Detecting Process Transitions from Wearable Sensors: An Unsupervised Labeling Approach

Sebastian Böttcher
Medical Center
University of Freiburg
Department of Neurosurgery
Freiburg, Germany
sebastian.boettcher@uniklinik-freiburg.de

Philipp M. Scholl
Embedded Systems
University of Freiburg
Computer Science Institute
Freiburg, Germany
pscholl@ese.uni-freiburg.de

Kristof Van Laerhoven
Ubiquitous Computing
University of Siegen
Electrical Engineering and Computer Science
Siegen, Germany
kvl@eti.uni-siegen.de

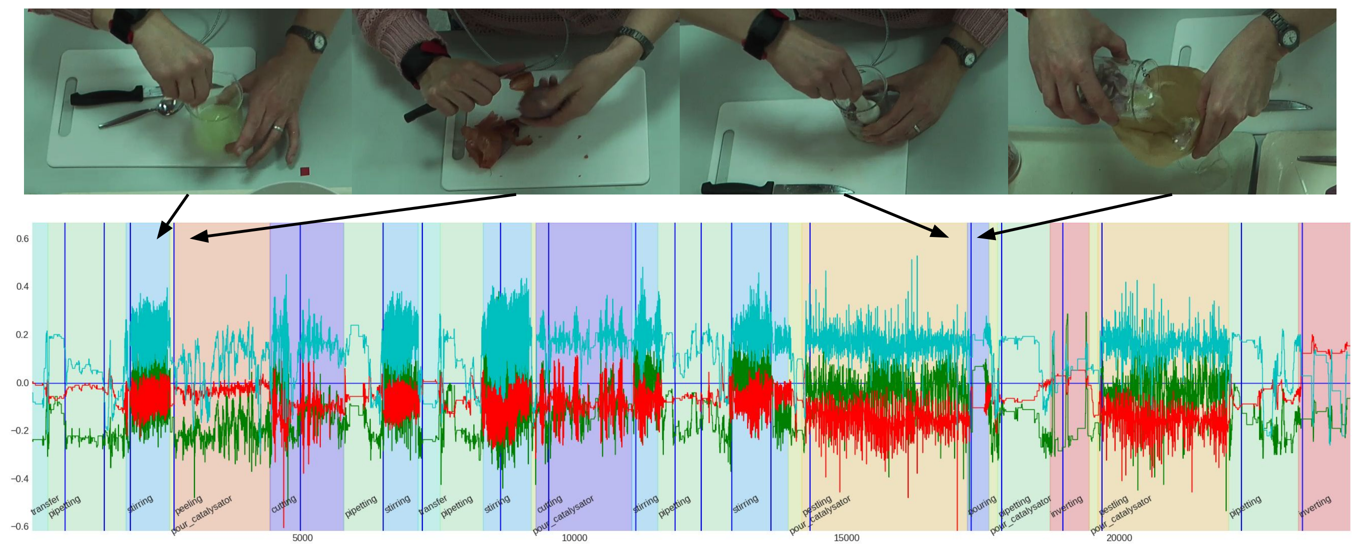

Figure 1. An example of the clustering result for one DNA extraction experiment. Shown are the acceleration time series (data, red, blue, green), ground-truth labels (background colors), and transitions found from clustering (blue vertical bars). Additionally, some video stills at the top show the process at different moments in time (arrows; l.t.r.: stirring, peeling, pestling, pouring). The cut marks are extracted from wrist acceleration measurements.

ABSTRACT

Authoring protocols for manual tasks such as following recipes, manufacturing processes, or laboratory experiments requires a significant effort. This paper presents a system that estimates individual procedure transitions from the user's physical movement and gestures recorded with inertial motion sensors. Combined with egocentric or external video recordings this facilitates efficient review and annotation of video databases. We investigate different clustering algorithms on wearable inertial sensor data recorded on par with video data, to automatically create transition marks between task steps. The goal is to match these marks to the transitions given in a description of the workflow, thus creating navigation cues to browse video repositories of manual work. To evaluate the performance of unsupervised clustering algorithms, the automatically generated marks are compared to human-expert created labels on publicly available datasets. Additionally, we tested the approach on a novel data set in a manufacturing lab environment, describing an existing sequential manufacturing process.

ACM Classification Keywords

H.5.m INFORMATION INTERFACES AND PRESENTATION (e.g., HCI): Miscellaneous

Author Keywords

Activity Recognition; Authoring; Guidance; Manual Workflows; Laboratory Processes

iWOAR '17, September 21–22, 2017, Rostock, Germany

ACM ISBN 978-1-4503-5223-9/17/09...$15.00
https://doi.org/10.1145/3134230.3134233

INTRODUCTION

Identifying steps of manual work either on real-time or off-line recordings has the potential to support such work through different outputs. A possible approach to this problem, followed in this paper, is to detect transitions between steps instead of distinguishing what is actually executed. While this does limit possible applications, since it will not allow to query for particular activities, it could provide marks in concurrently recorded video material to provide a first set of navigation cues for later refinement. Possible areas of application include laboratory experiments, preparing food, manual labor or any kind of repetitive activity that follows a (semi-)fixed procedure. The order and/or number of steps in the process may be known beforehand, such that a classifier must only detect transitions from one state to another. However, instead of detecting the type of step executed at a particular point in time, as usually done, we only try to detect the duration of each step - simplifying the problem. Such an unsupervised approach, even though it might be limiting, removes the necessity for labeled data.

This paper's approach aims at an automatic detection of such transitions based on inertial data recorded on the human body, with simultaneous and synchronized video recording for visual inspection afterwards. Point-of-view video recordings of manual processes have become easier to record, but tend to be hard to browse, both due to their length and because interesting marks in those videos are hard to find. An unsupervised clustering of the body motion provides here a first index into such repositories, e.g. being able to skip sequences where there is little or no movement.

Figure 1 shows an example of the clustering approach applied to a recording of wrist motion and documentation video data of a DNA extraction experiment. Blue vertical bars indicate transitions between clusters in the data, found via KMeans clustering. The graph also shows the groundtruth clustering, in the form of labeled actions during the process shown in different background colors. Video stills from the recording show the process at various points in time. The whole video can thus efficiently be traversed by jumping from one transition mark to another. In an extended consideration of the problem, the recognition of step transitions should be possible regardless of the scenario or data provided, motivating the use of unsupervised methods beyond their obvious advantage of not having to hand-label training data.

The remainder of this paper is structured as follows: First, a short survey of related research is compiled to situate the paper amongst the current state of the art. Following this, we present the system design, which is capable of efficiently solving the problem of separating possible process steps. The system is then applied to three real-world data sets that include video and inertial data, including one new and thus far unpublished one. An evaluation is then performed on these to gather recall performance for different approaches. In the last sections, the experiments' results are discussed in-depth, and a summary of contributions, conclusions, and a short outlook are given.

RELATED WORK

Inertial sensors as used in this work are a commonly used and widely accepted method of sensing human motion for activity recognition. IMUs can provide proper acceleration, angular acceleration and magnetic field strength via integrated accelerometer, gyroscope and magnetometer, respectively. IMUs are easily available as standalone sensors and nowadays built into most consumer wearables like smartphones or smartwatches. In [5, 3] for example a smartphone is used to record IMU data for use in daily life activity recognition, and [22] investigate the feasibility of smartwatches for hand and finger gesture recognition. [11, 2] on the other hand use a custom-made sensor platform for AR, which additionally gives the opportunity to use multiple sensor platforms in a small network to gather data from various body positions at the same time like in [21]. Data collected from IMUs furthermore lends itself to the extraction of a multitude of different features, most prominently in time and frequency domain [20]. Raw data has however also been used in research in some cases [21], especially if realtime applications are investigated. With respect to classification of motion gestures with inertial data, both supervised and unsupervised (e.g. [9, 14]) methods have been used, with unsupervised methods gaining popularity in recent years.

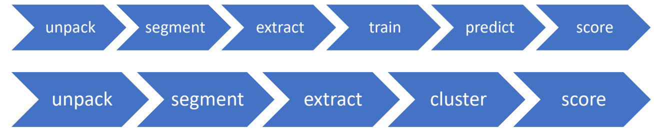

Figure 2. The *Grtool* pipeline. Each step represents an independent, parallelizable module. Compare the supervised (top) and unsupervised (bottom) pipelines, where the train and predict steps in the former are replaced by the clustering step in unsupervised AR

Detecting process steps to guide users through a workflow, or for providing post-hoc recordings was attempted several times already. RFID tags for example have been successfully used to track the movement of manufacturing parts throughout a process workflow [1, 19]. Another popular approach is to use cameras and motion tracking algorithms [18, 17], for example in factory or surveillance applications to track activities. Recent work is even using projectors to augment a users workflow in a process [7] to guide its users. A different environment is the kitchen where meal preparation could be of interest to health monitoring [23]. However, most aligned with the idea of this paper is the GlaciAR system [10], an authoring system for point-of-view videos based on object detection. We however employ wrist motion to detect significant steps. Another example is the wet lab support system proposed in [16], which is able to automatically document an experiment in a laboratory with a known protocol and wrist motion.

SYSTEM DESIGN

Since unsupervised methods often provide a multitude of tunable parameters that can greatly influence the outcome of the clustering and respectively the quality of automated indexing of video recordings, we propose a hyper-parameter evaluation approach in which test recordings are used to find a best parameter combination for unsupervised transition detection. In order to easily and efficiently process large numbers of parameter combinations, a lightweight tool geared towards parallel processing is needed [15]. It is implemented as a Unix command line utility which provides several largely independent modules that can be employed by themselves or linked to form an activity recognition pipeline (Figure 2). Furthermore, *grtool* is based on the Matroska video format [12] and uses audio tracks in conjunction with subtitles to store arbitrary, labeled activity data. *Grtool* provides an interface for both supervised (via *GRT* [8]) and unsupervised (via *scikit-learn* [13]) classification methods.

For unsupervised clustering, the experiments presented in the next chapter use the KMeans clustering, agglomerative clustering and gaussian mixture model (GMM) algorithms, all of which are implemented in *scikit-learn*. The iterative KMeans partition clustering algorithm provides a baseline for unsupervised recognition as it is a basic, efficient and widely used algorithm that can provide good results without much parametrization. Agglomerative clustering on the other hand is a bottom-up hierarchical clustering method that uses a certain linkage type to determine similar clusters to merge. Lastly, GMMs can be seen as a generalization of KMeans clustering, where the cluster covariance is an additional variable, i.e. a GMM fits a given number of Gaussian distributions to the data and provides probabilities for each sample to be in a certain cluster.

Classification with real-world data is not perfect, and noise in the classification output is a common occurrence in activity recognition. In most AR applications this does not largely hinder the recognition of actions in general. However in our scenario of unsupervised process step recognition, the clustering output has to be filtered to robustly detect transitions between states. We use a simple hysteresis filter to smooth the output and thus prevent false state transitions when multiple immediately concurrent samples are classified into different clusters. The filter works in that it only changes its output when a certain number of input samples in the future are the same as the one it is currently regarding.

EXPERIMENTAL SETUP AND EVALUATION

To evaluate and compare the performance of the different clustering methods, we carried out an evaluation on three different data sets from independent sources. All three datasets include labeled human motion data, along with some form of video evidence as extra documentation.

1. **DNA Extraction [16]** dataset has 13 recordings of a DNA extraction experiment performed in a biological experiment laboratory setting. Motion data from a single wrist accelerometer at 50 Hz is combined with videos from a fixed camera above the experimentation area. Experiments include 9 process steps, which may occur multiple times in one recording and in a semi-variable order.

2. **CMU's Kitchen - Brownies [4]** dataset contains 9 recordings of participants preparing a simple cookie baking recipe. Motion data from two arm and two leg IMUs was recorded at 62 Hz. Video recordings from multiple angles, including a head-mounted camera, are included as well. In total, the recipes consist of 29 variable actions.

3. **Prototype Thermoforming.** A third dataset was recorded by ourselves and consists of two recordings of a thermoforming process of a microfluidic 'lab-on-a-chip' disk [6]. It combines IMU data at 50 Hz from a smartwatch and *Google Glass*, and video recordings from the *Google Glass*. The dataset's process contains 7 fixed process steps in a known order (see Fig. 3).

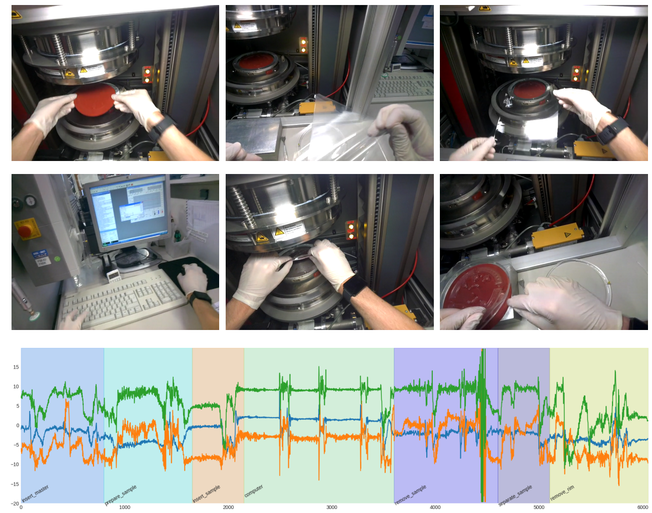

Figure 3. The third dataset in this paper uses video frames from a *Google Glass* recording of steps in a thermoforming process (top), combined with wrist accelerometer data from the same recording (bottom timeseries plot), to extract the different process steps' information (background colors in bottom plot).

Method	win / feat / marg	recall 1 / 2 / 3	F_1-score 1 / 2 / 3
KMeans	80 / mean / 4	.92 / .88 / .95	.92 / .88 / .95
Agglo.	90 / time / 5	.93 / .88 / .95	.93 / .88 / .95
GMM	90 / mean / 3	.88 / .86 / .95	.87 / .86 / .95

Table 1. Top scoring parameter combinations over all data sets (1 / 2 / 3) per method. The results show that these combinations have high scores for all data sets, not just individual experiments.

While the sensor setups across all three data sets are somewhat heterogeneous, each data set provides data from a unique scenario where linear processes are sequentially executed by the subjects. The heterogeneous sensor setups also give a challenging proving ground for the evaluation across all data sets, which is described in more detail later.

The raw data from each data set is preprocessed before it is given to the clustering algorithms. First, the data is stripped of samples that are not or negatively labeled (e.g. labeled `NULL`), which may considerably clean the input of noise, depending on the data set and individual recording. The data is then segmented by applying a sliding window of variable length, without overlap between consecutive windows. For each segment, a number of features are extracted, specifically the mean and variance of the segment, in addition to the min-max range and the median.

After preprocessing and feature extraction the unsupervised clustering algorithms are applied to the feature data: KMeans and agglomerative clustering along with Gaussian Mixture Models (GMM) are regarded for the experiments.For KMeans and agglomerative clustering the results are evaluated per-

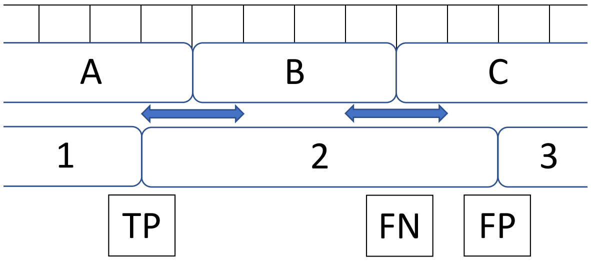

Figure 4. Our approach matches individual activity labels (top: A, B, and C) with clustered segments (bottom: 1, 2, and 3) to score transitions as true positives, false positives or negatives.

participant, since no actual training is necessary, and for the GMM clustering leave-one-participant-out cross-validation is performed.

To score the performance of clustering, the time-series data is clustered and the resulting cluster edges are compared to the labeled ground truth. Cluster edges are regarded as process step transitions and, within a certain *margin*, considered true positives if they coincide with the transition of a ground truth label. If however a ground truth transition is not met by the clustering, a false negative event is registered, and vice versa a false positive event if a cluster edge happens with no corresponding ground truth transition (see Figure 4).

Furthermore, we argue that false positives might in a real-world application not be as harmful as false negatives, since generating additional indices in an archived video file may overall not necessarily be a bad thing. Hence, when scoring the experiments presented here, special attention is given to the recall measure, which provides a performance measure of how many of the ground truth transitions have a corresponding cluster edge.

To find the best parameter combination for unsupervised transition detection, a hyper-parameter approach is applied, in which each data recording is processed multiple times with many parameter combinations. The parameters that are varied per experiment run are the segmentation window length between 10 and 100 samples, the extracted *feature* between mean, variance and an aggregated time feature which combines mean, variance, range and median features, and the transition scoring *margin* mentioned above, between 1 and 5 samples. Each unique combination of these parameters is then applied to each combination of dataset recording, modality and clustering algorithm already described.

Since there is high variation in experiment parameters, each of the three data sets can produce a very large amount of scoring information. Different combinations of recording, sensor modalities, segmentation parameter, extracted feature and scoring parameter can yield hundreds of thousands of separate scores. To bring structure to the results, they are first regarded for each data set individually, followed by an overall analysis of the scores. Summarized results can be found in Table 1 and 2.

Figure 5. Process benchmark of the CMU kitchen experiments (y-axis log-scale). Durations (in sec) of only the clustering process (top), as well as the whole experiment run (bottom), including preprocessing and feature extraction, are shown. Benchmarked algorithms are KMeans (blue), agglomerative clustering (red) and GMM (green).

To get more information on a possible best parameter set for detecting process step transitions, the overall analysis of the scores is done in a three-step approach:

1. The results of each method across the three data sets with a recall score of $\geq$ 0.85 are intersected across the window, feature and margin parameters, since these are the pipeline parameters applicable to all methods and data sets. The intersection removes duplicates.

2. The experiment runs where the three parameters are the same as each parameter combination from the intersection are extracted, per method and data set and again with $\geq$ 0.85, which gives three new tables per parameter combination (for each data set).

3. The results are sorted and aggregated according to the recall scores of the DNA extraction data set, since it provides a large number of individual recordings, simple modality and a relevant set of actions, which makes it the most useful data set for measuring performance.

Table 1 compiles the highest scoring combinations of the resulting table and thus shows the performance of each method on all three data sets combined and not just the individual scenarios, as in Table 2.

Figure 5 shows benchmarking data for the CMU kitchen data set. The processing time of the clustering, or training in case of GMM, was logged for each run of the experiment with different parameters (top). Additionally, the runtime of each parallel job is logged as well, showing the overhead that preprocessing and feature extraction create in the pipeline. Note that both graphs have a logarithmic y-axis scale. The recurring pattern of higher and lower processing times stems from the way the hyper-parameter approach iterates through possible combinations of parameters. Some combinations result in higher processing overhead, e.g. for GMM the higher dimensionality when using all modalities at the same time results in very costly operations. Summarizing the benchmark results also for the other data sets, it can be seen that GMM performs on average much worse than KMeans, and agglomerative clustering only slightly better than KMeans.

DISCUSSION AND CONCLUSION

Looking at individual top scoring experiment runs, there is a parameter combination for each data set and each detection method that can yield good recall scores for a robust detection of process steps. Comparing best runs for the three data sets, there is however no clear winner for the used method. Furthermore, segmentation window lengths of around two seconds are set in all top scoring runs and thus seem to be the best choice for this parameter, as is the mean feature which is sufficient in most cases to yield good results. Table 1 further shows that there are indeed parameter combinations that will yield good results in all of the regarded scenarios.

The results show that even very basic clustering of the mean acceleration of the wrist can already robustly distinguish between single steps in a linear process. This is a rather surprising result, since usually much more elaborate methods need to be employed to provide good recognition results. However, our goal was not to identify particular steps in a protocol but simply detect significant changes which indicate a possible transition to a different step. This is a much simpler problem, hence the surprisingly good results from this rather basic approach. Still, our approach can provide transition marks for a potential automatic labeling system that provides indices for archival video footage or documentation, supporting skipping over uneventful video segments with little changes to wrist motion.

In addition, several other factors need to be considered which may have influenced the results in a positive way. The applied smoothing step after clustering for example removes much uncertainty in the transitions, which may not be the case in comparable work. This factor is even further reinforced by the allowed margin of error applied when scoring the results. Furthermore, the data sets used in the experiments were specifically chosen for this application, i.e. they all provide clear cut, distinct steps in a linear process. Dataset 3 also provides very little variance, in individual process composition as well as in overall recordings, which explains the especially good results in this case. Another factor is the complete disregard of `NULL` samples, i.e. samples where the original labeling provides no classification. These samples were removed from the input before clustering, giving the clustering algorithms a very clean input. Noteworthy is also the fact that all three clustering algorithms used in the unsupervised experiments are in essence very similar, explaining the overall small variance in scores. GMMs can be seen as a generalization of the KMeans method, where the cluster covariance is an additional variable. The implementation of agglomerative clustering used in the experiments applies ward linkage, which minimizes the sum-of-squares error in each cluster, similar to the KMeans algorithm.

Method	modality	win / feat / marg	recall	F_1-score
KMeans	-	80 / mean / 4	0.92	0.92
Agglo.	-	90 / time / 5	0.93	0.93
GMM	-	90 / mean / 3	0.88	0.87
KMeans	l_leg mag	100 / time / 3	0.97	0.97
Agglo.	l_leg gyr	100 / time / 3	0.97	0.97
GMM	r_arm acc	100 / var / 1	0.9	0.9
KMeans	wrist acc	80 / mean / 2	0.96	0.95
Agglo.	head acc	90 / time / 2	0.96	0.95
GMM	wrist mag	100 / mean / 2	0.95	0.95

Table 2. Results for individual top experiment runs, per data set (from top: DNA ext., CMU kitchen, thermof.). Feature extraction *window* as number of samples (@50 Hz); Extracted *feature* is mean, variance or time (time = mean/var/range/median); Transition scoring *margin* as number of samples.

This system is still work in progress, and the next logical step is to assess the usefulness of the generated marks. An experiment where participants are asked to perform a manual process, which is recorded with cameras and body-worn inertial sensors is planned. Participants will later be asked to cut this video into sequences, which represent the steps of the protocol. We will then compare whether this cutting task will be performed quicker if it is pre-cut with an automatic system, or if such a pre-cut has detrimental effects. To facilitate this usability study, we furthermore plan to extend the thermoforming prototyping data set to a more valuable size, and eventually release it for open use in the scientific community. Another point of future optimization are the classification and clustering methods used in the experiments. Further optimization beyond the absolutely necessary model parameters, like number of clusters for the unsupervised clustering methods, is planned for future work. While adding more varied parameters and greater value ranges might yield better performance, each added parameter variation multiplies the number of runs by the size of the parameter range, so a balance has to be found where the improvement in score still warrants higher computational cost.

ACKNOWLEDGEMENTS

We would like to thank Hahn Schickard, and in particular Dr. Felix von Stetten and Harald Kühnle, for their collaboration and assistance in recording the data for the Thermoforming data set. We would also like to thank the Carnegie Mellon University Multimodal Activity Database team for the publication of their research data. The CMU kitchen data set used for this research was obtained from http://kitchen.cs.cmu.edu/ and their data collection was funded in part by the National Science Foundation under Grant No. EEEC-0540865. Support for the data collection and analysis for this paper was funded in part by the collaborative EU research project RADAR-CNS, which receives funding from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 115902.

REFERENCES

1. Sebastian Bader and Mario Aehnelt. 2014. Tracking Assembly Processes and Providing Assistance in Smart Factories.. In *ICAART (1)*. 161–168.

2. Eugen Berlin and Kristof Van Laerhoven. 2012. Detecting Leisure Activities with Dense Motif Discovery. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*. ACM, New York, NY, USA, 250–259.

3. E. Büber and A. M. Guvensan. 2014. Discriminative time-domain features for activity recognition on a mobile phone. In *Proc. IEEE Ninth Int Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) Conf*. Institute of Electrical & Electronics Engineers (IEEE), 1–6.

4. Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. 2008. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. *Robotics Institute* (2008). http://kitchen.cs.cmu.edu/

5. S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook. 2012. Simple and Complex Activity Recognition through Smart Phones. In *Proc. 8th Int Intelligent Environments (IE) Conf*. Institute of Electrical & Electronics Engineers (IEEE), 214–221.

6. Moritz Faller. 2016. Hahn-Schickard: Lab-on-a-chip + analytics. (2016). http://www.hahn-schickard.de/en/services/lab-on-a-chip-analytics/ Accessed: 2016-12-06.

7. Markus Funk, Oliver Korn, and Albrecht Schmidt. 2014. An Augmented Workplace for Enabling User-defined Tangibles. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 1285–1290.

8. Nicholas Gillian and Joseph A. Paradiso. 2014. The Gesture Recognition Toolkit. *J. Mach. Learn. Res.* 15, 1 (Jan. 2014), 3483–3487.

9. Tâm Huynh, Ulf Blanke, and Bernt Schiele. 2007. Scalable Recognition of Daily Activities with Wearable Sensors. In *Location- and Context-Awareness*. Springer, 50–67.

10. Teesid Leelasawassuk, Dima Damen, and Walterio Mayol-Cuevas. 2017. Automated Capture and Delivery of Assistive Task Guidance with an Eyewear Computer: The GlaciAR System. In *Proceedings of the 8th Augmented Human International Conference (AH '17)*. ACM, New York, NY, USA, Article 16, 9 pages.

11. Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. 2006. A Practical Approach to Recognizing Physical Activities. In *Lecture Notes in Computer Science*. Springer, 1–16.

12. Steve Lhomme. 2005. What is Matroska? — Matroska. (2005). https://www.matroska.org/technical/whatis/index.html Acessed: 2016-11-09.

13. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (Nov. 2011), 2825–2830.

14. H. K. Peng, P. Wu, J. Zhu, and J. Y. Zhang. 2011. Helix: Unsupervised Grammar Induction for Structured Activity Recognition. In *2011 IEEE 11th International Conference on Data Mining*. Institute of Electrical & Electronics Engineers (IEEE), 1194–1199.

15. Philipp M. Scholl. 2017. Grtool. (2017). https://github.com/pscholl/grtool Accessed: 2017-01-05.

16. Philipp M. Scholl, Matthias Wille, and Kristof Van Laerhoven. 2015. Wearables in the Wet Lab: A Laboratory System for Capturing and Guiding Experiments. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 589–599.

17. B. Song, A. T. Kamal, C. Soto, C. Ding, J. A. Farrell, and A. K. Roy-Chowdhury. 2010. Tracking and Activity Recognition Through Consensus in Distributed Camera Networks. 19, 10 (Oct. 2010), 2564–2579.

18. C. Stauffer and W. E. L. Grimson. 2000. Learning patterns of activity using real-time tracking. 22, 8 (Aug. 2000), 747–757.

19. T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster. 2008. Wearable Activity Tracking in Car Manufacturing. 7, 2 (April 2008), 42–50.

20. D. Trabelsi, S. Mohammed, Y. Amirat, and L. Oukhellou. 2012. Activity recognition using body mounted sensors: An unsupervised learning based approach. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*. Institute of Electrical & Electronics Engineers (IEEE), 1–7.

21. D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat. 2013. An Unsupervised Approach for Automatic Activity Recognition Based on Hidden Markov Model Regression. *IEEE Transactions on Automation Science and Engineering* 10, 3 (July 2013), 829–835.

22. Chao Xu, Parth H. Pathak, and Prasant Mohapatra. 2015. Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition Using Smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15)*. ACM, New York, NY, USA, 9–14.

23. K. Yordanova, S. Whitehouse, A. Paiement, M. Mirmehdi, T. Kirste, and I. Craddock. 2017. Whats cooking and why? Behaviour recognition during unscripted cooking tasks for health monitoring. In *Proc. IEEE Int. Conf. Pervasive Computing and Communications Workshops (PerCom Workshops)*. 18–21.