Time-of-Flight based multi-sensor fusion strategies for hand gesture recognition

Thomas Kopinski, Darius Malysiak, Alexander Gepperth, Uwe Handmann

To cite this version:

Thomas Kopinski, Darius Malysiak, Alexander Gepperth, Uwe Handmann. Time-of-Flight based multi-sensor fusion strategies for hand gesture recognition. IEEE International Symposium on Computational Intelligence and Informatics, Nov 2014, Budapest, Hungary. hal-01098695

HAL Id: hal-01098695
https://hal.inria.fr/hal-01098695

Submitted on 28 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Time-of-Flight based multi-sensor fusion strategies for hand gesture recognition

Thomas Kopinski
Hochschule Ruhr West
Computer Science Institute
Bottrop, Germany
thomas.kopinski@hs-rw.de

Darius Malysiak
Hochschule Ruhr West
Computer Science Institute
Bottrop, Germany
darius.malysiak@hs-rw.de

Alexander Gepperth
ENSTA ParisTech
828 Blvd des Maréchaux
91762 Palaiseau, France
alexander.gepperth@ensta-paristech.fr

Uwe Handmann
Hochschule Ruhr West
Computer Science Institute
Bottrop, Germany
uwe.handmann@hs-rw.de

***Abstract*—Building upon prior results, we present an alternative approach to efficiently classifying a complex set of 3D hand poses obtained from modern Time-Of-Flight-Sensors (TOF). We demonstrate it is possible to achieve satisfactory results in spite of low resolution and high noise (inflicted by the sensors) and a demanding outdoor environment. We set up a large database of pointclouds in order to train multilayer perceptrons as well as support vector machines to classify the various hand poses. Our goal is to fuse data from multiple TOF sensors, which observe the poses from multiple angles. The presented contribution illustrates that real-time capability can be maintained with such a setup as the used 3D descriptors, the fusion strategy as well as the online confidence measures are computationally efficient.**

***Index Terms*—gesture recognition, support vector machines, neural networks, efficient classification, tof sensors**

I. Introduction

Hand gestures or hand poses are gradually introduced into everyday life, be it by e.g. Kinect-type sensors in our homes or public places to interact with various kinds of systems. There is a multitude of approaches exploiting the advantages of the various kinds of sensors coming into use but since daylight applicable Time-of-flight(ToF) sensors are becoming less expensive and steadily improve in terms of signal quality, the range of applications naturally increases. Opposed to other approaches, we are able to set up a system working under daylight conditions whilst having more challenging outdoor parameters to deal with. In this article we present a real-time applicable system to classify a set of ten static hand poses (contrasted to hand gestures being dynamic). This is achieved by fusing the data coming from two ToF sensors. Each camera's data is captured and transformed via a chosen pointcloud descriptor which is again used as input for the training of a multilayer perceptron (MLP) and a support vector machine (SVM). We build upon previously acquired understandings[1] namely the fact, that our approach leads to satisfying results when carefully choosing the right parameters. We hereby demonstrate that we are able to significantly boost our results by utilizing a support vector machine and give an outline of how this may impact our upcoming results in the near future. We will first discuss the related work relevant for our research (Sec. II) and then go on to describe the sensors and the used database in Sec. III. Subsequently, in Sec. IV we will give an account of the used different holistic point cloud descriptors and explain the meaning of the parameter variations we will test. The key questions we will investigate in Sec. VI and Sec. VI-B concern the proper **choice of parametrized descriptors**, furthermore the **added value of a second ToF sensor**, and lastly the issue of **efficient neural network based fusion strategies** which are contrasted to the **support vector classification techniques**. In Sec. VII, the obtained results will be discussed in the light of these questions.

II. Related Work

When recognizing hand poses, depth sensors allow for a simple and robust solution, as they can easily deal with tasks as segmentation of the hand/arm from the body by simple thresholding as described in [2]. Several studies have made use of this feature with different approaches to segmentation. Moreover it is possible to make use of the depth information to distinguish between ambiguous hand postures [3]. Nevertheless, it has not been possible to achieve satisfactory results utilizing only a single depth sensor. Either the range of application was limited or the performance results were dissatisfying. Usually a good performance result was achieved with a very limited pose set or if designed for a specific application [4]. ToF-Sensors - although working at stereo-frame rate - generally suffer from a low resolution which of course makes it difficult to extract proper features. Improved results can be achieved when fusing Stereo Cameras with Depth Sensors, e.g. in [5]. In [6] a single ToF-Sensor is used to detect hand postures with the Viewpoint Feature Histogram. Various approaches make use of the Kinect's ability to extract depth data and RGB data simultaneously [7]. However this approach relies heavily on finding hand pixels in order to be able to segment the hand correctly. Moreover, approaches utilising the Kinect sensor will always suffer from changing lighting conditions which in our case is no drawback as ToF-sensors show robust results in such situations. [8] also make use of the Kinect sensor's ability to acquire RGB and depth data simultaneously albeit using a hand model as a basis for hand pose detection. Nevertheless this algorithm also relies on finding skin-colored pixels to allow for segmentation in 2D and 3D as well as tracking the hand.

Beneath the technology development research is conducted on how to design intuitive user interfaces. Bailly et al. investigate and compare different menu techniques in [9]. Wilson and Benko developed a system with several projectors and depth cameras named LightSpace [10]. tables, etc. by gestures as these are recognized based by several cameras from the Kinect. are not applicable for reasons of hygiene, where a sterile user interface is required. In [11] such approaches are validated.

In-car scenarios have been developed for several years as the 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 is given in [12]. Human-Robot Interaction [13]

Such scenarios demand robust data extraction techniques which is provided by the aforementioned ToF-sensor. Our approach shows that it is possible to achieve satisfactory results relying solely on depth data when detecting various hand poses. In merging information from a second depth sensor we are able to boost our results significantly while always retaining the applicability under various lighting conditions - one of the greatest advantages of ToF-sensors compared to e.g. the frequently used Kinect sensor.

To our knowledge there exist no profound studies examining the fusion of data coming from multiple ToF-sensors which is then used to solve the given task of classifying the set of hand poses by neural networks and support vector machines.

III. Database

The data was recorded using two ToF-Sensors (Figure 1 and 2) of type Camboard nano which provides depth images of resolution 165x120px with a frame rate of 90fps. The illumination wavelength is 850nm which makes the cameras applicable in various light conditions whilst maintaining robustness versus daylight interferences. Since the ToF-principle works by measuring the time the emitted light needs to travel from the sensor to an object and back pixel-wise the light is modulated by a frequency of 30MHz in order to be able to distinguish it from interferences. In a multi-sensor setup however this may lead to a distortion of measurements since both sensors have the same modulation frequency. To avoid such measurement errors, the data was recorded by taking alternating snapshots from each sensor. As can be seen in Figure 1 the cameras are mounted in a fixed position at a distance of approx 49.5cm and a perpendicular angle from the recorded object. This allows for a recording of the database such that the hand can be placed in an equal distance of about 35cm from each camera to the centroid of the resulting point cloud data set and therefore each camera can also be calibrated to its needs. For the current experiments, focus has been put on the recognition of static hand gestures which are contrasted to dynamic hand gestures. Each set of poses was recorded with a variation of the hand posture in terms of translation and rotation of the hand and fingers. This results in an alphabet of ten hand poses: *point*, *fist*, *grip*, *L*, *stop* and counting from 1-5 (cf. Figure 2). For each pose, a set of 2000 point clouds was recorded for each camera. Since we recorded hand poses from four different persons independently, this yields a data set of 160.000 samples. Additionally, we rotated one camera by 60° towards the other camera and recorded the same set now from an angle of 30° and compared the results to each other resulting in another data set of 160.000 point clouds. The database is randomly split into two parts of equal size for training and evaluation purposes.

IV. Point cloud descriptors

All used global descriptors were calculated using methods of the publicly available Point Cloud Library (PCL).

A. The ESF-Descriptor

The ESF-Descriptor (Ensemble of Shape Function) [14] is a global descriptor which does not rely on the calculation of the normals. First, 20000 points are sub-sampled from the input point cloud. Then, the algorithm repeatedly samples three points, from which four simple measures are calculated, which are discretized and used for histogram calculation.

B. The VFH-Descriptor

The VFH-Descriptor(Viewpoint Feature Histogram) [15] is a global descriptor partially based on the local FPFH (Fast Point Feature Histogram)[16] descriptor. It uses normal information, taking into consideration the view angle between the origin of the source and each point's normal. It furthermore includes the SPFH (Simplified Point Feature Histogram) for the centroid of the cloud, as well as a histogram of distances of the points in the cloud to the centroid. When calculating the VFHs for the various hand poses we have to take into consideration the influence of the normals on the results. In the described case the search parameter r guides the influence of the surrounding for the calculation of the normal. Choosing a small r can result in low descriptive power while a large r results in high computational load. We empirically chose a value of $r = 5cm$ and denote the resulting descriptor VFH5.

C. Neural network classification and fusion

With M cameras, N descriptors will be produced per frame (here: M=N) according to the methods described above. We use a multilayer perceptron (MLP) network[17] to implement the multi-class decision, which is either based on the the concatenation of all N descriptors ("early fusion"), or on each descriptor individually, with a subsequent combination of results ("late fusion"). The MLP training algorithm is "RProp"[17], with standard hyperparameters $\eta^{+} = 1.2$, $\eta^{-} = 0.6$, $\Delta_0 = 0.1$, $\Delta_{\text{min}} = 10^{-10}$ and $\Delta_{\text{max}} = 5$. Network topology is NK-150-10 (hidden layers are fixed to 1[17], hidden layer sizes from 10-500 were tested), K indicating the method-dependent descriptor size, and N the number of cameras, here $N = 2$. Us usual, activation functions are sigmoid throughout the network. MLP classifiers have 10 output neurons (one per gesture class) with activities o_i. Thus, the final classification decision is obtained by taking the class of the neuron with the highest output. However, we do not

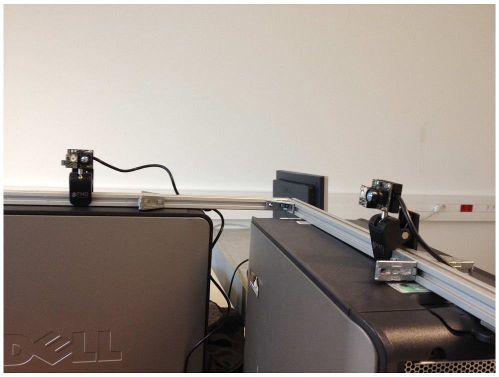

Fig. 1: The current setup for 90°

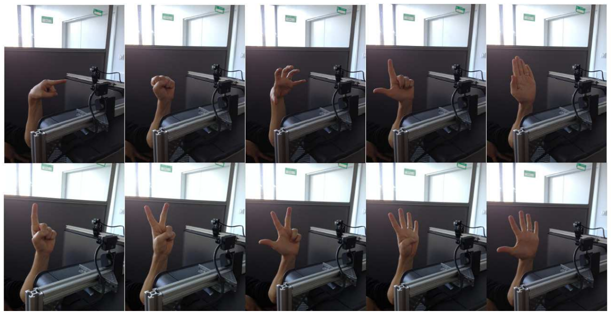

Fig. 2: The hand pose database

necessarily wish for every classification to be taken seriously, and we define several confidence measures $\text{conf}(\{o_i\})$ to this effect. Final decisions are thus taken in the following way:

$$\text{class} = \begin{cases} \text{argmax}_i o_i & \text{if conf}(\{o_i\}) > \theta_{\text{conf}} \\ \text{no decision} & \text{else} \end{cases}$$

We test three ad hoc confidence measures, which perform a mapping from $\mathbb{R}^{10} \rightarrow \mathbb{R}$: "confOfMax", "diffMeasure" and "varianceMeasure". Each of these measures is derived from the idea of approximating an entropy calculation, based on the information-theoretic idea that low entropy means high information content. The precise definitions are as follows:

$$\begin{aligned} \text{confOfMax}(\{o_i\}) &= \max\ o_i \\ \text{diffMeasure}(\{o_i\}) &= \max_i\ o_i - \max_i^2\ o_i \\ \text{varianceMeasure}(\{o_i\}) &= \frac{1}{N}\sum_i (o_i - E(\{o_i\}))^2 \end{aligned} \tag{1}$$

where $\max_i^2\ o_i$ indicates the second-strongest maximum over the neural outputs. For performing late fusion, that is, obtaining two independent classifications o_i^1, o_i^2 based on each camera's features, we simply calculate the arithmetic mean of both output vectors: $o_i^F = 0.5(o_i^1 + o_i^2)$. This intrinsically takes into account the variance in each response, as an output distribution strongly peaked on one class will dominate a flat (or less peaked) distribution. The resulting output distribution o_i^F can then be subjected to the decision rule of Eqn. (1).

V. Support Vector Classification

In its nature a support vector machine (SVM) is a binary classifier, yet there exist several methods to extend it for multi-class problems (e.g. one-versus-all or one-vs-one). In our experiments we chose to use the one-vs-one ([18]) approach in combination with crossvalidation. The method can be roughly described as follows; let N be the number of classes, $x \in \mathbb{R}^k$ a feature vector and $k_p : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ an arbitrary kernel parametrized by a set p. In one-versus-one the amount of SVMs to be trained is $N(N-1)/2 =: P$, i.e. one trains an SVM for each possible two-class combination. Formally stated, if $\tilde{x}_{i,\gamma,\delta}$ represents the i-th support vector (SV) for classes γ, δ, $\tilde{\alpha}_{i,\gamma,\delta}$ the corresponding coefficient and $b_{\gamma,\delta}$ the bias. Then the multi class classification result is given by first evaluating

$$c = (1 + \text{sgn}(b_{\gamma,\delta} + \sum_i^{M_{\gamma,\delta}} \tilde{\alpha}_{i,\gamma,\delta} k_p(\tilde{x}_{i,\gamma,\delta}, x)))/2 \tag{2}$$

followed by evaluating

$$v_{\gamma,\delta,x} = c\gamma + (1-c)\delta \tag{3}$$

which will assign $v_{\gamma,\delta,x}$ the value of either γ or δ. This can be seen as a vote for one of the involved classes, thus in a second step the class with the most votes among the P SVMs is determined and considered to be the final classification result. In order to measure the performance of the obtained SVM, i.e. the SVM for a certain parameter set p, we use the following n-fold crossvalidation (CV) strategy. Let S be the CV data set, first this S will be split into n equal sized disjoint subsets S_1 by randomly selecting elements from S. If $n \nmid |S|$ then only $n-1$ subsets will be created. Afterwards a multi class SVM will be trained on $S \backslash S_i$ followed by a performance evaluation on S_i. As the individual SVMs for a fixed parameter set will (if $n \mid |S|$) be evaluated on all subsets, the overall result will give a prediction of how good the SVM will perform on new data. Thus the crossvalidation corresponds to a parameter estimator. Finding the correct parameter set is a difficult and time consuming task, the canonical method is the so called grid search. If we assume a simple kernel $k(u,v) =< u, v >$, the only parameter for SVM training remains the constant $C \ni [C_s, C_e]$, which regulates the trade-off between classification errors and weight vector length in the transformed feature space (with e.g. a simple max-loss)

$$\min_w \left(||w||^2 + C \sum_i \max(0, 1 - y_i f(x_i)) \right) \tag{4}$$

During a grid search one probes with selected C values how the SVM performs, in our experiments we used an exponential sampling of $[C_s, C_e]$. This strategy corresponds to a brute force search and quickly becomes less efficient for parametrized kernels as e.g. the gauss kernel

$$k(u,v) = \exp(-\gamma ||u - v||^2) \tag{5}$$

The required time for such an approach can take between hours up to days, yet once the range for a specific type of data has been narrowed down, one can utilize this knowledge for future training procedures.

VI. Experiments

A. Neural Networks

We implement a multilayer perceptron (MLP) as described in Sec. IV-C using the freely available OpenCV library[19] and its C++ interface. Each experiment is performed 10 times with different initial conditions for the MLP, and the best result is retained. In these experiments, we systematically evaluate the influence of different confidence measures(”confOfMax”,”diffMeasure” or ”varianceMeasure”, see Sec. IV-C) on the fusion strategy (”add”, see Sec. IV-C) while measuring the performance of the first camera, the second camera as well as an ”early fusion” or a ”late fusion” of the two cameras. In order to test the influence of different 3D descriptors, we perform an identical evaluation except that the VFH5 point cloud descriptors is replaced by ESF. Additionally, we perform the same evaluation on an analogous database using the VFH5 descriptor where the angle between ToF sensors is 90 deg. Results are evaluated by default according to whether one among the S strongest output neurons coincides with the true class of a point cloud (”S-peak measure”). Unless explicitly states, we use $S = 1$. Results are given in Fig. 3. Several important aspects may be perceived: first of all, fusion strongly improves results in comparison to any single sensor, w.r.t. to the efficiency of sample rejection but also in absolute terms when no samples are rejected, corresponding to the intersection of the graphs with the right boundary of the coordinate system. Secondly, early fusion has slightly superior performance than late fusion but the difference is marginal, potentially giving a preference to late fusion due to reduced computational complexity. Lastly, the different confidence measure are consistently ranked throughout all experiments, with the ”diffMeasure” being the best-performing one, closely followed by ”confOfMax”. This is encouraging as especially confOfMax is computationally very lightweight, again favoring real-time execution. Thirdly, the angle between cameras does not seem to play a crucial role even though individual camera results differ considerably. Here, the beneficial aspects of fusion can be clearly demonstrated. And lastly, the ESF descriptor seems to perform slightly better than VFH5, which might lead us to prefer this descriptor as it is computationally simpler and requires constant execution time regardless of point cloud size. An interesting observation is that the two-peak measure enormously improves classification rates in all conditions. This is very useful for an application, especially for temporal filtering, as the behaviour of the second-strongest output can obviously also provide valuable information about the true pose class.

Training times are around 10min per single experiment, which outperforms an equivalent SVM-based (Support Vector Machine) ”one-versus-all” implementation by a large margin. Average execution times vary between 1-5 Hz depending of the use of the descriptor (ESF: 0.2s/0.2s for 30/90 deg. between cameras, VFH5: 0.4s/0.9s) whereas NN execution time is $< 0.005s$. On average the point clouds contain 1300-1600 points, depending on the angle between cameras and the distance of the recorded hand to each camera.

B. Support Vector Machines

In our experiments we evaluated two types of kernels, the ordinary scalar product and the gauss kernel. The parameters have been obtained via a grid search over an exponentially sampled parameter space. The crossvalidation set for ESF descriptors consisted of a total of 40000 feature vectors $x \in \mathbb{R}^{1280}$. All numerical values were sampled with double precision and special attention towards compare operations. In order to narrow down the search range we first applied a coarse crossvalidation with $n = 5$, as shown in [20] this reduces the variance of the obtained parameters yet increases the corresponding bias. Thus, once the area had been narrowed down we commenced a thorough gridsearch with $n = 100$. This method can be regarded as a two stage approach, Fig. 4 depicts the first stage results for an RBF kernel and VFH descriptors. One can see that the kernel parameters influence on the classification result becomes smaller for large penalty factors.

Using the scalar product, the best SVM obtained a classification performance of $\approx 98.7\%$ for 30 and 98.8% for 90 degrees. The time needed for the grid search was approximately 16 hours. Regarding the gauss kernel we obtained a classification rate of 99.8% for 30 and 99.6% for 90 degrees with a total training time of approximately 2 days.

The setup and evaluation procedure for the VFH descriptors is identical to the ESF case. The only difference lies in the reduced size of the VFH descriptors which contain only 616 elements. The results are summarized in table I. It must be mentioned that the VFH descriptors allowed a much faster parameter estimation due to the reduced data volume. Using the scalar product the needed time was 3 hours while ≈ 23 hours were needed for the gauss kernel. Although training has proven to be a slow procedure, classification of a single feature vector can be done very efficiently and takes only fractions of a second (e.g. 4μs / 250kHz for ESF descriptors on a modern CPU).

Descriptor	ESF 30	ESF 90	VFH 30	VFH 90
Classif. rate scalar kernel	98.7%	98.8%	96.9%	94.2%
Classif. rate gauss kernel	99.8%	99.6%	98.8%	93.1%

TABLE I: Classification results for both descriptor types. The SVMs were obtained through a two-stage grid search.

VII. Discussion and outlook

Analyzing the results in the light of the key research questions formulated in Sec. I, we can state that, first of all, fusion with data from a second ToF sensor improves results tremendously in all investigated conditions, camera setups

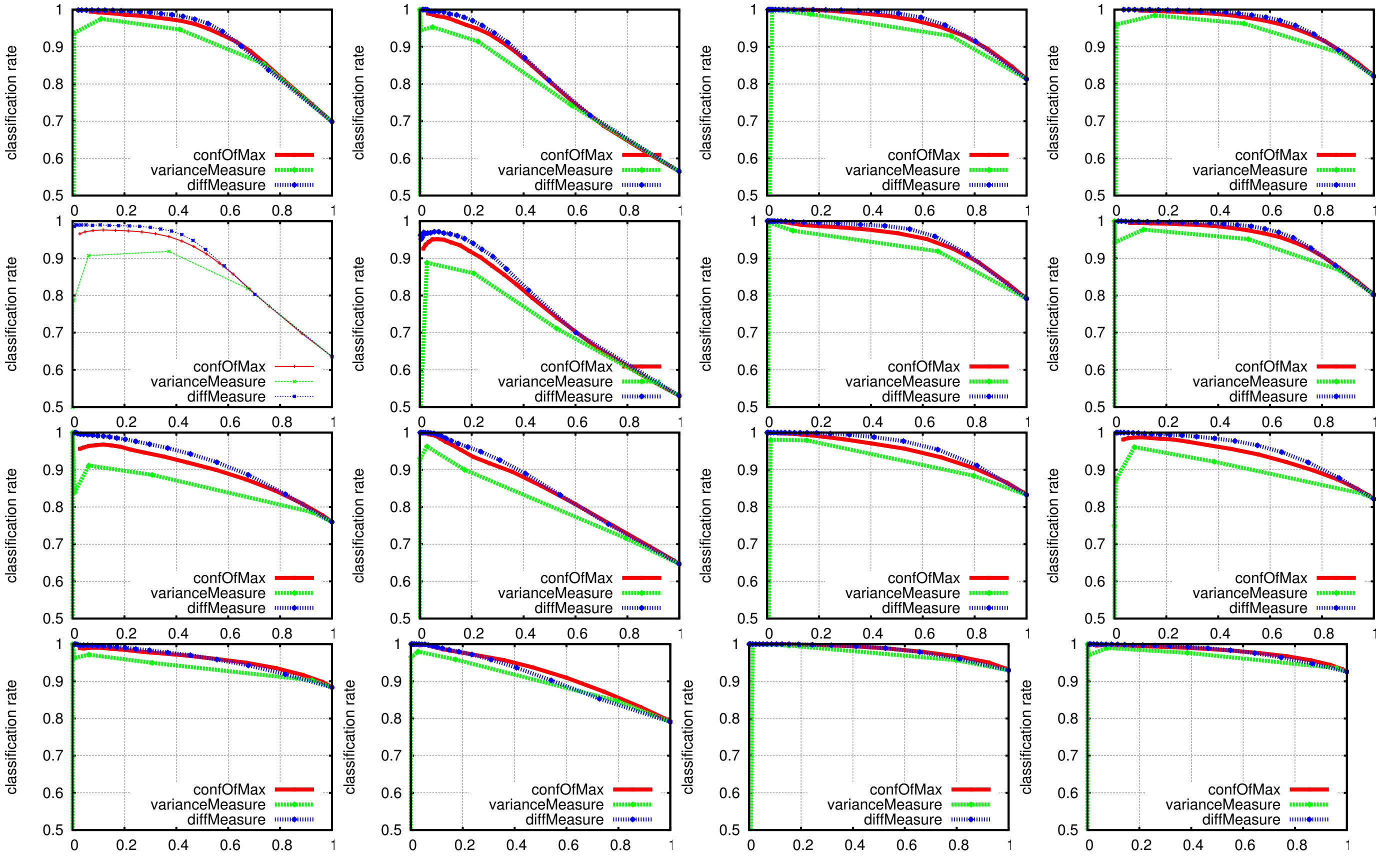

Fig. 3: Experimental results. First row: VFH5 descriptor, 30 degrees between cameras. Second row: VFH5, 90 degrees between cameras. Third row: ESF descriptor, 30 degrees between cameras. Last row: Same as third row, only classification errors evaluated using the two-peak measure, see text. In all rows, the order of diagrams is, from left to right: 1,2) first/second sensor 3) late fusion 4) early fusion. Individual plots show the effects of varying confidence thresholds on classification accuracies for several possible online confidence measures. We do not show the method-dependent confidence thresholds but rather the acceptance rates which vary if thresholds are varied. At the far right of each diagram, we recover the classification performance obtained when not rejecting anything, naturally leading to reduced performance.

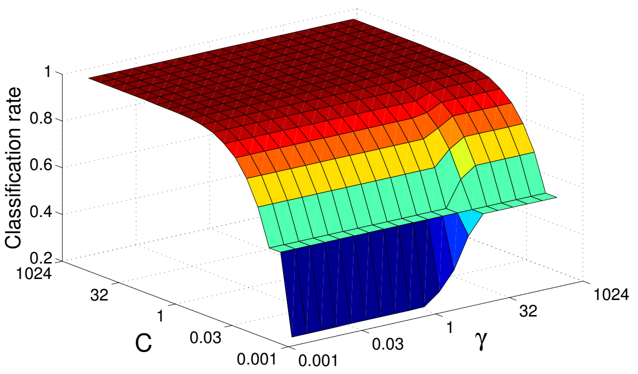

Fig. 4: Development of classification rate during a grid search for VFH descriptors using an RBF kernel. For large values of the penalty factor C the kernel parameter γ looses some of its significance.

and point cloud descriptors. Interestingly, late fusion performs globally just as well as early fusion, which is important as it has the potential to be much more computationally efficient. However, even when considering individual ToF sensors, the computation of confidence measures from output activity distributions is of tremendous impact as well. Confidence can be efficiently extracted at execution time (no need to see the class labels for this) and used to avoid classification decisions when they are likely to be incorrect anyway. We tested a number of information-theoretically motivated measures and luckily the most efficient measures seem to perform best. Concerning the influence of the used 3D descriptors: the ESF descriptor yields best performance with or without fusion. As this descriptor does not require normals computation and has approximately constant scaling behavior w.r.t. point cloud size, it is the most appropriate choice for real-time applications in the targeted automotive domain.

The use of support vector machines showed classification rates similar to these of neural networks. Gauss (or RBF) kernels are very common when it comes to SVM-based classification as they usually show the better results. In our case we found that for ESF descriptors, the standard scalar product allows the training of SVMs with a recognition rate close to RBF-based SVMs. The difference between both kernel types lies in the range of $\approx 1\%$ (with the RBF SVMs being better). There is effectively no difference between ESF descriptors for 30 or 90 degrees. The VFH descriptors showed very similar results, they as well lie only slightly apart when comparing both kernel types. Interestingly the VFH descriptor degenerates in its classification power for the 90 degree case. Overall the VFH descriptors seem to perform less effectively compared to the ESF features, which seems to indicate that the ESF descriptor retains more of the samples variance. Thus, using the scalar product, one can construct classifiers which are faster and structurally simpler than neural networks. As only a single scalar product has to be evaluated compared to more complex matrix operations of neural networks. Yet with the inherent drawback of huge training times one has to carefully assess the applicability of support vector machines. Neural networks should be favoured especially if it comes to online or mini-batch learning, whereas SVMs should be used in areas which do not require retraining and rapid deployment.

Summarizing, we have presented an adaptive data fusion approach for multiple ToF sensors addressing the generic task of 3D point cloud categorization in a multi-class setting. The fact of using a neural network for this purpose is of high advantage (besides very favorable database size scaling and multi-class issues) as the ensemble of normalized output confidences contains valuable information as well that can be efficiently exploited at runtime to improve results. Neural network learning furthermore removes the need for precise multi-sensor calibration as long as only categorization is targeted. Further work will include the comparison of both classification approaches in an automotive environment and the inclusion of a much larger database. Moreover we want to compare the live performance of our system under more challenging environmental conditions.

References

[1] T. Kopinski, A. Gepperth, S. Geisler, and U. Handmann, “Neural network based data fusion for hand pose recognition with multiple tof sensors,” 2014.

[2] 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*. IEEE, 2012, pp. 2748–2751.

[3] 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*, vol. 5, no. 3, pp. 334–343, 2008.

[4] S. Soutschek, J. Penne, J. 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*. IEEE, 2008, pp. 1–6.

[5] 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*. IEEE, 2012, pp. 72–77.

[6] 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*. Springer, 2014, pp. 403–414.

[7] M. Tang, “Recognizing hand gestures with microsofts kinect,” *Palo Alto: Department of Electrical Engineering of Stanford University:[sn]*, 2011.

[8] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, “Efficient model-based 3d tracking of hand articulations using kinect.” in *BMVC*, 2011, pp. 1–11.

[9] G. Bailly, R. Walter, J. Müller, T. Ning, and E. Lecolinet, “Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus,” in *Human-Computer Interaction–INTERACT 2011*. Springer, 2011, pp. 248–262.

[10] A. D. Wilson and H. 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*. ACM, 2010, pp. 273–282.

[11] R. Johnson, K. O’Hara, A. Sellen, C. Cousins, and A. Criminisi, “Exploring the potential for touchless interaction in image-guided interventional radiology,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 3323–3332.

[12] C. A. Pickering, K. J. Burnham, and M. J. Richardson, “A research study of hand gesture recognition technologies and applications for human vehicle interaction,” in *3rd Conf. on Automotive Electronics*. Citeseer, 2007.

[13] A. Znagui Hassani, B. van Dijk, G. Ludden, and H. Eertink, “Touch versus in-air hand gestures: evaluating the acceptance by seniors of human-robot interaction,” *Ambient Intelligence*, pp. 309–313, 2011.

[14] W. Wohlkinger and M. Vincze, “Ensemble of shape functions for 3d object classification,” in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2987–2992.

[15] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3d recognition and pose using the viewpoint feature histogram,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 2155–2162.

[16] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 3212–3217.

[17] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall, 1999.

[18] S. Knerr, L. Personnaz, and G. Dreyfus, “Single-layer learning revisited: a stepwise procedure for building and training a neural network,” in *Neurocomputing*, ser. NATO ASI Series, F. Souli and J. Hrault, Eds. Springer Berlin Heidelberg, 1990, vol. 68, pp. 41–50. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-76153-9_5

[19] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O’Reilly Media, Incorporated, 2008.

[20] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996. [Online]. Available: http://books.google.de/books?id=2SzT2p8vP1oC