A light-weight real-time applicable hand gesture recognition system for automotive applications

Thomas Kopinski, Stéphane Magand, Alexander Gepperth, Uwe Handmann

A light-weight real-time applicable hand gesture recognition system for automotive applications

Thomas Kopinski[1], Stéphane Magand[1], Alexander Gepperth[2] and Uwe Handmann[1]

***Abstract*— We present a novel approach for improved hand-gesture recognition by a single time-of-flight(ToF) sensor in an automotive environment. As the sensor's lateral resolution is comparatively low, we employ a learning approach comprising multiple processing steps, including PCA-based cropping, the computation of robust point cloud descriptors and training of a Multilayer perceptron (MLP) on a large database of samples. A sophisticated temporal fusion technique boosts the overall robustness of recognition by taking into account data coming from previous classification steps. Overall results are very satisfactory when evaluated on a large benchmark set of ten different hand poses, especially when it comes to generalization on previously unknown persons.**

I. Introduction

Being able to interact with a system naturally is becoming ever more important in many fields of Human-Computer-Interaction (HCI). Free-hand poses and gestures are one means which are attributed in this way, thus developing new interaction techniques is desirable in every-day situations, especially with declining sensor costs. Depending on the environment there are parameters restricting the applicability of these devices. Therefore, any approach has to be tailored to its needs. In the automotive environment, factors as daylight interference, obstacle occlusion, limited accessibility are only some of the problems which require consideration.

We present an approach to recognize hand poses with a single ToF-camera mounted on the center console. The so-called point clouds coming from the camera are transformed into a histogram, which in turn is used as input for the training and classification of the hand pose by a Multilayer-Perceptron (MLP). Our approach is purely data-driven in that all relevant information comes from a large database containing samples belonging to 1 out of 10 different hand poses. The use of a PCA-based hand cropping technique and robust point cloud descriptors, together with a neural network-based multi-class learning approach, make our system invariant to rotation, translation and deformation issues, which moreover works without the need to formulate a possibly complicated hand model. Using a ToF sensor additionally provides robustness against daylight interferences.

Building upon prior results[1], [2], we extend our database to have more variance in the data (by adding more persons), and present a novel temporal fusion technique which boosts recognition rates by taking into account preceding recognitions as well. Moreover, our temporal fusion of data lets us take an initial step towards defining dynamic gestures via static hand poses by taking into consideration several snapshots taken over time.

We will first discuss the related work relevant for our research (Sec. II). We then go on to describe the parametrization of the sensors and the setup of our system within an automobile environment in Sec. III. Subsequently we describe the recorded database in Sec. IV and afterwards give an outline of the PCA algorithm used for the cropping of the point clouds in the database as well as in real-time (Sec. V). In Sec. VI we give an account of the used different holistic point cloud descriptors and explain the meaning of the parameter variations we will test. Sec. VII describes the temporal fusion technique as well as the parametrization of the MLPs. The key questions we will investigate in Sec. VIII concern the generalization error of the MLP and the performance of our system in a live demonstration as well as offline. Lastly in X we give an outlook of potential improvements of our system as well as our next steps.

II. Related work

With the ability to easily separate hand and forearm from the background as well as robust applicability under daylight conditions depth sensors in general and ToF-sensors in particular provide an efficient means to deal with tasks such as hand gesture recognition [3]. When trying to differentiate between static hand postures relying solely on depth information seems to be a viable approach as can be seen in [4]. In most cases best results have been achieved when combining RGB data and depth data simultaneously however it remains difficult to achieve satisfactory results in real time without exploiting RGB data, as typically either the range of application is limited or the performance results are dissatisfying. For example in [5] an existing system is improved by adding a ToF-sensor but this approach strongly relies on RGB information to disambiguate pixels and is thus not feasible under difficult lighting conditions. Usually a good performance result was achieved with a very limited pose set or if designed for a specific application [6]. In contrast we aim at a system being able to distinguish difficult hand poses - sometimes varying only by a few data points - in order to be able to realize an interesting in-car application. The biggest disadvantage of ToF-Sensors is a low resolution paired with strong noise which of course makes it difficult to extract robust yet informative features. Improved results

*This work was not supported by any organization

[1]Thomas Kopinski, Stéphane Magand and Uwe Handmann are with the Department of Informatics, University Ruhr West firstname.lastname@hs-rw.de

[2]Alexander Gepperth is with the ENSTA ParisTech b.d.researcher@ieee.org

can be achieved when fusing Stereo Cameras with Depth Sensors, e.g. in [7]. The authors of [8] utilize a single ToF-Sensor in order to detect hand postures with the Viewpoint Feature Histogram which is related to our approach as it relies on the extraction of normal information from point clouds. However we aim at a faster method and thus accordingly improve our approach by simplifying the features as needed and additionally have a more complex gesture set. Moreover we aim at a real-time applicable system which is essential for our application.

The Kinect has become popular in such application scenarios as it extracts RGB and depth data simultaneously (e.g. cf. [9]). However this approach relies heavily on finding hand pixels in order to be able to segment the hand correctly. Employing a sensor as the Kinect is not feasible in our scenario as we aim for small sensor dimensions in order to demonstrate a working system but first and foremost have to avoid the system failing when it is exposed to direct sunlight. Additionally we provide a data-driven approach as we learn hand postures from a large dataset and thus avoid having to define a complex model as in [10]. Moreover the system mentioned again strongly depends finding skin-coloured pixels as well, to allow for segmentation in 2D and 3D as well as hand-tracking. To our knowledge there is no comparable work which is placed in the automotive environment. An extensive overview of the methods and applications used for hand gesture recognition is provided in [11]. One of their insights is that most applications are in the field of robot control, interactive displays/tabletops/whiteboards or sign language recognition. In [12] a case study is made of how the Kinect sensor can be utilised to control E-Mail functions in a car through set of six hand gestures. While the results remain unclear, except for the fact that gestures could be well accepted as a means of control in a car, the gesture set remains small and the effect of different lighting conditions on the results is not discussed. We aim at a specific scenario with a more complex application in the infotainment are thus our defined gesture set allows a broader application range. More comprehensive overviews are given in [13] and [14].

Besides these technological issues, considerable research is conducted on how to design intuitive user interfaces, potentially based on hand postures and gestures. In [15], the authors investigate and compare different menu techniques, whereas [16] presents a system using several projectors and depth cameras named ”LightSpace”. The user is surrounded by surfaces which are filled with content by the projectors. He can interact with the walls, tables, etc. by gestures as these are recognized based by several cameras from the Kinect. Special use cases and fields of applications are considered by several authors. In medical environments, touch gestures are not applicable for reasons of hygiene. Alternative touch-free approaches are explored in [17] and [18].

More recently the Leap motion controller emerged as a new means to obtain 3D data for hand gesture recognition. The authors of [19] compare the performance of the controller with the performance of the Kinect on a gesture set of ten different hand gestures. Their key insights are the fact that the Kinect sensor provides a complete depth map as compared to the Leap motion controller which in turn provides only a limited set of features such as the palm position of the hand. The main difference is a much smaller database compared to ours, the use of SVMs which increases training time and the lighting environment. The whole setup allows for a controllable scenario in terms of noise and light fluctuations and is therefore not practicable for our intentions.

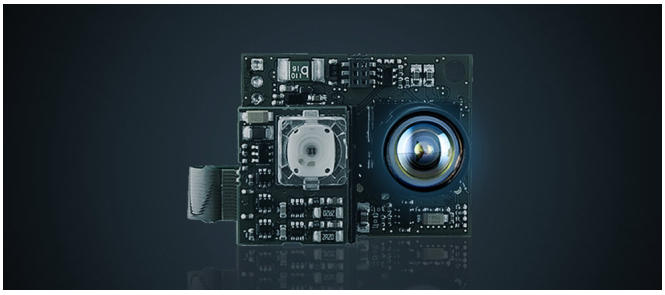

Fig. 1. The camboard nano

Fig. 2. Driver interacting with the system.

In-car scenarios have been in the focus of development for several years, as the driver can keep his hands close to the steering wheel while being able to focus on the surrounding environment. Pointing capabilities could be interesting to control content in the head-up displays. A good overview of automotive HMI is given in [20]. Hassani describes advantages in user acceptance of in-air hand gestures in comparison to touch gestures seniors of Human-Robot Interaction [21].

Scenarios as these call for robust data extraction techniques which becomes feasible when making use of a ToF-sensor. We demonstrate that it is able to achieve satisfactory results for the disambiguation task of ten hand poses setting up the recognition pipeline as described in the following chapters.

III. System Setup

We integrate a single ToF Camera (the Camboard nano - see Fig.1) into a car, fix it to the center console and connect it to a Laptop with a Linux system installed, handling the

computation task. No calibration is necessary other than the exposure time set to 0.8ms and optimized for close-range interaction. The camera itself has comparably small dimensions (37x30x25mm) and measures distances by the time-of-flight principle which makes it useful in any outdoor scenario. The benefit of our approach is the fact that little to no calibration is necessary as the robustness of our descriptor and the neural networks compensate for minor changes in camera setup. The camera is recording the designated inner part of the car interior in which the driver is able to interact with the system (see Fig.2). We focus on recognising the subject's (i.e. driver) right hand for the desired hand poses. Therefore we defined the designated Volume of Interest (VOI) within which we want to identify user input. Due to driver behaviour, possible range and convenience as well as obstacle occlusion (e.g. steering wheel) our VOI is of trapezoidal shape with a depth of 27-35cm, a maximum width of 60cm and a minimum width of about 45cm enclosed by the Field of View(FoV) of the camera frustum. The total height covered by our camera ranges from 30-35cm. Furthermore we cover a space big enough to recognize the most important movements in the car. Usually the driver has his hand on the steering wheel or close to it or in other situations leans onto the armrest, which differs significantly in position and allows for longer interaction with our system. By defining our VOI as described, we are able to cover all of these possibilities.

IV. Hand Gesture Database

We record data from 16 persons, each displaying 10 different hand poses (cf. Figure 3). For each gesture, 3000 samples are recorded, summing up to 30000 samples per person and a total database of 480000 samples. In order to induce some variance into the data, during the recording phase each participant is asked to rotate and translate their hand in all possible directions. Moreover, to tackle the task of scaling, for each gesture we define 3 different distance ranges, in which the participant is asked to perform the hand gesture in order to ensure sufficient sample coverage for various distances. Each frame is recorded at a resolution of 165x120px at 90fps with the Camboard nano, making it robust to daylight interferences and thus applicable in any outdoor scenario. This results in an alphabet of ten hand poses: Counting from 1-5 and *fist*, *stop*, *grip*, *L*, *point* denoted by *a-j* (cf. Figure 3). For the chosen probands, both male and female, the size of the hand ranges from 8,5cm - 9,5cm in width and from 17,0cm - 19,5cm in length.

V. Hand-forearm segmentation with PCA

A. Finding the principal axis of a point cloud

The main directions of the cloud are found using Principal Component Analysis (PCA) [22]. PCA aims to find uncorrelated basis vectors for an arbitrary set of data vectors. Eigenvectors (also termed "principal components") are ordered by the variance of data points projected onto them, allowing efficient data compression by omitting principal components of low variance. This algorithm is applied as shown below,

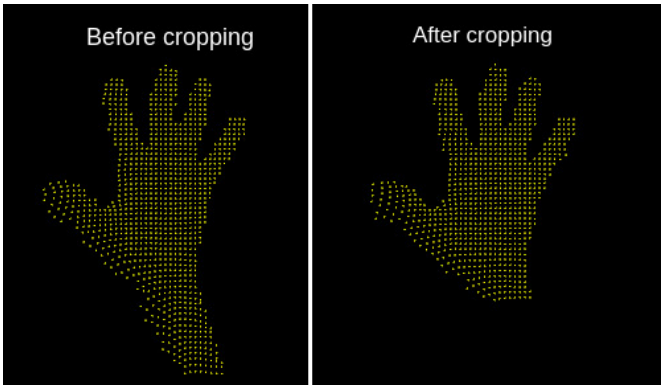

Fig. 4. Point cloud before PCA-cropping (left) and after (right).

using as input the set of n 3D coordinates of points in a point cloud denoted x_j, $j \in [0, n]$).

- The mean value $\bar{x} = \frac{1}{n} \cdot \sum_{j=1}^{n}(x_j)$ is computed.
- The scatter matrix is calculated :

$$S = \sum_{j=1}^{n}(x_j - \bar{x})(x_j - \bar{x})^\top$$

  This matrix can be used as maximum-likelihood estimate of the covariance matrix.
- The Eigenvectors of this matrix yield the principal components.

We intend to cut off 'unnecessary' parts of the cloud, i.e. outliers and elongated parts of the forearm. In this case, the principal components correspond to orthogonal vectors that represent the most important directions in the point cloud. The vector with the most important y-component allows to recognize the axis hand-forearm.

The wrist, as the link between the hand and the forearm, is detected in order to determine a limit for the cropping. The employed method assumes that the distance between the endpoint of the fingers and the centroid is an upper bound of the distance between the centroid and the wrist.

To find the endpoint of the hand towards the direction of the fingers, tests are made along the axis, starting at the centroid and moving progressively upward. At each step, we determine whether there are points within a designated small neighborhood around the axis. The upper end of the hand is marked if this number of neighboring points equals 0. Then the bottom limit for the wrist is fixed at the same distance from the centroid, but in the inversed direction along the y-axis. All points below this wrist limit are cut out which is exemplarily shown in Fig.4.

VI. The PFH-Descriptor

The PFH-Descriptor (PFH-Histogram) [23] is a local descriptor which relies on the calculation of normals. It is able to capture the geometry of a requested point for a defined k-neighbourhood. Thus, for a query point and another point within its neighbourhood, four values (the point features or PFs) are being calculated, three of which are angle values and the fourth being the euclidean distance between these

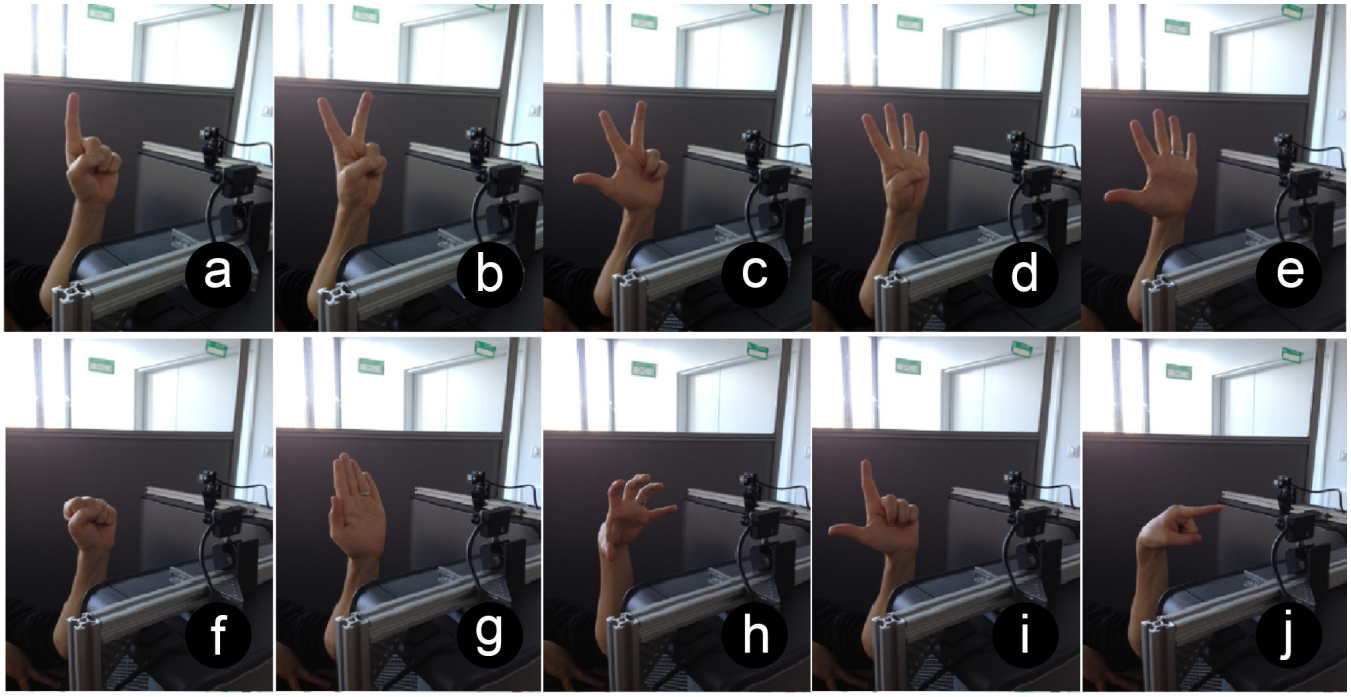

Fig. 3. The hand gesture database consisting of 10 different static hand poses.

two points. The angle components are influenced by each point's normal, so in order to be able to calculate them, all the normals have to be calculated for all points in the cloud. Therefore we are able to capture geometric properties of a point cloud in a sufficient manner, depending on the chosen parameters. These parameters have been thoroughly examined in our previous work which led for example to an optimal choice for the parameter n, the radius for calculation of the sphere which encloses all points used to calculate the normal of a query point. One major drawback is the fact that the PFH-descriptor cannot be easily embedded into a real-time applicable system as the computation cost becomes too high, when we extend it to be a global descriptor. To overcome this issue, we present a modification of the PFH-Descriptor in the following section.

A. Modification of the PFH-Descriptor

Our version of the PFH-Descriptor makes use of its descriptive power while maintaining the real-time applicability. Using the PFH in a global sense would mean having to enlarge the radius so that every two point pairs in the cloud are used to create the descriptor. This quickly results in a quadratically scaling computation problem as a single PFH-calculus would have to be performed 10000 times for a point cloud of 100 points. Given the fact that our point clouds have a minimum size of 200 points up to 2000 points and more, this is not feasible for our purposes. Therefore we randomly choose 10000 point pairs and use the quantized PFs to build a global 625-dimensional histogram. We calculate one descriptor per point cloud which forms the input for the neural network.

VII. Training setup and MLP parametrization

The novelty of this contribution is mainly embedded in the core of the recognition pipeline, namely the fusion of features and neuron outputs. This leads on the one hand to a real-time applicable system and on the other hand to an efficient boosting to the overall generelization performance possibly extendable over many time steps. We show that by employing fusion in this way, satisfactory results for a complex task can be achieved effortlessly. The advantage of MLPs over other methods such as SVMs lies in the quickly parameterizable networks associated with a short training time which is highly relevant as depending on the fusion strategy or the application the optimal strategy requires extensive research and testing. We have conducted thorough testing on the database at hand and achieved similar scores for selected cases with MLPs and SVMs in term of precision and recall.

A. Temporal integration of information

In order to improve the overall recognition rate of our MLP-based approach, we present a temporal fusion technique. To this end the training and testing data have to be prepared as follows: Firstly, we split training and testing data for each run in such a way that data coming from one person is not included in the training set, thus being able to measure the generalization performance of our system on previously unknown data. Secondly the training data has to be presented in a chronological way in order to make the fusion technique viable. The overall procedure is split into a 2-step approach. In an initial step an MLP is trained on all the data from the training set. To induce temporal information into the second MLP, the training step has to be modified in such a way that, at a given point in time t, not only the input from the

feature vector is presented to the MLP, but also the values of the output neurons of the first MLP classifying the sample at time $t-1$. To achieve this, the training data for MLP 2 have to be presented in a chronological order. Therefore the size of the input layer of the second MLP is determined by the length of the feature vector + number of the output neurons of MLP 1 and for our case sums up to (625 + 10). This approach can be motivated as follows: During the interaction of the user with the system, multiple snapshots are taken from the camera for a single hand pose. Thus information considered 'over time' i.e. for classifications coming time points shortly before the current point in time can be used to stabilize the results.

B. Neural network topology and training parameters

For the described two-step fusion approach the networks are trained with standard parameters with variations on the network topology. The input for the first network is formed by the modified point feature histogram(MPFH) of a processed point cloud as described in Sec. VII. The input for the second network is formed by first classifying the previous sample with the first network, calculating the output neurons' activities, and then concatenating these activities with the MPFH of the current time step. Thus, the input layer of the second network is of size $n+10$, since both networks have as many output neurons as there are classes in the classification problem, i.e., 10. The network topologies are therefore $625-h-10$ and $635-h-10$, respectively, with variations possible in the hidden layer size h. We conduct several experiments to obtain good values for h, which we find in the range of 30-50 hidden neurons.

The network is implemented using the FANN library [24]. The training algorithm is the standard RPROP algorithm and the activation function is the sigmoid function for both hidden and output layers.

C. Real-time applicability

Our current system is able to perform feature extraction and classification in real-time. More specifically, the segmentation and cropping of the hand, the calculation of features and the classification task are realisable at a frame rate of 30-50Hz, depending on the size of the point cloud.

VIII. Experiments and Results

We perform tests on the data set comprising all 16 persons. In order to compare our approach for the regular MLP and the fusion technique, two classifications are conducted for any input at a given point in time. Moreover, we vary the number of neurons in the hidden layer to measure whether we are able to improve the performance of our algorithms. Tab. VIII sums up the most important results. Each row shows classification accuracy for an MLP trained on all the data except the person it is tested on, each test being a generalization performance test. As an example, column 1 then represents the performance of all four MLPs, trained on persons 2-16 and tested on person 1. The overall classification accuracy of an MLP - averaged over all persons

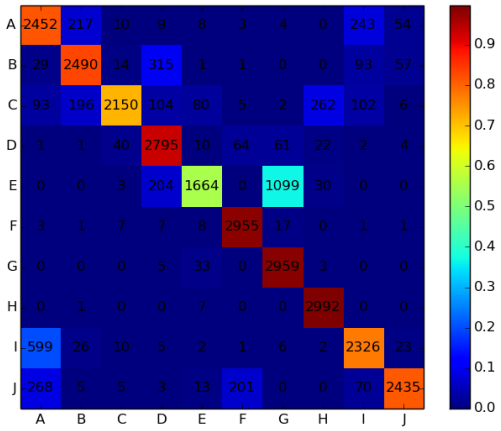

Fig. 5. Confusion matrix for the standard MLP.

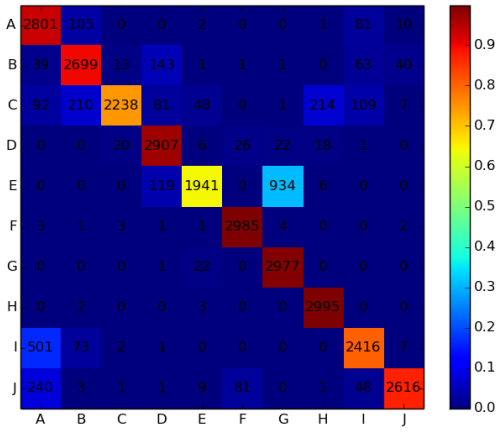

Fig. 6. Confusion matrix for the fused MLP.

- depending on the number of neurons in the hidden layer and the kind of fusion technique is shown in the last column.

First of all, it is observable that our fusion approach, denoted MLP2, outperforms the regular approach (MLP1) as the average MSE for the fused input is about 3% lower averaged over all persons. For some individual cases we are able to improve the results by up to 8% - here averaged over all gestures of a single person - as for instance for person 13 (improved from 90% to 98%) in the case of the larger MLP. Increasing the number of neurons from 30 to 50 has a negligible effect as the improvement of our approach compared to the standard MLP stays around 3% for all variations of the sizes of the hidden layer. However, it is possible to lower the average overall MSE by around 1.5% from 14.75% down to 13.06%, when comparing the upper two rows of Tab. VIII with the lower rows.

For those cases which perform worst in terms of recognition rate, we are still able to improve the results by an

participant	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Acc.
MLP1-30	81%	50%	69%	55%	76%	59%	56%	68%	79%	68%	88%	72%	95%	72%	87%	79%	72.1%
MLP2-30	83%	51%	72%	60%	79%	62%	58%	72%	85%	74%	89%	73%	97%	75%	91%	83%	75.3%
MLP1-50	83%	49%	69%	58%	79%	62%	57%	68%	84%	70%	89%	75%	90%	74%	90%	80%	74.0%
MLP2-50	86%	52%	74%	63%	83%	65%	60%	72%	89%	74%	90%	75%	98%	75%	93%	82%	77.0%

TABLE I

GENERALIZATION RESULTS FOR ALL 16 PERSONS AND BOTH MLPS, EACH WITH 30 AND 50 NEURONS RESPECTIVELY

average of 2-3% by the fusion technique. It is only for three test persons that our fusion approach has little to no effect (persons 11, 12 and 14) as the change of MSE remains around 1% or below. On the other hand, in no case the fusion approach leads to a deterioration of performance. A more precise evaluation of the results shows that the temporal fusion approach reduces the disambiguation problem to two candidates.

More specifically, the confusion matrices (cf. Fig.5 and Fig.6) show the correct classifications as well as the false positives, for all ten gestures, here presented exemplarily for person 13. Each row represents the (mis-) classifications for each gesture shown on the left-hand border. So gesture A ('one') has been classified 2451 times (Fig.5) correctly with the standard approach and 2801 times correctly with the improved fusion technique (Fig.5). At the same time, the same gesture has been mistaken 217 times for gesture 'two' and this number has been lowered to 105 with the improved approach. Simultaneously the number of false positives for this gesture being mistaken for gesture I dropped from 243 to 81.

Moreover, both approaches work best for gestures D, F, G, and H nearing 100% recognition rate, followed by gesture A, B and J (around 90%) and C and I (around 75-80%) referring to the standard MLP in Fig.5. For this person, both approaches perform worst for gesture E ('five') being classified correctly with a precision of around 55% and 65% respectively. More interestingly, it is observable that this gesture is being mistaken for gesture G ('flat hand') which can be explained by the similarity of appearance of both kinds of gestures.

Furthermore, the gesture J ('point') was mistaken around 201 times (cf. Fig.5) for gesture F ('fist') before our fusion approach and 81 times (Fig.6) after fusion. This is a drop of more than 60% and it reduces the problem to disambiguating only gestures J and F from each other, which we are able to handle individually and makes the recognition task much easier. Similar observations can be made for gesture pairs B $\rightarrow$ *D* and E $\rightarrow$ D for the example depicted as well as across the group of participants as a whole.

IX. IN-CAR INFOTAINMENT APPLICATION

The framework was connected to a tablet running a simulated infotainment system as can be seen in Fig.7. The camera is mounted on the front console and connected to a standard laptop with Ubuntu running the recognition pipeline. The laptop in turn is linked via a wireless interface with the tablet simulating an infotainment system with switchable audiochannels, a USB and CD interface as well as simulated incoming phone calls. The user can user the standard gesture alphabet to interact with each of the functions being able to control the system with her/his right hand during a car drive.

Fig. 7. Complete system: sensor, laptop and tablet mounted to the front console visualizing the infotainment system

X. OUTLOOK

In this contribution, we realised a real-time hand gesture recognition system based on inexpensive and robust time-of-flight cameras, intended for human-machine interaction in an automotive environment. The whole system was developed and implemented into an in-car environment allowing the driver to use a gesture alphabet of ten static gestures to interact with an infotainment system. The recognition pipeline was realised by setting up a large database, training two different MLPs fusing information coming from features and neuron outputs. This method of fusing MLP output coming from preceding time steps with feature coming from current point clouds allows for a significant generelisation improvement which we verified on our database. Tests show very satisfying generalisation performance for a set of 10 static gestures. Our system works with an average frequency of 30-50Hz being limited mainly by the ToF-camera itself, depending primarily on the number of points in the captured point cloud. The implemented PCA algorithm crops superfluous parts of

the forearm which further stabilizes the results as we assumed this was a major problem of the disambiguation problem so far. Our approach fuses information coming from classifications made before the current point in time which, as we are able to prove, provides even more stability to our system.

The next steps consist of adding a disambiguation module for the most difficult cases as well as using our fusion technique to extend our recognition from static hand poses to dynamic hand gestures. Since we are able to reduce the disambiguation problem with our current approach to two cases in most situations this will result in a more stable recognition module. Furthermore we can add even more information over an extended time span as well as adding confidence measures to our current decision module. We intend to fuse our algorithms to create an improved gesture recognition system by allowing interaction via static hand poses and dynamic hand gestures. As we allow dynamic interaction further features in the infotainment system such zooming in/out in map applications will become viable.

REFERENCES

[1] Thomas Kopinski, Alexander Gepperth, Stefan Geisler, and Uwe Handmann. Neural network based data fusion for hand pose recognition with multiple tof sensors. *ICANN*, 2014.

[2] Thomas Kopinski, Stefan Geisler, Louis-Charles Caron, Alexander Gepperth, and Uwe Handmann. A real-time applicable 3d gesture recognition system for automobile hmi. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2616–2622. IEEE, 2014.

[3] S. Oprisescu, C. Rasche, and B. Su. Automatic static hand gesture recognition using tof cameras. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 2748–2751. IEEE, 2012.

[4] E. Kollorz, J. Penne, J. Hornegger, and A. Barke. Gesture recognition with a time-of-flight camera. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):334–343, 2008.

[5] Michael Van den Bergh and Luc Van Gool. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 66–72. IEEE, 2011.

[6] S. Soutschek, J. Penne, Jo. Hornegger, and J. Kornhuber. 3-d gesture-based scene navigation in medical imaging applications using time-of-flight cameras. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008.

[7] Y. Wen, C. Hu, G. Yu, and C. Wang. A robust method of detecting hand gestures using depth sensors. In *Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on*, pages 72–77. IEEE, 2012.

[8] T. Kapuściński, M. Oszust, and M. Wysocki. Hand gesture recognition using time-of-flight camera and viewpoint feature histogram. In *Intelligent Systems in Technical and Medical Diagnostics*, pages 403–414. Springer, 2014.

[9] Matthew Tang. Recognizing hand gestures with Microsoft's kinect. *Web Site: http://www.stanford.edu/ class/ee368/Project_11/ Reports/Tang_Hand_Gesture_Recognition. pdf*, 2011.

[10] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, pages 1–11, 2011.

[11] Jesus Suarez and Robin R Murphy. Hand gesture recognition with depth images: A review. In *RO-MAN, 2012 IEEE*, pages 411–417. IEEE, 2012.

[12] Andreas Riener, Michael Rossbory, and Alois Ferscha. Natural dvi based on intuitive hand gestures. In *Workshop UX in Cars, Interact*, page 5, 2011.

[13] Zhou Ren, Jingjing Meng, and Junsong Yuan. Depth camera based hand gesture recognition and its applications in human-computer-interaction. In *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pages 1–5. IEEE, 2011.

[14] Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Yael Edan. Vision-based hand-gesture applications. *Communications of the ACM*, 54(2):60–71, 2011.

[15] Gilles Bailly, Robert Walter, Jörg Müller, Tongyan Ning, and Eric Lecolinet. Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus. In *Human-Computer Interaction–INTERACT 2011*, pages 248–262. Springer, 2011.

[16] Andrew D Wilson and Hrvoje Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 273–282. ACM, 2010.

[17] Rose Johnson, Kenton O'Hara, Abigail Sellen, Claire Cousins, and Antonio Criminisi. Exploring the potential for touchless interaction in image-guided interventional radiology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3323–3332. ACM, 2011.

[18] Guilherme Cesar Soares Ruppert, Leonardo Oliveira Reis, Paulo Henrique Junqueira Amorim, Thiago Franco de Moraes, and Jorge Vicente Lopes da Silva. Touchless gesture user interface for interactive image visualization in urological surgery. *World journal of urology*, 30(5):687–691, 2012.

[19] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. Hand gesture recognition with leap motion and kinect devices. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 1565–1569. IEEE, 2014.

[20] Carl A Pickering, Keith J Burnham, and Michael J Richardson. A research study of hand gesture recognition technologies and applications for human vehicle interaction. In *3rd Conf. on Automotive Electronics*. Citeseer, 2007.

[21] Anouar Znagui Hassani, Betsy van Dijk, Geke Ludden, and Henk Eertink. Touch versus in-air hand gestures: evaluating the acceptance by seniors of human-robot interaction. *Ambient Intelligence*, pages 309–313, 2011.

[22] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[23] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3384–3391. IEEE, 2008.

[24] Steffen Nissen. Implementation of a fast artificial neural network library (fann). *Report, Department of Computer Science University of Copenhagen (DIKU)*, 31, 2003.