Human Activity Recognition from Automatically Labeled Data in RGB-D Videos

David Jardim[1,2,3,4], Luís Nunes[2,3,4], and Miguel Dias[2,3,4]

[1]Microsoft Language Development Center, Lisbon, Portugal
Email: t-dajard@microsoft.com

[2]Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
Email: {luis.nunes, miguel.dias}@iscte.pt

[3]Instituto de Telecomunicações, Lisbon, Portugal

[4]ISTAR-IUL, Lisbon, Portugal

Abstract—**Human Activity Recognition (HAR) is an interdisciplinary research area that has been attracting interest from several research communities specialized in machine learning, computer vision, medical and gaming research. The potential applications range from surveillance systems, human computer interfaces, sports video analysis, digital shopping assistants, video retrieval, games and health-care. Several and diverse approaches exist to recognize a human action. From computer vision techniques, modeling relations between human motion and objects, marker-based tracking systems and RGB-D cameras. Using a Kinect sensor that provides the position of the main skeleton joints we extract features based solely on the motion of those joints. This paper aims to compare the performance of several supervised classifiers trained with manually labeled data versus the same classifiers trained with data automatically labeled. We propose a framework capable of recognizing human actions using supervised classifiers trained with automatically labeled data.**

I. Introduction

The goal of human activity recognition is to successfully classify an action performed by an individual or a group of people from a video observation. Although significant progress has been made, HAR remains a challenging area with several problems to solve. Manual analysis of video is labour intensive, fatiguing, and error prone. Solving the problem of recognizing human activities from video can lead to improvements in several application fields like surveillance systems, human computer interfaces, sports video analysis, digital shopping assistants, video retrieval, gaming and health-care [1]–[5]. We are interested in recognizing high-level human activities, ideally our recognition algorithm should be robust to changes in relative distance between the body and the sensor (Kinect), skeleton orientation, and speed of an action. In order to abstract ourselves from computer vision problems the Kinect sensor will be used to extract 3D skeleton data. Usually manually labeled data is used to perform some kind of training of classifiers that will then recognize the human activities. What if this labeling could be achieved automatically? How would the results compare? This paper shows that automating the data labeling process for the type of actions studied is possible and results in a minor loss in accuracy.

II. Related Work

Human activity recognition is a classification problem in which events performed by humans from video data are automatically recognized. Driven by application demands, this field has seen a relevant growth in the past decade. The previous approaches all used computer vision (CV) techniques to extract meaningful features from the data. Motion capture data (MOCAP) has also been used in this field, a relevant approach found was [6] where they pose the problem of learning motion primitives (actions) as a temporal clustering one, and derive an unsupervised hierarchical bottom-up framework called hierarchical aligned cluster analysis (HACA). HACA finds a partition of a given multidimensional time series into m disjoint segments such that each segment belongs to one of k clusters representing an action. They were able to achieve competitive detection performances (77%) for human actions in a completely unsupervised fashion. Using MOCAP data has several advantages mainly the accuracy of the extracted features but the cost of the sensor and the required setup to obtain the data is often prohibitive.

There are some approaches which combine motion information and object properties [7], [8]. In [7] the authors abstract the problem in two stages. First, by recognizing general motions such as moving, not moving or tool used. Second, by reasoning about more specific activities (Reach, Take, etc.) given the current context, i.e. using the identified motions and the objects of interest as input information. They've obtained an accuracy classification of 92%. [8] propose a two-level hierarchical action segmentation (HAS) approach that take into account contact relations between human end effectors, the scene, and between objects in the scene, using 6D pose trajectories extracted from marker-based tracking system. This work shows that HAS allows the identification of meaningful segments in complex human demonstrations without over-segmentation and without omitting important demonstration key frames.

With cost in mind Microsoft released a sensor called Kinect, which captures RGB-D data and is also capable of providing joint level information in a non-invasive way allowing the developers to abstract away from CV techniques. A previous study using Kinect [9] consider the problem of extracting a descriptive labeling of the sequence of sub-activities being performed by a human, and more importantly, of their interactions with the objects in the form of associated affordances. The learning problem is formulated using a structural support vector machine (SSVM) approach, where labelings over various alternate temporal segmentations are considered as latent variables. The method obtained an accuracy of 79.4% for affordance, 63.4% for sub-activity and 75.0% for high-level activity labeling.

In [10] the covariance matrix for skeleton joint locations over time is used as a discriminative descriptor for a sequence of actions. To encode the relationship between joint movement and time, multiple covariance matrices are deployed over sub-sequences in a hierarchical fashion. Their experiments show that using the covariance descriptor with an off-the-shelf classification algorithm one can obtain an accuracy of 90.53% in action recognition on multiple datasets.

In a parallel work [11] authors propose a descriptor for 2D trajectories: Histogram of Oriented Displacements (HOD). Each displacement in the trajectory votes with its length in a histogram of orientation angles. 3D trajectories are described by the HOD of their three projections. HOD is used to describe the 3D trajectories of body joints to recognize human actions. The descriptor is fixed-length, scale-invariant and speed-invariant. Experiments on several datasets show that this approach can achieve a classification accuracy of 91.26%.

The method developed by [12] addressed an interesting problem of transferring depth information to a target of RGB action data (depth data is not available) and used both RGB data and the learned depth data for action recognition. By borrowing an auxiliary dataset, with both RGB and depth data they are capable of uncovering missing depth information in the target data, couple two modalities (RGB and depth) and capture structure information. From their experiments they achieved superior performance over existing methods with accuracy values of 92.09%.

Recently and more directly related to our research, [3] developed a system called Kintense which is a real-time system for detecting aggressive actions from streaming 3D skeleton joint coordinates obtained from Kinect sensors. In two multi-person households it achieves up to 90% accuracy in action detection.

III. Proposed Pipeline

According to [13] human activity can be categorized into four different levels: gestures, actions, interactions and group activities. This paper will focus on the actions and interactions category. Several datasets are available from different sources like LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information) dataset [14], CMU (Carnegie Mellon University) MoCap dataset[1], MSR-Action3D and MSRDaily-Activity3D dataset [1].

In a real world situation we expect to have a subject perform a sequence of actions instead of a single isolated action like portrayed in the previous datasets. With this in mind we recorded a dataset containing sequences of actions performed by a 12 different subjects. We used Kinect indoors with artificial lighting to record the dataset with sequences of combat movements composed of 8 different actions: *right-punch; left-punch; elbow-strike; back-fist; right-front-kick; left-front-kick; right-side-kick; left-side-kick*. Using combinations of those 8 actions we created 6 distinct sequences (each sequence contains 5 actions). Of the 12 subjects recorded, each subject performed 6 different sequences. A total of 72 sequences, 360 actions was recorded. The data is recorded in .xed files which contains RBG, depth and skeleton information, and also in .csv format containing only the skeleton data. Kinect is able to track 20 joints of a subject's skeleton. Skeleton frames are generated at the rate of 30 frames per second, and each frame consists of the 3D coordinates of 20 body joints along with their tracking states (tracked, inferred, or not tracked). The dataset [2] is available for public usage. Although we recorded RGB and depth information, our framework relies solely on the position of the skeleton joints to extracts relevant features.

A modular framework was built with several task-oriented modules organized in a workflow (Fig.1) as follows:

1: **Feature Extraction**: extract features (absolute speed) that will be used to automatically divide a sequence in temporal segments

2: **Temporal Segmentation**: automatically find temporal segments that represent the actions of the sequence

3: **Feature Extraction**: extract meaningful features (3D velocity, joint angle and bone orientation) for temporal segment found

4: **Clustering and labeling**: use clustering to automatically group similar actions and thus label them (in the control experiments this task is done manually in a frame-by-frame basis)

5: **Feature Extraction**: extract meaningful features (3D velocity) for each labeled action

6: **Dedicated classifiers**: train a classifier per action using the previous labeling

7: **Action Recognition**: recognize an action in real-time

A. Temporal Segmentation

In our previous work [15] we proved that given a sequence of contiguous actions (Fig. 2) it is possible to automatically divide the sequence into what we called temporal segments that correspond to individual actions, and that these actions correspond relatively well to our own intuitive classification. A temporal segment is a sub-set of a sequence corresponding to a particular action being performed in that time frame. The sequences of our dataset are composed by 5 different actions, ideally our temporal segmentation algorithm should divide the sequence in 5 segments each corresponding to an action.

[1] http://mocap.cs.cmu.edu/

[2] https://github.com/DavidJardim/precog_dataset_16

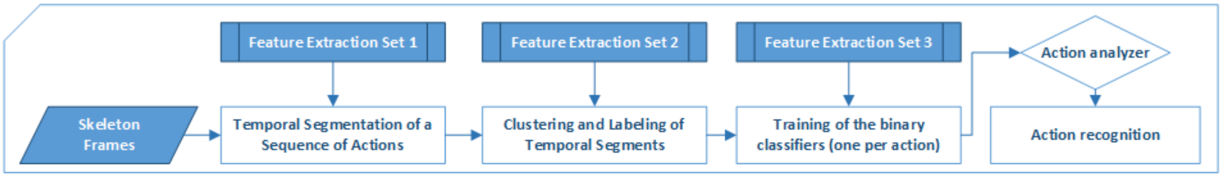
Feature Extraction Set 1
Feature Extraction Set 2
Feature Extraction Set 3
Action analyzer
Skeleton Frames
Temporal Segmentation of a Sequence of Actions
Clustering and Labeling of Temporal Segments
Training of the binary classifiers (one per action)
Action recognition

Fig. 1. Modular framework for action recognition

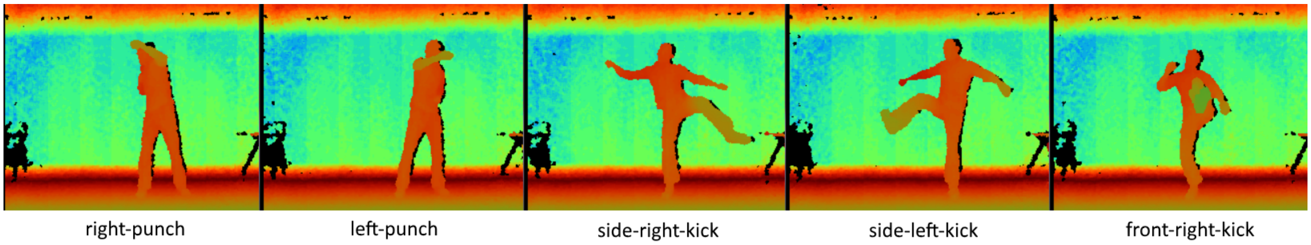

Fig. 2. Example of a recorded sequence with five actions (depth view)

Figure 3 illustrates an example result of our automatic segmentation method. Each color of the plot represents a temporal segment to which we assigned a joint as being the dominant joint for that action. We obtained 5 temporal segments which successfully correspond to the number of actions that the sequence contains, in this case: *right-punch; left-punch; side-right-kick; side-left-kick; front-right-kick*.

B. Sampling

In order to perform sampling the program selects all the temporal segments found, ideally 5 per sequence which corresponds to the number of actions that compose the sequence. Then the most active joint of the skeleton is assigned to that segment. Based on the window-frame of the segment found for a specific joint, we create new temporal segments for the remaining joints on the same exact window-frame. This can be portrayed as stacking the joints timeline one on top of another and making vertical slices to extract samples of data that correspond to temporal segments where an action has occurred.

C. Action labeling

An action can be seen as a sequence of poses over time. Each pose respects certain relative positions and orientation of joints of the skeleton. Based on the positions and orientations of the joints we extracted several features that will be used to model the movements performed by the subjects. We have experimented with features like absolute speed, velocity, joint angle and bone orientation that will be used to constitute the feature vectors for the clustering algorithm. We concluded that K-Means performed better when a combination of features were used. The results presented below extend those presented in [15] not only because the impact on supervised sequence classification is tested, but also because different features are used for the action clustering (using K-Means).

Experiments were made with new features like angle of the joints, bone orientation and other clustering algorithms. From the pool of clustering algorithms used the one which had the best performance was Hierarchical Clustering. Some research [16] refers that K-Means is usually more efficient in terms of its run-time, specially when dealing with large datasets. On the other hand Hierarchical Clustering, although slower in execution, has better clustering results. Since our dataset is relatively small and we are performing clustering on sub-sets of identical sequences performed by different subjects Hierarchical Clustering revealed more appropriated obtaining better results in the tests (Table I). Still, as seen in previous results obtained with K-Means, results show an understandable confusion between different (although similar) actions of the same body part (right leg), that are classified in the same clusters. Our efforts to distinguish clearly and unequivocally between similar movements of the same limb have not been entirely successful so far. This can be seen as noise in the following Supervised Learning process.

TABLE I
HIERARCHICAL CLUSTERING RESULTS FOR SEQUENCE 1

Action	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Right punch	100,0%	0,0%	0,0%	0,0%	0,0%
Left punch	0,0%	100,0%	0,0%	0,0%	0,0%
Front right kick	0,0%	0,0%	75,0%	0,0%	25,0%
Side right kick	0,0%	0,0%	16,67%	0,0%	83,33%
Side left kick	0,0%	0,0%	0,0%	100,0%	0,0%

D. Action classification

At this point, using our temporal segmentation approach and an off-the-shelve algorithm to perform action clustering, we were able to automatically assign a label to an action. Based on the assigned cluster and the set of available actions, a procedure was created to replace the cluster with the corresponding action. This allowed us to create an automatically labeled training set. In order to verify the accuracy of our automatically labeled training set, the original dataset was

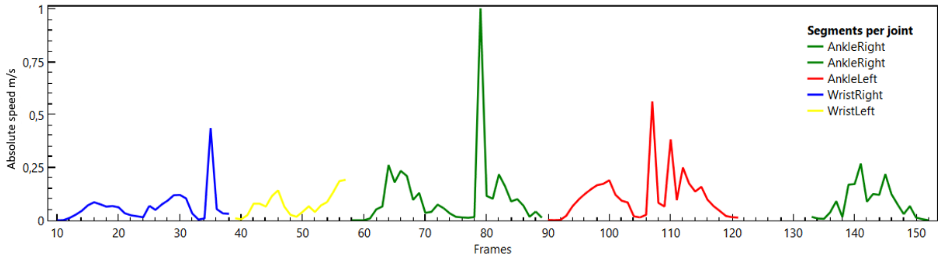

Fig. 3. Visual representation of our action segmentation method

manually labeled to be used as our ground truth. The main goal of this paper is to compare the performance of our action recognition framework trained with data automatically labeled versus data manually labeled. To verify our hypothesis we intend to use several supervised learning models, like artificial neural networks (ANN), support vector machines (SVM) and random decision forests. Also *k*-fold cross-validation will be used to sample the data in randomly partitioned *k* equal sized sub-samples with a single sub-sample being left out as a test-set.

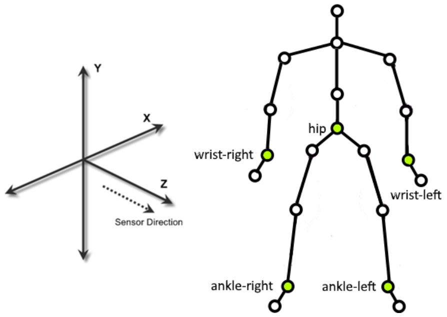

Fig. 4. Visual representation of skeleton joints selected for feature extraction

As previously said, Kinect is able to track 20 joints of a subject's skeleton. Of those 20 joints, only four were selected to extract features (wrist-right; wrist-left; ankle-right; ankle-left) as shown in Fig 4. The 3D coordinates are with respect to a frame of reference centered at Kinect. Frames from the camera are converted into feature vectors which are invariant to relative position and orientation of the body. We achieved this by re-calculating all the joints positions relative to the hip joint.

One of the main tasks in any Machine Learning problem is to select the adequate set of features to represent the learning examples. Feature vectors are a numerical representation of an object (in this case an action). An action can be seen as a sequence of poses over time. Each pose respects positions and orientation of joints of the skeleton. To reduce the dimension of the feature vector we divided the temporal segment representing an action in 6 sub-segments containing approximately the same number of frames. For each sub-segment we calculated the 3D average velocity. This was repeated for each of the selected joints in Fig. 4. The feature vector dimension is $6 * 3 * 4 = 72$ and it will be used to train the classifiers.

TABLE II
CLASSIFICATION ACCURACY (%) FOR DIFFERENT CLASSIFIERS TRAINED WITH ALL ACTIONS USING MANUALLY LABELED DATA AND CORRESPONDING STANDARD DEVIATION BETWEEN TRIALS

Action	MLP	SVM	RF
right-punch	69,80 ±0,83%	72,08 ±0,17%	80,89 ±1,00%
left-punch	70,22 ±0,77%	72,03 ±0,21%	81,30 ±1,28%
front-right-kick	70,11 ±1,03%	72,01 ±0,20%	81,37 ±0,88%
front-left-kick	69,99 ±0,92%	72,11 ±0,17%	81,39 ±1,11%
side-right-kick	69,97 ±0,72%	72,07 ±0,19%	81,57 ±0,83%
side-left-kick	70,10 ±0,87%	72,10 ±0,17%	81,54 ±1,67%
backfist	69,88 ±0,80%	72,10 ±0,17%	80,97 ±0,81%
elbow-strike	70,04 ±0,75%	72,04 ±0,20%	81,12 ±0,80%

IV. EXPERIMENTS

In this section, we explain our experimental results using our dataset. Several classifiers will be trained to compare the results of using manually labeled or automatically labeled training sets. These classifiers are trained to recognize an action from a sample of skeleton frames provided by Kinect on which we perform feature extraction from 4 main joints (*wrist-right, wrist-left, ankle-right and ankle-left*). The results obtained in the tables below represent the average recognition accuracy values of 30 trials using random seed values with the corresponding standard deviation to quantify the amount of variation in performance that occurred in each trial.

We experimented with the following classifiers: Multilayer Perceptron (MLP) as in [17]; Support Vector Machines (SVM) using pairwise classification [18] and Random Forests (RF) which are a combination of tree predictors [19]. Table II shows the average classification obtained for several classifiers using all the actions in the manually labeled training set. The results are significantly below the state-of-the-art [3] accuracy of 90%, obtained using binary classifiers. There is also a clear difference in performance between the classifiers. RF has nearly 10% increase in performance compared to MLP and SVM.

The next experiment was to train eight binary supervised classifiers using manually labeled data for recognizing the eight aggressive actions contained in our dataset. These classifiers are binary and the training-set for each classifier contains each instance of a given action labeled as positive examples and all other actions labeled as negative examples. Each

TABLE III
CLASSIFICATION ACCURACY (%) OF THE BINARY CLASSIFIERS USING MANUALLY LABELED DATA AND CORRESPONDING STANDARD DEVIATION BETWEEN TRIALS

Action	MLP	SVM	RF
right-punch	94,24 ±0,44%	91,52 ±0,17%	90,08 ±0,37%
left-punch	89,09 ±0,44%	92,50 ±0,26%	92,21 ±0,37%
front-right-kick	88,14 ±0,96%	87,95 ±0,21%	93,20 ±0,53%
front-left-kick	89,96 ±0,79%	90,42 ±0,28%	91,97 ±0,48%
side-right-kick	91,22 ±0,16%	91,92 ±0,07%	94,53 ±0,57%
side-left-kick	83,62 ±0,97%	84,76 ±0,23%	91,74 ±0,51%
backfist	92,55 ±0,32%	92,77 ±0,00%	93,58 ±0,46%
elbow-strike	95,02 ±0,28%	96,66 ±0,00%	96,66 ±0,00%

TABLE IV
CLASSIFICATION ACCURACY (%) OF THE BINARY CLASSIFIERS USING AUTOMATIC LABELED DATA AND CORRESPONDING STANDARD DEVIATION BETWEEN TRIALS

Action	MLP	SVM	RF
right-punch	83,82 ±0,81%	88,29 ±0,16%	89,40 ±0,48%
left-punch	82,43 ±1,31%	90,20 ±0,00%	90,84 ±0,33%
front-right-kick	81,22 ±0,74%	90,75 ±0,07%	90,00 ±0,49%
front-left-kick	89,99 ±0,76%	87,91 ±0,13%	90,99 ±0,25%
side-right-kick	82,80 ±1,18%	87,88 ±0,07%	89,57 ±0,57%
side-left-kick	84,99 ±0,86%	90,28 ±0,05%	90,56 ±0,68%
backfist	83,09 ±1,44%	87,60 ±0,00%	90,05 ±0,41%
elbow-strike	95,90 ±0,31%	96,83 ±0,00%	96,83 ±0,00%

classifier is trained to distinguish one action from all others. This approach (binary classifiers) produced the best results in [3] using SVM classifiers. In Table III we can see that the advantage of using binary classifiers is obvious with average accuracies above 91% in recognizing an action. In this case the difference in accuracy between classifiers is reduced, but again RF manages to obtain the best results.

Table IV shows the results of the repetition of the previous experiment with the fundamental difference of using a training set labeled by our automatic labeling pipeline. Again the MLP classifier has the worst performance and RF performs the best. MLP in comparison to the values of Table III has the largest decrease in performance, in some cases more than 10%. Concerning SVM and RF the difference is much less, never surpassing 3%.

As expected the usage of automatic labeling has affected the accuracy of the classifiers. This can be explained by the error that our automatic labeling method introduces. The temporal segmentation method can add frames to a segment that do not belong to that action or remove frames from a segment that do still belong to the same action where the segmentation performed when using the manually labeled data is inferred from the labeled frames. Also our clustering and labeling method confuses similar actions (Tab. I) which can lead to an incorrect labeling of the actions. Nonetheless, the difference is relatively small, and depending on the application, it could be negligible and remove completely the necessity of having to rely on human resources to manually label data. Finally, in Table V we calculate the difference in performance for each classifier accuracy using manually labeled data and automatically labeled data. In some cases the classifier that was trained using automatically labeled data outperformed his counterpart.

TABLE V
DIFFERENCE IN PERFORMANCE (%) BETWEEN THE TWO APPROACHES (MANUAL VS AUTOMATIC) FOR EACH BINARY CLASSIFIER PER ACTION

Action	MLP	SVM	RF
right-punch	-10,42 %	-3,23 %	-0,68 %
left-punch	-6,66 %	-2,30 %	-1,37 %
front-right-kick	-6,92 %	2,80 %	-3,2 %
front-left-kick	0,03 %	-2,51 %	-0,98 %
side-right-kick	-8,42 %	-4,04 %	-4,96 %
side-left-kick	1,37 %	5,52 %	-1,18 %
backfist	-9,46 %	-5,17 %	-3,53 %
elbow-strike	0,88 %	0,17 %	0,17 %
average	-4,95 %	-1,09 %	-1,97 %

V. CONCLUSION

In this paper, we described a framework capable of recognizing human actions using supervised classifiers trained with automatically labeled data. We used our own dataset of sequences of actions recorded with Kinect. We performed automatic temporal segmentation of a sequence of actions, automatically labeled the actions using a clustering algorithm (where we improved our previous results), and compared the performance of several supervised classifiers used on the state-of-the-art to recognize human activity, using manually labeled and automatically labeled training sets. Previous studies, extended here, showed how clustering and filtering techniques can be combined to achieve unsupervised labeling of human actions recorded by a camera with a depth sensor which tracks skeleton joints that will be used to train a supervised classifier. This work clarified the difference between using manually, versus automatically labeled data for simple action sequences such as the ones used. The objective was to measure the impact of the noise introduced by automatic labeling on action classification. Of the three supervised classifiers used (MLP, SVM and RF) to recognize an action, Random Forests had the best performance. The best possible results were obtained when manually labeled data was used to train the classifiers. Using automatically labeled data did introduce a decrease in performance due to the error that our automatic labeling method introduces. Nonetheless, the difference is relatively small and, depending on the application, the use of automatic labeling can indeed be considered as an option.

Our results proved that, for a dataset of simple combat actions, obtained with a standard Kinect camera with no special acquisition conditions, a temporal segmentation and clustering algorithm can be used to label identical actions performed by different users. Also, we have established that this labeling can be used to train supervised classifiers that will be capable of identifying specific actions in a RGB-D video feed without relying on any human resources, with a minor loss of precision relative to training with human labeled data.

We would like to replicate these results using other existing datasets. Recent research had its efforts shifted to the problem of action prediction. As future work, we would like to add action prediction capabilities to our action recognition framework using conditional random fields, enabling prediction of a future action executed by a subject in the context of a sequence of actions like in our dataset.

Acknowledgment

This research was sponsored by a joint scholarship between Fundação para a Ciência e a Tecnologia (FCT) and Microsoft Portugal under the grant SFRH/BDE/52125/2013. The project was hosted by Microsoft Language Development Center (MLDC) in Lisbon, Portugal.

References

[1] W. Niu, J. Long, D. Han, and Y. F. Wang, "Human activity detection and recognition for video surveillance," in *International Conference on Multimedia and Expo*. IEEE Press, 2004, pp. 719–722.

[2] S. S. Intille and A. F. Bobick, "A Framework for Recognizing Multi-agent Action from Visual Evidence," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference*, no. 489. AAAI Press, 1999, pp. 518–525. [Online]. Available: http://dl.acm.org/citation.cfm?id=315149.315381

[3] S. Nirjon, C. Greenwood, C. Torres, S. Zhou, J. a. Stankovic, H. J. Yoon, H. K. Ra, C. Basaran, T. Park, and S. H. Son, "Kintense: A robust, accurate, real-time and evolving system for detecting aggressive actions from streaming 3D skeleton data," in *International Conference on Pervasive Computing and Communications*. IEEE Press, 2014, pp. 2–10. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6813937

[4] M. Popa, A. Kemal Koc, L. J. M. Rothkrantz, C. Shan, and P. Wiggers, "Kinect sensing of shopping related actions," in *Communications in Computer and Information Science*, vol. 277 CCIS. Springer, 2012, pp. 91–100.

[5] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila, "Active pedestrian safety by automatic braking and evasive steering," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1292–1304, 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5936735

[6] F. Zhou, F. D. L. Torre, and J. Hodgins, "Hierarchical Aligned Cluster Analysis (HACA) for Temporal Segmentation of Human Motion," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3. Citeseer, 2010, pp. 1–40.

[7] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Transferring skills to humanoid robots by extracting semantic representations from observations of human activities," *Artificial Intelligence*, no. June 2016, 2015.

[8] M. W??chter and T. Asfour, "Hierarchical segmentation of manipulation actions based on object relations and motion characteristics," *Proceedings of the 17th International Conference on Advanced Robotics, ICAR 2015*, vol. 270273, pp. 549–556, 2015.

[9] H. Koppula, R. Gupta, and A. Saxena, "Learning Human Activities and Object Affordances from RGB-D Videos," in *The International Journal of Robotics Research*, vol. 32, no. 8. SAGE Publications, 2013, pp. 951–970. [Online]. Available: http://arxiv.org/abs/1210.1207v2

[10] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations." in *International Joint Conference on Artificial Intelligence*. AAAI Press, 2013, pp. 2466–2472.

[11] M. a. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban, "Histogram of Oriented Displacements (HOD): Describing trajectories of human joints for action recognition," in *International Joint Conference on Artificial Intelligence*, vol. 25. AAAI Press, 2013, pp. 1351–1357.

[12] C. Jia, Y. Kong, Z. Ding, and Y. R. Fu, "Latent tensor transfer learning for rgb-d action recognition," in *Proceedings of the ACM International Conference on Multimedia*. ACM Press, 2014, pp. 87–96.

[13] J. K. Aggarwal and M. S. Ryoo, "Human Activity Analysis: A Review," *ACM Computing Surveys*, vol. 43, no. 3, pp. 1–43, 2011.

[14] C. Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandrea, C.-E. Bichot, C. Garcia, and B. Sankur, "Evaluation of video activity localizations integrating quality and quantity measurements," in *Computer Vision and Image Understanding*, vol. 127. Elsevier, 2014, pp. 14–30. [Online]. Available: http://liris.cnrs.fr/Documents/Liris-5498.pdf

[15] D. Jardim, L. Nunes, and M. S. Dias, "Automatic human activity segmentation and labeling in rgb-d videos," in *Intelligent Decision Technologies: KES-IDT 2016*. Springer International Publishing, 2016, pp. SIST 56, p. 383 ff.

[16] M. Kaur and U. Kaur, "Comparison Between K-Mean and Hierarchical Algorithm Using Query Redirection," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 7, pp. 1454–1459, 2013.

[17] M. Kubat, "Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7," in *The Knowledge Engineering Review*, vol. 13, no. 4. Cambridge Univ Press, 1999, pp. 409–412.

[18] J. C. Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization," in *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998, pp. 185 – 208. [Online]. Available: http://research.microsoft.com/apps/pubs/?id=68391

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.