draw tick”, and “draw circle” include some skeleton models very similar to each other and for such frames the A-SOM elicited the same centres of activity.

Unlike the results obtained with the same experiment conducted on the posture vectors described above, where some images similar to each other could have different centres of activity, see Figure 10, in this experiment conducted on skeleton postures the results show that all the skeletons that are similar to each other are always represented very close to each other in the A-SOM, see Figure 13. We argue that the A-SOM can easily recognise an input never encountered before, if it is similar to another already known. The analysis based on comparison between centres of activity fired by known skeletons and those fired by never seen skeletons can be carried out with very encouraging results.



Winner Values	
9 3	*
4 8	**
4 5	***
4 4	****

Figure 13. Similar skeleton models elicit the same centre of activity. The representations of similar skeleton models are closely located in the A-SOM.

3.2. *Action simulation*

In this experiment the objective was to verify whether our A-SOM-based architecture was capable of simulating the most likely continuation of a partially seen action. We fed the trained A-SOM with incomplete input patterns and expected it to continue to elicit activity patterns corresponding to the remaining part of the action. We used two different A-SOM configurations for the action simulation test in order to find the best solution to improve the performance of the model. Thus, we set up two systems: one was the same we used for the action recognition experiment (see the leftmost part of Figure 6), the other was made of the A-SOM connected to itself with two sets of time delayed ancillary connections, see the rightmost part of Figure 6. The time delay for the two ancillary connections of the second configuration was set equal to 1 for the first set of connections and equal to 2 for the second set of connections. The second configuration aimed at improving the precision of the simulation by allowing the A-SOM to associate its activity with not only the activity elicited in the previous iteration but also with the activity elicited in the iteration before that. The experiment was conducted by using both configurations.

In the simulation experiments with the two A-SOM configurations we used a dataset derived from the INRIA 4D repository-based dataset described in the previous section. This dataset was created by subtracting consecutive images in the original action sequences to get a dataset focused on the dynamical parts of the scene. In this derived dataset each of the 13 actions in the original dataset is represented by a sequence of 9 frames representing the parts of the postures that change, i.e. only the part of the human body involved in the action, see Figure 4. In this case the training phase corresponded to 10,000 randomly selected actions.

As in the discrimination experiment with the original dataset derived from the INRIA 4D repository described in the previous section, we used a boundary box to cut the images to include only the part of the body involved in the action. Thus anything not involved in the movement was eliminated. This procedure can be seen as simulating an attentive process in which the human eye observes and follows only the salient parts of the action.

The obtained sequences of frames representing the dynamic parts of the scene at each instance in the actions were shrunk to $N_H \times N_W$ matrices and were vectorised to produce posture vectors $p \in R^D$, where $D = N_H \times N_W$. In this way every action, consisting of nine frames was represented by a sequence of vectors $p_i \in R^D$, $i = 1 \ldots 9$. In the simulation experiments $N_H = 30$, $N_W = 30$ and $N_H = 50$, $N_W = 50$ have been used. The obtained vector sequences were used as input to the A-SOM.

These preprocessing operations reduced the number of frames for each movie to nine, without affecting the overall quality of the video (e.g. the "check watch" action in Figure 5).

We tested the fully trained A-SOM configurations in the following way. First, all nine vectors of each sequence were received by the A-SOMs to record the sequences of activity patterns (approximated by the most activated neurons) that represent each action. Then we tested the A-SOMs by providing the initial parts of the action sequences while observing if they were

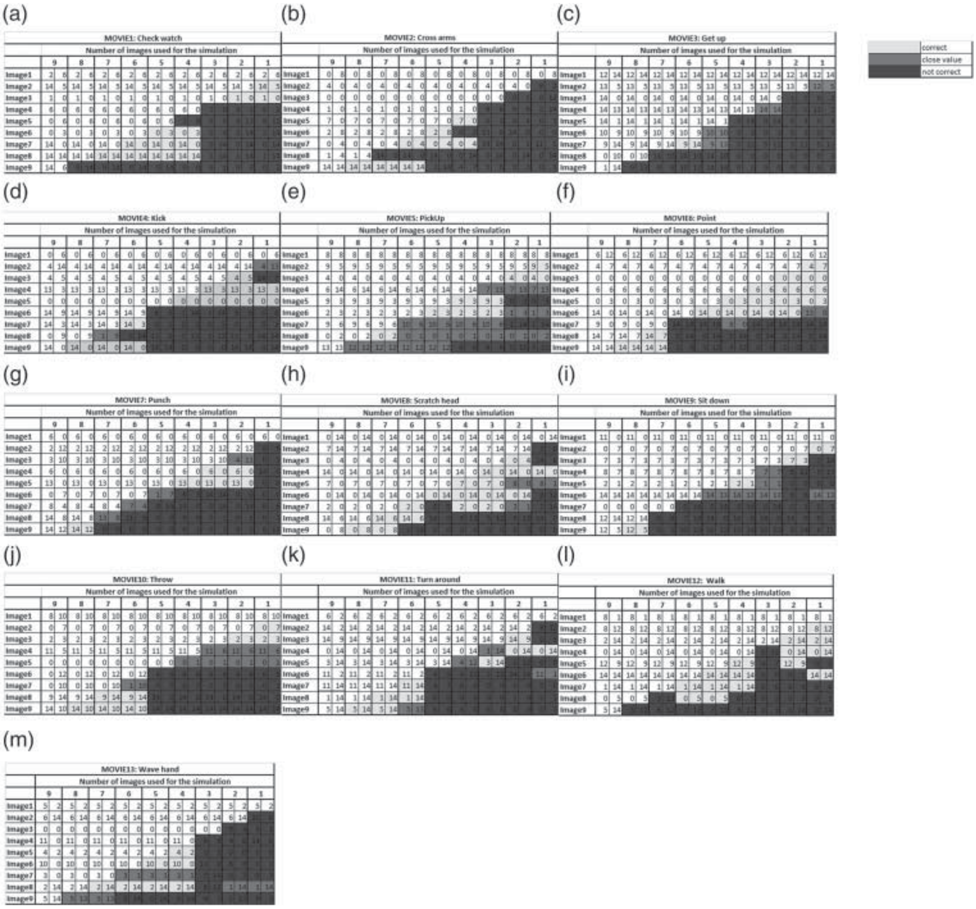
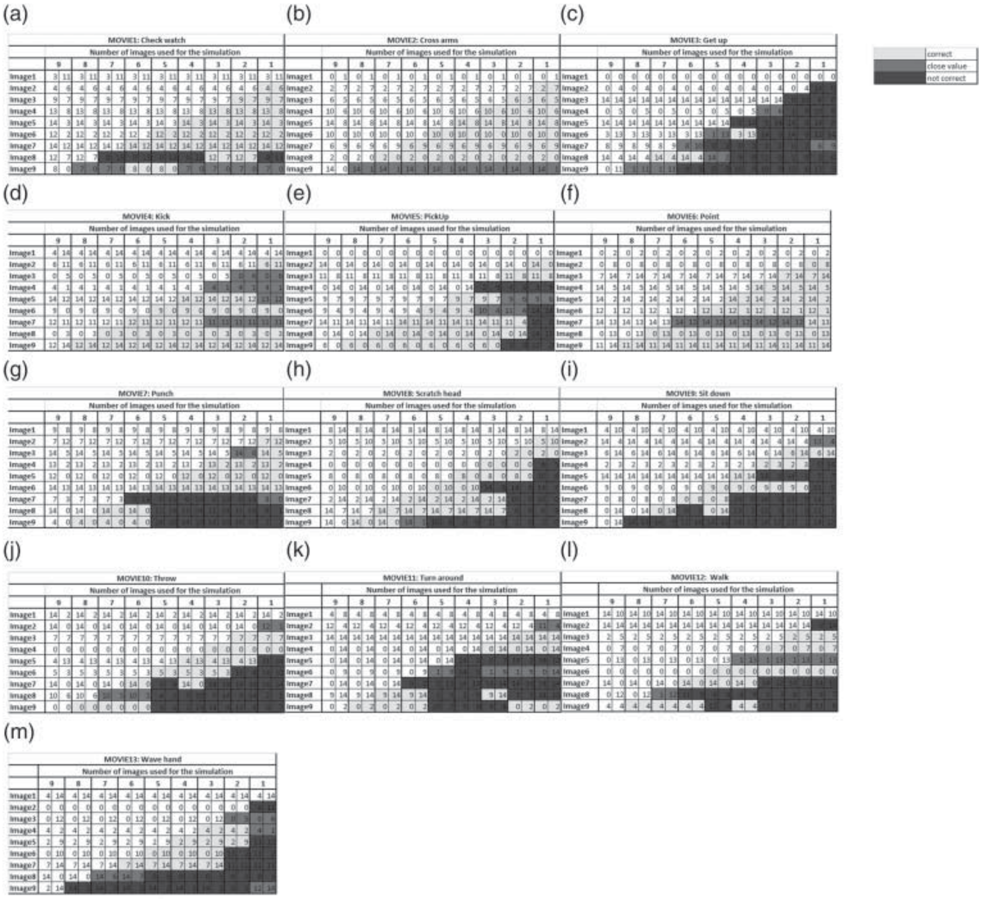

(a)
MOVIE1: Check watch
(b)
MOVIE2: Cross arms
(c)
MOVIE3: Get up
correct
close value
not correct
(d)
MOVIE4: Kick
(e)
MOVIE5: PickUp
(f)
MOVIE6: Point
(g)
MOVIE7: Punch
(h)
MOVIE8: Scratch head
(i)
MOVIE9: Sit down
(j)
MOVIE10: Throw
(k)
MOVIE11: Turn around
(l)
MOVIE12: Walk
(m)
MOVIE13: Wave hand
Number of images used for the simulation
9 8 7 6 5 4 3 2 1
Image1 Image2 Image3 Image4 Image5 Image6 Image7 Image8 Image9

Figure 14. Simulation results for the configuration made of one A-SOM connected to itself with one set of ancillary connections: the tables show the ability of the A-SOM to continue the likely continuation of an observed behaviour. A light gray colour indicates that the A-SOM is able to simulate, a dark gray colour indicates that A-SOM simulates an activity pattern similar to the right one, and a black colour indicates that the A-SOM was unsuccessful at simulating the right activity pattern. The system needs between four and nine inputs to internally simulate the rest of the sequence.

able to complete the remaining parts of the activity sequences representing the corresponding actions by internal simulation, i.e. without receiving any input. This was done repeatedly for all possible lengths of the initial parts of the sequences, i.e. for initial parts of length = 1, length = 2, . . . , length = 8. The centres of activity approximating the activity patterns were collected in tables (Figures 14 and 15), and colour coding was used to indicate the ability (or the inability) of the A-SOM to simulate the continuation of the action. A light gray colour indicates successful internal simulation (the most activated neurons was the same as with input); a dark gray colour indicates the internal simulation of an activity pattern similar to the right one (the most activated neurons close to the most activated neurons while receiving input); and a black colour indicates unsuccessful internal simulation (the most activated neurons not close to the most activated neurons while receiving input).

As Figure 14 shows, the ability of the configuration with one set of ancillary connections to internally simulate the continuation of the actions varies with the type of action. For actions like “sit down” and “punch”, the A-SOM needed eight images to simulate the reception of the final image in the sequence; whereas for the “walk” action, the A-SOM needed only four images to


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)
(m)
correct
close value
not correct

MOVIE1: Check watch																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	3	11	3	11	3	11	3	11	3	11	3	11	3	11	3	11	3	11
Image2	4	6	4	6	4	6	4	6	4	6	4	6	4	6	4	6	4	6
Image3	9	7	9	7	9	7	9	7	9	7	9	7	9	7	9	7	9	7
Image4	13	8	13	8	13	8	13	8	13	8	13	8	13	8	13	8	13	8
Image5	14	3	14	3	14	3	14	3	14	3	14	3	14	3	14	3	14	3
Image6	12	2	12	2	12	2	12	2	12	2	12	2	12	2	12	2	12	2
Image7	14	12	14	12	14	12	14	12	14	12	14	12	14	12	14	12	14	12
Image8	12	7	12	7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	12	7	12	7	[illegible]	[illegible]
Image9	8	0	7	0	7	0	8	0	8	0	7	0	7	0	7	[illegible]	7	0

MOVIE2: Cross arms																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Image2	2	7	2	7	2	7	2	7	2	7	2	7	2	7	2	7	2	7
Image3	6	5	6	5	6	5	6	5	6	5	6	5	6	5	6	5	6	5
Image4	10	6	10	6	10	6	10	6	10	6	10	6	10	6	10	6	10	6
Image5	14	8	14	8	14	8	14	8	14	8	14	8	14	8	14	8	14	8
Image6	10	0	10	0	10	0	10	0	10	0	10	0	10	0	10	0	10	0
Image7	6	9	6	9	6	9	6	9	6	9	6	9	6	9	6	9	6	9
Image8	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0
Image9	14	0	14	1	14	1	14	1	14	1	14	1	14	1	14	1	14	1

MOVIE3: Get up																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Image2	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	4	[illegible]	[illegible]
Image3	14	14	14	14	14	14	14	14	14	14	14	14	14	14	[illegible]	[illegible]	[illegible]	[illegible]
Image4	0	5	0	5	0	5	0	5	0	5	0	5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image5	14	14	14	14	14	14	14	14	14	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image6	3	13	3	13	3	13	3	13	[illegible]	[illegible]	3	13	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image7	8	9	8	9	8	9	8	10	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image8	14	4	14	4	14	4	14	4	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image9	0	11	[illegible]	11	[illegible]	11	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

MOVIE4: Kick																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	4	14	4	14	4	14	4	14	4	14	4	14	4	14	4	14	4	14
Image2	6	11	6	11	6	11	6	11	6	11	6	11	6	11	6	11	6	11
Image3	0	5	0	5	0	5	0	5	0	5	0	5	0	5	[illegible]	[illegible]	[illegible]	[illegible]
Image4	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1
Image5	14	12	14	12	14	12	14	12	14	12	14	12	14	12	14	12	13	12
Image6	9	0	9	0	9	0	9	0	9	0	9	0	9	0	9	0	9	0
Image7	12	11	12	11	12	11	12	11	12	11	12	11	11	11	11	11	11	11
Image8	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3
Image9	12	14	12	14	12	14	12	14	12	14	12	14	12	14	12	14	12	14

MOVIE5: PickUp																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Image2	14	0	14	0	14	0	14	0	14	0	14	0	14	0	14	0	14	0
Image3	11	8	11	8	11	8	11	8	11	8	11	8	11	8	11	8	11	8
Image4	0	14	0	14	0	14	0	14	0	14	0	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image5	9	7	9	7	9	7	9	7	9	7	9	7	9	7	9	6	9	6
Image6	9	4	9	4	9	4	9	4	9	4	9	4	10	4	11	4	[illegible]	[illegible]
Image7	14	11	14	11	14	11	14	11	14	11	14	11	14	11	11	4	[illegible]	[illegible]
Image8	0	14	0	14	0	14	0	14	0	14	0	14	0	14	0	14	[illegible]	[illegible]
Image9	6	0	6	0	6	0	6	0	6	0	6	0	6	0	[illegible]	[illegible]	[illegible]	[illegible]

MOVIE6: Point																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2
Image2	0	8	0	8	0	8	0	8	0	8	0	8	0	8	0	8	0	8
Image3	7	14	7	14	7	14	7	14	7	14	7	14	7	14	7	14	7	14
Image4	14	5	14	5	14	5	14	5	14	5	14	5	14	5	14	5	14	5
Image5	14	2	14	2	14	2	14	2	14	2	14	2	14	2	14	2	14	2
Image6	12	1	12	1	12	1	12	1	12	1	12	1	12	1	12	1	12	1
Image7	14	13	14	13	14	13	14	12	14	12	14	12	14	12	14	12	14	13
Image8	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13
Image9	11	14	11	14	11	14	11	14	11	14	11	14	11	14	11	14	11	14

MOVIE7: Punch																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	9	8	9	8	9	8	9	8	9	8	9	8	9	8	9	8	9	8
Image2	7	12	7	12	7	12	7	12	7	12	7	12	7	12	7	12	7	12
Image3	14	5	14	5	14	5	14	5	14	5	14	5	14	5	14	[illegible]	14	5
Image4	13	2	13	2	13	2	13	2	13	2	13	2	13	2	13	2	13	2
Image5	12	0	12	0	12	0	12	0	12	0	12	0	12	0	12	0	12	0
Image6	14	13	14	13	14	13	14	13	14	13	14	13	14	13	14	13	14	13
Image7	7	3	7	3	7	3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3	0
Image8	14	0	14	0	14	0	14	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image9	4	0	4	0	4	0	4	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

MOVIE8: Scratch head																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	8	14	8	14	8	14	8	14	8	14	8	14	8	14	8	14	8	14
Image2	5	10	5	10	5	10	5	10	5	10	5	10	5	10	5	10	5	10
Image3	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0
Image4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]
Image5	8	0	8	0	8	0	8	0	8	0	8	0	8	0	8	0	[illegible]	[illegible]
Image6	0	10	0	10	0	10	0	10	0	10	0	10	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image7	2	14	2	14	2	14	2	14	2	14	2	14	2	14	[illegible]	[illegible]	[illegible]	[illegible]
Image8	14	7	14	7	14	7	14	7	14	7	14	7	14	7	[illegible]	[illegible]	[illegible]	[illegible]
Image9	14	0	14	0	14	0	14	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

MOVIE9: Sit down																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	4	10	4	10	4	10	4	10	4	10	4	10	4	10	4	10	4	10
Image2	14	4	14	4	14	4	14	4	14	4	14	4	14	4	14	4	13	4
Image3	6	14	6	14	6	14	6	14	6	14	6	14	6	14	6	14	6	14
Image4	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	[illegible]	[illegible]
Image5	14	14	14	14	14	14	14	14	14	14	14	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image6	9	0	9	0	9	0	9	0	9	0	9	0	9	0	9	0	[illegible]	[illegible]
Image7	0	8	0	8	0	8	0	8	0	8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image8	0	14	0	14	0	14	[illegible]	[illegible]	0	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image9	0	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

MOVIE10: Throw																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	14	2	14	2	14	2	14	2	14	2	14	2	14	2	14	2	14	2
Image2	14	0	14	0	14	0	14	0	14	0	14	0	14	0	14	0	12	[illegible]
Image3	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Image4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Image5	4	13	4	13	4	13	4	13	4	13	4	13	4	13	4	13	[illegible]	[illegible]
Image6	5	3	5	3	5	3	5	3	5	3	5	3	5	3	[illegible]	[illegible]	[illegible]	[illegible]
Image7	14	0	14	0	14	0	14	0	[illegible]	[illegible]	14	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image8	10	6	10	6	10	[illegible]	10	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image9	0	0	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

MOVIE11: Turn around																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	4	8	4	8	4	8	4	8	4	8	4	8	4	8	4	8	4	8
Image2	12	4	12	4	12	4	12	4	12	4	12	4	12	4	12	4	11	4
Image3	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
Image4	0	14	0	14	0	14	0	14	0	14	0	14	0	14	0	14	0	14
Image5	0	14	0	14	0	14	0	14	0	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image6	0	9	0	9	0	9	0	9	1	9	[illegible]	[illegible]	1	9	1	9	0	14
Image7	0	14	0	14	0	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image8	9	14	9	14	9	14	9	14	[illegible]	[illegible]	[illegible]	[illegible]	9	14	[illegible]	[illegible]	[illegible]	[illegible]
Image9	0	2	0	2	0	2	0	2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	2	0	2

MOVIE12: Walk																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	14	10	14	10	14	10	14	10	14	10	14	10	14	10	14	10	14	10
Image2	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	[illegible]	[illegible]
Image3	2	5	2	5	2	5	2	5	2	5	2	5	2	5	2	5	2	5
Image4	0	7	0	7	0	7	0	7	0	7	0	7	0	7	0	7	0	7
Image5	0	13	0	13	0	13	0	13	0	13	1	13	1	13	1	13	1	13
Image6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Image7	14	0	14	0	14	0	14	0	14	0	14	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image8	0	12	0	12	1	12	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image9	4	4	4	4	4	4	4	4	[illegible]	[illegible]	4	4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

MOVIE13: Wave hand																		
	Number of images used for the simulation																	
	9		8		7		6		5		4		3		2		1	
Image1	4	14	4	14	4	14	4	14	4	14	4	14	4	14	4	14	4	14
Image2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]
Image3	0	12	0	12	0	12	0	12	0	12	0	12	0	12	[illegible]	[illegible]	[illegible]	[illegible]
Image4	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	2	4	[illegible]
Image5	2	9	2	9	2	9	2	9	2	9	2	9	2	9	2	9	[illegible]	[illegible]
Image6	0	10	0	10	0	10	0	10	0	10	0	10	0	10	[illegible]	[illegible]	[illegible]	[illegible]
Image7	7	14	7	14	7	14	7	14	7	14	7	14	7	14	[illegible]	[illegible]	[illegible]	[illegible]
Image8	14	0	14	0	14	[illegible]	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Image9	2	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	12	14

Figure 15. Simulation results for the configuration made of one A-SOM connected to itself with two sets of ancillary connections. The tables show the ability of the A-SOM to simulate the likely continuation of an observed behaviour. A light gray colour indicates that the A-SOM is able to simulate, a dark gray colour indicates that A-SOM simulates an activity pattern similar to the right one, and a black colour indicates that the A-SOM was unsuccessful at simulating the right activity pattern. The system needs between one and nine inputs to internally simulate the rest of the sequence.

simulate the remaining part of the sequence. In general, the system needed between four and nine inputs to be able to internally simulate the remaining parts of the actions. This is a reasonable result, since even humans cannot be expected to be able to predict the intended action of another agent without a reasonable amount of initial information. For example, looking at the initial part of an action like “punch”, we can hardly say what the person is going to do. It could be “punch” or “point”; we need more frames to exactly determine the performed action. In the same way, looking at a person starting to walk, we cannot say in advance if the person will walk or turn around or even kick because the initial postures are all similar.

The best results were obtained by using the A-SOM with two sets of time delayed ancillary connections. With this configuration, the system needed a less number of images to internally simulate the rest of the actions, as Figure 15 shows. Actions like “cross arm”, “kick”, and “point” needed only one image to simulate the rest of the sequence whereas with the other configuration almost nine images were needed. However even if the results are better than those we got with the other configuration, also in this case the ability to simulate varies with the type of action. Ambiguous actions, such as “wave hand” or “scratch head”, still need more images to simulate the likely continuation of their initial parts.

4. Conclusion

A new method based on the A-SOM, a novel variant of SOM, has been proposed to represent, discriminate, and to simulate actions. The proposed method highlights the strength of the A-SOM in classifying the observed action, and – thanks to its ability to remember perceptual sequences – the A-SOM is also able to simulate the likely continuation of the perceived behaviour of an agent. We tested two different configurations of the A-SOM: the first one with one set of time delayed recurrently connected ancillary connections; and one with two sets of time delayed recurrently connected ancillary connections, each set with a different time delay. The first configuration yielded good results in discrimination of actions, whereas the second considerably improved the system’s performance in the simulation of the likely continuation of partly seen actions. Moreover, we verified the ability of the A-SOM to recognise input never encountered before with encouraging results. In fact, the A-SOM recognises similar actions by eliciting close or identical centres of activity with a value of 62% of correctness. We are currently integrating robust size-, scale-, location-, and time-invariant features into the same architectural pattern and preliminary results are encouraging. I would like to stress again, however, that our goal is not to compete with state-of-the-art techniques for vision-based action recognition, but rather to investigate self-organising principles leading to the emergence of sophisticated social abilities such as action and intention recognition, and the work presented here is a step in that direction.

Many studies have been conducted with the aim of developing a system able to recognise as well as to simulate actions, but none of them offered a unique solution to cope with both the tasks. Pattern recognition methods, based on sophisticated statistical techniques, have been applied to recognise human actions, but our model goes beyond recognition of surface behaviour to simulate internal states and to predict future behaviour. Significant work in collaborative dialog and discourse theory rely on natural language processing for intention inference and plan recognition (Grosz & Sidner, 1988) whereas our work focuses on what can be understood from observing non-verbal behaviour. Relevant studies as those presented in Iosifidis et al. (2012) and in Huang & Wu (2010) use the SOM’s ability to cluster feature data and to reduce the data dimensionality to recognise actions that are similar to each other. In Iosifidis et al. (2012), the SOM created topological maps of features where similar actions were located close to each other and a multi-layer perceptron was used on top of these maps to perform the action recognition task. In

Huang & Wu (2010) the sequence of actions were mapped on the SOM lattice, creating trajectory representing the temporary evolution of the actions. The action recognition algorithm was based on, in this case, the longest common subsequence method. We obtained similar performance on the action recognition task by using similar classification procedures, but the bigger advantage of our system is given by the A-SOM's ability to remember perceptual sequence. Thanks to this ability the A-SOM is also capable of simulating the likely continuation of partially observed behaviour. Our system offers a unique solution to cope with both recognition and simulation tasks, and represents a step forward to the ambitious goal of developing a fully fledged system for understanding and anticipating others' short- and long-term intentions.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The authors gratefully acknowledge the support from the European Community's Seventh Framework Programme [grant agreement no. 612139].

Notes

1. The repository is available at http://4drepository.inrialpes.fr. It offers several movies representing sequences of actions. Each video is captured from five different cameras. For the experiments in this paper we chose the movie "Julien1" with the frontal camera view "cam0".
2. The repository is available at http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/. It offers several action datasets of depth sequences captured by a depth camera.
3. In the experiments presented in this paper we set $N_H = 15$ and $N_W = 15$.

References

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, *57*(5), 396–483.

Balkenius, C., Morén, J., Johansson, B., & Johnsson, M. (2010). Ikaros: Building cognitive models for robots. *Advanced Engineering Informatics*, *24*(1), 40–48.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. New York, NY: Oxford University Press.

Breazeal, C. (2004). *Designing sociable robots*. Cambridge, MA: The MIT Press.

Chella, A., Dindo, H., & Infantino, I. (2007). Imitation learning and anchoring through conceptual spaces. *Applied Artificial Intelligence*, *21*(4–5), 343–359.

Demiris, Y. (2007). Prediction of intent in robotics and multi-agent systems. *Cognitive Processing*, *8*(3), 151–158.

Dindo, H., & Schillaci, G. (2010, October 18–22). *An adaptive probabilistic approach to goal-level imitation learning*. 2010 IEEE/RSJ international conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, pp. 4452–4457.

Dindo, H., Zambuto, D., & Pezzulo, G. (2011, July 16–22). *Motor simulation via coupled internal models using sequential Monte Carlo*. Proceedings of the twenty-second international joint conference on artificial intelligence – Volume three, AAAI Press, Barcelona, Catalonia, Spain, pp. 2113–2119.

Giese, M. A., & Poggio, T. (2003). Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, *4*(3), 179–192.

Gkalelis, N., Tefas, A., & Pitas, I. (2008). Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition. *IEEE Transactions on Circuits Systems Video Technology*, *18*(11), 1511–1521.

Goldman, A. I. (2006). *Simulating minds: The philosophy, psychology, and neuroscience of mindreading*. Oxford: Oxford University Press.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as space–time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247–2253.

Grosz, B. J., & Sidner, C. L. (1988). *Plans for discourse* (Tech. rep.). DTIC document.

Huang, W., & Wu, Q. J. (2010, March 14–19). *Human action recognition based on self organizing map*. 2010 IEEE international conference on acoustics speech and signal processing (ICASSP), Dallas, TX, USA, pp. 2130–2133.

Iosifidis, A., Tefas, A., & Pitas, I. (2012). View-invariant action recognition based on artificial neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(3), 412–424.

Johnsson, M., Balkenius, C., & Hesslow, G. (2009, October 5–7). *Associative self-organizing map*. Proceedings of IJCCI, Madeira, Portugal, pp. 363–370 .

Johnsson, M., Gil, D., Balkenius, C., & Hesslow, G. (2010, July 14–16). *Supervised architectures for internal simulation of perceptions and actions*. Proceedings of BICS, Madrid, Spain.

Johnsson, M., Mendez, D. G., Hesslow, G., & Balkenius, C. (2011, May 24–26). *Internal simulation in a bimodal system*. Proceedings of SCAI, Trondheim, Norway, pp. 173–182.

Kohonen, T. (1988). *Self-organization and associative memory*. Berlin: Springer Verlag.

Li, W., Zhang, Z., & Liu, Z. (2010, June 13–18). *Action recognition based on a bag of 3d points*. 2010 IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW), San Francisco, CA, USA, pp. 9–14.

Lowe, D. G. (1999, September 20–27). *Object recognition from local scale-invariant features*. The proceedings of the seventh IEEE international conference on computer vision, Kerkyra, Greece, Vol. 2, pp. 1150–1157.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. Cambridge, MA: The MIT Press.

Pezzulo, G., Candidi, M., Dindo, H., & Barca, L. (2013). Action simulation in the human brain: Twelve questions. *New Ideas in Psychology*, *31*(3), 270–290.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*, *28*(6), 976–990.

Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, *1*(04), 515–526.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., ... Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, *56*(1), 116–124.

Tomasello, M., Carpenter, M., Call, J., Behne, T., & Moll, H. (2005). Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, *28*(5), 675–690.

Weinland, D., Ronfard, R., & Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, *115*(2), 224–241.