Universidad Carlos III de Madrid

TESIS DOCTORAL

Monitoring the Driver's Activity Using 3D Information

Autor:

Gustavo Adolfo Peláez Coronado

Directores:

José María Armingol Moreno

Arturo de la Escalera Hueso

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Leganés, 05/2015

TESIS DOCTORAL

Monitoring the Driver's Activity Using 3D Information

Autor: **Gustavo Adolfo Peláez Coronado**

Directores: **José María Armingol Moreno**

Arturo de la Escalera Hueso

Firma del Tribunal Calificador:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

This thesis is submitted to the Departamento de Ingeniería de Sistemas y Automática of the Escuela Politécnica Superior of the Universidad Carlos III de Madrid, for the degree of Doctor of Philosophy. This thesis is entirely my own work, and, except where otherwise indicated, describes my own research.

Dedicado a quienes me apoyaron durante
esta tarea. Vosotros sabéis quienes sois.

Dedicated to those who supported me
during this task. You know who you are.

“Levántame y arrójame donde quieras: Soy indiferente a ello. Pues allí tendré el espíritu que es en mi propicio; que está complacido y totalmente satisfecho siempre a disposición junto con sus acciones particulares, que son adecuadas y acordes a su constitución propia”-Marco Aurelio

“Take me and throw me where thou wilt: I am indifferent. For there also I shall have that spirit which is within me propitious; that is well pleased and fully contented both in that constant disposition, and with those particular actions, which to its own proper constitution are suitable and agreeable”–Marcus Aurelius

ACKNOLEDGEMENTS

If you were there when I was happy or thoughtful. Proud or disappointed. Energized or exhausted, I thank you sincerely for the time and patience you gave me. Either in the countryside, the laboratory, the Irish pub, in my home country, abroad or wherever we met, you stood next to me during this times. With advices full of wisdom, experience and if necessary, discipline, you helped me not to walk forward but to jump and reach my milestone. It may not seem as much but sometimes even the silence you gave me was just what I needed to meditate on an idea for my algorithm or to fix my attitude regarding a particular challenge. Because after a disappointing day with failed tests, wrong approaches, terrible optimizations and going back home, you were there to make me smile maybe for the first time during the day. And for that, I´m sincerely thankful because you can’t deny the fact that you cannot buy it in any supermarket.

To summarize, thank you all who made part of this journey.

RESUMEN

La supervisión del conductor es crucial en los sistemas de asistencia a la conducción. Resulta importante monitorizarle para entender sus necesidades, patrones de movimiento y comportamiento bajo determinadas circunstancias. La disponibilidad de una herramienta precisa que supervise el comportamiento del conductor permite que varios objetivos sean alcanzados como la detección de somnolencia (analizando los movimientos de la cabeza y parpadeo) y distracción (estimando hacia donde está mirando por medio del estudio de la posición tanto de la cabeza como de los ojos). En ambos casos, una vez detectado el mal comportamiento, se podría activar una alarma del tipo adecuado según la situación que le corresponde con el objetivo de corregir su comportamiento del conductor

Esta aplicación se distingue de otros sistemas avanzados de asistencia la conducción debido al hecho de que está orientada al análisis interior del vehículo en lugar del exterior. Es importante notar que las aplicaciones de supervisión interna son tan importantes como las del exterior debido a que si el conductor se duerme, un sistema de detección de peatones o vehículos sólo podrá hacer ciertas maniobras para evitar un accidente. Todo esto bajo las condiciones idóneas y circunstancias predeterminadas. Esta aplicación tiene el potencial para estimar si quien conduce está mirando hacia una zona específica que otra aplicación que detecta objetos, animales y peatones ha remarcado como importante.

Aunque en el mercado existen tecnologías disponibles capaces de supervisar al conductor, estas tienen un coste prohibitivo para cierto grupo de clientela debido a que no es un producto popular (comparado con otros dispositivos para el hogar o de entretenimiento) ni existe un mercado con alta oferta y demanda de dichos dispositivos. Muchas de estas tecnologías requieren de dispositivos externos e invasivos (colocarle al conductor uno o más sensores en el cuerpo) que podrían interferir con la naturaleza de los movimientos propios de la conducción bajo condiciones sin supervisar. Las aplicaciones actuales basadas en visión por computador toman ventaja de los últimos desarrollos de la tecnología informática y el incremento en poder computacional para crear aplicaciones que se ajustan al criterio de un método no invasivo para aplicarlo a la supervisión del conductor. Tecnologías como cámaras estéreo y del tipo “tiempo de vuelo” son capaces de sobrepasar algunas de las dificultades relacionadas a las aplicaciones de visión por computador como condiciones extremas de iluminación (diurna y nocturna),

saturación de los sensores de color y la falta de información de profundidad. Es cierto que la combinación y fusión de sensores puede resolver este problema por medio de múltiples escaneos de diferentes zonas o combinando la información obtenida de diversos dispositivos pero esto requeriría un paso adicional de calibración, posicionamiento e involucra un factor de dependencia de la aplicación hacia no uno sino los múltiples sensores involucrados ya que si uno de ellos falla, los resultados podrían no ser correctos. Recientemente han aparecido en el mercado de los videojuego algunos sensores, como es el caso de la barra de sensores Kinect de Microsoft, dispositivo de bajo coste, que ofrece información 3D junto con otras características adicionales y sin la necesidad de sistemas complejos de sistemas manufacturados que pueden fallar como se ha mencionado anteriormente.

La solución propuesta en esta tesis supervisa al conductor por medio del uso de información diversa del sensor Kinect (información de profundidad, imágenes de color en espectro visible y en espectro infrarrojo). La fusión de información de diversas fuentes permite el uso de algoritmos en 2D y 3D con el objetivo de proveer una detección facial confiable, estimación de postura precisa y detección de características faciales como los ojos y la nariz. El sistema comparará, con una velocidad promedio superior a 10Hz, la captura inicial de la cara con el resto de las imágenes de video, la comparación la hará por medio de un algoritmo iterativo previamente configurado comprometido con el balance entre velocidad y precisión. Con tal de determinar la fiabilidad y precisión del sistema propuesto, diversas pruebas fueron realizadas para el algoritmo de estimación de postura de la cabeza con una unidad de medidas inerciales (IMU por sus siglas en inglés) situada en la parte trasera de la cabeza de los sujetos que participaron en los ensayos. Las medidas inerciales provistas por la IMU fueron usadas como punto de referencia para las pruebas de los tres grados de libertad de movimiento. Finalmente, los resultados de las pruebas fueron comparados con aquellos disponibles en la literatura actual para comprobar el rendimiento del algoritmo aquí presentado.

Estimar la orientación de la cabeza es la función principal de esta propuesta ya que es la que más aporta información para la estimación del comportamiento del conductor. Sea para tener una primera estimación si ve hacia el frente o si presenta señales de fatiga al cabecear hacia abajo. Acompañando a esta herramienta, está el análisis de la imagen a color que se encargará del estudio de los ojos. A partir de dicho estudio, se podrá estimar hacia donde está viendo el conductor según la posición de la pupila. La orientación de la mirada ayudaría, junto con la orientación de la cabeza, a saber hacia dónde ve el conductor. La estimación de la orientación de la mirada es una herramienta de soporte que complementa la orientación de la cabeza. Otra forma de determinar una situación de riesgo es con el análisis de la apertura de los ojos. A través del estudio del patrón de parpadeo en el conductor durante un determinado tiempo se puede estimar si se encuentra cansado. De ser así, el conductor aumenta las posibilidades de causar un accidente debido a la somnolencia. La parte de la solución que se encarga de resolver este problema analizará un ojo del conductor para estimar si se encuentra cerrado o abierto de acuerdo al análisis de regiones de interés en la imagen. Una vez determinado el estado del ojo, se procederá a hacer un análisis durante un determinado tiempo para saber si el ojo ha estado mayormente cerrado o abierto y estimar de forma más acertada si se está quedando dormido o no. Estos 2 módulos, el detector de somnolencia y el análisis de la mirada complementarán la estimación de la orientación de la cabeza con el objetivo de brindar mayor certeza acerca del estado del conductor y, de ser posible, prevenir un accidente debido a malos comportamientos.

Es importante mencionar que el sensor Kinect está construido específicamente para el uso dentro de una habitación y conectado a una videoconsola, no para el exterior. Por lo tanto, es inevitable que algunas limitaciones salgan a luz cuando se realice la monitorización bajo condiciones reales de conducción. Dichos problemas serán mencionados en esta propuesta. Sin embargo, el algoritmo presentado es generalizable a cualquier sensor basado en nubes de puntos (cámaras estéreo, cámaras del tipo “time of flight”, escáneres láseres etc...); más caros pero menos sensibles a estos inconvenientes previamente descritos. Se mencionan también trabajos futuros al final con el objetivo de enseñar la escalabilidad de esta propuesta.

ABSTRACT

Driver supervision is crucial in safety systems for the driver. It is important to monitor the driver to understand his necessities, patterns of movements and behaviour under determined circumstances. The availability of an accurate tool to supervise the driver’s behaviour allows multiple objectives to be achieved such as the detection of drowsiness (analysing the head movements and blinking pattern) and distraction (estimating where the driver is looking by studying the head and eyes position). Once the misbehaviour is detected in both cases an alarm, of the correct type according to the situation, could be triggered to correct the driver’s behaviour.

This application distinguishes itself form other driving assistance systems due to the fact that it is oriented to analyse the inside of the vehicle instead of the outside. It is important to notice that inside supervising applications are as important as the outside supervising applications because if the driver falls asleep, a pedestrian detection algorithm can do only limited actions to prevent the accident. All this under the best and predetermined circumstances. The application has the potential to be used to estimate if the driver is looking at certain area where another application detected that an obstacle is present (inert object, animal or pedestrian).

Although the market has already available technologies, able to provide automatic driver monitoring, the associated cost of the sensors to accomplish this task is very high as it is not a popular product (compared to other home or entertaining devices) nor there is a market with a high demand and supply for this sensors. Many of these technologies require external and invasive devices (attach one or a set of sensors to the body) which may interfere the driving movements proper of the nature of the driver under no supervised conditions. Current applications based on computer vision take advantage of the latest development of information technologies and the increase in computational power to create applications that fit to the criteria of a non-invasive method for driving monitoring application. Technologies such as stereo and time of flight cameras are able to overcome some of the difficulties related to computer vision applications such as extreme lighting conditions (too dark or too bright) saturation of the colour sensors and lack of depth information. It is true that the combination of different sensors can overcome this problems by performing multiple scans from different areas or by combining the information obtained from different devices but this requires an additional step of calibration, positioning and it involves a dependability factor of the application on not one but

as many sensors included in the task to perform the supervision because if one of them fails, the results may not be correct. Some of the recent gaming sensors available in the market, such as the Kinect sensor bar form Microsoft, are providing a new set of previously-expensive sensors embedded in a low cost device, thus providing 3D information together with some additional features and without the need for complex sets of handcrafted system that can fail as previously mentioned.

The proposed solution in this thesis monitors the driver by using the different data from the Kinect sensor (depth information, infrared and colour image). The fusion of the information from the different sources allows the usage of 2D and 3D algorithms in order to provide a reliable face detection, accurate pose estimation and trustable detection of facial features such as the eyes and nose. The system will compare, with an average speed over 10Hz, the initial face capture with the next frames, it will compare by an iterative algorithm previously configured with the compromise of accuracy and speed. In order to determine the reliability and accuracy of the proposed system, several tests were performed for the head-pose orientation algorithm with an Inertial Measurement Unit (IMU) attached to the back of the head of the collaborative subjects. The inertial measurements provided by the IMU were used as a ground truth for three degrees of freedom (3DoF) tests (yaw, pitch and roll). Finally, the tests results were compared with those available in current literature to check the performance of the algorithm presented.

Estimating the head orientation is the main function of this proposal as it is the one that delivers more information to estimate the behaviour of the driver. Whether it is to have a first estimation if the driver is looking to the front or if it is presenting signs of fatigue when nodding. Supporting this tool, is another that is in charge of the analysis of the colour image that will deal with the study of the eyes of the driver. From this study, it will be possible to estimate where the driver is looking at by estimating the gaze orientation through the position of the pupil. The gaze orientation would help, along with the head orientation, to have a more accurate guess regarding where the driver is looking. The gaze orientation is then a support tool that complements the head orientation. Another way to estimate a hazardous situation is with the analysis of the opening of the eyes. It can be estimated if the driver is tired through the study of the driver's blinking pattern during a determined time. If it is so, the driver increases the chance to cause an accident due to drowsiness. The part of the whole solution that deals with solving this problem will analyse one eye of the driver to estimate if it is closed or open according to the analysis of dark regions in the image. Once the state of the eye is determined, an analysis during a determined period of time will be done in order to know if the eye was most of the time closed or open and thus estimate in a more accurate way if the driver is falling asleep or not. This 2 modules, drowsiness detector and gaze estimator, will complement the estimation of the head orientation with the goal of getting more certainty regarding the driver's status and, when possible, to prevent an accident due to misbehaviours.

It is worth to mention that the Kinect sensor is built specifically for indoor use and connected to a video console, not for the outside. Therefore, it is inevitable that some limitations arise when performing monitoring under real driving conditions. They will be discussed in this proposal. However, the algorithm presented can be used with any point-cloud based sensor (stereo cameras, time of flight cameras, laser scanners etc...); more expensive, but less sensitive compared to the former. Future works are described at the end in order to show the scalability of this proposal.

INDEX

INDEX OF FIGURES

INDEX OF EQUATIONS

CHAPTER 1. INTRODUCTION

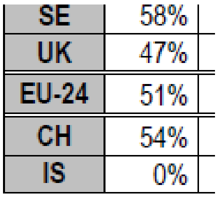
Until the year 1938, with the introduction of the “Beetle” model by the manufacturer Volkswagen, the vehicle industry was not as big as it is today mostly due to the high prices, lack of reliability and complexity associated to the first commercial vehicles. It was so complicated that it was not unusual that as a car owner would have to hire someone just for the driving and the maintenance as it required a high amount of knowledge and skills. Because motorized vehicles were so uncommon, the infrastructure associated with it (roads, traffic lights etc.) was not well developed. Nevertheless, accidents happened from time to time but it was considered as a normal situation when buying a motorized vehicle. As time went on and the massive manufacturing of vehicles took place, the traffic of cars and higher speed limit, due to more powerful engines, made accidents more frequent.

Today, the maximum speed limits have exceeded those from the early 20th century and the infrastructures are outdated but the accidents still prevail although many improvements have been made in attempt to prevent this. As an example, the most common and fundamental method to reduce the damage received by the driver and passengers is the safety belt. Developed by Swedish industry Volvo it has saved many lives since its implementation and is an example of a method to increase the safety of driving. Of course this method, although reduces drastically the chances of severe injuries to the passengers and drivers, cannot deal with the rest of the problems by itself thus other techniques such as anti-block systems for the breaks, airbags and on board computers take an important role to complement each other in order to make the driving experience as safe as possible. Still today, the numbers show that more and better efforts need to be done as the report from some institutions demonstrate that accidents there are still a large amount of accidents and some of them lethal.

The European Road Safety Observatory (ESRO) gathered data from the CARE database at the Directorate General for Energy and Transport of the European Commission [1]. The table below shows the data gathered by ESRO regarding the distribution of casualties by type of road user inside and outside urban areas in 2010 by country and the average for the EU.

TABLE I:

DISTRIBUTION OF CASUALTIES BY AREA AND COUNTRY



SE	58%
UK	47%
EU-24	51%
CH	54%
IS	0%

Also from ESRO, the following graph (Fig. 1-1) shows the number of casualties per month inside and outside urban areas. Notice that the amount of accidents during summer is higher outside urban areas but is lower during the other months. This can be due to the increase of travelling during holidays which will also increase the traffic flow outside the urban areas. During the month of July, the accidents are at their maximum. Compared to June, the increase is almost linear in case of the urban area but for the accidents that happen outside the urban areas the increase is more significant than in the previous 4 months. Again, this could be due to the travelling for holidays by people from the city.

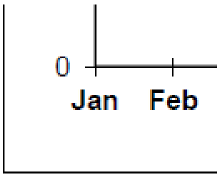


Figure 1-1: Accidents per month

One of the causes for car accidents is the fatigue and drowsiness. If a driver falls asleep for 4 seconds driving at 100km/h, the car will have travelled more than 110 meters without supervision. Lack or bad quality of sleep is one of the main factors although a small part of population (around 4%) has to deal with Obstructive Sleep Apnoea Syndrome (OSAS) [2]. OSAS does not present a spontaneous regression or improvement and the medications treat only the symptoms. A person who has been awake 17 hours has the same risk to crash as a person with a BAC (Blood alcohol content) of 0.05g/100ml and those who have been awake for 24 hours perform similarly to a person with BAC 0.1g/100ml [3]. According to the European commission, police reports show that the percentage of fatigue related crashes is between 1 and 4%. But additional questionnaire studies show that the estimated percentage of sleep related crashes varies in range of 10 to 25% higher than the police reports [4]. The exact percentage varies from study to study and from region to region. Horner and Reyner in [5] determined that 20% of the car accidents were related with sleep. In Germany, a crash study established the percentage of crashes related with fatigue at 24% [6].

The report “New Standards and Guidelines for Drivers with Obstructive Sleep Apnoea syndrome” [7] gives a clear explanation regarding this syndrome. They explain the symptoms, the population group that is most affected and how to deal with this. One of their main conclusions is the inclusion of the OSAS among the medical issues regarding driving license issues. In the section regarding the car accidents, they explain that 10% of all single vehicle accidents are related to fatigue. In the report it can be found also the warning signs for drowsy driving and they are:

- Yawning and blinking
- Impression of driving automatically: difficulty remembering the past few miles driven
- Missing exits
- Difficulty in maintaining a steady road position – drifting from one's lane –hitting a rumble strip
- Difficulty in maintaining a constant speed
- Nodding

The car manufacturer Volvo [8] explains that some of the accidents caused by distraction can be due to not paying attention to the road or due to drowsiness. The chance of a drowsiness accident to happen is 13 times higher between 4 and 5 in the morning. The indicators according to their investigations of a typical drowsiness-caused accident are:

- No skid marks on the road and/or hard or verge.
- No braking signs.
- The angle between the road and the trajectory is acute.
- Usually occurs at the final segment of a straight before the beginning of the next curve.
- Non-urban roads.

It is worth mentioning that not only the driver and the passengers can be victims but also those in the surroundings as the driver could lose control due to many factors on the car and drive away from the road hitting pedestrians in the vicinity. Therefore, every application that can help to reduce the casualties and thus the fatal victims is adequate in order to make the driving experience safer.

According to the document "Sleepiness at the wheel, white paper" [9] sleepiness is responsible for 20 to 25% of the accidents occurring on European roads. This has a huge economic impact because sometimes the vehicles crash against bridges, or critical structures like electric pylons and houses. They claim that in 2011, in the United States of America, the cost associated to motor vehicle crashes was of $230 billion. In the European Union, the direct and indirect costs related to car accidents were $160 billion in 2007.

Drivers can fall asleep because of OSAS but also due to the effect of psychoactive medications such as anxiolytics, anti-allergic drugs and antidepressants. Part of the DRUID (Driving Under the Influence of Drugs, alcohol and medicines) involved a survey conducted in 4 countries in Europe in order to get information of driving while under the influence of drugs. The selected group of drivers were between 18 and 75 years old and were using any psychotropic medicine. The figure 1-2 shows the result of the questionnaires. The amount of drivers interviewed for each country were: Belgium 136, Germany 146, The Netherlands 136 and Spain 215.

Figure 1-2: Results of the survey reported in the white paper.

The proposed solution in this thesis aims to prevent accidents caused by drowsiness (Fig. 1-3, taken from [9]) due to sleep deprivation, apnoea or drug side effects. By supervising the driver with modern and reliable algorithms that manipulate the 2D-3D data altogether in order to obtain not only an accurate head pose orientation but also a gaze orientation. This two factors combined can help to understand when the driver is falling asleep or being distracted so that an accident can be prevented. Also, the results can be used for further experiments and more complex layers such as the analysis of the behaviour under particular circumstances that may vary from urban environments (with pedestrian and near obstacles) to motorways (less dynamic and far objects but at higher speeds).

Figure 1-3: Drowsiness related accident

1.1 Driving Assistance Systems

There is a variety of driving assistance systems. Some of them are oriented to analyse the surroundings (pedestrians and obstacles) while others the inside of the vehicle (driver and its surrounding). The systems that analyse the outside will have the goal to warn the driver of a particular situation, whether it is a pedestrian crossing that was detected with a stereo set of cameras or the road sign in the nearby that is indicating a speed limit for the road. Depending on the situation, the sensors used to achieve the task will change although the most used are lasers, radars and cameras. Some of the applications need to use one sensor only to solve a determined problem; there are others, for example that combine two different sensors like one-layer lasers with colour cameras and then perform a fusion of the information to optimize the performance or complement the results of one sensor. The applications that supervise the inside of the vehicle use many different sensors too. Again, depending on the application to be performed the hardware to be used will be different. For example, when analysing the behaviour of the driver under different circumstances (different types of passengers) the use of microphones could deliver more useful information about the intensity of the noise. Because the analysis will be performed inside the vehicle, the nature of the sensors will vary as the area to be studied is delimited and smaller but for some applications like the supervision of the driver the hardware to be used will have certain requirements such as higher resolution and an adequate minimum distance (laser-based sensors). Combining different sources of information to supervise the driver can be performed too, it is not exclusive to the outside analysis.

By supervising the driver with a device that can predict drowsiness, distractions or another misbehaviour while driving according to the head movement and other methods, the chances to cause an accident can be reduced. The device should be able to determine the head pose accurate and fast in order to work with other systems on board to prevent accidents and make a more robust driving assistance system (ADAS). An example of ADAS can be seen in Fig. 1-4 where different cameras are assisting to detect obstacles but also supervising the driver.

Driving behaviour analysis can be performed by studying the results obtained from the application. As an example, profiles can be created according to the driving environment and situations (urban or motorway driving are different).

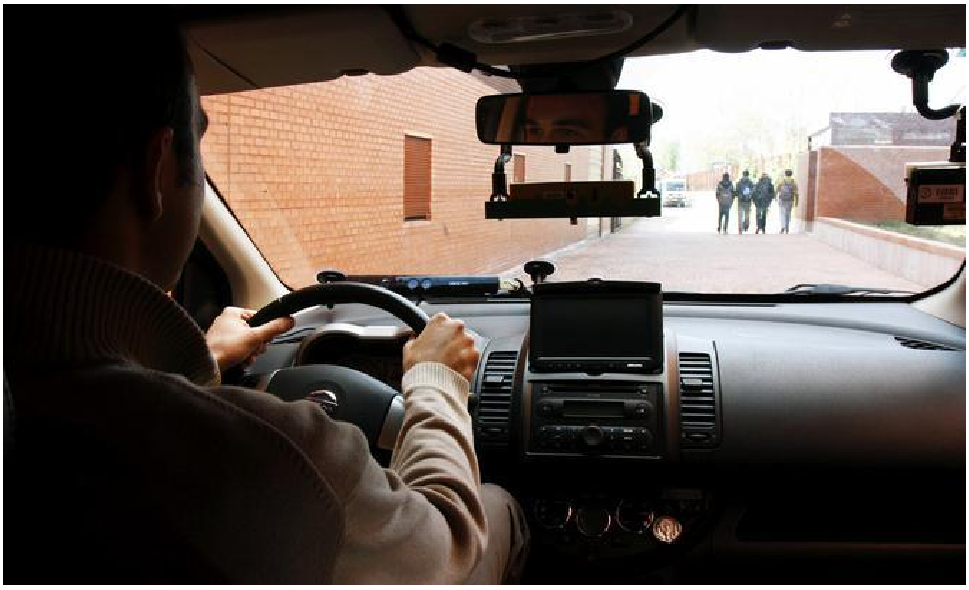


Figure 1-4: Devices on board to assist the driver while driving

If the microphones from the Kinect are used, loud and noisy environments can be determined and the driving behaviour could be monitored to establish if there is a correlation between its behaviour and the noise inside. The sensor should also be able to deal with some of the common problems of computer vision algorithms in the outside such as saturation of sensors, lack of visibility or problems with one unique source of information that a fusion of information and technologies could solve. By combining the 2D data stream from the colour camera with the 3D data obtained from the projection of the infrared dot pattern, the lack of data to do a head pose estimation is diminished due to the fact that the 3D estimation is independent of the lighting conditions that could affect the colour camera. It is then an opportunity to combine both algorithms (2D and 3D) into one application that works using the latest sensors embedded all in one device that allows to obtain a 3D head pose estimation with the support of reliable 2D algorithms. Overall, this low cost sensor allows the implementation of data fusion in ADAS at a price that other sensors could reach and the chance to combine different algorithms into one fast and reliable solution that will warn the driver properly.

1.2 Proposal

It is proposed in this thesis a solution that will supervise the driver and report if the driver is distracted or falling asleep in a fast and accurate way. It will combine both 2D and 3D algorithms to achieve an estimation of the 3 rotation angles (pitch, roll and yaw). In comparison with other solutions, this one allows the head pose estimation with infrared technology which allows the estimation with a 3D structure and therefore the rotation angles can be estimated instead of using approximations with the 2D information from a normal camera. Although oriented to work with the Kinect under the defined environment, it is not restricted to this conditions as the algorithm will work with any other sensor that can deliver a 3D cloud of points. The result will be an estimation of the head pose and the gaze orientation of the driver in a fast and accurate way. This results can be obtained in numeric format so that they can be used for other applications or in a visual representation that is more explicit and clearer than a numeric display in a console format. The proposal aims to reduce the chance of an accident due to drowsiness or misbehaviours from the driver. This in complement with additional tools to supervise the surroundings can help to avoid casualties while driving.

This solution was tested with more than 15 subjects of different age and sex under different lighting conditions and providing good results when contrasted with the ground truth that was the IMU. The obtained results can be used for drowsiness detection but also for other misbehaviours like distractions (looking away from the road). This is possible thanks to the accurate measurements obtained and by doing a temporal analysis, i.e. counting how long is the head positioned in a determined orientation or where the driver is looking at a determined moment. By doing a study on the time a driver is distracted or showing fatigue signals, a proper profile can be done for other solutions that study the human factors when driving. This application is immune to small changes and occlusions of the driver’s face. Whether yawning, talking, touching the nose or another small disruption, the solution can cope with it with a filtering process to remove noisy results using a temporary filter later described. This was implemented so that the application would be more robust to the common situations that happen while driving that cannot be equally elaborated at the laboratory due to harsh lighting conditions. An example of this is shown in Fig. 1-5 where the sensor bar was being tested in a dark environment.

Figure 1-5: Example of the location of the sensor bar

The fusion of 2D and 3D data can be complex and full of previous steps for calibration that can make the testing phase more complicated than what it should be. Therefore, the use of a device that can solve this problems at once brings a new approach to the data fusion by removing that calibration procedure that had to be done every time. And this can be done with a device that is affordable and easy to obtain in the current market thus opening the opportunity to test 2D and 3D algorithms with ease compared to any other approach.

This thesis will start with the description of the latest solutions that are somehow related to the aim of this proposal. The variety of solutions for supervising and monitoring the driver is big so as the other ADAS that supervise the surrounding environment of the vehicle for the sake of avoiding collisions and preventing accidents. All this applications are described and divided in two main categories so that the analysis can be clearer. Some definitions are also stated such as data fusion, driving assistance systems and the importance of on board devices and the importance of the placement for a proper usage.

Following the previous chapter is the general description of the system where the big picture is described. Both hardware and software used in this application are explained. The device used, the Kinect sensor bar is explained in this section and the data structure that is obtained from it that is the corner stone of this thesis as it allows to manipulate the 2D and 3D data at the same time without any calibration or synchronisation required. It also includes additional peripherals used during the experiment and algorithms that were used during different stages of this solution.

The next chapter explain with a deeper analysis all that is related to the spatial restrictions and optimizations. The assumptions taken into account to make the processing faster but still accurate and reliable are shown. Also described in detail is the estimation of the head pose and the importance of the correct parametrization. In the same way is the gaze estimation chapter that will focus on the interpretation of the position of the pupil in the eye. By taking into account several stages of image processing, the gaze estimator can establish if the driver is looking to the left, centre or right.

After the previous chapter, the next one will explain the results of both the gaze and head pose estimators. This chapter shows how to interpret the obtained data from the algorithms so that it can be known if the driver is asleep or distracted. Also if the driver is looking at a specific region during a determined moment of time. The last section will explain the ground truth used to confirm that the results were correct. Additional approaches that did not deliver good results are also described in this part of the thesis.

The last chapter is the conclusion of the thesis. It will explain the contributions achieved by doing this thesis. From affordable data fusion to a robust head pose estimation, the milestones reached are here described.

CHAPTER 2. STATE OF THE ART

2.1 Introduction

The ADAS have been changing significantly since the early days of its implementation, as stated before. Although before the ADAS, there were devices that improved the safety of driving like the first commercial seat belt. This improvements have progressed until today with the latest autonomous vehicles. The ADAS provide assistance in many ways and make the driving experience safer. Behind this, both laboratories and private companies have been working on new methods and improvements for the ADAS. And it is the collaboration between the laboratories and industries that allows a proper development for a correct ADAS due to the fact that the industry can bring actual data and experience from the real world and the laboratories can develop the solution according to this information. Different universities have experimental platforms or vehicles where to test the proposed solutions and they vary from size and main goal to equipment used and approaches to solve a specific or a set of problems. ADAS can vary from a simple indicator in a screen to indicate an event or it can take full control of the vehicle and make it autonomous. To do its job, ADAS can perform many tasks that go from the analysis of the surroundings of the vehicle to supervise the driver's behaviour at any moment. Because the goals and conditions are different, the sensors and techniques used to achieve the objective will vary. It will be described later in this chapter what is understood as an inside monitoring with examples; outside monitoring will be defined and explained too.

In this chapter, the definition of driving assistance systems and the different types of monitoring is given. Following the definitions are certain examples about each of the concepts here explained. The examples will show what is currently being done in the area of driving assistance systems, monitoring the driver and the different types of monitoring. This is so that there can be a comparison between this application and what the rest are doing.

2.2 Data Fusion

Data fusion is considered in this thesis as the sensor can return, through a set of sensors, information from the surroundings. Whether it is colour information or infrared through the optical devices or audio from the array of microphones. In this section, the definition of data fusion is explained in order

2.2.1 Definition

It was established during the decade of the 1980 in [10] and states that it is the process of dealing with the combination and proper association of information from one or many sensors in order to obtain a refined result of the algorithm which has the characteristic of performing constant adjustments to its estimations, results and temporal answers but also evaluating if the need of another provider of information is required.

Also, the Joint Directors of Laboratories (JDL) defines it in [11] as a “multi-level, multifaceted process handling the automatic detection, association, correlation, estimation, and combination of data and information from several sources”.

Later in [10] it can be found that the objective of the data fusion is to correct and complement the results that one sensor or data source can’t solve with the information and processing obtained from another data source. Finally, Steinerg and Bowman [12] say that it is the process to combine either data or information in order to predict estimates or states.

From the different definitions previously mention, it can be interpreted that the data fusion is the process of taking one or more information sources, analyse them in different ways and obtain a more reliable result compared to those that would be obtained through the use of only one information source. The information will be obtained from sensors of different nature that deliver different type of data. If necessary, a first step of calibration or association between the many information sources will be performed so that later the flaws or lack of performance from one sensor could be improved by the other information sources.

2.2.2 **Examples**

There are different applications that use the fusion of information for a variety of objectives to be achieved. Garcia F. et al. in [13] explain how to use the information from a colour camera and a one-layer laser to create a robust application to detect obstacles. Fig. 2-1 shows the obtained results when detecting obstacles using data fusion under different environments. This application is oriented for intelligent vehicles and can detect both pedestrians and vehicles by analysing the pattern obtained from the laser and the information of the colour camera. Different levels of fusion levels are explained. The application runs in real time as expected for an ADAS.

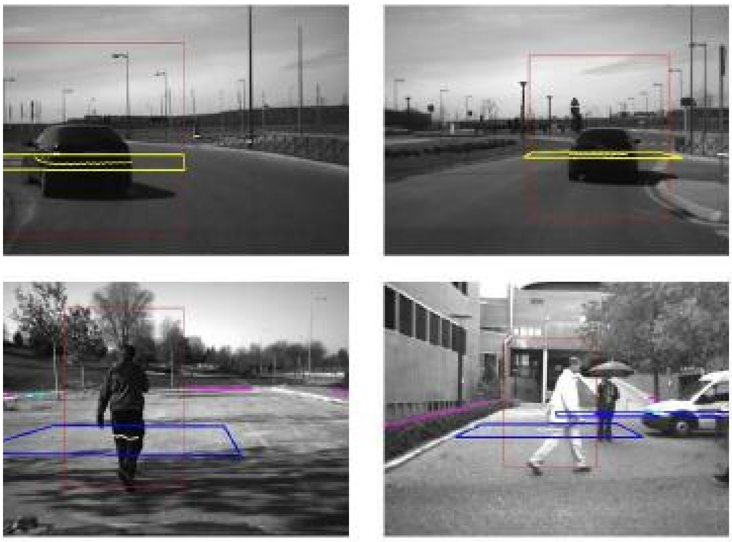


Figure 2-1: Detection of vehicles and pedestrians

Thomaidis, G. et al. show in [14] an analysis on the problems that may occur when the data fusion is performed; this also is oriented to intelligent transportations systems and vehicles. Specifically, they examine the problem of fusion for tracking targets and the detection of the surroundings including the road. Not only is the discussion done but also some solutions that could work at different levels of integration. The proposed solutions are shown finally with some tests.

In [15], a solution is proposed for overtaking vehicles using data fusion from a radar “Stop&Go” and a colour camera. They take into consideration the unprocessed information obtained from the radar and combines it with computer vision algorithms like optical flow. The results shown by the authors demonstrate that the algorithm has a high rate of success.

Finally, in [16], it is proposed the usage of automated vehicles for the construction environment by using pre-set paths as a reference for the vehicles. They combine three main parts that are the tracking vehicle system, the multi-sensor systems and the control of the vehicle. All this systems also are composed of the fusion of different sensors including ultra-sonic for proximity and safety. Another combination is done between the odometry from encoders and the IMU sensor in order to improve the position of the vehicle. The figure below, (Fig. 2-2) shows the configuration of their proposed system displaying a 2D grid from the data obtained from the ultra-sonic sensor.

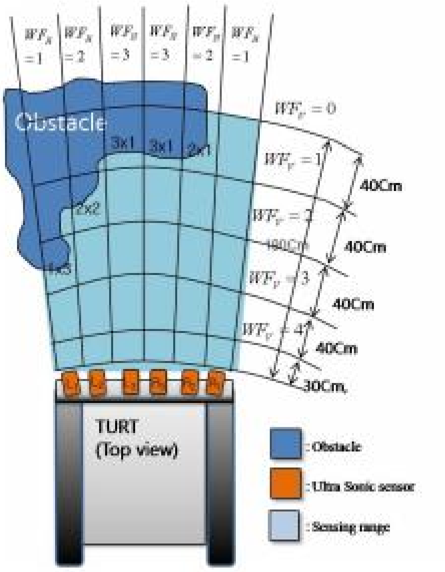


Figure 2-2: Configuration of the obstacle avoidance system

Although there are some other applications of data fusion, this are some of those closer to the application here described. For this thesis, the fusion happens between the combination of 2D information (colour camera) and the 3D information (depth) in order to obtain a single structure with spatial (*x, y, z*) and colour(*r, g, b*) information for each pixel that constitutes it.

2.3 Driving Assistance Systems

Driving assistance systems have changed significantly during the last 10 years as the computers have become faster, with more storage capacity, consuming less power and with smaller dimensions. Therefore what used to take a large space, such as the whole trunk of an average vehicle for 3 computers, can now store more powerful computers that execute powerful algorithms in a smaller space. Either to warn of a possible collision or by reducing the damage received by the occupants of the car during an accident, the ADAS will complement the vehicle's function and make it safer to drive. In order to explain the current state of the ADAS, two main branches are defined: inside and outside monitoring. Each of this divisions has different goals to achieve and will be explained in the next lines. Also, examples of some of the latest systems are mentioned.

2.3.1 Outside Monitoring

One of the first goals that the driving assistance systems had (with computer assistance) was the analysis of the surroundings of the vehicle. Either for automated parking or to detect obstacles on the road and adjust the route. This type of monitoring can prevent collisions with different obstacles by either warning the driver or by taking control of the car and perform the proper actions to avoid crashing. Most of the time the sensors will be located outside.

Definition

In this thesis, it is defined as the analysis of the surroundings of the vehicle with different sensors including thermal and colour cameras; one or multiple layer laser scanners; short or long proximity sensors based in ultrasound. The goals of this driving assistance systems is to scan for nearby objects of interest either for the information they present (road structures such as lanes and signs) or because they need to be avoided in order to prevent a collision (pedestrians, animals and other objects). Not only this applications can help to reduce accidents by warning the driver but in particular situations, they will take action i.e. change the direction of the vehicle, reduce the speed of the vehicle or in the case of aerial vehicles, increase the thrust if necessary.

Examples

In [17] it is proposed an information based approach for sensor resource management that is suited for vehicular applications with cooperative sensor systems. This is what they call the "principle of cooperative sensors". They apply this strategy to a pedestrian perception system and analyse the results in different critical traffic situations in comparison to other possible approaches for this problem. They use, as a way to justify the results, real world measurement data taken from an IEEE 802.11p prototype sensor at 5.9 GHz.

Krotsky et al [18] propose a multimodal trifocal framework composed of a pair of colour cameras to conform a stereo system and an infrared camera. This allows the demonstration of the significantly higher performance of detection when it is done with many information sources, in this case, colour, depth and infrared. With an experimental analysis, they show the different challenges to be solved when doing the approach of multiple sources and perspectives to identify the pedestrians. Later, a test rig is proposed consisting of two colour and two infrared cameras in order to test different approaches for pedestrian detections. They finish the article by providing an analysis of the infrared and colour features that were used to identify a detected obstacle as a pedestrian.

The industry is not totally isolated from the world of ADAS. Some have a close collaboration with technology institutes and academic world. For example, in [19] it is presented a method for understanding the surroundings using a reconstruction in 3D based on image labelling using a conditional random field. The labels can be of the semantic type i.e. car, obstacle, sidewalk etc… and it is with this labels that the other ADAS can work with such as free space estimation and obstacle avoidance. The application can do the 3D reconstruction as long as the vehicle is travelling below 63Km/h. It is also discussed the fact that modern vehicles are equipped with different cameras, its locations and types are mentioned too, for many applications and how the information that they deliver is important for the driver.Fig.2-3 shows, at the top left corner, the input camera image. At the top right the segmentation with tags and at the bottom the 3D model with textures.

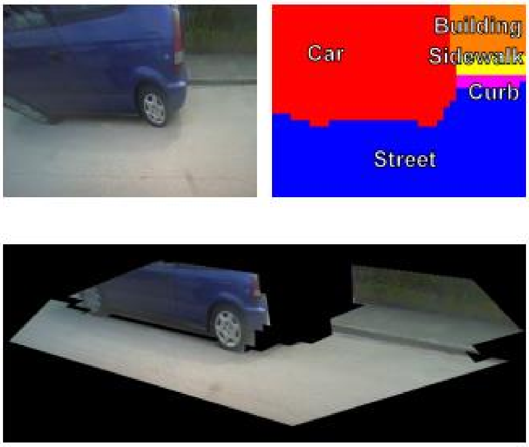


Figure 2-3: Results obtained from the proposed solution.

2.3.2 Inside Monitoring

Outside monitoring has many goals to achieve, from pedestrian and obstacle detection to context analysis like road detection and road signs detection and classification. But there is also another type of monitoring and that is the one that deals with what is happening inside the car, more specifically, with the driver. When supervising the driver, different conditions must be considered in order to develop an application that works properly. For example, the proximity of the driver to the dashboard where presumably the camera will be located; if it is too close, proximity and distance based sensors may have problems as they have a minimum distance between the object to be analysed and the sensor itself. Also the inside of the cabin is, most of the times, a closed area surrounded by crystals and different textiles or fabrics. Because of the static nature of this environment, a previous analysis of the scene may allow a fast detection of movement and the driver. But not only the driver detection and computer vision algorithms can be applied inside the vehicle. Analysis of the audio can be easily performed as there are not many interference as if it was performed in the outside where the wind would be a serious audio disrupter. The audio analysis allows the study of the driver's behaviour under calm and quite against loud and chaotic environments. In this thesis, the main goal is the application of computer vision algorithms to supervise the head pose movements and estimate whether the driver is distracted or falling asleep by detecting misbehaviours and drowsiness signs. Therefore, the positioning of the Kinect was important so that the minimum distance would not interfere with the measurements of the driver's head.

Definition

For this particular solution, the inside monitoring is defined as the analysis of the vehicle's inside with the purpose to estimate the pose of the head so that later the drowsiness can be estimated or to confirm if the driver is looking at a determined area that other applications consider as important due to the presence of a pedestrian, animal or obstacle. This supervision will be performed with computer vision algorithms that will work with the information obtained from the Kinect's sensors i.e. infrared 3D structure and RGB CMOS sensor. One of the advantages of doing the inside analysis is the fact that the object to be analysed, in this case the driver's head, will be most of the time around the same position unless extra conventional situations arise. Therefore, the computer vision algorithms can perform faster when searching for the face just to mention one example. Because of the crystals that filter some infrared radiation, the sensor in charge to obtain the 3D structure has a lower chance to saturate as if it was in the outside. Because this solution does not require to attach any sensor to the driver, it is considered a non-invasive method. This allows, through proper adjustments of the algorithms, to supervise the driver without interfering with the natural behaviour of its movements.

Examples of Non-Invasive Applications

Some experiments have been performed that can be catalogued as non-invasive when performing the inside monitoring i.e. based in computer vision. The last example is also non-invasive but it is not using computer vision algorithms. Instead, it is using sensors around where the driver makes contact with the vehicle.

In [20] they explain a real time 3D facial imaging that estimates the facial orientation and the gaze detection using 3 phase correlation image sensor i.e. the light encodes the surface normal into the amplitude and phase of the reflected light. They find the key points and curvature features of the face by using the normal vector map and intensity image. Later, with the key points found, the optical flow algorithm is used to estimate the face orientation. In comparison to other approaches, this one is done using 3D algorithms. It is remarked at the end that this is not a final application and that it is more a simple example of the future complex developments to be done. Fig 2-4 shows the schematic diagram of the proposed system from where they can obtain both dense and pixel-wise normal vector maps disregarding the surface reflectance.

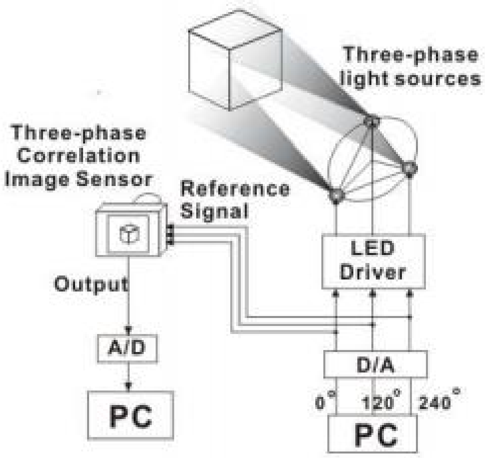

Three-phase light sources
Three-phase Correlation Image Sensor
Reference Signal
LED Driver
Output
A/D
PC
D/A
0°
120°
240°
PC

Figure 2-4: Schematic diagram of the proposed solution

Another approach is the one mentioned in [21] where a real time gaze zone estimator is presented based on the orientation of the driver's head using two rotation angles only, yaw and pitch. The method, they say, works for day and night and for drivers with glasses too. They calculate the driver's yaw rotation angle by modelling the face with an ellipsoidal model, not a cylindrical and for the calculation of the pitch they introduce new terms like the normalized mean and the standard deviation for the horizontal edge projection histogram. Vector machines and mentioned too for the gaze orientation estimation. To achieve the exact gaze position, 18 different zones are proposed. This zones correspond to different regions of interest from the driver's point of view. It is said that the error estimated for 200 000 images is under 7.

Heuer et. al describe in [22] a set of sensors suitable for the determination of the driver's state. They state that the distraction of the driver is kept to a minimum due to the fact that it is an unobtrusive integration of the sensors. This sensors are not necessarily cameras but textile capacitive electrodes already integrated in a commercial vehicle. This sensors are located in the steering wheel. The steering wheel, with all the sensors included, can monitor the electro cardiogram, skin conductivity and peripheral capillary oxygen saturation (ECG, SC and SpO2). Finally, they describe an embedded system customised to perform data acquisition and online processing of the bio-signals from the sensors. This system can work as an interface for the vehicle's information systems too.

The monitoring of the eyes to estimate the fatigue of the driver has been proved to be an adequate tool to achieve this goal. In [23] Singh et. al. describe an application where the tracking of the eyes is used to estimate the fatigue in the driver and trigger an alarm when it is necessary.

The way to alert the driver takes various forms. From graphical alerts, like text and figures, to acoustic alarms and vibrations from the seatbelt. A small hardware device was also built to deal with the measuring of the rotation of the wheel, the speed of the car and both haptic (vibration of the seatbelt) and acoustic alarms.

Another application that deals with the monitoring of the driver is presented by Friedrichs et. al in [24] where the prevention of accidents by detecting the drowsiness in the driver through the measurement of the odometry in the vehicle (yaw rate and vehicle speed). One of the main features in this article is that it does not need additional hardware as it relies only in inertial sensors available through the CAN bus of the vehicle. Further work will be done in order to resolve the problems with the separation between road curvature and vehicle lurching between the lane markings. A modelling of the whole system i.e. the vehicle is shown and how the Kalman filter is configured to correct the input data from the velocity, acceleration, yaw angle and others. Also it was proposed to use as an input signal the GPS although the refresh rate of the GPS is much slower (50 times slower refresh rate). The proposed solution was tested with 294 drivers in a simulator with a database from Mercedes Benz.

Thanks to the latest development in the field smartphones, it is possible to develop an application to monitor the driver like the one mentioned in [25] by Castignani et.al. They describe a device called Sensefleet to do profiles of the driver and keep a score regarding its behaviour. The score is calculated using the context information like the weather conditions and the topology of the road. The application is said to detect dangerous situations. Through the methodology of fuzzy logic, they determine the score for each driver. They claim to detect the acceleration, steering and other events through the fusion of sensors including GPS data.

2.3.3 On Board devices

The dashboard of the vehicle is where the most important controls for the driver are installed and are easy to access. By placing them in an area determined by the reach of the driver's arm, different controls (air conditioning, turning lights; emergency lights, radio and more) allow the customization of the driving experience. This controls will be placed next to indicators that will tell the driver the information that can be useful to take the proper decisions when driving. An example could be the radio telling news about traffic jams in a determined zone of the road or screens that will show not only the vehicle's condition (speed, revolutions per minute and more) but also the location of the car via a global positioning system (GPS) device. Many studies have been performed in this area. Human Machine Interfaces (HMI) are a specific branch of studies dedicated to the proper location of controls and displays in different environments with the main objective of making the control of a machine as intuitive as possible and the interpretation of information more clear and less ambiguous. The on board devices in the car are no exception to this rule.

An HMI makes any process easier to operate and understand so that there is less chance for mistakes and accidents. This mistakes can be due to random circumstances, physical or mental fatigue and many more. The HMI tackles this problem by making the complex process simpler and displaying only the controls and information that guarantee that the task to be done is executed correctly without mistakes. The importance of making applications and interfaces simpler is very important, so much that the International Engineering Consortium (IEC) estimates that the professional applications have up to 50% of the code oriented for the ease of use for the end user. Some of the rules and protocols to follow with an HMI can be found in the ISO 92411 which specifies the ergonomic requirements regarding office work with visual components. Some of the points this section covers are keyboards, working spaces and the design of a menu. Another one is ISO 14915 that is oriented to multimedia and is oriented to the way the user will navigate through the options of the control panel. Finally, another standard that is appropriate for on board devices is the ISO-IEG 11581 which explains all that is related to the graphical symbols on screens. Fig 2-5 (Taken from Béla G. Lipták, *Process Control*) shows an example of location of controls and displays for an operator. Measurements are in inches.

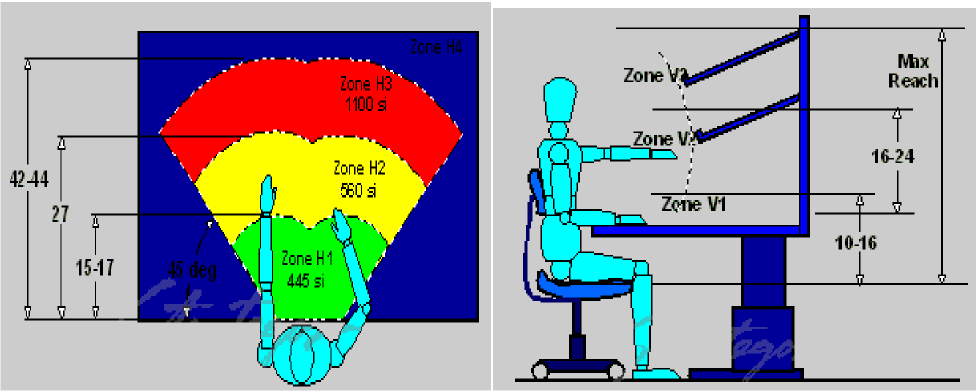


Figure 2-5: Correct positioning of controls according to HMI studies

In order to not interrupt the driver's behaviour and make it as unnoticeable as possible, the position of the Kinect sensor bar was decided to be put in front of the driver, on top of the dashboard. This location allows the best results for the head pose estimation algorithm while not distracting the driver with blinking LEDs or interfering with the field of view through the windshield. Fine adjustments were needed to achieve the final position as the different heights of the drivers demanded that the field of view of the Kinect should consider an area large enough to consider the smallest and tallest drivers. The tests performed with people between 1.60m and 1.80 were successful. Also, because it is on the dashboard, the wheel and hand positioning had to be considered because when the driver performs certain manoeuvres, the face may be blocked partially and therefore the estimation would not be possible.

2.4 Microsoft Kinect

The sensor used is capable of registering both 2D and 3D information from the scene captured by its 2 sensors. If the proper libraries are used, there is no need for a previous calibration process as it could happen with other 3D solutions like a stereo pair where the chessboard would be used. The Kinect will estimate the depth of an object by projecting a known infrared dot pattern and analyse the distortion caused by the object where it reflects and is detected by the infrared sensor (Figure 2-6). This is known as parallax from a stereo system i.e. analyse the shift of one of the projected dots projected from one point and captured-observed from another. It is one of the most economical offers available in the market for a 2D and 3D sensor and it is a robust device that needs no additional configurations other than the proper software, no additional hardware or firmware. As for this system, different drivers and libraries were tested during the development stage. In the end, the latest development tools and drivers available were implemented. Although new versions may be available, the slight changes implemented in this new versions may not work correctly when implemented.

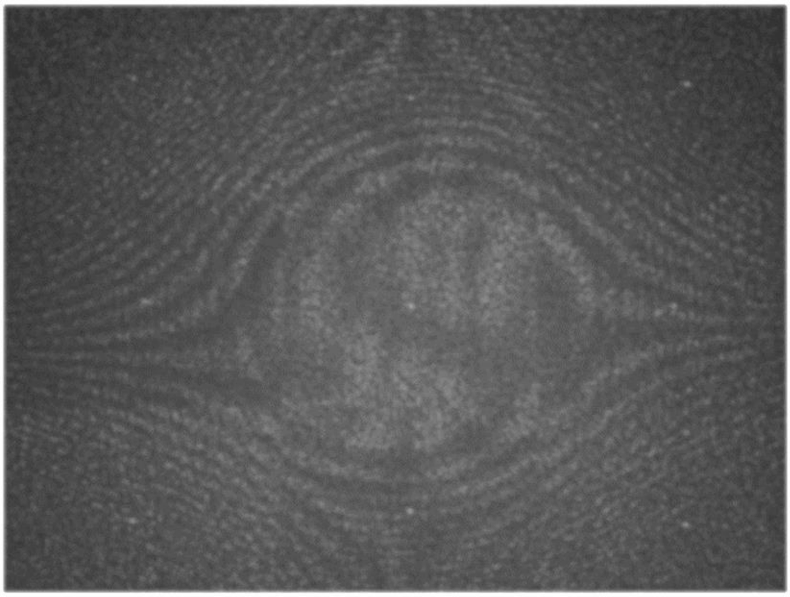


Figure 2-6: Infrared pattern projected on a wall

Because of the potential that a low cost 3D sensor could offer at the moment of the beginning of the thesis and the relative ease of use once properly configured, it was decided to implement the head pose estimation for the driver with this sensor (Fig. 2-7). In the next lines, the hardware that composes it and some examples are described.

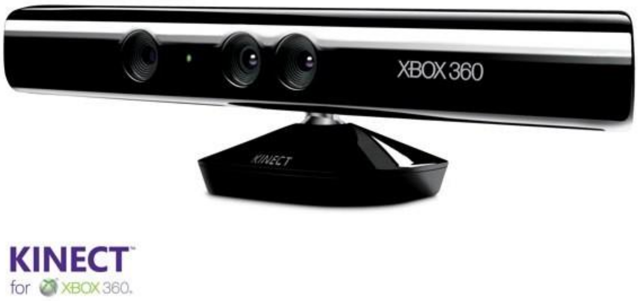


Figure 2-7: Kinect Sensor Bar

2.4.1 Hardware description

It has colour and depth sensing lenses with a horizontal field of view of 57º and the vertical of 43º. The physical tilt range varies from -27 to 27 degrees and the depth sensor has a range from 1.2m to 3.5m. The data streams for the cameras are different, for the colour camera one can have 640x480, 32bit depth at 30 frames per second (FPS) and for the depth camera 320x240

16 bit depth at 30 FPS. All this complemented by a 16 bit at 16 KHz audio stream. The frame rate of the colour image can be reduced in order to obtain a 1280x1024 resolution. This was not used due to the mismatching of the 2D and 3D streams which would have required a calibration process that trumps the sole purpose of using this sensor. Also, an accelerometer (2G) that has an accuracy of 1º as the upper limit. [26] [27]. The next figure (Fig. 2-8) shows the block diagram of how the Kinect works. Notice that in blue is the part corresponding to the colour camera due to the fact that other products from the manufacturer are similar but does not include this part.

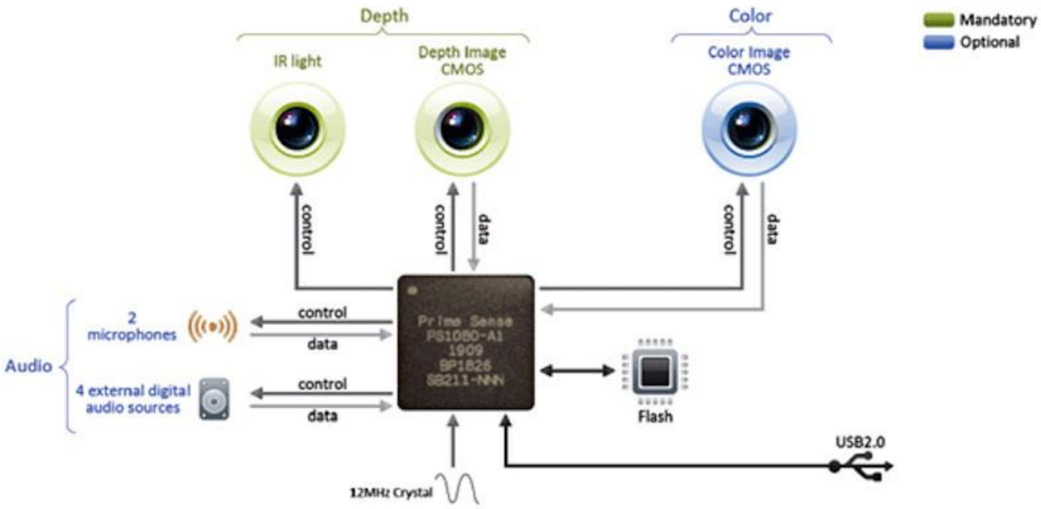


Figure 2-8: Diagram from the manufacturer PrimeSense

2.4.2 Examples

Some projects have been developed with the Kinect after the release of the first drivers available through the result of a competition partially sponsored by Adafruit Industries [28] by the end of 2010. A month later PrimeSense[29]], who designed the depth sensing in the Kinect, released their own drivers. Now, a large set of experiments have been developed in different branches. Many others are available in web pages such as Hackaday [30]].

The projects with the Kinect involve a wide variety of applications: As demonstrated in [31] the tracking of a hand is useful for applications oriented to virtual reality. Some others like [32] provide a solution that uses gestures as a way to interact with virtual objects in an augmented reality application. In [33] the detection of a human presence is achieved with a Kinect by using the depth information and 2D information associated to the head's contour. The Kinect can also be used as a complementary sensor in a more complex system such as in [34] where a mobile robot uses, as part of its localization system in an indoor environment, a Kinect for the location of landmarks to correct the robot's position.

In [35] Amorim Vaz et. al. describe a system to train young drivers in order to improve the learning of the traffic rules. They achieve this by using the Kinect sensor bar with an Arduino micro controller. The application itself consist of a marker that represents a vehicle (position and orientation) in a virtual world and different situations that could happen in real life are created. The system got the feedback from sociologist and psychologist in order to improve the applicability of it.

In [36] the Kinect sensor bar is used for ADAS too. They developed an application that will identify 4 situations that may cause an accident such as talking on a cell phone or looking at an external object. They alert with acoustic sounds in case a distraction is detected. This alert, as claimed in the article, is done for testing purposes only. The whole system was tested and installed in the medium-fidelity driving simulator of the Virginia Driving Safety Laboratory (VDSL. They claim that the system could be improved by refining the motion capture and eye tracking for some other dangerous situations. Finally, the fact that the commercial motion capture technology is available is considered as promising for further studies. The different tests were done on the team members only and it is claimed that it may not be representative.

Another application where the Kinect sensor is used for an ADAS is explained by Ohn-Bar, E. et. al in [37]. They developed a 2 stage method for the detection of a hand and hand-object interaction. As the application described earlier, the system can detect a determined set of activities. This predetermined activities are integrated in a classifier that is later tested with a Kinect captured dataset under real world driving situations consisting of more than 7000 sample frames. Different algorithms were tested for the RGB and depth images captured by the Kinect. The goal of the system is to study and infer more into the driver's distractions while driving.

2.5 Conclusion

The driving assistance systems have the main goal to make the driving experience safer. Since its early implementations, they are responsible for reducing the fatal tragedies associated with driving. From a seat belt and special brakes to the latest cameras on board, they are now an important part in any design of a modern vehicle. Some will monitor the outside in search of pedestrians, animals and obstacles, other will be used for context analysis such as the road sign detection and lane marks on the road. But this is only the outside analysis, there is also the inside where the supervision of the driver is the main objective. By supervising the driver with a small on board device, properly place, it is possible to detect misbehaviours such as falling asleep or distractions during critical moments. The device used , although not designed for applications like this, allows the combination of different algorithms and opens the door for testing some others that require more expensive sensors such as 3D point cloud algorithms. In this application, the usage of the Kinect makes the usage of 2D and 3D algorithms possible and it is possible to export this proposals to other systems with different hardware as the 3D structure does not have to come from the Kinect sensor bar and the colour image could be obtained from another sensor. This of course, assuming that the calibration and setup of the other system is correctly performed.

The application consist of a Kinect sensor located in front of the driver, on the dashboard just behind the wheel. This position allows the detection of the whole face without distracting the driver or blocking its field of view. The application will scan the inside of the car for a face. When the face is detected, it will build a 3D structure based on the associated points projected by the infrared device in the Kinect. The first face detected is used as a reference to compare with the next of faces to be captured. Once the next face is captured, the comparison between the current and first face will be performed using an iterative algorithm previously adjusted to obtain reliable results as fast as possible. The rotation angles are obtained and from this results different conclusions can be done according to the study to be performed such as the drowsiness detection, distractions and building profiles according to where the driver is looking during different scenarios (highway and urban for example) that will allow a better estimation on where to place different controls and displays in the vehicle.

CHAPTER 3. GENERAL DESCRIPTION

3.1 Introduction

In this chapter, the overall idea of the thesis is described. Again, the main goal of the proposed solution in this thesis is to estimate if the driver is falling asleep or distracted through the head pose of the driver in 3D with a low cost sensor and the gaze orientation. In order to achieve this, 2D and 3D algorithms are used in different stages of the solution. Compared to other solutions for head pose estimation, this one is more robust to lighting conditions when performing the head pose estimation thanks to the infrared capabilities of the Kinect that allow the construction of a 3D structure under more adverse environmental conditions that other colour sensors couldn't handle. Thus, the estimation of the rotation angles is more robust than those solutions that use only 2D algorithms like optical flow based on key-point features of the face. The platform test (the vehicle where it was tested) is also described.

3.2 The Platform

The thesis is part of a set of applications mounted in an experimental vehicle named IVVI 2.0 (Intelligent Vehicle based on Visual Information). This vehicle is used as a platform test for the latest developments of the ADAS done in the intelligent systems laboratory (ISL), it is equipped with a variety of sensors that will perform different tasks. From multilayer laser scanners to video cameras of different type. As opposed to the testing done in the laboratory, IVVI allows the test of the developed algorithms in real life conditions that are, most of the time, not ideal. Thus offering a great feedback from the behaviour of the tested solution in the vehicle. This is especially important with the computer vision algorithms as the lighting conditions can be a serious problem when performing any type of detection. The ADAS tested in IVVI can perceive people or different important objects such as road lanes, traffic signs (both detection and identification) and vehicles in the nearby. In Fig. 3-1 it can be seen the different ADAS being developed in the IVVI 2.0. Because of the constant evolution of the ADAS, the list of applications available is constantly changing, still below are listed some of the latest ADAS.

1. Pedestrian detection [38].
2. Driver monitoring under daylight and night conditions [39]. This is to be replaced by the proposed solution in this thesis.
3. Lane detection and classification system [40].
4. Traffic sign recognition [41].
5. Stereovision based obstacle detection and ego-motion [42].
6. Fusion procedures, using laser scanner and vision. [43]

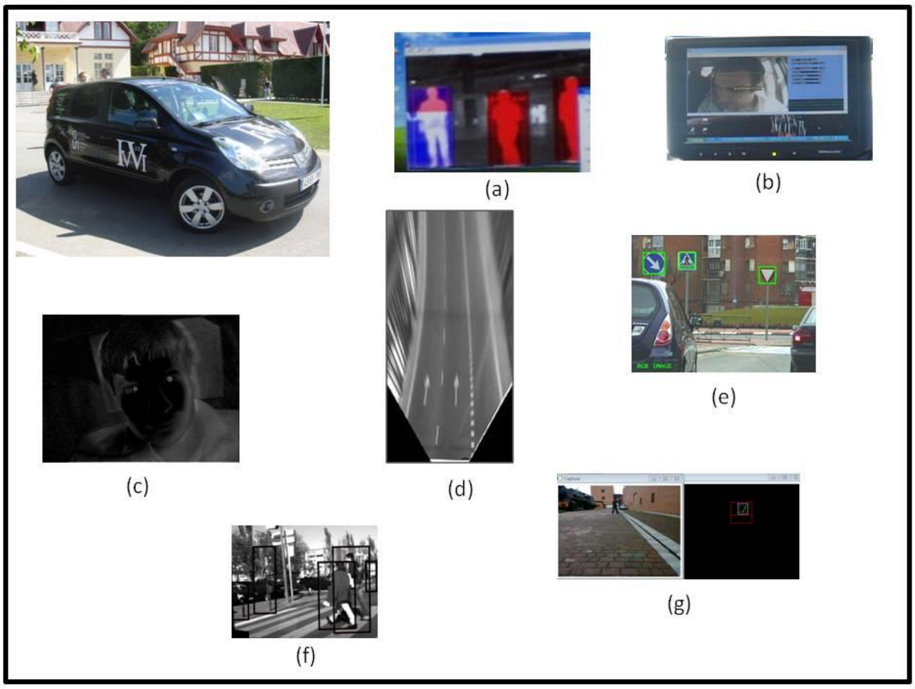


Figure 3-1: Technologies being developed and researched in IVVI 2.0

This solutions will change in time as they are upgraded and new sensors are mounted that allow a new approach to an existing solution. For example, the laser that allows the fusion with vision is being replaced by a 4 layer scanner. Also the computers on board are constantly upgraded so that the algorithms that can be executed are more powerful and thus a more robust ADAS can be achieved.

As stated earlier, a set of different sensors are mounted in the vehicle and this variety allows the correct detection of what one of the many mounted ADAS was intended to do. This sensors are updated constantly to keep with the latest development in the field of research. By keeping the latest available hardware in the platform, more precise ADAS can be done and the robustness of others can be improved. Part of the hardware mounted is:

1. A colour camera for driver monitoring in 2D. Later would be replaced by the proposal in this thesis.(Fig. 3-2a)
2. An on board colour monitor to see the results of the ADAS being used. (Fig. 3-2b)
3. Two laser scanners, a 4 layer type and a 1 layer type. The choice of which of the 2 scanners is used depends on the application to be done. (Fig. 3-2c)
4. A far infrared camera to detect pedestrians in dark environments. (Fig. 3-2d)
5. A colour camera used for the detection and classification of the road signs. (Fig. 3-2e)
6. One Bumblebee stereo system that is used for different obstacle detection and classification. Also used for telemetry and navigation. (Fig. 3-2f)
7. A GPS-inertial device that acquires information of the vehicle's movement and position. (Fig. 3-2g)

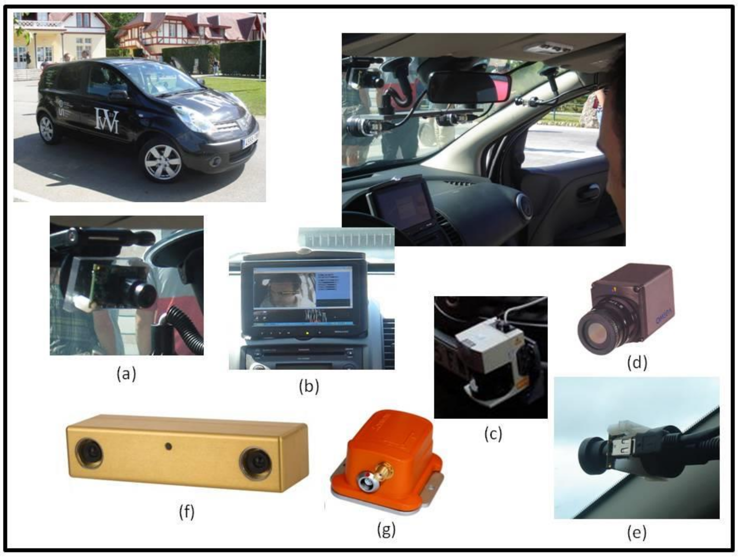


Figure 3-2: Sensor devices installed in IVVI 2.0

The hardware described previously makes possible the different ADAS implemented in the IVVI 2.0. Some of this sensors will be complemented by others and others will be replaced by new versions and firmware in order to keep the ADAS as modern as possible.

However, the set of sensors is one part of the ADAS, the other part is the communication between them and the computers that process the information of the sensors. The next list describes the support hardware that is mounted in IVVI 2.0.

1. One Wi-Fi router with the latest network protocols to allow the connection between the computers but also with an available exterior network through the 802.11.a/b/g/n protocol. It is installed in the bumper of the vehicle. (Fig 3-3a)
2. A variety of mobile devices like PDAs and Smartphones connected to the computers in order to allow the interaction with those on board. (Fig 3-3b)
3. To keep the equipment running, batteries, inverters, and a backup system are mounted in the trunk of the vehicle. (Fig 3-3c)
4. The whole processing of the information and storage location of the ADAS algorithms is in the computers located in the trunk. Constantly updated to keep with the more robust algorithms but also to reduce the time consumption of the ADAS. (Fig 3-3d)
5. Finally, a Can-Bus reader gives the opportunity to read the basic telemetry of the vehicle like speed, revolutions per minute and more. This can be used to get a general idea of the status of the vehicle. (Fig 3-3e)

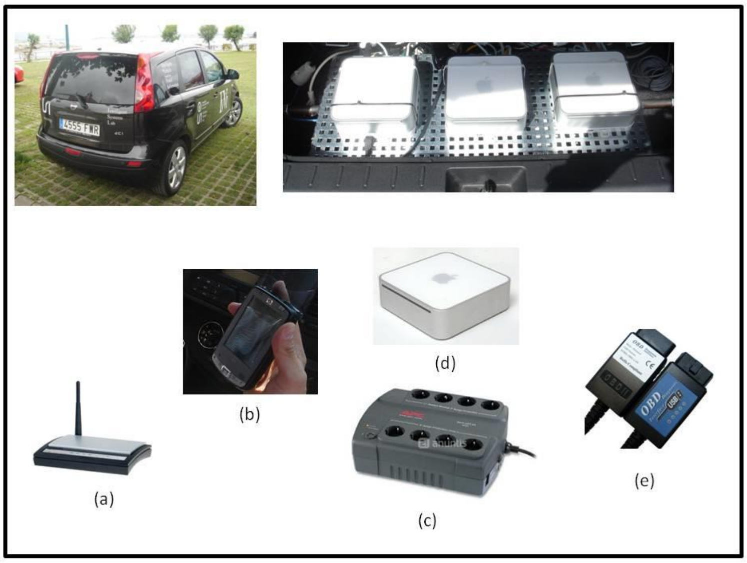


Figure 3-3: Processing units and other devices installed in IVVI 2.0

3.3 Point Clouds

The data captured by the Kinect is a 3D structure composed by points that have both 2D and 3D information and more. One point will have the *x, y, z* coordinates according to the reference system of the Kinect but also the colour in an RGBA compressed format. Therefore, from a point cloud, it can be obtained both 2D and 3D information that will be useful for the head pose estimation. Not only the 2D for the facial detection as any other camera would do but the 3D information in a structure that is calibrated and requires no previous alignment of objects or pattern recognition to guarantee that the system is calibrated. This 3D information is what distinguish this proposal as many others use 2D algorithms but do not take into account the depth of the analysed scene thus being susceptible to false positives and inaccuracies due to lighting conditions. The 3D data of the cloud handles this problems in a better way.

In order to handle these vital structures, the Point Cloud Library (PCL) were used. This library is a large scale open project aimed at the processing of both 2D and 3D data. Some of the latest algorithms are implemented and allow the customization of certain parameters in order to adjust them to the nature of the experiment to be done. Because it is cross-platform, it can be used successfully on Windows, Linux, MacOS and recently Android/iOS. Below is an example of a pointcloud (Fig. 3-4 taken from the website of the developer) representing an object with the shape of a cylinder. This image shows also the reference axels.

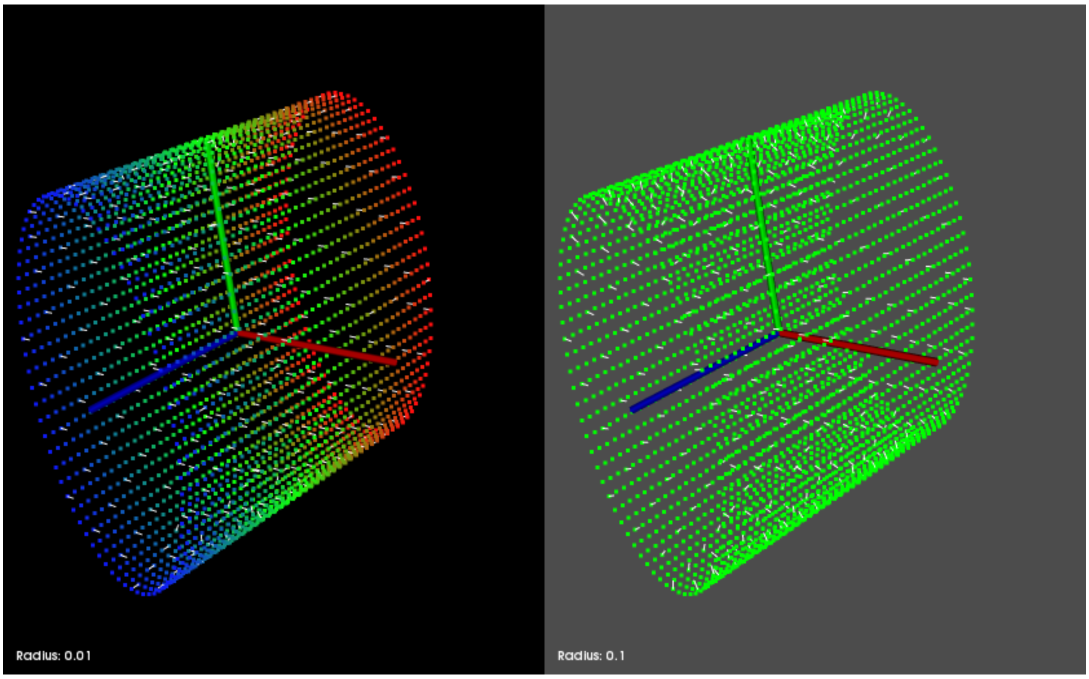


Figure 3-4: Sample of a pointcloud representing a cylinder.

3.3.1 Format of a Point Cloud

The point cloud file (.pcd) has a determined format. It is different to other 3D files as it is explicit in the information that it contains. The fields this class contains are: width, height (specifying the dimensions of the cloud), the points that compose the cloud in *XYZ* format and when colour is available, *XYZRGBA*. When the sensor allows it, as is the case for the Kinect sensor bar, the field of intensity *I* is available replacing the *RGBA* part of the cloud. A Boolean variable to know if the cloud is dense i.e. if there are points that contain infinite or invalid values. The orientation and origin of the cloud are also in the format but most of the algorithms do not make us of them. This format, the standard pcd cloud, was used for the whole analysis presented in this solution and can be seen in Fig. 3-5 with random data filling the fields of the variables corresponding to the format of the cloud.

One variation of this format is the compressed point cloud file that encodes the data so that the space in memory is reduced. This format is useful for special situations that require small clouds without losing information such as recording a sequence or transmitting to another device especially through limited communication channels. The compressed format was used for the offline analysis and the construction of a database as it allowed a fast recording with more clouds per second compared to the standard format.

```
# .PCD v.7 - Point Cloud Data file format
VERSION .7
FIELDS x y z rgb
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 213
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 213
DATA ascii
0.93773 0.33763 0 4.2108e+06
0.90805 0.35641 0 4.2108e+06
0.81915 0.32 0 4.2108e+06
0.97192 0.278 0 4.2108e+06
0.944 0.29474 0 4.2108e+06
```

Figure 3-5: Example of a header of the PCD file

3.4 Computer Vision

One of the main tools used in this proposal is the computer vision. Without this, it would have been more complex and challenging to achieve the same goal. Algorithms like Viola-Jones to detect the face are used so that other algorithms can proceed with their tasks. The application of this algorithm can be seen in Fig. 3-6 where the face is detected from the frame obtained from the colour stream. Although the haar-like features are very common in the world of computer vision, it is not the main method for this ADAS. 3D algorithms are used too like the Iterative Closest Point (ICP) and filtering according to the distance from the sensor. This is one of the main contributions of this ADAS due to the fact that it uses new tools to perform a task that others have done using 2D algorithms that can be susceptible to drastic lighting conditions. For further improvements in the application, more 2D algorithms are proposed like the opening and closing of the image obtained in infrared to reduce the noise caused by the projected dots. The latest available version of OPENCV at the time of development that were compatible with the rest of the other software tools were used. The analysis of the eyes is a good example of the application of a determined set of tools available in the world of computer vision. Not only are the tools necessary to solve the problem but also the correct parametrization and the order in which they are executed.

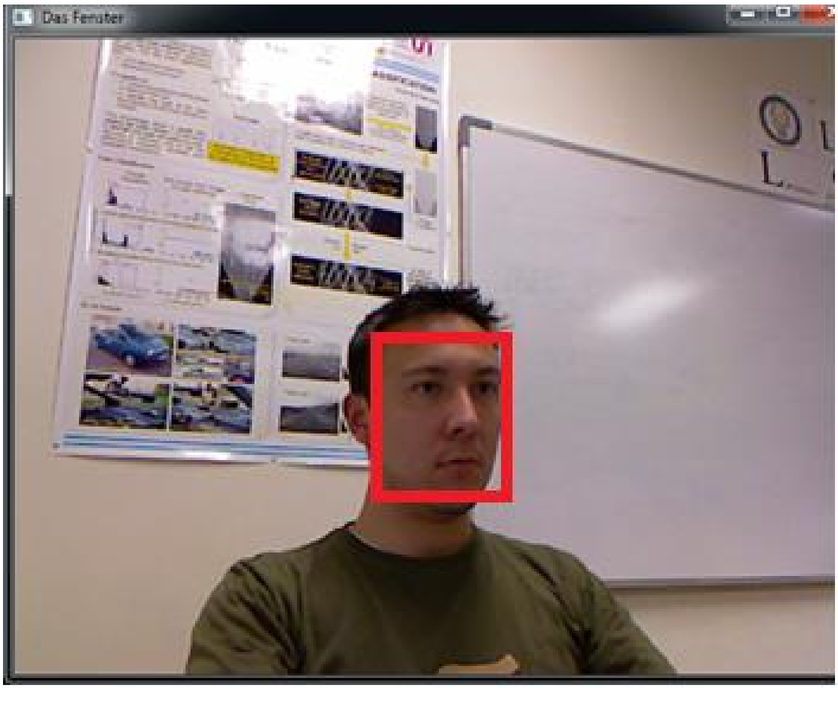


Figure 3-6: Example of the haar-like features detecting the face

3.5 Other Resources

An IMU manufactured by MTi-G [44] was used as a ground truth to compare the results obtained from the proposed solution. This device can give the 3 rotation angles according to its initial position; when attached with a GPS antenna, the coordinates of the current position are available too. Thanks to its small dimensions, it can be used in a variety of tests including the ground truth for ADAS oriented to autonomous navigation (accurate up to 10 meters if properly configured) or as a ground truth to determine if the rotation angles of the head match with those of the solution here proposed. Special attention was put on the fact that under certain scenarios a cumulative error was present in the measurements and was considered when comparing the results. The image below (Fig.3-7 taken from the website of the manufacturer) shows the IMU used as ground truth. The golden connector is for the GPS antenna and the silver connector is for the gyroscope data.

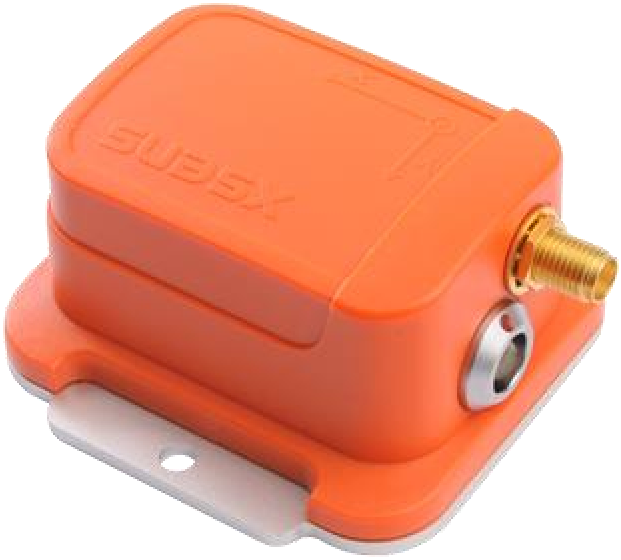


Figure 3-7: Example of the used IMU

Because of the nature of this ADAS, some information could be taken into account in order to improve the accuracy and speed of execution. For example, the head of the driver will be located in a determined position and distance from the Kinect. Therefore, the background can be eliminated by filtering those objects that are further away from the sensor. This allows the reduction of the space to search for a face and eliminate those regions that could give a false positive result from the haarl-like features algorithm due to the chromatic similitudes it may have.

3.6 Proposal Phases

The proposed thesis will achieve the main goals through the execution of many different stages that will be later described in detail. From capturing the first frame from the Kinect sensor bar to the analysis of the rotation angles to determine if the driver is falling asleep or looking away from the road, this sub-stages describe the process to achieve the final goal. Also, this phases are an attempt to divide the whole application in smaller blocks in order to make it easier to understand but also to facilitate the track of the progress of the proposed solution. The phases into which this thesis is divided are:

- Spatial restrictions and facial detection.
- Head pose estimation.
- Determination of misbehaviours.
- Gaze orientation.
- Drowsiness detection
- Results
- Conclusions and future works.

The whole application can be explained in a series of images as seen in the image below (Fig. 3-8). This steps are intended only to clarify the execution of the application and will be later explained into detail.

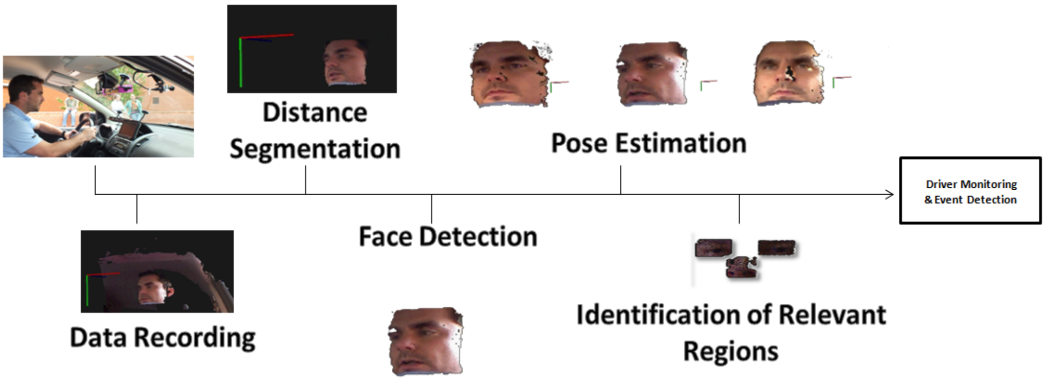

Distance Segmentation
Pose Estimation
Driver Monitoring & Event Detection
Face Detection
Data Recording
Identification of Relevant Regions

Figure 3-8: Example of the proposed solution

3.6.1 Spatial Restrictions and Face Detection

Because the thesis is done inside the vehicle, some considerations can be assumed in order to reduce the search speed. This is achieved by reducing the space where to search for a face. By taking this two considerations into account, the search is performed faster and does not compromise the reliability of the facial detection algorithm. Because this is the first step of the algorithm and is used as a reference, it is considered the most important. The next points are intended to give an overall view of what is to come in the next chapters, more information and in-depth analysis are later explained.

3.6.2 Estimation of the Head Pose

Once the face is detected, the estimation of the pose of the head is performed by using the iterative closes point algorithm (ICP). The ICP algorithm will take as an input two clouds and compare them to obtain the translation matrix composed by the rotation and translation. The first cloud is of the first face detected, the second cloud is the last captured by the Kinect. Again, different adjustments to the algorithm allow a fast calculation of the rotation matrix and accuracy of the estimations. From the matrix obtained, the Euler angles are extracted for the three rotation axis of the second cloud in reference to the first cloud (the current face compared to the reference).

3.6.3 Determination of misbehaviors

After the rotation angles are obtained, it is established what is considered a misbehaviour. The two misbehaviours are: distraction and falling asleep. By analysing a determined angle of rotation of the head, it is possible to establish if the driver is falling asleep (head falling down). This rotation is monitored for a determined time in order to avoid false alarms when the driver is just looking down. When the system determines that the driver is asleep, an alarm is triggered. The other misbehaviour, distraction, is defined as the driver’s head is not oriented to the front. A maximum rotation angle is established and when it is exceeded, the alarm of distraction is triggered.

3.6.4 Gaze estimation

It was implemented a gaze estimation system that can distinguish 3 regions where the driver is looking. This estimation is immune to the rotation of the head as it compensates the position of the pupil with the yaw angle and is intended to be a complementary tool for the head pose estimation in order to have a more precise description of what the driver is doing. Although the resolution of the camera was low, the 3 regions (left, centre and right) could be defined under non-ideal lighting conditions.

3.6.5 Drowsiness detection

An analysis of the closed eyes was performed in order to estimate if the driver is falling asleep while driving. This analysis takes into account the PERCLOS parameter to determine if an alarm must be triggered to alert the driver. PERCLOS was calculated by studying the amount of time the eye was kept close during an interval. This amount of time was calculated with the number of frames that the eye was closed. For this, a function was implemented that would detect if the eye was closed using different image processing algorithms and blob analysis. Overall, it delivered a high success rate and was tested with different individuals and lighting conditions.

3.6.6 Results

The proposed algorithm was tested with different users under a variety of lighting conditions and an IMU was used as the ground-truth for the rotation angles obtained from the head pose estimation algorithm. Overall, the algorithm worked as expected with fast and accurate estimations of the head pose and correct detections of misbehaviours performed by the driver. The tools to see, analyse and do interpretations of the results is described in this section. Although getting the rotation angles is the main goal, more complex layers of analysis can be done from this results.

3.6.7 Conclusions and Future Works

New ideas are explained here that were tested but not implemented for different reasons. One of them is the use of the infrared stream available from the Kinect. The image obtained from this stream is affected by the noise of the pattern projected by the Kinect to determine the depth. This pattern distortions the infrared image and adds a noise similar to the salt and pepper. Also, the frame rate of this stream is lower than the colour stream. A robust gaze orientation could be performed with more resolution from the camera or with additional techniques already tested to solve this problem. Currently, a discrete gaze orientation was tested that could establish if the user was looking to the centre, left or right side.

CHAPTER 4. ESTIMATION OF THE HEAD POSE

4.1 Introduction

This chapter will explain the first steps of what is to be the whole solution. First, the spatial restrictions that allow the optimization of the following algorithms are explained. Then, with the space restrictions applied and the space search reduced, the facial detection is performed. Without the optimization through the spatial restrictions, the solution would be inefficient as it would perform searches where it is unlikely to find the driver’s face and increase the time of execution while also increasing the chance to do a mistake.

Once the scene has been captured in 3D and the face has been detected, the point cloud of the face will be built. It is with this point cloud of the face that the estimation of the pose of the driver’s head will be achieved. Different algorithms will be done to the clouds of the face such as the reduction of the amount of points it contains. Then, the function that will obtain the rotation matrix between the reference and current cloud is explained with its correct parametrization to obtain fast an accurate results. Following this section, it is explained the importance of resizing the detected face to the dimensions of the reference face in order to increase the speed of execution. The final part will deal with the interpretation of the obtained results from the rotation matrix.

4.2 Spatial Restrictions

Due to the fact that the thesis is executed inside the vehicle, certain spatial restrictions can be assumed. The spatial restrictions will allow the efficiency of the proposed solution. Because the main goal is to estimate how the driver is behaving, the head must be detected first and in order to be detected fast, all the information that can reduce the space where it could be is beneficial.

The first restriction takes the design of the car into consideration. The way that the cockpit of the car is designed allows the assumption that the seat will be located at a determined distance from the sensor. Therefore, all the objects that are beyond this distance are eliminated. This removes space that does not need to be searched for a face and it will remove all objects in the back that could have chromatic similitudes to a face returning a false positive and thus compromising the reliability of the solution. In order to obtain adequate results, the objects that are more than 1 meter away from the sensor are discarded. After applying the depth threshold procedure, the filtered cloud (Eq. 1) is obtained.

$$\text{Filtered Cloud}=\begin{bmatrix} Pf_{11} & \cdots & Pf_{1n} \\ \vdots & \ddots & \vdots \\ Pf_{m1} & \cdots & Pf_{mn} \end{bmatrix} \qquad (1)$$

Where the z component of point Pf is lower than 1 meter.

It was necessary to implement an additional filter due to the nature of the hardware that would return an invalid value when the depth sensor could not calculate correctly the distance of one point in the image. This value was not a number and it was read as "*NaN*". By removing the *NaN* data, it was guaranteed that the cloud would be composed only of valid data so that the algorithms to come would have the right type of structure to work with although this brought additional challenges to the colour part of the captured image that will be later explained.

The next spatial restriction is related also with the design of the car. Because the driver will be seated at a certain distance with a variation of a few centimetres, the size of the face of the driver will not vary much and can be assumed with an average width, height and a small standard deviation of this measurements. This is useful for the next algorithm that will detect the face to estimate the driver's head pose.

The spatial restrictions were obtained from the dimensions of the testing vehicle IVVI and the parameters were set accordingly to filter the undesired space. This dimensions can be adjusted later if the environment where it will be used changes drastically from the first assumptions. During the latest testing stage, the solution was tested in a simulator in another laboratory where the distance between the wheel and the driver were different from the IVVI 2.0. Therefore, the spatial restrictions had to be adjusted in order to obtain the ideal results from the application. Because the speed is an important factor for this proposal, different optimization steps were performed besides the spatial restriction previously explained without compromising the reliability of the results. Other methods to increase the speed of the solutions are later explained.

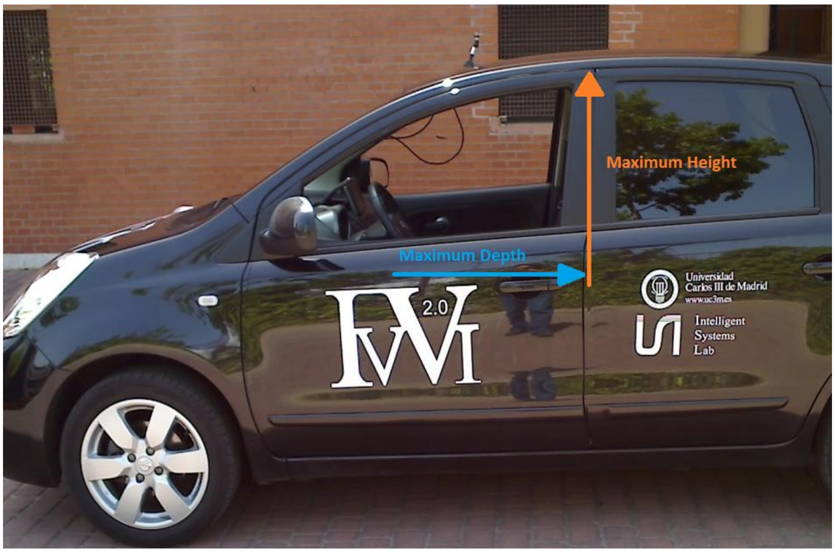


Figure 4-1: Considerations for the spatial restriction

4.3 Facial Detection

Once the search space is reduced by the previously described filters, the facial detection algorithm can proceed. This is done using the Viola-Jones [45] approach as it has proven to be a reliable and, if properly configured, fast algorithm to detect a face in an image. But to detect the face, the algorithm requires an image where to search for it i.e. a 2D and not a 3D structure thus, the first task is to obtain an image from the point cloud.

The search of the face is done by the Viola-Jones approach of detecting an object. The classifier is trained with a determined number of pictures of the object that wants to be detected. The training usually takes hundreds of samples of what it should detect but also with other images that represent what is not a desired object but that has similar features. This is the same training that is used when using neural networks where both positive and negative examples are necessary to obtain a robust solution. The algorithm will search for the desired object in a region of interest inside the input image. If the region has the chance to have the desired object, it will return a positive value and negative when it does not. This ROI is then moved across the image and its dimension can be changed so that the object can be found without knowing its size. In order to detect the object without knowing its size, it is necessary to perform multiple scans on the image with different window sizes.

With the trained classifier for detecting faces, it will scan the whole image looking for a face with different sizes by increasing the dimensions according to a configuration parameter. For this application, the increase was set at 10%. The main advantage of this search method is that it searches for a face in a determined region by applying only one feature at a time, this feature is very simple by itself but when combined with many others, up to 6 000, a face can be detected with a high success rate. This features can be seen in Fig. 4-2 (taken from the website of the developer). When one of this features fails, it will not apply the rest of the features and discard the region as a possible candidate for containing a face. If the first feature passes, it will apply the second and so on. This can be used for facial detection but also for other features as it is an object detector that, in this particular case, is oriented to detect a human face. It can also be trained to detect other objects such as cars, symbols or facial features like the eyes (single or pair), mouth, nose and ears.

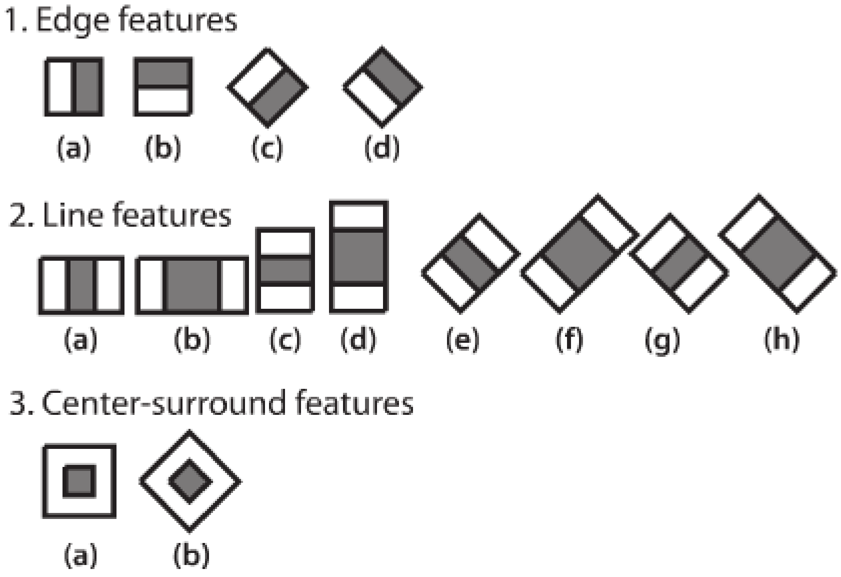


Figure 4-2: Example of the features for the facial detection algorithm

To obtain the image where the face, the first step is to create a matrix where each position belongs to a vector of 3 positions as seen in Eq.2.

$$\text{Cloud}=\begin{bmatrix} Pc_{11} & \cdots & Pc_{1n} \\ \vdots & \ddots & \vdots \\ Pc_{m1} & \cdots & Pc_{mn} \end{bmatrix} \qquad (2)$$

Where Pc_{mn} represents one point in the cloud that contains the *x, y, z* coordinates inside the cloud and the encoded colour component in RGBA format. The result can be seen in Fig. 4-3. The cloud has enough resolution to distinguish some facial features such as the nose and mouth using only the 3D data.

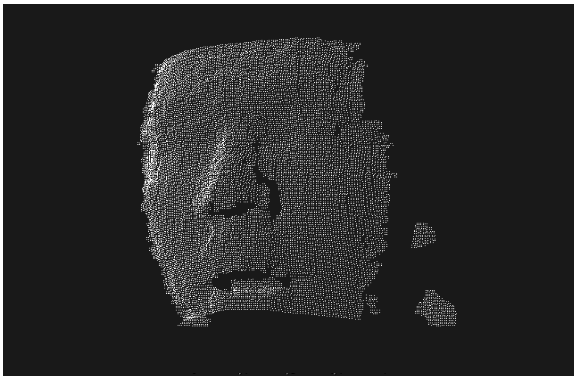


Figure 4-3: Example of the segmentation of the detected face, no cropping

The dimensions of this matrix are the same as the width and height of the point cloud. Each of the matrix positions will store a numerical value for each colour component blue, green and red (BGR) thus conforming one pixel of the image. The whole cloud is scanned and the BGR values are assigned to its corresponding position in the matrix to conform the final image or 2D representation of the point cloud. This results in the transformation of an *XYZRGBA* structure (the point cloud) to an *XYRGBA* structure (the image).

The image where the face is to be found contains only those pixels that passed the filtering processes previously described and black for the pixels that did not passed. By reducing the spatial search for the facial detection algorithm, the time for obtaining the result is reduced. Also the chances for detecting a false positive due to chromatic similitudes or another face that does not belong to the driver (the passengers in the back) is reduced.

This is not the only consideration to optimize the search of the face, the parametrization for the facial detection algorithm is another way to reduce the time of the search. By doing a search of the face first, the detection of the other important features of the face is optimized because if no face is found, the rest of the searches will not be executed thus saving time. The parameters for each of the objects to be found are carefully set. This means that parameters like the neighbour objects, the minimum size of the object found etc… is different when detecting a face or detecting the eyes and nose. This allows the discrimination of objects too large to be what was supposed to be (detect half of the face as one eye for example). The previous criterion is complemented with some geometrical restrictions. Considering drawing techniques for drawing faces, certain assumptions can be done that will improve the location of the desired object (Fig. 4-4, taken from the website artifactory). One of this considerations is the pair of eyes that are located half of the total height of the face.

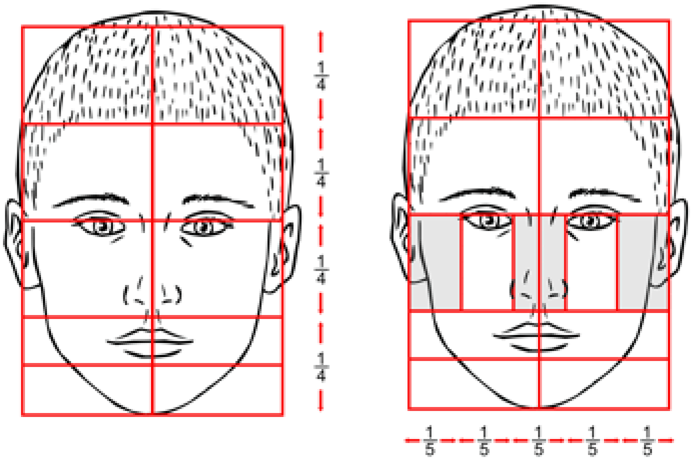


Figure 4-4: Drawing techniques used to optimize the features extraction

This allows the construction of a ROI inside the detected face where the pair of eyes can be found, same with the other features. As for the face itself, a minimum and maximum size were specified.

The assumption is that the driver will be at a certain distance from the sensor and the dimensions of the face will not change drastically due to the position of the seat that will not vary more than 1 meter. Therefore, by specifying an average size for the face to be found with a standard deviation, the face detection algorithm can perform faster. The first face detected, with its features and dimensions, are a corner stone to this solution as it will be used as a reference to compare with the rest of the faces to be detected in the future frames.

The faces to be detected later, after the spatial restrictions are considered, will be resized to the size of the reference face. This is so that the errors obtained from the Viola-Jones from detecting the same face in a different frame do not affect the next algorithms. As seen in Fig. 4-5, the resulting cloud contains the colour information of the original cloud but also the 3Ddata that will be later used. This errors can be due to different variables, one of them is the lighting conditions that change from one frame to another. By resizing the current detected face to the original, the errors from the head pose estimation algorithm can be reduced. This will be later explained.

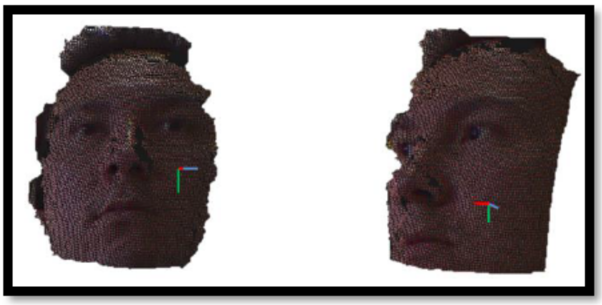


Figure 4-5: Example of the face detected

The detected faces in the image have an associated structure that stores the width, height and the upper left corner of the rectangle that envelopes them. Because the points in the colour image come from the point cloud, they can be used to obtain the depth at that precise point by reading the data of the point cloud at the determined pixel's coordinates. This is used to create a cloud with only those points that belong to the rectangle of the detected face. The resulting cloud is a 3D structure (*XYZRGBA*) of the detected face with the same dimensions as the reference face and can be explained by the next equation (Eq. 3):

$$\text{Facial Matrix}=\begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{m1} & \cdots & P_{mn} \end{bmatrix} \qquad (3)$$

Where P_{mn} represents one point in the cloud that contains the *x, y, z* coordinates inside the cloud and the encoded colour component in RGBA format.

Until now, the point cloud of the captured scene has been filtered by different filters under determined assumptions. Then, the face was detected and its features located too with a smart approach using ROIs. Finally, a 3D structure of the original and current face are built thanks to the coordinates of the detected face and the depth information of the point cloud. In the next section it is explained how to obtain the rotation angles of the head by using the two captured clouds (current and reference face), the correct parametrization of the algorithm and the filtering used to increase the speed while still delivering reliable results. The detection of the face was successful inside the vehicle as shown in Fig.4-6. The left image shows the 3D cloud of the whole scene and the right image shows the cloud that belongs only to the detected face.

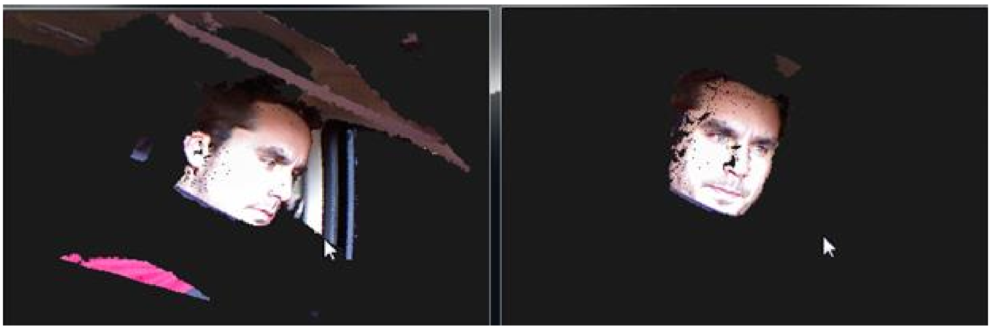


Figure 4-6: Sample of the facial detection inside IVVI 2.0

4.4 Down-sampling the cloud

The cloud is down-sampled using the Voxel Grid algorithm which takes a 3D structure and reduces the amount of points by taking small blocks of the structure and replace them with one single point (the centroid). The radius of the sphere that will be replaced by one point is very important for the rest of the solution because if the radius is too big, it will return a cloud with very few points that other algorithms will find challenging to work with; if the radius is too small, the change will be unnoticed and the efficiency improvement will not be reached. As seen in the equation Eq. 4, the point *p′* that will replace the cloud is defined as the centroid of a specific region.

$$p'(x',y',z')=\left(\frac{\sum_{i=0}^{N} x_i}{N},\frac{\sum_{i=0}^{N} y_i}{N},\frac{\sum_{i=0}^{N} z_i}{N}\right) \qquad (4)$$

A graphical representation of the down-sampling method can be seen in the figure below (Fig. 4-7). To the left is a cloud composed of 9 points. When applying the filter, the points in the outside are replaced by the centroid coloured in blue.

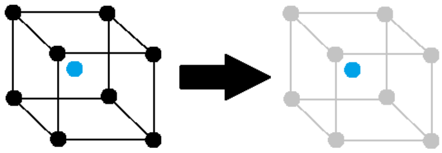


Figure 4-7: Example of the down-sampling method

The reason to do a down-sampling is because the amount of points in the cloud obtained from the face detection can slow down the overall solution without increasing the accuracy of the algorithm in a significant way. The advantage of this filter is that the shape of the structure is kept as opposed to those that would remove the points according to their coordinates or any other features like the pass-through filter used earlier to remove the background. The final value used in this application was obtained after several tests measuring the execution time of the down-sampling until it was under 10mS. The accuracy of the final results varied under 0.5º from what was obtained using the whole cloud. This parameter should be adjusted depending on the hardware that is doing the capture of the scene as the density of the cloud of points may vary.

4.5 Using the ICP algorithm

The Iterative Closest Point (ICP) algorithm [46] is used to determine the pose of the head. By performing a determined set of iterations, the algorithm will find the transformation between two clouds. ICP will return the transformation matrix composed of the rotation and translation data that exist, when possible, between the reference cloud and the target cloud. It can be configured to stop at a minimum error between the two clouds or the number of iterations it takes to obtain the rotation matrix. As for this project, both parameters were set so that the result would be as fast and accurate as possible. The steps the algorithm performs are:

1. Associate the points according to the nearest neighbor criterion (Eq. 5). I.e. Closest points between the two point clouds associated in order to allow the later comparison.

$$d(p_1, p_2) = \|p_1 - p_2\| \qquad (5)$$

Where the point of the model cloud is p_1 and $p_2$is the point of the cloud to be compared.

2. The transformation parameters rotation and translation are calculated between the two clouds using the minimum square equation (6).

$$MC = \frac{1}{n}\sum_{i=1}^{n}(p_1 - p_2)^2 \qquad (6)$$

Thus transformation parameters applied to the latest point cloud are obtained for the later comparison.

3. The previous parameters are applied to the cloud to be evaluated (7).

$$Mt = R(\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + T) \qquad (7)$$

Where *T* is defined as the translation vector $\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}$ for the 3 axis of the cloud, R is defined as the rotation matrix obtained from the multiplication of the 3 rotation matrixes for each axis R1 (8), R2 (9) y R3 (10) in the Z, Y and X respectively.

$$\text{R1=}\begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & 0 \\ sin(\Delta\theta) & cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (8)$$

$$\text{R2=}\begin{bmatrix} \cos(\Delta\delta) & 0 & \sin(\Delta\delta) \\ 0 & 1 & 0 \\ -\sin(\Delta\delta) & 0 & \cos(\Delta\delta) \end{bmatrix} \qquad (9)$$

$$\text{R3=}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Delta\varphi) & -\sin(\Delta\varphi) \\ 0 & \sin(\Delta\varphi\,) & \cos(\Delta\varphi\,) \end{bmatrix} \qquad (10)$$

4. The fitness error is calculated between the target cloud with the parameters applied and the model cloud.
5. The steps 1-4 are repeated until the fitness error in equation (11) is lower than the minimum error specified, or the numbers of iterations reaches its maximum previously specified. In (11) *N* stands for the number of points of the clouds, m_i is the point of the model cloud. f represents the fitness error derived from Eq.6 i.e. the minimum square comparison between the first face model retrieved by the sensor *mi*, and the last (*ti*). The rotation and translation parameters are represented by the transformation matrix *Mt*.

$$f(\mathrm{R,T}) = \frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}}\|\mathrm{m_i} - \mathrm{Mt(t_i)}\|^2 \qquad (11)$$

The ICP algorithm is applied in order to obtain the transformation matrix Mt determining the new coordinates of the target cloud, as shown in (7), corresponding to the rotation and translation of the original matrix.

From ICP, the matrix Mt (3x3) is obtained. It is from this matrix from where the Euler's rotation angles are obtained according to the equations (12) (13) and (14). Translation vector effect is eliminated by using the centroid of the point of clouds corresponding to the face obtained.

$$\Delta\theta = sin^{-1}(R_{2,1}) \qquad (12)$$

$$\Delta\delta = tan^{-1}\left(\frac{-R_{3,1}}{R_{1,1}}\right) \qquad (13)$$

$$\Delta\varphi = tan^{-1}\left(\frac{-R_{2,3}}{R_{2,2}}\right) \qquad (14)$$

Where $\Delta\theta, \Delta\delta$ and $\Delta\varphi$ are the rotation angles for Pitch, Roll and Yaw respectively between the 2 analysed clouds.

ICP needs to be properly configured in order to deliver reliable results. After a variety of tests, it was observed that the slight variations of the Viola-Jones algorithm when delivering the rectangle containing the face (therefore the cloud of points of the face) altered significantly the results obtained between measurements. In order to avoid these errors, a constant number of points is determined for the two clouds to be analysed by ICP. This is done by adjusting the window of the last face detected to that of the first frame which is the reference face. By readjusting the obtained window, the resulting cloud has the same dimensions as the reference cloud and the estimation of the ICP algorithm is adequate. The ICP algorithm will compare the reference cloud as seen in (Fig. 4-8 left) using a reference system to determine the rotation angles (Fig. 4-9 centre) with the current cloud (Fig. 4-9 right)

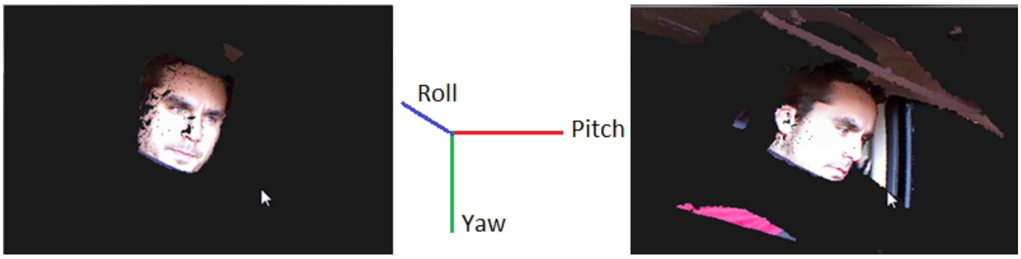


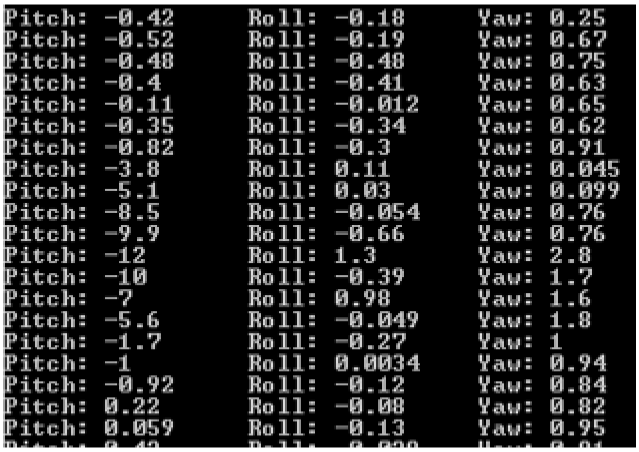
Figure 4-8: Reference cloud to the left, target cloud to the right

4.5.1 Configuring the ICP

The ICP algorithm was configured so that the results were accurate and fast. The number of iterations was set to 50 and the minimum error to that of the size of the radius in the Voxel Grid algorithm (1.0cm). The type of data that ICP would be working with were coloured point clouds so the right filter had to be built to work with this additional field. The modification of these parameters would change the resulting matrix transformation. By choosing these parameters, the results were stable up to one degree with static clouds i.e. loaded two pcd files and compared them. The error increased slightly, up to 4º, when using the frames captured by the Kinect sensor bar. Increasing the number of iterations did not deliver more accurate results and increased proportionally the time of execution. Due to the previous filtering process, the minimum error was that of the distance between points of the filtered cloud. Late tests showed that the dramatic reduction of the iterations (lower than 5) would deliver results that may vary significantly from test to test not only in the accuracy of the rotation angles but in the orientation of the resulting cloud sometimes delivering a result that was 180º rotated around the yaw axis.

4.5.2 **Results of the ICP**

Once ICP has been configured correctly, 2 clouds were used as input for the algorithm. The first cloud is the reference cloud and the second is the target cloud (the latest captured). Both are obtained from the driver's face and are down-sampled as explained in the earlier sections. Then, the algorithm will estimate the transformation matrix between the reference and the target cloud. After a determined number of iterations the transformation matrix is obtained. From this transformation matrix, containing rotation and translation, the matrix containing the 3 rotation angles (pitch, roll and yaw) is extracted. Then, the 3 angles are obtained by solving basic trigonometric equations previously explained. For the ease of comparison and interpretation, the results are converted to degrees as they are originally in radians. This makes the comparison easier with other devices like the IMU used as a ground truth. One of the early representations of the results was through a console window where the data would be printed to the screen as seen in Fig. 5-3. This output made possible the first evaluation of the obtained results.



```
Pitch: -0.42      Roll: -0.18      Yaw: 0.25
Pitch: -0.52      Roll: -0.19      Yaw: 0.67
Pitch: -0.48      Roll: -0.48      Yaw: 0.75
Pitch: -0.4       Roll: -0.41      Yaw: 0.63
Pitch: -0.11      Roll: -0.012     Yaw: 0.65
Pitch: -0.35      Roll: -0.34      Yaw: 0.62
Pitch: -0.82      Roll: -0.3       Yaw: 0.91
Pitch: -3.8       Roll: 0.11       Yaw: 0.045
Pitch: -5.1       Roll: 0.03       Yaw: 0.099
Pitch: -8.5       Roll: -0.054     Yaw: 0.76
Pitch: -9.9       Roll: -0.66      Yaw: 0.76
Pitch: -12        Roll: 1.3        Yaw: 2.8
Pitch: -10        Roll: -0.39      Yaw: 1.7
Pitch: -7         Roll: 0.98       Yaw: 1.6
Pitch: -5.6       Roll: -0.049     Yaw: 1.8
Pitch: -1.7       Roll: -0.27      Yaw: 1
Pitch: -1         Roll: 0.0034     Yaw: 0.94
Pitch: -0.92      Roll: -0.12      Yaw: 0.84
Pitch: 0.22       Roll: -0.08      Yaw: 0.82
Pitch: 0.059      Roll: -0.13      Yaw: 0.95
```

Figure 4-9: Example of the console output with the rotation angles

4.5.3 Interpreting Results

In order to know what number was representing what axis, different movements of the head were performed one axis at a time. This resulted in a drastic change for one of the 3 numbers obtained. Although the rest of the axis changed too, this were so small that were considered as error (changes under 1 degree). With the angles obtained and identified, the next step was to define when a driver was falling asleep or being distracted. The 3 rotation angles are just a measure obtained from the transformation matrix between the 2 faces. But the angles by themselves do not tell anything regarding the behavior of the driver. A proper interpretation has to be done in order to understand what rotation axis is can be used for what. In this case, the pitch angle is going to be used for one misbehavior and the yaw angle for another. These misbehaviors were defined differently. When the driver is looking to the sides, it was considered a distraction and when the driver was keeping its head down for a determined time, it was considered as falling asleep. It is worth noticing that the driver will perform different movements with the head according to the different scenarios (motorway or urban) and the adjustment of the thresholds that define a misbehavior will be necessary. In the circumstance that no face can be detected, the ICP algorithm will not be executed and no results will be delivered.

The next figure shows the block diagram (Fig.4-10) of the application. It starts with the capture of the scene, then the filtering and facial detection, the down-sampling and the application of the ICP algorithm to find the rotation angles that are later interpreted as misbehaviors when it corresponds.

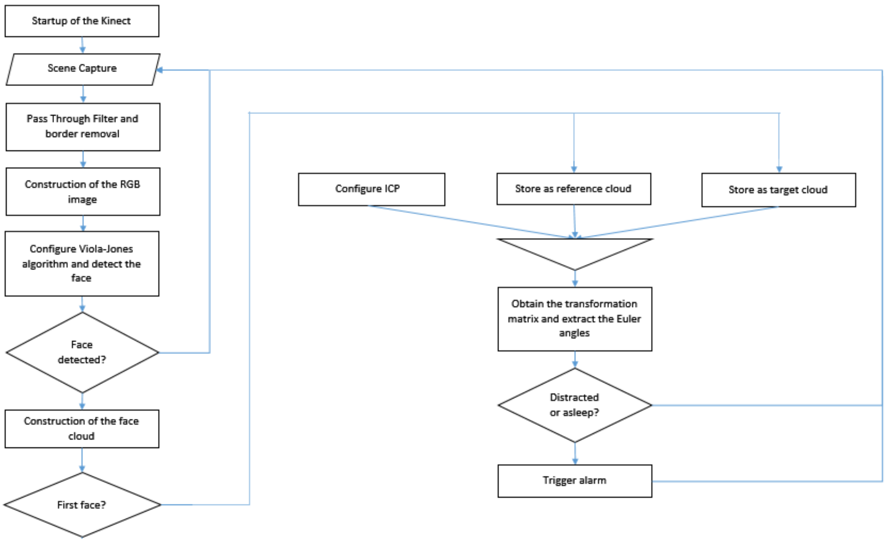

Startup of the Kinect
Scene Capture
Pass Through Filter and border removal
Construction of the RGB image
Configure Viola-Jones algorithm and detect the face
Face detected?
Construction of the face cloud
First face?
Configure ICP
Store as reference cloud
Store as target cloud
Obtain the transformation matrix and extract the Euler angles
Distracted or asleep?
Trigger alarm

Figure 4-10: Diagram of the whole application

CHAPTER 5. **Ocular Analysis**

5.1 Introduction

The head pose estimation of the driver can deliver a good representation on what the driver is doing; looking at the road, falling asleep and other misbehaviours. It has been described previously how to achieve this task in the previous chapters. But this estimation can only deliver so much information regarding where the driver is focusing its attention at a determined moment. Therefore, an additional tool had to be done in order to include additional information of the driver's behaviour. The additional tools should complement the head pose estimator by delivering results regarding the gaze orientation and the drowsiness of the driver.

In this chapter, a gaze orientation approach is described that has the intention to support the head pose estimation. This addition will allow the distinction of three regions (left, centre and right) of the pupil's position. By combining the gaze orientation and the head pose estimation, a more accurate guess of what the driver is doing can be achieved. It will be explained how the estimation of the gaze is done by calculating the gravity centre of the pupil. The definition of the 3 regions is described too as part of the interpretation of the numbers obtained from the gaze orientation algorithm. Following this section, the drowsiness detector will be explained in detail. This tool will determine if the driver is falling asleep through the analysis of the time that the eyes are closed during an interval of time.

5.2 Calculation of the Position of the Pupil

The position of the pupil will determine the direction where the driver is looking at. In order to do this, a cascade classifier was used in order to detect one eye. The classifier would return a region of interest from the colour image that corresponds to the eye. It is from this region that the pupil is searched through a series of image processing algorithms and from where its position in relation with the boundaries will be calculated. This position will be analyzed in order to know if the driver is looking to the left, centre or right.

The process of estimating the gaze starts with the enlargement of the picture of the face, specifically setting a ROI in the upper half as it is the only part that is of interest regarding the gaze orientation. As seen in Fig. 5-1, the face is detected from the captured frame and the ROI that constitutes it will be enlarged.

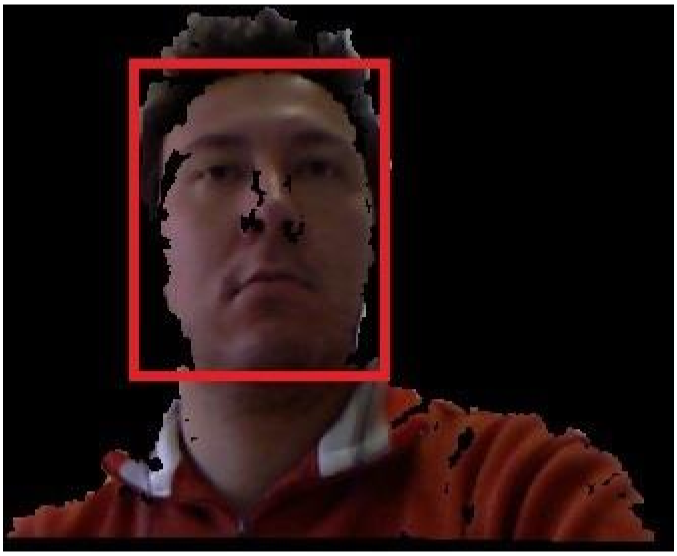


Figure 5-1: Face detected from the image extracted of the cloud

The decision to focus on the upper half of the image is based on the facial proportions studied by different drawing techniques explaining that the eyes are located at half of the face. This is not the only source to be taken into account but also the experimental results from the many variations obtained by the cascade classifier regarding the facial detection and the position of the eyes in the detected rectangle containing the face. This reduction of the searching space allows a more accurate and fast search of the eyes.

The image was enlarged using the bilinear interpolation method as it delivered the best results compared to the other interpolation methods. As the image was enlarged using the linear interpolation, the quality of the image was not large enough to corrupt the roundness of the pupil (artifacts) as some of the other interpolation methods did. It was also the fastest method compared to others that delivered also good results in term of image quality. In Figure 5-2 the results of the interpolation methods can be seen. From left to right the methods are: Nearest neighbor, bilinear, pixel area relation, bicubic over 4x4 pixel neighborhood and Lanczos over 8x8 pixel neighborhood. The second one (Fig. 5-2 b) was chosen.

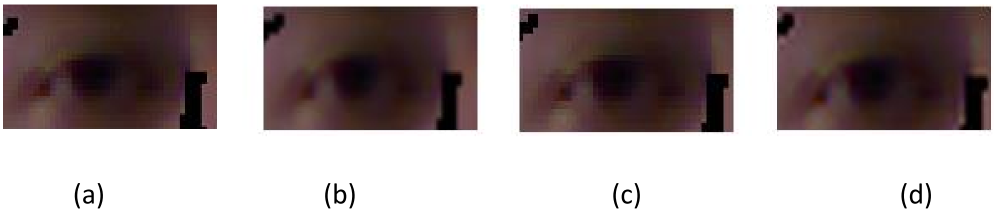


Figure 5-2: Different interpolations available

Once the image was enlarged, the search for an eye was performed. This required the adjustment of the parameters inside the cascade classifier so that the eye could be found in an image 4 times larger. It has to be taken into account the fact that this gaze orientation method is not meant to be used only with a frontal face picture but it should also work with the rotation of the head. But when the head is rotated, as when rotating to see the right mirror in the car, one part of the face is occluded from the camera and the occluded half of the face is darker than the frontal half. Lighting conditions are one problem, another problem is the partial occlusion of certain regions of the face like the eye being occluded by the nose. It was because of these problems that the search for one eye and not the pair of eyes was chosen. The model of the pair of eyes requires the clear presence of the two eyes in order to return a ROI when both are found but when the head is rotated, the cascade classifier using this model would not find the pair due to the occlusion and variable lighting conditions in each half. When searching for one eye, it does not matter if the detected eye is on the left or right as it is assumed that both pupils are doing the same movement. The result of the eye detection can be seen in Fig. 5-3 where the ROI of the detected eye is marked in green.

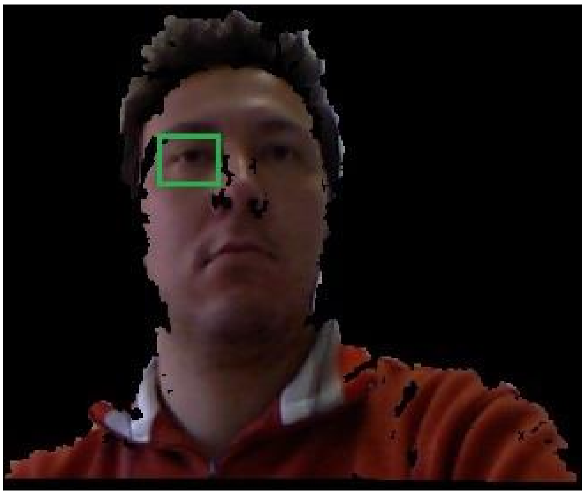


Figure 5-3: Detection of the eye on the colour image

With the eye detected, the image is then subject to different processes. The goal is to detect the pupil which is the darkest object in the region of interest of the eye. To do this, the first step is to convert the colour image to a grayscale image. The grayscale image is then equalized by analyzing its histogram. This is necessary because of the different lighting conditions that can be present when the head is rotated. With the image equalized, a conversion to black and white is performed by establishing a threshold that will allow the elimination of all the bright objects and leave only the darkest, like the pupil. This results in a black and white image that contains one blob that belongs to the pupil. The resulting image is again resized four times its original size.

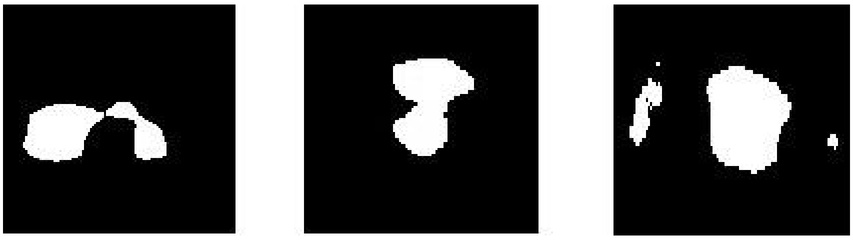
There are different methods to obtain the position of one object in relation to another used as a reference. Because of the irregularities, ease of implementation and robustness, the method of the gravity centre was chosen. This method consist in calculating the geometrical centre of a blob inside a region of interest and is obtained by calculating the average of the coordinate values of the pixels that compose the blob in this case, the pupil. The first step is to sweep the image and take into account only the pixels that belong to the blob. Then, the average of the *x* axis coordinates, using the equation below (Eq. 15), will return the geometrical centre of the blob that is the same as the position of the pupil in reference to the bounding box of the eye.

$$Position\ in\ X = \sum_{i=0}^{n} \frac{Cx_i}{p} \qquad (15)$$

Where n is the amount of points of the eye's ROI. Cx_i is the value of the *x* coordinate of the point that belongs to the blob. p is the number of points that belong to the blob.

The result is then established as a percentage where 0 is one extreme of the eye and 100 the other extreme. Therefore, when the driver is looking to the front, the returning value should be 50 as it is at half the position of the whole width of the bounding box of the eye.

Due to the nature of the sensor and the classifier, different factors have to be taken into account when doing this process. One of this factors is the dark region that may appear in the colour image due to sporadic misinterpretations of both the colour and IR sensor. This, followed by the pixels that could be considered as dark enough to be part of the pupil, can alter the estimation of the position of the pupil. To mitigate the effects of this problems (noise), opening and closing processes were applied to the black and white image. This allowed the preservation of the biggest dark object supposedly being the pupil and the removal of objects that should not be there like the noise from the sensor or dark spots in the eyeball just to mention some. The image below (Fig. 5-4) shows the variety of resulting objects after the previous image processing where the blob is divided (Fig.5-4 a), irregular (Fig.5-4 b) and with noise due to other regions with similar chromatic characteristics (Fig. 5-4 c).



(a) **(b)** **(c)**

Figure 5-4: Example of the binary conversion of the pupil's image

5.3 Establishing Regions and Interpretations

The algorithm will return the horizontal position of the pupil inside the eye with a number between 0 and 100. But this number means nothing for the end user or for the next applications that will take benefit from this. Although the raw data could be useful for some extra processing, it is more understandable if this numbers are translated into a representation that is more explicit. One of his approaches is to determine if the driver is looking to the left, right or centre based on the interpretation of one number. To do this, 3 regions were set depending on the values obtained from the gaze orientation algorithm. Therefore, after a variety of tests, the regions that established that the user was looking to the left, centre or right were set. The objective is to make it clearer for the end user or next algorithm where the user is looking. Also, because of the noisy and irregular nature of the gravity centre, it was decided that this would be the best approach to deliver the results. This irregularities were diminished by implementing a filter (Eq. 16) that would compare the current result with the previous one and replace it with the previous one if a certain criteria was achieved. An irregularity is considered to be a small fluctuation in the value (noise) or a drastic change mostly due to calculation errors from the cascade filter.

$$G = \begin{cases} G_{t0}, & Nt > ||d|| < Ct \\ G_{t-1}, & Ct \geq ||d|| \leq Nt \end{cases} \qquad (16)$$

G is the gaze orientation value, $G_{t0}$is the gaze orientation value at the current frame and G_{t-1} the value of the previous frame. Nt and Ct are the noise and calculation thresholds respectively. d is the difference between the current and the previous frame regarding the gaze orientation.

5.4 Challenging situations and establishing limitations

The Kinect sensor bar can obtain the 3D structure of an object but also the 2D information. It is from the 2D information that the eyes were detected and the gaze estimated as the 3D information available (that made the head pose estimation available) is not enough for the gaze orientation. The colour information obtained from the 3D structure makes the use of classifiers possible up to a certain point. The low resolution camera of the Kinect sensor bar (640x480) makes the detection of the eyes harder than with other devices with higher resolutions. Nevertheless, through a variety of previously described algorithms, the eye can be detected and the position of the pupil be determined. It is worth noticing that this sensor is adequate for analyzing larger objects than the eyes and at further distances than those that apply for this experiment.

5.5 Results of the gaze estimation

The algorithm was tested during different lighting conditions where the eye was detected and the gravity centre of the pupil was determined with success. Although not as robust as the facial detection, due to the larger dimensions of the face compared to the eye, it was still possible to estimate to which of the 3 regions the driver was looking. As a support tool for the head pose estimation, it will allow to perform a deeper analysis on the behavior of the driver by taking into account the gaze and the head pose. This means that the driver no longer has to be marked, defined or tagged as just looking to the right due to the head pose but also looking to the front, left or right. Figure 5-5 shows the result with the head oriented to the front looking to both sides and to the centre.

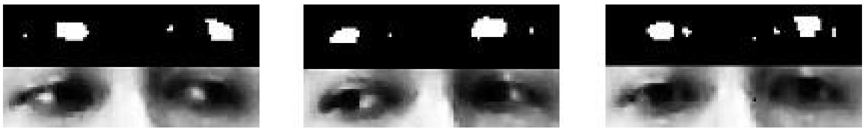


Figure 5-5: Binary image on top, grayscale image at the bottom

This solution, compared to others, does not require a previous calibration method like asking the driver to look at determined pre-established marks inside the car. It will work immediately and without the need of any previous initial conditions. The markings (Fig. 5-6) could be used later for understanding where the driver is looking at. For example, if the driver's gaze is oriented to the lateral mirrors, gauges, road and more. Also to divide the visible area, the front, into regions to discretize and analyze the behavior of the driver regarding this areas.

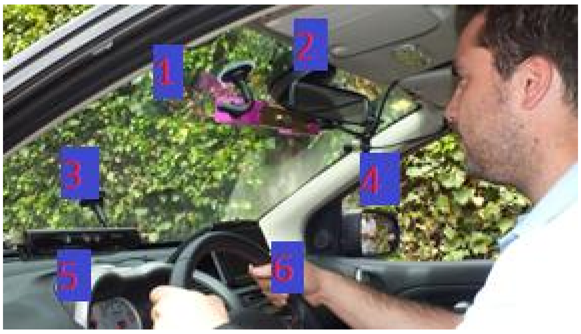
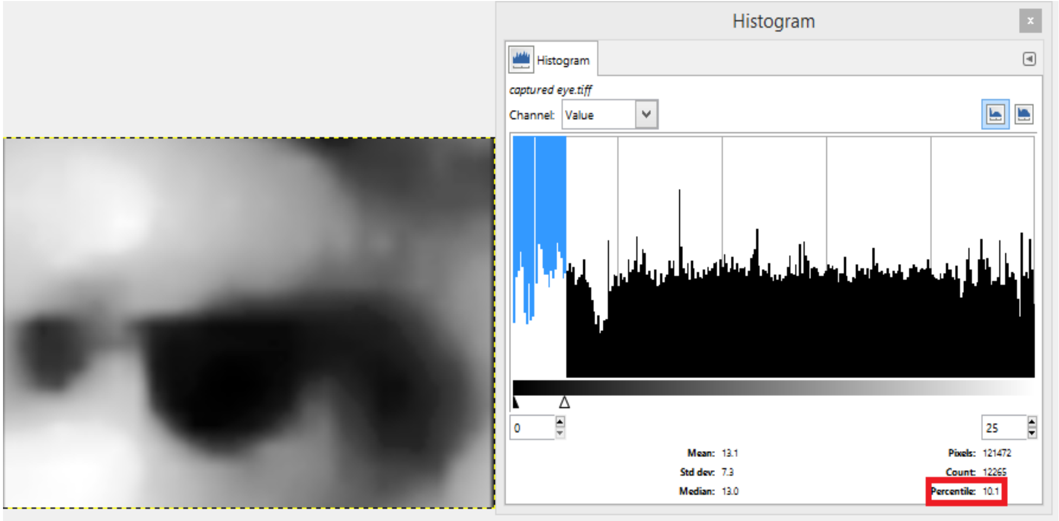


Figure 5-6: Proposal of the discretization of the visual area for the driver

5.6 Gaze estimation with Blob Analysis

Another approach taken to solve the problem of the gaze orientation with a low resolution camera was the analysis of the objects found through image processing. By performing different algorithms on the captured frame, the position of the object that belonged to the pupil could be analyzed and used for the computation of the gaze orientation in a similar way to the one described earlier.

This approach uses the equalization of the histogram to enhance brighter regions and make the image clearer followed by the conversion to a binary image through the calculation of a dynamic threshold that corresponds to a value in the percentile of the histogram. Finally, the blob analysis with the ellipse that fit is performed. An illustration of this can be seen in Fig. 5-7. To the left is the ROI of the eye and to the right the histogram previously equalized. The region in blue, the extreme left of the histogram, marks what belongs to the 10th percentile of the whole histogram. This value can be seen at the bottom right corner inside the red rectangle. The value, in grayscale, that belongs to the 10th percentile will be applied to the ROI of the eye. Once applied, the result is an image with the darkest region only as seen in Fig. 5-8.



captured eye.tiff

Channel: Value

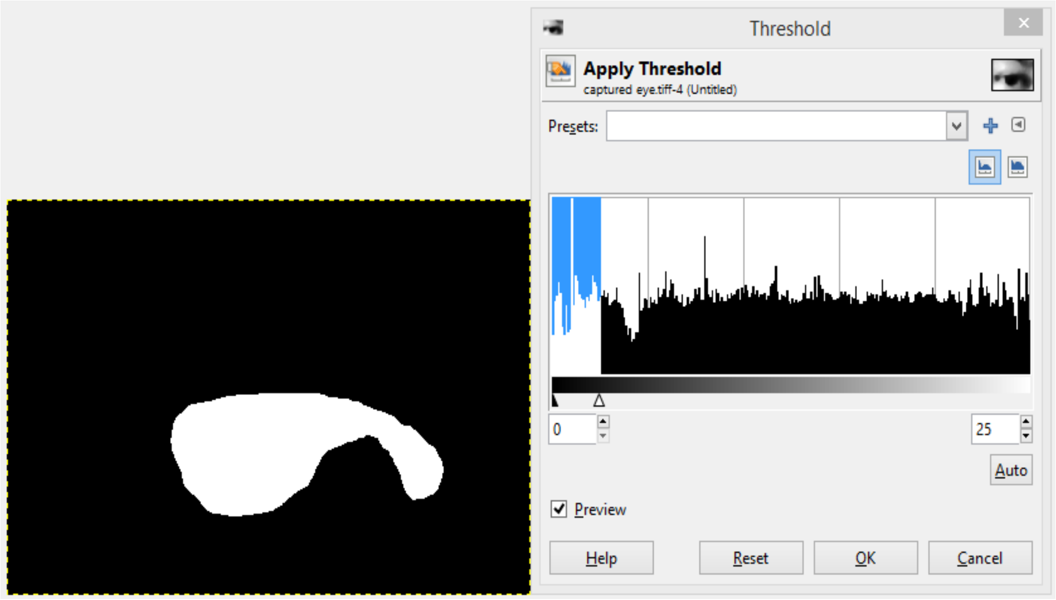
0

25

Mean: 13.1
Std dev: 7.3
Median: 13.0

Pixels: 121472
Count: 12265
Percentile: 10.1

Figure 5-7: 10th percentile of the histogram



captured eye.tiff-4 (Untitled)

Presets:

0

25

☑ Preview

Figure 5-8: Application of the threshold according to the 10th percentile

The position of the pupil was estimated by using the gravity centre of the bounding box that can be seen in colour blue in the pictures above. This box will contain inside the ellipse that fits the desired blob. Once the gravity centre of the bounding box is obtained, its position related with the image is estimated by the equation below (Eq. 17):

$$Position = \frac{GCB}{W} \qquad (17)$$

Where *GCB* is the Gravity centre of the box and *W* is the width of the image.

This equation will return a value between 0 and 1 depending on the position of the blob (the pupil). If the pupil is located at the extreme left of the image, the result will be closer to 0. On the contrary, if it is at the other end, the result will be closer to 1. This result was adjusted by taking into account the rotation of the head because although the eyes were still oriented to the centre, when the head was rotating to the left and right, the gravity centre was changing too in proportion to the yaw angle.

The table below shows some examples of the gravity centre of the eye when only the head was rotating. Instead of being constant, the estimated position was being affected by a proportional factor related to the yaw angle. Therefore, a correction value was introduced to compensate for this errors that can be found in the equation Eq. 18.

TABLE II:

RELATIVE POSITION OF THE PUPIL WHEN ROTATING THE HEAD

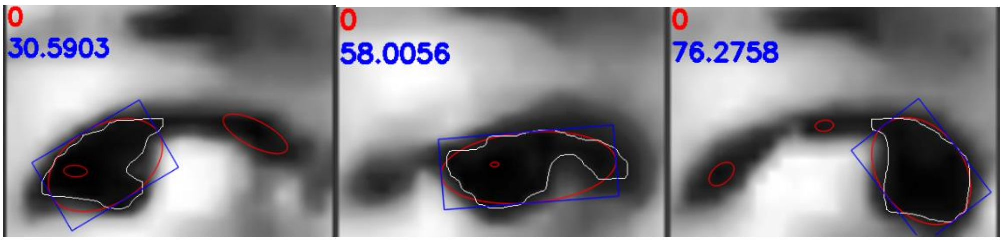
Yaw	-20º	-15º	0º	16º
Percentage	67	60	52	45

$$Corrected\,position = \frac{GCB}{W} + \frac{\varphi}{2} \qquad (18)$$

Where φ represents the rotation angle corresponding to the yaw obtained from the head pose estimation algorithm. This correction allowed a more robust estimation of the gaze orientation and helped with the changing lighting conditions. Because of the previous blob processing, this method offers improvements over other approaches because it will discard those objects that do not pass the geometrical requirements such as size and orientation. The rejection of such blobs improves the final result as they are not taken into account when calculating the position of the gravity centre.

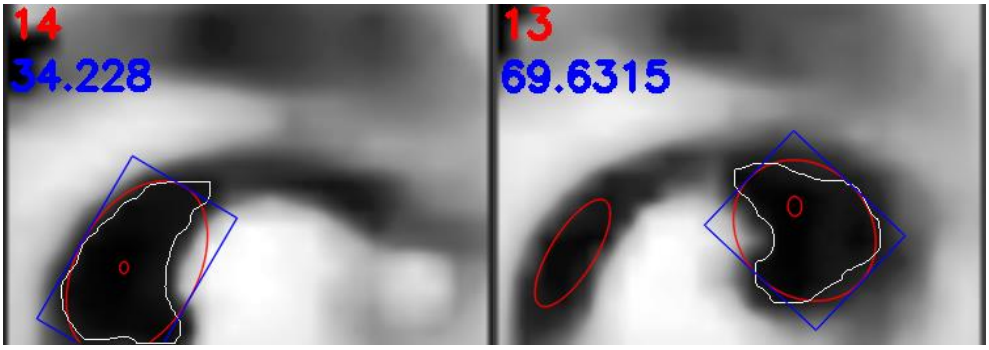
Figures 5-9, 5-10, 5-11, show the yaw value appears on the top left corner in red and below in blue the percentage of the gravity centre in reference to the width of the image. The borders of the resulting blob are drawn in white colour, the surrounding ellipse is drawn in red colour. In blue colour is the rectangle for the blob that is considered after the analysis of the blob's size and the width / height ratio considerations.

Figure 5-12 shows the result of the gaze estimation while moving the eyes to the left and right. The first part is flat as the eyes were not moving. The second part, where the orange line is moving up, corresponds to the eyes moving to the left and then going back to the centre. Finally, the third part, shows the result when the eyes are moving to the right and returning to the centre.



(a) **(b)** **(c)**

Figure 5-9: Estimation with yaw = 0



(a) **(b)**

Figure 5-10: Estimation with positive yaw

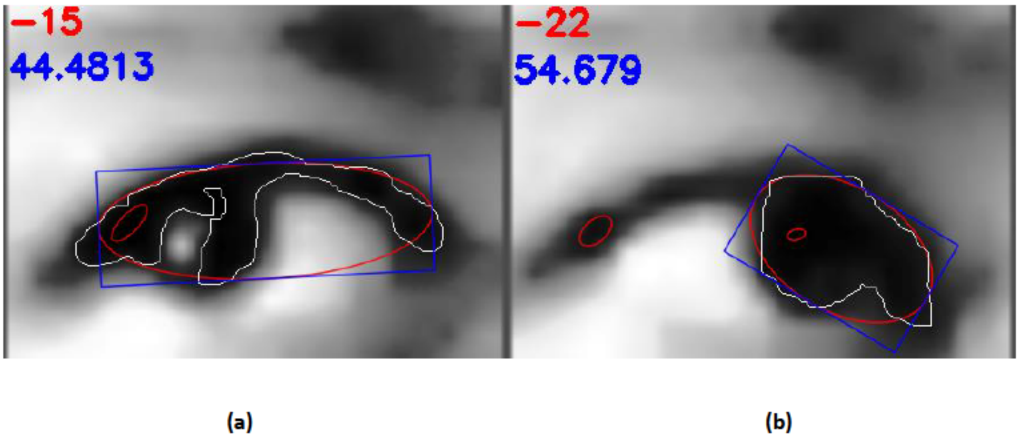


Figure 5-11: Estimation with negative yaw

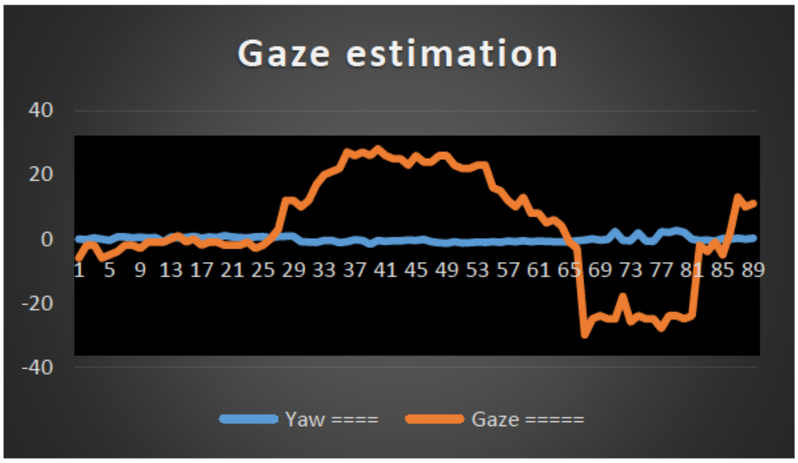


Figure 5-12: Gaze estimation and yaw

5.7 Drowsiness and PERCLOS

One of the parameters for detecting if the driver is falling asleep is the percent eye closure (PERCLOS) [47]. It is defined as the fraction of time that the eyes are closed at least 80% during a determined moment. When the eye is fully open during the specified time of measurement, PERCLOS is 100% and when closed during all the specified time it is 0%. PERCLOS has a high correlation with fatigue and drowsiness and has been used in different approaches to estimate if the driver is falling asleep or not as in [48] [49]. An overall evaluation is shown in [50] where 16 volunteers were driving during successive sessions with breaks in between. They recorded different output variables like PERCLOS, electroencephalogram (EEG) and Karolinska sleepiness scale (KSS) (Table III). It is shown in this study the high correlation between KSS and PERCLOS and thus the study of one of this parameters can be used to explain the other. They also state that the measurement of fatigue is a difficult task with large errors if it is expected to estimate the fatigue on an individual level and with a high temporal resolution. This means that if the eye is closed for more than a determined period of time, the driver would be falling asleep and therefore, an alarm must be triggered to prevent an accident.

TABLE III:

KAROLINSKA SLEEPINESS SCALE

1	**Extremely alert**
2	Very alert
3	Alert
4	Rather alert
5	Neither alert nor sleepy
6	Some sighs of sleepiness
7	Sleepy, but no difficulty remaining awake
8	Sleepy, some effort to keep alert
9	Extremely sleepy, fighting sleep

Thanks to the previous analysis of the face, the PERCLOS can be calculated. The whole image is scanned and with the face detected, the search for the eyes is performed with the correct classifier in a specific ROI. The classifier in charge of the search of the eye will return a value regarding the result of the search. If the eye is detected in the image, it means that certain requirements were achieved. One of this is the contrast between the pupil and the eyeball and in order for this to happen, the eye must be open. In certain cases, a closed eye was considered a positive candidate by the classifier. In order to avoid this situation, a fast analysis must be done in the region that belongs to the positive candidate (i.e. the closed eye). If the eye is open, the eyeball must be visible and thus a white region must appear in the image. The amount of white colour in the region of the eye will determine if the eye is closed or not. Therefore, by using the classifier to detect the eye and analyzing the amount of white, it can be established if the eye was open or not at the analyzed frame. This approach faced several problems with the illumination changes. Also, the colour of the eye ball was not always white because the driver could have tired eyes that make the eye ball more red due to the presence of blood vessels that are not visible when the person is well rested and in ideal conditions. Still, different equalizations were tried with no success and it proved to work only in very specific situations and it was not robust enough. It also depended on the person itself that was analyzed as the opening of the eye would determine how much white could be detected in the image. This means that a person with eyes that have a smaller gap between the lids than the average would not show much white colour when the image analysis was done. This gave the clue for the definitive approach that was implemented in order to find if the eye is closed or not.

Images like the one shown in Fig. 5-13 should be analyzed and through a series of algorithms it should be identified as that with closed eyes.

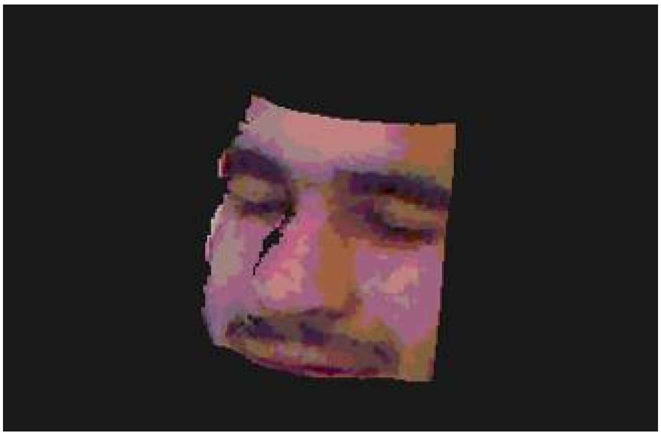


Figure 5-13: Capture face with the eyes closed

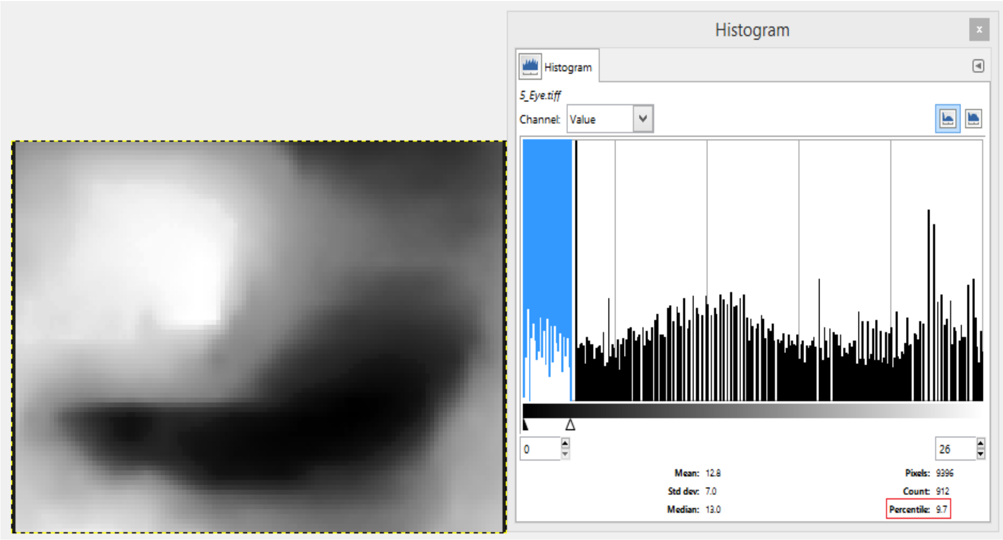
5.7.1 Blob analysis

Instead of detecting the white of the eyeball to detect if the eye was open, it was decided to detect if the eye was closed by analyzing the lids when they are closed. This approach takes into account the fact that a dark horizontal region is predominant when the two lids touch. The region is present only when the eyes are closed as when they are open, the white colour of the eyeball is predominant. As opposed to the analysis of the white colour, the black region will always be black. This was found after the analysis of the videos taken inside the laboratory but also inside the vehicle when the tests where done outdoors. To guarantee that the darkest region was found, the term “darkest region” had to be translated into numbers and equations in the program. Because of the illumination changes, the ROI in the eye would vary slightly from one test to another.

To solve this problem, 2 main process were done. The first is a histogram equalization, this allows the correction of the illumination channel from the problems when a certain region of the image is too bright and makes the rest of the image too dark to distinguish anything. By equalizing the histogram, the darkest regions get brighter and the detection of borders will return more results compared to the non-equalized scenario.

The second is the threshold determination for the conversion of the grayscale image to black and white. This value was calculated not by using a specific value for every scenario but a variable that will be in function of the illumination. As stated before, the frame's histogram is equalized but this is still affected by the lighting changes. Therefore, a percentile analysis was done in order determine the darkest region of the image. If this is too bright, the darkest region will take more pixels from the lower part of the histogram and if the image is too dark, it will take less pixels.

The analysis of the obtained regions by varying the percentile chosen showed that the best results were when the threshold was equal to the 10th percentile of the histogram as it can be seen in Fig. 5-14.



Channel: Value

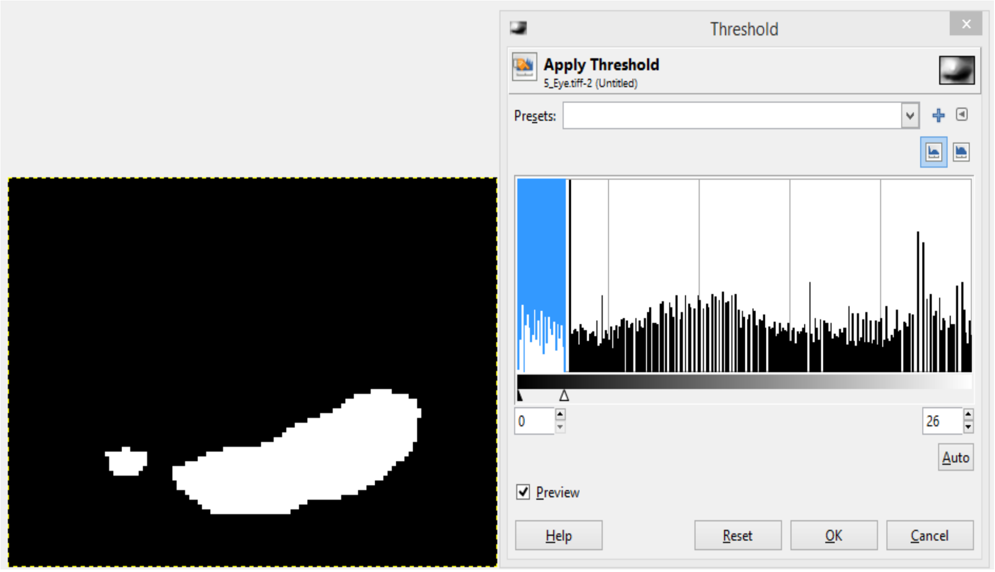
0

26

Mean: 12.8
Std dev: 7.0
Median: 13.0
Pixels: 9396
Count: 912
Percentile: 9.7

Figure 5-14: Histogram to the left, in blue the region belonging to the 10th percentile

This value, when applied to the threshold process, returned an image that contained only the ROI from between the lids and ignored most of the rest of the image as seen in Fig. 5-15.



0

26

☑ Preview

Figure 5-15: Application of the threshold keeping only the 10th lower percentile

The result is a black and white image where the blob corresponding to the two lids closed is predominant. Other blobs are from dark regions from the saturation of the IR sensor or spots in the skin just to mention some. By performing an opening operation on the image with the blobs, most of the undesired blobs were removed. It is from this resulting image that the blob analysis was performed. One of the methods to remove undesired blobs was the size of it. Those blobs what were too small (1% of the image size) or too big (70% of the image size) were discarded. After this processes, a model fit analysis was performed, specifically, an ellipse fitting model. By studying the ellipse model that fit to the blob, it could be determined if it was from an opening eye or a closed eye because when the eye is open, the ellipse would be in a vertical position and when closed the ellipse would be horizontal. The figure 5-16 shows 3 particular cases of the resulting blobs, bounding rectangles and ellipses.

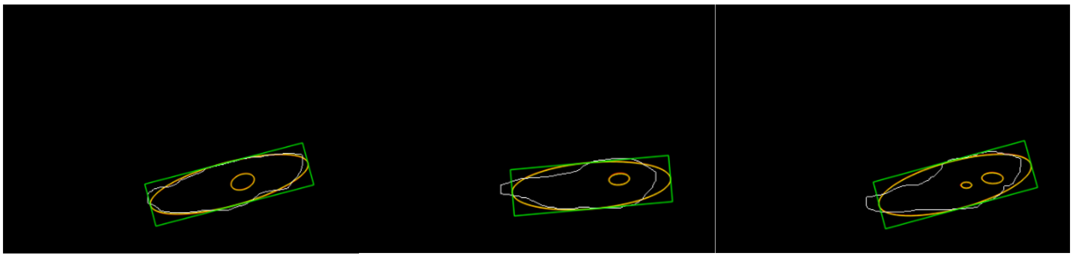


Figure 5-16: Borders detected and the ellipse surrounding the blob

The horizontal and vertical position were obtained from the comparison of the ratio between the width and height of the ellipse (Some mathematic models would return the two radius that are just the same as the width and height of the bounding box and thus the same analysis can be done). As the analysis of more than 510 frames proved, it is considered that the blob is horizontal when the width / height ratio is over 1.70. Figure 5-17 shows 3 successful detections of the eye being closed.

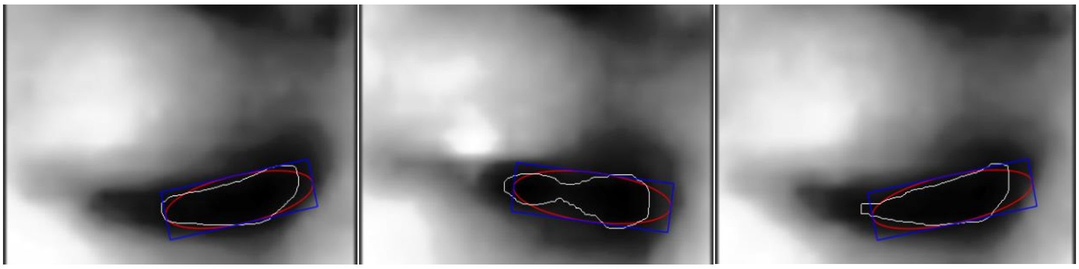


Figure 5-17: Superposition of the rectangle surrounding the blob, the ellipse and borders

The next step is to establish if a misbehavior is happening or not. As stated before, a misbehavior is defined as looking away from the region between the lateral mirrors (yaw angle) or when the head is oriented towards the ground (pitch angle). These two methods are solved with the resulting angles obtained from the transformation matrix that the ICP algorithm returns from the analysis of the 3D structure of the face and thus it is a task that belongs to the head pose estimator. Another misbehavior to be considered is what comes before the driver starts to nod its head and that is the PERCLOS parameter. The blinking pattern of the driver changes with its mood. When the driver is alert and in optimal conditions for driving, the blinking pattern will be different as if the driver was falling asleep and is tired. By using the definition of the PERCLOS parameter, it can be estimated if the driver is falling asleep. This means that the drowsiness on the driver can be detected by measuring the percentage of time that the eyes were closed during a certain interval.

5.7.2 Time based analysis

The function that detects if the eye is closed through the analysis of the closure of the eyes is responsible for this task. First, a frame is sent to the function to estimate if the present eye is closed or not. When the result is positive, i.e. the eye is closed, a counter is triggered. This counter has the goal of establishing during how many frames the eye was closed. This counter is reset after 1 second so that the number of frames that the eye was closed can be established for every interval. Due to the fact that the framerate will vary depending on the scene analyzed, it must be constantly calculated. This part is paramount because the function can only return if a frame contains a closed eye or not and the counter can only keep record of how many frames had the eye closed. In order to do a temporary analysis, it is necessary to calculate the number of frames per second and this is obtained through the correct calculation of the framerate in hertz. Once this is achieved, the number of frames where the eye was closed per second is obtained.

Another counter, a drowsiness counter, will record how many of this intervals had the eye closed for more than 60% of the time. The table below shows a theoretical explanation of the estimation of the drowsiness. If the frames per second calculated were 10Hz, it means that there will be 10 frames to be analyzed each second. For each frame, the result of the algorithm that calculates if the eye is closed is reflected in the second row of the table with a "+" when the eye is positively detected as closed and "-"when the eye is not closed.

TABLE IV:

EXAMPLE OF THE ANALYSIS OF A SET OF FRAMES

Frame No.	1	2	3	4	5	6	7	8	9	10
Result	-	+	+	+	-	+	+	+	-	+

For this particular example, the eye was closed for 70% of the time. This would mean that the drowsiness counter increases its value as it is above the 60% threshold. If this pattern persist for more than 3 seconds, the alarm will be triggered. The alarm will not go off until the percentage of the time was closed during the interval is lower than the threshold.

CHAPTER 6. Results

6.1 Introduction

The proposed solution estimates the head pose of the driver with a 3D structure based on the detection of the driver’s face. From here, after the application of filters and processing, the rotation angles are determined using the ICP algorithm. This algorithm calculates the transformation matrix that is composed of the rotation matrix and translation vector. From this composed matrix, the rotation angles are calculated. An IMU was used to confirm that the results were reliable. These results, although useful, do not explain the possible misbehaviors that the driver could do. Therefore, certain rules were set to explain what a misbehavior is and what rotation angle could determine it. Tables and figures complement the description of what the proposed solution can achieve.

So that the results of the algorithm could be better visualized and understood, a graphic representation of the pitch and yaw variations was done and will be explained in this chapter too. The roll angle was not represented due to the low frequency of variations compared to the other two rotation angles. This tool can be used not only to visualize the results of the head pose estimation but also, for debugging the modifications done to the code. Seeing a line that moves in a determined direction is more representative than seeing an angle.

Regarding modifications, some were done to improve the program; for example, enabling the storage of sequences that could be used later for “offline” study. With the changes to the solution, the program could take a whole scene captured and store it with all the information (2D and 3D) at a framerate of approximately 10Hz.The way this sequences are stored will be explained as it is not as simple as writing the cloud’s file to the hard drive.

6.2 Early tests

The first approach is usually not the best and will not be the way in which the final experiment will be carried out. Different approaches were performed before taking the definitive approach that was previously described. The decision to change the methodology and order of execution of algorithms is due to the fact that there is not a definitive pipeline or suggested method to follow for applications like this one as it was, until the decision was taken, something that was just being commented in small threads in a few forums in the internet. Below are some of the tested methods that were proposed and tested for the solution of the driving monitoring. These approaches were discarded due to different reasons that will be later explained.

6.2.1 RANSAC

This method approaches a cloud of points to a geometrical shape, by adjusting the parameters of the algorithm, the algorithm can return a match to a specific shape (sphere, plane, and cylinder). RANSAC [51] assumes that any cloud is defined as that which belongs to a parametrized geometrical model (inlier) and the points that do not belong to the model because they are too far away (outliers). The general steps of the algorithm are:

1. From the original data, a random subset is selected and is named “hypothetical inliers”.
2. A model (plane, sphere or cylinder) is going to be fitted to the set of hypothetical inliers.
3. The rest of the data is tested against the fitted model. Those points that fit the estimated model, are considered as part of the inlier set.
4. The estimated model is reasonably good if the number of points in the inliers is over the predefined threshold.
5. The parameters of the model are estimated again using all the identified inliers and stop.
6. Steps 1 to 5 are repeated a maximum number of times according to a specific parameter previously defined.

The parameters that can be configured for the RANSAC function are the geometrical shape, number of iterations and the maximum distance from the inlier. Fig. 6-1 (taken from the website of the developer) shows with a blue line the plane that fits the points in blue. In red are the outliers.

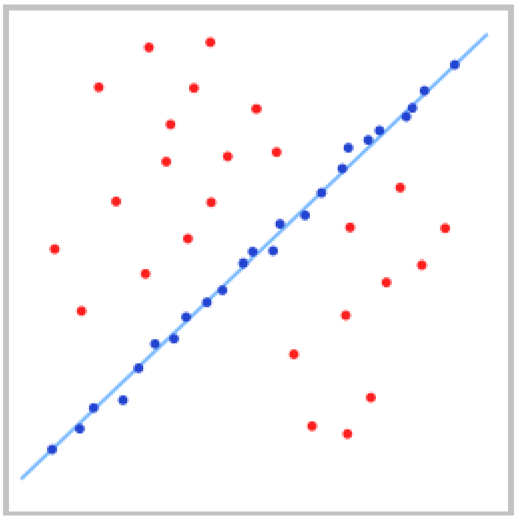


Figure 6-1: Example of the RANSAC line fit algorithm

The idea was to obtain a geometrical shape, a plane, from the cloud that represented the driver and calculate the normal to the surface. The plane would be parallel to the surface of the face and the normal vector to this surface would correspond to where the head was oriented. The figure 6-2 (taken from the website of the developer) shows in green the normal vector to the plane of the dots marked in yellow and red.

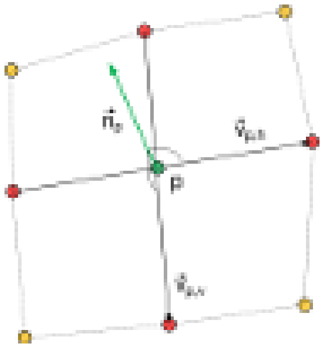


Figure 6-2: Normal vector to the surface

Although promising, it was discarded due to the large amount of false positives obtained when calculating the plane that fit to the driver’s cloud. If the driver was turning the head, the new plane was matched to the profile of the face and not the front as expected. Other shapes were tested too without success.

6.2.2 **Optical Flow**

Another approach to solve the head pose estimation problem was to use the 2D technique called Optical Flow. The idea was to obtain the orientation of the face by analysing the face and detecting important points from where the movement of the head could be estimated. The Lucas-Kanade method was used for detecting the important points (keypoints). This method is more robust against noise in the image but weak with regular surfaces as there are no points on where to fund the movement of the object. Therefore, it is important to do a previous processing step to enhance the details in the image so that more keypoints can be found which will give a better chance to get more accurate result from the optical flow algorithm. An example of optical flow applied to navigation can be seen in Fig. 6-3 (taken from the website hizook).

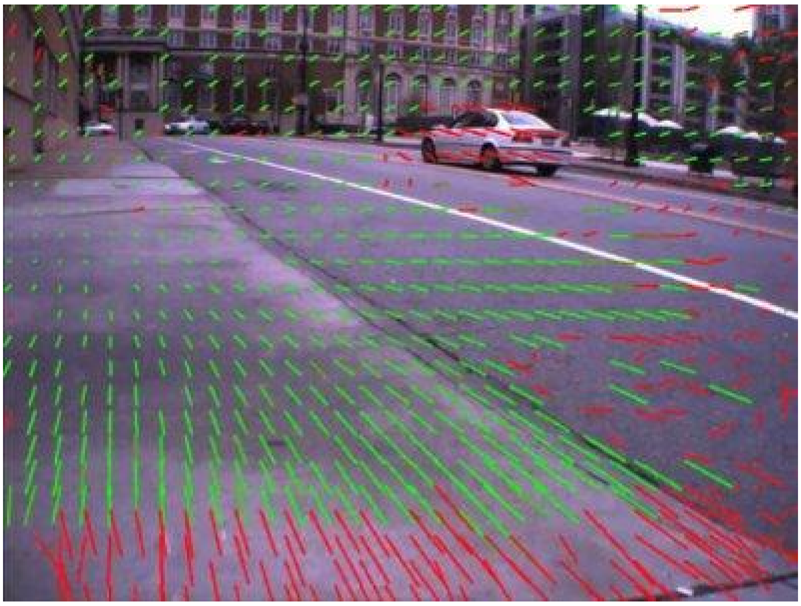


Figure 6-3: Coloured arrows show the optical flow result

This method was discarded due to the nature of the algorithm itself being iterative and useful for moving objects whereas the main objective is to get the position not the motion of the head. Also because at the moment of testing this method, the ICP approach (the one implemented in the final version) was showing promising results by using the 3D data built by the infrared projector which was the main advantage of this application compared to others that use only 2D information.

6.2.3 Clustering

The idea was to remove the part of the 2D analysis to search for the head and use only the cluster that would belong to the head. Once the cloud that represents the driver’s head is detected and isolated from the rest of the scene, the idea would be to use the ICP algorithm on these clouds. In the PCL libraries, the clustering method uses a Kd-Tree structure to group the points into clusters by using nearest neighbours.

The general steps for the Euclidean clustering are listed below (taken from the website of the developer of the libraries PCL [52]):

1. Create a Kd-tree representation for the input point cloud dataset *P*;
2. Set up an empty list of clusters *C*, and a queue of the points that need to be checked *Q*;
3. For every point $p_i \in P$, perform the following steps:
   - Add p_i to the current queue *Q*;
   - For every point $p_i \in Q$ do:
     - Search for the set P_k^i of point neighbours of p_i in a sphere with radius $r < d_{th}$;
     - For every neighbour $p_i^k \in P_i^k$, check if the point has already been processed, and if not add it to *Q*;

- When the list of all points in *Q* has been processed, add *Q* to the list of clusters *C*, and reset *Q* to an empty list

4. The algorithm terminates when all points $p_i \in P$ have been processed and are now part of the list of point clusters *C*

The figure 6-4 shows an example of the captured cloud where the subject is looking up but the algorithm did not segment correctly the cloud belonging to the head and the rest of the body due to the distance between points in the region of the neck. The following figures (6-5, 6-6 and 6-7) show additional examples of the subject looking to the front, to the left and to the right respectively.

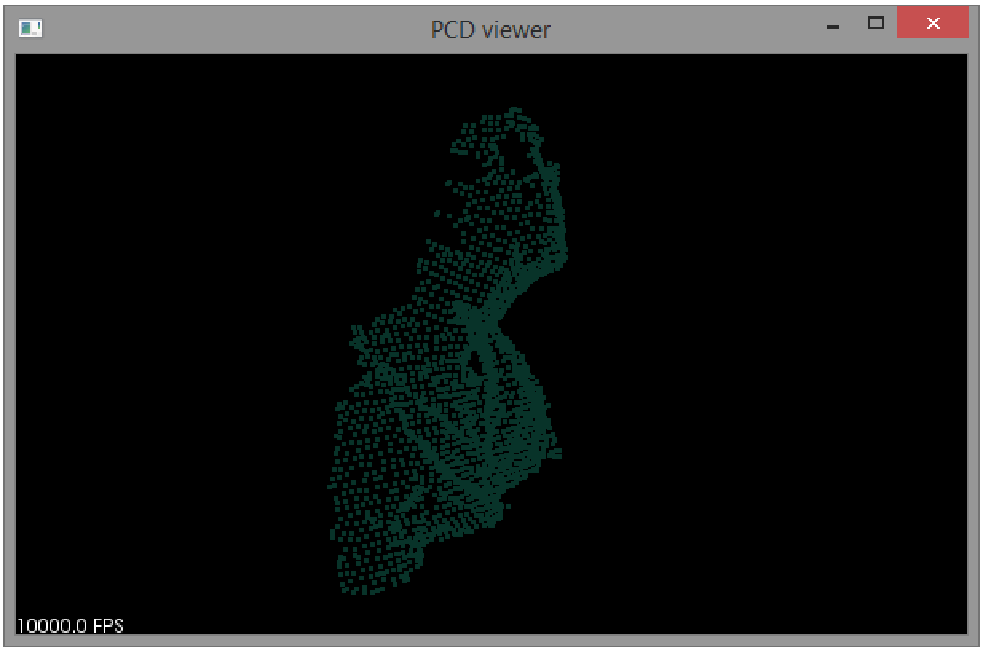


Figure 6-4: Subject looking up

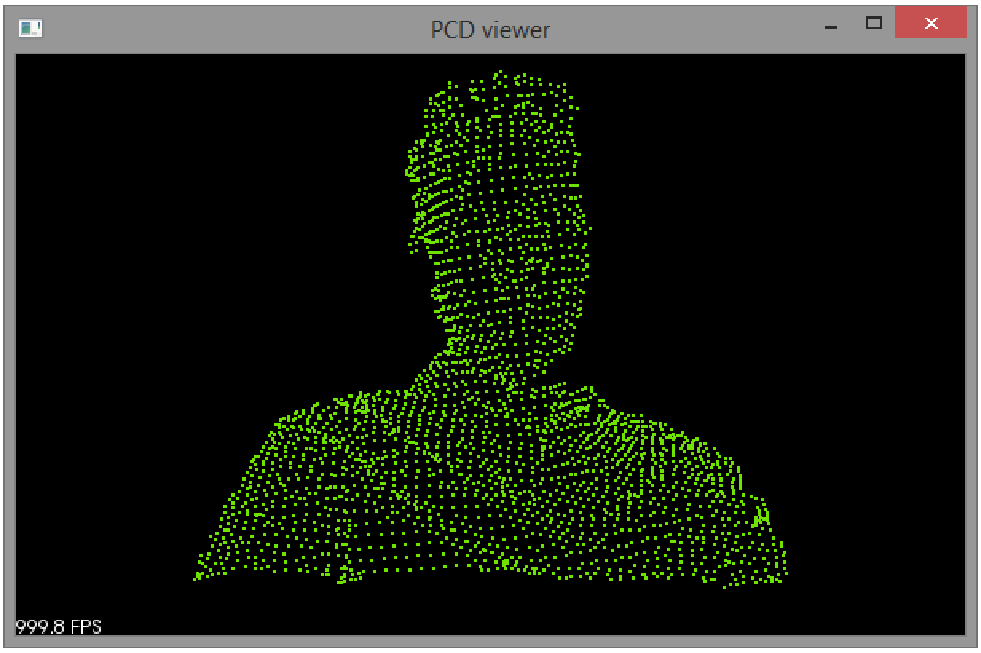


Figure 6-5: Subject looking to the front

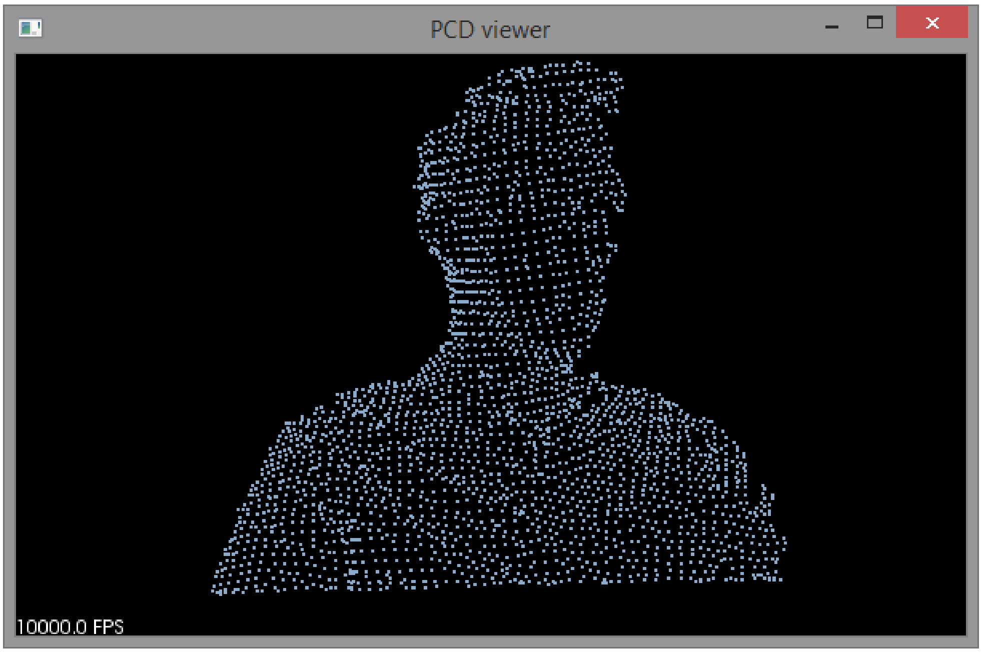


Figure 6-6: Subject looking to the left

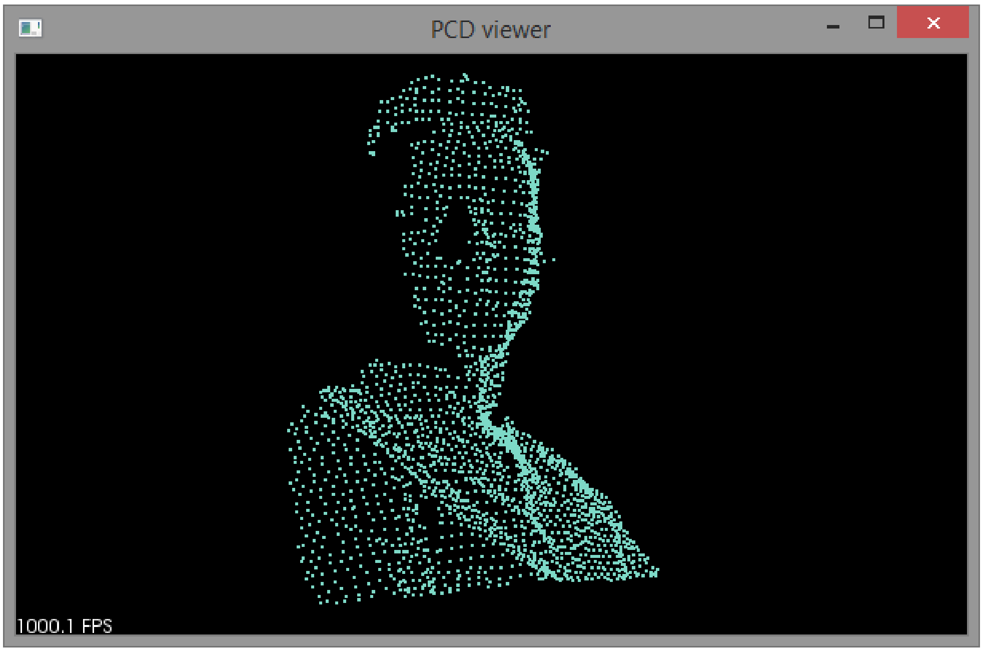


Figure 6-7: Subject looking to the right. Small gap in the neck

The two parameters that can be adjusted are the maximum distance between points to be considered as part of one cluster and the size limitations regarding the amount of points it can contain (maximum and minimum points).The main problem was that the parameters that would allow the fine adjustment for the clustering did not deliver the desired cloud and the different dimensions of the cloud of the head made ICP return bad results. Due to the nature of the movements of the head and the positioning of the camera, the clustering algorithm could not separate the cloud that belonged to the head from that of the rest of the driver's body. The figures above show an example of the clustering that successfully separated the body from the rest of the scene (Fig.6-8).

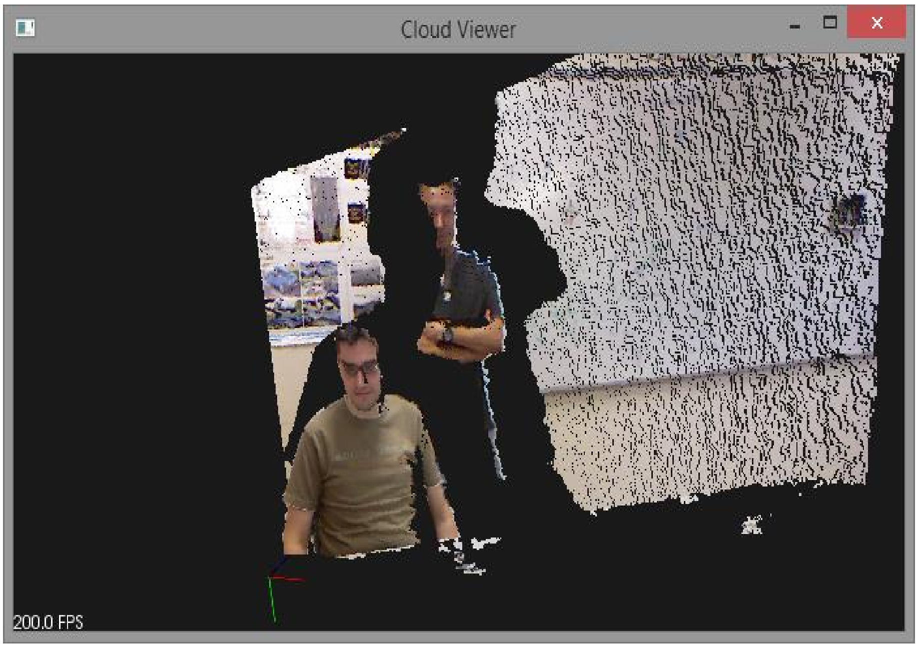


Figure 6-8: Captured cloud

This approach was tested in combination with RANSAC in order try and reduce the errors provided by both approaches. By doing first a clustering extraction and then a RANSAC model matching, the idea was to remove the unnecessary points from the cloud that belonged to the head. The distance between the points that conformed the head were too close to each other to separate them from the rest of the body. As seen in the figure 8-7, only in particular cases a small gap could occur and during the rest of the sequences this gap was not present. The results were not successful and therefore both options were discarded from the solution.

6.3 Ground Truth

As stated before, an IMU was used to confirm that the rotation angles obtained from the proposed solution were adequate. This was not the first test to evaluate the results obtained from the proposed solution. In earlier stages, the ICP algorithm was tested using rigid objects with an asymmetrical shape in order to avoid ambiguous interpretations. The results of these tests showed that the algorithm was indeed predicting the measurements in a correct way. As an example, when the object was rotated 45º the ICP algorithm returned 45º with a margin error of 3º from test to test. Because of the complexity and variability of the movements of the head, this approach was not replicated when testing the whole solution and the IMU was then chosen to be used as a ground truth.

The IMU had a differential error that was cumulative in time in one of the 3 axis. When testing to confirm if the obtained angles were accurate, this error was taken into account. The presence of this error could cause wrong differences between the algorithm's estimation and the IMU i.e. the estimation could be right but the result from the IMU would not be a close value to the estimation but something different that, with time, would be even more distant from the real value that the IMU should be returning. In order to reduce the impact of the error different measurements were performed. The goal was to establish an equation that could predict the current value of the error in time. These tests were more than 1 minute longer and the data received from the wrong axis was approximated with a linear equation. By estimating the error in time, the comparison between the IMU and the algorithm was more precise.

Figures 6-9, 6-10 and 6-11 show the output of the 3 rotation angles (pitch, roll and yaw) from the proposed solution and the IMU used as a ground truth. The correlation factor between the two outputs varied from test to test but the minimum was 0.90 and the maximum was 0.97. The correlation value was calculated with the equation below:

$$Correl(X,Y) = \frac{\sum(x-\bar{x})(y-y)}{\sqrt{\sum(x-\bar{x})^2 \sum(y-y)^2}} \qquad (19)$$

Where X corresponds to the data of the output of the solution and Y corresponds to the output of the IMU. Also $\bar{x}$ and $\bar{y}$ are the sample means from the output solution and IMU respectively.

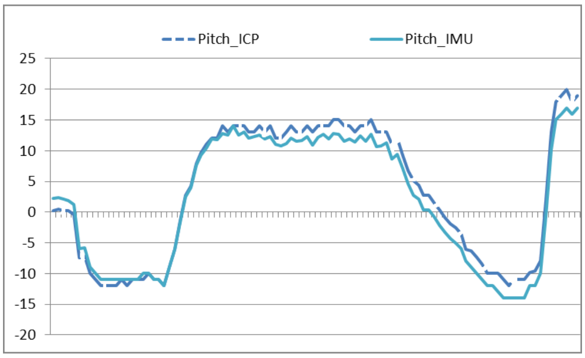


Figure 6-9: Comparison of Pitch between ICP and IMU

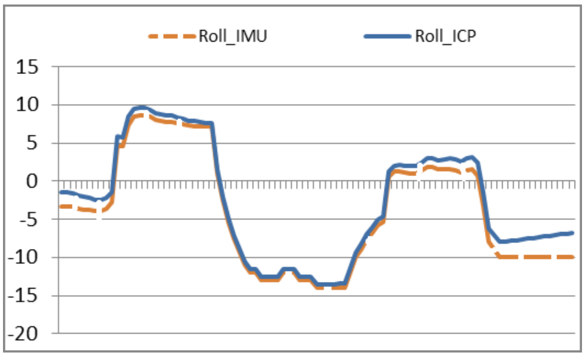


Figure 6-10: Comparison of Roll between ICP and IMU

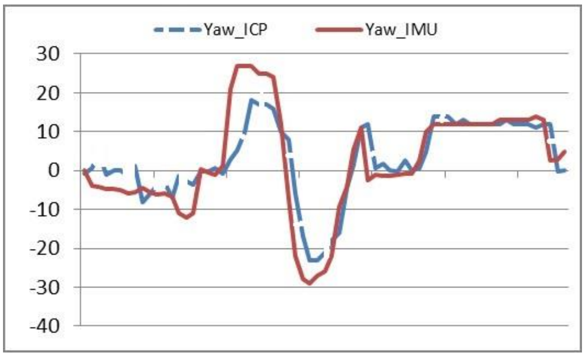


Figure 6-11: Comparison of Yaw between ICP and IMU

From the 2 data sets (IMU and output of the solution), the average error between both was calculated as a way to measure how accurate it was and the results are shown in Table V. The error was calculated with the equation below:

$$\text{Measuring Mean Absolute Error} = \frac{1}{n}\Sigma_{i=1}^{n}\left|Angle_{IMU} - Angle_{ICP}\right| \qquad (20)$$

TABLE V:

MEAN ANGLE ERROR IN THE OVERALL TEST BETWEEN THE IMU AND THE ICP ALGORITHM

Angle	Error
Pitch	2.1
Yaw	3.7
Roll	2.9

Results comparison provided by [53] shows a compendium of 38 different works with results regarding to fine pose estimation. The work presented here provides considerably better results (in terms of MAE) than 33 of them (approx. 87%), and very similar to the other 5. Furthermore, the results shown on this paper are similar to other more recent approaches [54] and [55]. Although a comparison of overall results is statistically insignificant, due to the diversity of the databases used, it is proved that the performance of the system, based on a low cost sensor, is close to other state of the art systems, even with the lack of a tracking stage that would allow a smoother error rates.

6.4 Test Conditions

Initially it was tested in the laboratory with low lighting conditions caused mostly by the sun light coming from the window. This light was not direct from the sun.

It was tested inside the car with variable lighting conditions. The variability of the illumination was not a problem for the algorithm as the 3D structure is built with infrared information that is immune to the changes in the visible light. Since the early tests, the robustness of the solution was evident. The first outdoor test was done inside an underground garage and then driving to the outside where the sun was shining. This approach was successfully tested during different exhibitions of the IVVI 2.0 with more than 50 subjects of different sex, race and age.

As with any other sensor, it can get its sensor saturated and the Kinect is no exception to this rule. Although robust to the lighting conditions, it can get saturated with particular sun light intensities. When the sun is rising or about to set, the projection of the sun rays may point directly to the infrared sensor and return a *NaN* value without colour or depth information. This is because the sun irradiates, among other, infrared light at the same frequency as the projector and the sensor inside the Kinect. It is worth noticing that this sensor was not meant to be used outdoors but inside a house. Figure 6-12 shows at the left the image captured under normal conditions and to the right is the image obtained when the sensor is saturated.

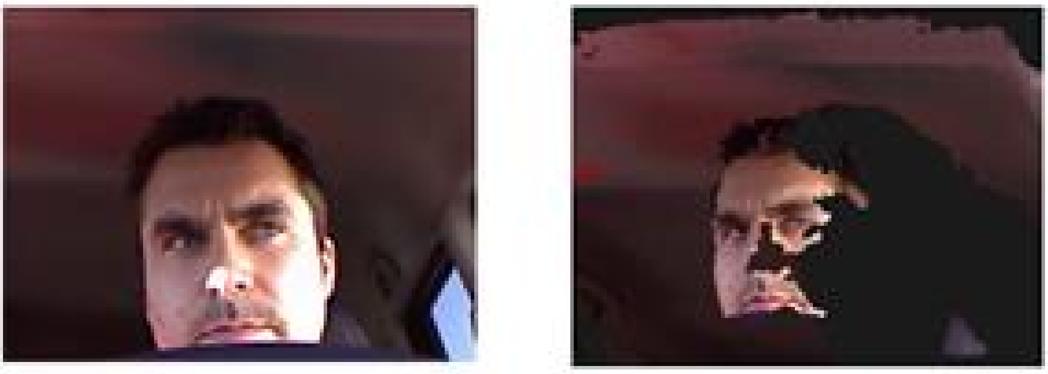


Figure 6-12: Excessive lighting reflected saturates the sensor

6.5 Detected Features

By using drawing techniques, different ROIs were set on the driver's face. This ROIs have the goal to reduce the searching space for a determined feature thus reducing the execution time and the possibility to detect a false positive. In order to analyse the driver and obtain more information from its behaviour, the chosen features were the eyes and the nose. The nose because it can be used for other approaches for the head orientation and positioning and the eyes in order to detect where the driver is looking and for other experiments that could take advantage of the picture of the driver's eyes. The mouth was discarded for this solution due to the fact that it is the part that has the most changes either because the driver is talking, shouting or yawning. It would not give, for this solution, any viable information that could not be found through the interpretation of the other detected features or the head pose estimation. The detection of the features was possible in both colour and infrared images. This means that there is the possibility to analyse the driver in complete darkness. Fig. 6-13 (a) shows the image that can be obtained from the infrared stream. After several image processing algorithms, the face was detected (Fig 6-13c) but the features like the eyes and nose could not be detected due to the noise present in the infrared streaming. Still as it can be seen in Fig. 6-13(c), the features of the face can be detected in the colour streaming.

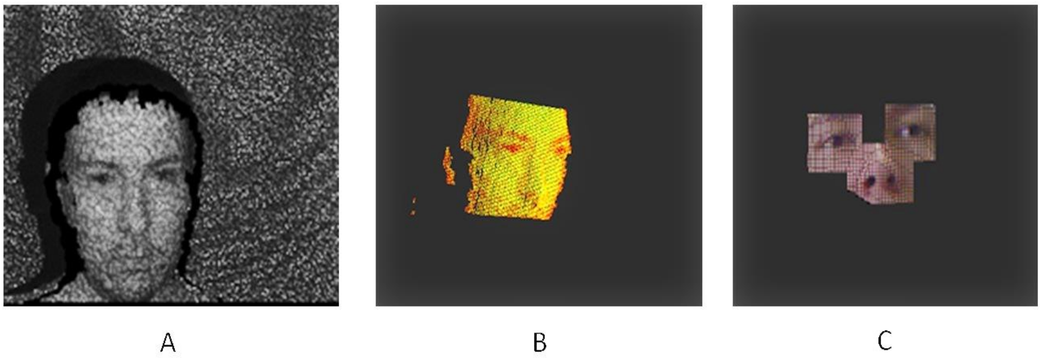


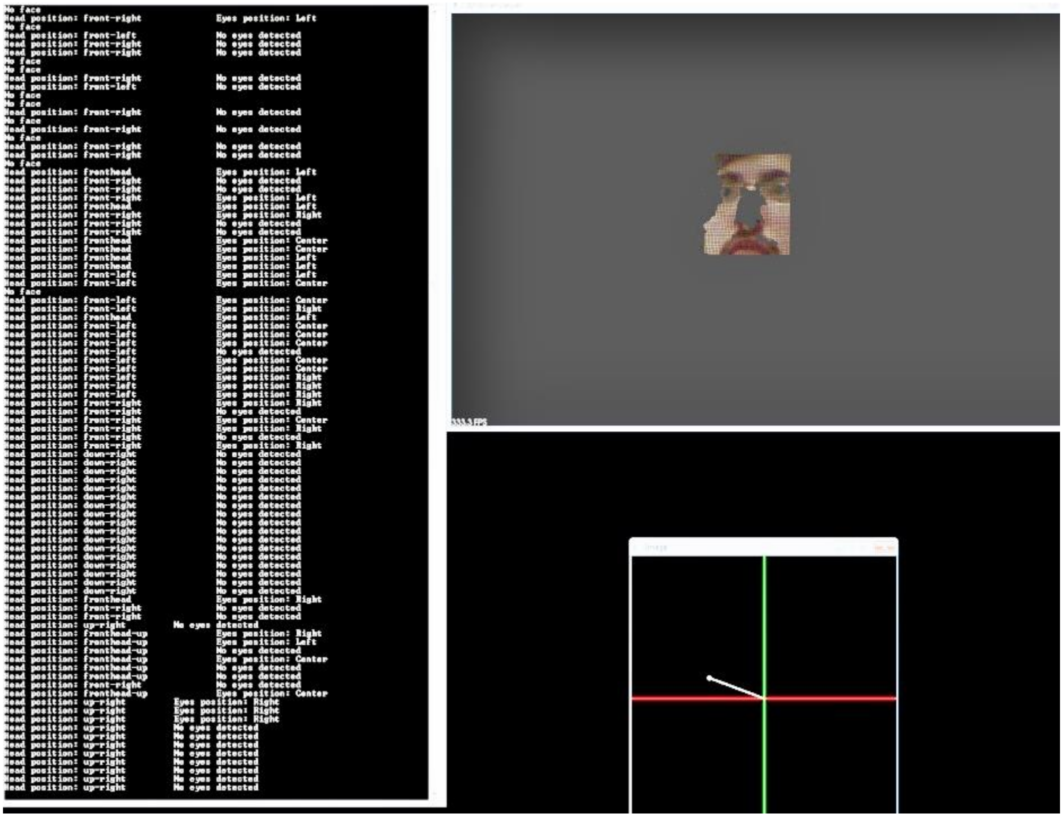
Figure 6-13: Tests done with the infrared streaming

6.6 Representation of the Results

The display of numbers in a console window can be useful when debugging the software but it is not the most clear or obvious way to represent results. Therefore, a graphical tool was developed that would represent in the way of a vector where the driver was looking at a certain moment. Starting from the centre of a reference system with *X* and *Y* axis, the vector will increase its magnitude proportional to the rotation angle of the head. The vector's angle is the same as that of the driver's head.

An acoustic alert is also included in order to help with the interpretation of the gaze estimation. This is because when moving the eyes, it is difficult to see the result displayed in the console or through the vector in the graphical representation window. A high pitch pulsing tone will be heard when the combination of the head orientation and gaze estimation is such that the driver is looking at the right mirror. For example, if the head is oriented to the centre and the eyes to the right, then the high pitch tone will be heard. Also if the head is oriented to the right and the eyes to the centre.

The data can be also displayed in another viewer by sending the data through a socket to another graphical tool. An example of the application working while showing the 3 different outputs can be seen in Fig. 6-14. To the left is the output to the console in text mode. At the top right corner is the graphical representation in 3D of the captured cloud belonging to the face. At the bottom right corner is the graphical representation of the head pose estimator where in white is the vector pointing in the direction in which the head is pointing. Additional displays can be included if necessary especially if more complex layers are to be developed with the behavior of the driver or the area where it is looking.



```
No face
Head position: front-right              Eyes position: Left
No face
Head position: front-left               No eyes detected
Head position: front-right              No eyes detected
Head position: front-right              No eyes detected
No face
No face
Head position: front-right              No eyes detected
Head position: front-left               No eyes detected
No face
No face
Head position: front-right              No eyes detected
No face
Head position: front-right              No eyes detected
No face
Head position: front-right              No eyes detected
Head position: front-right              No eyes detected
No face
Head position: fronthead                Eyes position: Left
Head position: front-right              No eyes detected
Head position: front-right              No eyes detected
Head position: front-right              Eyes position: Left
Head position: fronthead                Eyes position: Left
Head position: front-right              Eyes position: Right
Head position: front-right              No eyes detected
Head position: front-right              No eyes detected
Head position: fronthead                Eyes position: Center
Head position: fronthead                Eyes position: Center
Head position: fronthead                Eyes position: Left
Head position: fronthead                Eyes position: Left
Head position: front-left               Eyes position: Left
Head position: front-left               Eyes position: Center
No face
Head position: front-left               Eyes position: Center
Head position: front-left               Eyes position: Right
Head position: fronthead                Eyes position: Left
Head position: front-left               Eyes position: Center
Head position: front-left               Eyes position: Center
Head position: front-left               Eyes position: Center
Head position: front-left               No eyes detected
Head position: front-left               Eyes position: Center
Head position: front-left               Eyes position: Center
Head position: front-left               Eyes position: Right
Head position: front-left               Eyes position: Right
Head position: front-left               Eyes position: Right
Head position: front-right              Eyes position: Right
Head position: front-right              No eyes detected
Head position: front-right              Eyes position: Center
Head position: front-right              Eyes position: Right
Head position: front-right              No eyes detected
Head position: front-right              Eyes position: Right
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: down-right               No eyes detected
Head position: fronthead                Eyes position: Right
Head position: front-right              No eyes detected
Head position: front-right              No eyes detected
Head position: up-right         No eyes detected
Head position: fronthead-up             Eyes position: Right
Head position: fronthead-up             Eyes position: Left
Head position: fronthead-up             No eyes detected
Head position: fronthead-up             Eyes position: Center
Head position: fronthead-up             No eyes detected
Head position: fronthead-up             No eyes detected
Head position: front-right              No eyes detected
Head position: fronthead-up             Eyes position: Center
Head position: up-right         Eyes position: Right
Head position: up-right         Eyes position: Right
Head position: up-right         Eyes position: Right
Head position: up-right         No eyes detected
Head position: up-right         No eyes detected
Head position: up-right         No eyes detected
Head position: up-right         No eyes detected
Head position: up-right         No eyes detected
Head position: up-right         No eyes detected
Head position: up-right         No eyes detected
Head position: up-right         No eyes detected
```

Figure 6-14: Representation of results

The representation of the results is intended only to make the output of the solution more understandable and to help with the occasional debugging of the code when necessary although this should not be present when using this application in collaboration with another application due to the fact that the processing time to display the data to the console is very high. Also the calculation of the vector that shows the orientation of the head takes additional time.

6.7 Integration of the estimators

With the head pose estimation algorithm and the gaze algorithm explained, this section describes the combined result of the two approaches for determining what the driver is looking at. In order to keep the results clear and simple, the data obtained from both algorithms has been discretized into regions i.e. left, centre and right instead of numeric values for the rotation of the head and the position of the pupil of the eye. By combining the different results of the estimators, the software can determine if the driver is looking to the left mirror, the right mirror or the road.

The regions are described in the image below. Notice that the definition of what is left, centre and right can be changed in the code in order to adapt to the different requirements of the applications where the proposed solution is to be executed.

The graphics below (Fig. 6-15, 6-16 and 6-17) show the result of the combined work of the head pose estimator and the gaze estimator executed in the laboratory. The test consisted in fixing the head in one position at a time and moving the eyes to the 3 regions that the gaze estimator can distinguish (left, centre and right). Then, the head would move to another position and the movement of the eyes was repeated. The head was positioned in 3 different orientations (left, centre and right) that represent the most common movements of the head while driving.

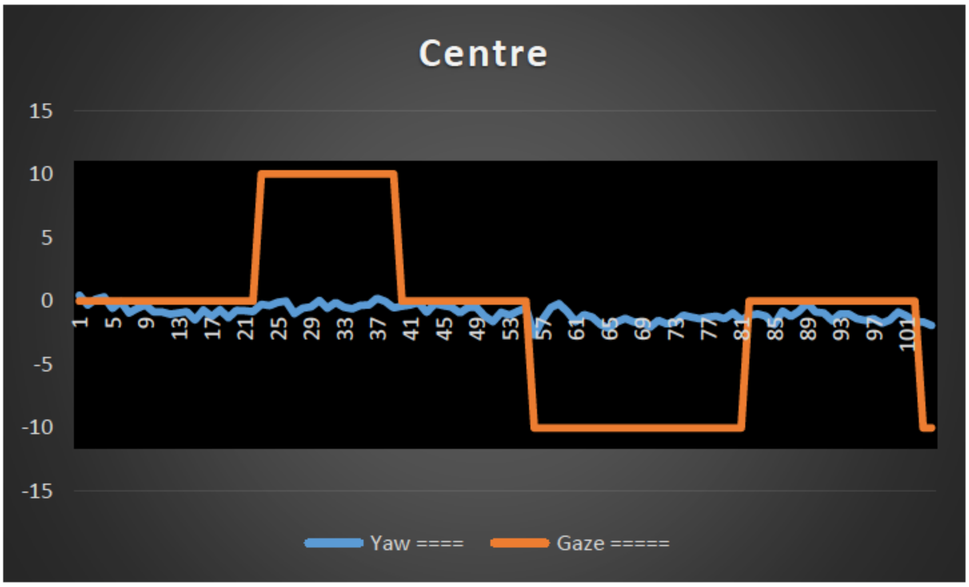


Figure 6-15: Gaze estimation with the head oriented to the centre

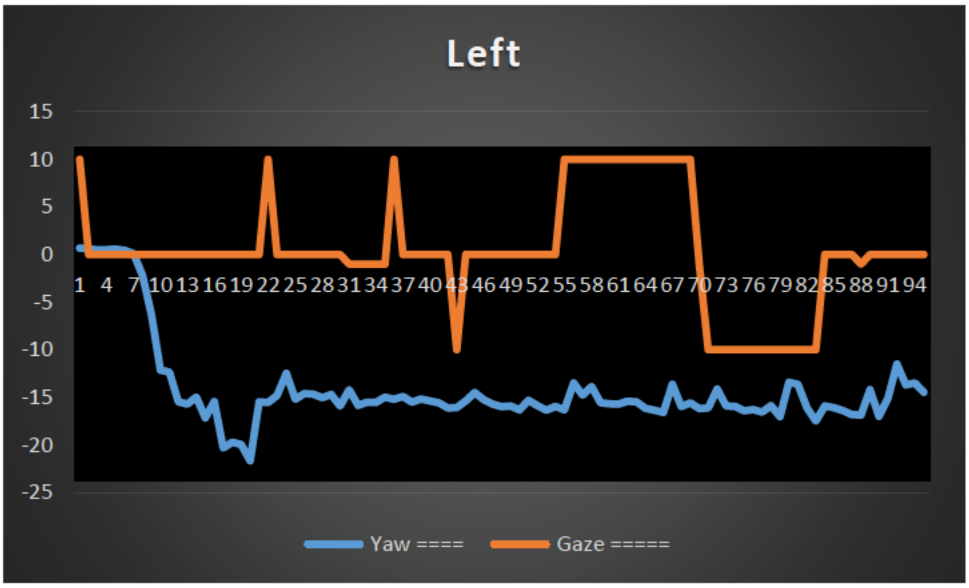


Figure 6-16: Gaze estimation with the head oriented to the left

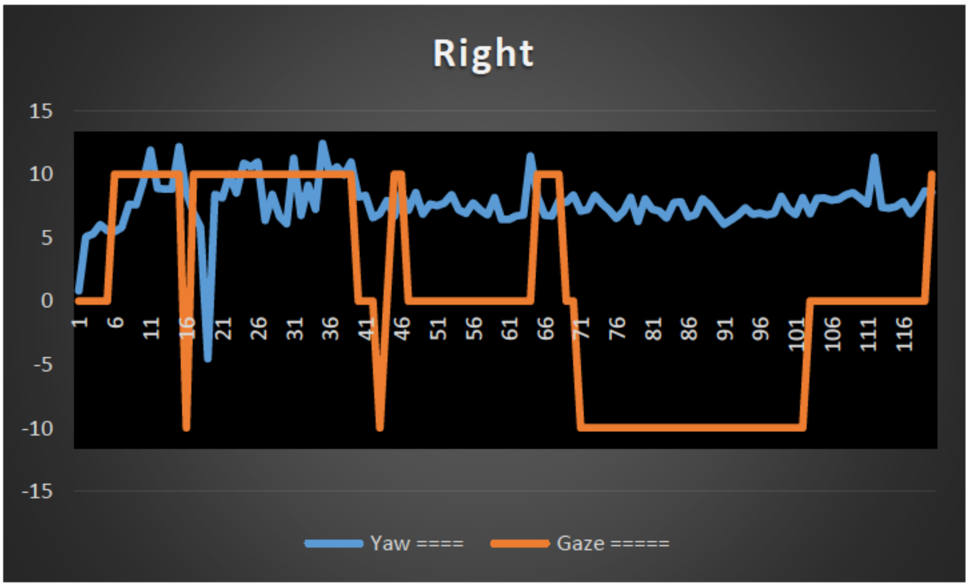


Figure 6-17: Gaze estimation with the head oriented to the right

The colour camera did not perform with the same reliability outside due to the saturation of the sensor either because of the position of the sun or the reflection of bright objects such as light coloured clothes that would reflect and cause too much brightness. An equalization of the luminance channel was performed to increase the detection rate of the cascade classifier. This equalization was done just after the face was detected and the eyes were to be searched. Once the image was equalized, the contrast of the pupil and the eyeball was greater and the gaze estimation was improved. The image, as expected, was not the best to work with due to the noise and interference present (Fig. 6-18 and 6-19).

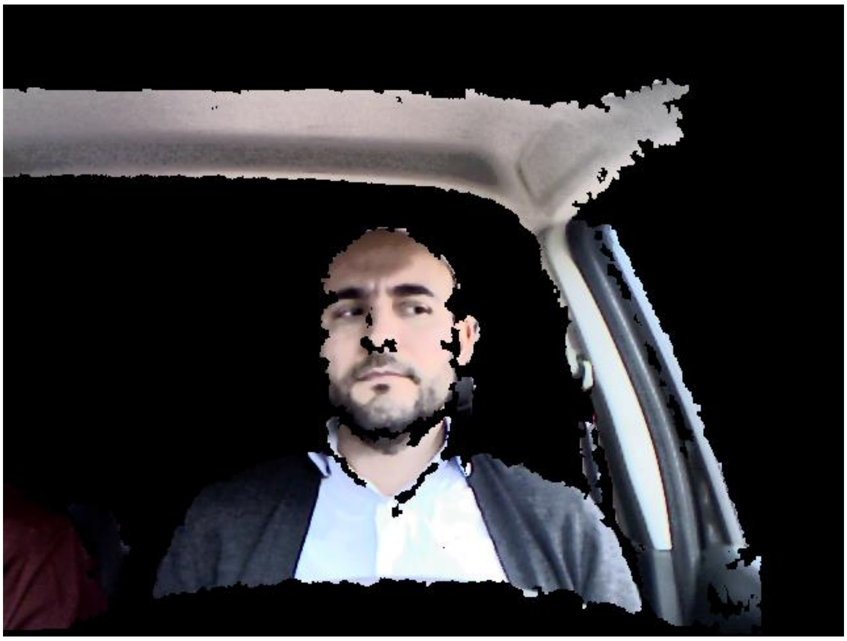


Figure 6-18: Invalid data in the region of the nose

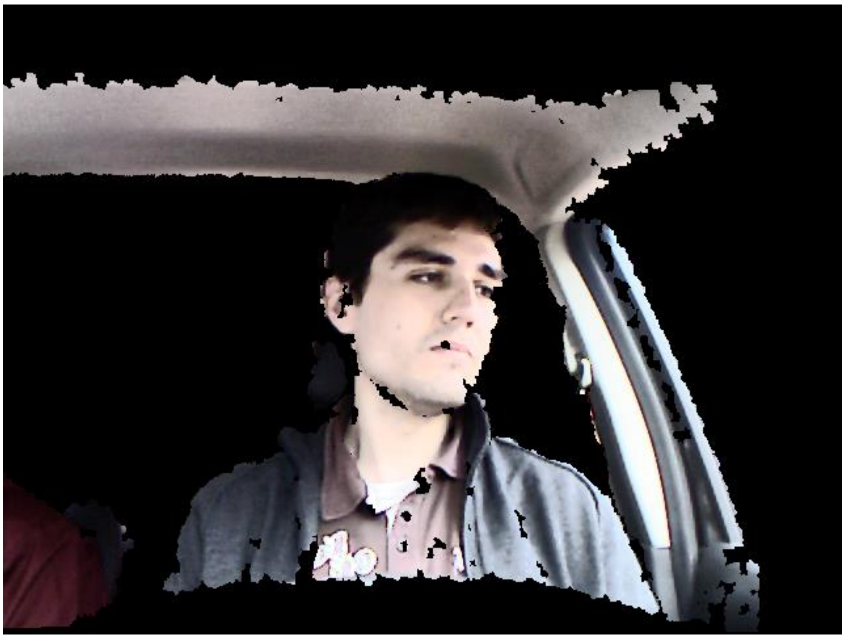


Figure 6-19: Occlusion of the left eye and high presence of invalid data

This made the estimation of the head pose and the gaze more challenging. Below are the graphical representations (Fig. 6-20, 6-21, 6-22) of the tests performed outside with adverse lighting conditions and an example of the image from where the estimations had to be done.

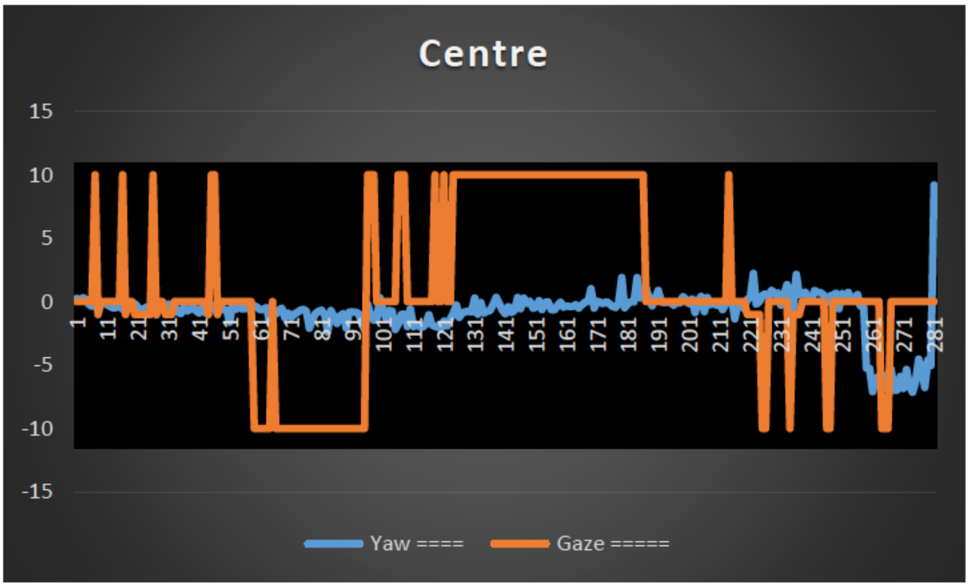


Figure 6-20: Outside testing while looking to the centre

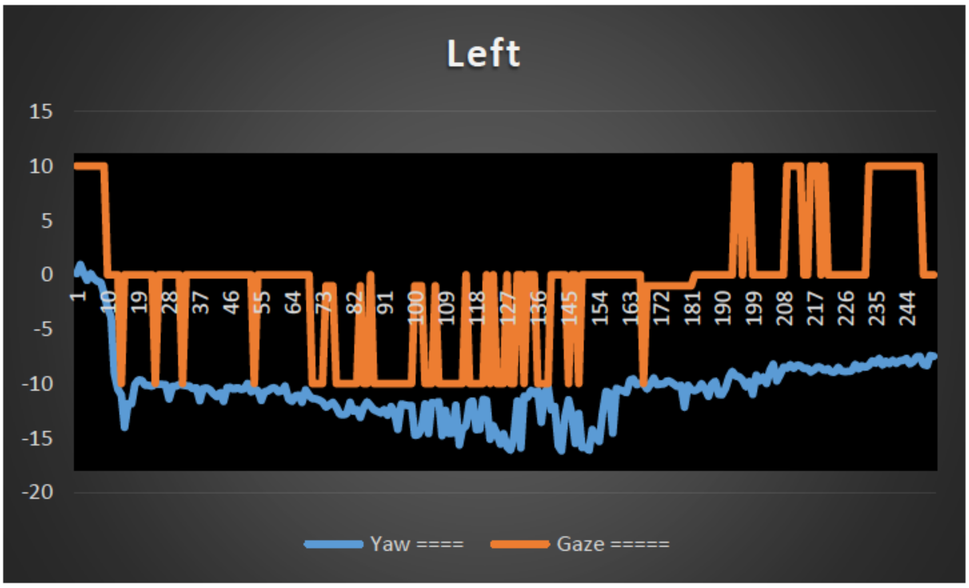


Figure 6-21: Outside testing while looking to the left

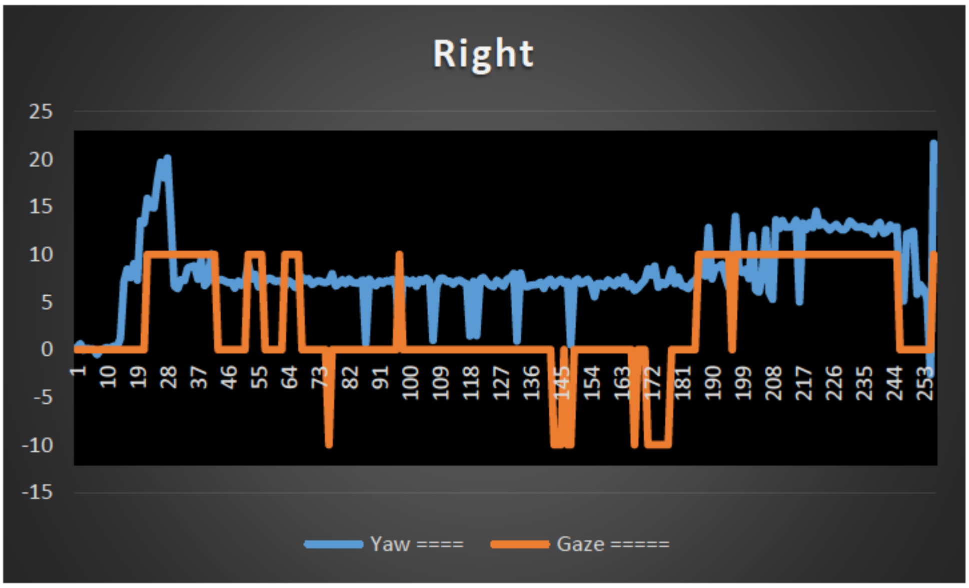


Figure 6-22: Outside testing while looking to the right

6.8 Drowsiness detector

The algorithm was tested with more than 2300 frames from different video sequences taken from different persons both in the laboratory and inside the vehicle under adverse lighting conditions. The subjects were both male and female, with different skin and eye colour in order to make the sample group as heterogeneous as possible. The images below (Fig. 6-23 6-24, 6-25) show the result of the application detecting the eye as closed. In blue is the rectangle surrounding the analyzed blob and in white is the blob.

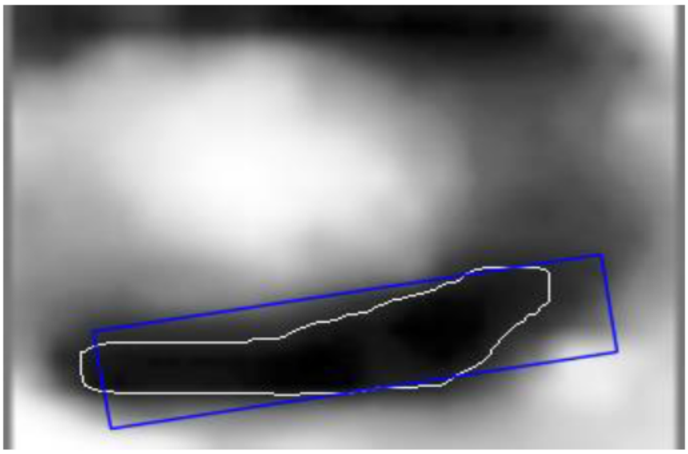


Figure 6-23: Eye detected as closed, female Caucasian.

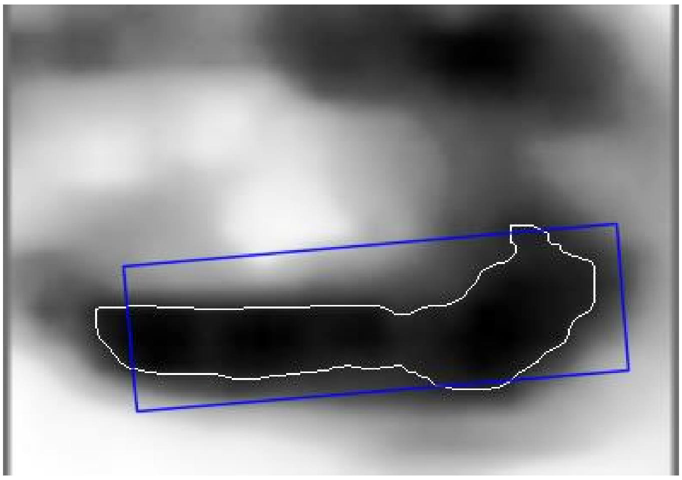


Figure 6-24: Eye detected as closed, male Caucasian

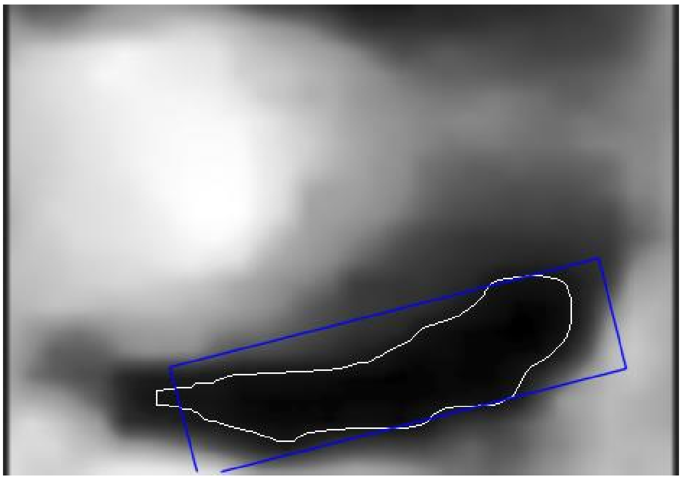
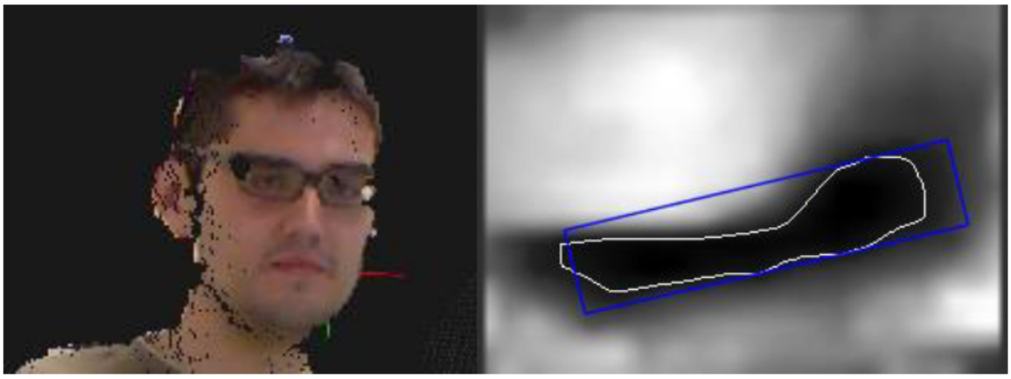


Figure 6-25: Eye detected as closed, male Latin American

Some of the results of the tests are shown in Table VI. All the sequences were analyzed frame by frame in order to identify the scenarios where the program tagged the analyzed frame as that of a closed eye when there was an eye closed but also when it was not tagged as closed because the eye was open. These two conditions are considered as the right identifications of this test and are shown in the first column of the table. The 2nd column shows the situations where the analyzed frame contained an open eye but the algorithm replies that the eye is closed. The 3rd column belongs to the situations where the eye was closed but the algorithm did not consider it as closed. It is worth mentioning that the subject from test 1 had glasses on when the sequence was taken therefore, the detection rate was lower than what was expected due to the false positives caused by the frame of the glasses and the interference of the crystals. Fig. 6-26a shows the coloured cloud captured by the sensor and Fig 6-26b shows a closed eye from a frame that was captured later. Overall the algorithm identified the situation correctly with a success rate of 84.81% as it can be seen on the bottom right cell of the table.



(a) (b)

Figure 6-26: Scenario with glasses

TABLE VI:

RESULTS OF THE DROWSINESS DETECTOR

Test No	Closed Eye Tagged Open Eye Not tagged	Open Eye Tagged	Closed Eye Not tagged	Total	Correct/Total ratio
1	108	29	28	165	0.65
2	172	6	5	183	0.94
3	144	9	13	166	0.87
4	136	4	3	143	0.95
5	118	13	9	140	0.84
6	148	9	8	165	0.90
7	167	40	2	209	0.80
8	160	32	0	192	0.83
Accumulated	1153	142	68	1363	0.85

The representation of the results of the drowsiness detector are shown in the graphical representation among the head orientation. It is a coloured bar that will increase its height in proportion to the amount of time the eyes are closed. The colour of the bar changes in proportion to the height too. When the threshold set to trigger the alarm is reached, it will be at its maximum height, the colour will be red and a text of warning will be shown in the graphical representation.

Next to the graphical representation, an intermittent acoustic alarm is triggered to warn the driver. Both alert systems (visual and acoustic) will not stop until the percentage of the time that the eye was closed during an interval is below 20%. The next figures (Fig. 6-27, 6-28, 6-29 and 6-30) are some examples of the application when estimating if the eyes are closed.

The alarms implemented for this application were chosen only to show the results of the overall solution and are not intended to be definitive. A deep study regarding what alarm to implement should be done before implementing in a definitive platform. This is in order not to cause an accident when alerting the driver that could make a reactionary move and causing an accident like making an abrupt movement with the wheel of the car.

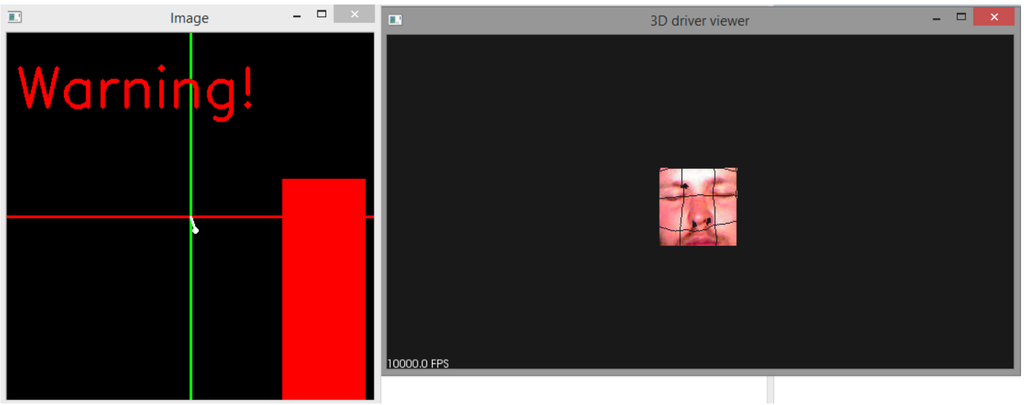


Figure 6-27: Results with Caucasian male

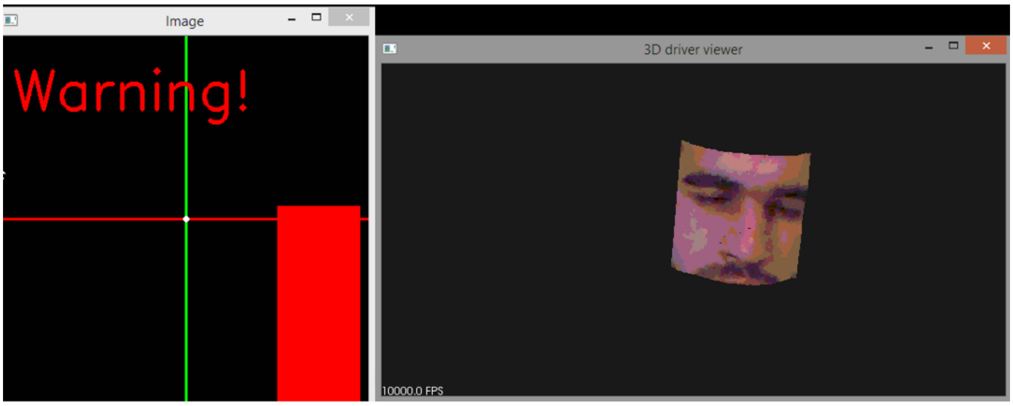


Figure 6-28: Results with Mediterranean male

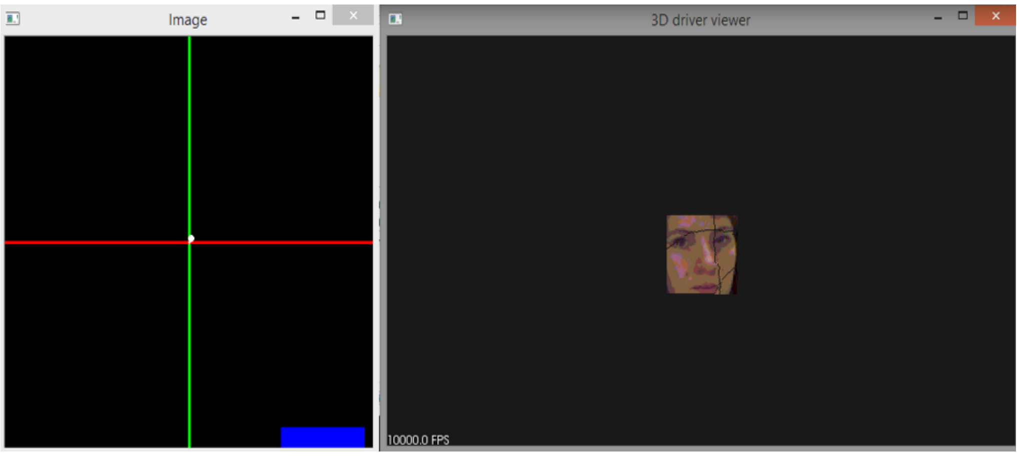


Figure 6-29: Results with Caucasian female

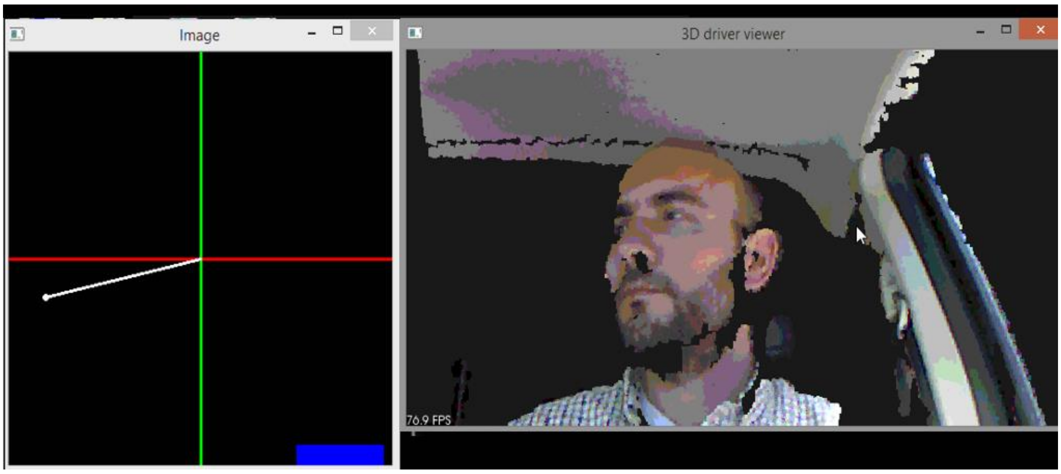


Figure 6-30: Results with Caucasian male outdoor

Many of the errors that happened when the algorithm detected that the eye was closed when it was actually open were due to invalid points obtained from the camera. The next image (Fig. 6-31) shows two consecutive frames of a sequence where the region corresponding to the invalid points at the left of the pupil is growing until it affects the dark region of the pupil. These errors happen due to the fact that when the sensor has an invalid point with a *NaN* value, it will not only give a *NaN* value for the 3D data but the colour information is also ignored. This of course altered the calculation of how many frames had a closed eye.

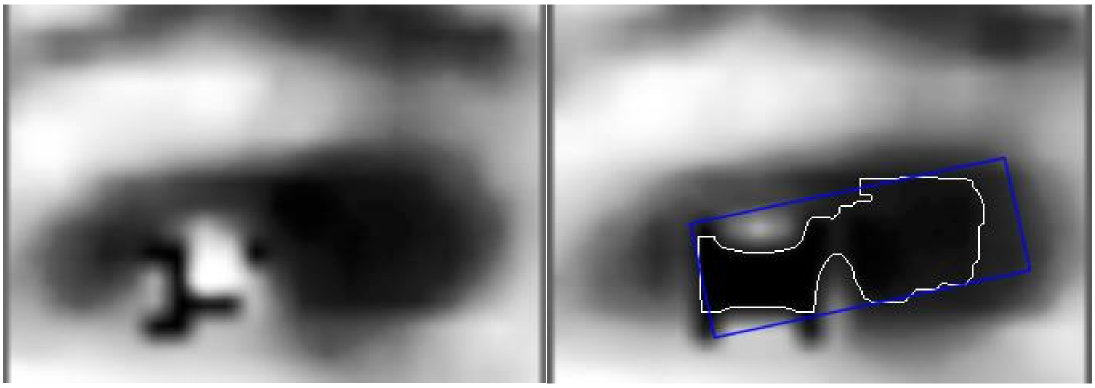


Figure 6-31: Wrong detection due to the error in the eye

Another factor that could contribute to a wrong identification is the shape of the eye. The narrower the eyes, the harder it will be to discern between open and close. This narrow gap could be found also in all of the tests performed when the subject was opening the eyes after blinking. Also when about to close the eyes but still not fully closed. The system would consider this transition as the eyes being closed. Although these were the most predominant contributors to false positives, there were other factors but their frequency was so low that they could be considered particular cases. One of this unique scenario, that caused a wrong detection, was the one caused by the reflection of the brightness on the makeup present in the eye. Another scenario found to give a false positive was the subtle movement of the eyebrow and make a horizontal dark line that the algorithm would detect as the closed lids. This last scenario could be prevented with a camera with higher resolution so that the position of the detected blob could be considered as another filtering stage. This means that the vertical position of the analyzed blob should be in the middle part of the image and not close to the upper or lower border.

CHAPTER 7. CONCLUSIONS

7.1 Introduction

The goal of the thesis was to implement an ADAS to estimate if the driver is distracted or falling asleep based on the orientation of the driver's head and estimate the gaze orientation with the Kinect sensor bar. It was possible to achieve the main goal and the results were explained in the early chapters of this thesis. The main advantage of using the Kinect sensor is the combination of both 2D and 3D data at the same time. Also the price of the sensor that is below 50€ makes this solution a good budget approach. The sensor had the sufficient resolution in its sensors to estimate the head pose and distinguish the 3 areas where the driver is looking. The accessibility to the 2D and 3D data allows the test of algorithms for processing clouds of points and colour images at the same time without the need of calibration techniques. By creating different ways of representing the results, the final user can interpret the estimated results through a visual tool that shows the rotation of 2 axis or by reading in the command line the 3 rotation angles.

Like any other application, this one has its own limitations like the saturation of the sensor or the accuracy of the gaze estimation. These problems can be solved using another sensor, modifying the existent or with new tools to come in the near future. But within the limits established by the hardware and software resources, the application worked properly and achieved the expected results.

7.2 Contributions

When doing a project with different modules, algorithms, sub-stages and pipelines, many small milestones are achieved during the development stage. Not that the final objective is important (it is, after all the main goal) but to ignore the middle steps and collateral effects would mean that the whole picture is not being seen. The development of this solution is no exception. Although not all, some of the most important achievements are listed below.

- Affordable driver monitoring: this application does not require an expensive or hard to find device to work. The current price is estimated, varies from shop to shop, to be under €100. It is available in many stores online as in retail stores too. Another advantage of this application is the fact that it is a non-invasive method that may affect the movements or the driver itself in any way as opposed to other solutions that require the attachment of a device to the body or to wear a heavy equipment on the head.

- Robust head pose estimation: The results show that the estimation of the head pose is robust enough to estimate where the driver's head is oriented. If the inside of the vehicle's cabin is divided in a matrix composed of 9 regions, it is possible to estimate with accuracy where the head is oriented. More robust results can be achieved if the gaze orientation is included. Thanks to the infrared construction of the 3D structure of the face, the estimation is very robust against lighting variations.

- 3D data manipulation with support of 2D data: With the information obtained from the cloud of points, it is possible to work with 2D and 3D data at the same time without complications of previous calibrations or large and complex adjustments. If the proper libraries are used, this will be done from the beginning and allow the testing stage to start immediately rather than spending time with set ups. The 3D data is associated with the colour information and thus the filters and tools used for colour images can be used to filter the 3D data as explained in this solution when detecting and building a cloud of the face.

- Improved analysis of the driver's behaviour: Using the head pose estimation and the gaze orientation together allows the end user to study the behaviour of the driver under different circumstances that can vary from urban to motorway behaviours. Other researchers, like the research group CAOS (Control Learning and Systems Optimization Group), can use this tool in a simulator and build a new layer of results based on what this solution can deliver such as the misbehaviours mentioned earlier as an example. More complex layers can be built from the information of the head pose estimation and the gaze orientation such as an attention index to the road during a determined session.

- Offline mode analysis: With additional modifications to the proposed solution, it is possible to store sequences at 10Hz. A special format of point clouds was used to achieve this speed without losing quality of the captured scene. The modified solution stores the information in different folders. One with the 2D information stored in Jpeg format, another with the 3D information stored in compressed point cloud format, a third folder with the IMU/GPS information and finally a text file that can be used as an index control. All the information stores the time stamp of the CPU and thus the synchronization of one image with its corresponding image and IMU information can be used to reconstruct the moment when it was captured.

7.3 Future Works

As stated before, the project has its limitations that define what it can do and under what conditions. The results speak for themselves when the right conditions are available but when the conditions are not ideal, the application will deliver results that are not as expected. New sensors are available that can improve the application's performance by using new technologies such as the Kinect sensor bar for the XBOX One videogame console that uses time of flight technology to estimate the depth of the scenario with more accuracy and detect new features like the heart rate.

With the current sensor, it is possible to analyse the infrared information. Further tests using this stream could guarantee that the application can work under any lighting conditions.

Another future work would be the optimization of the head pose estimation algorithm by using only the cloud composed by the ROIs of the eyes and nose. This would allow the complete integration of this solution with 3D data and thus removing any flaws that could come from the 2D analysis due to lighting conditions and other challenges.

Either with the next generation Kinect or another hardware it would be possible to improve the gaze orientation with a higher resolution camera. So far the gaze estimator can distinguish between 3 regions but with a better camera this could be extended to more regions that would allow a better understanding of the behaviour of the driver.

Finally, the integration with additional tools through a complex architecture or sockets communication in order to develop a more robust solution. As an example, the solution could be integrated in the ROS architecture to collaborate with other algorithms by sending the rotation angles of the head or the position of the eyes through “topics” that another application like the pedestrian detection could complement for a third application that would warn the driver from a dangerous situation.

REFERENCES

[1] “Traffic Safety Basic Facts” 2012. DaCoTA | Project co-financed by the European Commission, Directorate-General for Mobility & Transport. European Road Safety Observatory.

[2] http://ec.europa.eu/transport/road_safety/specialist/knowledge/fatique/index_en.htm last accessed 04/2015

[3] http://www.tac.vic.gov.au/road-safety/statistics/summaries/fatigue-statistics last accessed 04/2015

[4] http://ec.europa.eu/transport/road_safety/specialist/knowledge/fatique/fatigue_and_road_crashes/frequency_of_fatigue_related_crashes_en.htm last accessed 04/2015

[5] Horne, J.A. & Reyner, L.A. (1995) *Sleep related vehicle accidents*. British Medical Journal, 310, 565-567

[6] Langwieder, K. & Sporner, A. (1994) *Struktur der Unfälle mit Getöteten auf Autobahnen im Freistaat* Bayern im Jahr 1991, HUK-Verband, Büro für KfZ-Technik, Munich

[7] Report of the Obstructive Sleep Apnoea Working Group “New Standards and Guidelines for Drivers with Obstructive Sleep Apnoea syndrome“ Brussels 2013

[8] http://www.volvotrucks.com/SiteCollectionDocuments/VTC/Corporate/Values/ART%20Report%202013_150dpi.pdf European Accident Research and Safety Report 2013, Volvo Trucks

[9] Institut National du Sommeilet de la Vigilance, Autoroutes & Ouvrages Concédés (ASFA) “Sleepiness at the wheel, white paper” June 2013

[10] D. L. Hall and J. Llinas, *Handbook of Multisensor Data Fusion*. CRC Press, 2001, pp. 3-18

[11] F.E. White, Data Fusion Lexicon, Joint Directors of Laboratories, Technical Panel for C3, Data Fusion Sub-Panel, Naval Ocean Systems Center, San Diego, 1991.

[12] N. Steinberg, C. L. Bowman, and F. E. White, “Revisions to the JDL Model,” in *Proceedings of the SPIE Conference on Architectures Algorithms and Applications*, 1999.

[13] Garcia, F.; de la Escalera, A.; Armingol, J.M., "Enhanced obstacle detection based on Data Fusion for ADAS applications," Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on , vol., no., pp.1370,1375, 6-9 Oct. 2013

[14] Thomaidis, G.; Kotsiourou, C.; Grubb, G.; Lytrivis, P.; Karaseitanidis, G.; Amditis, A., "Multi-sensor tracking and lane estimation in highly automated vehicles," Intelligent Transport Systems, IET , vol.7, no.1, pp.160,169, March 2013

[15] Garcia, F.; Cerri, P.; Broggi, A.; de la Escalera, A.; Armingol, J.M., "Data fusion for overtaking vehicle detection based on radar and optical flow," Intelligent Vehicles Symposium (IV), 2012 IEEE , vol., no., pp.494,499, 3-7 June 2012

[16] Bo-Hyun Yu; Dong-Hyung Kim; Byung-Gab Yu; Seung-Yeol Lee; Chang-Soo Han, "Development of prototype of a Unmanned Transport Robot for transport of construction materials" Control, Automation and Systems, 2008. ICCAS 2008. International Conference on , vol., no., pp.448,452, 14-17 Oct. 2008

[17] Kloeden, H.; Damak, N.; Rasshofer, R.H.; Biebl, E.M., "Sensor resource management with cooperative sensors for preventive vehicle safety applications," Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2013 Workshop on , vol., no., pp.1,6, 9-11 Oct. 2013

[18] Krotosky, S.J.; Trivedi, M.M., "On Colour-, Infrared-, and Multimodal-Stereo Approaches to Pedestrian Detection," Intelligent Transportation Systems, IEEE Transactions on , vol.8, no.4, pp.619,629, Dec. 2007

[19] Haltakov, V.; Belzner, H.; Ilic, S., "Scene understanding from a moving camera for object detection and free space estimation," Intelligent Vehicles Symposium (IV), 2012 IEEE , vol., no., pp.105,110, 3-7 June 2012

[20] Shu Zhan; Kurihara, T.; Ando, S., "Facial Orientation and Eye-Gaze Detection Based on Real-Time 3D Facial Imaging By Using Correlation Image Sensor," Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on , vol.2, no., pp.1061,1065, 16-18 Oct. 2006

[21] Sung Joo Lee; Jaeik Jo; Ho Gi Jung; Kang Ryoung Park; Jaihie Kim, "Real-Time Gaze Estimator Based on Driver's Head Orientation for Forward Collision Warning System," Intelligent Transportation Systems, IEEE Transactions on , vol.12, no.1, pp.254,267, March 2011

[22] Heuer, S.; Chamadiya, B.; Gharbi, A.; Kunze, C.; Wagner, M., "Unobtrusive in-vehicle biosignal instrumentation for advanced driver assistance and active safety," Biomedical Engineering and Sciences (IECBES), 2010 IEEE EMBS Conference on , vol., no., pp.252,256, Nov. 30 2010-Dec. 2 2010

[23] Singh, H.; Bhatia, J.S.; Kaur, J., "Eye tracking based driver fatigue monitoring and warning system," Power Electronics (IICPE), 2010 India International Conference on , vol., no., pp.1,6, 28-30 Jan. 2011

[24] Friedrichs, F.; Miksch, M.; Bin Yang, "Estimation of lane data-based features by odometric vehicle data for driver state monitoring," Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on , vol., no., pp.611,616, 19-22 Sept. 2010

[25] Castignani, G.; Derrmann, T.; Frank, R.; Engel, T., "Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring," *Intelligent Transportation Systems Magazine, IEEE* , vol.7, no.1, pp.91,102, Spring 2015

[26] http://openkinect.org/wiki/Protocol_Documentation#Cameras (Last accessed 04/2015)

[27] http://msdn.microsoft.com/en-us/library/jj131033.aspx (Last accessed 04/2015)

[28] Industries, Adafruit. (2012, March) Adafruit Industries. [Online]. www.adafruit.com

[29] Industries, Prime Sense. (2012, March) Prime Sense Industries. [Online]. www.primesense.com

[30] Hack a day community. (2012, March) Hack a Day. [Online]. www.hackaday.com

[31] V. Frati and D. Prattichizzo, "Using Kinect for hand tracking and rendering in wearable haptics," World Haptics Conference (WHC), pp. 317-321, June 2011.

[32] E.S. Santos, E.A. Lamounier, and A. Cardoso, "Interaction in Augmented Reality Environments Using Kinect," Virtual Reality (SVR), 2011 XIII Symposium on, pp. 112-121, May 2011.

[33] Lu Xia, Chia-Chih Chen, and J.K. Aggarwal, "Human detection using depth information by Kinect," Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 15-22, June 2011.

[34] N. Ganganath and H. Leung, "Mobile robot localization using odometry and kinect sensor," International Conference on emerging signal processing applications (ESPA), pp. 91-97, January 2012.

[35] Amorim Vaz, F.; de Souza Silva, J.L.; Sol Dos Santos, R., "KinardCar: Auxiliary Game in Formation of Young Drivers, Utilizing Kinect and Arduino Integration," Virtual and Augmented Reality (SVR), 2014 XVI Symposium on , vol., no., pp.139,142, 12-15 May 2014

[36] Gallahan, S.L.; Golzar, G.F.; Jain, A.P.; Samay, A.E.; Trerotola, T.J.; Weisskopf, J.G.; Lau, N., "Detecting and mitigating driver distraction with motion capture technology: Distracted driving warning system," Systems and Information Engineering Design Symposium (SIEDS), 2013 IEEE , vol., no., pp.76,81, 26-26 April 2013

[37] Ohn-Bar, E.; Trivedi, M.M., "The Power Is in Your Hands: 3D Analysis of Hand Gestures in Naturalistic Video," Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on , vol., no., pp.912,917, 23-28 June 2013

[38] D. Olmeda, A. De La Escalera, and J. M. Armingol, “Detection and tracking of pedestrians in infrared images,” 2009 3rd International Conference on Signals Circuits and Systems SCS, pp. 1-6, 2009

[39] M. J. Flores, J. M. Armingol, and A. Escalera, “Real-Time Warning System for Driver Drowsiness Detection Using Visual Information,” *Journal of Intelligent Robotic Systems*, vol. 59, no. 2, pp. 103-125, 2009.

[40] J. M. Collado, C. Hilario, A. De La Escalera, and J. M. Armingol, “Adaptative road lanes detection and classification,” in *Transform*, 2006, vol. 4179, pp. 1151-1162.

[41] J. P. Carrasco Pascual, “Advanced Driver Assistance System based on Computer Vision using Detection, Recognition and Tracking of Road Signs,” Universidad Carlos III de Madrid, 2009

[42] B. Musleh, A. D. L. Escalera, and J. M. Armingol, “Obstacle Detection and Localization Using U-V disparity with Accelerated Processing Time by CUDA,” in *IEEE International Conference on Robotics and Automation 2011 ICRA*, 2011.

[43] F. García, F. Jiménez, J. J. Anaya, J. M. Armingol, J. E. Naranjo, A. de la Escalera, “Distributed Pedestrian Detection Alerts Based on Data Fusion with Accurate Localization” Sensors. Key A, Vol. 13, pp. 11687, 11708, 2013

[44] “MTI-G X-sens.” [Online]. Available: http://www.xsens.com/en/general/mti-g. [Accessed: 05-Mar-2012].

[45] Viola, P.; Jones, M., "Rapid object detection using a boosted cascade of simple features," Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on , vol.1, no., pp.I-511,I-518 vol.1, 2001

[46] Besl, P.J.; McKay, Neil D., "A method for registration of 3-D shapes," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.14, no.2, pp.239,256, Feb 1992

[47] Wierwille W, et al. (1994) Research on vehicle-based driver status / performance monitoring: development, validation, and refinement of algorithms for detection of driver drowsiness. Report No DOT HS 808 247. NHTSA, Washington D.C.

[48] H. Ueno, M. Kaneda, and M. Tsukino, Development of drowsiness detection system, Proc. of Vehicle Navigation and Information Systems Conf., Yokohama, Japan, 1994, pp. 15-20.

[49] R. Grace, V. Byrne, D. Bierman, J. Legrand, D. Gricourt, R. Davis, J. Statszewski, and B. Carnahan, A Drowsy Driver Detection System for Heavy Vehicles, Proc. of 17th Digital Avionics Systems Conf., Bellevue, USA, Oct. 1998, Vol. 2, pp. I36/1-I36/8

[50] Sommer, D.; Golz, M., "Evaluation of PERCLOS based current fatigue monitoring technologies," Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE , vol., no., pp.4456,4459, Aug. 31 2010-Sept. 4 2010

[51] M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

[52] www.pointclouds.org/documentation/tutorials/cluster_extraction.php (last visited 20/05/2015)

[53] E. Murphy-Chutorian and M. M. Trivedi, “Head pose estimation in computer vision: a survey.,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 4, pp. 607–626, 2009.

[54] R. Oyini Mbouna, S. G. Kong, and M.-G. Chun, “Visual Analysis of Eye State and Head Pose for Driver Alertness Monitoring,” Intell. Transp. Syst. IEEE Trans., vol. 14, no. 3, pp. 1462–1469, 2013.

[55] E. Murphy-Chutorian and M. M. Trivedi, “Head Pose Estimation and Augmented Reality Tracking: An Integrated System and Evaluation for Monitoring Driver Awareness,” IEEE Trans. Intell. Transp. Syst., vol. 11, no. 2, 2010