

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/261878630>

Hand Gestures to Control Infotainment Equipment in Cars

Conference Paper · June 2014

DOI: 10.1109/IVS.2014.6856614

CITATIONS

44

READS

4,588

3 authors:



Jose Luis Alba-Castro

University of Vigo

122 PUBLICATIONS 1,679 CITATIONS

SEE PROFILE



Elisardo González Agulla

University of Vigo

20 PUBLICATIONS 489 CITATIONS

SEE PROFILE



Francisco Parada Loira

University of Vigo

6 PUBLICATIONS 70 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Biosecure [View project](#)



Biometrics [View project](#)

Hand Gestures to Control Infotainment Equipment in Cars

Francisco Parada-Loira^{1,2}, Elisardo González-Agulla¹, José L. Alba-Castro¹, *Member, IEEE*

Abstract—Controlling multiple devices while driving steals drivers’ attention from the road and is becoming the cause of accidents in 1 out of 3 cases. Many research efforts are being dedicated to design, manufacture and test Human-Machine Interfaces that allow operating car devices without distracting the drivers’ attention. A complete system for controlling the infotainment equipment through hand gestures is explained in this paper. The system works with a visible-infrared camera mounted on the ceiling of the car and pointing to the shift-stick area, and is based in a combination of some new and some well-known computer vision algorithms. The system has been tested by 23 volunteers on a car simulator and a real vehicle and the results show that the users slightly prefer this system to an equivalent one based on a touch-screen interface.

I. INTRODUCTION

Driver distraction is one of the principal causes of car accidents [1] [2]. Interaction with the infotainment devices like mobile phones, GPS, radio, and, recently, the use of tablets to view web pages, watch videos or look at maps, is one of the causes with increasing prevalence. Since infotainment devices are increasing their presence in the new car models, there is also an increasing interest to develop safer user interfaces that allow drivers to keep their eyes on the road [3]. In the last few years, many systems were developed to provide the user with a natural way to interact with the infotainment. These systems lie in four principal categories: haptic devices (buttons, knobs, tablets or other touching devices) [4], electronic sensors [5], voice controls [6] or vision based systems detecting features like eye gaze, head pose or hand gestures. All these systems have their pros and cons. Voice-based interaction is maybe the most natural, but, even when this technology is commercially available in many models, it only works accurately in acoustically clean environments provoking some degree of stress for the user, otherwise. Touching-based devices are comfortable when they are located in reachable ergonomic places and the user has memorized the location associated to every function, but as the ergonomics decreases and functionality increases, the necessity to glance at the device to search for the desired function, also increases. Gesture-based devices are midway between touching the knobs and speaking to the system, so it inherited some of the pros and cons of both worlds. On the one hand, drivers are accustomed to using a hand to access car parts, like the shift-stick or the controls of the front panel, so, gesturing is just an evolution of those actions but

with the advantage of not needing a glance to the target. On the other hand, gestures, like uttered words, are not always executed exactly equal and are influenced by acquisition conditions, so they are prone to errors due to ambiguity on the realization and “noise” effects of imaging.

Detecting, tracking and segmenting hands in the interior of a moving vehicle is a difficult task due to strong and sudden changes in light intensity and color temperature. The vast majority of previous works on hand detection and gesture recognition were developed in controlled scenarios, many of them in indoor environments. The most often used techniques to detect hands include skin segmentation [7][8], color clustering, boosting features [9], image differencing [10], background segmentation, optical flow, motion detection [11], HOG descriptors [12], thermal imaging [13], multi-cue analysis like HOG+context+skin [14] or more recently the use of depth sensors like Kinect or similar devices [15] [16]. But even resorting to a depth camera solution has problems. In our initial viability tests, the use of Kinect-type sensors was discarded due to malfunction when the reflection of bright sunlight occludes the infrared light of the sensor, a similar problem with infrared reflecting surfaces in the pioneering work of Aykol et al. [17]. To avoid these and other difficulties, other systems rely in the use of multiple cues to add robustness to the system, like HOG+SVM+Skin+Depth [18] [19]

The design of our system was guided by the necessity of an easy to use, intuitive, robust and non-intrusive interface, so we resorted to an interface based on hand gesture recognition that monitored the gear shift area. After the preliminary tests with depth cameras, we decided to rely only on visible and infrared imaging. In this article we explain the vision algorithms we have integrated, modified or created to handle a great variety of driving scenarios, both for gestures made by the driver and the passenger, and with largely different lighting conditions. We have also performed a survey of 23 subjects comparing a touch-based interface with a gesture-based interface.

The rest of the paper is structured in the next sections. First we give an overview of the whole system and we explain the rationale of our designing decisions. Section III delves into the vision-based algorithms and explains with more detail the main algorithmic contribution of our work. Section IV is devoted to the usability survey and Section V discuss the results and concludes the paper.

II. OVERALL DESCRIPTION OF THE SYSTEM

A. Global Functionality

The system had to be designed to control several functions of the infotainment service, like moving forward

¹Elisardo González and Jose L. Alba are with University of Vigo, Spain. e-mail:eli.jalba@gts.uvigo.es).

²Francisco Parada were with University of Vigo and now is with the Galician Automotive Technological Center (CTAG), Spain (e-mail: francisco.parada@ctag.es).

This work has been supported by funds from the Galician Government through contract IN827J and project CN 2012/260 “Consolidation of Research Units: AtlantTIC”.

and backward through menus, selecting options, increasing/decreasing the volume or other continuous parameters, and selecting from a numbered list (0-9). The user would have to perform the predefined set of static and dynamic hand gestures in an ergonomic predefined area. The functionality had to be the same with daylight and infrared light. When the system is ON the predefined area should be continuously monitored. A Finite-State-Machine should be aware of the state of the system at any moment.

Fig. 1 shows some code-word examples of the hand gestures in the codebook. This is composed by a set of static gestures and another set of dynamic gestures. Static gestures are the numbers from one to five as shown in the top row of Fig. 1, open/closed hand, as shown in second row, and an “OK” gesture (third row, left). The set of dynamic gestures comprises hand raising (third row, right), down, left and right movement; left-right shaking, XY mouse-like movement and left-right twist of the wrist. Similar code-words, like passing the open hand from left to right, shaking the hand or making a left-to-right hand mouse-like movement, are differentiated by taking into account the speed of the gesture and the overlap with the predefined target area, as shown in Fig. 2. All the gestures can also be performed by the passenger, so, the coding must accept gestures produced by right and left hands indistinctively.



Figure 1: Examples of several hand gestures in the code-book

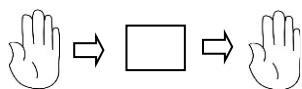


Figure 2: dynamic gesture code-word of left to-right movement passing through the target area.

B. Global Technical Restrictions

For the system to operate properly the captured image should have a minimum dynamic range and the percentage of overexposed and/or underexposed area should be small or shouldn't affect too much the predefined target area. The integration time of the camera should be small enough to ensure no motion blur affects the image of the hand. The frame rate should be at least 15 frames/sec. These two parameters limit the maximum speed to perform a dynamic gesture. The minimum speed of the dynamic gesture should be bounded to avoid confusion with static gestures. The depth of field of the image should ensure that the hand appears focused in all the height range needed to perform the gestures. All these technical restrictions are easy to fulfill in a large percentage of driving situations. The system should be prepared to react, through the Finite State Machine, to those rare cases where some of them are not met.

C. Constrains of the Application Scenario

The computer vision algorithms that have to extract the data from the frames to operate the infotainment interface, must exploit information from the application scenario. A clear understanding of it and its constraints can save thousands of code lines and make the application work robustly.

Given a fixed location for the camera on the ceiling and between the driver and passenger seats, the captured image has a static background (the cabin components) only perturbed by small vibrations and changing light and shadow areas. Objects of a specific range of sizes that move in the gestural target area can be considered hands, with high probability. To tell apart a hand from other objects invading the gestural area it is possible to code hand physical properties like, color (when working in visible spectrum), shape, size; and to use constraints of the application scenario, like the gestures' codebook and the speed and trajectory of the hand. So, any object that appears as foreground above the cabin components, and it is not a hand performing a gesture of the codebook, can be considered as background after a prefixed amount of time (legs, clothes, random objects, or even a hand that keeps on grabbing the gear shift).

One of the more restrictive constrains of the application scenario is the necessity of the interface to respond in real-time to the hand gestures. This precludes the use of computationally heavy algorithms. In the next section we will explain the computer-vision solution that we designed to fulfill all the requirements.

III. COMPUTER-VISION ALGORITHMS

The core of the system is composed by the computer-vision algorithms, that process the input video stream, detecting, segmenting, tracking, classifying and coding the hand gesture to feed the interface manager. The interface manager interprets the code-words in the context and takes an operational decision. The next subsections explain the main parts of the computer-vision algorithms.

A. Detection and segmentation of the hand

A versatile modelling of the background scenario is a key step to obtain a robust detection and segmentation of the hand. The model must have a dynamic behavior to accommodate the multiple scene changes due to varying lighting conditions (sunlight, overcast, infrared), shifts of shadows due to turning the car or driver/passenger movements, partial occlusion of the background with body parts or other objects, etc.

Three modules work in parallel to analyze the scene and segment a hand if it is present:

- Edge-based Foreground-Background Model (EBM)
- Mixture of Gaussians background Model (BG_MoG)
- Maximally Stable Extremal Regions segmenter (MSER)

These three modules are fed with the current frame (grayscale or color) and they combine their outputs to make more robust decisions. We have developed the FGBG_Edges module from scratch and modified the BG_MOG and MSER modules of the OpenCV library [20].

Among the features that are not affected by changes in illumination are the edges (except in the case of oversaturation or under illumination of the scene). So, we model the edges of the background of the image and use the temporal changes of those edges to analyze candidates of occluding hands.

There are two ways of constructing the model of the edges of the scene: static and dynamic. A static model is calculated averaging a large number of images of the edges of the background and thresholding the result. This training can be made offline, and has the advantage that no new calculations are needed on running time. Of course, it has the disadvantages that it must be trained for every car model, suffers with the vibration of the car and it doesn't adapt to new edges introduced by shadows or occluding things.

The dynamic model solves some of these issues learning the model of the background edges in real time. The learning process works with the following equation:

$$EBM_t = \alpha EBM_{t-1} + (1-\alpha)EI \quad (1)$$

where EBM is the edge background model, α is the learning rate, and EI is the current edge image, output of a Canny filtering on the input grayscale image [21]. To speed up the computation time, we add directionality to the edges, classifying the results of the Canny image in 4 classes using the Local Contour Patterns (LCP) features [22]. The classes are horizontal edges, vertical edges, diagonal left and diagonal right edges.

$$EBM_t(c) = \alpha EBM_{t-1}(c) + (1-\alpha)EI(c) \quad (2)$$

where $c = 1 \dots 4$ refers to the possible values of the selected LCP classes for each pixel.

The main objective of this modelling is to detect the new edges and those that have been occluded in each frame. To detect these edges we work with 2 models: a short term

model and a long term model. The long term model ($LEBM$) has a large learning rate α . This means that it takes more time to adapt to changes in the edges (caused by the users of by changes in lights or shadows). On the other hand, the short term model ($SEBM$) adapts very quickly to the new edges or the occluded ones. Setting $\alpha \sim 0$ we avoid a delay in the adaptation of the model to the appearing or disappearing edges, however this α makes the short-term model noisier.

Once the models are updated in every frame, they are thresholded to get rid of noisy edges (Th_{LEBM} , Th_{SEBM}). From the two updated models we define the images with Old Edges, New Edges and Hidden Edges (occluded):

$$OE_t(c) = Th_{LEBM}_t(c) \cap Th_{SEBM}_t(c) \quad (3)$$

$$OE_t = \bigcup_{c=1, \dots, 4} OE_t(c) \quad (4)$$

$$NE_t(c) = Th_{SEBM}_t(c) > Th_{LEBM}_t(c) \quad (5)$$

$$NE_t = \bigcup_{c=1, \dots, 4} NE_t(c) \quad (6)$$

$$HE_t = Th_{LEBM}_t > OE_t \quad (7)$$

These three edge images are the output of the Edge-based Foreground-Background Model that will be combined with the other two modules' outputs. Fig. 3 shows an example of a gestured frame (top-left) and the corresponding LCP image (top-right, showing different colors for horizontal, vertical and diagonal edges), $LEBM$ binary image (center-left), OE binary image (center-right), NE binary image (bottom-left) and HE binary image (bottom-right).

The other two modules whose outputs are combined with these three binary images are explained next.

Modelling the image color with a Mixture of Gaussians to obtain a background model, is one of the best techniques to detect and segment objects in the foreground when there aren't sudden illumination changes in the scene. To obtain a model of the background we use the algorithm [23] that is implemented in the OpenCV library, but with the next modifications to improve the segmentation of objects in the foreground when working with color images.

We use the method with a color space that separates the luminance and the chroma, so our modifications are still compliant with grayscale images, like the infrared ones used in the night mode of the system. The original method in the OpenCV library assigns the same variance to the three color channels of the image (RGB isotropic variance), but with our modification we allow more variance to the luminance channels than to the chroma channel. This way, changes due to varying lighting conditions are less likely to yield a foreground candidate than changes due to an occluding hand with a different chromaticity than the background.

The other modification to the OpenCV algorithm, that comes handy for this scenario, consist of classifying both shade and bright areas as those that have slightly smaller or bigger luminance than the BG_MoG model. This modification helps to avoid false candidates in neighborhood areas of an entering or exiting object that moves too fast to be integrated into the background model.



Figure 3: Original frame (Top-Left), LCP (Top-Right), *LEBM* (Center-Left), *OE* (Center-Right), *NE* (Bottom-Left), *HE* Bottom-Right)

The MoG model produces then, for every new frame, a labeled image with three pixel categories: foreground candidates, shade/bright and background.

The third module segments the image into uniform regions using the Maximum Stable Extremal Regions (MSER) algorithm [24]. The implementation of this algorithm in the OpenCV library is quite efficient. This algorithm works both for grayscale and color images, but we found no advantage of applying it to color images in this scenario. The output of the MSER module is an image of segmented regions. This image is readily combined with the output of the EBM module to assign a probability for each region as a function of the number of new edges, the hidden edges and the old edges, that the region enclosed. Fig. 4 shows the combination of both modules in the upper row, where lighter gray means more probability of being foreground (left image) or being background (right image). In the bottom row the resulting masks after thresholding are shown.

This way, the combined MSER+EBM outputs two candidate image masks: foreground (MSERFG-Edges) and background (MSERBG-Edges). These masks are then finally combined with the output of the MoG module to label undecided pixels as foreground or background. A region in the MSERBG-Edges mask cancels out all the Foreground pixels of the output of the FG-MoG (eq. 8), and the regions in the MSERFG-Edges that do not appear in the FG-MOG output are added to it (eq. 9).

$$FG-MOG \leftarrow FG-MOG \cap (\text{not MSERBG-Edges}) \quad (8)$$

$$FG \leftarrow FG-MOG \cup MSERFG-Edges \quad (9)$$

Fig. 5 shows an example of the FG-MoG output after being combined with the outputs of the MSER and EBM.



Figure 4: Combination of MSER and EBM modules. Top Left: Probability of Foreground regions, Top-Right: Probability of Background regions. Bottom row: thresholded probability images.

B. Finding, coding and classifying candidate gestures.

Once all the Foreground Masks have been obtained for a particular frame, it is time to check if one of the foreground regions is a hand producing a gesture of the codebook. At this point some other constraints of the application scenario can be applied. First of all, regions with a size smaller or larger than a threshold are filtered out. Then the silhouette of all the segmented regions in the foreground are extracted and smoothed. The smoothing filter is implemented by truncation of the high frequency coefficients of the Fast Fourier Transform of the closed contour. After smoothing the contour, the silhouette is coded as a series of concave and convex curves that represent the fingers and the valleys among them. The arm parts are also filtered out by locating the center of the hand using the Distance Transform and the Minimum Bounding Box of the candidate region. Fig. 5 shows the silhouette of two candidate regions, one is the hand is the other is a false positive.

Now, the filtered silhouettes are added to a tracking list. We use the Kalman filter to track every candidate across successive frames. Each tracked candidate is assigned an identification number, so each trajectory of its mass center and Bounding-Box can be analyzed to check if it make up meaningful dynamics for the system.

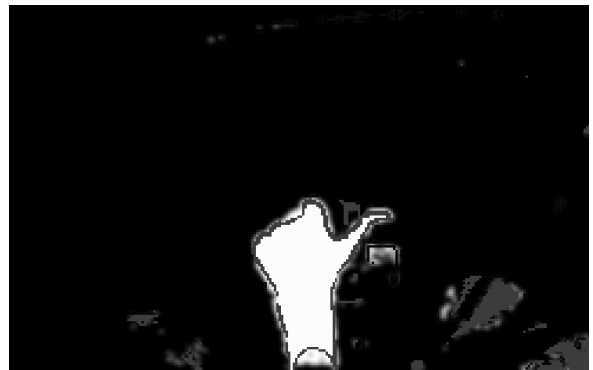


Figure 5: Combined output of MoG, MSER and EBM. Silhouette two candidate regions are overlaid.

The silhouette codes in the tracking list are compared against the static code-words in the system code-book. A previous screening is made by checking the number of convex points of the code. The candidate is discarded if that number is not equal to a valid one. Each code-word contains the normalized distance and angle of each convex point with respect to the center of the hand and the direction of the arm, respectively, so the code is rotation invariant. Fig. 6 shows some pre-coding examples. The best-match is associated to the silhouette code in each frame and accumulated across frames. Only when the trajectory of the tracker intersects with the target region, the computer-vision module is allowed to send information to the application control interface. The accumulated list of best-match static code-words is combined with the trajectory dynamics to decide on the static or dynamic gesture that has been produced. Fig. 6 shows the overlaid area that needs to be intersected to launch an event. There is another area defined in the upper part of the image, where some car models can have a touch screen. Gestures detected while intersecting with that area could be touching gestures, so no message is sent to the control interface.

C. Quantitative Results.

The system was tested with a dataset composed by 189 video sequences, 98+71 with driver and passenger gestures, respectively, recorded by 4 people. The sequences contained at least 10 instances of 8 static and 9 dynamic gestures. The system was developed using two different hands (1 driver and 1 passenger) and tested with the other two. Static gestures yielded a 96% recognition rate. Dynamic gestures were more difficult to tell apart mainly due to the similarity of left and right movement; left-right shaking, and left-right twist of the wrist. The results fell to 87%. We argue that this figure can increase with more adaptation of the users to the system.

IV. SURVEY

The survey has been designed to validate the performance of the proposed system and compare it against a touch screen system, both in a car simulator and in a prototype vehicle. The survey has been designed and carried out by the team at the CTAG (Galician Automotive Technological Center). They recruited 23 volunteers (14 male and 9 female), divided into 2 groups (12 and 11) and with ages between 22 and 38. The volunteers had to evaluate two systems (Touch-Screen and Hand Gestures) scoring each question from 1 to 10. The order of execution was switched for each group to avoid any bias. Table I reflects the test protocol.

The systems were tested with the prototype car (Audi A6) stopped to avoid hazardous situations. The car simulator was allowed to be driven up to 120 Km/h. The sequence of commands for the infotainment was the same for all the participants and for the two cars. The participant had to select the album $n \rightarrow$ song $m \rightarrow$ increase/decrease the volume \rightarrow play the song \rightarrow change the song \rightarrow increase/decrease the volume \rightarrow menu back \rightarrow menu back \rightarrow select songs \rightarrow chose song $l \rightarrow$ play the song.

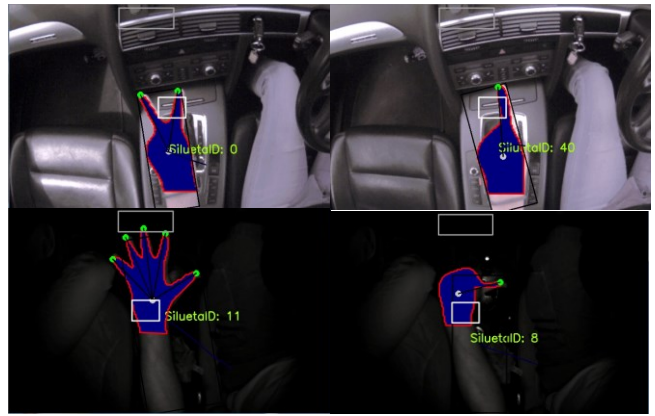


Figure 6: Some examples of tracked hands. The top two images correspond to a session recorded in a trip with visible light and color images, and the bottom two correspond to a session recorded in another car model with another driver and in a night trip with infrared illumination of the cabin.

TABLE I: TEST PROTOCOL

	1 st day: test in prototype	2 nd day: test in simulator
Group 1	1st Touch-screen	1st Hand gestures
Group 1	2nd Hand Gestures	2nd Touch screen
Group 2	1st Hand gestures	1st Touch screen
Group 2	2nd Touch screen	2nd Hand gestures

Participants were asked several questions. Fig. 7 show the results for each question:

1. Reliability (top-left)
2. perceived security for driving (top-right)
3. subjective distraction (2nd row left)
4. usefulness (2nd row right)
5. ease of use (3rd row left)
6. necessity (3rd row right)
7. desirability (bottom left)
8. willingness to pay for it (bottom right)

Blue bars refer to the Touch screen, purple to the Hand gestures. The first bar for each color refers to the test in the prototype car and the second to the test in the car simulator.

V. DISCUSSION AND CONCLUSIONS

Results of the survey seem to highlight a couple of interesting conclusions. Volunteers found the touch-screen (blue bars) more reliable and slightly easier to use, and hand-gestures interface (purple bars) more secure, less distracting and slightly more useful. Moreover, they found the hand-gestures system a little bit more desirable and worth buying. These opinions are the same regardless the vehicle used (1st -2nd bar for each color). A closer look to the comparison between both cars indicates that the prototype (1st bar) was perceived as more reliable, secure, useful and easy to use than the car simulator (2nd bar), for both interfacing systems. It is quite likely that these opinions were influenced by the fact that the prototype car was stopped and the drivers didn't have to pay real attention to the road, and also that the car simulator was used with the infrared illumination.

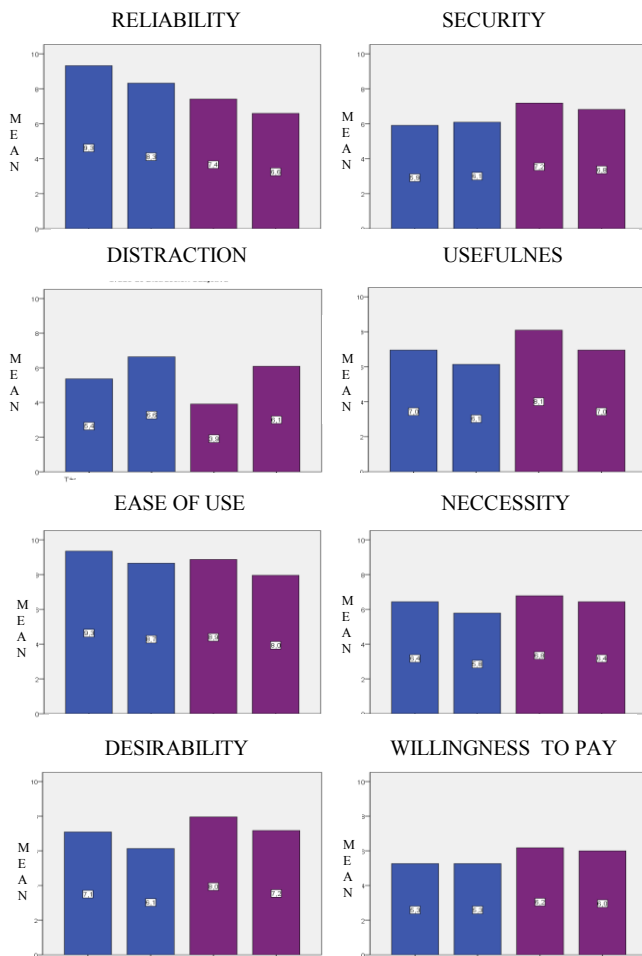


Figure 7: Results of the survey

In short, we have designed a hand-gesture-based interface system using a visible-infrared camera mounted at the cabin ceiling and reading gestures performed in the shift-stick area by the driver or the passenger. The biggest effort of the project was aimed to end up with a robust, reliable, easy-to-use and responsive system. It works in real-time at 10 frames/s, running in an Intel© dual core at 2800MHz, so it is fast enough for the task. It works quite independently on the lighting conditions, so it is robust. However, the quantitative performance and survey results reveals that there is still some work to do to improve reliability, at least enough to surpass the touch-screen based interface. Nevertheless, the volunteers perceived the hand gesture system as more secure.

REFERENCES

- [1] Klauer, S. G., Guo, F., Sudweeks, J., & Dingus, T. A. (2010). *An analysis of driver inattention using a case-crossover approach on 100-car data: Final report* (No. HS-811 334).
- [2] McEvoy, S. P., Stevenson, M. R., & Woodward, M. "The impact of driver distraction on road safety: results from a representative survey in two Australian states." *Injury prevention*, 12(4), 242-247.2006
- [3] Pickering, C.A., "The search for a safer driver interface: a review of gesture recognition human machine interface," *Computing & Control Engineering Journal*, vol.16, no.1, pp.34,40, Feb.-March 2005
- [4] Rümelin, S. & Butz, A. "How to make large touch screens usable while driving." *In Proceedings of the ACM 5th International*

Conference on Automotive User Interfaces and Interactive Vehicular Applications (pp. 48-55). October, 2013.

- [5] Geiger. "BerLührungslose Bedienung von Infotainment-Systemen im Fahrzeug". PhD thesis, TU MLunchen, 2003
- [6] Pflieger, B., Schneegass, S., & Schmidt, A. "Multimodal interaction in the car: combining speech and gestures on the steering wheel." *In Proceedings of the ACM 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. pp. 155-162. October 2012.
- [7] Askar, S., Kondratyuk, Y., Elazouzi, K., Kauff, P., & Schreer, O. "Vision-based skin-colour segmentation of moving hands for real-time applications." *In IET Visual Media Production, 2004.(CVMP). 1st European Conference on*. pp. 79-85.March, 2004
- [8] Zhu, X., Yang, J., & Waibel, A. "Segmenting hands of arbitrary color." *In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. pp. 446-453, 2000
- [9] Kolsch, M., & Turk, M. "Analysis of rotational robustness of hand detection with a viola-jones detector." *In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 3, pp. 107-110.2004
- [10] Herrmann, E., Makrushin, A., Dittmann, J., Vielhauer, C., Langnickel, M., & Kraetzer, C. "Hand-movement-based in-vehicle driver/front-seat passenger discrimination for centre console controls." *Image Processing: Algorithms and Systems VIII*. Proceedings of the SPIE, Volume 7532, article id. 75320U, 9 pp. (2010).
- [11] Althoff, F., Lindl, R., Walchshausl, L., & Hoch, S. "Robust multimodal hand-and head gesture recognition for controlling automotive infotainment systems." *VDI BERICHTE, 1919*, 187. BMW Group Research and Technology Hanauerstr. 46, 80992 Munich, Germany 2005.
- [12] Cheng, S. Y., & Trivedi, M. M. "Vision-based infotainment user determination by hand recognition for driver assistance." *Intelligent Transportation Systems, IEEE Transactions on*,11(3), 759-764.2010
- [13] Cheng, S. Y., Park, S. & Trivedi, M. M. "Multi-spectral and multi-perspective video arrays for driver body tracking and activity analysis." *CVIU, 106(2)*, 245-257. 2007
- [14] Mittal, A., Zisserman, A., & Torr, P. "Hand detection using multiple proposals".In *BMVC 2011*
- [15] Van den Bergh, M., & Van Gool, L. "Combining RGB and ToF cameras for real-time 3D hand gesture interaction." *In Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. pp. 66-72. January 2011.
- [16] Kurakin, A., Zhang, Z., & Liu, Z. "A real time system for dynamic hand gesture recognition with a depth sensor." *In Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. pp. 1975-1979. 2012
- [17] Akyol S., Canzler U., Bengler K., Hahn W. "Gesture Control for Use in Automobiles" in *Proceedings of the IAPR MVA 2000 Workshop on Machine Vision Applications*, pp. 349-352, November 28-30, 2000, Tokyo
- [18] Ohn-Bar, E.; Trivedi, M.M., "In-vehicle hand activity recognition using integration of regions," *Intelligent Vehicles Symposium (IV), 2013 IEEE*, vol., no., pp.1034,1039, 23-26 June 2013
- [19] Ohn-Bar, E., & Trivedi, M. M. "The Power is in Your Hands: 3D Analysis of Hand Gestures in Naturalistic Video."IEEE Conference on Computer Vision and Pattern Recognition Workshops. Pp 912-917, June 2013.
- [20] OpenCV Documentation. <http://www.opencv.org>
- [21] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, June 1986
- [22] Parada-Loira, F.; Alba-Castro, J.L., "Local Contour Patterns for fast traffic sign detection," *Intelligent Vehicles Symposium (IV), 2010 IEEE*, vol., no., pp.1,6, 21-24 June 2010
- [23] Zivkovic, Z. "Improved adaptive Gaussian mixture model for background subtraction". *In 17th International Conference on Pattern Recognition (ICPR 2014)*, Vol. 2, pp. 28-31. 2014.
- [24] J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Proc. 13th British Machine Vision Conf.*, pp. 384-393, 2002.