Pointing at the HUD: Gesture Interaction Using a Leap Motion

Daniel Brand
University of Salzburg
Salzburg, Austria
daniel.brand@stud.sbg.ac.at

Kevin Büchele
University of Salzburg
Salzburg, Austria
kevin.buechele@stud.sbg.ac.at

Alexander Meschtscherjakov
Center for HCI
University of Salzburg
Salzburg, Austria
alexander.meschtscherjakov@sbg.ac.at

Abstract
Head Up Displays (HUDs) have the advantage to be visible in the line of sight of the driver and thus minimize visual distraction. Otherwise, it is not easy to manipulate them since they are virtually positioned behind the windscreen. We used a Leap Motion controller to achieve a gesture controlled HUD. We have conducted a simulator study with two variations of a simplified HUD: one with three segments and one with four segments. We show that the three segment HUD is superior to the four segment HUD in terms of interaction time and error rate. We provide data on the horizontal angle a HUD segment is chosen with the index finger of the right hand when selecting one of the three respectively four segments of the HUD. Our results can inform HUD interaction designers on interpreting mid-air pointing gestures to achieve higher success rates.

Author Keywords
Gesture recognition; Head Up Display; Leap Motion.

ACM Classification Keywords
H.5.2 [User Interfaces]: Interaction styles (e.g., commands, menus, forms, direct manipulation)

Introduction
Head Up Displays (HUDs) have the advantage that information is visible for the driver near the line of sight when look-

ing at the road ahead. Modern cars visualize different kinds of information in the HUD such as the actual speed, the actual speed limit, turn-by-turn navigation information, information from the adaptive cruise control system (e.g., target speed or speed limiter), or even information form the communication and entertainment system (e.g., caller ID, song name). This increased number of features of a HUD makes it crucial to be able to manipulate elements in a HUD. For example, one might add features or remove them, or even rearrange their order.

From an interaction design point of view this is not trivial since a direct manipulation as for example on a touch screen, where one can for example drag and drop objects, is not easily achievable. The visibility of a HUD is high, whereas its reachability is low. With some manufacturers (e.g., BMW or Mercedes) the HUD may be controlled with buttons on the steering wheel. They allow both the manipulation of the design of the HUD itself as well as the content it visualizes. This has the disadvantage to be not a direct manipulation and input and output modality are separated from each other.

Our approach is to use mid-air gestures as means to manipulate the content of a HUD. By simply pointing at different areas within the HUD with the index finger it should be possible to mark items inside the HUD. By performing a gesture (e.g., moving the index finger along with the whole arm towards the marked object) would select this item. We prototyped mid-air gestures by using a Leap Motion controller. This allows us to select items within a HUD without any other devices such as a touchpad, knobs etc.

Based on these considerations we were interested in how accurate such mid-air pointing gestures would be in in the setting of an in-vehicle HUD and how drivers would perform such gestures. In particular we were interested in how accurate and how fast a driver could select segments of a HUD and if there is any difference when we provide a different numbers of segments to be chosen. In our case we compared a HUD with three segments with a HUD with four segments. Second, we wanted to figure out how participants positioned their index finger when selecting a segment i.e. at which angle drivers would position their index finger. In order to answer these research goals we conducted a driving simulation study with ten participants. Our results show a difference in average interaction time and error rate with the two types of HUDs (3-segments vs. 4-segments). We also could see that the angle under which the segments were hit depended on the HUD type but also on the position of the segment itself.

Related Work

There have been an increasing number of mid-air gestures studies in the automotive UI community. Rümelin et al. [5] have shown that pointing as a lightweight form of gestural interaction is reliable achieving a recognition rate of 96 % in the lab. Ohn-Bar [3] have presented hand gestures to steer infotainment systems. May et al. [2] have shown that multimodal air gestures have advantages over conventional touch systems in navigating menus in the vehicle in terms of safety, but longer task completion and mental workload. Riner et al. [4] have provided a first standardization of the in-car gesture interaction space. Lauber et al. [1] have presented a Head-mounted displays (HMDs) which was indirectly manipulated by pointing gestures.

Prototype

We implemented a prototype which enabled us to highlight and select segments on HUD with mid-air gestures. For identifying mid-air gestures we used the the Leap Motion, which recognizes each finger including bones and joints. For visualizing the HUD we used a TFT monitor and a very

Figure 1: Simulated HUD with 3 segments labeled 1 to 3. Currently the user is pointing at element 3 and thus it is highlighted in blue. When the user would move the index finger of the right hand towards the segment it would be selected and highlighted in green.

Size snd distance

The size of the HUD was 900px (i.e. 24.5 cm) times 200px (i.e. 5.2 cm). The size of the elements in the 3-segment HUD was 200px times 300 px (i.e. 8.16cm), and 200px times 225px (i.e. 6.12 cm) in the 4-Segment HUD.

The driver's head was approx. 100cm away from the screen.

simple design. The visualization shows a rectangle with a predefined hight and width. The HUD could be divided into an adjustable number of similar segments. Thus, the more segments were chosen, the lower the width of each segment was. Segments were numbered increasingly starting with 1 from left to right. These numbers were used to instruct users which HUD segment they should select.

In order to combine gesture input from the Leap Motion and the HUD visualization we used the JavaFX framework. Certain gestures above the Leap Motion would cause an interaction with the JavaFX window (i.e. HUD visualization). The Leap Motion was set up to track two major gestures: pointing at a certain field and selecting it by moving the pointing finger slightly in direction to the HUD. The current marked element would get a blue background, while the selected element will be shown with a green background. In the case of our prototype, the Leap Motion framework and the JavaFX framework are running on the same computer.

Figure 1 shows the HUD visualization with 3 segments. They are labeled with numbers 1 to 3 from left to right. When the Leap Motion recognizes the index finger of the right hand pointing at one of the segments it would be highlighted in blue in order to visualize that it is marked. The user could select this segment by moving the index finger in the direction of the segment. This would result in highlighting the selected segment with green if the correct segment was chosen.

As mentioned before, the number of segments could be varied dynamically. For our study we decided to compare a HUD with three segments and one with four segments. Explorations with two and five segments showed, that selecting an element in the two segment HUD was too easy and to hard with five segments. In order to match the mid-air gestures to the selection of a segment we had to define under which angle the index finger of the right hand had to approach the HUD. In our setup we were only using the horizontal angle of the index finger of the right hand. It was assumed that the HUD was positioned in front of the user and the Leap Motion was aligned accordingly between the user and the HUD. When the user pointed with the right index finger in the center of the HUD this would result in a 90° angle between the finger and the HUD. When the finger would be turned to the right so that the finger and the HUD would be parallel to each other this would be an angle of 0°. Turning the finger in the other direction (i.e. left) would mean an angle of 180°.

For the study, we had to decide which angle intervals would correspond selecting which segments for both, the 3-segment HUD and the 4-segment HUD. In order to define these values, we explored different index finger positions. It became quite obvious in the beginning, that the anatomy of our hand allows a slightly bigger movement range of the right index finger to the left compared to the right and it seems to allow a more accurate granularity. It was also clear form the beginning that we could extend the interval for the outer segments to its limits. Based on these considerations and on a set of explorative studies with colleagues we ended with the following angle intervals. For the 3-segment HUD segment 1 was chosen when the finger approach in an angle between 180° and 100°, segment 2 between 100° and 80°, and segment 3 between 80° and 0°. For the 4-segment HUD the following intervals were chosen: segment 1 between 180° and 100°, segment 2 between 100° and 95°, segment 3 between 95° and 85°, and segment 4 between 85° and 0°.

Study Setup

Our study setup was built up in the driving simulator which consists of a wireframe mock-up and OpenDS as the driv-

Figure 2: Flattend steering wheel allows gestures to be made above the steering wheel rim.

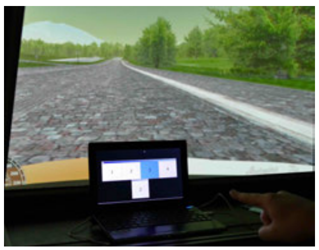

Figure 3: Test setup during the case studies with the driving simulator. Driver selects segment 3 of the 4-segment HUD. Below the 4 segments the to be chosen segment number is displayed.

ing simulation environment. A netbook running eclipse executed the program and simulated the HUD. The Leap Motion which was plugged into this netbook was placed right behind a Fanatec steering wheel. It was shaped flattened at the top and the bottom so that gestures were possible to be made above the steering wheel (see figure 2).

For the study we used an in-between subject design with two conditions (3-segment display and 4-segment display). The study setup for each participant was as follows: After filling in an informed consent and answering some demographic questions participants were allowed to get comfortable with the driving simulator and also had the possibility to explore the gesture interface. Then each participant had to complete the two conditions while driving in the simulator. Conditions were counter balanced to reduce order effects. For each condition participants had to select a given segment by pointing at the corresponding segment. The segment number was randomly chosen and visualized in a small window below the HUD (see figure 3). When the participant pointed with the index finger of the right hand at the desired element is marked blue. Then the participant can select it by moving the pointing finger slightly forward (similar to a screen tap gesture). After successfully (or not) selecting a segment the next segment number was visualized after a delay of 3.5 sec. This was done so that the participant could put his/her hand back to the steering wheel.

This makes sure that his hand does not keep on pointing onto the selected segment which would make the next selection easier if the same segment has to be selected. Furthermore, we think it is more realistic when the user's hand has to be moved from the steering wheel to the HUD. This process was repeated 15 times for each condition. Throughout the study, participants had to keep on driving in the driving simulator since we wanted our study setup to be as realistic as possible. After the study each participant had to fill in a short questionnaire and took part in a semi-structured interview.

Each gesture was logged in a log file. The logfile includes basic settings such as number of segments, implemented angles, condition order, touched elements and the duration of the whole test. Above that it holds for each interaction the angle at which the gesture was performed as well as hit and error rates (i.e. if the visualized segment was hit or not).

Study Results and Discussion

Ten participants conducted the study (5f, 5m, aged between 20 and 30). Participants owned a driving license and were right-handed. They could practice selecting elements in the HUD while driving. It became soon obvious that each test person got familiar with the control quite fast.

When comparing the two conditions we could see that the 3-segment HUD was superior to the 4-segment HUD. Table 1 shows that the average time it took for a participant to select a segment in the 3-segment HUD condition was 4 seconds, whereas it was 10 seconds in the 4-segment HUD condition. Also the error rate for the 3-segment HUD condition (18.1%) was significantly lower than for the 4-segment HUD condition (24.8%). This is not very surprising, since with a 3-segment HUD the middle segment could be much wider than in the 4-segment condition. What surprises is the amount how better participants performed with the 3-segment solution.

	3-segment HUD	**4-segment HUD**
Avg. time	*4 sec.*	*10 sec.*
Avg. error rate	*18.1%*	*24.8%*

Table 1: Test results with 3 and 4 elements

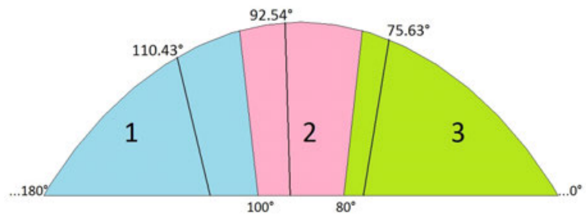

Figure 4: Three segments and the angles of the average hit points.

	Seg. 1	Seg. 2	Seg. 3
Interval	*180° - 100°*	*100° - 80°*	*80° - 0°*
Avg. hit	*110,43°*	*92,54°*	*75,63°*

Table 2: Average hit points with the 3-segment HUD.

Regarding our second research goal (i.e. at which angle gestures are made) we found that for the 3-segment HUD that the average hit angle for segment 1 was 110.43°, for segment 2 it was 92.54°, and for segment 3 it was 75.63° (see Figure **??** and Table 2). This means that participants' gestures had a slight offset to the left. For example, the theoretically ideal angle to select the segment in the middle (i.e. segment 2) would be at 90°, but our participants rather hit it on average 2,54° left to that. This is not only true for the segment in the center but also for the left and right segment. We chose a value between 180° (left border) and 100° (right border) for the left segment (i.e. segment 1) and it was hit on average 10.43° left of the right border. In comparison to that the right segment (i.e. segment 3) was hit only 4.37° right of its left border (i.e. 80°).

Similar results can be seen with the 4-segment HUD (see Figure 5 and Table 3). Segment 1 was hit on average 8.97° left to its right border (i.e. 100°), whereas segment 4 was

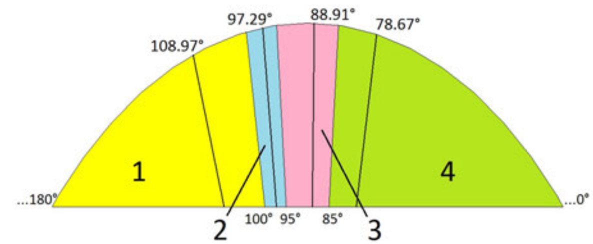

Figure 5: Four segments and the angles of the average hit points.

	Seg. 1	Seg. 2	Seg. 3	Seg. 4
Interval	*180° - 100°*	*100° - 95°*	*95° - 85°*	*85° - 0°*
Avg. hit	*108.97°*	*97.29°*	*88.91°*	*78.67°*

Table 3: Average hit points with the 4-segment HUD.

hit on average 6.33° right from its left border (i.e. 85°). Segment 2 was hit at an angle of 97.29° and segment 3 was hit at an angle of 88.91°. Here we have to mention that we decided not the distribute angle intervals equally between these segments, since our pre-explorations suggested that we can give segment 2 a smaller interval in comparison to segment 3.

Another finding was, that despite the fact that in both conditions the width of the HUD was exactly the same, that average hit points for the 4 segment conditions were closer to the center than in the 3 segment condition (108.97° and 78.67° vs. 110,43° and 75,63°). This suggest that the a more fragmented visualization leads to a smaller range of gestures.

The interview and questionnaire at the end of the study revealed that the proposed system seems to be rather intuitive to use and has potential to be used in the future.

However, it was also mentioned that the system may distract the driver too much. Another limitations of our study is that it took place in a simulator. We assume that it might be harder to perform gestures in a driving vehicle with centrifugal forces and bumpy roads. Second, we limited our studies to 3 and 4 segments and chose specific angle intervals for these two conditions. Thus, results might not be generalizable. Third, we chose not to implement a real HUD but used a screen to visualize the HUD without the ability to see through it. A real HUD might even be harder to target at. Finally, we chose to position the Leap Motion above and behind the steering wheel. This had the advantage that it is closer to a real world gesture of pointing at something with your index finder, but might be problematic while driving.

Conclusion and Future Work

In this paper we have presented an approach to use mid-air gestures to manipulate the content in a HUD. We have shown that segmenting the HUD in three parts is superior to a 4 segment solution in terms of interaction time and error rate. We have also shown that drivers tend to hit a target slightly to the right, which could be an important information when designing such systems. finally, we have shown that a more fragmented HUD would lead to smaller pointing gestures in terms of angle interval. The main goal of our prototype was to figure out which system settings are the best for a comfortable control of a HUD. In our opinion, we came to a satisfying result regarding this goal. When it comes to the angle intervals, one has to mention that they might differ from car to car as they depend on the location of the Leap Motion. Regarding gestures, we tried to find a set of those that are recognized by the Leap Motion in a satisfying rate. However, there is still some room for improvements such as different gestures. Furthermore, more functionalities can be added in case of future studies on this topic.

REFERENCES

1. Felix Lauber, Claudius Böttcher, and Andreas Butz. 2014. You'Ve Got the Look: Visualizing Infotainment Shortcuts in Head-Mounted Displays. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '14)*. ACM, New York, NY, USA, Article 3, 8 pages.

2. Keenan R May, Thomas M Gable, and Bruce N Walker. 2014. A Multimodal Air Gesture Interface for In Vehicle Menu Navigation. In *Adjunct Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '14)*. ACM, New York, NY, USA, 1–6.

3. Eshed Ohn-Bar, Cuong Tran, and Mohan Trivedi. 2012. Hand Gesture-based Visual User Interface for Infotainment. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*. ACM, New York, NY, USA, 111–115.

4. A. Riener, A. Ferscha, F. Bachmair, P. Hagmüller, A. Lemme, D. Muttenthaler, D. Pühringer, H. Rogner, A. Tappe, and F. Weger. 2013. Standardization of the In-car Gesture Interaction Space. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '13)*. ACM, New York, NY, USA, 14–21.

5. Sonja Rümelin, Chadly Marouane, and Andreas Butz. 2013. Free-hand Pointing for Identification and Interaction with Distant Objects. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '13)*. ACM, New York, NY, USA, 40–47.