A real-time applicable 3D gesture recognition system for Automobile HMI

Thomas Kopinski¹, Stefan Geisler¹, Louis-Charles Caron², Alexander Gepperth² and Uwe Handmann¹

Abstract—We present a system for 3D hand gesture recognition based on low-cost time-of-flight (ToF) sensors intended for outdoor use in automotive human-machine interaction. As signal quality is impaired compared to Kinect-type sensors, we study several ways to improve performance when a large number of gesture classes is involved. Our system fuses data coming from two ToF sensors which is used to build up a large database and subsequently train a multilayer perceptron (MLP). We demonstrate that we are able to reliably classify a set of ten hand gestures in real-time and describe the setup of the system, the utilised methods as well as possible application scenarios.

I. INTRODUCTION

As "intelligent" devices enter more and more areas of everyday life, the issue of man-machine interaction becomes ever more important. As interaction should be easy and natural for the user and also not require a high cognitive load, non-verbal means of interaction such as hand gestures will play a decisive role in this field of research. With the advent of low-cost Kinect-type 3D sensors, and more recently of low-cost ToF sensors (400-500€) that can be applied in outdoor scenarios, the use of point clouds seems a very logical choice. This presents challenges to machine learning approaches as the data dimensionality and sensor noise are high, as well as the number of interesting gesture categories. In this article we build upon earlier results [1] and demonstrate how a system can be developed and integrated into a car in order to be able to classify a gesture alphabet of ten hand poses. Our approach is purely data-driven, i.e. by extending and applying a Pointcloud descriptor to our needs we are able to set up a real-time applicable system which is robust versus daylight interferences, invariant to rotation and translation problems and moreover works without the need to formalise a possibly complicated hand model. We will first discuss the related work relevant for our research (Sec. II) and then go on to describe the setup of our system within an automobile environment. Subsequently we describe the sensors and the used database in Sec. IV. In Sec. V we go on to give an account of the used different holistic point cloud descriptors and explain the meaning of the parameter variations we will test. Sec. VI summarises the implemented NN classes and the choice of parameters. The key questions we will investigate in Sec. VII concern the generalisation error of the NN and the performance of our system in a live demonstration as well as offline and then we outline the procedure of the Live Demonstration in VIII. Lastly in IX we give an outlook of potential improvements of our system as well as our next steps.

II. RELATED WORK

Depth sensors allow for an easy and robust solution for recognising hand poses as they can easily deal with tasks as segmentation of the hand/arm from the body by simple thresholding as described in [2]. Several surveys have made use of this feature with various 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 utilising 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]. However for our purposes we need to employ several hand poses which are partly very difficult to disambiguate. 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-coloured pixels to allow for segmentation in 2D and 3D as well as tracking the hand. To our knowledge there is no comparable work which is placed in the automotive environment. [9] give an extensive overview of the methods and applications used for hand gesture recognition. One of their insights is that most applications are in the field of robot control, interactive displays/tabletops/whiteboards or sign language recognition. In [10] a case study is made of how the Kinect can be utilised to control E-Mail functions in a car through set of

在技术发展研究中，如何设计直观的用户界面。Bailly等研究了不同菜单技术的影响[13]。Wilson and Benko开发了一个系统，该系统使用多个投影仪和深度摄像头称为LightSpace[14]。

汽车场景已经开发了数年。在这种情况下，驾驶员可以将双手放在方向盘附近，以便能够专注于周围环境。点触功能可能有助于控制内容，同时能够专注于周围环境。良好的概述请参见[15]。

这些场景需要提供可靠的深度数据抽取技术，这是由上述提及的ToF传感器实现的。我们的方法表明，它可能实现 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.

### III. System Setup

我们集成两个ToF传感器到汽车中，分别固定到中心台架和连接到一台装有Linux系统的计算机上，处理计算任务。考虑到我们之前的结果，我们发现一个30度的角度对于摄像头的设置是非常合适的，以便可以对难解的手势进行区分。在这种情况下，我们在驾驶员的视角下放置了一个摄像头，由于明显的空间要求（参见图2）。与我们的前期研究相比，我们专注于识别主题（例如驾驶员）右手和希望的手姿。因此，我们定义了一个所需体积的兴趣（VOI），其中我们需要识别手姿。由于驾驶员的行为，可能出于位置和便利性以及与障碍的重叠（例如，方向盘）VOI具有梯形形状，具有深度范围27-35cm，最大宽度60cm和最小宽度约45cm，由FoV（Field of View）范围覆盖。相机范围从30-35cm。重要的是要注意，两个相机都被设置在同一参数下，以便粗糙地具有相同的距离到记录对象。此外，我们认为覆盖一个足够大的空间来识别手姿，与ToF传感器相比，传统的Kinect传感器。

### IV. The Database

该数据库使用两个ToF传感器（参见图1和2）收集数据，这些传感器由Camboard nano提供，可以提供深度图像的分辨率165x120px，并保持90fps的帧率。

照明波长为850nm，适用于各种照明条件。保持的稳健性优于日光干扰。由于ToF原理通过测量光从传感器到目标并返回所需的频率来工作，避免了测量误差，数据是通过交替捕捉每个传感器的快照来记录的。如图2所示，相机被安装在一个固定位置的距离大约为25cm，30度的角度从记录的物体。这允许对数据库中记录的物体进行建模，其中手可以放置在等距的点云数据集和因此可以对干扰进行区分。至少需要一个传感器，因此可以对手姿进行建模。研究重点已放在对静态手姿的识别上，这与动态手姿形成对比。每组手姿都与手部姿势有关，包括手势和手指。此外，驾驶员可以将手臂放在扶手上，这是合理的，因为扶手的位置。
to allow for creating a reproducible scenario which is also realistic in terms of applicability as possible (longer) user interaction might well occur in this manner. 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 each pose, a set of 2000 point clouds was recorded for each camera. Since we recorded hand poses from ten different persons independently, this yields a data set of 400,000 samples. Each person was asked to move and rotate the hand to a sufficient degree as to ensure enough variance in the point clouds. This is needed, since our approach is meant to be as invariant to translation and rotation as possible, also considering the fact that hand size varies significantly between the probands.

The main advantage of using a ToF camera is that it allows for outdoor use as e.g. contrasted to a Kinect. This is well demonstrated in our database as well as in the live demonstration as we recorded many samples over varying lighting conditions during bright or dim daylight without impeding the performance of the system. In the chosen probands, both male and female, the size of the hand ranged from 8,5cm - 9,5cm in width and from 17,0cm - 19,5cm in length.

V. POINT CLOUD DESCRIPTORS

All used global descriptors were calculated using methods of the publicly available Point Cloud Library (PCL) which were adapted to our needs.

A. The PFH-Descriptor

The PFH-Descriptor (PFH-Histogram) [16] is a local descriptor which relies on the calculation of the normals. It is able to capture the geometry of a requested point for a defined k-neighbourhood. So for a query point and another point within this 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 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.

B. 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 camera and concatenate all obtained descriptors in order to serve as inputs to the neural network.

VI. NEURAL NETWORK CLASSIFICATION AND FUSION

With M cameras, our system produces M descriptors per frame according to the methods described above. We use a multilayer perceptron (MLP) network[17] to model the multi-class classification function, either implicitly by concatenating all N descriptors followed by NN classification (“early fusion”), or explicitly, by performing classification individually on each of the N descriptors and then combining results (“late fusion”). Neural networks contain a bias unit at each layer, the training algorithm is “RProp”[17] with hyperparameters $\eta^+ = 1.2, \eta^- = 0.6, \Delta_0 = 0.1, \Delta_{\min} = 10^{-10}$ and $\Delta_{\max} = 5$. Network topology is $NK$-16-10 (hidden layer sizes from 10-500 were tested without finding significant performance improvements), K indicating the size of the used descriptors and N the number of cameras, here $N = 2$. Activation functions are sigmoid throughout the network. As the MLP classifiers have 10 output neurons, corresponding to the 10 gesture classes, with activities $o_i$, the final classification decision is obtained by taking the class of the neuron with the highest output. However, we do not necessarily wish for every classification to be taken seriously, and thus implement a simple extension based on a confidence measure $\text{conf}(\{o_i\})$. Final decisions are thus either taken by determining the neuron with maximal output, or by applying the confidence measure in the following way:

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

The confidence measure performs a mapping from $\mathbb{R}^{19} \rightarrow \mathbb{R}$ and shall be denoted "confOfMax":

$$\text{confOfMax}(\{o_i\}) = \max_i o_i$$

(1)

For performing late fusion, that is, obtaining two independent classifications $o_{1}, o_{2}$ based on each camera’s features, we simply calculate the arithmetic mean of both output vectors: $o_{\text{F}} = 0.5(o_{1} + o_{2})$. 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 $o_{\text{F}}$ can then be subjected to the decision rule of Eqn. (1).

VII. THE EXPERIMENTS

The experiments were conducted in such manner as to test the influence of the various parameters on the classification results. We tested various settings for the NN parameters and ended up having three layers of 1250, 16 and 10 neurons -
i.e. one hidden layer - for the input, hidden and output layer respectively. The number of neurons for the hidden layer was chosen as such because the overall performance of the NN peaked at this value during the course of the experiments.

The used NN code was taken from the OpenCV library. We used Rprop as the training algorithm and a sigmoid activation function. The input for the network is formed by taking the input cloud and create the customised descriptor for it. Doing this for each camera and concatenating them together forms the input which is fed into the NN. This procedure is done for training, testing as well as for the live demonstration described later on.

A. Baseline performance

The first experiment splits all gesture and person samples 50/50 for training and classification, which leads to an overall recognition rate of 95 percent without even applying the confidence measure as indicated in eqn.(1). Table I displays the classification error of the NN. From Tab. I we obtain an overall classification error of 6.3%. A few things can be observed:

- overall the performance of the net seems to vary between gestures: most notably c, e and f are the ones on which the NN performs worst, as opposed to gestures a and g, which are recognised best - 9503, 9758 respectively out of nearly 10000 samples recognised correctly.
- for every gesture - but above all, for the ones recognised worst - we can identify the one, for which it was mistaken the most as for instance gesture b is likely to be held for gesture c and vice versa; in other words, 'two' is likely to be held for 'three' and 'fist' is likely to be held for 'one'. This makes sense as these poses differ only slightly in terms of pointcloud size, shape or appearance in general. Similar observations can be made for gesture pairs (c ↔ e), (e ↔ g) or (f ↔ a).
- it is important to note, that while one gesture may be mistaken for another, the reverse is not always the case, at least not in the frequency (compare f → a and a → f)

B. Generalization to unknown persons

Table II represents the performance of the NN, trained using all persons except one, on data from the person not contained in the training set. Hence each row now adds up to ∼2000 samples. The overall classification error for this experiment is about 14% - nearly twice as large if compared to the table before. We can also derive a set of interesting observations here:

![Fig. 3: The hand gesture database: A set of ten hand gestures, counting from 1-5 and fist, flat, grip, L, point](image)
• the network performs best on poses f, h, i while the worst results are on a and e

• similar to the experiment described above, there are poses likely to be mistaken for others - here most notably: a ↔ i and e ↔ d. While this makes sense overall (compare the hand gestures visualised in the database in Fig. 3 for similarities), the described cases differ from the ones above and it is left to interpretation where this difference comes from. If a certain hand pose from a person differs significantly from the equivalents included in the training set, this case is prone to error.

• we can derive, that it is obviously possible generalise well by training a NN on our database for our given task while the more interesting fact is that this works better for some cases that others. It is e.g. clearly observable that hand pose a is very likely to be interpreted as hand pose i which can occur due to the fact that this certain proband poses in a similar way for this specific task.

We are able to retain most of the true positives and improve the overall classification error which drops to 3.63%

• The disambiguation problem remains for the most difficult cases - as an example gesture f is likely to be held for gesture a (cf. Fig. 3)

<table>
<thead>
<tr>
<th>a</th>
<th>30</th>
<th>b</th>
<th>40</th>
<th>c</th>
<th>50</th>
<th>d</th>
<th>60</th>
<th>e</th>
<th>70</th>
<th>f</th>
<th>80</th>
<th>g</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>110</td>
<td>120</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

When applying a very high threshold of Θ = 0.95, the following observations can be made:

• raising Θ close to 1 results in many samples being excluded from accepting as confident (cf. Tab. IV) as more than a third (34005) of all samples were rejected

• in most cases the recognition of false positives drops to (or close to) 0; those cases which are difficult to disambiguate are reflected clearly in the results

• some classes 'suffer' more than others in terms of recognition or rejection rate, e.g. hand poses c and e are subject to many exclusions which can be interpreted in such a way as that these gestures are probably more ambiguous than others

• we can also state the fact that as (fairly) many samples remain as false positives in some cases (300 in the case of gesture f → a). Confidence must be high for several neurons (i.e. hand poses) in cases like this.

• we are able to lower the classification error to 1.3% on average at the cost of the fact that some gestures are better recognised than others - this can be seen in the video of our live demonstration: we stabilise the overall performance of the system as a whole while recognition results fluctuate for a few examples

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

C. Improvement of baseline performance by thresholding confidence

Tables III and IV refer to a set of experiments conducted on the whole data set split into equal parts (~100000 samples for training and testing each) but this time varying the confidences to a threshold Θ, in order to classify only those samples we deem to be 'confident enough'. As before, several experiments have been run and we shall outline the observations:

• we have tested various confidence parameters Θ for the output neurons ranging between values [0,5,0,95] with 0.05 values for the steps taken; we chose to exemplary show the effect by taking 2 sample results for the values 0.65 and 0.95

• we allow only for accepting the classified category by the NN if the value of the highest neuron is above the chosen confidence value, else we reject the sample. This leads to the fact that with a higher chosen confidence value more samples are rejected (6776 rejected samples for Θ = 0.65 - Tab. III and 34005 rejected samples for Θ = 0.95 - Tab. IV).

• The desired effect of sorting out cases in which we are unsure is achieved by this parameter if we compare these results to the previous ones (cf Tab. I) as the number of false positives drops in all cases.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

D. Improvement of generalization to unknown persons

Tab. V and Tab. VI show the results of our system when evaluating it on data of a person excluded from the training set while employing the confidence measure. We
show exemplary results for the confidence parameters $\Theta = 0.75$ and $\Theta = 0.85$ (cf. Tab. V and Tab. VI respectively). Analogous to the experiments before we derive the following observations:

- The effect of introducing $\Theta$ when generalising on unknown data can be described as similar albeit there exist slight differences, namely the fact that we are testing on a smaller number of samples which quickly leads to a significant drop in the recognition of true positives. Overall the system performs well offline - though slightly worse compared to when employing the whole database.
- we achieve error rates close to 0 (or 0) in most of the cases with a (comparatively) small choice of $\Theta$ (cf. Tab. V)
- a good choice of $\Theta$ is crucial for our classification task as too many samples may be ruled out (cf. gesture g in Tab. VI) too early which leads to poor performance results in some cases.
- we are unable to achieve an equally good classification error as before due to the fact that too many samples are rejected (more than 65% in the case of $\Theta = 0.85$) which makes our system too unstable for some cases. As we describe later on, we are averaging results over time when utilising our system. Hence it remains rather acceptable to expect some false positives - which are then ruled out by a simple maximising function - than to increase the confidence in such manner as that too few samples are included. The latter case results in some hand gestures nearly not being recognised at all.
- averaging results over time does not help if $\Theta$ is chosen too high. We achieve best results for our Live Demonstration with $\Theta$ ranging in [0.6,0.8]
- an overall error measurement has to be taken cautiously, as the number of true positives varies significantly in between the classes $a$-$j$

| a | 0.02 | b | 0.72 | c | 0.06 | d | 0.00 | e | 0.00 | f | 0.00 | g | 0.00 | h | 0.00 | i | 0.73 | j | 0.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | 0.00 | b | 0.79 | c | 0.00 | d | 0.00 | e | 0.00 | f | 0.00 | g | 0.00 | h | 0.00 | i | 0.74 | j | 0.00 |
| a | 0.00 | b | 0.79 | c | 0.00 | d | 0.00 | e | 0.00 | f | 0.00 | g | 0.00 | h | 0.00 | i | 0.74 | j | 0.00 |

TABLE V: The classification results with confidence threshold $\Theta = 0.75$, the classification error is 7.89% with 9560 rejected samples

VIII. THE REAL-TIME SYSTEM

The real-time implementation is set up and described as in section III. The proband is seated and positioned in the same manner as the other probands during the recordings. The right arm is placed onto the armrest in order to be able to interact with the system properly. The lighting conditions for this experiment were the same (bright sunlight) as those for the other probands before. The proband tests all ten hand poses in an arbitrary manner, the cameras record the environment as described in Sec. III + Sec. IV and feed the transformed data into the NN which classifies the result. Every recognized hand pose is displayed, with the system running at a frequency of 5-6 Hz.

Test results are in general comparable to the offline system although an exact measurement is not possible as annotations were not yet created for the real-time in-car scenario. Based on a visual inspection of results, we observe very few errors if probands’ hand gestures are a part of the training set, otherwise performance suffers but not beyond a certain point. The recognition rate in the latter case is impeded by the fact that many of the chosen poses are difficult to disambiguate (see the used gesture alphabet in Fig. 3), e.g., hand pose 'two' vs. 'three' or 'L' vs. 'one'.

We were able to boost the performance of the real-time system by using the "confOfMax" confidence measure as in the offline experiments. Confidence thresholds in the range of [0.6,0.9] yield satisfactory results, by determining whether the system is ‘sure’ enough to recognise and classify a sample. Of course a balance has to be found between the need to have very accurate results, and the need to maintain a sufficiently high frequency of results. As we achieve already a satisfactory frame rate which can conceivably be boosted by optimizing the code, we believe that it is acceptable to reject half of the incoming samples in exchange for a high recognition accuracy.

IX. CONCLUSION AND OUTLOOK

In this article, we present 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. Tests on an offline database show excellent generalisation performance for a set of 10 static gestures. We furthermore present an in-car, real-time implementation of this system which is tested, in a non-rigorous way for the time being, on persons whose hands are not at all present in the training data, while still achieving very good performance given that a classification problem with 10 classes is studied. In particular, what we demonstrate is that already a computationally simple confidence measure boosts performance considerably at the cost of ignoring uncertain samples, which to our mind is strongly preferable to proposing an incorrect classification.

In future work, we will build upon these foundations and try to use more powerful heuristics to bring the recognition
rate close to 100 percent. Particularly, we intend to investigate the use of other, more advanced confidence measures, as well as the principle of temporal continuity (i.e., classifications do not usually fluctuate very rapidly). Beyond that, it is conceivable to devise schemes that compare output neurons not only based on their activation but also on statistical co-occurrence properties established during training, which might improve the disambiguation of problematic classes.

**References**


