Hierarchical SOM-Based Detection of Novel Behavior for 3D Human Tracking

German Ignacio Parisi and Stefan Wermter

Abstract— **We present a hierarchical SOM-based architecture for the detection of novel human behavior in indoor environments. The system can unsupervisedly learn normal activity and then report novel behavioral patterns as abnormal. The learning stage is based on the clustering of motion with self-organizing maps. With this approach, no domain-specific knowledge on normal actions is required. During the tracking stage, we extract human motion properties expressed in terms of multi-dimensional flow vectors. From this representation, three classes of motion descriptors are encoded: trajectories, body features and directions. During the training phase, SOM networks are responsible for learning a specific class of descriptors. For a more accurate clustering of motion, we detect and remove outliers from the training data. At detection time, we propose a hybrid neural-statistical method for 3D posture recognition in real time. New observations are tested for novelty and reported if they deviate from the learned behavior. Experiments were performed in two different tracking scenarios with fixed and mobile depth sensor. In order to exhibit the validity of the proposed methodology, several experimental setups and the evaluation of obtained results are presented.**

I. Introduction

ANALYSIS of human activity has attracted great interest from researchers due to its promising applications in many areas such as assistance for the elderly, automatic surveillance and human-robot interaction [5]. Generally, activity recognition is presented in terms of extraction and classification of time varying feature data, i.e. matching test sequences with a group of reference sequences representing typical behaviors [2]. In this context, two major challenges must be addressed: a reliable estimation of the target position, and the definition of high-level knowledge for interpreting behaviors.

From a computer vision perspective, techniques for the estimation of real-world coordinates from 2D images remain a challenge in data-driven 3D tracking [1]. Recently, measurements from depth sensors have been used to estimate 3D motion and detect falls. In [7][8][9][10], the authors use depth information to accurately estimate the target's body posture and detect falls. However, most of the proposed algorithms on fall detection rely on prior scene analysis, e.g. the estimation of ground surface. Furthermore, they estimate domain-specific threshold values for abnormal body velocity. In fact, it should be considered that human actions can be influenced by many factors. Therefore, a system for learning behavioral patterns should not assume a substantial amount of prior knowledge. It has been shown that normal actions can be learned unsupervisedly and a subspace of typical actions can be obtained adaptively [17]. The analysis of motion patterns is an effective method to better understand human behavior [18].

From a machine learning perspective, the detection of novel behavior can be seen as the identification of new data that a system is not aware of during training [3]. Novelty detection has been researched within diverse application domains to identify patterns that do not conform to the expected normal behavior [4]. Among different neural architectures for unsupervised learning, self-organizing networks have shown to be suited to behavior classification when motion can be expressed in terms of multi-dimensional flow vectors [5].

We present a 3D tracking framework for the detection of novel human behavior in indoor environments. The system learns unsupervisedly normal activity and detects novel behavioral patterns as abnormal. For the 3D tracking, we extract spatio-temporal properties that describe human motion in the scene. We estimate target position, velocity and body orientation from depth map video sequences. Tracked motion from depth sequences is extracted in terms of multi-dimensional flow vectors. For the learning stage, we propose a hierarchical architecture based on four self-organizing map (SOM) networks. To tackle the effect of tracking errors, a first SOM is used to remove outliers from the training motion vectors. Preprocessed vectors are encoded into three classes of descriptors: trajectories, body features and directions. Each class of motion descriptor is then used to train an independent SOM network. At detection time, new observations that deviate from the learned behavior are reported as abnormal. We performed experiments in two different tracking scenarios: (1) fixed depth sensor for tracking multiple targets and (2) active tracking with a mobile robot platform.

This paper is organized as follows. In Section II we present the related work on SOM-based novelty detection. Section III describes our study of human motion and the extraction of 3D motion features. We describe the preprocessing stage for the removal of outliers caused by tracking errors and the encoding of motion descriptors. In Section IV we introduce our hierarchical architecture for the unsupervised detection of novelties. The experimental setups for motion acquisition and two different tracking scenarios are presented in Section V. We discuss the achieved results and evaluate the algorithm under different detection conditions in Section VI. We present our concluding remarks in Section VII.

German Ignacio Parisi is with the Department of Informatics, University of Milano-Bicocca, Milan, Italy (email: g.parisi4@campus.unimib.it).

Stefan Wermter is with the Department of Informatics, University of Hamburg, Germany (email: wermter@informatik.uni-hamburg.de).

II. Related Work

Some previous approaches combine computer vision and computational intelligence to learn extracted representations of motion patterns. In this field, extensions of the SOM network have shown to be a powerful tool in detecting patterns not presented during the training. Hew et AL. (2004) [15] propose a SOM model for learning patterns of vehicle trajectories. The architecture consists of a hierarchical SOM used for learning the distribution of normal patterns. Novel patterns are then considered as anomalies. In [14], the authors used a fuzzy self-organizing map (fuzzy SOM), where each output neuron directly corresponds to a class of trajectories. To obtain an accurate model of the scene, many neurons are required. H. Al-Khateeb et al. (2011) [18] propose an extended fuzzy SOM with a small number of nodes to detect abnormal trajectories of pedestrians. In [19], the authors train a SOM to recognize human actions from a sequence of images. Most of these approaches extract motion from 2D color images. An important limitation is a substantial computational effort for the extraction of features, which in some cases does not allow the detection of abnormal behaviors in real time. With the use of depth information, it is possible to capture both spatial and temporal properties of 3D motion. This approach allows a more flexible representation of human motion and a better performance at detection time.

III. 3D Human Tracking

A human body can be modeled as a spatially extended object with body parts connected by joints. The distribution of the body masses will then change depending on the posture. Cognitive evidence on motion tracking suggests the heuristic estimation of a center of mass to represent the point where all the masses of the body concentrate [6]. We will now describe our model for 3D tracking of human bodies and the preprocessing stage to address tracking errors.

A. Human Body Representation

We estimate the position of a moving target based on a 3D model of the human skeleton. Each body joint is represented as a point sequence of real-world coordinates $C = (x, y, z)$. We consider two centers of mass. The upper-centroid U describes the position of the upper-body with respect to the torso and the shoulders. The lower-centroid L describes the position of the lower-body with respect to the torso and the hips. The estimation of U and L is independent from the rotation of the body, therefore the centroids are computed if the tracked person is frontal to the sensor, turned on their side or back. To describe the overall body orientation we estimate the orientation of the torso T, expressed as the slope of the segment between U and L with respect to the image plane, i.e. the segment is vertical if the body is upright (see Fig. 1).

We define the body velocity S_i as the difference in pixels of the upper-centroid U between two consecutive frames i and $i-1$. We then encode S_i as horizontal speed H_i and vertical speed V_i with respect to the image plane. The former refers to the target moving on the width and depth axis, i.e. closer, further, right, and left. The latter represents the speed with respect to height, e.g. negative if the target is moving down. Horizontal and vertical body speeds are expressed respectively as $H_i = \sqrt{S_i^2(x) + S_i^2(z)}$ and $V_i = S_i(y)$. Finally, we estimate the relative direction of 3D motion in space at time i as $D_i = \{S_i(x)/D, S_i(y)/D, S_i(z)/D\}$ with $D = \sqrt{S_i^2}$.

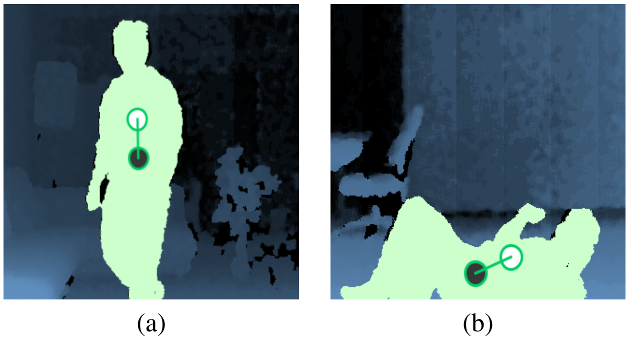


Fig. 1. Body representation for 3D motion tracking of human actions. We estimate two centers of mass: one for the upper-body (white dot) and one for the lower-body (black dot). The torso orientation (green line) is estimated as the slope of the segment between the two centers with respect to the image plane. We show the orientation for a body in upright position (a) and a body falling down (b).

For every tracked target at time i, we obtain a flow motion vector

$$M_i = (U_i, H_i, V_i, T_i, D_i)\,. \tag{1}$$

Flow motion vectors as defined by Eq. 1 are able to capture both spatial and temporal information of human motion. On the one hand, their advantage is low computational complexity and a simple implementation. In our case, measurements are stand-alone and time is not considered as a feature. On the other hand, flow motion vectors are susceptible to tracking errors, e.g. non-linearities of the sensor. Depth sensors such as the Microsoft Kinect[1] and ASUS Xtion[2] have the potential to be used in applications for 3D tracking where the requirements for accuracy are less strict. Therefore, systematic and random errors can occur during tracking activities. Noisy observations can be caused by tracked motion from highly occluded targets or tracking in scenarios with a moving sensor, e.g. active tracking. In a scenario with multiple users populating the scene, occlusion errors may also occur because of often overlapping bodies.

As shown by our experiments, outliers in the training data may lead to non-linear distortions of the subspace of normal behaviors. Therefore, the detection and removal of outliers must be addressed.

B. Preprocessing

The preprocessing stage consists of two sequential operations: (1) the detection and removal of outliers from

[1] Kinect for Windows. `http://www.microsoft.com/en-us/kinectforwindows`

[2] ASUS Xtion Pro Live. `http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO_LIVE/`

extracted raw motion vectors and (2) the encoding of motion descriptors.

An outlier is defined as an observation that does not follow the pattern suggested by the majority of the observations belonging to the same data cloud [16]. From a geometrical perspective, outliers are to be found detached from the dominating distribution of the subset for normal actions. In our case, we assume that the behavior of a moving target must be consistent over time. This means that it will be concentrated to one or a couple of regions of the feature space of the training observations [13]. Therefore, we consider inconsistent changes in body velocity and torso orientation to be caused by tracking errors rather than real tracked motion.

To tackle the effect of noisy observations, we use a SOM-based outlier detection that removes outliers from large multi-dimensional data sets. After removing outliers from the data cloud, preprocessed flow motion vectors are encoded into three classes of motion descriptors:

1) *Trajectories*: sequences of target's tracked positions for visited areas in the scene

$$T_n = \{U_1, ..., U_n\}; \tag{2}$$

2) *Body features*: sequences of body speeds (vertical and horizontal) and torso orientations

$$F_n = \{(H_1, V_1, T_1), ..., (H_n, V_n, T_n)\}; \tag{3}$$

3) *Directions*: sequences of relative directions of motion

$$R_n = \{D_1, ..., D_n\}. \tag{4}$$

This representation with three different classes of descriptors aims to adapt the detection of novel behavioral patterns for different tracking scenarios and environments.

On the one hand, trajectories describe patterns on geometrical properties of the environment, i.e. visited areas in a room. On the other hand, body features and directions are geometry-independent descriptors and are still relevant in different environments and scenarios with a mobile sensor.

IV. SOM-based Novelty Detection

The self-organizing map (SOM) is a competitive neural network introduced by Teuvo Kohonen [11]. The SOM projects statistical relationships between high-dimensional data items into a low-dimensional discretized representation of the input space. The network learns by iteratively reading each training vector. The training algorithm computes the models so that they describe the domain of observations. It adopts a neighborhood function to preserve the topological properties of the training data [11].

For the detection of novelties, we propose a hybrid neural-statistical architecture [13], which is extended towards multiple layers for 3D posture recognition in real time. The concept is to approximate the normal behavior with specific trained SOM networks. This approach is unsupervised and therefore no a priori information on class labels of training data is necessary. Fig. 2 illustrates the flow chart for the learning stage. The proposed learning architecture consists of four SOM networks. A first network S_o is used to detect and remove outlier values from the extracted motion vectors. Preprocessed vectors are then encoded into three classes of motion descriptors. An independent SOM network for each descriptor is trained. We denote the networks for trajectories, body features and directions as S_1, S_2, and S_3 respectively.

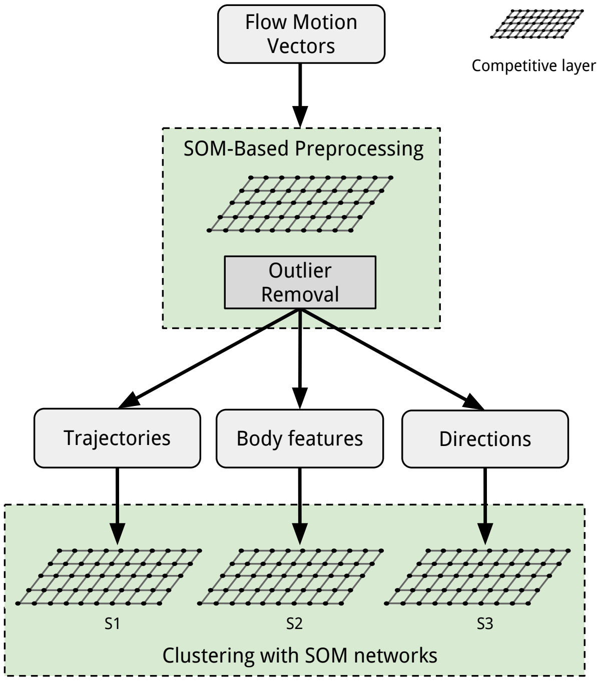


Fig. 2. Hierarchical SOM-based architecture for the training stage. A first SOM is used to detect and remove outliers from raw extracted motion vectors. Preprocessed vectors are then encoded into three classes of descriptors and fed to three independent SOM networks as training observations.

At detection time, new observations from tracked motion are tested for novelty. Test observations are encoded and each motion descriptor is processed in relation to the specific trained network. Fig. 3 illustrates the flow chart for the novelty detection phase.

We will now describe the hierarchical training algorithm for learning a subset of normal actions and the detection algorithm to test new observations for novelty.

A. Training Algorithm

A SOM network consists of an input layer and a competitive layer. In the classical SOM, the number of units and their topological relations are set from the beginning. Every unit is connected to adjacent units by a neighborhood relation that dictates the structure of the map.

For our hierarchical architecture we consider two-dimensional networks with competitive units arranged on a hexagonal lattice in the Euclidean space. The hexagonal shape is generally preferred because all 6 neighbors of a unit will then be at the same distance, instead of 8 neighbors in a rectangular fashion. Each unit i has an associated d-dimensional model vector $m_i = [m_{i1}, m_{i2}, ..., m_{id}]$. When

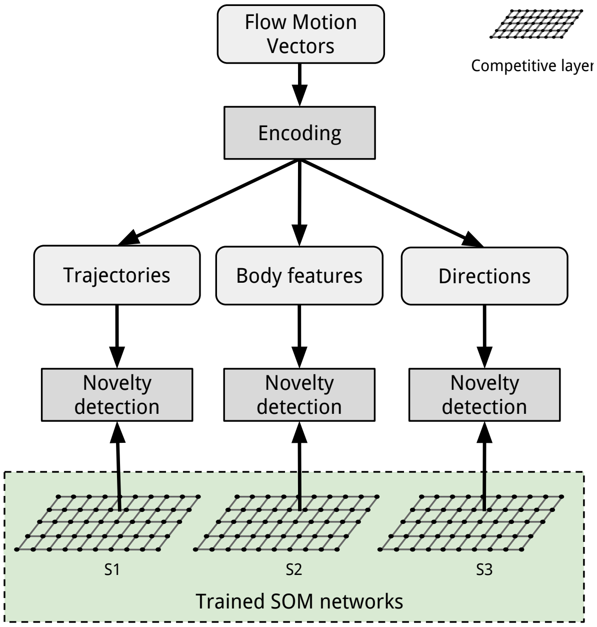


Fig. 3. Neural-statistical architecture for novelty detection based on trained SOM networks. For every test observation, novelty values are calculated from each SOM.

an input vector $x = (x_1, ..., x_d)$ is presented to the network, the units compete and the best matching unit c for x_j is selected by the smallest Euclidean distance as

$$c = \arg\min_i \|x_j - m_i\|. \quad (5)$$

For an input vector x_i, the quantization error q_i is defined as the distance from the best matching unit c_i.

The number of units of each map is set to $m = n/10$ with n training vectors. As shown by our experiments with different network configurations, this value represents a good interplay between accuracy and generalization capabilities of the mapping. At the initialization phase, the weights of neurons are set to random values from the domain of the input vectors.

Each network is trained with a batch variant of the SOM algorithm. This approach requires fewer parameters and converges much faster than the traditional stepwise recursive algorithm [12]. In batch learning, the entire training set is presented at once. Only afterwards is the map updated with the network effect of all the samples. The updating is done by replacing the model vector m_i with a weighted average over the samples, where the weighting factors are obtained from the neighborhood function.

The update step for batch learning is formally defined as

$$m_i(t+1) = \frac{\sum_{j=1}^{n} h_{ic(j)}(t) x_j}{\sum_{j=1}^{n} h_{ic(j)}(t)}, \quad (6)$$

where c is the best matching unit as defined by Eq. 5, $h_{ic(j)}$ is the neighborhood function as defined by Eq. 7, and n is the number of sample vectors.

As the neighborhood relation, we use the Gaussian function

$$h_{j,i}(x) = exp\left(\frac{-\|r_c - r_i\|^2}{2\sigma^2(t)}\right), \quad (7)$$

where r_c is the location of unit c on the map grid and $\sigma(t)$ is the neighborhood radius at time t.

Input vectors for the first SOM consist of flow motion vectors as defined by Eq. 1. The SOM networks for trajectories, body features and directions use as input the preprocessed motion descriptors as defined by Eq. 2, 3 and 4 respectively.

Before each training phase, input vectors are normalized to avoid range-biased clustering. Scaling of variables is of special importance since the SOM algorithm uses the Euclidean distance to measure distances between vectors. It is of our interest that variables with a different range of values are equally important. A well-known approach to achieve this is to linearly scale all variables so that the variance of each is equal to one. Given a set of n multi-dimensional training vectors, we perform a standard score single-variable transformation.

B. Detection Algorithm

The goal of the detection algorithm is to test if the most recent observation is novel or not. For this purpose, the degree of novelty for every test observation is expressed with the estimation of a P-value. If the P-value is smaller than a given threshold, then the observation is considered to be novel and reported.

For each new test observation x_{n+1} presented to the S_i, the algorithm is summarized as follows [13]:

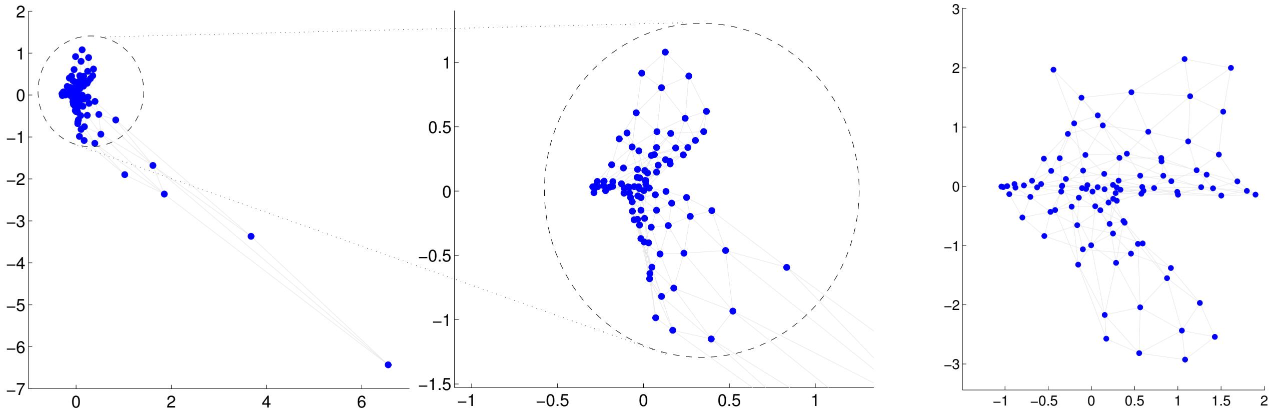
1) Normalize x_{n+1} with respect to the training set D_i.
2) Estimate $q_{(n+1)}$ with respect to S_i.
3) Define B as the number of quantization errors $(q_1, ..., q_n)$ from S_i greater than $q_{(n+1)}$.
4) Define the novelty P-value as $P_{(n+1)} = B/n$.

In the case of S_o, observations with P-values under the novelty threshold T_o are considered as outlier values and therefore removed from the training set. For S_i with $i = \{1, 2, 3\}$, if $P_{(n+1)}$ is smaller than T_i, the test observation x_{n+1} is considered as novel.

As an extension of the algorithm proposed in [13], a different novelty threshold is estimated for each trained network S_i with $i = \{1, 2, 3\}$. The choice of convenient threshold values that take into account the characteristics of the distributions can have a significant impact on the successful rates for novelty detection. We now empirically define two different thresholds that consider the distribution of the quantization errors from each trained SOM. For the detection of outliers with S_o, we set

$$T_o = \sqrt{\overline{Q_o} + \sigma(Q_o) + \max(Q_o) + \min(Q_o)} * 0.5, \quad (8)$$

where Q_o is the set of quantization errors of S_o, $\overline{Q_o}$ is the mean value of Q_o, and $\sigma(Q_o)$ is the deviation standard.



(a) (b)

Fig. 4. Effects of outliers in the clustering of training data. The first SOM (a) was trained with the full set of extracted motion vectors. As seen in the zoomed area in (a), the presence of highly noisy observations decreased the sparcity of the feature map. The second SOM (b), which was trained after the removal of outliers, gave a more representative clustering of the observations from tracked motion.

For S_i with $i = 1, 2, 3$, the novelty threshold is set to

$$T_i = \left[\frac{\overline{Q_i} + \sigma(Q_i)}{\max(Q_i) + \min(Q_i)} \right] * 0.1. \qquad (9)$$

Our experiments show that the novelty thresholds, as defined by Eq. 8 and 9, represent a balanced trade-off to filter outliers and tolerate observations outside the boundaries of the dominating data cloud.

Fig. 4 illustrates the visual results of our hierarchical learning stage to show the effects of outliers in the training data. A first SOM was trained with the full set of extracted flow motion vectors. As shown in Fig. 4.a, outliers in the data decreased the sparcity of the feature map. These noisy observations were detected by our algorithm and removed from the training set. The second SOM was then trained with the preprocessed motion descriptors. The removal of outliers allowed a more representative clustering of the motion vectors for normal activity (Fig. 4.b).

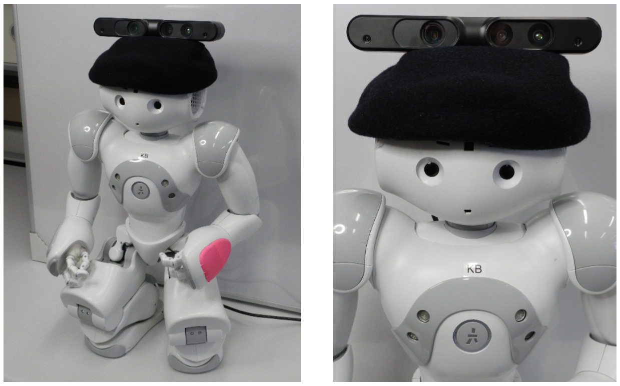
V. Experimental Setups

A. Motion Acquisition and Clustering

For the acquisition of the training data, we monitored a home-like environment with a Kinect. The sensor was installed on a platform 1,30 meters above the ground and positioned parallel to the horizontal surface. Depth maps were acquired with a VGA resolution of 640x480 and the depth operation range was set from 0.8 to 3.5 meters. The angular field of view was 57 degrees horizontally and 43 degrees vertically. Video sequences were sampled at a constant frame rate of 30 Hz. For each point sequence of real-world coordinates, we calculated the median value of the last 3 measurements. This technique returns depth values that are more robust to random noisy measurements from the sensor.

TABLE I
Parameters for the SOM training algorithms.

Parameter	Value
Map grid	Hexagonal lattice
Distance function	Euclidean
Neighborhood function	Gaussian
Initialization	Random
Training algorithm	Batch



(a) (b)

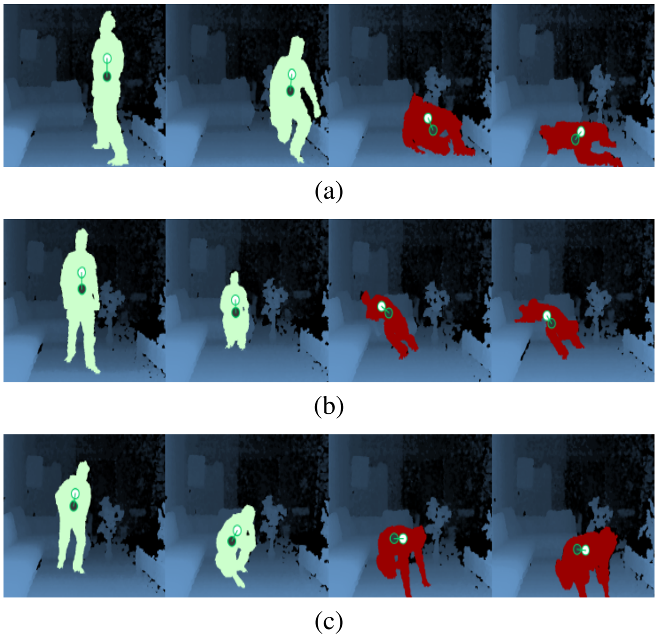
Fig. 5. Humanoid robot Nao with Xtion depth sensor for active tracking. This approach allows to exploit Nao's head motion capabilities to actively track a moving person in the environment.

The training sequences consisted of 20 minutes of video with domestic actions. The actions were performed by different actors and included walking around the environment, sitting down, and picking up objects from the ground. From the training sequences, a total of 35.062 flow motion vectors were extracted. For the clustering, vectors were considered independently. The parameters for the SOM training algorithms are listed in Table I.

B. Detection Scenarios

At detection time, we performed experiments in two different tracking scenarios. The first scenario consisted of a fixed sensor 1,30 meters above the ground. The number of actors temporarily in the scene varied from one to three.

In order to overcome the limitation on the reduced field


(a)

(b)

(c)

Fig. 6. Novelty detection in depth video sequences: (a) falling down, (b) fainting on a sofa, and (c) crawling. Red body area indicates novel behavior.

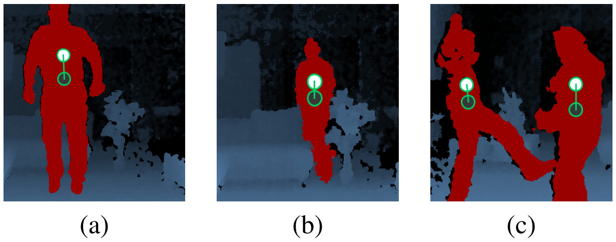
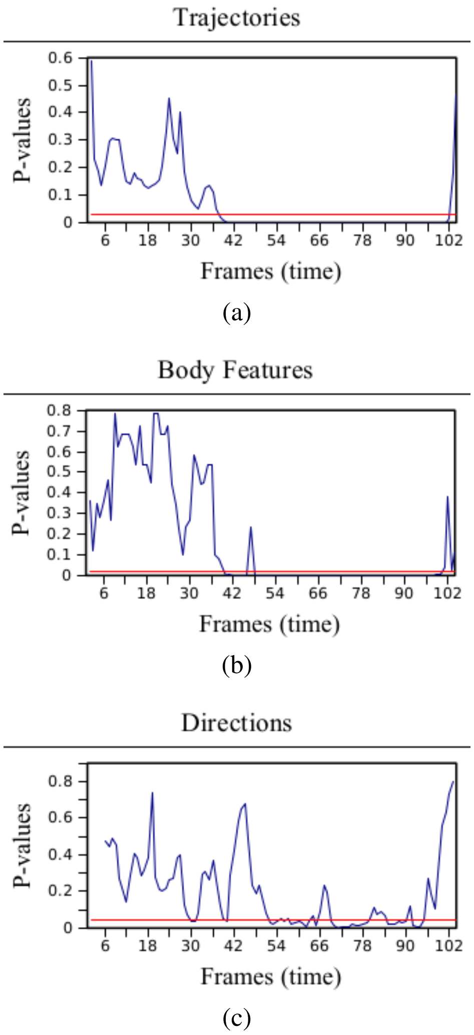

(a) (b) (c)

Fig. 7. Novel behaviors detected with fixed sensor: (a) jumping, (b) visiting a novel area behind the sofa, and (c) fighting.

of view of the camera, we set a second scenario with a mobile sensor. The human target was actively tracked around the environment. For this purpose, we installed the depth sensor on top of the humanoid robot Nao[3]. As shown in Fig. 5, we extended the Nao with a new ASUS Xtion Pro Live sensor. The device is based on the same technology as Kinect, but smaller and lighter. As shown by our experiments, the reduced weight of the Xtion did not affect the overall stability of the Nao. The modified Nao could correctly turn its head (pan, tilt angles) and walk without falling down. The tracking framework was responsible for computing the operations required to keep the target in the scene.

For both scenarios, experiments were performed in different environments and light conditions. Detection of novelties was tested from real time video streams and recorded data sets. As an approach to reduce false positives caused by tracking errors, the system reported a novel behavior only if three consecutive P-values were below the novelty threshold.



(a)

(b)

(c)

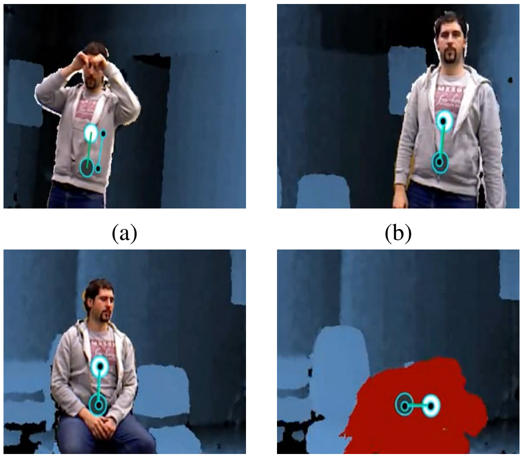
Fig. 8. P-values of motion descriptors for a falling down sequence: trajectories (a), body features (b) and directions (c). P-values for novel behavior lie under the novelty threshold (red line).

VI. Results and Discussion

The system successfully detected behavioral patterns not presented during the training. The first SOM network detected and removed 502 outliers from the training vectors. At detection time, the actions reported as abnormal included video sequences with actors falling down, fainting, crawling, jumping, visiting novel areas in the environment, and starting to fight (Fig. 6 and 7). The system also reported novel behavioral patterns for multiple users populating the scene (Fig. 7.c). The expectation was that estimated P-values for sequence frames with novel activity were under the novelty threshold for the corresponding motion descriptor. As shown in Fig. 8 for a fall sequence, P-values were under the thresholds for trajectories, body features and directions. In this case, the fall sequence led to a novel area visited in the scene, increased body speed, novel torso orientation, and a novel direction of motion towards the ground.

For the detection scenario with the humanoid Nao, the moving target was successfully tracked around the room

[3] Nao humanoid robot developed by Aldebaran Robotics. `http://www.aldebaran-robotics.com/`



(a) (b)

(c) (d)

Fig. 9. Novel behavior detection with active tracking. The person moves around the environment (a-c) and faints on a chair (d). For a better visualization of the target, depth and color information were calibrated.

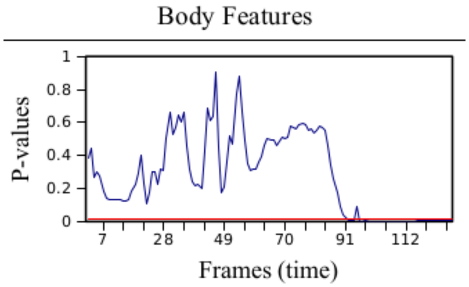


Fig. 10. Active tracking for novelty detection on a fainting sequence. P-values for the novel behavior are below the novelty threshold (red line).

and the system reported abnormal behaviors in different environments. At detection time, only P-values for body features were considered, i.e. the novelty detection was focused on the speed and posture, since the configuration of the environment kept changing and the sensor was mobile. Fig. 9 illustrates a target moving around the room and then fainting while sitting on a chair. Different from a general falling down sequence, the fainting sequence did not end up on the ground. Furthermore, the body speed was not necessarily increased with respect to other actions performed during the training sequences. As shown in Fig. 10, P-values for body features were under the threshold for action frames with the target lying on the chair.

The processing of depth video sequences showed a reduced computational effort in comparison with the use of color information. This approach allowed to obtain same accuracy rates with novelty detection in real time and from recorded video sequences. These results could have significant relevance in scenarios for ambient assisted living [20] and learning robots for indoor human behavior [21].

A. Evaluation

We evaluated the detection algorithm using the standard measurements defined in [22]:

TABLE II
RESULTS FOR THE EVALUATION OF OUR NOVELTY DETECTION ALGORITHM FOR THE FIXED SENSOR SCENARIO.

	Preprocessed				Raw			
	TP	TN	FP	FN	TP	TN	FP	FN
1 actor a	12	8	0	0	11	8	0	1
1 actor b	11	8	1	0	11	7	1	1
2 actors a	11	7	1	1	9	7	1	2
2 actors b	8	7	3	2	9	4	3	4
	42	30	5	3	40	26	5	8

TABLE III
RESULTS FOR THE EVALUATION OF OUR NOVELTY DETECTION ALGORITHM FOR THE ACTIVE TRACKING SCENARIO.

	Preprocessed				Raw			
	TP	TN	FP	FN	TP	TN	FP	FN
Actor a	11	9	0	0	11	7	1	1
Actor b	11	6	2	1	10	5	3	2
Actor c	9	7	3	1	9	7	2	2
Actor d	10	7	1	2	9	6	3	2
	41	29	6	4	39	25	9	7

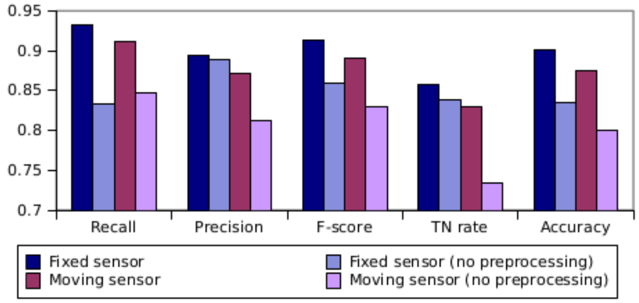


Fig. 11. Evaluation of our detection algorithm under four different conditions: fixed and moving sensor with and without the removal of outliers from the training data. For both tracking scenarios, preprocessing before the learning stage increased accuracy rates at detection time.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{11}$$

$$\text{F-score} = 2 * \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}, \tag{12}$$

$$\text{True negative rate} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{13}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{14}$$

A true positive (TP) was obtained if a novelty was detected between the first and the last frame where the novel action took place. True negatives (TN) refered to normal actions not detected as novel. False positives (FP) and false negatives (FN) refered respectively to normal actions reported as novel and novel behaviors not reported by the system.

Actors performed 20 different normal and abnormal actions several times. Normal actions included domestic activities such as walking, sitting down and standing up, and picking up objects. Abnormal actions consisted in behavioral patterns not presented during the training, e.g. falling down, crawling, jumping, and visiting novel areas. Novelty detection was performed for different tracking scenarios and number of actors in the scene. The results for the evaluation of our novelty detection algorithm are summarized in Tables II and III.

The use of the SOM in the first layer of the hierarchical architecture led to increased accuracy rates at detection time. As shown by our experimental results, only a small number of outliers in the training data caused a distortion of the subspace of normal actions. The removal of outliers from the training data increased accuracy rates for both scenarios of 7% and 9% respectively. To provide a qualitative idea of the improvement introduced by the use of the first SOM, Fig. 11 illustrates the accuracy rates for the detection scenarios under different conditions.

VII. Conclusions

We presented a hierarchical SOM-based architecture for the detection of novel human behavior in indoor environments. Unlike other research presented in the field of abnormal action detection [7][8], our approach did not require prior scene analysis, e.g. the estimation of ground surface. Experimental results showed the validity of the proposed methodology for different tracking scenarios. The removal of outliers from the training set represented an important stage before the clustering with SOM networks. The use of the SOM in the first layer of the hierarchical architecture led to increased accuracy rates for novelty detection. With the use of the humanoid Nao and a depth sensor, it was possible to actively track a moving target in the environment. This approach overcame the limitations of a reduced field of view of fixed cameras and range sensors. The system could detect novelties in real time and from recorded data sets with the same detection performance.

There are different areas that can benefit from robust vision-based human motion analysis, e.g. automatic surveillance, ambient assisted living with learning robots and human-robot interaction. Our representation of human 3D motion extracted relevant characteristics of human activity expressed in terms of multi-dimensional vectors. The hybrid neural-statistical architecture based on self-organizing maps has substantial potential for clustering flow motion vectors and detecting novelties.

References

[1] F. Talantzis, A. Pnevmatikakis and A. G. Constantinides, “Audio-visual person tracking. A practical approach.,” *Communications and Signal Processing*, vol. 4, Imperial College Press, UK, 2012.

[2] D. Gowsikhaa, S. Abirami and R. Baskaran, “Automated human behavior analysis from surveillance videos: a survey,” *Applied Imagery Pattern Recognition Workshop*, pp. 1-8, 2008.

[3] M. Markou and S. Singh, “Novelty detection: A review - part 2: Neural network based approaches,” *Signal Processing*, vol. 83, 2003.

[4] V. Chandola, A. Banerjee and V. Kumar, “Anomaly detection: A survey,” *ACM Computer Surveillance*, Article 15, 2009.

[5] W. Hu, T. Tan, L. Wang and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, no. 3, pp. 334-352, 2004.

[6] G. A. Alvarez and B. J. Scholl, “How does attention select and track spatially extended objects? New effects of attentional concentration and amplification,” *Journal of Experimental Psychology*, General 134, pp. 461-476, 2005.

[7] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte and J. Meunier, “Fall detection from depth map video sequences,” *Proceedings of the 9th international conference on Toward useful services for elderly and people with disabilities: smart homes and health telematics*, pp. 121-128, 2011.

[8] C. Kawatsu, J. Li and C. Chung, “Development of a fall detection system with Microsoft Kinect,” Department of Mathematics and Computer Science, Lawrence Technological University, MI, USA, 2012.

[9] G. Mastorakis and D. Makris, “Fall detection system using Kinects infrared sensor,” *Journal of Real-Time Image Processing*, Springer-Verlag, 2012.

[10] R. Planinc and M. Kampel, “Introducing the use of depth data for fall detection,” *Personal and Ubiquitous Computing*, Springer-Verlag, 2012.

[11] T. Kohonen, “Self-Organizing Maps,” *Series in Information Sciences*, vol. 30, Springer, Heidelberg. Second ed., 1995.

[12] T. Kohonen, “Essentials of the self-organizing map,” *Neural Networks*, vol. 37, pp. 52-65, 2013.

[13] A. J. Hglund, K. Htnen and A. S. Sorvari, “A computer host-based user anomaly detection system using self-organizing maps,” *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. 5, pp. 411-416, 2000.

[14] W. Hu, D. Xie, T. Tan, and S. Maybank, “Learning activity patterns using fuzzy self-organizing neural network,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 3, pp. 1618-1626, 2004.

[15] W. Hu, D. Xie and T. Tan, “A hierarchical self-organizing approach for learning the patterns of motion trajectories,” *IEEE Transactions on Neural Networks*, vol. 15, no. 1, pp. 135-144, 2004.

[16] A. K. Nag, A. Mitra, S. Mitra, “Multiple outlier detection in multivariate data using self-organizing maps title,” *Computational Statistics*, vol. 2, no. 2, pp. 245-264, 2005.

[17] T. Nanri and N. Otsu, “Unsupervised abnormality detection in video surveillance,” *IAPR Conference on Machine Vision Applications*, pp. 574-577, 2005.

[18] H. Al-Khateeb and M. Petrou, “An Extended fuzzy SOM for anomalous behaviour detection,” *In Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference*, pp. 31-36, 2011.

[19] T. da Rocha, F. de Barros Vidal and A. R. S. Romariz, “A proposal for human action classification based on motion analysis and artificial neural networks,” *IEEE World Congress on Computational Intelligence*, pp. 10-15, 2012.

[20] W. Yan, E. Torta, D. van der Pol, N. Meins, C. Weber, R. H. Cuipers, S. Wermter, “Learning robot vision for assisted living,” *In Garcia-Rodriguez, J., Cazorla, M., editors, Robotic Vision: Technologies for Machine Leaning and Vision Applications*, ch. 15, pp. 257-280, IGI Global, 2013.

[21] W. Yan, C. Weber, S. Wermter, “A neural approach for robot navigation based on cognitive map learning.,” *Proceedings of the International Joint Conference on Neural Networks*, pp. 1146-1153, 2012.

[22] C. J. Van Rijsbergen, “A computer host-based user anomaly detection system using self-organizing maps,” *Information Retrieval*, Butterworth-Heinemann, 2nd edition, London, 1979.