

# Image recognition of multi-perspective data for intelligent analysis of gestures and actions

Egbert Gedat\*, Pascal Fechner, Richard Fiebelkorn, Jan Vandenhouten, Ralf Vandenhouten

## Zusammenfassung

Das Projekt BERMUDA konnte im Januar 2015 begonnen und nach knapp drei Jahren im August 2017 erfolgreich abgeschlossen werden. Es wurden ein technischer Aufbau und eine Bildverarbeitungs- und Analysesoftware entwickelt, mit denen sich multiperspektivische Videos aufzeichnen und auswerten lassen. Basierend auf zwei in größerem Abstand gewinkelt positionierten Kameras wurden synchrone Videos sowohl im Labor als auch in realen Umgebungen aufgenommen. Die Auswertung umfasst die Hintergrundeliminierung, die Körperteilklassifikation, ein Clustering, die Zuordnung zu Personen und schließlich die Rekonstruktion der Skelette. Ausgehend von den Skeletten wurden Methoden des maschinellen Lernens zur Erkennung der Haltungen und darauf aufbauend zur Gestenerkennung entwickelt. Beispielfähig konnte die im Sicherheitskontext relevante Handlung des Schlagens mit einer Genauigkeit von 51,3% und einer Trefferquote von 60,6% erkannt werden.

## Abstract

The BERMUDA project started in January 2015 and was successfully completed after less than three years in August 2017. A technical set-up and an image processing and analysis software were developed to record and evaluate multi-perspective videos. Based on two cameras, positioned relatively far from one another with tilted axes, synchronized videos were recorded in the laboratory and in real life. The evaluation comprised the background elimination, the body part classification, the clustering, the assignment to persons and eventually the reconstruction of the skeletons. Based on the skeletons, machine learning techniques were developed to recognize the poses of the persons and next for the actions performed. It was, for example, possible to detect the action of a punch, which is relevant in security issues, with a precision of 51.3% and a recall of 60.6%.

## 1. Introduction

The aim of the project is the automated recognition of actions in a security context, such as violent behavior in public places like airports and train stations. Therefore, 3D data of persons in videos are required. Two methods are available to achieve this, either by the use of depth images generated by infrared light patterns, e.g. Microsoft Kinect (Kinect 2 camera for Windows (Microsoft Corp., Redmond, WA, USA) with OpenNI (<http://structure.io/openni>)), or stereoscopic viewing. The latter technique was extended to multi-perspective viewing in this project. Hence it was possible to position the cameras further from each other, possibly compensating for occlusions of the persons. The image preprocessing was carried out in 2D up to

the different body parts that were detected in the images and assigned to the persons. With a triangulation of corresponding points, the 3D-skeletons of the persons in the videos were obtained. There are some methods known for the analysis of such skeletons with respect to pose recognition for one person (Andriluka et al. 2009, Buys et al. 2014, Tran & Trivedi 2008), for multiple humans (Andriluka et al. 2012, Belagiannis et al. 2014, Luo et al. 2010) and for action recognition (Bigontina et al. 2015, Laptev et al. 2008). The pose of the humans was assigned a pose letter (Zhou et al. 2016) after pose recognition. The pose recognition was done with an artificial neural network. Then, in the style of speech recognition algorithms (Rabiner 1989, Gales & Young 2007, Trentin & Gori 2003), the sequence of the pos-

es, i.e. letters, was analyzed by hidden Markov models (Kulić et al. 2008) to find the best fitting action or word for the action being observed. As a result, at each point in time the most probable action of the person was obtained.

## 2. Image preprocessing and skeleton reconstruction

The scope of image preprocessing covers all steps on 2D images up to the point of calculation of the body part joints and centers of gravity. The technical details of the camera set-up are given in section 4. The background elimination, the body part classification and the clustering were written in Java using OpenCV (OpenCV, <https://opencv.org>) and the assignment of body parts to persons and the reconstruction of the

\* corresponding author

skeletons were written in Matlab (MathWorks, Natick, MA, USA). First, the humans must be found in the frames of the videos – or vice versa, the background must be eliminated. The background elimination was realized by an OpenCV implementation of Gaussian mixture-based background/foreground segmentation (Zivkovic 2004). The algorithm distinguishes three classes: background, foreground and shadow. Multiple application of the algorithm with different parameters and weightings yielded greyscale foreground probability images. Due to strongly varying light, in some cases a normalization of the image overall intensity was performed. The thresh-

olded foreground pixels were filtered by an OpenCV human detection algorithm and the remaining pixels were subjected to the body part classification.

The body part classification was done pixel-wise on each foreground pixel. The algorithm was based on a decision forest as proposed by Shotton et al. (Shotton et al. 2013). That algorithm was firstly programmed and trained as given there in 3D and secondly extended to 2D data. The 2D body part classification was based on the extraction of silhouettes (Bigontina et al. 2015) of the humans in the videos, i.e. the foreground pixels. The silhouettes were of ternary nature in the simplest case and

of greyscale nature with more runs of the background/foreground segmentation. The training of the decision forests was done with manually classified data. It took several days of computing time on a standard personal computer. The algorithms trained this way assigned one of 21 predefined body parts to each pixel (Gedat et al. 2016). The point clouds that were found for each body part were arranged into different areas of the same class by a clustering process. The clustering was done with a nearest-neighbor search. The resulting clusters were the candidates for the assignment to the skeletons.

In principle there can be more or fewer body part clusters for one body part than humans in the frames of the videos. A selection of the optimal body part clusters and the assignment to the different persons were solved with graph theoretical methods. Initially a graph was defined that describes the human skeleton (Gedat et al. 2016). The graph consists of six part-graphs, that all stem from the neck and emanate either left or right into the head, the arm and the leg, respectively. The best body part clusters of this k-shortest-disjoint paths problem were selected by an adaption of the Suurballe algorithm (Suurballe 1974) for k-shortest-disjoint paths search using a cost function based on heuristic anatomic distances of body parts (Gedat et al. 2016, 2017b).

For an improved anatomical representation, an anatomical correction of the body parts was established. For upper and lower arms and legs, the body part point clouds were processed with a principle component analysis yielding proximal and distal joints of the body parts (Gedat et al. 2016). For all other body parts, the center of gravity of the point cloud was used and shifted with a heuristic metric in medial direction to account for the superficial view of the volumes. At this point of the procedure in each of the (two) multi-perspective images, a number of skeletons had been determined in 2D. The number of skeletons was derived from the number of persons detected by the OpenCV method used for detection of humans in the images. Before the 3D skeleton reconstruction can be applied, an as-

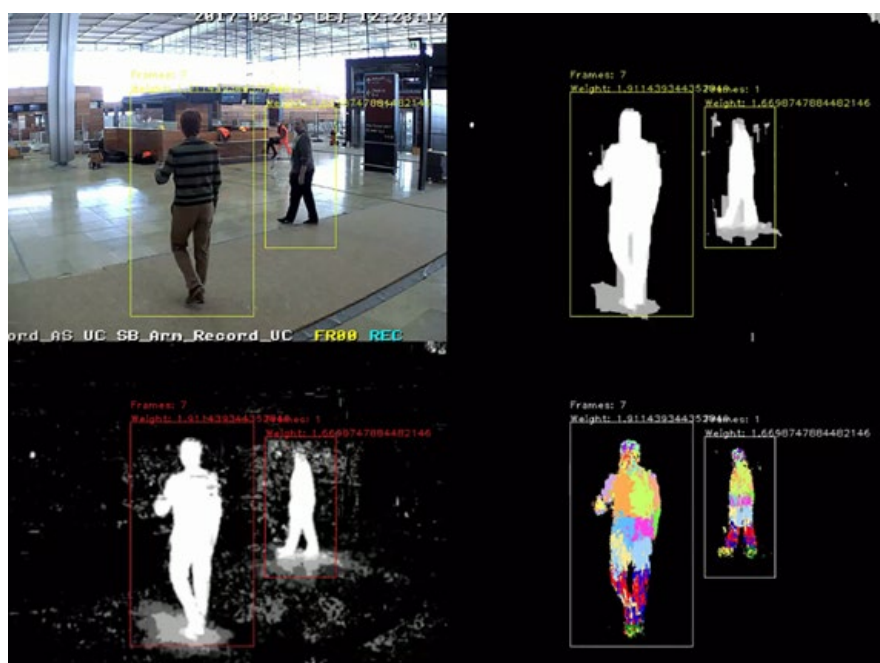


Fig. 1) Example of the preprocessing up to the 2D body part classified image (bottom right) from the BER experiment (see section 4). Original image (top left), background/foreground greyscale image (bottom left) and thresholded background/foreground image (top right). The frames are the bounding boxes from the OpenCV person detection algorithm.

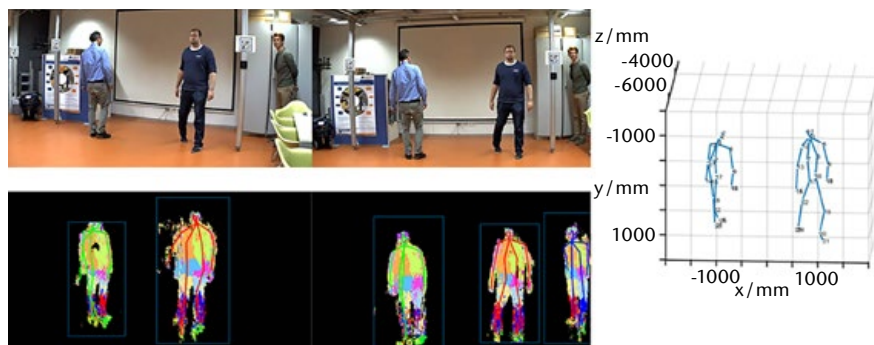


Fig. 2) Example of the skeleton reconstruction from the Lab experiment (see section 4). Left side: Top: left and right camera images. Bottom: Body part classified images with reconstructed skeletons and color code for the assignment of skeletons to persons and person detection bounding boxes. The third person was not detected in the left image. Images are cropped at top and bottom. Right side: The corresponding reconstructed 3D skeletons.

segment of the 2D-skeletons to the humans in the images had to be accomplished. This was done by an estimation of the 3D coordinates of the center of gravity of the skeletons with the height of the OpenCV human detection bounding box as body height and the intercept theorem. Then the Hungarian method, also called the Kuhn-Munkres algorithm, was applied to calculate the minimal global pairwise distance of all skeletons and the skeletons were accordingly assigned to the humans.

Now the reconstruction of the 3D-skeletons could be performed by triangulation of corresponding body part centers and joints. Therefore, the intrinsic and extrinsic matrices of the camera system must be known; see section 4.

To be able to work independently of scale and distance in the following steps, the 21 3D-skeleton coordinates were transformed to 16 angles that characterize the skeletons.

### 3. Pose and action recognition with machine learning methods

Our pose and action recognition works with skeleton angles as input. These can originate from any applicable first-hand data, i.e. a software model, laboratory videos, or real-life videos acquired with an arbitrary setup of 2D or 3D cameras. For pose and action recognition, Matlab was used, including the neural networks toolbox.

The pose recognition method is based upon a predefined poses alphabet (Gedat et al. 2017a). The poses alphabet is in principle infinite in size and must be adjusted to a set of expected poses in the data to be analyzed. The poses alphabet used in this study is given in Tab. 1.

For the pose recognition, an artificial neural network (ANN) was trained and applied. The ANN consisted of 32 input neurons, one hidden layer of 15 sigmoid neurons and one output layer of 10 softmax neurons (Fig. 4). The 32 input neurons correspond to the 16 skeleton angles and one flag for each angle indicating if it was present in the input vector. The 10 output neurons correspond to the ten poses present in the training data. For train-

**Tab. 1:** The poses alphabet. Images for each pose are given in Gedat et al. 2017a.

pose letter	pose description
a	walking – right leg in front
b	walking – left leg in front
c	punch – right arm stretched forward
d	punch – left arm stretched forward
e	hook – right arm bent forward
f	hook – left arm bent forward
g	both arms in front of body
i	walking – stand straight on left foot
j	walking – stand straight on left foot
k	stand straight
n	stand – arms bent
o	stand – hands in front of chest
p	stand – left hand in front of chest, right arm bent
q	stand – right hand in front of chest, arm bent

ing of the ANN, key frames (Zhou et al. 2016) of a MOCAP (CMU Graphics Lab Motion Capture Database, <http://mocap.cs.cmu.edu/>) software model (see Section 4) were manually classified according to the poses alphabet in Tab. 1. For the ANN, Matlab default settings were used, i.e. scaled conjugate gradient back propagation, splitting of the data into 70 % training, 15 % testing and 15 % validation on a random choice basis.

This trained ANN delivers the probabilities of the poses from the pose alphabet for any input skeleton. For further processing in the action recognition module, the most probable pose was chosen and saved as a letter in the pose sentence.

As practiced in automated speech recognition, one separate hidden Markov model (HMM) was calculated for each action that shall be detected. A HMM consists of three matrices: the start matrix, the transition matrix and the emission matrix. Given these matrices, a score for the particular action can be calculated. The three matrices

are not based on training but can be readily calculated from classified data. The training data were manually classified with respect to poses from the poses alphabet and actions performed including

- (i.) walking – left step
- (ii.) walking – right step
- (iii.) boxing – right stroke
- (iv.) standing, including various smaller motions.

The matrices were calculated according to the pseudocode in Fig. 3.

For the evaluation of the letter input from the neural network an averaging over three time steps was performed, the scores were multiplied with a heuristic duration-of-action compensating exponential function and a cut-off score was defined for each action, below which it was not further considered for that point in time. After these steps, the action with the greatest score was chosen to be the detected action.

```

for each action  $A_k$ 
  for all frames  $F_i$ 
    if  $F_i$  shows action  $A_k$ 
      if  $F_{i-1}$  not shows  $A_k$ 
        start $_k$ (state( $F_i$ ))++
      if  $F_{i+1}$  shows  $A_k$ 
        transmit $_k$ (state( $F_i$ ), state( $F_{i+1}$ ))++
        emit $_k$ (state( $F_i$ ), emission( $F_i$ ))++
  start $_k$ =start $_k$ /sum(start $_k$ )
  transmit $_k$ =transmit $_k$ /sum(transmit $_k$ )
  emit $_k$ =emit $_k$ /sum(emit $_k$ )

```

Fig. 3) Pseudocode for the calculation of the hidden Markov model matrices. Red font indicates the three matrices needed for the HMMs. The last three lines are normalization to a total probability of 1 for each matrix.

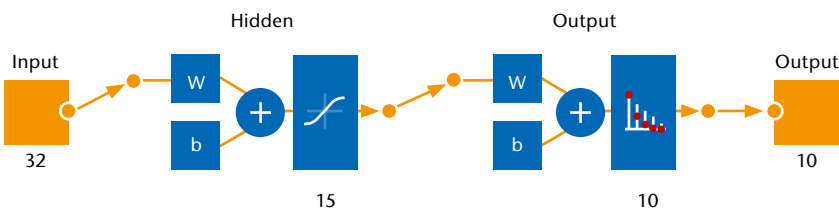


Fig. 4) Schematic Matlab representation of the used artificial neural network with the weights matrices  $W$  and biases vectors  $b$ .

## 4. Experiments

### I. MOCAP software model

To set up the machine learning tools, a software model of human actions was rendered in Blender (The Blender Foundation, Amsterdam, The Netherlands) for selected MOCAP motions displaying strokes and walking. Therefore, the MOCAP motions, which consist of joint coordinates for each time frame, were scaled down to 3.75 fps with the software *bvhacker* ([www.bvhacker.com](http://www.bvhacker.com)), given a skin with the software *MakeHuman* (*MakeHuman* team, [www.makehuman.org](http://www.makehuman.org)) and eventually rendered in Blender with a texture resembling the color code of the body part classification tool described above from two views in an angle of 45 degrees. For details, see Gedat et al. 2016.

For the training of the ANN 370 key frames of the MOCAP software model were manually classified with respect to the pose of the person. The performance of the training was – depending on the randomly separated training/validation/testing data – around 85 % correct pose assignments.

For the calculation of the HMM matrices 1,891 frames of the MOCAP software model were manually classified with respect to poses and actions performed. For details, see Gedat et al. 2017a.

The results for action recognition from Gedat et al. 2017a with 83.2 % precision and 83.7 % recall are very promising. However, these were achieved with a software model and not real-life data.

### II. BER – Real life scenario at the Berlin airport terminal

The software was tested in a real-life scenario at the Berlin International Airport (BER) terminal during the construction period. A set-up was used consisting of two pairs of *Mobotix* (*Mobotix* AG, Langmeil, Germany) MX-S15 dual cameras, each with two L32 lenses. They were mounted pairwise on the ends of 1 m long crossbars mounted on tripods. Each of the MX-S15 was equipped with an embedded Linux operating system which communicated with a browser on a *Lenovo* X230 ThinkPad laptop via LAN-cable. The two connected L32 cameras of each MX-S15 were synchronized by the embedded camera software. The RGB image size was 640 × 480 pixels. This mobile set-up was mounted in the terminal in different arrangements with respect to view angles, distances and heights of the cameras. For each arrangement, a calibration procedure was performed with a checkerboard being recorded in roughly thirty dif-

ferent positions for calculation of the intrinsic and extrinsic matrices of the cameras with a Matlab camera calibration tool. Then a set of predetermined actions, including (i) walking, (ii) waving one hand, (iii) boxing, (iv) avoiding being hit by a stroke, were performed by one to five persons. One example of the BER session is shown in Fig. 1. The data were recorded onsite and transferred to computers in the laboratory for further analysis.

The on-site experiment at BER generated 23.7 GB of image data. The data were used to successfully calibrate the cameras in 5 of 7 set-ups, to develop and to train the 2D body part classifier, to deal with synchronization issues and to develop methods for the assignment of the skeletons to persons (see above in section 3).

Ultimately, it was not possible to satisfactorily process the data with respect to action recognition, but some key issues could be learned for the preparation of the final laboratory session: (i) varying background light must be avoided, (ii) background color similar to foreground color should be avoided (where possible), (iii) synchronization of the cameras must be improved for camera heads not connected to the same camera board, (iv) overlap of humans must be avoided, because they would be merged into one common bounding box.

### III. Lab - Real life movies in the laboratory

With the knowledge obtained from the BER session, a laboratory set-up was realized that could be used for real life laboratory action recognition of punches. One pair of L32 camera lenses with a *Mobotix* MX-S15 camera was mounted in multi-perspective arrangement with an angle of approximately 30 degrees and a mutual distance of about 2 meters about 3 to 5 meters away from the spot of action to be recorded. Light from outside the laboratory was excluded by blinds. Clothing was chosen not to have the color of the background. Three volunteers performed, first separately, then jointly predetermined actions, namely, (i) standing, (ii) walking, and (iii) boxing. One example of the Lab session is shown in Fig. 2.

Six movies were recorded, the first four with one single person performing the actions for training the machine learning algorithms, the fifth with the persons performing the actions one after the other and the sixth with the persons acting at the same time. The two latter were intended for testing of the method after the training phase had been finished.

The training of the ANN with the new data yielded an accuracy of only 30 %. After ignoring left and right (i.e. accounting for symmetry), an increase to 50 % was possible. Further improvement was not possible. However, with the use of the Blender software model, trained ANN and HMM punches could be well detected in the fifth video of the Lab session. Only scenes with at least 10 consecutive valid skeletons were evaluated with respect to overlap between manual classification and calculated action. The results of the action detection are given in Tab. 2. There was some variance in the accuracy of the method with respect to the three humans in the videos. The results are given in Fig. 5.

Although the accuracy was not high enough to detect all strokes and only strokes, the results indicate that it is possible to use the proposed method for action recognition. However, further work is necessary.

### 5. Discussion and outlook

The action recognition with the software model (precision 83.2 % and recall 83.7 %) was very effective compared to values in previous studies. Kang and Wildes report a mean accuracy for various algorithms applied to the Human Motion Data-

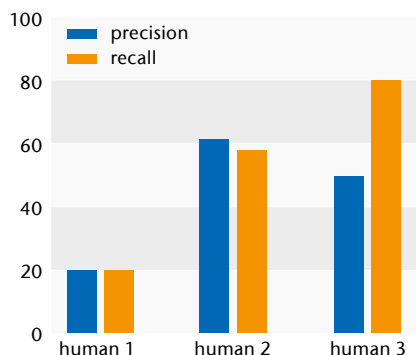


Fig. 5) The accuracy of the method for the three humans in the video.

Tab. 2: Results of the action recognition. N is the number of punches in the movie, TP is true positive, FP is false positive and FN is false negative.

N	TP	FP	FN	precision	recall
33	20	19	13	51.3	60.6

base (HMDB51, see (Kang & Wildes 2016)) of  $62.7 \pm 6.4$  %, to the University of Central Florida data (UCF101, see (Kang & Wildes 2016)) a mean accuracy of  $88.5 \pm 7.4$  % and to the Stanford University's Sports-1M dataset (Sports-1M, see (Kang & Wildes 2016)) of  $67.0 \pm 8.6$  % (Kang & Wildes 2016). However, the values obtained with the software model are not directly comparable to those given by Kang and Wildes, because in this experiment idealized data were used that exhibited minimal flaws. But it can be deduced from the good accuracy that the method proposed here, combining a neural network with hidden Markov models and all preprocessing steps, is in principle capable of doing the job of action recognition.

The hidden Markov model appears to operate in the way of a smoothing filter in the time domain. For example, the pose sentence obtained from the neural network for one walking action was `posesentence{1} = 'bjiii-jaaaaajjcbnbbiiijaaanajbbbbbiiiaa'` and the corresponding action recognition was `actions{1} = '-llrrrrrrrrllllllllllrrrrrrrrllllllrrr'` with 'l' representing a left step and 'r' a right step. Although there are some deviations from the ideal walking pose sentence, composed of sequences like `b i a j b i a j` and so on, the action recognition correctly gives alternating left and right steps. The two dashes at the beginning of the pose sentence are due to the averaging over three time steps.

Comparing the results of the laboratory experiment to the values given by Kang and Wildes, the accuracy of 51.3 % precision and 60.6 % recall is not far from literature values. However, a drastically minimized action repertoire – detecting only punches – was used. Idealized laboratory conditions were also applied to minimize errors. On the other hand, the unwanted conditions which should be avoided to make the system work more reliably could be learned from the BER experiment.

In the next step, the idealizations that were necessary to detect actions in the laboratory environment must be reduced by transferring the duty to the background/foreground segmentation and to the classification algorithm. The background/foreground segmentation must be independent of the color of clothing, while changing light must be compensated. The body part classification must work on silhouettes containing more than one person that can in principle not be separated in 2D images. Also, the people detection algorithm, which did not find all persons and also had several false positives, should be improved. Further, the range of actions to be detected must be enlarged step by step, implying also an extension of the poses alphabet.

The detection of strokes in the laboratory experiment was achieved with the machine learning tools trained and calculated from the software model. Novel training with the laboratory data did not result in good accuracy. On the other hand, the variance in accuracy for the three humans in the Lab experiment may indicate only restricted applicability of the models trained with artificial data. It remains to be elucidated whether learning from idealized data is superior to learning from flaw-prone real-world data.

### Acknowledgement

This work was supported by the Federal Ministry of Education and Research in the program FHprofUnt, support code 13FH009PX3.

The MOCAP data used in this project was obtained from `mocap.cs.cmu.edu`. The database was created with funding from NSF EIA-0196217.

## REFERENCES

- Andriluka M, Roth S, Schiele B (2009) Pictorial structures revisited: People detection and articulated pose estimation. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Miami, FL, pp 1014–1021. doi: 10.1109/CVPR.2009.5206754
- Andriluka M, Roth S, Schiele B (2012) Discriminative Appearance Models for Pictorial Structures. *Int J Comput Vis* 99(3):259–280. doi: 10.1007/s11263-011-0498-z
- Belagiannis V, Amin S, Andriluka M, Schiele B, Navab N, Ilic S (2014) 3D Pictorial Structures for Multiple Human Pose Estimation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, pp 1669–1676. doi: 10.1109/CVPR.2014.216
- Bigontina A, Herrmann M, Hoernig M, Radig B (2015) Human Body Part Classification in Monocular Soccer Images. In: Paulus D, Fuchs C, Droege D (eds) Proceedings of the 9th Open German-Russian Workshop on Pattern Recognition and Image Understanding, 2014, Koblenz. Active Vision Group, Koblenz, pp 128–131
- Buys K, Cagniard C, Baksheev A, Laet T de, Schutter J de, Pantofaru C (2014) An adaptable system for RGB-D based human body detection and pose estimation. *Journal of Visual Communication and Image Representation* 25(1):39–52. doi: 10.1016/j.jvcir.2013.03.011
- Gales M, Young S (2007) The Application of Hidden Markov Models in Speech Recognition. *FNT in Signal Processing* 1(3):195–304. doi: 10.1561/20000000004
- Gedat E, Fechner P, Fiebelkorn R, Vandenhousten R (2016) Multiple human skeleton recognition in RGB and depth images with graph theory, anatomic refinement of point clouds and machine learning. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics. Conference proceedings, Budapest, Hungary. IEEE, Piscataway, NJ, pp 627–631. doi: 10.1109/SMC.2016.7844310
- Gedat E, Fechner P, Fiebelkorn R, Vandenhousten R (2017a) Human action recognition with hidden Markov models and neural network derived poses. In: 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia. IEEE, pp 157–162. doi: 10.1109/SISY.2017.8080544
- Gedat E, Fechner P, Fiebelkorn R, Vandenhousten R (2017b) Szenenanalyse und Unterscheidung der Skelette mehrerer Menschen in digitalen Bildern mit Graphentheorie durch eine k-kürzeste-disjunkte-Wege-Suche. *Wissenschaftliche Beiträge* 21:31–35. doi: 10.15771/0949-8214\_2017\_4
- Kang SM, Wildes RP (2016) Review of Action Recognition and Detection Methods. arXiv:1610.06906 [cs.CV]. Accessed 07 Dec 2017
- Kulić D, Takano W, Nakamura Y (2008) Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains. *The International Journal of Robotics Research* 27(7):761–784. doi: 10.1177/0278364908091153
- Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, pp 1–8. doi: 10.1109/CVPR.2008.4587756
- Luo X, Berendsen B, Tan RT, Veltkamp RC (2010) Human Pose Estimation for Multiple Persons Based on Volume Reconstruction. In: 2010 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, pp 3591–3594. doi: 10.1109/ICPR.2010.876
- Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77(2):257–286. doi: 10.1109/5.18626
- Shotton J, Girshick R, Fitzgibbon A, Sharp T, Cook M, Finocchio M, Moore R, Kohli P, Criminisi A, Kipman A, Blake A (2013) Efficient human pose estimation from single depth images. *IEEE Trans Pattern Anal Mach Intell* 35(12):2821–2840. doi: 10.1109/TPAMI.2012.241
- Suurballe JW (1974) Disjoint paths in a network. *Networks* 4(2):125–145. doi: 10.1002/net.3230040204
- Tran C, Trivedi MM (2008) Human body modelling and tracking using volumetric representation: Selected recent studies and possibilities for extensions. In: 2008 Second ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), Palo Alto, CA, USA, pp 1–9. doi: 10.1109/ICDSC.2008.4635733
- Trentin E, Gori M (2003) Robust combination of neural networks and hidden Markov models for speech recognition. *IEEE Trans Neural Netw* 14(6):1519–1531. doi: 10.1109/TNN.2003.820838
- Zhou L, Li W, Ogunbona P (2016) Learning a pose lexicon for semantic action recognition. In: 2016 IEEE International Conference on Multimedia and Expo (ICME), Seattle, WA, USA, pp 1–6. doi: 10.1109/ICME.2016.7552882
- Zivkovic Z (2004) Improved adaptive Gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, Cambridge, UK, 28-31 Vol.2. doi: 10.1109/ICPR.2004.1333992

## AUTHORS

**Prof. Dr. rer. nat. Ralf Vandenhousten**  
**Dr. rer. nat. Egbert Gedat**  
**Pascal Fechner, Dipl.-Inform.**  
**Richard Fiebelkorn, Dipl.-Ing. (FH)**  
Fachbereich Ingenieur- und Naturwissenschaften  
Technische Hochschule Wildau

**Jan Vandenhousten**  
ixellence GmbH Wildau

E-Mail for correspondence:  
egbert.gedat@th-wildau.de

