IZMIR KATIP CELEBI UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

A STEREOPHOTOGRAMMETRIC APPROACH FOR DRIVER ASSISTANCE SYSTEMS

M.Sc. THESIS

Emre ÖZDEMİR

Department of Geomatics Engineering

Geomatics Engineering Program

Thesis Advisor: Assoc. Prof. Dr. Özşen ÇORUMLUOĞLU

DECEMBER 2015

IZMIR KATİP ÇELEBİ UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

A STEREOPHOTOGRAMMETRIC APPROACH FOR DRIVER ASSISTANCE SYSTEMS

M.Sc. THESIS

Emre ÖZDEMİR
(Y130108002)

Department of Geomatics Engineering

Geomatics Engineering Program

Thesis Advisor: Assoc. Prof. Dr. Özşen ÇORUMLUOĞLU

DECEMBER 2015

İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SÜRÜŞ DESTEK SİSTEMLERİ İÇİN STEREOFOTOGRAMETRİK BİR YAKLAŞIM

YÜKSEK LİSANS TEZİ

Emre ÖZDEMİR
(Y130108002)

Harita Mühendisliği Anabilim Dalı

Harita Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Özşen ÇORUMLUOĞLU

ARALIK 2015

Emre Özdemir, a **M.Sc.** student of Izmir **Katip Celebi University Graduate School of Science and Engineering** student ID Y130102002, successfully defended the **thesis** entitled “**Stereophotogrammetric Approach for Driver Assistance Systems**” which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor: **Assoc. Prof. Dr. Özşen ÇORUMLUOĞLU** …........
Izmir Katip Celebi University

Jury Members: **Assoc. Prof. Dr. Mehmet ÇETE** …........
Izmir Katip Celebi University

Assist. Prof. Dr. Adem EREN …........
Izmir Katip Celebi University

Date of Submission: 18 December 2015
Date of Defense : 28 December 2015

To my family,

FOREWORD

I would especially like to thank my advisor, Assoc. Prof. Dr. Özşen Çorumluoğlu, for his healthy degree of optimisms when experiments disappointed and all seemed lost. I would also like to thank him for the warm and friendly atmosphere he garnered in his group; it encouraged sharing of ideas, insightful discussions and a productive work environment.

Next, I would like to thank Dr. İbrahim Asri for his both technical and motivational contributions for this thesis project.

Without great friends and family, this endeavor would have concluded before it began. I would like to thank them for believing in me, encouraging me to continue going, and providing distractions from work when they were needed.

I want to thank my parents. It is with their help for all my life that I became who I am today. Thanks for always being there for me, believing in me, and motivating me to set out on my own path. I cannot begin to describe how lucky I feel for having them as my parents. All opportunities and accomplishments I owe to them.

December 2015 Emre ÖZDEMİR

TABLE OF CONTENTS

Page

ABBREVIATIONS

GIS	**:**	Geographic Information System
NIR	**:**	Near Infrared
3D	**:**	Three Dimension
RGB	**:**	Red Green Blue
KYM	**:**	Kırmızı Yeşil Mavi
CAD	**:**	Computer Aided Design
CCD	**:**	Charge Coupled Device
CMOS	**:**	Complementary Metal Oxide Semiconductor
UAV	**:**	Unmanned Aerial Vehicle
GPS	**:**	Global Positioning System
IMU	**:**	Inertial Measurement Unit
GCP	**:**	Ground Control Point
FPS	**:**	Frames Per Second
ABS	**:**	Antilock Braking System
TCS	**:**	Traction Control System
ESP	**:**	Electronic Stability Program
ACC	**:**	Adaptive Cruise Control
RADAR	**:**	Radio Detecting And Ranging
CSV	**:**	Comma Separated Values
ASCII	**:**	American Standard Code for Information Interchange
GUI	**:**	Graphical User Interface
DSLR	**:**	Digital Single Lens Reflex
GUI	**:**	Graphical User Interface
SURF	**:**	Speeded-Up Robust Features
BRISK	**:**	Binary Robust Invariant Scalable Keypoints
FAST	**:**	Features from Accelerated Segment Test
MSER	**:**	Maximally Stable Extremal Regions
GPU	**:**	Graphics-Processing Unit
CPU	**:**	Central Processing Unit
FPGA	**:**	Field Programmable Gate Array
EXIF	**:**	Exchangeable Image File Format

LIST OF TABLES

Page

LIST OF FIGURES

Page

A STEREOPHOTOGRAMMETRIC APPROACH FOR DRIVER ASSISTANCE SYSTEMS

SUMMARY

Anyone will agree with that one of the sectors is the automotive sector that has been showing great improvements since the first car put into the market. Great impact of innovations and technological advantages provided by computer technologies are being seen in the sector for last couple of decades. The most important usage of these high technologies are about safety and comfort of driving. Such systems are aiming vehicles, which are not crashing and comfortable ones.

As it is well known fact, the amount of vehicles in the traffic always increases in great extents year by year. In addition to other causes of traffic collisions, this increase becomes an important factor for the rise in number of traffic accidents. This situation also increases the risk of traffic accident that one of us probably finds himself or herself involved in. The producers in the sector have been taking precautions and preventions to decrease the accident risks. Followings are some of those; designing of stable, solid, strong, able bodied, safe and secure vehicles, improving road conditions and secure road design, broadcasting public ads. Furthermore, in addition to designs of roads, vehicles and traffic, the most influencing factors are drivers' faults. Therefore, in that era of the automotive security systems, so many researches are going on and so many safety driving systems and support systems are being developed, improved, and produced to decrease the amount and the risk of traffic accidents caused by driver mistakes. Those systems provide aid, warning and even autonomous driving opportunities to reduce to minimize human effect.

In this research and development area, plenty of automotive firms develop various driving safety systems and driver assistance systems for several purposes and in different contents and concepts by using today's technological opportunities offered.

Here in this thesis project, suggested such a driver assistance system that can take benefit of cameras as the only sensors, rather than RADAR (Radio Detecting and Ranging) or other ultrasonic sensors. The project depends on a system that collects data through a pair of cameras, detects objects of interest and makes image measurements with the aid of digital image processing, then, calculates three dimensional real-world position of the interested object with respect to the vehicle, using stereo-photogrammetry technique. Therefore, the system produces 3D coordinates of the object ahead only. With the aid of these points of the object, the distance between the vehicle and the object ahead can be calculated and controlled with the safe following distance, according to speed of vehicle. As the stereo-photogrammetric processing will be handled for only limited number of points, the system is observed to be efficient in terms of 3D coordinate quality and performance.

SÜRÜŞ DESTEK SİSTEMLERİ İÇİN STEREOFOTOGRAMETRİK BİR YAKLAŞIM

ÖZET

İlk araçtan günümüze en çok ilerleme kaydeden sektörlerden birinin de otomotiv sektörü olduğu herkes kabul edecektir. Özellikle bilgisayar ve dijital çağın getirdiği yenilik ve teknolojik avantajların pek çok alanda ortaya çıkan çarpıcı etkisi araç teknolojilerinde de etkin bir şekilde kendisini hissettirmektedir. Günümüz araç teknolojilerinin geldiği noktada öne çıkan yeni araç teknolojilerinden en önemlilerinin de araçlardaki sürüş güvenliğini arttırmaya yönelik güvenli sürüş destek sistemleri olduğunu görürüz.

Bilindiği üzere trafiğe çıkan araç sayısı gün ve gün artmaktadır. Araç sayısındaki bu artış, diğer nedenlere ek olarak trafik kazalarının sayısını her geçen gün arttıran önemli bir neden olarak gösterilmektedir. Bu durum da herhangi birimizin dahil olabileceği bir trafik kazası riskini arttırmaktadır. Kazaları azaltacak önlemler olarak da, genellikle daha sağlam araçların tasarlanması, teknik açıdan kaza olasılıklarını azaltacak şekilde yolların tasarlanması, sürücülere yönelik hazırlanan kamu spotları gibi yöntemlerden faydalanılmaktadır. Halbuki, yol, araç, trafik gibi nedenlerden çok daha etkin olan bir diğer kaza sebebi sürücü hatalarıdır. Bu nedenle günümüzde en çok üzerinde durulması gereken konulardan biri de araçlarda sürücü hatalarını minimize edecek, sürüş güvenliğini arttıracak ve sürücüye daha fazla uyarı, hatta yardım sağlayacak, bunun da ötesinde sürüşü otomatikleştirecek sistemler olduğundan dünyada bu konu üzerine çok ciddi çalışmalar yapılmaktadır.

Bu çalışmalar kapsamında hemen her otomotiv firması günümüz teknolojisinin avantajlarını kullanarak basit ve/veya kompleks değişik içerik ve amaca yönelik olarak çeşitli sürüş destek sistemleri geliştirmektedir.

Bu tez çalışmasında, sensör olarak, RADAR (Radio Detecting And Ranging / Radyo ile Tespit Etme ve Mesafe Tayini) veya öteki ultrasonik sensörler yerine, yalnızca bir kamera çiftinden güç alan bir sürücü destek sistemi önerilmiştir. Proje, veriyi kamera çiftinden alan, dijital görüntü işleme desteği ile ilgili objeyi görüntüde bulan ve resim ölçmelerini gerçekleştiren, sonrasında da objenin araca göre üç boyutlu gerçek dünya koordinatlarını stereo fotogrametri tekniği ile hesaplayan bir sistem göstermektedir. Bu sayede, sistem yalnızca önündeki objeye ait noktalar için üç boyutlu koordinatlar üretir. Bu obje noktaları sayesinde, obje ile araç arası mesafe hesaplanır ve araç hızına göre belirli güvenli takip mesafesi ile kıyaslanır. Stereo fotogrametrik işleme yalnızca belirli sayıda nokta için yapılacağından, sistemin 3B nokta kalitesi ve performans bakımından verimli olduğu gözlemlenmektedir.

1. INTRODUCTION

Traffic accidents have been an important problem for humanity to deal with. To reduce the number of accidents and decrease the effects of them, many driver assistance systems have been scientifically researched, developed, and applied into the practice for last few decades. In this thesis study, the main concern is to provide a technique that collects necessary data for a driver assistance system through cameras. In the following sections, photogrammetry and digital image processing techniques, traffic and driver assistance systems will be discussed in detail in terms of their history and implementation.

1.1. General Background of Photogrammetry

The word ‘photogrammetry’ consists of three Greek words as follows, ‘photos’, ‘gramma’, and ‘metron’. English meanings of these words are respectively ‘light’, ‘something written or drawn’ and ‘to measure’ (Duggal, 2006). With the aforementioned explanation, one can mind what photogrammetry is about. One of the most widely accepted definition describes photogrammetry as science, technology and art of getting trustworthy metric data of an existent phenomenon and environment via documenting, measuring and interpreting photographs or images (Şeker & Duran, n.d.).

The mathematical modeling and basics of photogrammetry depending on perspective geometry, is originated from Italian Renaissance, 15th century, including famous artists as Brunelleschi, Piero della Francesca and Leonardo da Vinci (Ostermann & Wanner, 2012). This fundamental approach was used by architect Meydenbauer in 1858 with terrestrial photographs for the plans of Wetzlar Cathedral in Germany, and by Aime Laussedat, who was a military photographer, in 1859 to produce Paris’ city plans via terrestrial photographs (Konecny, 2014).

Photogrammetry has been evolving since the first application in terms of technologies and techniques used for example capturing photographs or images and other related data. Even the perspective geometry was replaced by central projection after a short time it emerged.

Going back to first decades of photogrammetry, all the equipment were completely mechanical and optical ones. The cameras were equipped with films or tablets and the processes were all analog with expensive, huge, heavy, complicated, and hard to use analogue stereo plotting machines. These processes include orientation of the photographs, image measurements and all other calculations.

Later on, as the technology developed and computer usage widened, the concept of photogrammetry also started to change rapidly and headed towards using computers by analytical photogrammetry. Analytical photogrammetry mainly relies on algorithmic reconstruction concept of photogrammetry with digitized image measurements and transmitting these data into a computer system such as a CAD (Computer Aided Design) database. The time span of the hardware used can be seen in the Figure 1.1 below.

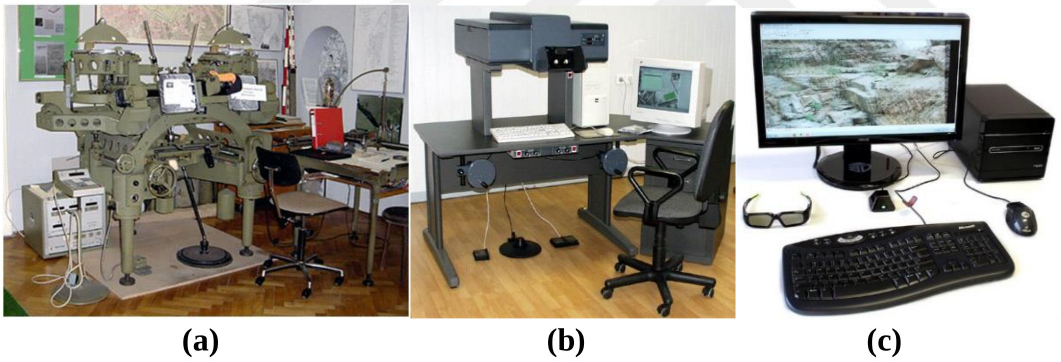


Figure 1.1 : Stereo-Photogrammetric Time Span, respectively, Wild A8 Analogue Stereo Plotting Machine (a) (Grant, 2011), Analytical Photogrammetric Station “Stereoanagraph” (b) (GeoSystem, n.d.), Current Stereophotogrammetric Workstations (c) (Birch, 2011).

As the analogue photogrammetry and analytical photogrammetry are not being used widely nowadays, and digital photogrammetry is the most recent and dominating new age photogrammetry technique and used in this thesis study, these two techniques are not detailed more in here. Therefore, digital photogrammetry technique is the newest one among all others, which can be thought as a product of developed technologies such as those in computers and cameras. This technique is suitable with both digitally captured images and scanned (digitized) images.

The cameras used before were generally metric cameras with known interior geometry in analogue and analytical photogrammetry. However, in digital photogrammetry, digitized or digitally captured images are being used, which is a game-changing step because of annihilation of both film costs and impracticability of films. Today, the high reachability rate of professional non-metric digital cameras makes them to be used commonly in plenty application areas with the help of different camera calibration softwares. An analogue camera's film needed to be changed and kept under certain conditions in order to use it again in the future. The advantage of digital cameras arise here. With the help of CCD (Charge Coupled Device) and CMOS (Complementary Metal Oxide Semiconductor) sensors in digital cameras, all the images can be saved into a memory card in digital form, which is considerably smaller than films and tablets and can store many more photographs when they are compared with previous hardcopy photos. With the help of digitally stocking of imagery, archives are reserving less place and offering easier access to it.

In digital age, computers can handle almost every study. So does photogrammetric processing. As mentioned before, in analogue and analytic photogrammetry stereo plotter was a necessary tool for the stereo-photogrammetric processing. Yet, in digital photogrammetry, every step of photogrammetric processing, including, orientation, image measurements, and calculations are done via computers. There are many paid and freeware softwares (Demirel & Şeker, 2015) for photogrammetric applications. However, one can create a tool easily with some programming abilities using any of the programming languages.

As being a successful technique for gathering three-dimensional reliable data, photogrammetry have been used for mapping, documentation, and industrial applications.

For mapping purposes, aerial photogrammetry technique is widely used. In such applications, photographs are taken from above ground using air vehicles such as aircrafts, unmanned aerial vehicles (UAV), helicopters etc. The exterior orientation parameters of images are both gathered from a GPS/IMU equipment, which is integrated to the camera, and calculated using ground control points (GCPs), which are points geodetically placed on the Earth's surface, with bundle block adjustment calculation that takes the system stability and precision to a higher level. Products of these applications are generally used to build up a GIS database or for topographic maps.

Terrestrial (also called as close range) photogrammetry technique is the one that widely used for modelling objects on the Earth’s surface. Photographs are generally taken on terrain, on a tripod or hand-held usually, using control points on the object. Exterior parameters of the photographs are calculated using these points, in common. Products of this technique are possibly used in areas of GIS, documentation, restoration, construction, accident scene investigation, archeology etc.

Close range photogrammetry became so useful through the years of technology it is also used in industrial applications since the mid-1980s with great success (Fraser & Brown, 1986). Industrial application areas of close range photogrammetry includes vehicles’, automobiles’ body deformation control, crash tests, aerospace industry for production quality control, wind energy systems and so on (Luhmann, 2010). Photographs are generally taken with mounted and fixed camera stations but also hand-held shooting is possible. Control points are located on the subject with a projector if it is necessary. For industrial purposes, especially in mass production factories, the high production speed is compulsorily protected. So, high-speed cameras, such as a few thousand frames per second (FPS) capable ones, are widely used with high megapixels. Industrial application of photogrammetry can supply 3D coordinate production of fast moving objects. All these precise equipment produce data in high accuracy in order to maintain high quality and speed production for the products. The developments in technology led close-range photogrammetry to work even above a car with mounted cameras for engineering purposes (Asri, Çorumluoğlu, & Güner, 2012).

1.2. Digital Image and Processing

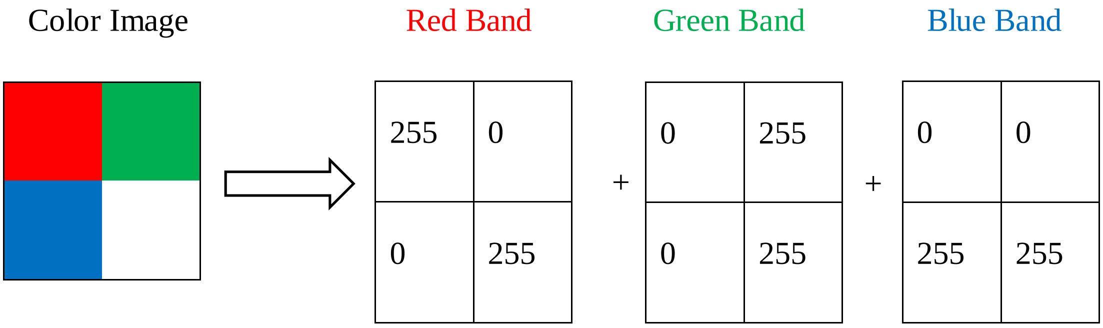
People have always been excited in documenting the moment. First examples of such documentation can be thought as drawings in caves. By years, painting, literature and other forms of art have been used to document memories. As the centuries passed, and technological improvements came up, photography also became one of those forms for documentation.

The Figure 1.2 below shows the first photograph, View from the Window at Le Gras, by Joseph Nicéphore Niépce. The photograph is said to be firstly presented in year 1827. The one above is reproduced version of the original one by Helmut Gernsheim and Kodak Research Laboratory (Austin, n.d.).

Figure 1.2 : The first photograph (Austin, n.d.).

Digital image is representation of an image in the digital environment. Either a digital image is captured as a digital image, via digital cameras, or an analogue image can be scanned and digitized, as well. In both scenarios, the final product is a digital image. Digital image can be defined as a coordinated two-dimensional plane, where each point, in other words each pixel, has its own intensity value (Gonzalez & Woods, 2002). A digital image consists of finite number of pixels, which is defined as "Contraction of picture element" according to Seventh Edition of Modern Dictionary of Electronics (Graf, 1999). These elements are the smallest substances in a digital image. Each pixel has its own digital values, along with their exact locations, which are records of brightness or intensity or gray level of the related spectral band. For example, a true color image consists of three bands as follows; red, green, and blue. Each band has data for each pixel as its brightness value. Actually, each band is a matrix including digital numbers for each pixel.

As seen in the Figure 1.3 below, a true color image, which has four pixels in total, has three bands of data stands for red, green, and blue. The red pixel has maximum brightness value for red band, and minimum value for the other bands, so it is seen as red. The same is applicable for green and blue bands' pixels. The white pixel, which is located at the bottom right corner of the image, has the maximum brightness value for all three bands, and this is the why it is seen as white.



Red Band

255	0
0	255

Green Band

0	255
0	255

Blue Band

0	0
255	255

Figure 1.3 : Bands of a true color digital image.

Digital image processing is considered as handling digital images via a computer (Gonzalez & Woods, 2002) and is performed pixel-by-pixel operations. The technique has evolved a lot with the developing technology. Different types of image processing techniques are providing image enhancements and extraction of meaningful information through image analysis. These enhancements can be generally named as digital image editing, which includes, cropping, resizing, coloring, sharpening, softening, contrast adjustment, brightness changes, noise cancelling etc. However, with image processing techniques, it becomes possible to detect, count, or recognize objects, get or set their colors, analyze their size etc.

Capabilities of digital image processing are also helping industrial necessities. Such systems consists of digital high-speed cameras, high-speed connection between production line, cameras and computers. For such high-speed acquisition, computers are also needed to be fast. Such systems are generally used to check the status of products and decide whether if they are OK or not. Then, this status information is sent to the production system using the connection between them. This helps faster production controls and reduces the control costs.

In addition to industrial applications for quality control and accident scene reconstruction, it is believed that digital photogrammetry is also useful for driver assistance systems. This would become possible with specially developed photogrammetric processing algorithms. Digital image processing would be used for object detection and image measurements in order to obtain necessary data from photographs. These digital image-processing algorithms would make photogrammetric processing in real-time.

1.3. Driver Assistance Systems: Why, Traffic and Traffic Accidents

We live in an era in which transportation is one of the most important daily routines of a civilized human being. People have the demand of transportation more than ever as the population increases and new necessities occurs in last centuries, such as business trips, visiting family elders, or holidays. For this reason with the aid of developing technology, it became possible to transport through the air, water and of course land. Every single person living in a civilized environment have to use at least one of these transportation types to go places like home, work, cinema, shopping or just as their entertaining activity. All these necessities of transportation and different types of it created one concept, called 'traffic'. The word 'traffic' is described as follows according to Oxford Dictionaries; 'Vehicles moving on a public highway' (Oxford, 2015).

The problem with the traffic is the increasing number of cars in the traffic day by day along with other vehicles (TSI, 2015a). As the number of cars are increasing every day, accidents are taking place more frequently and these accidents are causing in physical damages, injuries, and deaths.

As seen in the Figure 1.4 below, road length is not getting more as much as the number of the cars (the lessening of road lengths are caused by exclusion of village roads in 2004 and 2014 due to current law in Turkey). The relation between the number of cars and road length leads to increasement in two main problems; traffic jams and accidents. Traffic jam is a serious problem, which causes spending more time for travelling and petrol consumption in addition to psychological effects.

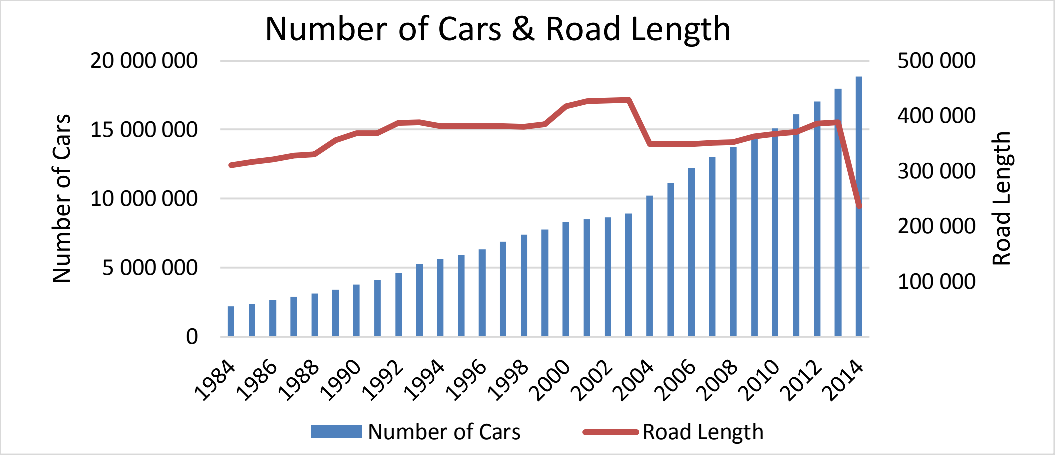


Figure 1.4 : Number of cars and road length by year (TSI, 2015b), (TSI, 2015c).

On the other hand, the number of accidents in the traffic is getting more day-by-day according to the statistical records of traffic accidents in Turkey. The Figure 1.5 shown below is one of the evidences as it shows the total number of cars in traffic and ratio of number of accidents to number of vehicles.

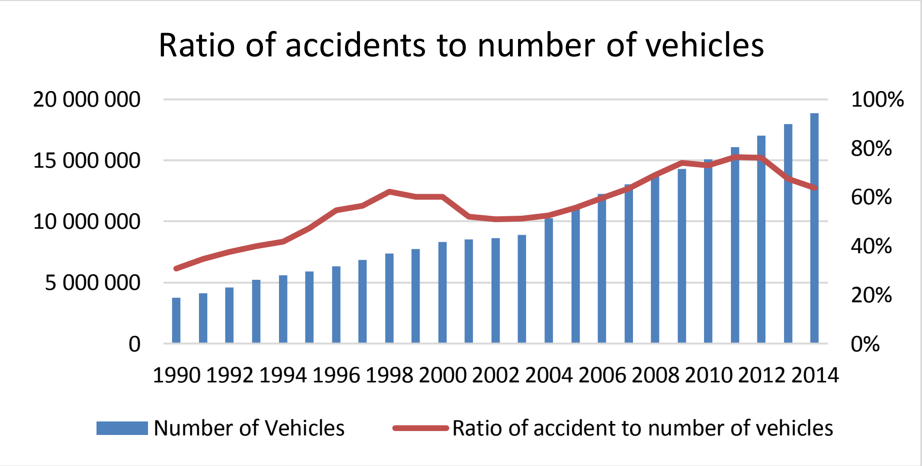


Figure 1.5 : Ratio of accidents to number of vehicles (TSI, 2015d).

As seen in the Figure 1.5 above, unsurprisingly number of vehicles and ratio of number of accidents to number of vehicles are both rising. It can be seen that more than sixty percent of cars are involved in accident. The number of deaths and injuries are also showing an increasing trend as seen in the Figure 1.6 below according to Turkish Statistical Institute.

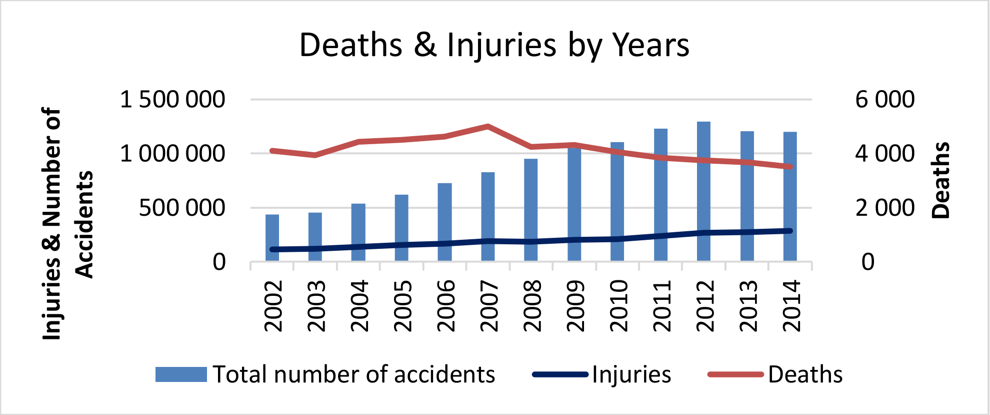


Figure 1.6 : Number of accidents, injuries and deaths (TSI, 2015e).

With the aid of high technology vehicles, better roads and traffic designs, number of deaths are getting lower after 2007, yet, number of injuries are still increasing. Which means the accidents lose effect on human life every year. However, it is sure that this is not enough for civilization especially in this era of information and technology.

As seen in the Figure 1.7 below deaths caused by motor vehicle accidents holds just below 1% when analyzed in 50 major causes of death.

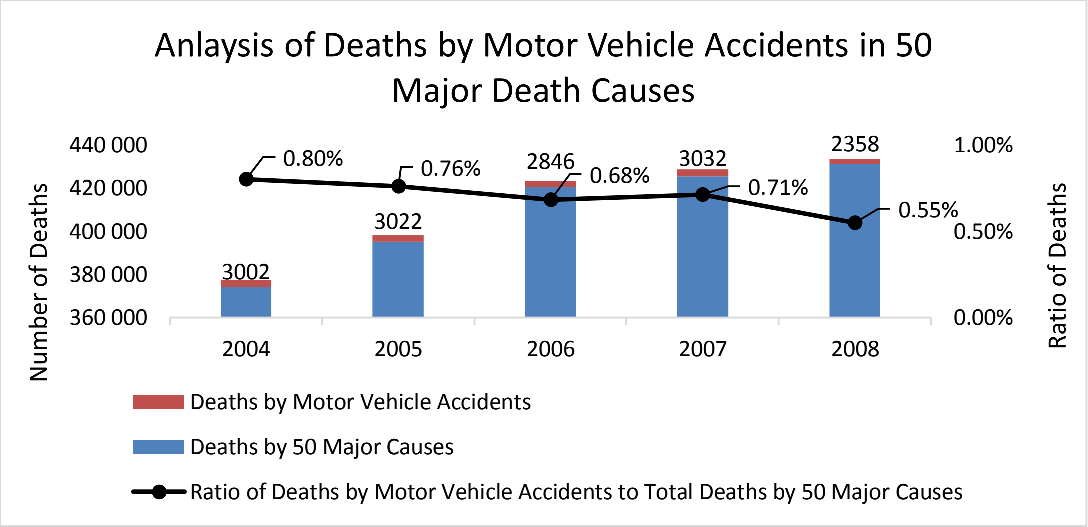


Figure 1.7 : Deaths by motor vehicle accidents in 50 major death causes (TSI, 2015f).

To lower the number traffic accidents, their reasons must be examined in order to define correct solutions. The Figure 1.8 below shows the faults causing road traffic accidents.

The Figure 1.8 below shows the number of accidents, passengers' faults' ratio, pedestrians' faults' ratio, road defects' ratio, vehicles' defects' ratio, and drivers' faults ratio. As detailed information about material damage accidents were not collected only by the year 2008, since the ratios are a little deviated. The ratio of drivers' faults in traffic accidents seems to be more than 90 percent almost every year. Therefore, the figure above clearly states that the main cause of all these traffic accidents is dominantly drivers' faults. To reduce the number of accidents, drivers' faults must be examined and reduced. To accomplish this, with the help of current technologies many driver assistance systems have become available on the market, which are detailed in the following section.

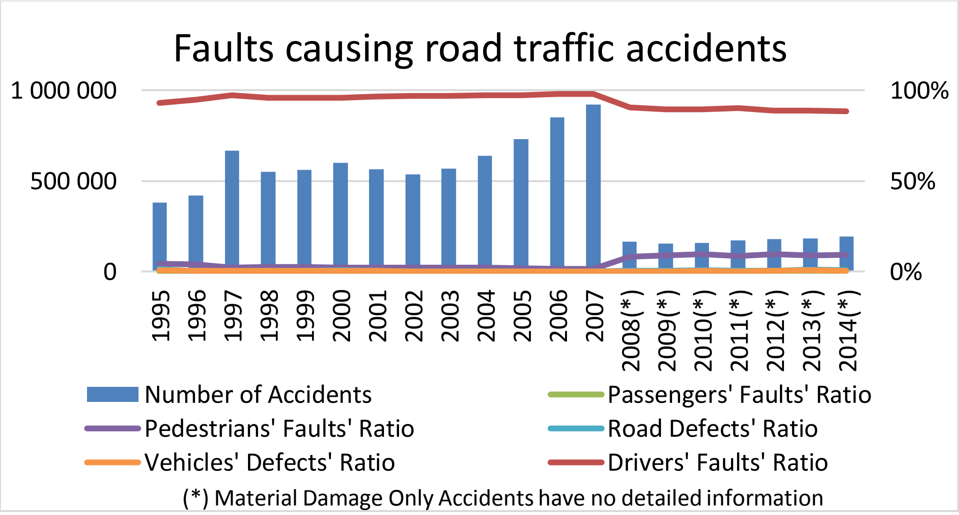


Figure 1.8 : Faults causing road traffic accidents (TSI, 2015g).

2. DRIVER ASSISTANCE SYSTEMS

The traffic accidents have become huge problems in terms of money and human life as mentioned in the previous section. Therefore, vehicle manufacturers began to design and implement new systems to their vehicles in order to prevent them from accidents as much as possible. Those systems were mainly about vehicle dynamics rather than driver and drivers' faults and these systems accepted by majority and some of them are even became compulsory in the EU (Bosch, 2011). In this section, widely available, known, and used ones of these systems will be detailed in order to emphasize the main trend of driver assistance systems.

2.1. Systems Monitoring Vehicle Dynamics

Antilock braking system (ABS) is one of the very first systems that implemented in driving environment, as being patented in 1981 (GRAD[DE], 1982). Antilock braking system is created to prevent lock of wheels in case of a sudden strong brake made by driver. The reason behind this is loss of control. If a wheel is locked, than it cannot be used to steer the vehicle, which is definitely an unwanted condition. The aim of this system is to prevent the vehicle from locked wheels. Consequently, the system checks the status of the wheels and if one of them has tendency to be locked then the brake pressure of this wheel is adapted in real time.

Following ABS, traction control system (TCS) showed up, boosting ABS. Traction control system was designated to check the wheels' situations to prevent them from spinning. In case of a spinning, system annihilates this undesirable behavior of the wheel by braking it (Bosch, n.d.).

Electronic stability program (ESP) is developed to prevent vehicle from skidding caused by oversteering, understeering or quick and strong maneuvers etc. The system checks the status of each wheel, using the functions of ABS and TCS as aid, and applies brake to one or more wheel according to the faced situation. If it is also needed to slow down the vehicle, system also handles this via engine's control unit. According to press information from Mercedes-Benz, this system was first implemented C140 Series of SEC in year 1992 (Daimler, 2010).

2.2. Systems Monitoring Vehicle Surrounding

So far, until the mid-1990s the systems were generally about driving dynamics except car park aid. These systems, which are enhanced to control driving dynamics, use sensors monitoring actions within car, while car park aid is the only one with monitoring car surrounding with ultrasonic sensors. After the millennium, systems are concerned more about the environment of the vehicle using cameras and other ultrasonic sensors. Examples of such systems are detailed below.

Car park has been a challenging act for drivers as park areas are getting smaller and more limited with the increase in number of cars and people are having small dents or scratches around their vehicle. To prevent from this, automobile sector came with ultrasonic sensor based system, which warns driver with sounds according to distance between vehicles or other objects around the vehicles while parking. The frequency of the sound depends on the distance. Some systems are also available with visualized warnings. First examples of such systems are observed about mid-1990s.

Adaptive cruise control (ACC) system consists of electronic distance measurement unit attached to vehicle's main control computer. The system is an enhanced type of a cruise control system, which makes the vehicle stay at the exact same speed using its computer system. The ACC differs from a conventional cruise control system with the distance measurement unit. Cruise control systems use only the vehicles computer and no other sensors to make the vehicle stay on the same speed. On the other hand, ACC calculates necessary following distance according to vehicle's speed and uses RADAR or laser equipment to measure the distance to the vehicle ahead. Comparing these two, system makes a decision about the vehicle's current speed. System can stay at the same speed, slow down the vehicle if needed, and then speed up the vehicle to the fixed speed when the danger disappears. This system became available on the market around year 2000.

In addition to the systems concerning about driving safety, comfort oriented ones are also became available by years. Parking assistant system is one of them. The system consists of camera(s) and/or ultrasonic sensor(s), used to automate or semi automate the parking process. The number of sensors, including cameras, depends on the complexity of the system. System detects and measures the parking space, checks whether the vehicle can fit or not. Then, handles the parking maneuver with steering wheel, while fully automated systems can also handle throttle and brake functions. Such systems are started to be widely used on the market around year 2010.

Lane keeping assistance system is another safety driving system. The system checks the lane markings positions using a camera, therefore it can understand if the vehicle is moving left or right with respect to the lane markings. Some advanced systems can also use fusion of cameras and ultrasonic sensors looking and searching rear or sides around the vehicle to produce a better solution. In case of such scenarios, the system first warns the driver through visual displays, vibrations, or sounds. Some more advanced systems check if the driver responds or not, then takes over the control of the vehicle and steers it to position between lane markings if necessary. System goes offline as the driver uses signals to change the lane. Such systems became available in the late 2000s from different manufacturers.

All these technologies' names may differ by manufacturers.

As the technology developed in every aspect, including, hardware, and software, this helped a lot on the enhancement of the systems, too. The development speed of the technology pushed and evolved driving safety systems to driver assistance systems and this would not stop here. The technological process, in the last decade particularly, is moving driver assistance systems into autonomous driving systems.

Many of the car manufacturers are working on autonomous driving, including Volvo Car Group, Daimler AG, BMW, Ford Motor Company, General Motors, Robert Bosch GmbH, Continental AG etc. Autonomous driving systems, in other words self-driving systems, use fusion of RADAR sensors, cameras, ultrasonic sensors, and laser scanners to monitor all around the vehicle. In addition to these sensors, most of them uses online maps, which are updated simultaneously via cloud computing technology, to handle navigation and to have traffic information. Most of the systems are designed to work on highways in the beginning rather than urban areas, since the traffic flow is more stable in highways. However, there are many ongoing studies that working on autonomous driving in urban areas.

Almost all the companies are claiming that the more automated driving is the safer and more economical it becomes. As detailed in the previous section, about more than ninety percent of traffic collisions occur due to the drivers' error. With the aid of autonomous driving, this huge amount of errors will be decreased as low as it is possible with the aim of zero accidents. As an example, Google claims that their self-driving prototype had no accidents for 300,000 miles (Urmson, 2012). Full autonomous vehicles are still in development as prototypes. However, semi-autonomous vehicles are now on the market. Even some manufacturers made autonomous driving possible with software upgrades, i.e. Tesla (Tesla, 2015).

Besides all these technological improvements and explorations, there is also a legislation side of the autonomous driving. In the US, National Highway Traffic Safety Administration (NHTSA) published a policy (NHTSA, 2013) concerning about self-driving cars. The policy states five levels for self-driving vehicles, from no-automation to full self-driving automation. Level zero describes 'No-Automation' as the driver has all the control of brake, steering, throttle, and motive power as long as the vehicle moves on. Level one describes 'Function-Specific Automation', which claims this level of automation consists of at least one specific control options. Level two details 'Combined Function Automation' and it states the circumstance of two or more main control units are automated to work together. Level three stands for 'Limited Self-Driving Automation'. This level is described as follows; driver gives control to vehicle under certain conditions to perform safety-critical functions. The policy's last level, level four, is 'Full Self-Driving Automation' and details as this level of automated vehicles are the ones designed to handle all safety-critical driving functions for entire trip.

3. PHOTOGRAMMETRY AND DIGITAL IMAGE PROCESSING

3.1. Photogrammetry

As the main concern of the study is to examine applicability of photogrammetry to driver assistance systems, the technique is detailed below. Photogrammetry technique relies on central perspective projection geometry. The geometry is as shown in the Figure 3.1 below.

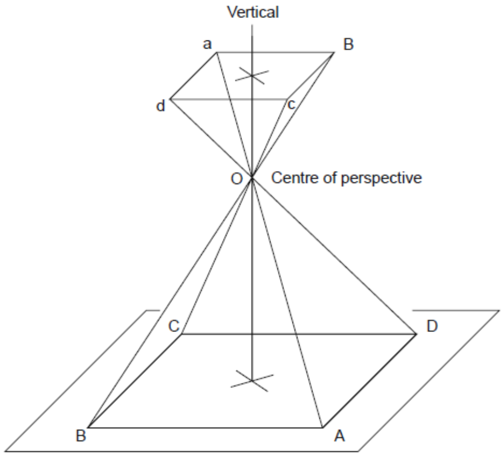


Figure 3.1 : Schematic representation of central perspective geometry (Gomarasca, 2009).

As seen in the Figure 3.1 above, point O is the center of perspective and 2D or 3D object space points are projected through this center onto a 2D plane. In photogrammetric approach, this geometrical solution is applied to camera and real world. The plane in the upper portion of the figure, which is the small one, can be considered as the image plane in the camera, and the distance between this small plane and center of perspective matches focal length of a camera’s lens. The plane in the bottom portion of the figure, which is the bigger one, can be considered as a real world plane. In stereophotogrammetry, the geometry consists of two central perspective geometry with an overlap, which is the three dimensionally producible part.

The Figure 3.2 below demonstrates object point P, image points p' and p" of P, projection centers of both cameras O_1 and O_2, focal length c, target of camera T, base distance B between cameras and coordinate systems of cameras which are centered at O_1 and O_2. The triangle of P, $O_{1,}$ and O_2 summarizes stereophotogrammetry very briefly.

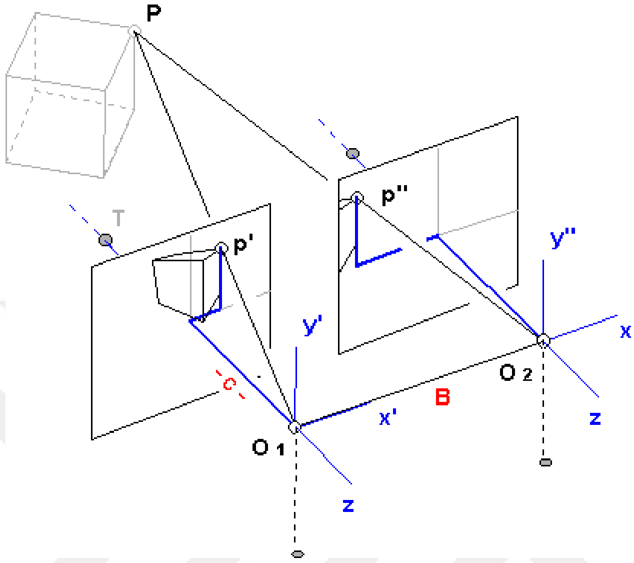


Figure 3.2 : Schematic representation of stereophotogrammetry (Haggrén, n.d.).

The reflectance of the real world object point P is measured on the stereo images' 2D planes as image points p' and p". However, these image measurements are affected by lens distortion and they are a part of camera's interior geometry. Therefore, it is compulsory to define interior geometry of camera and lens in interest, which consists of, focal length, sensor's dimensions, location of principle point, and lens distortions. All these components of a camera and its product, an image, are calculated with camera calibration. Camera calibration principally relies on shooting a network of well-known points that are precise and accurately determined in a 3D reference coordinate frame and making calculation based on this network to obtain all these components.

Focal length is the distance between the point of convergence in the lens and the image sensor. This distance is generally measured in millimeters and it is calculated in camera calibration process.

The sensor is one of the most important elements of a camera setup. It is essential to obtain the size of the sensor to form photogrammetric geometry. Moreover, the resolution of digital image sensor is necessary for a photogrammetric process.

Dimensions of a sensor are typically measured in millimeters and resolution is in pixels. Pixel size is also calculated along with these parameters, which is a size of a pixel and expressed in microns.

Principle point is the point where optical axis, which passes through the lens, intersects the image plane, or sensor. This point is considered as the image's center and distortion is zero at this point. Principle point (O) is also frequently known as projected optical center (perspective center) onto 2D image plane along camera optical axis, which is perpendicular to the image plane.

Lens distortion is misplacement of light arrays caused by lens imperfection. Lenses have two types of distortion, radial distortion and tangential distortion. To explain radial distortion, fish-eye effect is a good example. It is just a type of radial distortion. Tangential distortion occurs when lenses lean. These distortions are modelled in order to make the image measurements correctly. The Figure 3.3 represents interior orientation parameters such as; focal length, perspective center, and image coordinate system.

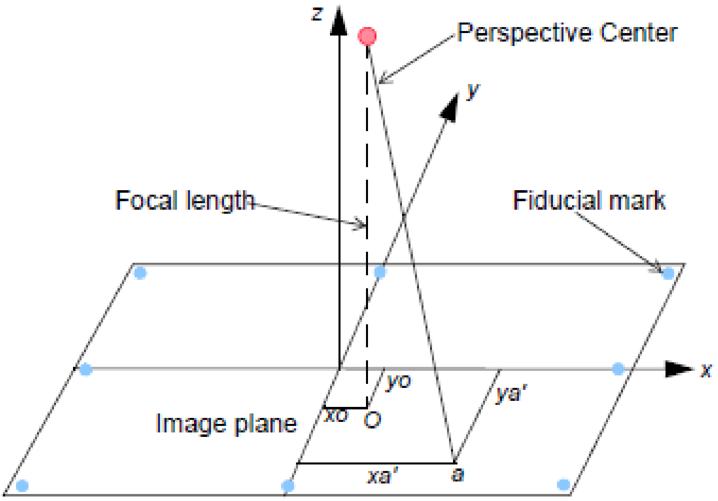


Figure 3.3 : Schematic representation of camera interior (HexGeoWiki, 2014).

In addition to interior parameters, exterior parameters of a camera is also required for photogrammetric evaluation. Exterior parameters are about position and rotations of the camera at the time of exposure. Position of the camera is needed to be known in three dimensions as X_0, Y_0, and Z_0 in an object coordinate system and rotations of the camera axis named as omega (ω), phi (φ), and kappa (κ), with respect to the X, Y, and Z-axes of that object coordinate system respectively. All these parameters are solved within a bundle block adjustment computation with an only aid of GCPs formerly. Currently, GPS aid is added much for the solution of these exterior parameters into this bundle block adjustment calculation. GPS are used to define center of projections seen in the Figure 3.4 below.

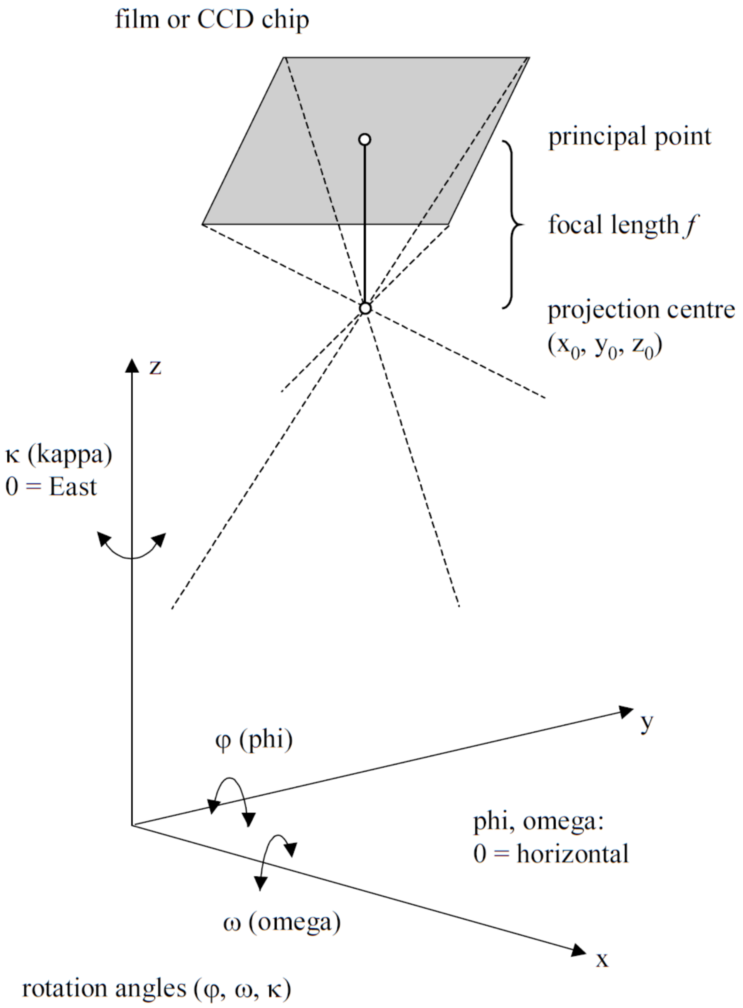


Figure 3.4 : Exterior orientation of an image sensor (Linder, 2006).

Interior and exterior orientations forms the central projection model through collinearity condition. The Figure 3.5 below describes interior orientation parameters with image plane.

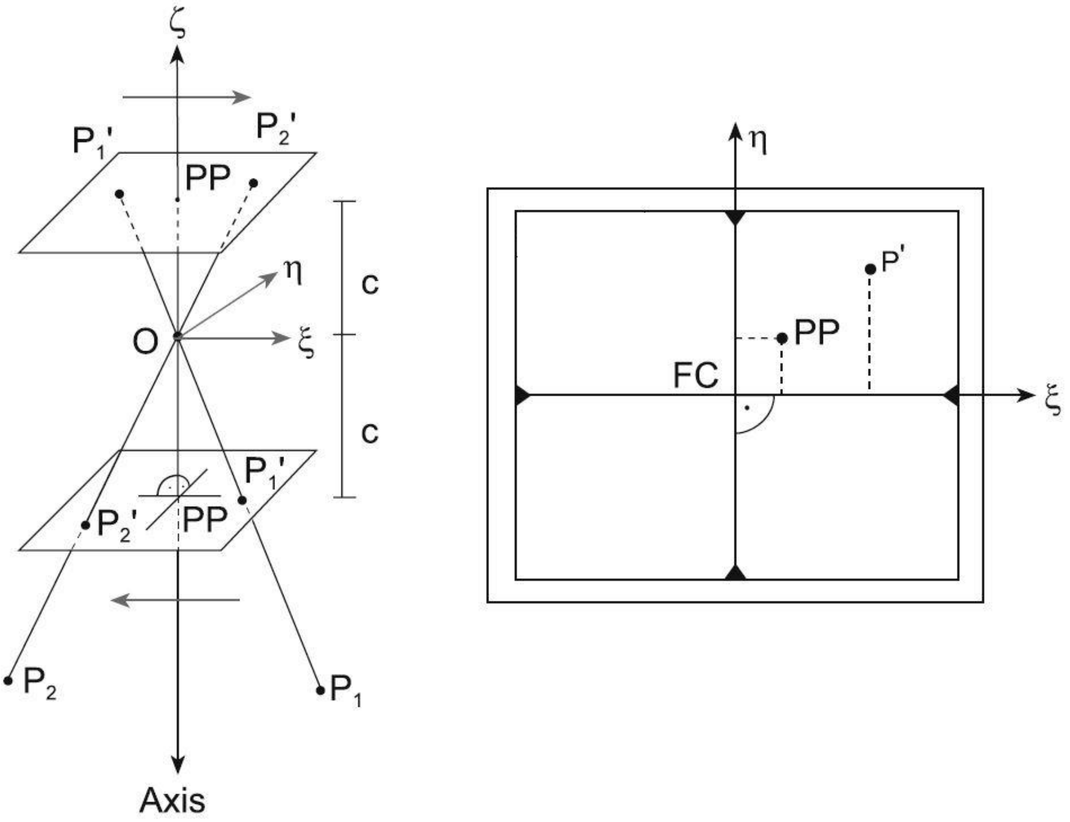


Figure 3.5 : Interior orientation geometry (Gomarasca, 2009).

The Figure 3.5 above leads to collinearity condition using the Equations **(3.1)** below.

$$\xi = \xi_0 - c\frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)}$$
$$\eta = \eta_0 - c\frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \qquad \textbf{(3.1)}$$

Mathematical model of photogrammetry is based on central projection as mentioned before. Thus, the relationship between image and real world coordinate systems are described as in the following Figure 3.6 below, where X, Y, Z are the coordinates in the object coordinate system, x, y, z are the coordinates in the 2D image coordinate system and, ξ_0, η_0, ζ_0 are image coordinates of principal point (Gomarasca, 2009).

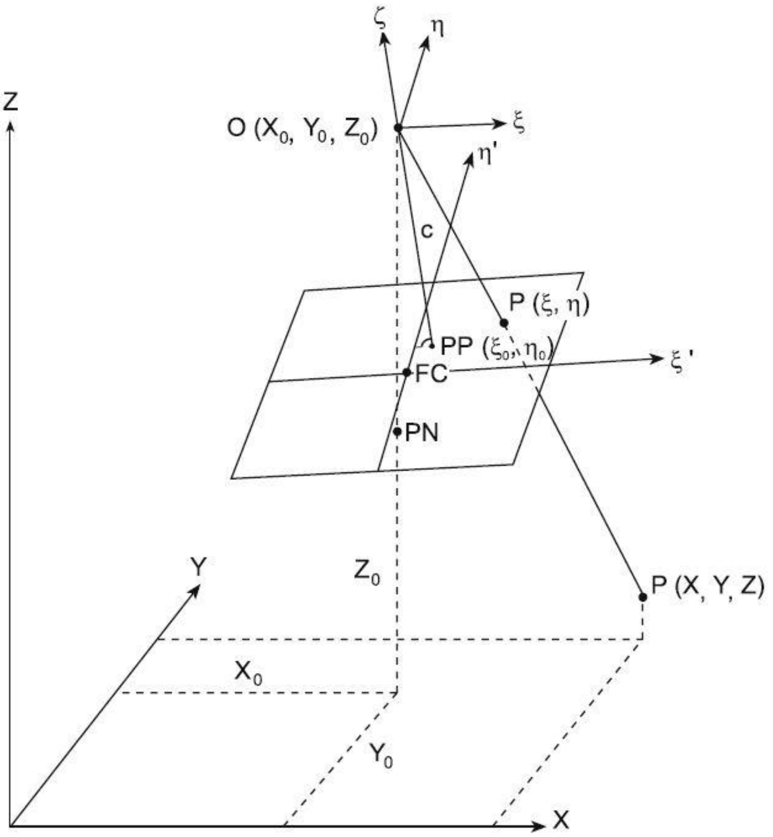


Figure 3.6 : Relationship between image and real world coordinates (Gomarasca, 2009).

According to the Figure 3.6 above, mathematical relationship between image and real world (or object) coordinate systems are as in the Equations **(3.2)** below (Gomarasca, 2009).

$$X = X_0 + (Z - Z_0)\frac{r_{11}(\xi - \xi_0) + r_{12}(\eta - \eta_0) - r_{13}c}{r_{31}(\xi - \xi_0) + r_{32}(\eta - \eta_0) - r_{33}c} \qquad \textbf{(3.2)}$$

$$Y = Y_0 + (Z - Z_0)\frac{r_{21}(\xi - \xi_0) + r_{22}(\eta - \eta_0) - r_{23}c}{r_{31}(\xi - \xi_0) + r_{32}(\eta - \eta_0) - r_{33}c}$$

Where r_{ij} are the elements of the rotation matrix detailed in the Equation **(3.3)** below.

$$R_{\omega\phi\kappa} = \begin{bmatrix} \cos\phi \ast \cos\kappa & -\cos\phi \ast \sin\kappa & \sin\phi \\ \cos\omega \ast \sin\kappa + \sin\omega \ast \sin\phi \ast \cos\kappa & \cos\omega \ast \cos\kappa - \sin\omega \ast \sin\phi \ast \sin\kappa & -\sin\omega \ast \cos\phi \\ \sin\omega \ast \sin\kappa - \cos\omega \ast \sin\phi \ast \cos\kappa & \sin\omega \ast \cos\kappa + \cos\omega \ast \sin\phi \ast \sin\kappa & -\cos\omega \ast \cos\phi \end{bmatrix} \qquad \textbf{(3.3)}$$

The angles of rotation matrix are as follows; ω stands for primary axis' rotation, ϕ for secondary axis' rotation, and κ for tertiary axis' rotation.

3.2. Digital Image Processing

Digital image processing has two tasks in this project. First task is to determine and locate the object in both stereo images and latter is to make stereo image measurements. Actually, it was also possible to make image measurements without specific object detection. However, this scenario would cause many points to process. In this case, image measurements for both entire stereo images and 3D point production could affect performance of the system in a negative way. This is the reason, why system is designed to make image measurements and 3D photogrammetric evaluation in a limited area determined in both images.

There are many different applications with digital image processing such as; thresholding, filters, edge detection, smoothing, shape description, object recognition, object detection, color transformation, segmentation, corner point detection, and other different image interpreting techniques.

In this thesis study, thresholding, smoothing and connected components interpretation methods are used to object detection purposes. While, point detection, feature extraction, and matching functions are employed for image measurement purposes.

Object detection algorithms are based on characteristics of the objects. These characteristics include; shape, texture, color etc. (Cyganek, 2013). In this study, image-processing techniques are used as tools to detect vehicles in every frame captured by stereo camera setup on a vehicle and provide image measurements for later sequential stereo photogrammetric processes.

Object detection has always been a challenge in image processing. Different image processing algorithms have been developed to handle object detection. One of them is Viola-Jones Algorithm (Viola & Jones, 2001), which is used widely. The algorithm is said to be a machine learning approach. One of the common usages of the algorithm is face detection, as seen in the Figure 3.7 below.

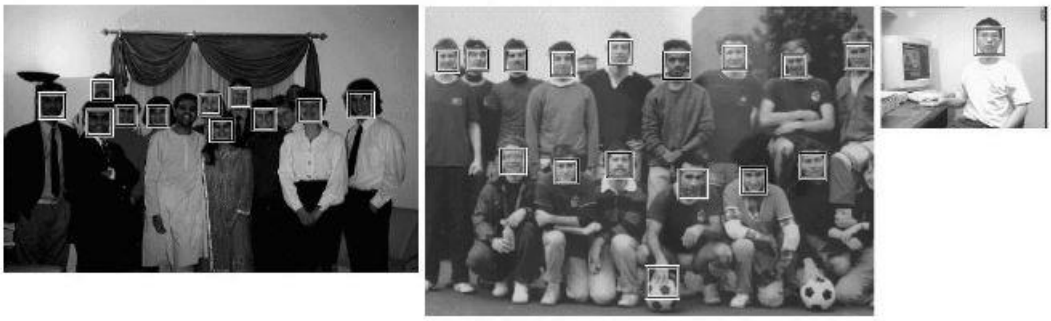


Figure 3.7 : Test outputs of Viola-Jones Algorithm (Viola & Jones, 2001).

Viola-Jones Algorithm computes an integral image, which consists of speedily calculated box structures expressing the image. These integral images are calculated with the Equation **(3.4)** below, where x, and y are the coordinates of the pixels (Viola & Jones, 2001).

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y') \qquad \textbf{(3.4)}$$

These calculations are combined with other specific mathematical operations and in the end; objects became possible to be detected with those representations as shown in the Figure 3.8 below.

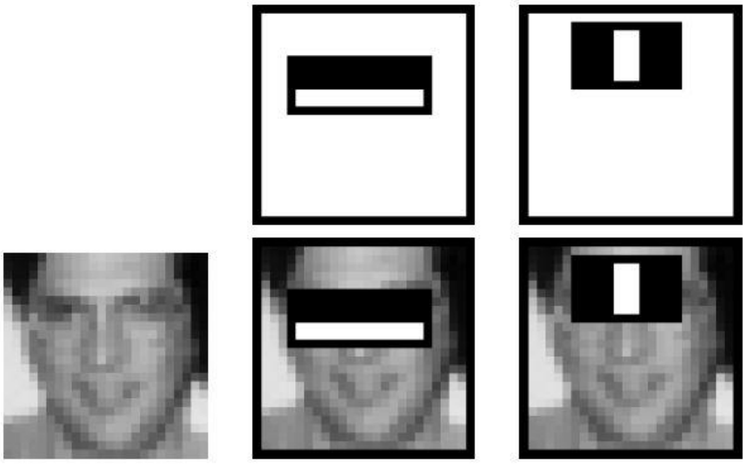


Figure 3.8 : Representative images from Viola-Jones Algorithm (Viola & Jones, 2001).

Viola-Jones algorithm is possible to implement in this study using MATLAB’s cascade object detector training system object (MathWorks, 2015a). Yet, a custom object detection approach is developed for test purposes and detector training is left for future work. The custom object detection function developed using thresholding, smoothing and connected components analysis.

The used thresholding function is ‘im2bw’, which is a built-in function in MATLAB. The function takes the inputs of a color or gray image and transforms it to a binary image by thresholding. The function can be used with either automatic threshold level or a given threshold level. The example in the webpage of software producer is given in the Figure 3.9 below.

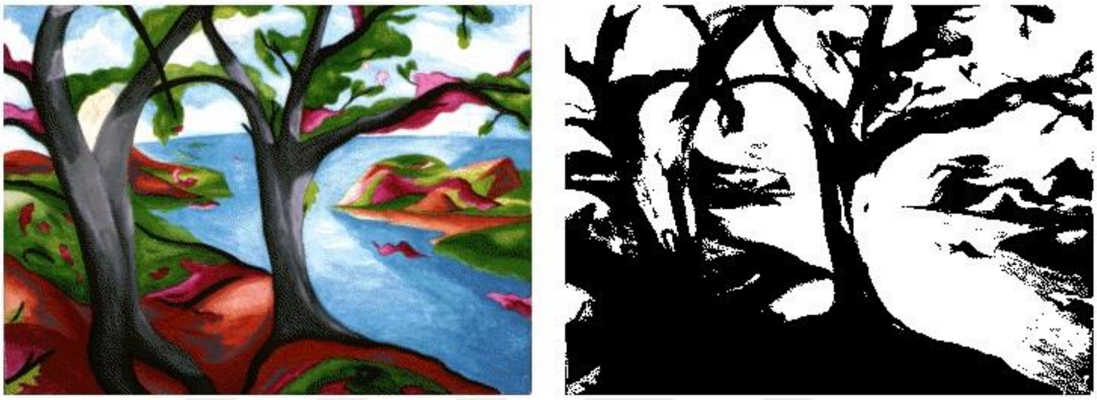


Figure 3.9 : Automatic thresholding a color image (MathWorks, 2015b).

In the Figure 3.9 above, the color image, on the left, is processed with automatic threshold using ‘im2bw’ function. The output of the process is given on right.

For image smoothing purposes, again, a built-in function in MATLAB ‘wiener2’ is used. This function is an adaptive noise cancellation that can output both noise and filtered image. The function is able to take neighborhood dimension and noise. If the noise is given as input, the function outputs only the filtered image. Otherwise, it detects the noise and outputs is, too. The function estimates local means and creates a filter based on these mean values. The Equation **(3.5)** below represents calculation of variance and local mean pixel-by-pixel, where, η is N-by-M local neighborhood of pixels (MathWorks, 2015c).

$$\mu = \frac{1}{NM} \sum_{n_1, n_2, \in \eta} a(n_1, n_2) \tag{3.5}$$

An example of such noise removal is shown as in the Figure 3.10 below by software documentation web page.

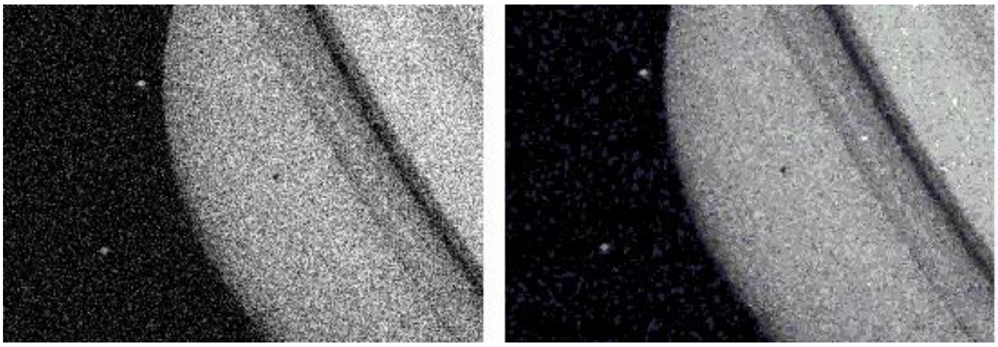


Figure 3.10 : Noise removal from a noisy image (MathWorks, 2015d).

The image on the left in the Figure 3.10 above is the one with noise, while the one on the right is noise-cancelled using ‘wiener2’ function.

Connected component analysis is handled via MATLAB’s built-in function ‘bwconncomp’. The function catches linked components in a binary image. The function takes binary image as input and outputs number of objects, connectivity of components, size of the binary image, and locations of caught objects. The algorithms seeks for an unlabeled pixel, then, with a flood-fill algorithm, it labels pixels of the caught object. These iterations goes on until all the pixels are labeled. Following example is given in the official web page for the function.

The Figure 3.11 below consists of two images. The one on the left is the original one. The one on the right is processed one. The process consists of connected component analyze and the largest component is erased with the help of ‘bwconncomp’ function.

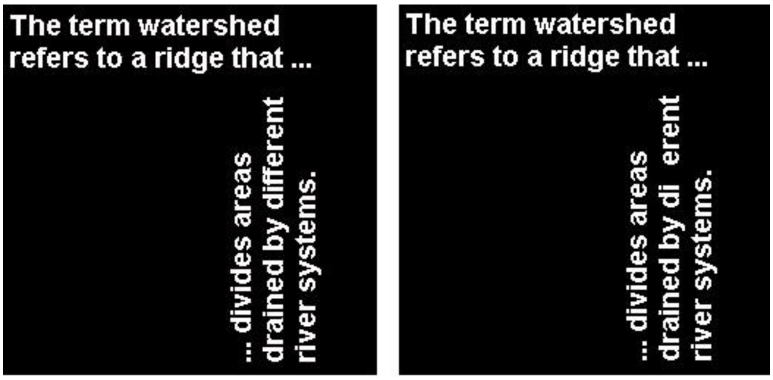


Figure 3.11 : Erasing the largest object in the binary image (MathWorks, 2015e).

Image measurements in the region of interest then take place after the detection of the object points the custom object detection function, point detection, feature extraction and matching.

For point detection, various alternatives exists, i.e. SURF (Speeded Up Robust Features), BRISK (Binary Robust Invariant Scalable Keypoints), FAST (Features from Accelerated Segment Test), Harris-Stephens, MSER (Maximally Stable Extremal Regions) etc. However, after some trials, SURF features are seen as the one with best results for this project (Bay, Ess, Tuytelaars, & Van Gool, 2008).

The built-in ‘detectSURFFeatures’ function in MATLAB takes inputs of a grayscale image, metric threshold value for stronger point detection, number of octaves for point search area, number of scale levels per octave to adjust scale increments, and region of interest to define the working area. The function outputs objects’ details, count, location, scale, metric value, sign of Laplacian value, and orientation of features. Metric is used to describe strength of the features. Sign of Laplacian value stands for intensity values’ representation, which is used for feature matching. If metrics for two features are the same but their sign of Laplacian values are different, this means those features do not match each other. An example from MATLAB’s web page is given below.

The Figure 3.12 below shows detected SURF features with green points with a plus sign and green circles around them.

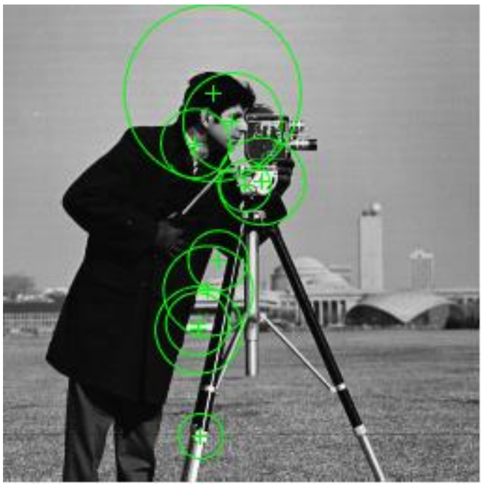


Figure 3.12 : Detected SURF Features on the image (MathWorks, 2015f).

Due to mismatches of detail points, the strongest detected features are desired in this project. This is done by ‘selectStrongest’ property of the points class (MathWorks, 2015g). This function simply selects the points with the strongest metrics, and outputs these points.

Feature extraction is done to gather valid points from detected features with built-in ‘extractFeatures’ function (MathWorks, 2015h). This function takes binary or intensity image and points as inputs. The outputs of the function are the features and valid points.

Extracted features are matched using ‘matchFeatures’ function in MATLAB (MathWorks, 2015i). This function is designed to take input of features and it outputs indices for matching features and match metrics. Indices are the representations of the points and match metrics are the distances between these matched points.

4. CASE STUDY

The aim of this thesis study is to provide three-dimensional position information of the objects ahead of a vehicle on the road for a driver assistance system using stereophotogrammetry technique. The main concern about using stereophotogrammetry was the image measurements. Image measurements in photogrammetry are held by operator using computers or stereo plotters as detailed previously. Nevertheless, a driver assistance system needs to work in real time in order to work properly and meet the demand. For this purpose, image measurements and photogrammetric evaluation had to be handled in real time. Digital image processing techniques are used to find most accurately matching details appearing in each stereo-image pairs to make image measurements automatically and also photogrammetric calculations done using these measurements for driving assistant systems.

The necessary sample data were collected with pre-designed scenario in the field. Moreover, a computer program was developed to make all the necessary image-processing steps, photogrammetric processes, and display processes to warn the driver. This section details equipment of the study, photogrammetry concept and the way it is used in the project, digital image processing step of the project, field works, and the development of a computer program with a graphical user interface (GUI).

4.1. Equipment of the Study

In this study for a driver assistance system development proposes using stereo-photogrammetry, two DSLR (Digital Single Lens Reflex) cameras are used for shooting, and a total station set is used to build a control network and make measurements on a vehicle (a car) on the road. Usages of these equipment will be detailed in the field works section.

The cameras are the main components of the system. There are two Nikon D5200 cameras used here (Figure 4.1). Both were with their 18-105mm kit lens. The cameras were equipped with a 23.5mm x 15.6mm CMOS sensor, housing 24,710,000 pixels, while 710,000 of pixels were not being used. Cameras are able to capture images in 6000 x 4000 resolution, which ends up with 24,000,000 pixels in total. In each sensor, there are 39 focus points those making the camera focus fast. Cameras are also providing 30 FPS video in 1920 x 1080 resolution.

Figure 4.1 : Nikon D5200 DSLR Camera with 18-105mm kit lens.

Cameras attached to a chipboard with ~15cm base distance in the beginning, later the base distance is decided to be ~150cm. Reason of this will be detailed in the upcoming sections.

Total station used in the study was a Spectra Precision FOCUS 8 Series. Angle accuracy of the instrument is between 2”-5”. Device is able to measure with the accuracy of ± (2 + 2 ppm x distance) mm with prism. Without using the reflector, it can measure with the accuracy of ± (3 + 2 ppm x distance) mm. Inside the device, Windows CE operating system was installed. The shortest possible range is 1.5 m for the device. Measuring intervals are about 1.5 seconds for precise mode, while in normal mode it can measure under 1 second, using prism. Reflectorless operation takes ~2 seconds in precise mode, and ~1 second in normal mode. Least count given in the datasheet is 1mm for reflectorless precise mode and 10mm for reflectorless normal mode. In this study, device is mostly used in reflectorless precise mode. All the information above is taken from official datasheet and can be reached online (Trimble, 2014).

In addition to these devices, a car is needed because the image processing technique will be used to detect that car appearing in images as an object ahead of the vehicle where the stereo-cameras are mounted on its dashboard. All these processes are coded in MATLAB environment.

4.2. System Design

The system is designed to show that photogrammetry is viable technique to be used to supply data for a driver assistance system.

First, technical expectations in Table 4.1 below were calculated with respect to a series of experimental camera setups designed for the field tests in this study. These calculations include pixel size on target, planimetric accuracy, and depth accuracy. Table 4.1 below shows possible outputs based on these setups.

Table 4.1 : Expectations from setup.

Pixel Size (micron)	Focal Length (mm)	Base length (m)	Object Distance (m)	Image Measurement Accuracy (% of pixel)	Pixel Size on Target (mm)	Planimetric Accuracy (mm)	Depth Accuracy (cm)
3.85	18.00	0.15	50.00	50%	10.69	5.35	178.24
3.85	18.00	0.15	100.00	50%	21.39	10.69	712.96
3.85	50.00	0.15	50.00	50%	3.85	1.93	64.17
3.85	50.00	0.15	100.00	50%	7.70	3.85	256.67
3.85	50.00	1.50	50.00	50%	3.85	1.93	6.42
3.85	50.00	1.50	100.00	50%	7.70	3.85	25.67

Pixel size, focal length, base length, object distance, and image measurement accuracy are all inputs for these calculations. All equations (**4.6)**, (**4.7)**, and (**4.8)** below are based on central projection geometry ("How Accurate is Photogrammetry? - Part 2," 2010).

Pixel size on target is,

$$Pixel\ Size\ on\ Target = \frac{Object\ Distance}{Focal\ Length} * Pixel\ Size\ of\ Image \qquad \textbf{(4.6)}$$

Planimetric accuracy is,

$$Planimetric\ accuracy = Image\ Measurement\ Accuracy * Pixel\ Size\ of\ Image \tag{4.7}$$

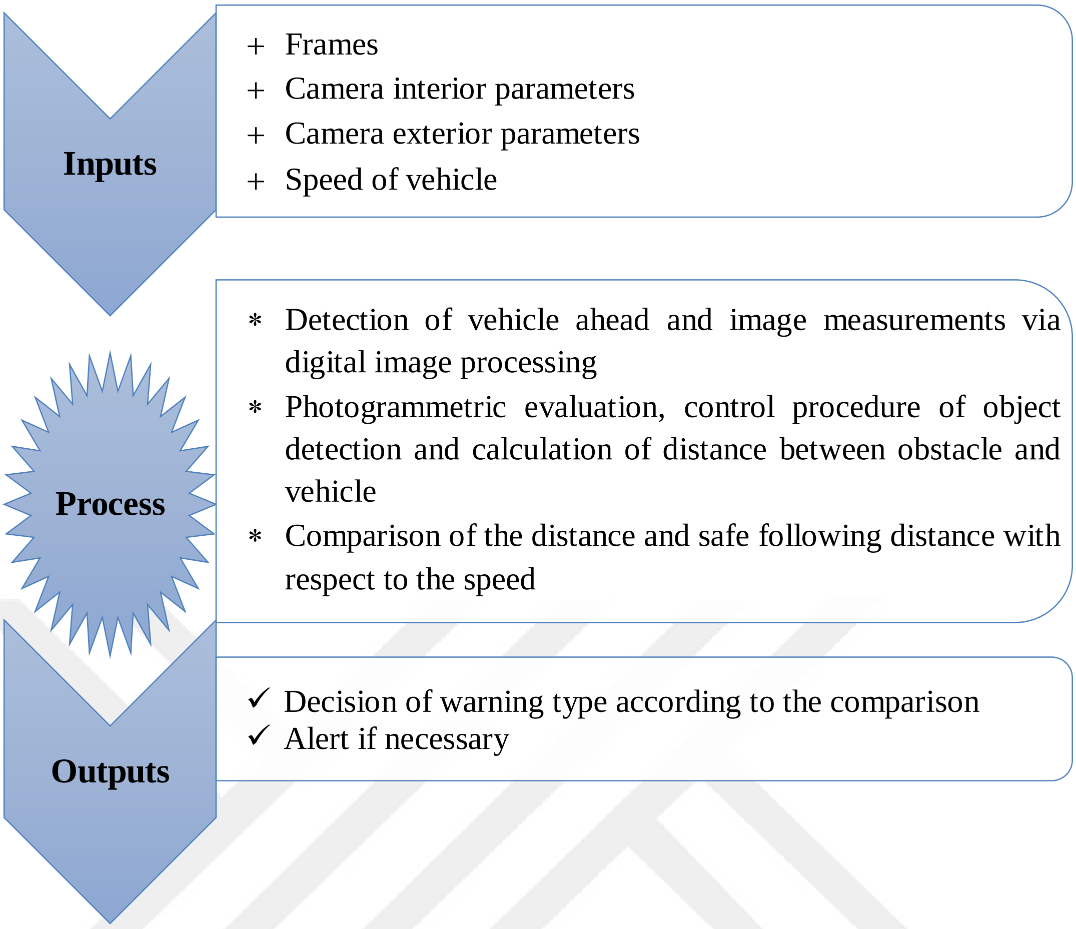
Depth accuracy is,

$$Depth\ Accuracy = \frac{Object\ Distance}{Base} * Planimetric\ Accuracy \tag{4.8}$$

The calculated values in the Table 4.1 above are expectations from a perfect geometry. The best fitting results of this experimental work fulfilling the main purpose of this thesis as it can be seen from the table above, which shows that the most suitable camera setup is the setup with 50mm focal length and 1.5-meter base length. The geometry is defined by interior and exterior parameters of cameras. Problems about camera calibration process are detailed in the following sections, which defines interior and exterior orientation parameters.

The system with the power of software which is developed in the study is able to; load images from both cameras for stereo-image processing at every certain seconds, to use the interior and exterior orientation parameters determined for both cameras, to make stereo image measurements of matched details appearing on the relevant stereo images pair, to automatically generate three-dimensional coordinates of these detail points found and matched on these stereo pair by the help of stereo-photogrammetry, to calculate distances between these details which are possibly the parts of an obstacle ahead the vehicle and the vehicle itself where the system is installed, to compare the distance with safe following distance of the vehicle speed, and finally set a warning signal corresponding to this comparison.

As seen in the Figure 4.2 below system has three main stages as follows, inputs, processing, and output.



- + Frames
- + Camera interior parameters
- + Camera exterior parameters
- + Speed of vehicle

- * Detection of vehicle ahead and image measurements via digital image processing
- * Photogrammetric evaluation, control procedure of object detection and calculation of distance between obstacle and vehicle
- * Comparison of the distance and safe following distance with respect to the speed

- ✓ Decision of warning type according to the comparison
- ✓ Alert if necessary

Figure 4.2 : Schematic representation of system design.

Inputs of the system contain frames, camera interior and exterior parameters, and speed of the vehicle for the real world test of the system. For this test purposes, frames are gathered as images in full resolution that the camera can provide, which is specified in the previous section. Interior and exterior orientation parameters of the cameras are collected from single camera calibration (for both cameras individually) and stereo-camera calibration processes in a lab environment and in a test field respectively. Speed of the vehicle is entered to the system as a constant value. However, this variable can be provided in real-time from vehicle’s computer or a navigation system.

The main concerns of the system are detection of the vehicle ahead and making image measurements of matched details on both stereo pairs from the run of developed digital image processing functions. Image measurements are held only for the matched details in the concerned area, which is the area of detected obstacle in the image. In addition to these, photogrammetric procedure takes place to produce the 3D coordinates of the detail points obtained on the obstacle by image matching algorithms, and with these

3D points, calculation of distance between the obstacle and the vehicle becomes possible. However, object detection with image processing may be mistaken. For this reason, a control procedure developed to check whether these detail points are the parts of the road or not. This procedure makes some more measurements in the area below the detected object and calculates average road height from these road points. Then, this road height is compared with the height of detected object's points. If the height of object's points are in between given range above the road level, then those points are treated as object points, other points will be removed from the main process. After the control procedure, safe following distance is calculated according to the speed of the vehicle and compared to the distance between obstacle and vehicle.

Several types of warnings are then sent to the system display with regard to the comparison results to warn the driver if necessary. If the distance is more than the safe following distance, the output no need to return a warning. Other than this situation, is detailed in the following sections.

4.3. Camera Calibration and Determination of Interior Parameters

All mathematical processes have been automated using different photogrammetry softwares. However, all these softwares are in need of an operator to perform. The operators load images, give camera parameters, and make image measurements. However, for driver assistance systems, photogrammetric process must be automatic. For this purpose, some specific algorithms are needed to handle the operators' job. In this thesis study, these algorithms are developed to clarify if photogrammetry technique can be used for driver assistance systems or not. With this aim, camera calibration process was performed to get camera parameters.

Camera calibration for such applications is one of the important photogrammetric steps depending on the study type (Asri, Çorumluoğlu, & Kalaycı, 2010). Some systems are designed with multiple sensors, such as GPS, IMU, laser scanners etc., while some others are designed with cameras only. This study intends to use cameras only to detect a possible obstacle ahead and measure the distance between. Thus, camera calibration is a critical step to produce precise results.

The interior parameters of both cameras used in the study are obtained through camera calibration process in PhotoModeler (EOS, n.d.) environment with both single sheet, and multi-sheet calibration methods depending on preferred focal lengths in the study. The methodology of calibration is changed due to the field of view, which is directly affected by preferred focal lengths. As the focal length goes up, the field of view becomes smaller at the certain distance to camera. Consequently, multiple sheets become too difficult to shoot in such a small area, the distance between camera and sheets needs to be more. Multi-sheet calibration method is used for focal lengths of 15mm, 35mm, and 50mm. While, single sheet calibration is done for 50mm, and 70mm focal lengths. The effect of focal length change experienced in the study will be detailed in the following sections.

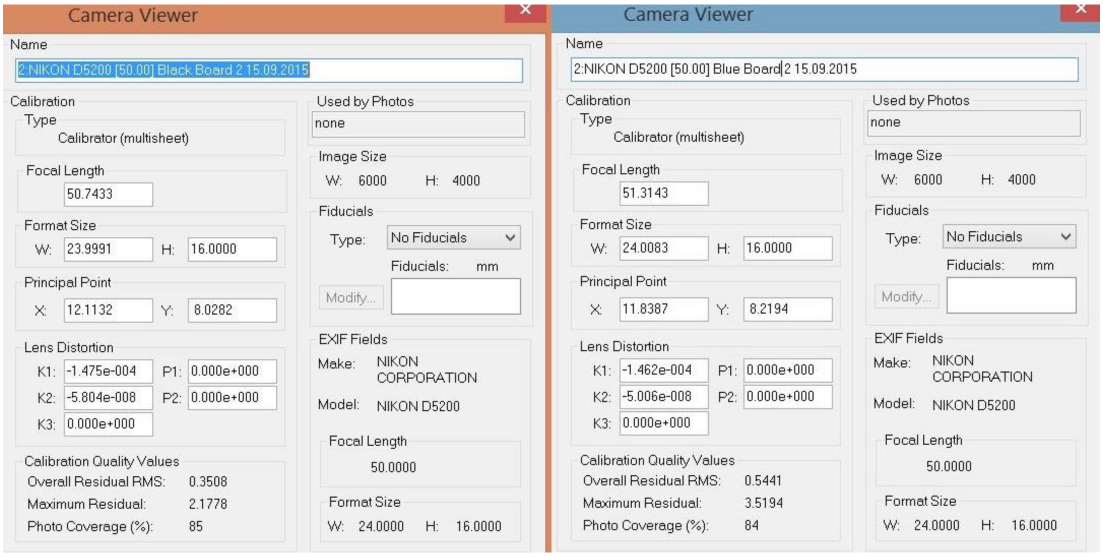
The main problem encountered here is that the software takes sensor information, both sensor size and resolution, from images' EXIF (Exchangeable Image File Format) data, then it makes calculations with these data. This became a huge problem for these cameras because Nikon put slightly different information into the data of an image taken by it. The sensor size is 23.5 mm x 15.6 mm as given in the technical specifications table provided by the manufacturer (Nikon, n.d.). However, this information is placed in the image data file as rounded integers as 24.0 mm x 16.0 mm as seen in the screen shots in Figure 4.3 below. In addition to this, the camera does not use all the pixels in the sensor. According to manufacturer, the sensor has 24.71 million pixels in it (Nikon, n.d.). Nevertheless, the image output of the camera is exactly 24 million pixels, which is the highest available resolution in the settings menu of the camera. The error in sensor area is 17.4mm^2, which is about 5% of the real sensor area. Beside this error in the sensor's size, the error in resolution is about 3% of the real resolution (Table 4.2). All these improper values in the image data cause an error in the pixel size calculation of the software. Thus, all the interior geometry is solved with errors when the PhotoModeler software is used for the camera calibration.

Table 4.2 : Nikon D5200 Camera Specifications Analysis.

Nikon D5200 Camera Specifications Analysis		
	Data in image	**Sensor's Specifications**
Sensor Width (mm)	24.00	23.50
Sensor Height (mm)	16.00	15.60
Sensor Area (mm^2)	384.00	366.60
Error (mm^2)	17.40	
Error * (%)	-4.75%	
Resolution (pixels)	24,000,000	24,710,000
Error in Resolution (pixels)	710,000	
Error in Resolution * (%)	2.87%	
Pixel Size (micron)	4.00	3.85
* Percentages are calculated as (1-(Data in image / Sensor’s Specification))		

The errors above caused the software to calculate the sensor size different from the real sensor size, which is given by the manufacturer in the technical specifications of the camera. Dimensions of the sensors are possible to see in the figure below under ‘Format Size’ title. To overcome this problem the best choice would be changing the sensor size values in the image data, at least. Nonetheless, PhotoModeler software was careful about any changes in image data. As well, the software does not allow user to edit these data. Hence, all processes are carried out in this way.

Another faced problem is about lenses. Even the lens is fixed at a certain focal length using a tape to hold the objective at the focal length, the focal length saved in the image data has changed sometimes in continuous shooting. Because of this, some of the images were rejected by the software. Therefore, these photographs are manually added into the calibration project, despite the software’s warnings, and outputs are as shown in the Figure 4.3 below.


Camera Viewer

Name: 2:NIKON D5200 [50.00] Black Board 2 15.09.2015

Calibration
Type: Calibrator (multisheet)
Focal Length: 50.7433
Format Size: W: 23.9991 H: 16.0000
Principal Point: X: 12.1132 Y: 8.0282
Lens Distortion: K1: -1.475e-004 P1: 0.000e+000 K2: -5.804e-008 P2: 0.000e+000 K3: 0.000e+000
Calibration Quality Values
Overall Residual RMS: 0.3508
Maximum Residual: 2.1778
Photo Coverage (%): 85

Used by Photos: none
Image Size: W: 6000 H: 4000
Fiducials: Type: No Fiducials; Fiducials: mm
EXIF Fields: Make: NIKON CORPORATION; Model: NIKON D5200
Focal Length: 50.0000
Format Size: W: 24.0000 H: 16.0000

Camera Viewer

Name: 2:NIKON D5200 [50.00] Blue Board2 15.09.2015

Calibration
Type: Calibrator (multisheet)
Focal Length: 51.3143
Format Size: W: 24.0083 H: 16.0000
Principal Point: X: 11.8387 Y: 8.2194
Lens Distortion: K1: -1.462e-004 P1: 0.000e+000 K2: -5.006e-008 P2: 0.000e+000 K3: 0.000e+000
Calibration Quality Values
Overall Residual RMS: 0.5441
Maximum Residual: 3.5194
Photo Coverage (%): 84

Used by Photos: none
Image Size: W: 6000 H: 4000
Fiducials: Type: No Fiducials; Fiducials: mm
EXIF Fields: Make: NIKON CORPORATION; Model: NIKON D5200
Focal Length: 50.0000
Format Size: W: 24.0000 H: 16.0000

Figure 4.3 : Output of camera calibration process.

4.4. Stereo Camera Calibration and Determination of Exterior Orientation Parameters

Exterior orientation parameters are other key inputs for this study. The system is designed to produce three-dimensional coordinates of the concerned points by stereophotogrammetry and since there are only two cameras in the system design, it means no adjustment calculation opportunity. For adjusted computation of the parameters, there must be more measurements than the required amounts, like, three-dimensional coordinates of a point can only be calculated via at least a stereo pair. In addition, the errors in interior orientation parameters affect all exterior orientation parameters, too. Because exterior orientation parameters can only be calculated with known interior orientation parameters.

Exterior orientation parameters of the stereo setup are calculated with a designed control network. Thus, a network consisting of 21 control points and 18 tie points, 39 points in total was designed and used in the study. Control points are located with the total station and point list is transferred to PhotoModeler and used as fixed points. Stereo image couples were processed to gather exterior orientation parameters.

The Figure 4.4 below shows the calibration network from one point of view. The 3D process is done with 1.23 pixels of error and exterior orientation parameters (Table 4.3) are obtained with respect to the control network.

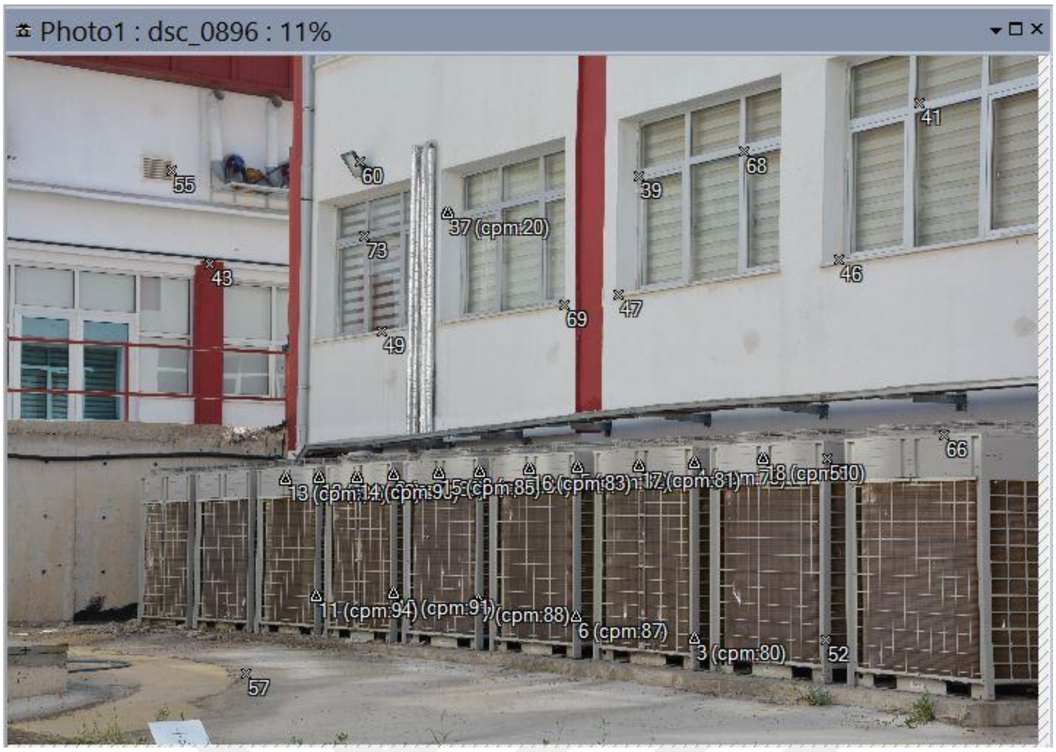


Figure 4.4 : Image measurements of control network.

Table 4.3 : Exterior orientation parameters of stereo-cameras of the last study.

	Left Camera	**Right Camera**
Omega	94.035	93.947
Phi	-23.880	-22.518
Kappa	1.849	1.874
X	997.258	998.634
Y	1,012.724	1,012.110
Z	997.844	997.846
Omega, Phi and Kappa are in degree X, Y and Z are in meters		

The distance between cameras, base length, is measured as 1.51m with steel measuring tape, and the data from the above network design outputs as 1.506m, which confidently matches and confirms the measured value.

4.5. Code and Procedures for Photogrammetric Process

Stereo-photogrammetric process is handled via MATLAB code, which is taken from ‘GPS Supported Mobile Terrestrial Photogrammetry System’ PhD thesis (Asri, 2011) and optimized for this study. The optimize code takes inputs of, camera parameters, exterior orientation parameters, image measurements and outputs three-dimensional coordinates of the corresponding points as a list. One of the optimizations done in the code is about image measurements. The code is used to take image measurements as matrices for each image. However, for this study, this part of the code is changed to obtain image measurements automatically or manually. Automatic image measurements are detailed in image processing title and manual image measurements are detailed below. The code was originally designed to give outputs in a .txt (ASCII Text File) file and changed to manage outputs directly in the MATLAB environment or in a .csv (Comma Separated Values) file (Figure 4.5). Other optimizations done to the code are mainly about performance enhancements.

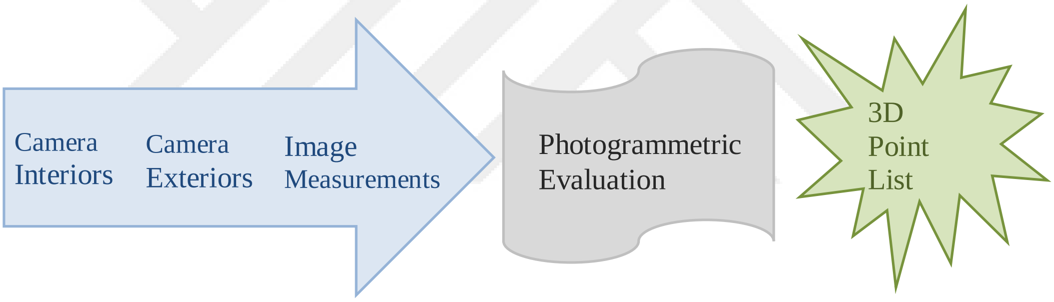
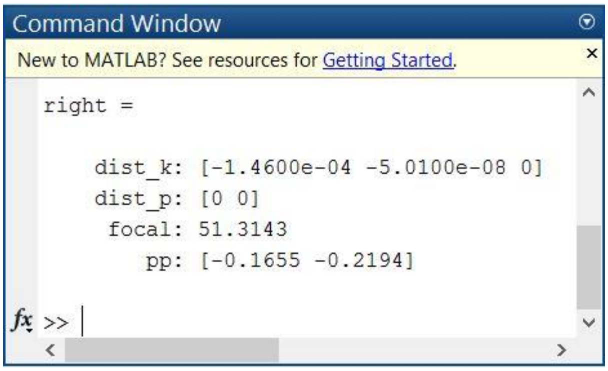


Figure 4.5 : Photogrammetric Evaluation Flowchart.

Camera interior orientation parameters are stored in a .mat file as a struct in MATLAB environment. Each camera parameters are stored in different structs with four fields as seen in the Figure 4.6 below. These fields stand for radial distortion, tangential distortion, focal length, and coordinate of principle point.



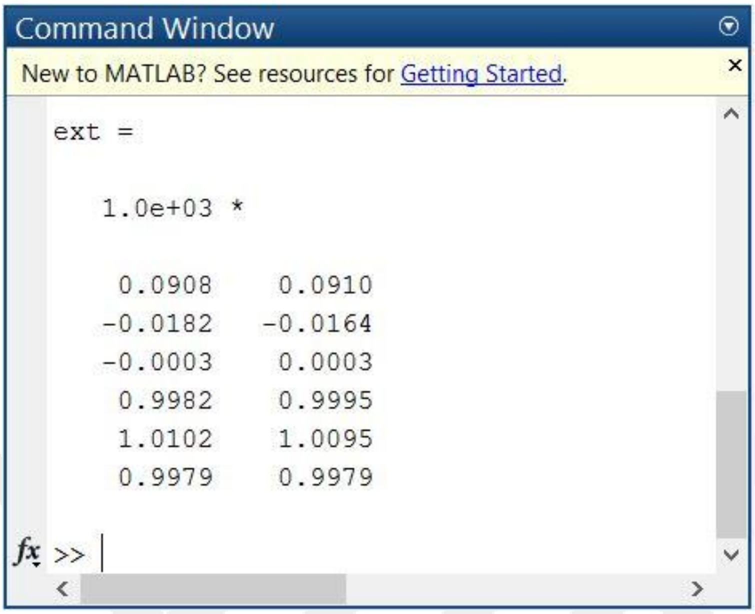
```
right =

    dist_k: [-1.4600e-04 -5.0100e-08 0]
    dist_p: [0 0]
     focal: 51.3143
        pp: [-0.1655 -0.2194]

fx >>
```

Figure 4.6 : Camera parameters stored in struct.

Exterior parameters are also stored in .mat file. Differing from camera interiors, exterior parameters are directly stored as a matrix. Each column represents one camera station and rows are placed as follows; omega, phi, kappa, x, y, and z (Figure 4.7).



```
ext =

   1.0e+03 *

    0.0908    0.0910
   -0.0182   -0.0164
   -0.0003    0.0003
    0.9982    0.9995
    1.0102    1.0095
    0.9979    0.9979

fx >> 
```

Figure 4.7 : Exterior parameters stored as matrix from a different camera setup.

Image measurements are not needed to be stored. They are automatized with digital image processing techniques and it will be detailed in the following section. To control the system and developed software, a control environment was designed with several signed marks at top of surveying poles. Then, to compare the locations of the car and control points distributed in the control environment, it was necessary to make image measurements manually, in the MATLAB environment with graphical user interface (GUI). The GUI simply works on a figure, lets the operator zoom or pan before measurement. After zoom and pan operations, it is expected from the operator to hit a key from keyboard to activate the marker to measure. This process is shown in the Figure 4.8 below. The output of this step is directly taken as input for the next step and not stored.

Output of the algorithm is shown in the Figure 4.8 below. Unit of output is meter and one space is placed between coordinates in the output file. Each row stands for a point while columns are in order of X, Y, and Z coordinates.

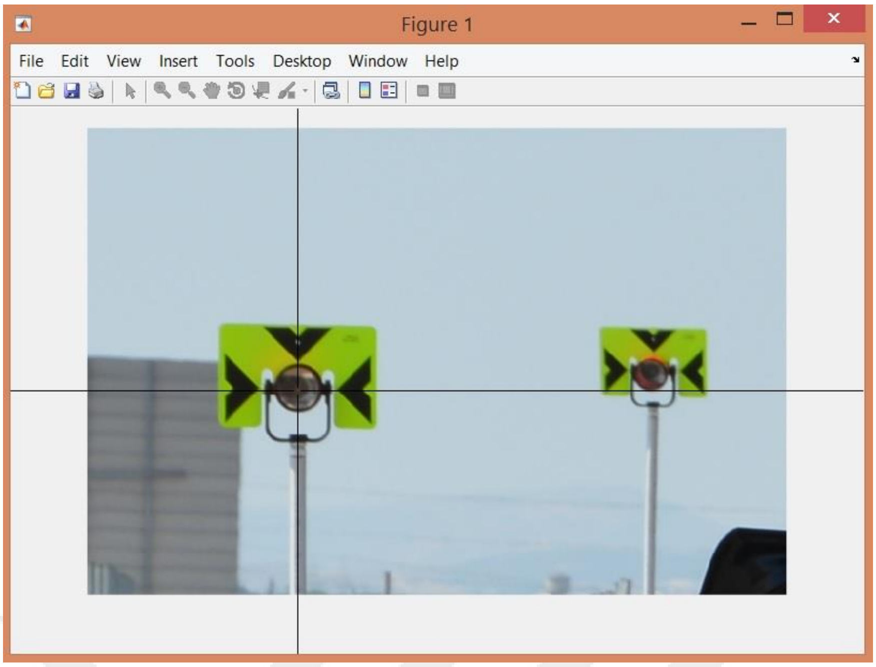
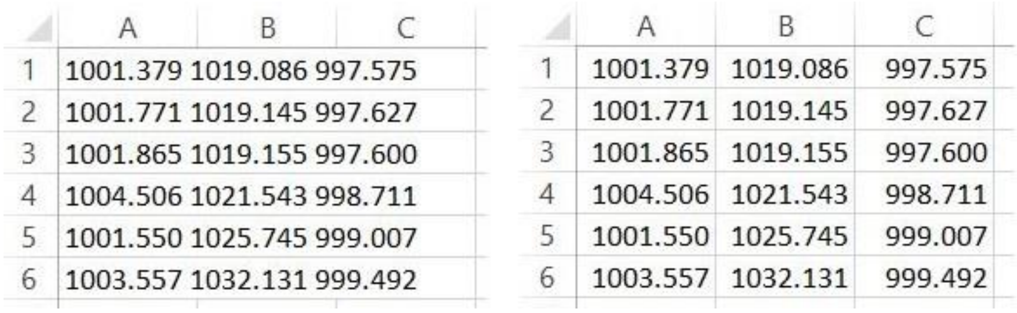


Figure 4.8 : Manuel image measurements in MATLAB.

The file format of the output is chosen as .csv in order to make it usable for many software. However, to use these values in office software, such as Excel, each data must be in another cell. So, 'Text to Columns' function is used to split these data (Figure 4.9).



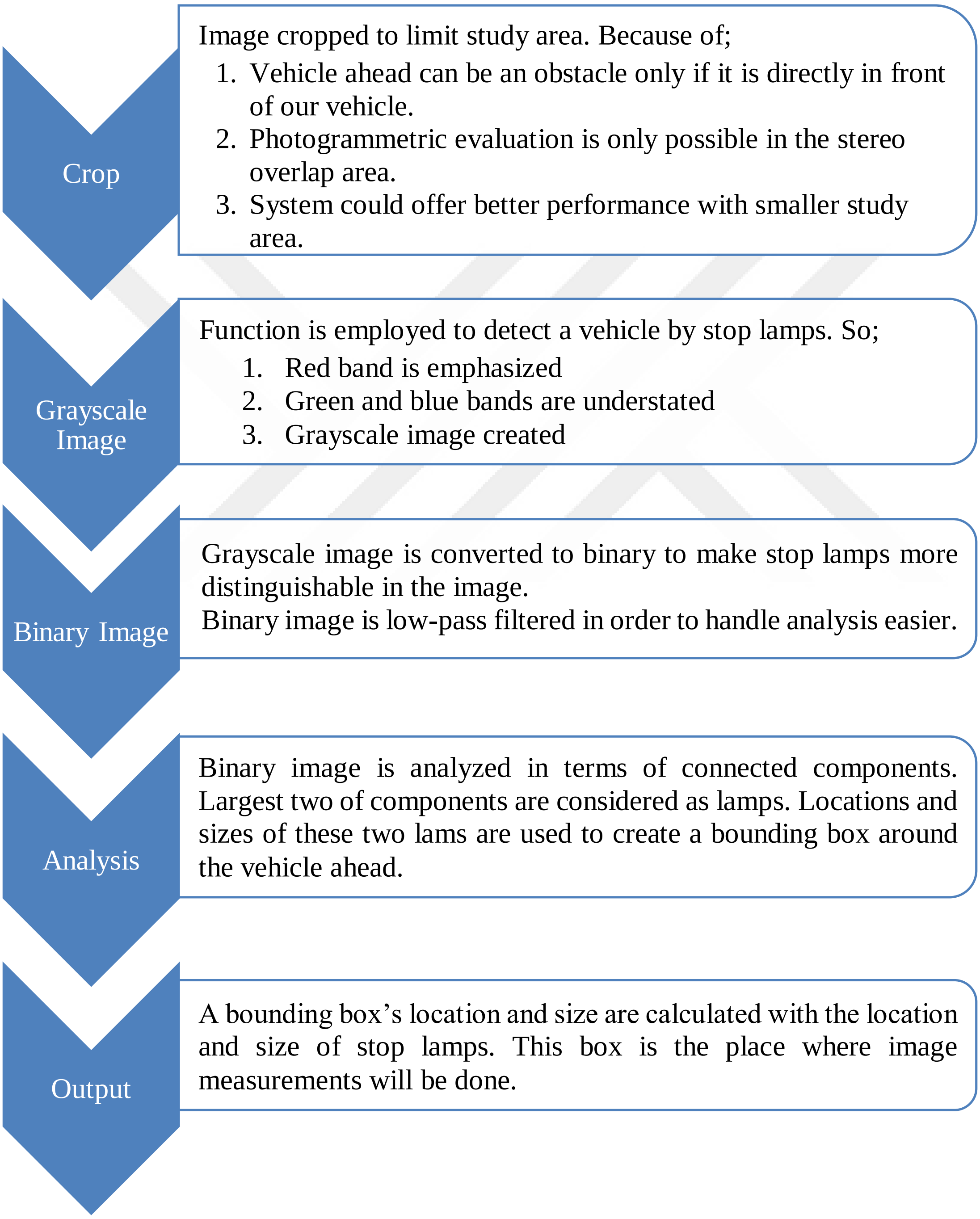
	A	B	C
1	1001.379 1019.086 997.575		
2	1001.771 1019.145 997.627		
3	1001.865 1019.155 997.600		
4	1004.506 1021.543 998.711		
5	1001.550 1025.745 999.007		
6	1003.557 1032.131 999.492		

	A	B	C
1	1001.379	1019.086	997.575
2	1001.771	1019.145	997.627
3	1001.865	1019.155	997.600
4	1004.506	1021.543	998.711
5	1001.550	1025.745	999.007
6	1003.557	1032.131	999.492

Figure 4.9 : Output of the algorithm, raw data on the left, split data in different cells is shown on the right.

4.6. Digital Image Processing Algorithm

For object detection purposes, a function is designed to find a vehicle in the image. The function is developed to find a vehicle by its stop lamps. The reason for this is stop lamps are always in red and there are always two of them for every vehicle. Steps of object detection function are shown in the Figure 4.10 below.



Crop

Image cropped to limit study area. Because of;

1. Vehicle ahead can be an obstacle only if it is directly in front of our vehicle.
2. Photogrammetric evaluation is only possible in the stereo overlap area.
3. System could offer better performance with smaller study area.

Grayscale Image

Function is employed to detect a vehicle by stop lamps. So;

1. Red band is emphasized
2. Green and blue bands are understated
3. Grayscale image created

Binary Image

Grayscale image is converted to binary to make stop lamps more distinguishable in the image.
Binary image is low-pass filtered in order to handle analysis easier.

Analysis

Binary image is analyzed in terms of connected components. Largest two of components are considered as lamps. Locations and sizes of these two lams are used to create a bounding box around the vehicle ahead.

Output

A bounding box's location and size are calculated with the location and size of stop lamps. This box is the place where image measurements will be done.

Figure 4.10 : Object detection function schematic representation.

To clarify the steps of the object detection function, each level of the function is given with screenshots as shown in the following couple of figures below. The first one is the base, image from one of the fieldworks.

The function has two inputs. One of them is the base image as shown in the Figure 4.11 below, and the other one is the position of the middle of the stereo overlap area. The middle of the stereo overlap area is shown with yellow line. Position of the line is given in image coordinate system to the function. The function seeks for the taillights of the car around the given line, within the given base image.

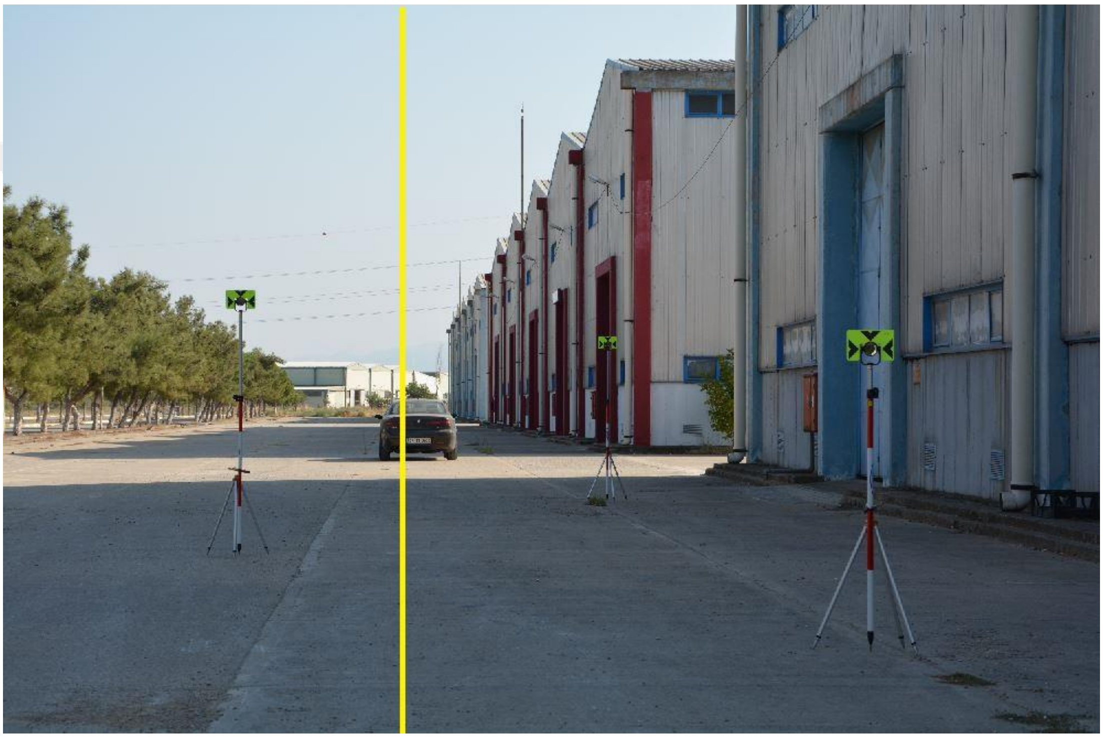


Figure 4.11 : Base image and the middle of the stereo overlap area shown with yellow line.

The function crops the image around the given line. The crop area is calculated with respect to the image size adaptively and named as scan area. The width of this area is named as scan length in the function. Latter, red band's values are exaggerated to bring taillights into the forefront. After this, a low-pass filter is applied to illuminate these taillights and output a binary image (Figure 4.12).

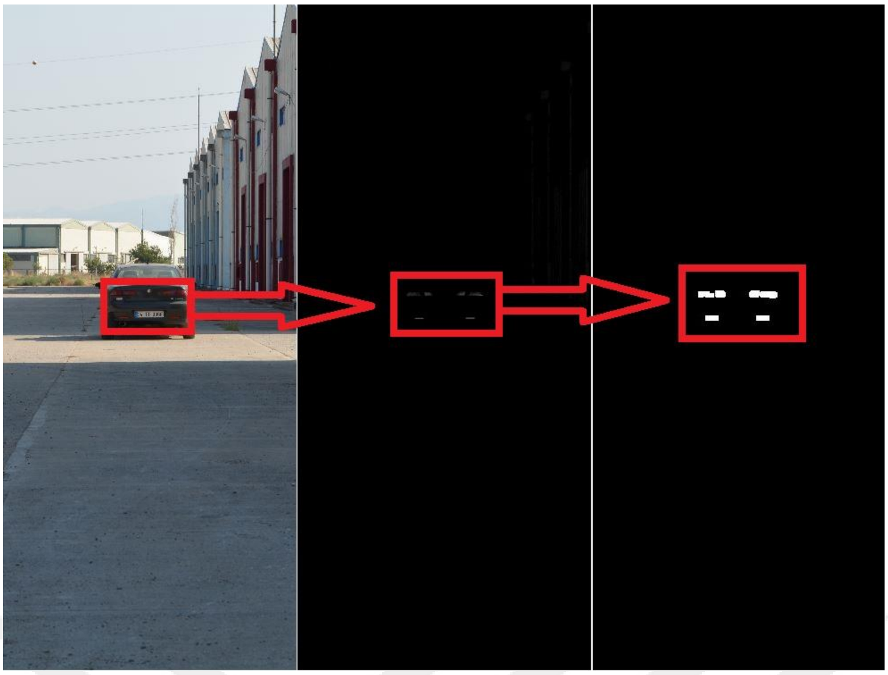
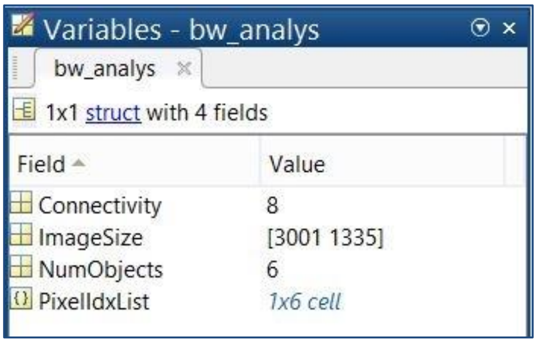


Figure 4.12 : Steps of the object detection function, in order of, crop, emphasize, and detection.

The red boxes and arrows are drawn to highlight the taillights of the car as shown in the Figure 4.12 above. The binary image is given to the built-in function, ‘bwconncomp’ to analyze the image for the connected components (Figure 4.13).



Variables - bw_analys

bw_analys

1x1 struct with 4 fields

Field	Value
Connectivity	8
ImageSize	[3001 1335]
NumObjects	6
PixelIdxList	*1x6 cell*

Figure 4.13 : Result of the connected component analyze.

There are six objects detected in the example above. The function is designed to calculate left and right limits of the object using two largest details of these objects. These limits are calculated with the index values of the detected objects with the Equation (**4.9)** below, where *mi* and *ma* are indexes of the detected components, *image* is the frame in which the function is searching for an object, *miny* is the left limit of the object, and *maxy* is the right limit of the object.

$$[\sim, miny] = ind2sub(size(image), min(mi))$$

$$[\sim, maxy] = ind2sub(size(image), \max(ma)) \qquad \textbf{(4.9)}$$

The top and the bottom limits of the object are determined with another approach. To get these top and bottom positions, the function calculates sum of the each line of the binary image top to bottom. The sum of the line is used to check if there is an object in the binary image or not. In other words, if ten percent of the pixels in the row are different from the other pixels in the same row, then this row is considered as the limit of the object. The equations used for this approach is shown below, where *x1* and_*y1* represents dimensions of the image, and *scan_length* is the value defining the crop area around the middle line **(4.10)**.

```
for i = 1:x1
    check_top = [1,i,y1 - 1,0];
    crop_top = imcrop(image,check_top);
    value_top = sum(int16(crop_top));
      if value_top >= scan_length*2*0.05
        top_box_value = i;
        break
      end
end
```

(4.10)

The function is designed to calculate the object's limits in the image as described above. However, if anything goes wrong, function outputs top and bottom values with respect to the images dimension. In this case, bottom border of the bounding box is directly accepted as the bottom row of the image, while top border is calculated as thirty percent space from the top row of the image due to the sky area in the image.

The object's top, bottom, left, and right limits does not describes the output of the function directly. These values are base for the bounding box, which is the output. Left border of the bounding box is calculated with respect to the middle line of the stereo area, the scan length, and the left limit of the detected object. Right border is calculated with the detected object's size. Top border of the bounding box is calculated with respect to the image size and top limit position of the detected object. Bottom border of the bounding box is calculated with respect to the detected object's size.

Output example of the function is visualized and can be seen in the Figure 4.14 below. Function output is a vector, containing coordinates of the upper left corner of the box, width, and height information. This box data defines where image measurements must be done for photogrammetric process.

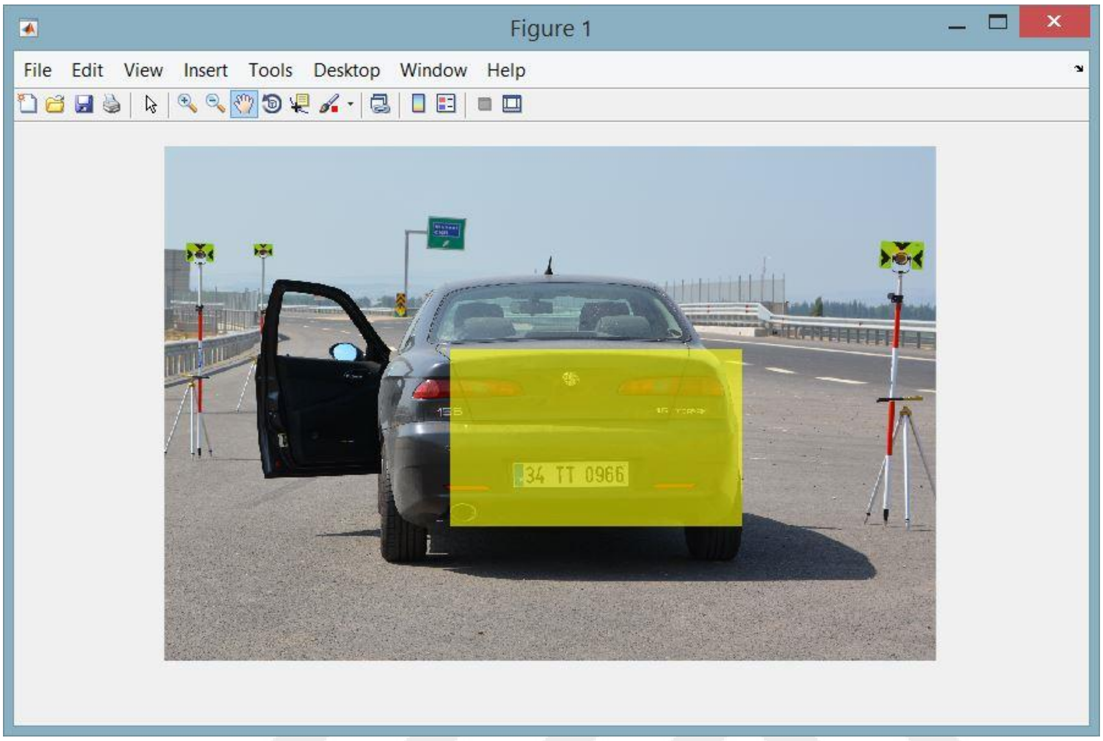


Figure 4.14 : Object detection function output.

The function is designed to work with the scan area, described by the middle of the stereo overlap area and image size, as mentioned before. However, there are many scenarios, in which the vehicle ahead does not appear in the cropped scan area. For such cases, the function simply moves the scan area to the left and right. This quantity of the motion is based on the scan length.

The Figure 4.15 below shows three positions of the scan area. The first scan area is shown with yellow rectangle, which is the initial scan area. In case of a failure with the object detection for the initial area, function moves the scan area to the left, which is shown with blue rectangle. If it fails again, the scan area is moved to the third place, which is shown with red rectangle. The function seeks for an object for each movement of the scan area. Thus, the failures due to the scan area limitations are avoided in a vast scale. The function is designed to move the scan area for four times, yet, not all movements shown in the figure since it could be difficult to understand.

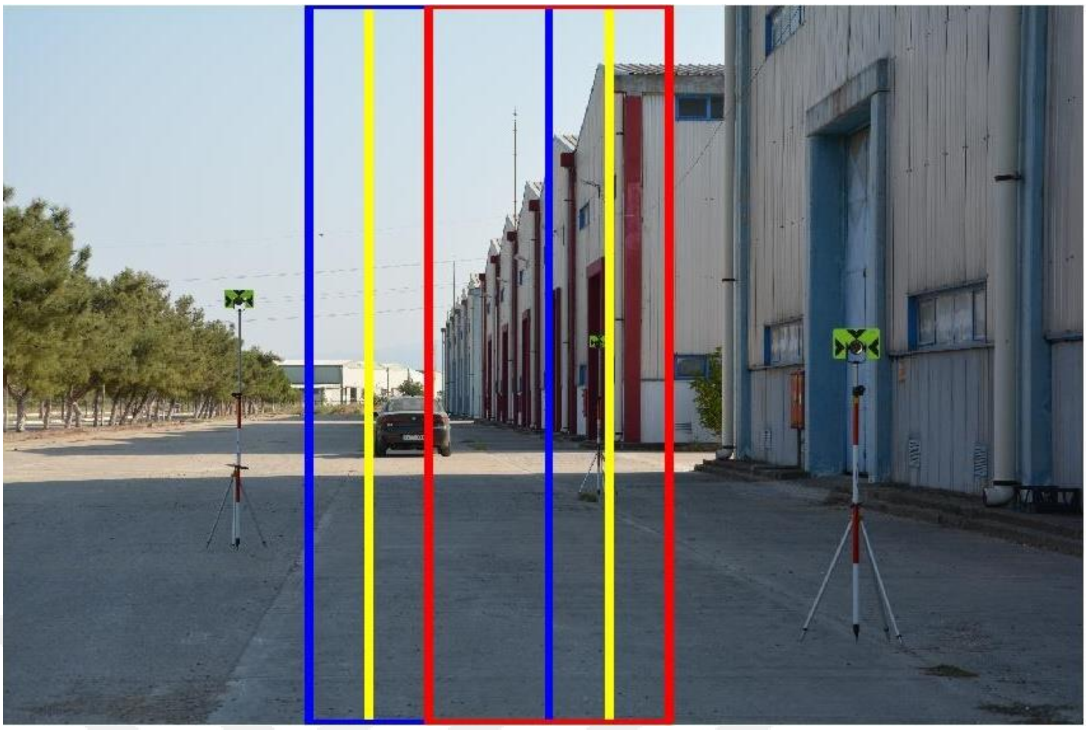
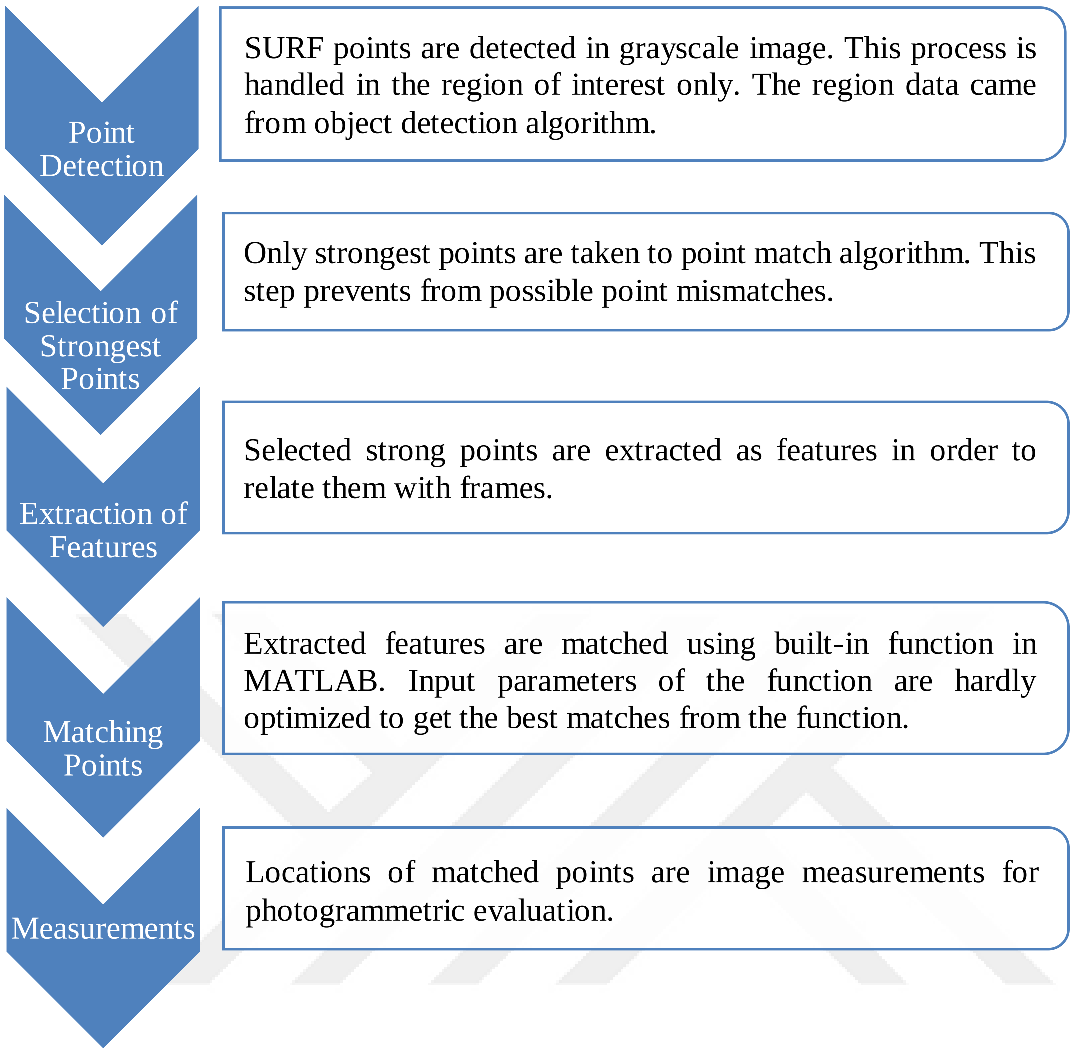


Figure 4.15 : First (yellow), second (blue) and third (red) positions of the scan area.

Image measurement procedure can be summarized in two main steps, image processing step for point detection, selection, and matching and measurement evaluation step for these points. Image processing can also be divided in three sub steps. In the first sub-step, random points are detected in the previously determined area. Second sub-step stands for selection of strongest points. Then, in the third sub-step, matching strong points from left and right images are then determined for the stereo-photogrammetric measurement.

Point detection is handled by ‘detectSURFFeatures’ (Speeded-Up Robust Features) built-in function in MATLAB. This function takes grayscale image and region of interest and outputs SURFPoints object. This object includes, count, location, scale, metric and other data as stated in the previous section. Selection from this data is handled with ‘selectStrongest’ function, which takes inputs of SURFPoints and desired number points. This selection is done in both frames with the purpose of lowering the number of mismatches. Strong points are then extracted with respect to the frames using ‘extractFeatures’ function. Locations of matched points are used in the following image measurements evaluation step (Figure 4.16).



SURF points are detected in grayscale image. This process is handled in the region of interest only. The region data came from object detection algorithm.

Only strongest points are taken to point match algorithm. This step prevents from possible point mismatches.

Selected strong points are extracted as features in order to relate them with frames.

Extracted features are matched using built-in function in MATLAB. Input parameters of the function are hardly optimized to get the best matches from the function.

Locations of matched points are image measurements for photogrammetric evaluation.

Figure 4.16 : Image measurement function diagram.

Even with these precautions, some mismatches may occur due to closeness of the points. However, this situation causes unusual coordinates, which is easy to handle. Output of this function is visualized in MATLAB environment as seen in the figure below.

In the Figure 4.17 below, black star markers are the points, which are input for photogrammetric process as image measurements.

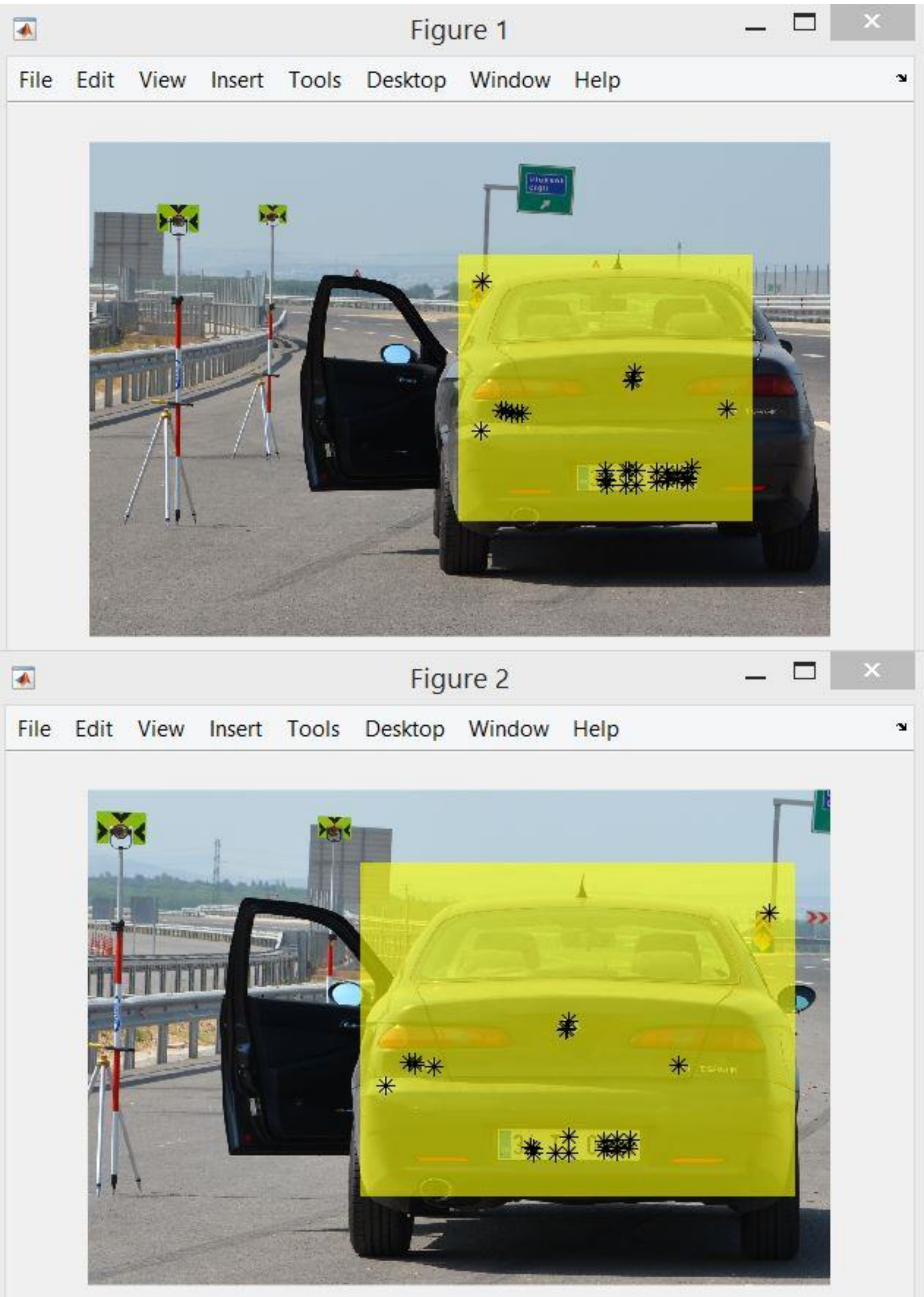


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help

Figure 4.17 : Image measurement output from the stereo pair of left (top) and right (bottom) frames.

4.7. Fieldworks

Fieldworks are designed to get position vectors of a car by stereo photogrammetry and total station measurements for a comparison. While doing this, different base distances and focal lengths are used to experience performance of setups.

The Figure 4.18 below shows data collection steps in brief. In the beginning, while the car is standing still, frames are captured and measurements are done with total station. The measurements are done using predetermined points on the car, which are corner points on the plate. In addition, three control points added to field, which are reflectors, as seen in the Figure 4.15. These control points are used to control movements of the vehicle.

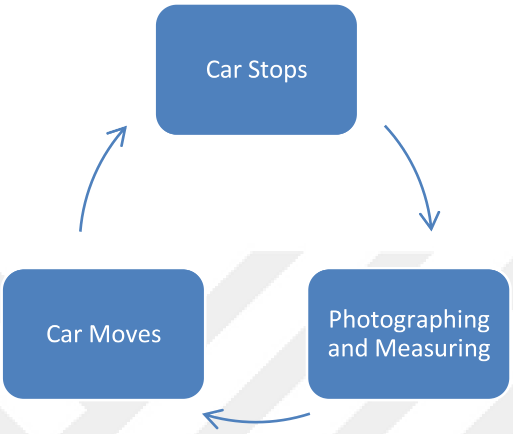


Figure 4.18 : Data collection progress in fieldworks.

Total station is placed on an unknown point and X, Y, and Z coordinates for this point are given as 100.0, 100.0, and 100.0 respectively in meters.

As seen in the Figure 4.19 below, camera setup is fixed on two tripods and car is waiting in front of the setup. At this point, predefined points' locations on the car measured by total station and photographs are taken. Then the car is moved forward and then the same procedure is repeated. At the end of the session, vehicle's movements are recorded with both photogrammetric technique and total station measurements. The main purpose of this approach is to compare the results from photogrammetric technique and total station measurements.

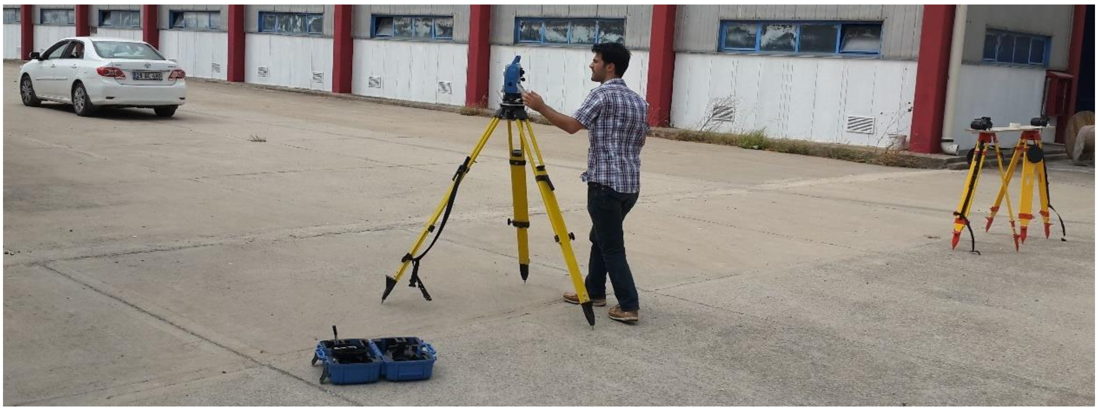


Figure 4.19 : A view from fieldwork.

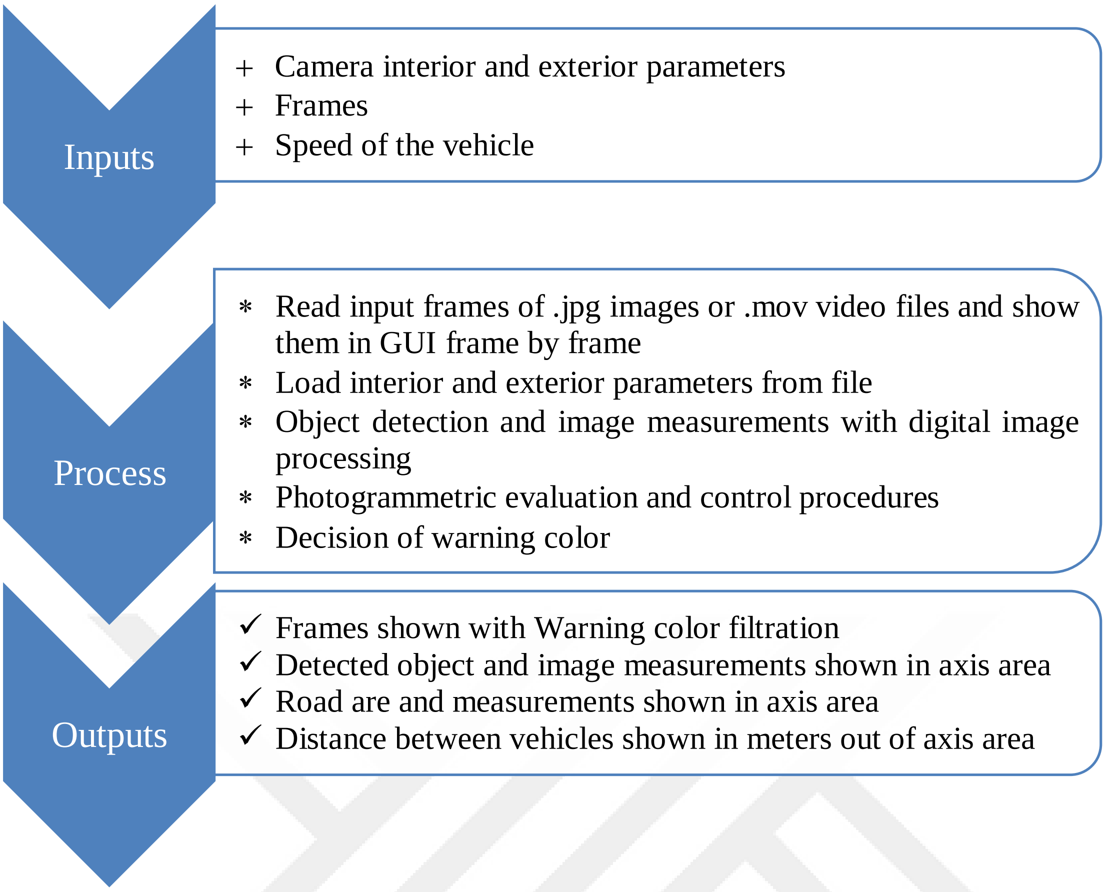
Besides from photography, video recording also done in order to evaluate frame by frame. Conversely, for video recording no total station measurements could be done.

Fieldworks are repeated for several base distances, focal lengths, addition of control points, and re-calibration of cameras. Each adjustment came with improvement for results. These results will be detailed in the following section.

4.8. Graphical User Interface Design and Software Development

A graphical user interface is needed to make the system be useful for real users. Using MATLAB’s built in ‘guide’ function, the user interface designed. The interface is designed to take all necessary inputs from user and give the visualized outputs.

The inputs of the software are very similar to photogrammetric function including, exterior and interior camera parameters, left and right frames and speed of the car. Each input is given with different buttons. This prevents from giving the wrong file to software. Speed of the camera-attached car is given by hand because system is not a real time working one. However, this speed input can be given in real time in the future work. The calculated distance between two vehicles is also shown in the graphical user interface. The software’s flowchart is given in the Figure 4.20 below.



Inputs

- Camera interior and exterior parameters
- Frames
- Speed of the vehicle

Process

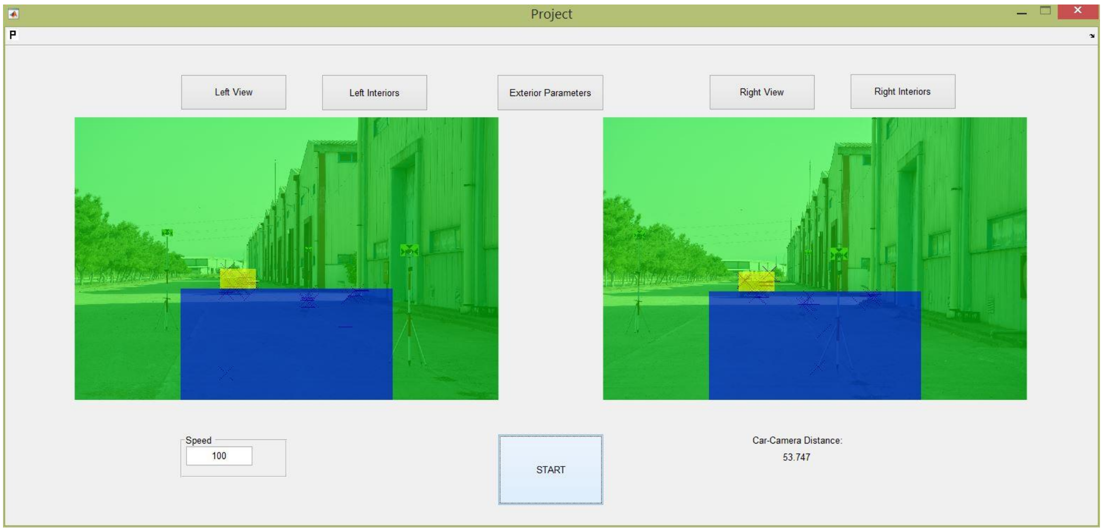
- Read input frames of .jpg images or .mov video files and show them in GUI frame by frame
- Load interior and exterior parameters from file
- Object detection and image measurements with digital image processing
- Photogrammetric evaluation and control procedures
- Decision of warning color

Outputs

- ✓ Frames shown with Warning color filtration
- ✓ Detected object and image measurements shown in axis area
- ✓ Road are and measurements shown in axis area
- ✓ Distance between vehicles shown in meters out of axis area

Figure 4.20 : GUI software diagram.

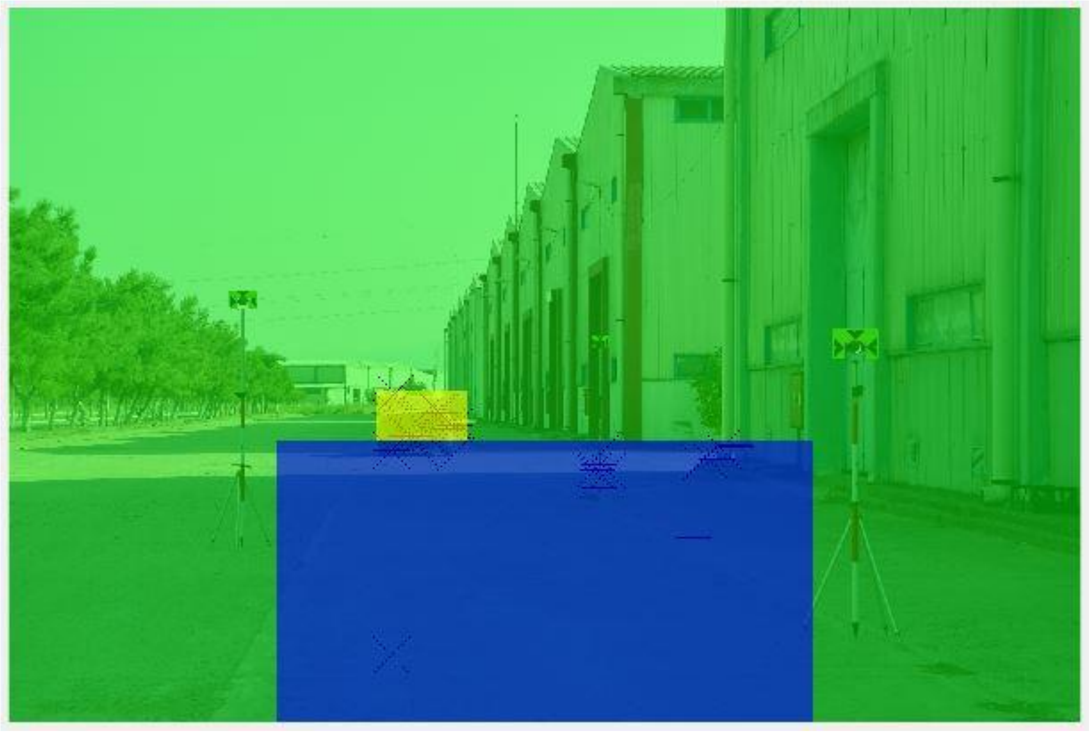
The Figure 4.21 below shows the output of the software. The button at top of it are as follows respectively, left view, left interiors, exterior parameters, right view, and right interiors. The figure in the middle are left and right frames, respectively. The figures are mainly painted as green that means the following distance is safe. The yellow boxes are the obstacles for each image and they are generated by image processing functions. The blue boxes stand for areas of road points, in which image measurements are done for road height determination. The points of image measurements are also shown as black stars below, yet, they are almost invisible in the original screen shot due to resolution. The start button triggers the calculations and then let the software reaches and displays outputs. This is developing version of the purposed software, not the final one.



100

Car-Camera Distance:
53.747

(a)

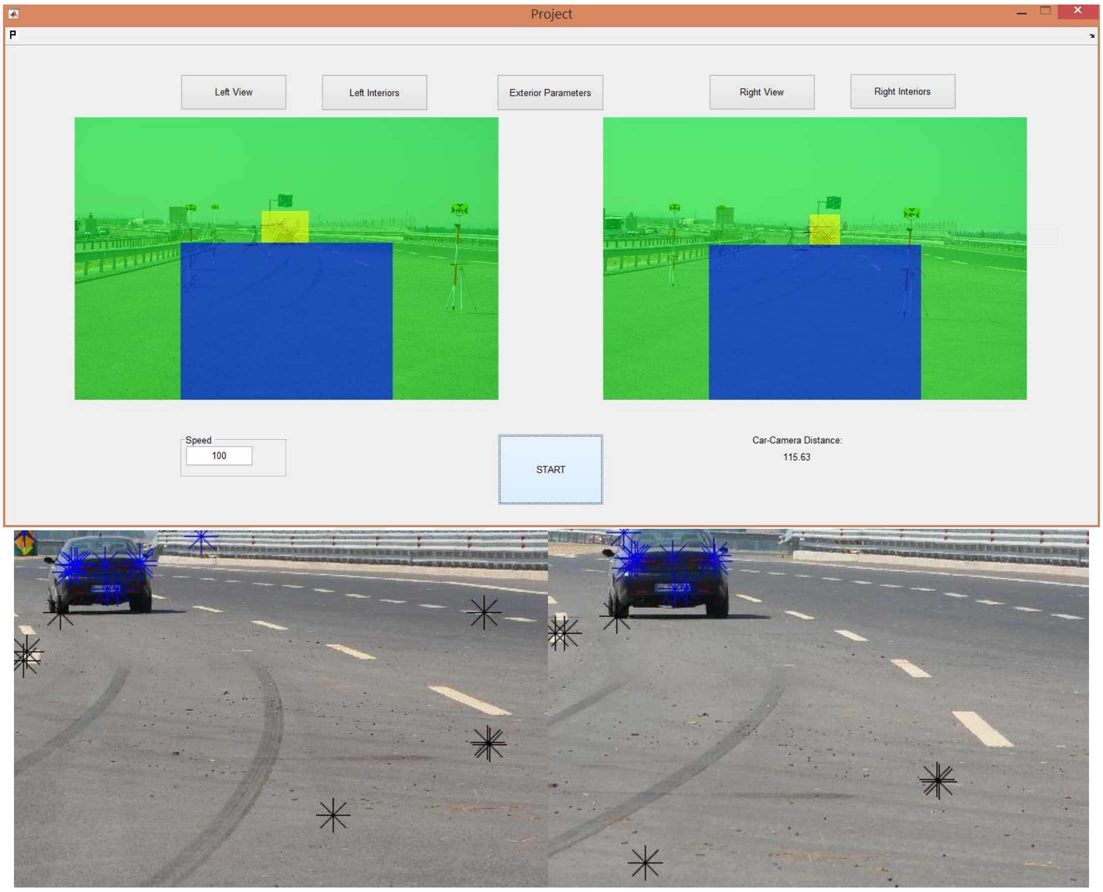


(b)

Figure 4.21 : Output of the software (a) and zoomed in left frame (b).

The approach is also tested for a real scenario from highway. Images collected for this scenario includes lane markers and it is seen that even the markers far away from the cameras are detected and matched by the developed software. The screenshot with cropped areas is shown below.

The Figure 4.22 below shows the stereo-matched images from highway, and zoomed in versions of them are shown without color filters. Black asterisks are the points that belong to road, and blue asterisks are points belonging car.



Speed 100

Car-Camera Distance: 115.63

Figure 4.22 : Detected and stereo-matched points from highway.

5. CONCLUSION

After several system design for different options in terms of focal length and base length and field trials, project mainly achieved its objective with acceptable results, a working software, and several future works. This section includes the results and conclusions of the study, and remaining works to do.

5.1. Results

As mentioned in the previous sections, several tests were done for different focal lengths and base distances. Distance between car and base between base cameras are calculated with respect to average location of predetermined points on the car plate for both total station measurements and photogrammetric calculations. First experience was the one with the worst results, as seen in the Table 5.1 below.

Table 5.1 : Results of the first study.

Car-Camera Distance Total Station Measurements (m)	**Car-Camera Distance Photogrammetry (m)**	**Difference (m)**
5.553	3.157	2.396
12.374	5.246	7.128
21.593	5.988	15.605
31.845	6.279	25.566
43.465	6.437	37.028
56.171	6.449	49.722
66.606	6.853	59.753
78.216	6.294	71.922
88.863	6.541	82.322
100.652	6.523	94.129
110.981	6.353	104.628
Base: 15cm **Focal Length:** 18mm		

As seen in the Table 5.1 above, 15cm base and 18mm focal length did not help to provide required results for the project. Contrarily, these results would disappoint.

However, when the relation between base distance and object distance from camera is considered, these results are thought as acceptable, because, trial was out of mathematical limits of the photogrammetry.

The Figure 5.1 below shows the results of the first study in graphic. The difference between distances, shown on the right side of the figure, are too high, which means this setup was useless for such study.

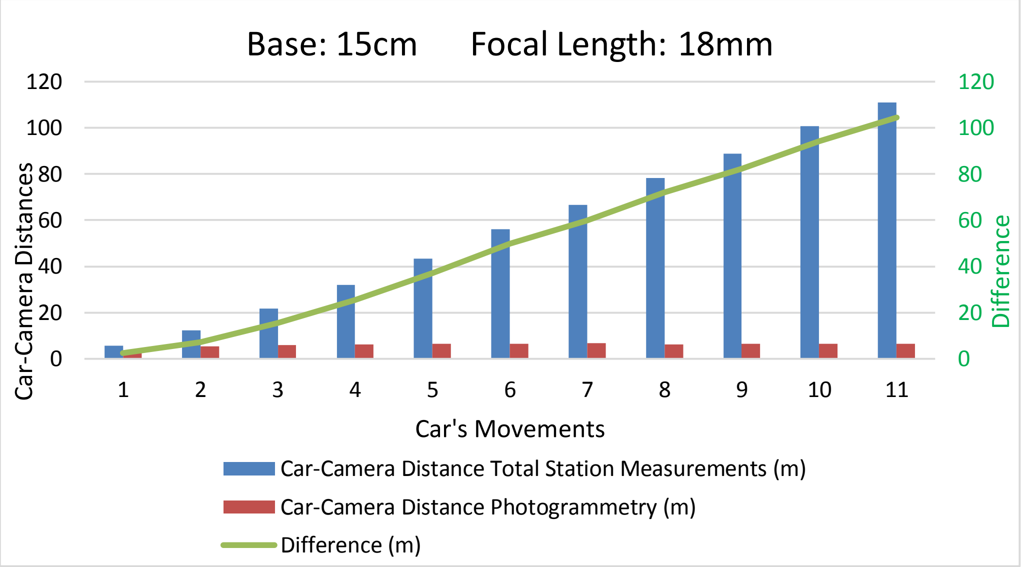


Figure 5.1 : Results of the first study.

Base to distance ratio is accepted around 1/10, in general. For a driver assistance system, it is necessary for the system to work for 50 meters or more. In this study, it is desired to sense up to 100 meters. So, base is fixed as 1.5 meters and focal length is kept at 18mm. The results for this scenario are as follows in the Table 5.2 below.

Table 5.2 : Results of the second study.

Car-Camera Distance Total Station Measurements (m)	**Car-Camera Distance Photogrammetry (m)**	**Difference (m)**
3.296	3.092	0.204
5.342	5.285	0.057
8.104	8.288	0.184
11.022	11.650	0.628
14.172	15.482	1.310
16.749	18.609	1.860
19.637	22.448	2.811
Base: 150cm **Focal Length:** 18mm		

As seen in the Table 5.2 above, increasing the base made results better. However, errors are still more than what the system requires.

The graphic in the Figure 5.2 below shows the results of the second study. The differences are again shown on the ride side and the others on the left side.

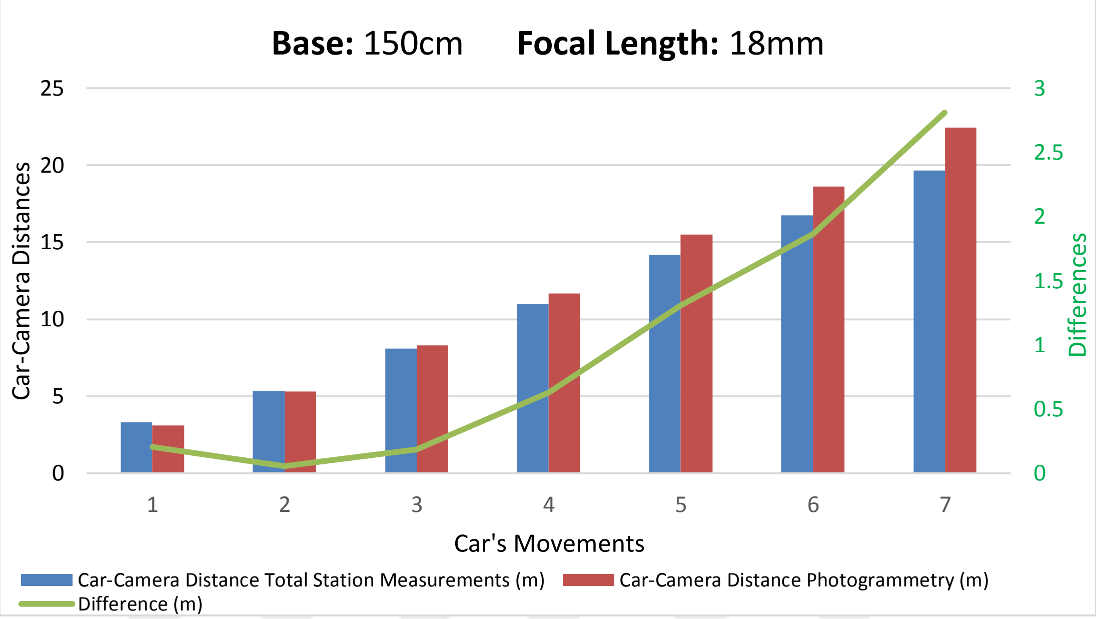


Figure 5.2 : Results of the second study.

Therefore, the base was left the same as 1.5m; focal length is changed to 35mm, 50mm, 70mm, and 105mm. The optimum focal length is observed as 50mm because of vertical and horizontal field of views, for existent camera setup. After various trials of camera calibrations, results became better and better. The last results for this study is shown in the Table 5.3 below.

Table 5.3 : Results of the last study.

Car-Camera Distance Total Station Measurements (m)	Car-Camera Distance Photogrammetry (m)	Difference (m)
11.116	11.208	0.091
18.428	18.584	0.156
24.261	24.450	0.189
30.402	30.668	0.266
38.268	38.608	0.340
45.748	46.244	0.497
52.95	53.544	0.594
60.389	61.063	0.674
64.253	64.948	0.695
69.697	70.437	0.740
75.439	76.524	1.085
81.144	82.084	0.940
91.648	92.759	1.110
Base: 150cm **Focal Length:** 50mm		

The Figure 5.3 above shows the graphic about the last study. The differences between distances are shown on the right side, while others are shown on the left side. The differences seem to be acceptable for the last study.

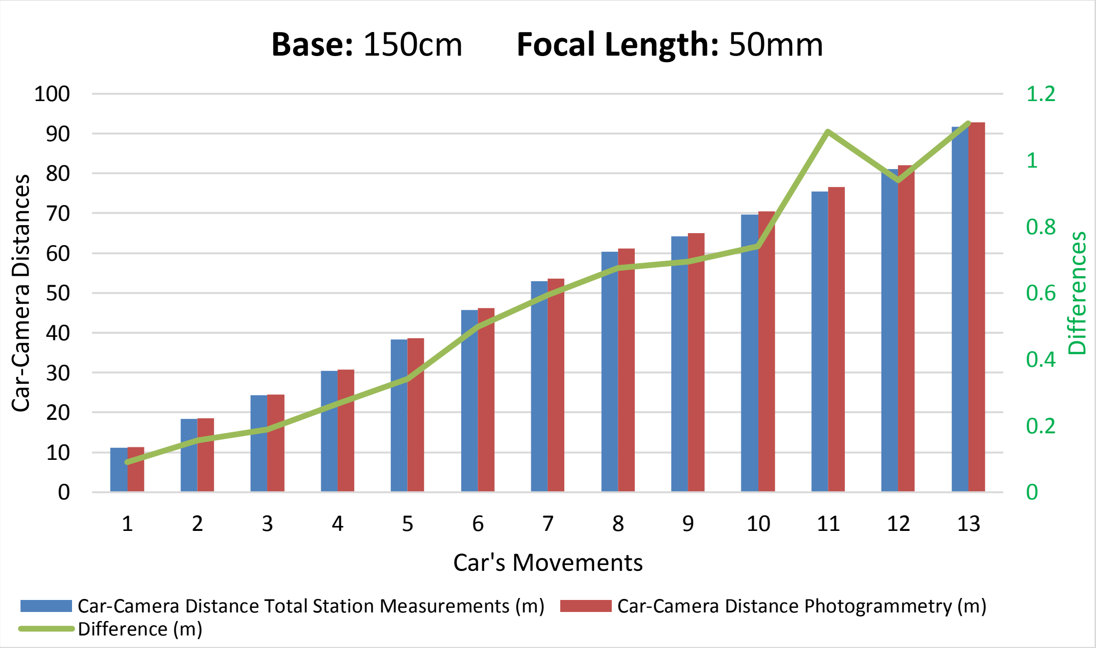


Figure 5.3 : Results of the last study.

Those distances were calculated in different coordinate systems for every move of the car. Photogrammetric computation are done in the control network's local coordinate system that is set up during stereo-camera calibration step, while total station measurements are done in another local coordinate system set up during each fieldwork. It may not be compulsory, yet, it is definitely better to make comparison in one local coordinate system. Thus, photogrammetric evaluation results are transformed to total station's local coordinate system. To do so, the widely known coordinate transformation formula is used **(5.11).**

Coordinate transformation is held by,

$$New\ Coordinates = Rotation\ Matrix * Old\ Coordinates$$

(5.11)

Rotation matrix is, shown in the Formula **(5.12)** below. (Gomarasca, 2009)

$$\begin{bmatrix} \cos ß * \cos\theta & -\cos ß * \sin\theta & \sin ß \\ \cos\alpha * \sin\theta + \sin\alpha * \sin ß * \cos\theta & \cos\alpha * \cos\theta - \sin\alpha * \sin ß * \sin\theta & -\sin\alpha * \cos ß \\ \sin\alpha * \sin\theta - \cos\alpha * \sin ß * \cos\theta & \sin\alpha * \cos\theta + \cos\alpha * \sin ß * \sin\theta & -\cos\alpha * \cos ß \end{bmatrix}$$

(5.12)

Where, α, ß, and θ are rotations around axes of, X, Y, and Z, respectively. However, rotation angles were not used in this study. Instead, each element of matrix is taken as another variable. As there are nine elements in rotation matrix, nine equations are needed to solve these parameters. During fieldwork, three control points were located with reflectors in order to generate such transformation more reliable. The coordinate transformation equation is written for these three points. Then, each element solved using these equations.

In the end, coordinate transformation formula is applied as shown in the Formula **(5.13)**.

$$\begin{bmatrix} Nx \\ Ny \\ Nz \end{bmatrix} = \begin{bmatrix} r1 & r4 & r7 \\ r2 & r5 & r8 \\ r3 & r6 & r9 \end{bmatrix} * \begin{bmatrix} Ox \\ Oy \\ Oz \end{bmatrix} \quad \textbf{(5.13)}$$

The results are compared and differences are shown in the Table 5.4 below.

Results in the Table 5.4 below are accurate enough. However, these results are still in need of enhancement due to camera calibration issues caused by both camera manufacturer and calibration software.

Table 5.4 : Results after coordinate transformation.

Car-Camera Distance Total Station Measurements (m)	**Car-Camera Distance from Transformed Coordinates (m)**	**Difference (m)**
11.116	11.382	0.266
18.428	18.720	0.292
24.261	24.555	0.294
30.402	30.748	0.346
38.268	38.650	0.382
45.748	46.257	0.510
52.950	53.517	0.567
60.389	61.000	0.611
64.253	64.869	0.616
69.697	70.333	0.636
75.439	76.387	0.948
81.144	81.920	0.775
91.648	92.544	0.895
Base: 150cm **Focal Length:** 50mm		

As seen in the Table 5.4 above, even the object's distance is more than it should be, the system still produces reasonable results. Therefore, it is possible to claim that photogrammetry is capable of supporting driver assistance systems, especially with the certain conditions of highways. The Figure 5.4 below represents the differences with the transformed coordinates.

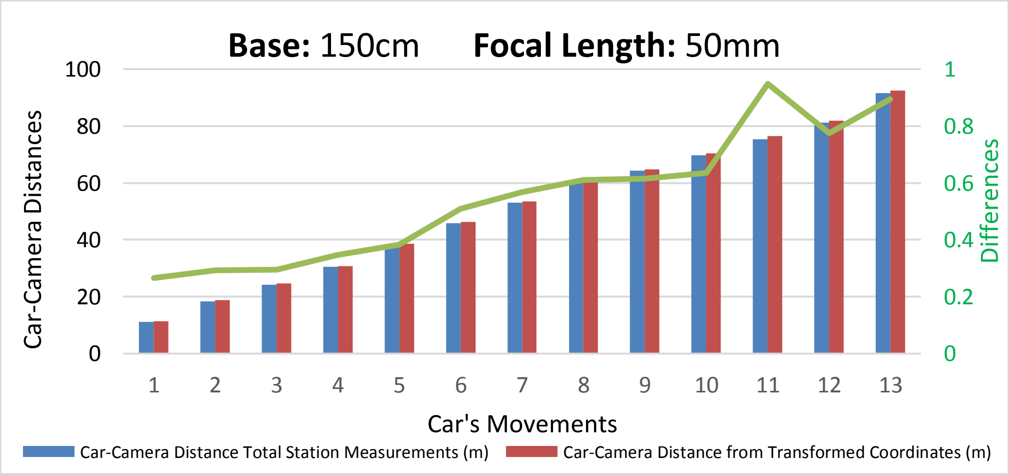


Figure 5.4 : Results after coordinate transformation.

The Figure 5.5 below shows the comparison of differences of these three studies.

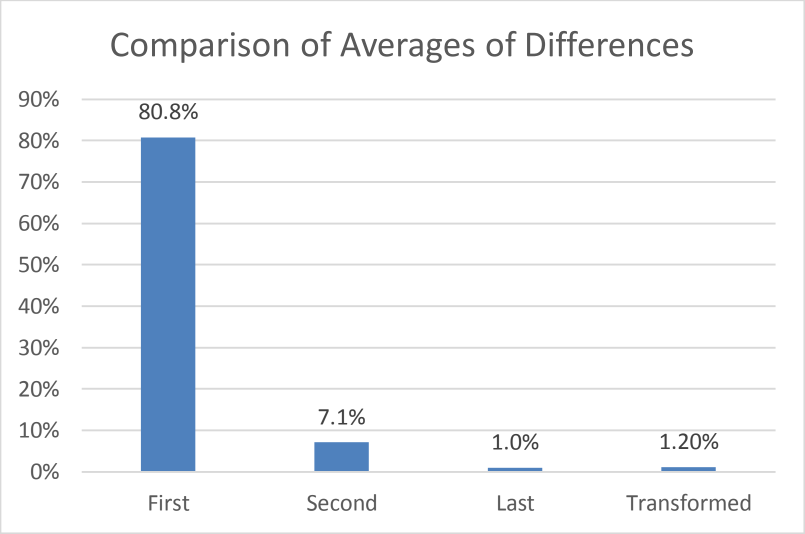


Figure 5.5 : Comparison of averages of differences.

The percentages shown in the Figure 5.5 above are calculated as in the Formula **(5.14)** below.

$$Percentage = \frac{Difference}{Distance\ Measured\ with\ Total\ Station} \qquad \textbf{(5.14)}$$

5.2. Discussion

In this thesis study, it is desired to see if photogrammetry can be employed for a driver assistance system. For this purpose, stereo-photogrammetry and digital image processing techniques and related equipment were selected with respect to the planned test scenarios.

Stereophotogrammetry is used to measure distance between the obstacle ahead and cameras. Cameras’ interior parameters are acquired using PhotoModeler software’s calibration skills. For exterior orientation parameters, a control network was designed. This helped figuring out exterior orientation parameters of the stereo cameras fixed on the chipboard.

Digital image processing is employed to detect the vehicle ahead appeared in the stereo images and to make image measurements for photogrammetric computations. For these reasons, special functions are developed and implemented into system. The object detection function is not a perfectly working one, however, it is just in the beginning stage. In addition, the object detection function is only developed to complete the system for test results. Image measurement approach in this study worked fine so far, yet, it still needs some developments to eliminate all the mismatches.

Fieldworks are done to collect data. During fieldworks, stereo-photogrammetric image couples are collected. In addition to this, total station measurements are done for control.

Each fieldwork's results are examined for every detail of every step and causes of faults, too. Different factors of every test setup are adjusted including focal length, base length, and additional control points.

Considering the results all together, it is deducted that interior parameters of the cameras have to be resolved perfectly in order to calculate fixed exterior parameters better and make stereophotogrammetric evaluation with a higher accuracy.

Faced limitations are software and hardware based problems, which are possible to be solved. Software based problems are seen as follows; wrong data in the image's database about sensor size and no editing option about this error in the calibration software.

In addition to photographs, video is also tested. Aim of video capture was to make the process frame by frame in real-time. However, the resolution of the recorded video could not enough to serve for the purpose. Therefore, studies with video-based processing are not included here. Lower resolution compared to photographs, became a limitation for point catching from a frame and such fewness of points led the system to fail. For this purpose, it is believed that high frame rate photographs are better for photogrammetric implementation in driver assistance systems. Moreover, the photographs that are used in this study are in 24 megapixels. Yet, the majority areas of the photographs are not used because these portions of the images were showing sky or areas out of road. This situation can be eliminated with better camera positioning.

Bearing in mind all the aspects, experiments, and results, it is strongly believed that all the errors faced are possible to be corrected using different softwares and/or hardware.

5.3. Future Work

It is possible to claim that this thesis study succeeded to show photogrammetry technique is practicable of supporting driver assistance systems. However, this thesis does not reveal a working prototype system.

For future work, a closed circuit driver assistance system would be designed and realized. New system should be based on; NIR cameras with high-resolution real time transition capabilities, a unit for real time processing, and an IMU unit.

NIR capable cameras can be used in both day and night conditions. Camera calibration should be handled via different software in order to obtain more useful interior camera parameters. Further, control network would be better to be established with solid, invar materials and surveyed with a more precise instrument; such network might be used to get interior parameters of cameras, too.

An IMU unit may be used to track movements of the vehicle in three-dimension, which also provides the speed of the vehicle. This three-dimension track data would be used to predict future track of the vehicle with Kalman Filter on the road, so that image processing area can be reduced to a certain margin to improve the driving safety conditions (Grewal & Andrews, 2015).

A processing unit shall be employed to handle real time processing necessities. A hardware with a graphics-processing unit (GPU) would be useful in terms of parallel processing. Parallel processing capabilities of GPUs are much better than a central processing unit (CPU), which is the demand of real-time image processing. Another alternative for such application is field programmable gate array (FPGA), yet, such hardware are harder to program.

In addition to hardware design, object detection algorithm should be enhanced in order to increase detection ratio and detect different objects, such as pedestrians, bicycles, motorcycles etc. For object detection purposes, Viola-Jones algorithm will possibly be used. For the image measurements level of the approach, the number of mismatches can be lowered with different parameters or functions.

REFERENCES

Asri, İ. (2011). *GPS Destekli Mobil Yersel Fotogrametri Sistemi.* (PhD Thesis), Selçuk Üniversitesi, https://tez.yok.gov.tr.

Asri, İ., Çorumluoğlu, Ö., & Güner, S. (2012). Hareketli Yersel Fotogrametrik Ölçme Sisteminin Hacimsel Analiz Çalışmalarında Kullanılabilirliğinin Araştırılması. *Electronic Journal of Map Technologies, 4*(3), 1-6.

Asri, İ., Çorumluoğlu, Ö., & Kalaycı, İ. (2010). *Hareketli Yersel Fotogrametrik Ölçme Sisteminin Kalibrasyonu*. Paper presented at the 5. Ulusal Mühendislik Ölçmeleri Sempozyumu, Zonguldak.

Austin, H. R. C. T. U. o. T. a. (n.d.). History: The First Photograph. Retrieved 09.10.2015, 2015, from http://www.hrc.utexas.edu/exhibitions/permanent/firstphotograph/history/#top

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer vision and image understanding, 110*(3), 346-359. doi: 10.1016/j.cviu.2007.09.014

Birch, J. (2011, 18.07.2011). Stereo Display Hardware. Retrieved 11.09.2015, 2015, from http://www.adamtech.com.au/Blog/?p=327

Bosch. (2011). EU regulation enters into force in November 2011: ESP® compulsory in all new car models [Press release]. Retrieved from http://www.bosch-presse.de/presseforum/details.htm?txtID=5412&locale=en

Bosch. (n.d.). Driver assistance systems: past, present and future. Retrieved 07.10.2015, 2015, from http://products.bosch-mobility-solutions.com/en/de/specials/specials_safety/automated_driving/technology_and_development_1/driver_assistance_systems_88/driver_assistance_systems_89.html

Daimler. (2010). The CL-Class from Mercedes-Benz [Press release]. Retrieved from http://www.daimler.com/Projects/c2c/channel/documents/1892377_CL_Klasse_e_2010.pdf

Demirel, H., & Şeker, D. Z. (2015, 21-23.05.2015). *Fotogrametrik Açık Kaynak Kodlu Yazılımlar: Yeni Bir Dönem.* Paper presented at the TUFUAB VIII. Teknik Sempozyumu, Konya, Türkiye.

Duggal, S. (2006). *Surveying* (Fourth Edition ed. Vol. Volume 2). New Delhi: McGraw Hill Education (India) Private Limited.

EOS. (n.d.). PhotoModeler - Close-Range Photogrammetry and Image Based Modeling. Retrieved 14.12.2015, 2015, from http://www.photomodeler.com/index.html

Fraser, C. S., & Brown, D. C. (1986). Industrial Photogrammetry - New Developments and Recent Applications. *Photogrammetric Record, 12*(68), 197-217.

GeoSystem. (n.d.). Analytical Photogrammetric Station «Stereoanagraph». Retrieved 11.09.2015, 2015, from http://www.vingeo.com/anagraf.html

Gomarasca, M. A. (2009). *Basics of Geomatics*.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing*.

GRAD[DE], G. H.-J. I. (1982). DE3233313 (A1). B. G. ROBERT[DE].

Graf, R. F. (Ed.) (1999) (Seventh ed.).

Grant, M. (2011). A Day in the Life of a Photogrammetry Manager. Retrieved 11.09.2015, 2015, from http://www.blom-uk.co.uk/2011/02/a-day-in-the-life-of-a-photogrammetry-manager/

Grewal, M. S., & Andrews, A. P. (2015). *Kalman Filtering Theory and Practive Using MATLAB* (Fourth Edition ed.): JohnWiley & Sons Inc.

Haggrén, H. (n.d.). Seven pictures for stereography. Retrieved 19.10.2015, 2015, from http://www.foto.hut.fi/publications/paperit/hhaggren/bildteknik_2002/stereo_photography.htm

HexGeoWiki. (2014, 22.04.2014). Interior Orientation. Retrieved 08.12.2015, 2015, from https://wiki.hexagongeospatial.com/index.php?title=Interior_Orientation

How Accurate is Photogrammetry? - Part 2. (2010). Retrieved from http://www.adamtech.com.au/Blog/?p=167

Konecny, G. (2014). *Geoinformation Remote Sensing, Photogrammetry, and Geographic Information Systems Second Edition*

Linder, W. (2006). *Digital Photogrammetry A Practical Course*.

Luhmann, T. (2010). Close range photogrammetry for industrial applications. *ISPRS Journal of Photogrammetry and Remote Sensing, 65*(6), 558-569. doi: 10.1016/j.isprsjprs.2010.06.003

MathWorks. (2015a). Detect Objects Using the Viola-Jones Algorithm - MATLAB. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-class.html

MathWorks. (2015b). Convert Image to Binary Image, Based on Threshold - MATLAB. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/images/ref/im2bw.html

MathWorks. (2015c). 2-D Adaptive Noise-Removal Filtering - MATLAB. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/images/ref/wiener2.html

MathWorks. (2015d). Noise Removal - MATLAB & Smulink. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/images/noise-removal.html#buh9ylp-72

MathWorks. (2015e). Find Connected Components in Binary Image - MATLAB. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/images/ref/bwconncomp.html

MathWorks. (2015f). Object for Storing SURF Interest Points - MATLAB. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/vision/ref/surfpoints-class.html

MathWorks. (2015g). Return Points with Strongest Metrics - MATLAB. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/vision/ref/cornerpoints.selectstrongest.html

MathWorks. (2015h). Extract interest point descriptors - MATLAB. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/vision/ref/extractfeatures.html

MathWorks. (2015i). Find Matching Features - MATLAB. Retrieved 09.12.2015, 2015, from http://www.mathworks.com/help/vision/ref/matchfeatures.html

NHTSA. (2013). *Preliminary Statement of Policy Concerning Automated Vehicles*. http://www.nhtsa.gov/: National Highway Traffic Safety Administration

Retrieved from http://www.nhtsa.gov/About+NHTSA/Press+Releases/U.S.+Department+of+Transportation+Releases+Policy+on+Automated+Vehicle+Development.

Nikon. (n.d.). Nikon | Imaging Products | Specifications - D5200. Retrieved 19.10.2015, 2015, from http://imaging.nikon.com/lineup/dslr/d5200/spec.htm

Ostermann, A., & Wanner, G. (2012). *Geometry by Its History*.

Oxford. (Ed.) (2015) Oxford Learner's Dictionary (Vols. 2015). http://www.oxforddictionaries.com/: Oxford University Press.

Şeker, D. Z., & Duran, Z. (n.d.). Terrestrial & Numerical Photogrammetry. Retrieved 10.09.2015, 2015, from http://web.itu.edu.tr/~seker/files/Terrestrial%20and%20Numerical%20Phot.-1.PDF

Tesla. (2015). Your Autopilot has arrived. Retrieved from https://www.teslamotors.com/blog/your-autopilot-has-arrived

Trimble. (2014). FOCUS 8 Series Total Stations Datasheet. Retrieved 15.10.2015, 2015, from http://www.spectraprecision.com/media/custom/upload/File-1418898434.pdf

TSI. (2015a). *Number of road motor vehicles by aim of use*. http://www.tuik.gov.tr/: Turkish Statistical Institute Retrieved from http://tuikapp.tuik.gov.tr/medas/?kn=89&locale=tr.

TSI. (2015b). *Number of road motor vehicles*. http://www.tuik.gov.tr/Start.do: Turkish Statistical Institute Retrieved from http://www.tuik.gov.tr/Start.do.

TSI. (2015c). *Road lengths*. http://www.tuik.gov.tr/Start.do: Turkish Statistical Institute Retrieved from http://www.tuik.gov.tr/Start.do.

TSI. (2015d). *Number of traffic accidents and results*. http://www.tuik.gov.tr/Start.do: Turkish Statistical Institute Retrieved from http://www.tuik.gov.tr/Start.do.

TSI. (2015e). *Number of accidents, persons killed and injured by year*. http://www.tuik.gov.tr/Start.do: Turkish Statistical Institute Retrieved from http://www.tuik.gov.tr/Start.do.

TSI. (2015f). *Seçilmiş 50 neden, cinsiyet ve yaş grubuna göre ölümler*. http://www.tuik.gov.tr/Start.do: Turkish Statistical Institute Retrieved from http://www.tuik.gov.tr/Start.do.

TSI. (2015g). *Faults causing road traffic accidents*. http://www.tuik.gov.tr/Start.do: Turkish Statistical Institute Retrieved from http://www.tuik.gov.tr/Start.do.

Urmson, C. (2012). The self-driving car logs more miles on new wheels. Retrieved from https://googleblog.blogspot.co.uk/2012/08/the-self-driving-car-logs-more-miles-on.html

Viola, P., & Jones, M. (2001). *Rapid Object Detection Using a Boosted Cascade of Simple Features*. Paper presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

APPENDICES

APPENDIX A - Graphical User Interface Code

```
%This is the code of the graphical user interface. The GUI is coded
%to take inputs of camera parameters, speed, and frames then, output
&the  frames  with  the  corresponding  warning  type.  The  GUI  calls
%functions, which are specifically designed for this project.

function varargout = Project( varargin )
gui_Singleton = 1;
gui_State = struct( 'gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Project_OpeningFcn, ...
    'gui_OutputFcn', @Project_OutputFcn, ...
    'gui_LayoutFcn', [ ] , ...
    'gui_Callback', [ ] );
if nargin && ischar( varargin{ 1 }o )
    gui_State.gui_Callback = str2func( varargin{ 1 } );
end

if nargout
    [ varargout{ 1:nargout } ] = gui_mainfcn( gui_State, varargin{ :
} );
else
    gui_mainfcn( gui_State, varargin{ : } );
end
function Project_OpeningFcn( hObject, eventdata, handles, varargin )
handles.output = hObject;

guidata( hObject, handles );

function varargout = Project_OutputFcn( hObject, eventdata, handles )
varargout{ 1 } = handles.output;



function pushbutton2_Callback( hObject, eventdata, handles )
 [ filename, pathname ] = uigetfile( { '*.jpg;*.jpeg;*.mov;*.mp4' },
'File Selector' );
global extensionL;
extensionL = strsplit( filename, '.' );
global img;
global videoReaderL;
global videoFrameL;
global imge;
imge = strcat( pathname, filename );
if strcmp( extensionL{ 2 }, 'JPG' ) || strcmp( extensionL{ 2 }, 'JPEG'
)
    img = imread( imge );
    axes( handles.axes2 );
    imshow( img );
elseif strcmp( extensionL{ 2 }, 'MOV' ) || strcmp( extensionL{ 2 },
'mp4' )
    videoReaderL = vision.VideoFileReader( imge );
    videoFrameL = step( videoReaderL );
    axes( handles.axes2 );
    imshow( videoFrameL );
end



function pushbutton3_Callback( hObject, eventdata, handles )
 [ filename, pathname ] = uigetfile( { '*.jpg;*.jpeg;*.mov;*.mp4' },
'File Selector' );
global extensionR;
extensionR = strsplit( filename, '.' );
global img1;
global videoReaderR; global videoFrameR;
global image1;
image1 = strcat( pathname, filename );
if strcmp( extensionR{ 2 }, 'JPG' ) ||strcmp( extensionR{ 2 }, 'JPEG'
)
    img1 = imread( image1 );
    axes( handles.axes4 );
    imshow( img1 );
elseif strcmp( extensionR{ 2 }, 'MOV' ) || strcmp( extensionR{ 2 },
'mp4' )
    videoReaderR = vision.VideoFileReader( image1 );
    videoFrameR = step( videoReaderR );
    axes( handles.axes4 );
    imshow( videoFrameR );
end



function pushbutton4_Callback( hObject, eventdata, handles )
global fname1;
[ fileName1, pathname1 ] = uigetfile( { '*.mat' }, 'File Selector' );
fname1 = strcat( pathname1, fileName1 );



function pushbutton5_Callback( hObject, eventdata, handles )
global fname2 ;
[ fileName2, pathname2 ] = uigetfile( { '*.mat' }, 'File Selector' );
fname2 = strcat( pathname2, fileName2 );


function pushbutton6_Callback( hObject, eventdata, handles )
global fname3;
[ fileName3, pathname3 ] = uigetfile( { '*.mat' }, 'File Selector' );
fname3 = strcat( pathname3, fileName3 );


function edit2_Callback( hObject, eventdata, handles )


function edit2_CreateFcn( hObject, eventdata, handles )

if  ispc  &&  isequal(  get(  hObject,  'BackgroundColor'  ),  get(  0,
'defaultUicontrolBackgroundColor' ) )
    set( hObject, 'BackgroundColor', 'white' );
end


function pushbutton8_Callback( hObject, eventdata, handles )
global img; global img1; global noPoints; global fname1 ;
global fname2; global fname3; global Sped;global extensionR;
global videoReaderL;global videoFrameL;global videoReaderR;
global videoFrameR;global imge;global image1;global checkValue;
Sped = str2double( get( handles.edit2, 'String' ) );

sLeft = load( fname1 );
sRight = load( fname2 );
sExt = load( fname3 );

leftInterior = structCaller( sLeft );
rightInterior = structCaller( sRight );
ext = structCaller( sExt );

if strcmp( extensionR{ 2 }, 'JPG' ) || strcmp( extensionR{ 2 }, 'JPEG'
)

    bboxl = vehicleDetector2( img, 2973 );
    bboxr = vehicleDetector2( img1, 2600 );
    [ r1, r2, carPoints ] = photog6( img, img1, sLeft.( leftInterior
), sRight.( rightInterior ), sExt.( ext ), bboxl, bboxr );

    if ( bboxl( 2 ) + bboxl( 4 ) ) > size( img, 1 )*0.95 || ( bboxr(
2 ) + bboxr( 4 ) ) > size( img1, 1 )*0.95
        r1r = [ 0, 0 ]; r2r = [ 0, 0 ];
        roadPoints = 0;
        roadBoxLeft = [ 0, 0, 0, 0 ];
        roadBoxRight= [ 0, 0, 0, 0 ];
    else
        [ roadBoxLeft, roadBoxRight ] = roadBoxer( bboxl, size( img
), bboxr, size( img1 ) );
        [  r1r,  r2r,  roadPoints  ]  =  photog6(  img,  img1,  sLeft.(
leftInterior ), sRight.( rightInterior ), sExt.( ext ), roadBoxLeft,
roadBoxRight );
    end

    onePoint = carPointSelector( roadPoints, carPoints );

    distCar = pointDistance( onePoint, cameraMidPoint( sExt.( ext ) )
);
    warnColor = warner( distCar, Sped );

    axes( handles.axes2 );
    imag = insertShape( img, 'FilledRectangle', [ 0, 0, 6000, 4000 ],
'Color' , warnColor , 'Opacity', 0.5 );
    imag = insertMarker( imag, [ r1;r1r ], 'marker', 'star' );
    imag   =   insertShape(   imag,   'FilledRectangle',   roadBoxLeft,
'Color', 'blue' );
    imag = insertShape( imag, 'FilledRectangle', bboxl );
    if strcmp( warnColor, 'black' ) ==1
        imag = insertText( imag, [ 10 size( imag, 1 )/2 ], 'Unknown
Following Distance Error' );
    end
    imshow( imag );

    axes( handles.axes4 );
    imbg = insertShape( img1, 'FilledRectangle', [ 0, 0, 6000, 4000
], 'Color', warnColor , 'Opacity', 0.5 );
    imbg = insertMarker( imbg, [ r2;r2r ], 'marker', 'star' );
    imbg   =   insertShape(   imbg,   'FilledRectangle',   roadBoxRight,
'Color', 'blue' );
    if strcmp( warnColor, 'black' ) == 1
```

```
        imbg = insertText( imbg, [ 10 size( imbg, 1 )/2 ], 'Unknown
Following Distance Error' );
    end
    imbg = insertShape( imbg, 'FilledRectangle', bboxr );

    checkValue = num2str( distCar, 5 );
    set( handles.text2, 'String', checkValue );
    imshow( imbg );

elseif strcmp( extensionR{ 2 }, 'MOV' ) || strcmp( extensionR{ 2 },
'mp4' )

    while ~isDone( videoReaderR ) && ~isDone( videoReaderL )
        bboxl = vehicleDetector2( videoFrameL, 700 );
        bboxr = vehicleDetector2( videoFrameR, 500 );

        [ r1, r2, carPoints ] = photog6Video( videoFrameL,
videoFrameR, sLeft.( leftInterior ), sRight.( rightInterior ), sExt.(
ext ), bboxl, bboxr );

        if ( bboxl( 2 ) + bboxl( 4 ) ) > size( videoFrameL, 1 )*0.95
|| ( bboxr( 2 ) + bboxr( 4 ) ) > size( videoFrameR, 1 )*0.95
            r1r = [ 0, 0 ]; r2r = [ 0, 0 ];
            roadPoints = 0;
            roadBoxLeft = [ 0, 0, 0, 0 ];
            roadBoxRight= [ 0, 0, 0, 0 ];
        else
            [ roadBoxLeft, roadBoxRight ] = roadBoxer( bboxl, size(
videoFrameL ), bboxr, size( videoFrameR ) );
            [ r1r, r2r, roadPoints ] = photog6( videoFrameL,
videoFrameR, sLeft.( leftInterior ), sRight.( rightInterior ), sExt.(
ext ), roadBoxLeft, roadBoxRight );
        end
        if exist( 'roadPoints', 'var' )==0
            roadPoints = 0;
        end
        onePoint = carPointSelector( roadPoints, carPoints );
        distCar = pointDistance( onePoint, cameraMidPoint( sExt.( ext
) ) );
        warnColor = warner( distCar, Sped );

        videoFrameL = step( videoReaderL );
        imag = insertShape( videoFrameL, 'FilledRectangle', [ 0, 0,
1920, 1080 ], 'Color' , warnColor , 'Opacity', 0.5 );
        imag = insertMarker( imag, [ r1;r1r ], 'marker', 'star' );
        imag = insertShape( imag, 'FilledRectangle', roadBoxLeft,
'Color', 'blue' );
        if strcmp( warnColor, 'black' ) ==1
            imag = insertText( imag, [ 10 size( imag, 1 )/2 ], 'Unknown
Following Distance Error' );
        end
        imag = insertShape( imag, 'FilledRectangle', bboxl );

        visualFrame( handles.axes2, imag );


        videoFrameR = step( videoReaderR );
        imbg = insertShape( videoFrameR, 'FilledRectangle', [ 0, 0,
1920, 1080 ], 'Color' , warnColor , 'Opacity', 0.5 );
        imbg = insertMarker( imbg, [ r2;r2r ], 'marker', 'star' );
        imbg = insertShape( imbg, 'FilledRectangle', roadBoxRight,
'Color', 'blue' );
        if strcmp( warnColor, 'black' ) ==1
            imbg = insertText( imbg, [ 10 size( imbg, 1 )/2 ], 'Unknown
Following Distance Error' );
        end
        imbg = insertShape( imbg, 'FilledRectangle', bboxr );
        visualFrame( handles.axes4, imbg );

        checkValue = num2str( distCar, 5 );
        set( handles.text2, 'String', checkValue );
    end
end


function uipanel1_SizeChangedFcn( hObject, eventdata, handles )
```

APPENDIX B – A Part of Photogrammetric Evaluation Code

```
%This is a part of the photogrammetric evaluation function. The
%function specifically desined to take inputs of frames, camera
%parameters, and region of interest for each frame to make image
%measurements and output 3D coordinates for these points and their
%image measurements.

function  [r1, r2, points]=photog6 ( colourImage1, colourImage2, sol,
sag, ext, bbox1, bbox2 )


ro=pi/180;
grayimage1 = rgb2gray ( colourImage1 );
grayimage2 = rgb2gray ( colourImage2 );

par = ext;
c1 = sol.focal;
c2 = sag.focal;
pix_val1 = 3.91;
pix_val2 = 3.91;
xo1 = [sol.pp ( 1, 1 ); sol.pp ( 1, 2 )];
xo2 = [sag.pp ( 1, 1 ); sag.pp ( 1, 2 )];

[res_size1 ( 1 ), res_size1 ( 2 ), ~]= size ( colourImage1 );
[res_size2 ( 1 ), res_size2 ( 2 ), ~]= size ( colourImage2 );

k11 = sol.dist_k ( 1, 1 ); k12 = sol.dist_k ( 1, 2 ); k13 = sol.dist_k
( 1, 3 );
p11 = sol.dist_p ( 1, 1 ); p12 = sol.dist_p ( 1, 2 );
k21 = sag.dist_k ( 1, 1 ); k22 = sag.dist_k ( 1, 2 ); k23 = sag.dist_k
( 1, 3 );
p21 = sag.dist_p ( 1, 1 ); p22 = sag.dist_p ( 1, 2 );

points1 = detectSURFFeatures ( grayimage1, 'ROI', bbox1 );
points2 = detectSURFFeatures ( grayimage2, 'ROI', bbox2 );

strong_p1 = points1.selectStrongest ( bbox1 ( 3 ) * bbox1 ( 4 ) * 1/50
);
strong_p2 = points2.selectStrongest ( bbox2 ( 3 ) * bbox2 ( 4 ) * 1/50
);
```

```
 [feat1, valid_points1] = extractFeatures ( grayimage1, strong_p1 );
[feat2, valid_points2] = extractFeatures ( grayimage2, strong_p2 );

index_pairs = matchFeatures ( feat1, feat2, 'Method', 
'NearestNeighborRatio', 'MaxRatio', 0.6, ...
    'Metric', 'normxcorr', 'MatchThreshold', 35 );

matched_points1 = valid_points1 ( index_pairs ( :, 1 ), : );
matched_points2 = valid_points2 ( index_pairs ( :, 2 ), : );

r1 = double ( matched_points1.Location );
r2 = double ( matched_points2.Location );
…
```

APPENDIX C – A Part of Object Detection Code

```
%This is a part of the object detection code. The function calls the
%other custom developed function to detect object. In case of failure,
%it calls the function with a different region of interest information

function vehicleBox = vehicleDetector2( image, middle_line_x )
[ x1 , y1 , ~ ] = size( image );
scan_length = uint16( y1 / 9 );
try
  vehicleBox = rgbFinder4( image, middle_line_x, x1, y1, scan_length
);
catch
  try
    vehicleBox = rgbFinder4( image, uint16( middle_line_x - 
scan_length / 2, x1, y1, scan_length ) );
  catch
    try
      vehicleBox = rgbFinder4( image, uint16( middle_line_x + 
scan_length / 2, x1, y1, scan_length ) );

    catch
      try
        vehicleBox = rgbFinder4( image, uint16( middle_line_x - 
scan_length * 3 / 4, x1, y1, scan_length ) );
      catch
        try
          vehicleBox = rgbFinder4( image, uint16( middle_line_x + 
scan_length * 3 / 4, x1, y1, scan_length ) );
        catch
          vehicleBox = rgbFinder4( image, uint16( middle_line_x + 
scan_length, x1, y1, scan_length ) );
        end
      end
    end
  end
end
```

APPENDIX C - Coordinate Transformation Code

```
%This is coordinate transformation function. This function calculates
%coordinate transformation parameters (rotation matrix) using three
```

```
%points' coordinates for two different coordinate systems. And outputs
%the new coordinates of another (fourth point) in the new coordinate
%system

function [ transformedCoords, R ] = coordTrans (  ko1, ko2, ko3, kn1,
kn2, kn3, oldCoords )

syms r1 r2 r3 r4 r5 r6 r7 r8 r9

eq1 = kn1 ( 1 ) == r1  *  ko1 ( 1 ) + r4  *  ko1 ( 2 ) + r7  *  ko1 (
3 );
eq2 = kn1 ( 2 ) == r2  *  ko1 ( 1 ) + r5  *  ko1 ( 2 ) + r8  *  ko1 (
3 );
eq3 = kn1 ( 3 ) == r3  *  ko1 ( 1 ) + r6  *  ko1 ( 2 ) + r9  *  ko1 (
3 );

eq4 = kn2 ( 1 ) == r1  *  ko2 ( 1 ) + r4  *  ko2 ( 2 ) + r7  *  ko2 (
3 );
eq5 = kn2 ( 2 ) == r2  *  ko2 ( 1 ) + r5  *  ko2 ( 2 ) + r8  *  ko2 (
3 );
eq6 = kn2 ( 3 ) == r3  *  ko2 ( 1 ) + r6  *  ko2 ( 2 ) + r9  *  ko2 (
3 );

eq7 = kn3 ( 1 ) == r1  *  ko3 ( 1 ) + r4  *  ko3 ( 2 ) + r7  *  ko3 (
3 );
eq8 = kn3 ( 2 ) == r2  *  ko3 ( 1 ) + r5  *  ko3 ( 2 ) + r8  *  ko3 (
3 );
eq9 = kn3 ( 3 ) == r3  *  ko3 ( 1 ) + r6  *  ko3 ( 2 ) + r9  *  ko3 (
3 );

solveIt = solve ( eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8, eq9, [r1,
r2, r3, r4, r5, r6, r7, r8, r9] );

R = [solveIt.r1, solveIt.r4, solveIt.r7;...
    solveIt.r2, solveIt.r5, solveIt.r8;...
    solveIt.r3, solveIt.r6, solveIt.r9];
R = double (  R  );
transformedCoords = zeros (  size (  oldCoords  ) );
for i = 1:size (  oldCoords, 1  )
transformedCoords (  i,  :  ) = R  *  oldCoords (  i,:  )';
end
```

APPENDIX D - Point Selection from Car Point List Code

```
%This function eliminates the points from car point list, which are
%not suitable for a car height

function carAvgPoint = carPointSelector ( roadPoints, carPoints  )
carPoints =  ( sortrows ( carPoints, 3 ) );
if roadPoints ~= 0
    roadMean = meanSelectedPoints ( roadPoints );
    for i=1:size ( carPoints,1 )
        if carPoints ( i, 3 ) >  ( roadMean ( 3 ) + 0.25 ) &&  carPoints
( i, 3 ) <  (  roadMean ( 3 ) + 1.75  )
            carContainer ( i, : ) = carPoints ( i, : );
        else
            carContainer = carPoints;
        end
    end
    carAvgPoint = mean ( carContainer );
elseif roadPoints == 0
    carAvgPoint = mean ( carPoints );
end
```

APPENDIX E - Warning Color Decision Making Code

```
%This function outputs warning type according to the distance between
%and velocity

function color = warner ( distance, velocity )

safeFolDist = 8 + 0.3 * velocity;
ratio = distance / safeFolDist;

if ratio >= 1.07
    color = 'g';
elseif ratio <= 0.93
    color = 'r';
elseif ratio > 0.93 && ratio < 1.07
    color = 'y';
else
    color = 'black';
end
```

APPENDIX F - Road Area in Image Calculation Code

```
%This  function  outputs  a  ROI  of  road  area  in  each  frame.  This
%calculation is based on detected object locations in the image.

function  [ boxRoadLeft,   boxRoadRight  ]  =  roadBoxer(  carBoxLeft,
frameSizeLeft,  carBoxRight,  frameSizeRight )

boxRoadLeft  = [ frameSizeLeft( 2 )*0.25,  carBoxLeft( 2 )+carBoxLeft(
4 ),  frameSizeLeft( 2 )*0.5,  frameSizeLeft( 1 ) - ( carBoxLeft( 2
)+carBoxLeft( 4 ) ) ];

boxRoadRight   =   [   frameSizeRight(   2   )*0.25,     carBoxRight(   2
)+carBoxRight( 4 ),  frameSizeRight( 2 )*0.5,  frameSizeRight( 1 ) -
( carBoxRight( 2 )+carBoxRight( 4 ) ) ];
```

APPENDIX G - Selection from Points with Mean Height Code

```
%This function calculates the mean of the coordinates. It first sorts
%the coordinates with respect to the Z axis. And calculates mean of
%the coordinates based on the points in the middle.

function onePoint = meanSelectedPoints( pointsMa )

[x, ~, ~] = size( pointsMa );
pointsMa = sortrows( pointsMa, 3 );
onePoint = mean(  pointsMa( int16( x/2 ) - 2 : int16( x/2 )+2, :  )
);
```

APPENDIX H - Drawing Axis with Edited Frame Code

```
%This function is designed to place frames to the axis graphics object
%in the GUI.

function visualFrame( axisName,frame )

frame = im2uint8( frame );
axesChild = get( axisName, 'Children' );
set( axesChild,'cdata',frame ); drawnow;
```

CURRICULUM VITAE

Name Surname: Emre Özdemir

Place and Date of Birth: İstanbul / 14.09.1990

E-Mail: emrozdmr@gmail.com

EDUCATION:

B.Sc.: Istanbul Technical University / Geomatics Engineering 2012

PROFESSIONAL EXPERIENCE AND REWARDS:

- Harmoni Real Estate Appraisal Company, Assistant Expert, June 2012 – September 2013
- Izmir Katip Celebi University Geomatics Engineering Department, Research Assistant, September 2013 - …

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Emre Özdemir, Özşen Çorumluoğlu, İbrahim Asri (2014). Stereofotogrametrik Sürüş Destek Sistemi. *7. National Engineering Surveying Symposium*, October 15-17, Çorum, Turkey

OTHER PUBLICATIONS, PRESENTATIONS, AND PATENTS:

- İbrahim Asri, Emre Özdemir , Özşen Çorumluoğlu (2014). 3D Photogrammetric Modelling of Cultural Heritage: Samples in Turkey. Turkish-German 3D Spatial Information Technologies Workshop
- İbrahim Asri, Özşen Çorumluoğlu, Emre Özdemir (2015). Uydu Verilerinden Elde Edilmiş Sayısal Yükseklik Modeli ile CBS Ortamında Orman Yangın Gözetleme Kulelerlerinin Görebilirlik Analizi: İzmir Örneği. Turkish National Photogrammetry and Remote Sensing Association, *VIII. Symposium*, May 21-13, Konya, Turkey
- İbrahim Asri, Özşen Çorumluoğlu, Emre Özdemir (2015). CBS destekli Orman Yangını Risk Dağılım Analizi; Antalya Örnegi. *15. Turkish Scientific and Technical Mapping General Assembly*, March 25-28, Ankara, Turkey