LEHRSTUHL FÜR VERKEHRSTECHNIK - TECHNISCHE UNIVERSITÄT MÜNCHEN

UNIVERSITÄTSPROFESSOR DR.-ING. FRITZ BUSCH

Arcisstraße 21, 80333 München, Tel. (089) 289 – 22438

MASTER'S THESIS

Accurate Estimation of Travel Times on the Traffic Data, Extracted from Aerial Image Time Series

Author:

Mihir Shah

Supervisors:

Dr.-Ing. Franz Kurz (Deutsches Zentrum für Luft und Raumfahrt, DLR)
Dr.-Ing. Matthias Spangler (LSt Verkehrstechnik, TUM)
Dr.rer.nat. Dominik Rosenbaum (DLR)

Munich, November 2013

TABLE OF CONTENTS

ABSTRACT

The ground stationary measurement devices are barely able to handle the daily traffic volumes and thus there is a high chance they could fail in situations arising from catastrophes or traffic disasters. Therefore, a real-time airborne monitoring system for such unforeseen circumstances was developed in the last years at the German Aerospace Centre (DLR). This system has the capacity to estimate road traffic information such as vehicle positions and vehicle velocities by tracking through image sequences. This Thesis focuses on developing a robust and feasible algorithm for automatic estimation of travel times on a motorway from the extracted traffic data using aerial image sequences. The used aerial image time series were captured on the motorways of Cologne and Munich by the airborne monitoring system. Preliminary analysis was carried out on the image data and individual vehicle velocities were derived from the image time series and consolidated. NAVTEQ road database was used to detect road areas along the motorway so that the tracked vehicles can be assigned to a road segment and point-to-point travel times for shorter distances can be obtained. Travel times for the route are derived and also predicted for some road segments by taking into account the traffic flow. The approach for the calculation of travel times is based on the identification of traffic state. For the validation of proposed methods, results are also compared with the manual deductions of ground stationary infrastructure as well as with the times recorded through the ground run campaign along a 16 km motorway segment of Munich by an ADAC reference vehicle. The conclusion and shortcomings of the approach are discussed at the end of the thesis.

<u>Problem:</u> Implementing interfaces between the extracted traffic data and developing a concept to accurately estimate travel times; developing a robust algorithm to calculate travel times in real-time by detecting different traffic states/flows in real-time; Comparison of Results

<u>Objective:</u> Near real-time estimation of travel times based on the traffic data extracted from Aerial Image time series is the focus

<u>Approach:</u> Getting familiar with the traffic data already obtained from aerial image sequences; developing a concept to obtain travel times; Analysis/Comparison of the results with the travel times calculated from conventional stationary measurement system and ADAC reference vehicle.

1 INTRODUCTION

1.1 Background and Motivation

The on-going trend of urbanisation and growth of population in the cities makes it necessary to have adequate means of road traffic monitoring. Most cities in the world today greatly rely on the use of ground stationary infrastructures such as induction loops, radar sensors and traffic cameras for traffic monitoring on the motorways. But the fact of the matter is that they could fail altogether in situations arising from mass events or disasters where there is damage to the stationary infrastructures. The result would be a complete lack of information or blackout. Mobile measurement units such as floating car data (FCD) which flow with the traffic are well suited to estimate travel times but the data is incomplete in terms of wide area coverage and insensitive to short-term congestions (Schaefer et al. 2002), (Busch et al. 2004). Thus, an airborne traffic management system as shown in the Figure 1 finds its application in these situations.

The background and motivation of the thesis comes from Project VABENE undertaken by the Department of Photogrammetry and Image Analysis in DLR (German Aerospace Centre). Real-time monitoring of traffic is one important part of the project (DLR). The thesis aims at developing a robust algorithm to estimate travel times and predict them whenever necessary. Information of accurate travel times is valuable not only for traffic management but also for security related organisations. They are interested in faster transfer of logistics to the disaster area and thus need to know the travel times around the particular disaster area. In general, travel time estimation is based on the data from ground measurement systems. One of the disadvantages of ground measurement methods (except FCD) is the low spatial resolution whereas Ernst (2005) stated that airborne imagery provides high spatial resolution combined with acceptable temporal resolution depending on the flight repetition rate (not in this system) and wide area coverage. The big advantage of this airborne system is that it is flexible and can be applied nearly everywhere (with the exception of tunnel segments, bad weather and nights) without being dependent on third party infrastructure. Even the ground station is designed to be self-sustaining, supplied by a power generator. An application or use of the travel times derived is the generation of isochronal maps, which shows up-to-date travel times from each map position to the accident scene (Kurz et al., 2007). But one of the disadvantages of airborne imagery is that it requires complex image analysis methods and traffic models to derive the desired traffic parameters (Ernst, 2005). “The system can also be combined with the disaster management tool to provide information and aerial image sequences to the relief forces in real-time during an event of breakdown of other infrastructures which can prove to be really useful" (Frassl et al. 2010).

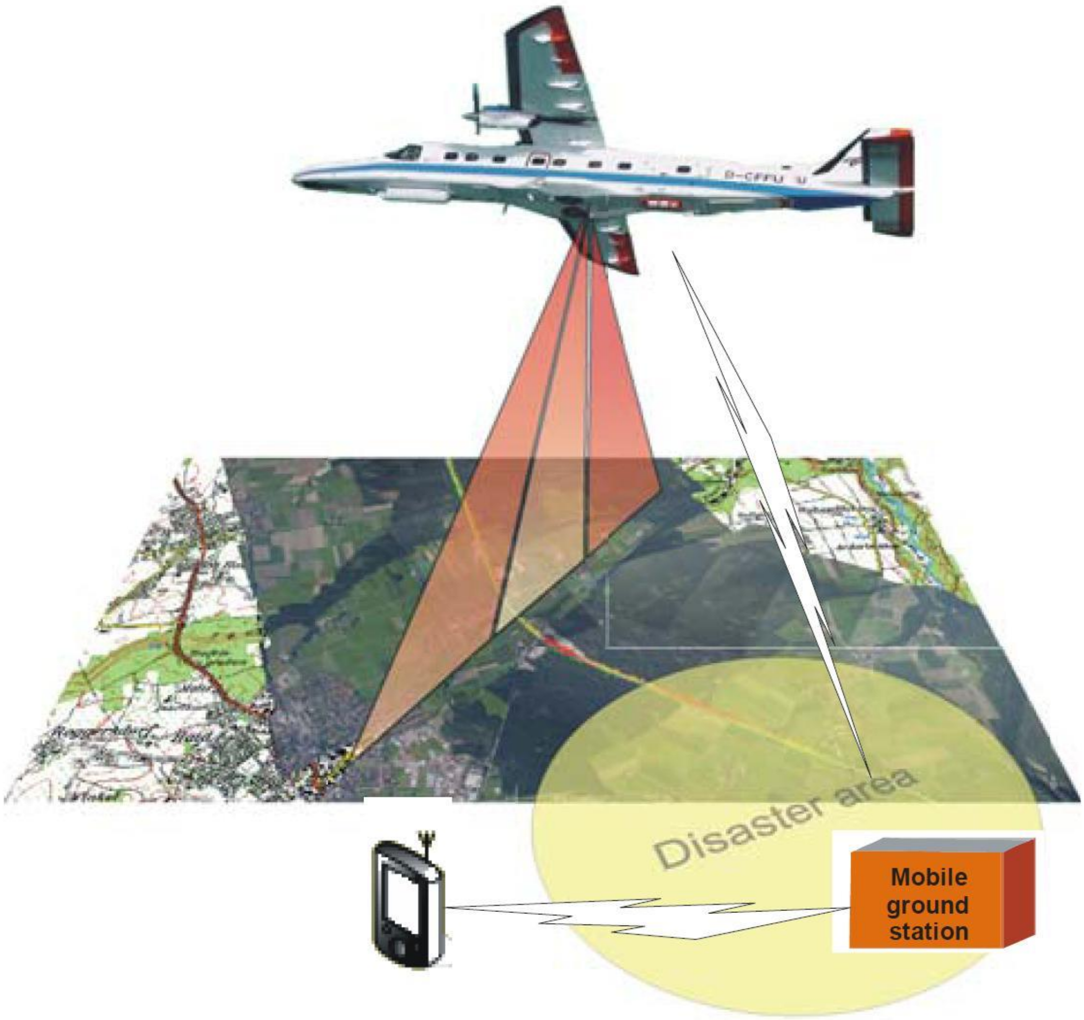

Figure 1: Aerial overview of the system (DLR 2013)

In this thesis, the approach to develop an algorithm for accurate estimation of travel times from the geocoded aerial image time series is discussed. The various methods taken into account for the development are addressed and the quality of extracted travel times are validated and evaluated.

1.2 Research Approach

The target of the study is to automatically estimate travel times on motorways based on the traffic data extracted using aerial image time series. It requires development of a robust algorithm.

The initial work relies on understanding the system hardware and the pre-processing, processing and collection/extraction of data from image sequences. The next step is to get familiar with the extracted traffic data and review the available literature regarding previous research. The interfaces are then implemented between the huge amounts of data. In addition to providing the interfaces, a concept is developed to obtain the travel times. Algorithm is conceived in MATLAB.

This system hardware can generate road traffic information such as vehicle positions and even track vehicles from the aerial image sequences. The individual vehicle velocities are derived from the known vehicle positions in two consecutive geocoded images. It is done by calculating the distance covered over time elapsed. The generated traffic information is then aggregated and consolidated. NAVTEQ road database is used to detect road areas around road segments which are projected into the orthorectified aerial images and the detected vehicles are assigned to the corresponding road segment so that instantaneous velocities can be calculated. These instantaneous velocities are then aggregated and local as well as momentary velocities for each segment are derived. Since the length of the each route is known, Travel times for the sections are derived and also predicted (in some cases with no recorded data) by taking into account the traffic flow in real-time. The method of prediction changes according to the identification of traffic state. The flow is estimated according to the regulations of MARZ textbook.

The travel times and velocities are assessed for varied traffic dynamics such as smooth traffic flow with low densities and traffic jams with high densities for the validation of proposed methods. The results are also compared with the manual deductions of ground stationary infrastructure as well as with the times recorded through the ground run of an ADAC reference vehicle for the measurement of accuracy. At the end of this thesis, some important conclusions as well as deductions of the system are discussed.

1.3 Structure of the Thesis

This thesis is divided into additional chapters, each of which is briefly described below:

Chapter 2: Overview of the system

This chapter does a literature review, in which different aspects of the system are described. It includes introduction into system hardware as well as the methods adopted for the generation and extraction of traffic data. Previous attempts to record traffic data from aerial surveys have been summarized. The algorithm source code written for this thesis has some foundation in the topics discussed here.

Chapter 3: Estimation of Travel times

This chapter thoroughly explains the process of accurately estimating travel times and developing a robust algorithm. It begins by describing how and when traffic data is acquired; it then discusses the method to estimate individual vehicle velocities and the consolidation of extracted traffic data. Then it focuses on the aggregation of traffic data by creating routes and road segments. Lastly, the prediction of travel times is described by a decision flowchart of developed model.

Chapter 4: Analysis of Generated Results

This chapter focuses on the generated results. Firstly, the results are summarized and discussed. Then they are compared with the results from different sources and sensors. The reason for differences in the values is also provided.

Chapter 5: Conclusion

A Summary of the research is made and some general conclusions are stated based on the comparison analysis.

2 OVERVIEW OF THE SYSTEM

2.1 System Hardware

The real-time system can be divided in to several sub-systems, namely on-board system, on-board computer and the ground station. The aircraft system consisting of the on-board system and computer is controlled by two on-board operators. The on-board system consists of a sensor system, a computer network system and a radio link. The on-board computer network system consists of industrial PCs with up-to-date hardware (Core 1-7 CPU, 16 GB RAM, SSD Drives, NVIDIA GeForce 9800 GTX GPU / 512 MB memory / computer capability 1.1) and a gigabit switch. The on-board computer reads the images out from cameras and performs steps like image storage and image/navigation data stream synchronisation, direct geo-referencing / orthorectification and generation of road traffic information. And the ground system consists of a receiving antenna and a computer network. Flight position and attitude is recorded synchronously by an IGI GPS/INS navigation unit with real-time capabilities. (Leitloff et al. 2013).

The camera system consist of three non-metric off-the shelf cameras (3K = 3Kopf). The cameras are arranged in such a way that two of them look in obliquely sideward direction and one looks in the nadir direction as shown in Figure 2 on the next page. The camera system is coupled to the navigation unit mentioned above so as to enable the geo-referencing of 3K optical images. Due to the limited capacities of cameras for the high frame rates, the sensor system records images in the so called image burst mode. These bursts are short sequences with a high repetition rate. After an image burst, the camera stops for several seconds until the next burst is triggered. This burst might contain 3 or 4 images. The data recorded in Cologne has an image burst containing 3 images while the data recorded in Munich has a burst of 4 images. Generally, an image burst (consists of three or four images) is recorded every 7 seconds whereas every image of this burst is recorded approximately every 0.7 seconds. The 3K camera has a significantly higher frame rate than metric camera systems, which allows it to record movements on the ground like road traffic ((Kurz et al. 2007) (Leitloff et al. 2013)). In the research, only the image bursts in the nadir direction are taken into account because the cameras placed obliquely do not cover motorways. Vehicle detection is always performed on the first image of the burst and vehicle tracking is performed using consecutive geo-coded image pairs within the burst.

Figure 2: 3K+ System consisting of three Cameras (Gellert et al. 2013)

2.2 Generation and Extraction of Traffic data

Two datasets were used for the estimation of travel times. The images were captured in Cologne and Munich. The first dataset was acquired between 14:01 and 15:11 (German time) on 2nd September 2006 in Munich on the motorway A8, south of Munich. The other dataset was acquired on 17th September 2011 from the different motorways near Cologne. The times are stored in the form of GPS timestamps.

Image pre-processing is required before the processing steps are implemented. In the first pre-processing step, the image stream of each camera and the data stream of the GPS/IMU navigation system are synchronised in each camera PC. Orthoimage preparation is performed for each camera viewing direction for the purpose of vehicle detection. The images are overlaid with road axes obtained from a Navteq database. This is done in order to reduce the search area for the detection of vehicles and limit it to road areas (Leitloff et al. 2013)

The processing steps necessary for the generation of road traffic information are performed on board the aircraft. Only the conversion of traffic information is done on the ground to fulfil interface specification of the internet traffic portal. The two main tasks of processing the images include vehicle detection and vehicle tracking. Fast vehicle detection procedure is done by employing various algorithms which help in the pre-classification, blob detection and final classification of the vehicles. These algorithms take into account the following geometric features (Steger et al. 2008) and radiometric features (Haralick et al. 1973). The geometric features include area, circularity, rectangularity, compactness and eccentricity whereas the radiometric features include entropy, local homogeneity, energy, correlation and contrast. “The proposed vehicle tracking is performed using explicit shape models generated on each vehicle detected on the roads. Tracking takes place between two consecutive geo-coded images of an image burst. It is performed by matching the vehicles in first image with the vehicles in the following image. The search area for vehicles in the second image is based on the detected position, assumed travel direction and the maximum vehicle speed expected from first image. Vehicle with the best matching score is assumed to be the correct match. The entire processing chain is real-time capable. Thus, an image burst for each camera’s looking direction can be processed and analysed by the processing chain before recording of the next burst has been finished” (Leitloff et al. 2013).

The detecting and tracking of vehicles can also be done manually to extract the information from aerial images. It was done during the research so as to get an idea of the extraction and be familiar faster with the huge amount of data. iDibias and xDibias softwares were used to detect and track vehicles.

2.3 Related work

The various methods of travel time data collection are fairly well established. Several of the emerging or non-traditional ground measurement techniques are based on using point vehicle detection such as induction loops or video cameras. The different methods employed are extrapolation method, vehicle signature matching, platoon matching and aerial surveys.

Extrapolation method estimates travel times based on the instantaneous speeds calculated by ground detectors between detection sources for a short road segment as done later for the comparison. Vehicle signature matching estimates travel times by matching vehicle signatures between different points of detection whereas platoon matching estimates travel times by matching characteristics of vehicle platoons, such as positions and distances, between different points of observation. And aerial surveys measure travel times by calculating vehicle speeds and densities. The first three methods mentioned above are based on the ground stationary infrastructure.

Several types of air surveys have been used or tested to measure traffic flow and other parameters. These surveys were conducted using fixed-wing aircraft, helicopter, observation balloons, or even satellites. One of these surveys dates back to 1965 when a transportation consultant in Maryland used a fixed wing aircraft to collect congestion and traffic information. In those days, vehicles within a given section were counted manually to give traffic density. Using known ground distance they were able to calculate vehicle speeds and thus the level of service. Trained observers could estimate the density and corresponding level of service without calculating actual vehicle densities ((Jordan, 1965) (Skycomp Inc, 1995)).

Researchers from the University of Karlsruhe in Germany examined the matching of vehicle images from aircraft in 1987. The system was tested in Austria with some problems occurring because of the curvature and slope of the area. Several improvements were suggested though (Wiiliams, 1997). The potential of using observation balloons and satellite imagery was discussed in two papers. They primarily focused on the collection of traffic volume and densities as well as observations of congested traffic conditions ((Littleton, 1996) (Merry et al. 1996)). Georgia Tech Research Institute of the USA also tested a traffic surveillance drone that can capture video images from a high altitude of 8 to 16 km. The prototypes were capable of 30 minutes of flight at maximum speed of 30 knots. (ITS World, 1997).

In the recent past, Dr. Kurz (2007) and his team at the German Aerospace Centre (DLR) developed a camera system to capture images and record traffic information. An aerial overview of the data collection is shown in Figure 1. They proposed an instantaneous method to calculate travel times but the methods used are outdated. The image matching is introduced and detection as well as vehicle tracking is real-time capable. Thus, no concrete work has been done in precisely estimating travel times from Aerial Images. Hence, this thesis develops a robust algorithm to accurately estimate and predict travel times in varied traffic dynamics such as smooth traffic flow with low densities and/or traffic jams with high densities.

3 ESTIMATION OF TRAVEL TIMES

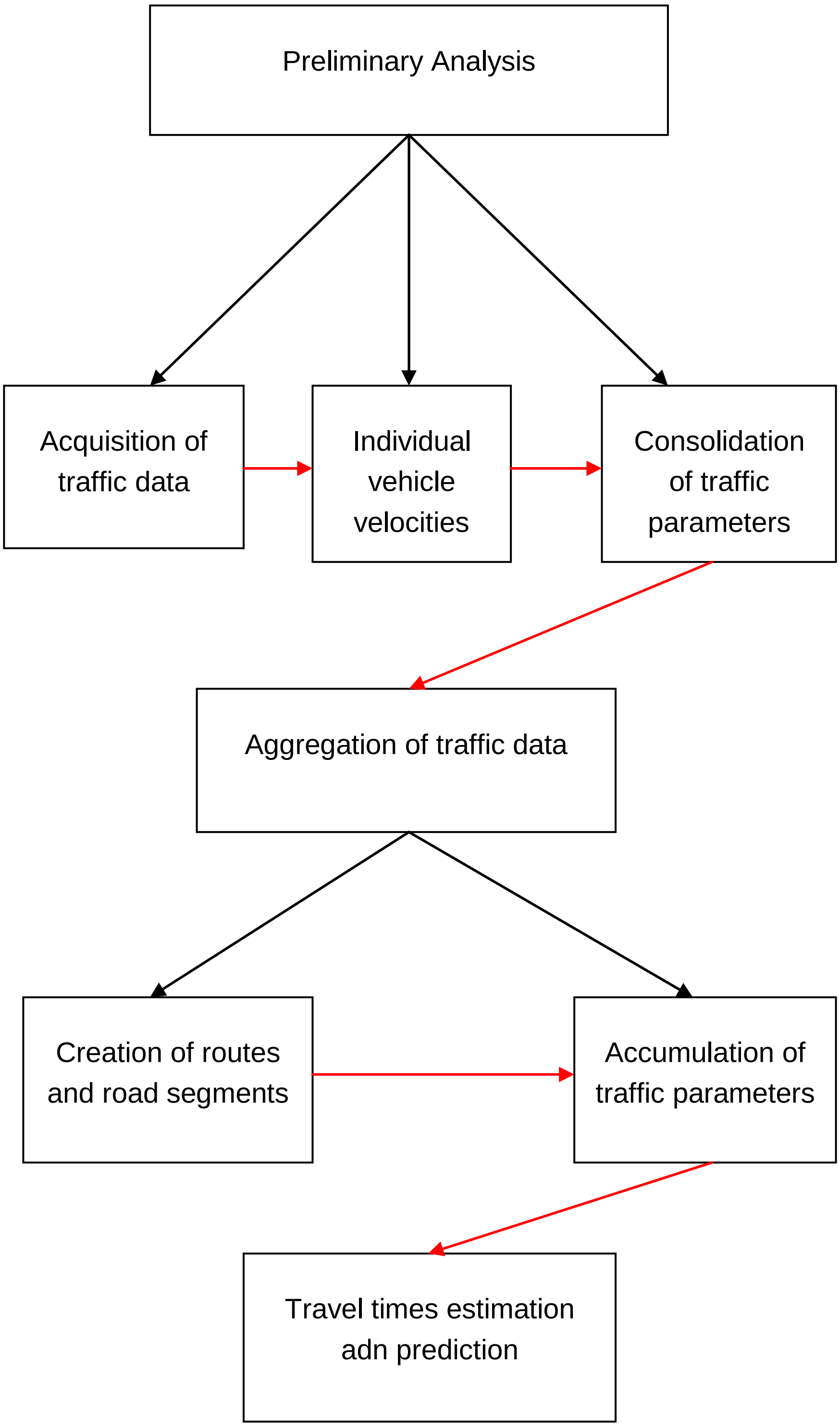

Figure 3: Flowchart for estimation of travel times (→ flow)

3.1 Preliminary Analysis

3.1.1 Acquisition of Traffic Data

For the estimation of travel times, two datasets were used. The images were acquired in Cologne and Munich. The first dataset was acquired between 14:01 and 15:11 on 2nd September 2006 in Munich on the16 km motorway strip (A8), south of Munich between motorway junctions Hofolding and Weyarn. Each image burst consisted of 4 consecutive geo-coded images. 22 image bursts were recorded with the time difference between each of them calculated at around 7 seconds. Image numbers are ranging from 11 to 100.

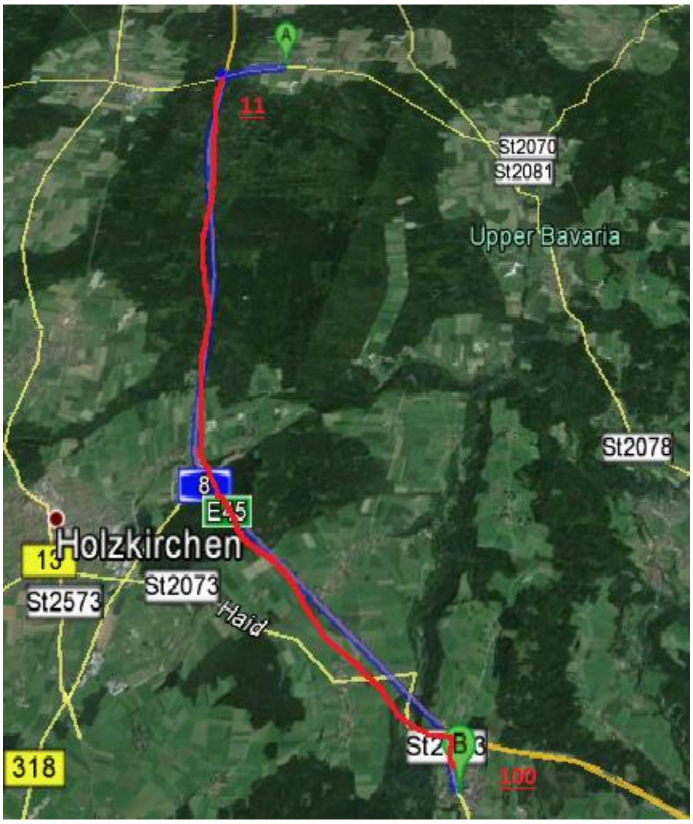

Figure 4: 16 km motorway strip on A8. A-Hofolding, B-Weyarn (Google Maps)

The other dataset was acquired on 17th September 2011 from the different motorways near Cologne. The images selected for the research purposes were captured between 12:57 to 12:59 and 13:10 to 13:12 German standard time. However in this data set, 49 image bursts were recorded and each burst contained 3 images. The times are stored in the form of GPS timestamps. Location of the captured images is shown in Figure 4 and Figure 5.

The flight path of the aircraft in Cologne is shown in figure 5 below. Some part of the flight path for images 53-80 and 108-143 overlaps.

Heavy traffic with high densities was expected in the section south of Munich caused by the homebound travellers, whereas smooth traffic flow with low densities was expected in the preferred sections on motorways near Cologne.

Figure 5: Approximate Flight path around the motorways in Cologne (Google Maps)

The aerial images taken by the aircraft are geo-referenced and orthorectified to the UTM (Universal Transverse Mercator coordinate system) coordinates. Thus the locations of cars detected and tracked are stored in UTM coordinates. This coordinate system is different from the latitude and longitude method.

The software used to acquire and display the data manually is iDibias. The information of the detected cars such as their locations is saved in files 'ON000x.car_detected' for every image. The tracking information was extracted by side-by-side mode in iDibias as shown in figure 6. It is stored in '_cpdb1_ON000x_1' file where x is number of the image. Thus for example, a file of name '_cpdb1_ON0004_1' will have the location of the individual cars tracked from Image 4 to Image 5 in side-by side mode. The same is done for all the images taken with the camera looking in Nadir direction.

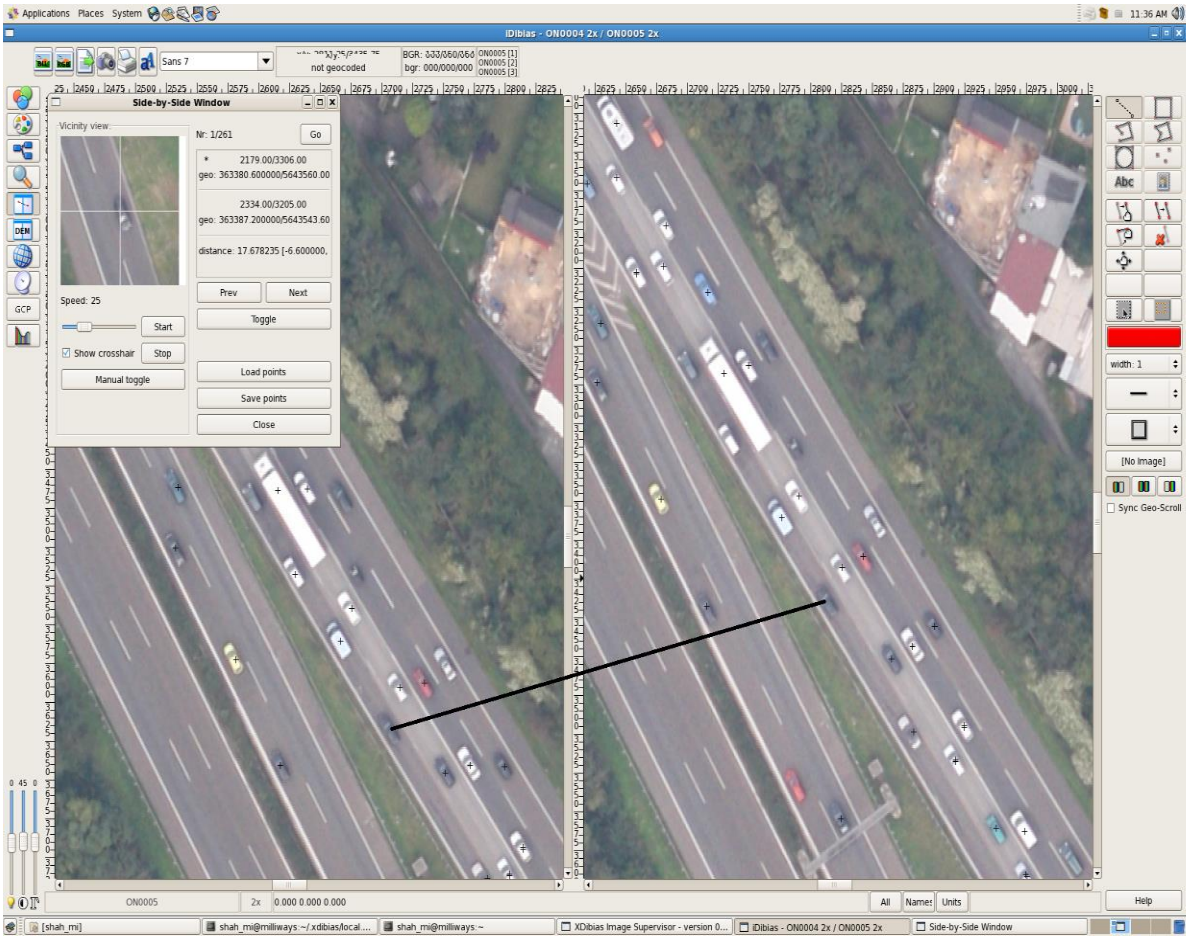

Figure 6: iDibias side-by-side mode for tracking vehicles

The images taken from the two sideward cameras were deemed not important for the research. Because images acquired by the obliquely sideward cameras cover the small/side roads linked to the motorways which are not required for travel time calculation. The geo-referencing data of the images is recoded in file '_aux' file which includes the UTM co-ordinates of the image and their GPS timestamps. The timestamps are calculated from the base date of 10th September 2011 in case of the

Cologne dataset. The unit of storage is seconds. All the extracted information from every image is stored in a folder named ‘ON000x’.

3.1.2 Estimation of Individual Vehicle Velocities

The first step in the accurate estimation of travel times is to derive individual vehicle velocities. Thus, data in files ‘ON000x.car_detected’ and ‘_cpdb1_ON000x_1’ are read in MATLAB (MATLAB 2012b) to get information about the detected and tracked vehicles. These files contain the pixel and UTM coordinates of the vehicles in addition to the object amount (total number of vehicles per image), GPS timestamps and correctness, completeness and quality of the data. The correctness and completeness of the data plays a major role in knowing the quality of the extracted data. The Tracking data contains the UTM co-ordinates of the same vehicle twice from the first and second image of the burst respectively. Data is mixed with numbers and strings, so the format of the sources is defined in the algorithm. The third image of the burst is not used for tracking as first and second image has a better overlap to track more vehicles because of a small time gap of 0.7 seconds whereas the time gap is 1.4 seconds between first and third image.

All the images are taken into consideration from the Munich dataset because the entire flight path is on motorways. However, only Images 27-52 and 81-107 are considered from the Cologne dataset because the other images acquired show traffic on the urban roads.

The individual vehicle velocities are then derived from the known vehicle coordinates in two consecutive geocoded images by calculating the distance covered over elapsed time as illustrated in the equation below. The elapsed time equals the time difference calculated between the capturing of first and second image of each burst. In the similar way, velocities of every individual vehicle tracked in every image are derived and stored. The velocities are stored in an array matrix of a*b where a (rows) is the amount of vehicles tracked and b (columns) is the image number. The number of vehicles (a) tracked in different images are different, so some values of velocities are stored as 0 because of a*b matrix. It is depicted in Figure 7 on page 18.

$$v = \frac{\sqrt{(X_1^{UTM} - X_2^{UTM})^2 + (Y_1^{UTM} - Y_2^{UTM})^2}}{\Delta t} \tag{3.1}$$

Where,

$v =$ Velocity of the individual vehicle

$X_1^{UTM} =$ X – Coordinate of a particular vehicle in first image

$X_2^{UTM} =$ X – Coordinate of the same vehicle in second image

$Y_1^{UTM} =$ Y – Coordinate of a particular vehicle in first image

$Y_2^{UTM} =$ Y – Coordinate of the same vehicle in second image

$\Delta t =$ Time difference between the GPS timestamps of first and second image

The velocities derived by the above equation are instantaneous velocities because of the small time difference.

In some cases such as the flight path from Images 27-52, it is also possible to segregate velocities and vehicles according to their driving directions that is either northbound or southbound. It can be done by using simple Geometry. Hence, you have a count of the vehicles moving in a particular direction which can be used later to check the automation capacity of the code.

3.1.3 Consolidation of Traffic Data

Ideally, all the false positives (Vehicles which are detected and tracked even if non-existent) should be identified and deleted and the false negatives (Vehicles which exist but not detected or tracked) should be added to the data before beginning the analysis. False positives are obtained due to the misidentification of the shadow of trees or marks of the road as vehicles (Rosenbaum et al, 2010). But in this thesis, the information at microscopic level (related to every individual vehicle) is not important as the travel times are to be derived for the entire route or road segment and not for individual vehicles. So, the false negatives resulting from the automatic vehicle detection and tracking algorithms is not corrected. However, most of those false positives are deleted with the help of below mentioned methods.

There are certain inaccuracies in the preliminary analysis which can influence the results. In order to get accurate results, the omission of such discrepancies is necessary such as the amount of outliers. For example: If the derived velocity of a vehicle is less than 10 km/h on a motorway where free flow is detected, then it is considered as an outlier as shown in Figure 7 on the next page. They occur because of the false positives or due to the vehicles tracked in the small urban roads linked to the motorways. Thus, an upper limit and lower limit is set for the derived velocities. This is done by creating the percentile function. The percentile values used have a statiscal meaning and they are assumed as outliers. The limits used in this thesis are 5%ile and 95%ile. Another possibility would be to take the median value which is not influenced by a small number of outliers.

Variables - velocity

velocity

velocity <30x8 double>

	1	2	3	4	5	6	7	8	9	10
1	72.1789	130.1804	113.6333	137.9607	128.1323	144.6369	133.6365	145.0041		
2	134.3609	110.6148	148.0873	101.6451	115.0970	136.6778	135.2869	140.3494		
3	133.7477	123.1840	125.1986	121.8390	120.6759	114.3716	142.2377	110.4019		
4	120.6835	113.9404	128.5630	112.0453	129.2109	114.4416	117.6390	128.3575		
5	119.7906	141.0159	119.5362	132.4956	115.4238	112.0615	109.0983	110.3964		
6	86.0290	122.8781	119.4247	122.3460	113.3529	115.6754	122.3129	95.0263		
7	122.7569	125.3077	122.2672	103.4478	108.4222	113.5063	125.6290	122.5805		
8	116.2469	89.6786	119.3147	105.9783	109.2331	83.9789	117.0288	106.9874		
9	99.8795	122.1155	123.3080	119.5697	95.0543	83.8118	96.4319	88.2487		
10	109.4639	87.0042	125.0601	110.5832	101.4400	85.9292	120.4750	88.4439		
11	108.5038	113.2220	116.2081	102.9343	105.8229	18.0721	119.7263	87.7991		
12	3.4533	99.3181	107.9461	0.9390	91.9318	76.6542	125.7066	88.7339		
13	101.8399	142.4180	89.8355	131.2838	101.4445	138.5264	123.6911	88.9139		
14	123.3425	129.5344	138.2976	137.1459	144.6886	143.1266	108.1204	116.9408		
15	139.1071	137.4903	117.9268	65.0125	132.5730	137.6500	120.4783	118.9951		
16	123.8106	139.5355	121.3726	119.2399	136.0518	8.6622	128.0422	124.0898		
17	137.9389	131.5636	108.1042	117.8061	140.8392	123.0398	124.2128	120.9524		
18	134.0550	83.9619	88.4553	1.5868	130.4922	130.3898	91.4808	119.7734		
19	134.4810	108.8972	0	0	131.5126	114.8626	91.3720	109.1506		
20	129.3771	0	0	0	134.8997	124.5827	102.7596	116.7539		
21	114.9894	0	0	0	134.5415	116.3936	103.2772	93.6266		
22	114.8282	0	0	0	120.2662	116.9382	13.0130	95.0173		
23	0	0	0	0	117.0115	110.9992	95.3755	96.3871		
24	0	0	0	0	91.7781	90.7925	100.9453	87.4254		
25	0	0	0	0	110.1383	89.0977	95.4210	9.2357		
26	0	0	0	0	91.7607	60.6148	0	0		

Figure 7: False Tracking 1 - Inaccuracies in derived instantaneous velocities

The UTM coordinates of the vehicles contributing to the outliers are removed as well. This is done to eradicate another inaccuracy related to the driving direction. Another effective method of building road segments/sections by creating polygons is also implemented (Section 3.2.1). The polygons are created by taking into account the number of lanes per direction, thus removing all the other vehicles driving in the false direction or other lanes.

In Figure 8 on the next page, directions/arrows 1 and 2 on the right side of the vertical line are acceptable, since the lateral movement of the vehicle is possible during the lane changing behaviour or during a curve on the motorway. But the other directions 3 and 4 are inacceptable as the vehicle is moving in the other direction of the road. This particular segregation is done in order to exclude the vehicles from the opposite direction because they might have a different state of traffic flow. However these vehicles are automatically included in road segments (Section 3.2.1) of the opposite direction thus resulting in no loss of detected vehicles.

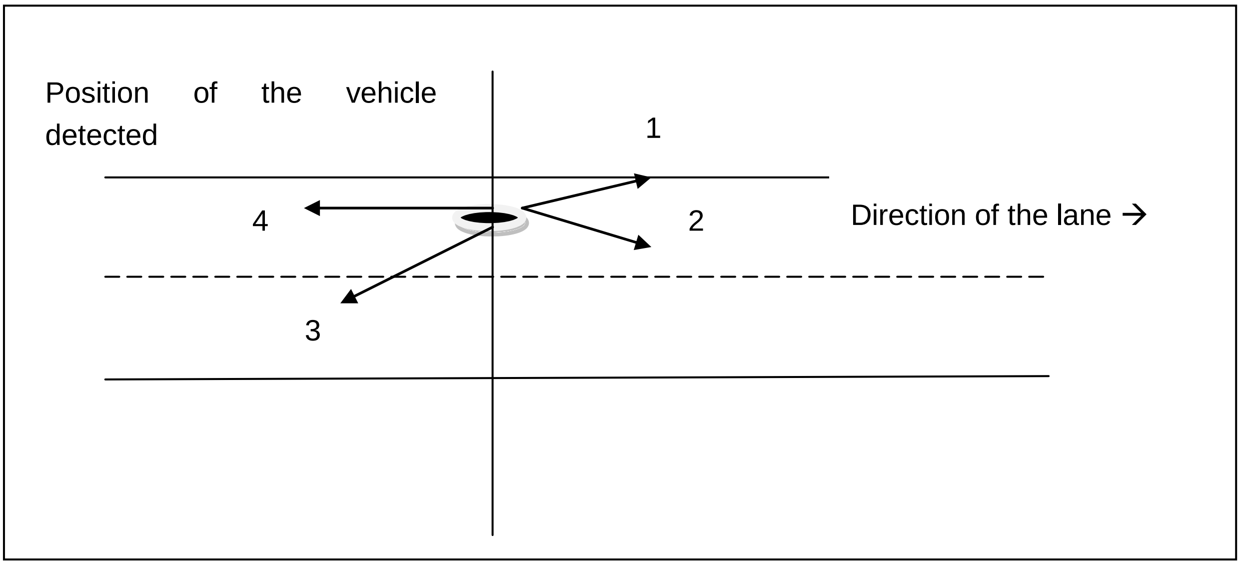

Figure 8: False tracking 2 – False driving directions

3.2 Aggregation of Traffic Data

3.2.1 Creation of Routes and Road segments

Data from a road database is used a priori information for the creation of ‘Routes’ and ‘Road segments’ along the flight path. One of these road databases is produced by the NAVTEQ Company (NAVTEQ 2006). Navteq files are extracted from this database which provides with the UTM coordinates of the centre line of the roads (nodes) captured in images. In addition to the coordinates of the nodes (centre line points), the files consist of section ids’, node ids’, Navteq ids’ and the number of lanes per each direction for every image. This information is stored in ‘ON000x.navteq_roads’ file. Validations of position accuracy by Dr. Kurz (2007) of NAVTEQ road lines resulted in 5m accuracies for motorways and recent validations assumed it to be 3.5m for motorways.

A new algorithm is developed to create routes along the flight trajectory. Routes in this case are defined as the specific arrangement of the nodes and their coordinates neighbouring each other. A new codes’ need arises due to the fact that centre line nodes are not aligned and its sorting is necessary to detect road areas according to the fight path. It needs to be arranged accordingly to build proper road polygons for each direction of the traffic flow so that the vehicles can be assigned to each polygon/segment.

The code reads information from ‘ON000x.navteq_roads’ for every first image of the burst, randomly selects any id and starts looking for the closest neighbouring nodes by the help of section ids and Navteq ids until it cannot find any other node on the same direction of the traffic flow. Hence, after the completion of 1 loop, a route in 1 direction of the road is formed. Information for each route is saved in a new created file ‘Routexx.txt’, where xx is the route number. The file contains node ids with UTM coordinates arranged systematically. In addition to the coordinates, the file also contains information on number of lanes at the position/location of the node, which is quite helpful to know the width of the motorway. The 51 routes created by the algorithm for Cologne dataset is shown in the figure 9 and figure 10.

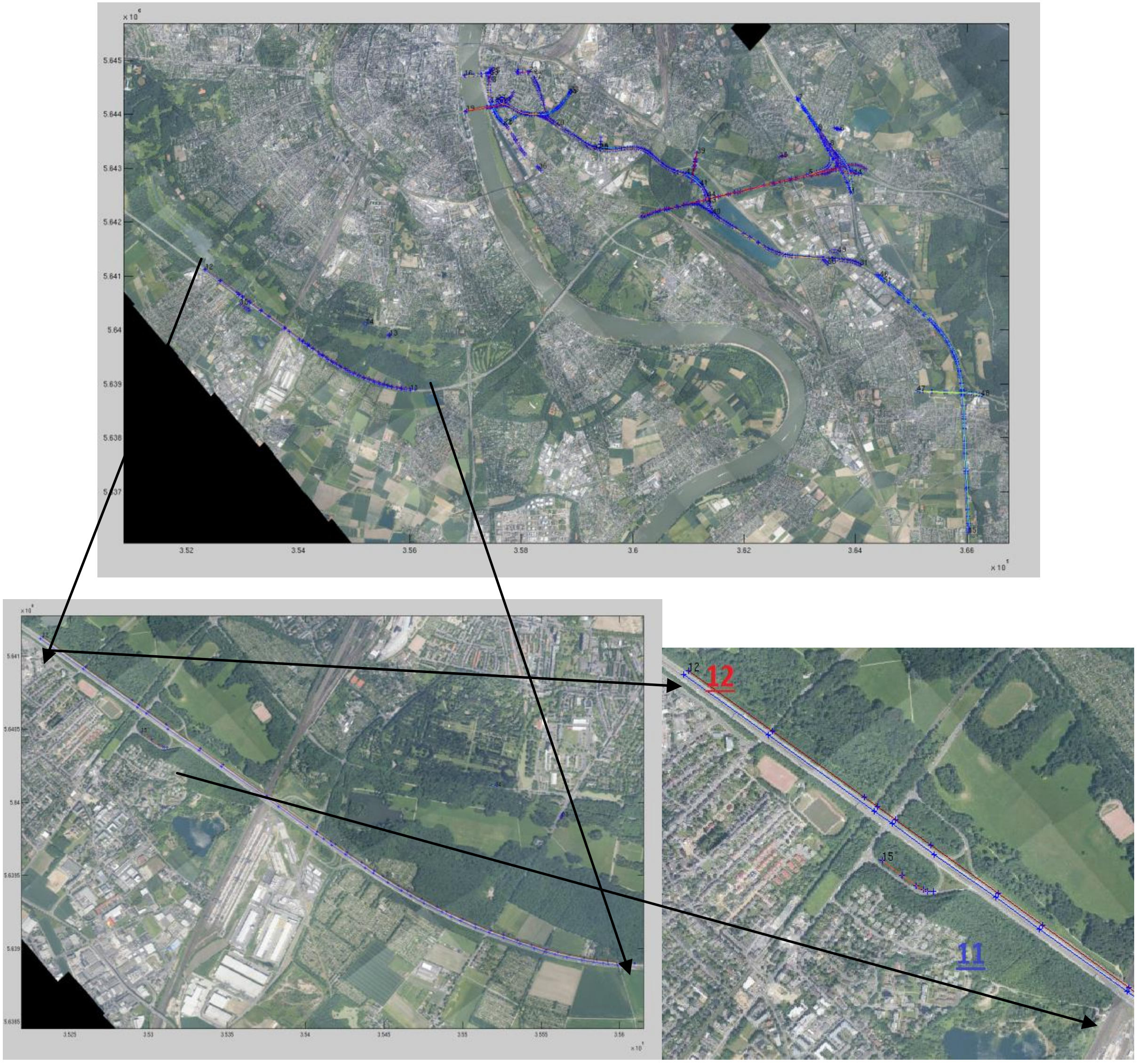

Figure 9: Magnified Routes for Cologne Dataset

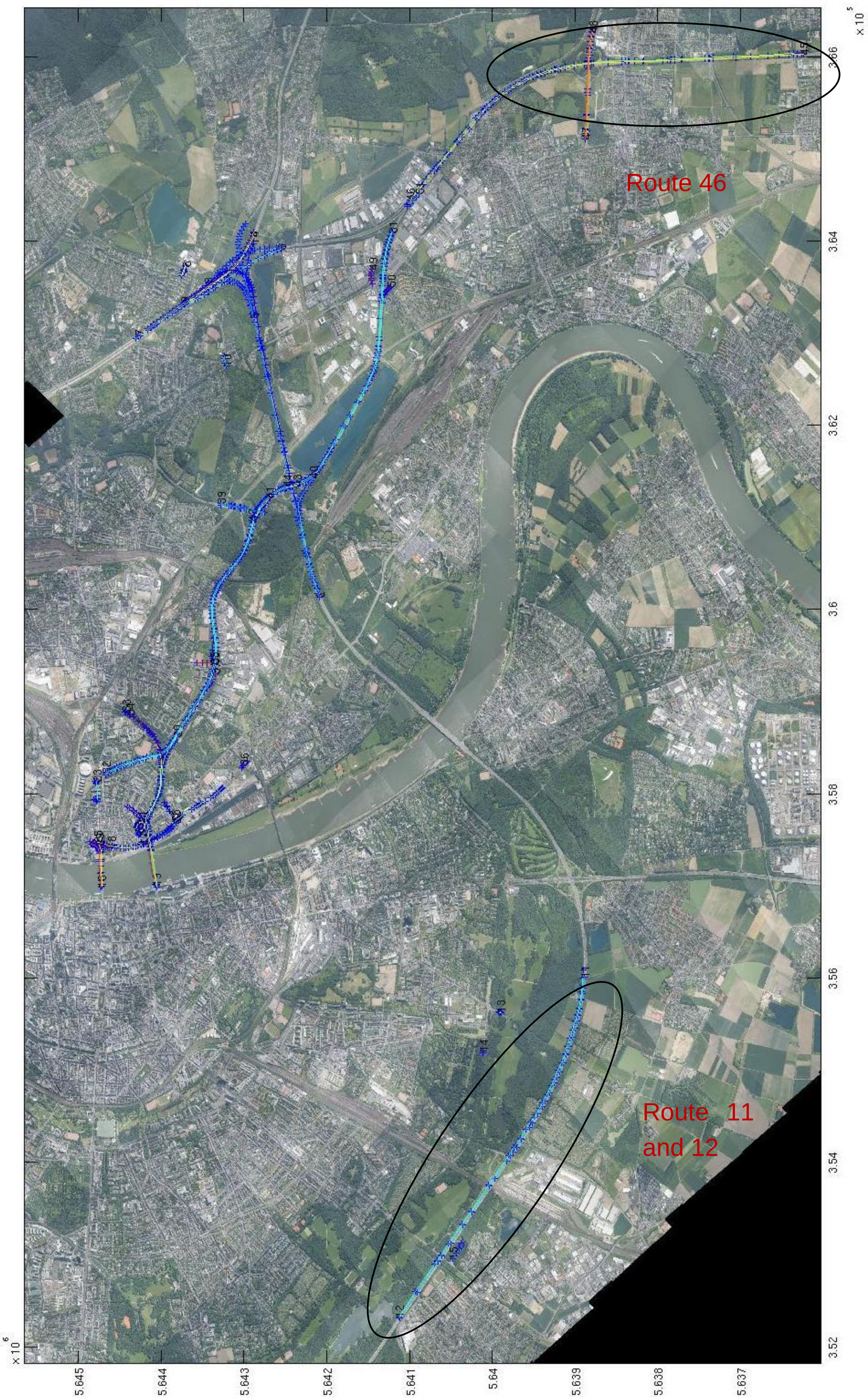

Figure 10: Routes

There are in total 51 routes which are shown in different shades of blue, orange and green. They are numbered as well from 1 to 51 (Figure 10). Route number 11 and 12 are highlighted in Figure 9 as they fall along the flight path from Image 27-52. ‘Route12.txt’ contains centre line coordinates of the road where traffic is flowing northward and ‘Route 11.txt’ contains southward flowing traffic.

The same is done for Munich dataset except the background image of city. However the derived 37 routes are manually aggregated to only two routes by adding nodes to only two files (Figure 11) for simplification and better comparison of travel times. One of the routes is moving northward and the other moving southward for the entire16 km motorway strip (Figure 11).

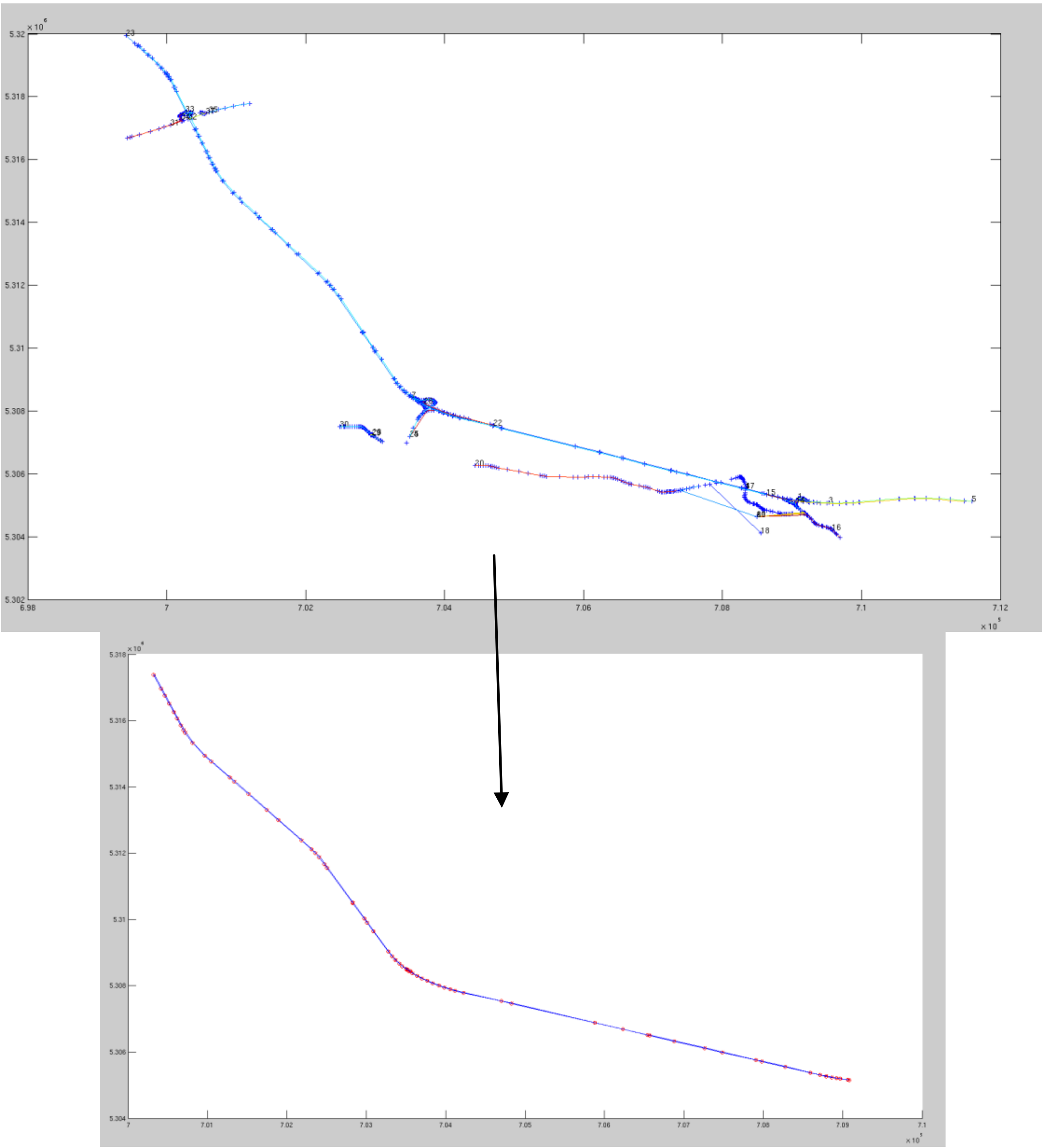

Figure 11: Aggregated Routes of the Munich Dataset

After the creation of the road polygons, it is essential to build road segments/polygons because the boundaries of these polygons are used to judge if the UTM coordinates of the tracked vehicles are inside or outside of the drawn section. The width of the motorway at every point is known because of the mentioned 'number of lanes' in 'Routexx.txt' files. The road polygons are formed by the following concept:

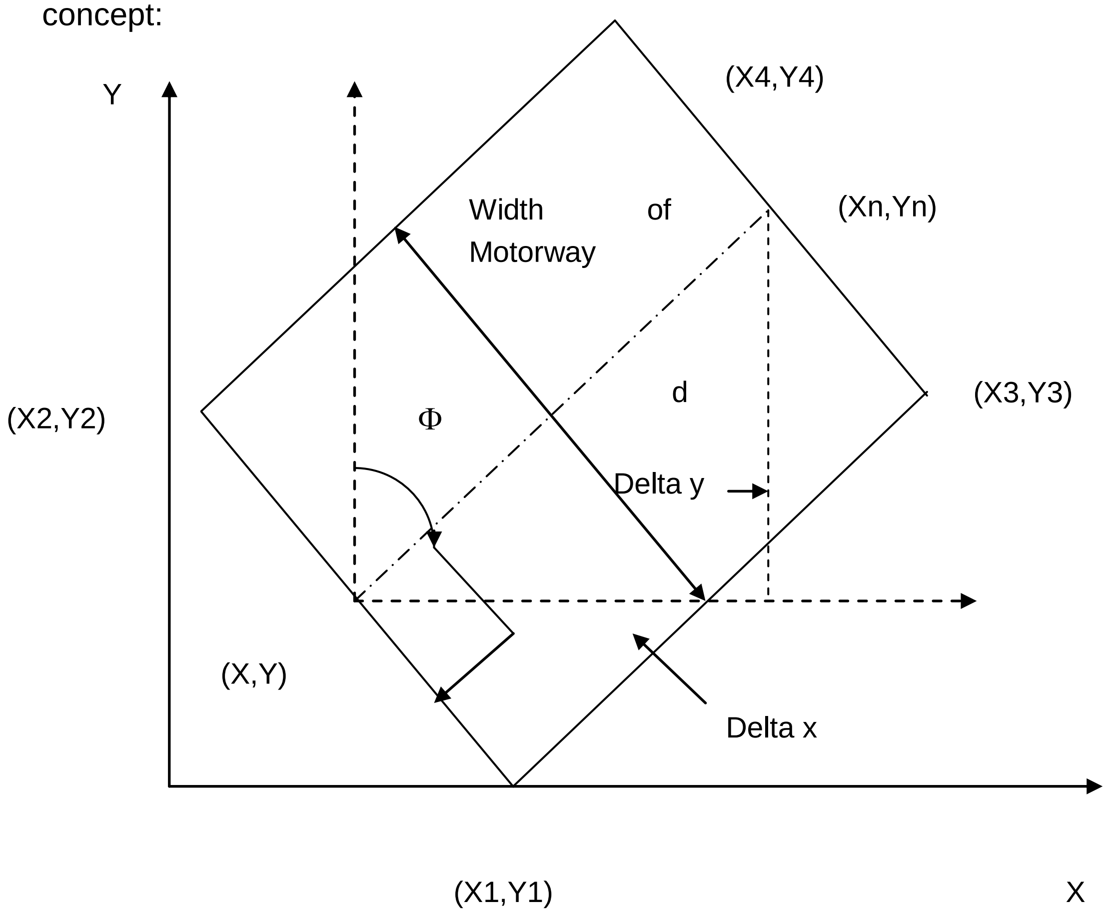

Figure 12: Building a road polygon

There are 2 methods to create a road segment from this concept. Here the coordinates of the 2 nodes are known. They are arranged systematically one after the other.

(X,Y) = UTM coordinates of the 1st node of the route

(X_n,Y_n) = UTM coordinates of the 2nd node of the route

(X_1,Y_1) = UTM coordinates of the 1st corner of the segment

$(X_2,Y_2),(X_3,Y_3),(X_4,Y_4)$= UTM coordinates of 2nd, 3rd and 4th corner of the polygon respectively

$Phi =$ angle Φ

$\sigma = numberoflanes * \frac{lanewidth}{2}$

$Deltax = \Delta x$ = Change in x coordinate from 1st to 2nd Node

$Deltay = \Delta y$ = Change in Y coordinate from 1st to 2nd Node

$d =$ Length of the segment or the distance between the first and second node

The equations used to calculate the accurate positions of the four corners of a polygon are illustrated below.

$$\tan\Phi = \frac{\Delta x}{\Delta y}$$

$$\begin{aligned}
X_1 &= \sigma\cos(\Phi+90) + X; Y_1 = \sigma\sin(\Phi+90) + Y \\
X_2 &= \sigma\cos(\Phi-90) + X; Y_2 = \sigma\sin(\Phi-90) + Y \\
X_3 &= \sigma\cos(\Phi+90) + X_n; Y_3 = \sigma\sin(\Phi+90) + Y_n \\
X_4 &= \sigma\cos(\Phi-90) + X_n; Y_1 = \sigma\sin(\Phi-90) + Y_n
\end{aligned}$$

Or

$$\begin{aligned}
X_1 &= (-1*\frac{\Delta y}{d}*\sigma) + X; Y_1 = (\frac{\Delta x}{d}*\sigma) + Y \\
X_2 &= (-1*\frac{\Delta y}{d}*-\sigma) + X; Y_1 = (\frac{\Delta x}{d}*-\sigma) + Y \\
X_3 &= (-1*\frac{\Delta y}{d}*\sigma) + X_n; Y_3 = (\frac{\Delta x}{d}*\sigma) + Y_n \\
X_3 &= (-1*\frac{\Delta y}{d}*-\sigma) + X_n; Y_3 = (\frac{\Delta x}{d}*-\sigma) + Y_n
\end{aligned} \tag{3.2}$$

This process of finding the four corners is repeated for the entire route. Second node in first road polygon becomes the first node in the next determination of segment boundaries. Hence, the road segments are drawn according to the coordinates and covers entire width of motorway for all routes as shown in Figure 13 on the next page. Length of the segment varies according to the centre line coordinates (nodes) given in the NAVTEQ files. The function of ‘inpolygon’ is used to handpick all the vehicles (points) present inside each road segment along the length of each route. Therefore, number of vehicles for each road segment is obtained and since the length of each segment is known, vehicle density can be derived. Vehicles detected in those segments are sorted according to their respective segments. Whereas, the corresponding instantaneous velocities derived previously are sorted according to their respective segments and images. Thus, a three dimensional variable is created for velocities.

Figure 13: Road segments

3.2.2 Accumulation of Traffic parameters

Accumulation of the derived traffic parameters is required to perform further analysis. Vehicle density and vehicle velocities are sorted out assigned to their segments. Now the velocities have to be aggregated to a single value per road segment. Densities for each segment are already known.

There are at least two ways to determine mean speeds:

Locally: $v_l = \sum_{i=1}^{M} v_i \frac{m_i}{M}$ (3.3)

Momentarily: $v_m = \sum_{i=1}^{N} v_i \frac{n_i}{N}$ (3.4)

Where, m/M = weight ratio of the individual vehicle speed

n = Number of vehicles with same speed

N = Total number of vehicles

Average local velocity is defined as the mean velocity of all the vehicles in a known cross-section over different periods of time and average momentary velocity is defined as the mean velocity of all the vehicles at a specific point in time (Busch, 2011). Thus momentary velocity is simply the mean of all derived velocities.

The sorted velocities are first aggregated according to the images and then according to the drawn segments. Therefore, local and momentary velocity for each segment as well as image is known. Traffic parameters are then scaled to Metric units of km/h and veh/km. Travel times in seconds for every segment were estimated by dividing the average local or momentary velocity (per segment so single aggregated value) over the length of respective segment.

3.3 Prediction of Travel Times

Travel times for shorter distances are thus known. But the total travel time for entire route is yet unknown due to the segments with no detected vehicles. Times for those segments have to be predicted in real-time and not at some point in near future.

One such method to predict the travel times is by surveying the historical database. The database should have contained traffic records showing trends on that particular segment which differ with different days as well as with different times during the days. Different traffic trends generated on public holidays or peak hours throughout the year should have been also included in this database. A rough idea of the format of this inexistent database has been given in the table below. If this database would have been available, then the prediction would have to be performed manually. This contradicts with the aim of this thesis to develop an algorithm to automatically calculate the results. However this database is not available and another method is implemented.

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
00:00	-----	----	----				
01:00	-----	----	----				
02:00	-----	----	----				
03:00							
04:00							

Table 1: Rough Idea on the historical database

Another method is based on the classification of traffic flow. Travel times can be predicted much more accurately by taking into account varying traffic dynamics at that point of time. MARZ document, which is sponsored by the Federal ministry of transport, building and housing serves as basis of identifying the flow of traffic for every segment with known velocities and densities. Several conditions for classification are shown below.

	1 lane		2 lanes		3 lanes		4 lanes	
Traffic Flow	*V [km/h]*	*D or K [veh/km]*	*V [km/h]*	*D or K [veh/km]*	*V [km/h]*	*D or K [veh/km]*	*V [km/h]*	*D or K [veh/km]*
Free Flow	≥ 80	$\geq 0, \leq 20$	≥ 80	$\geq 0, \leq 30$	≥ 80	$\geq 0, \leq 40$	≥ 80	$\geq 0, \leq 50$
Dense Flow	≥ 80	$>20, \leq 50$	≥ 80	$>30, \leq 60$	≥ 80	$>40, \leq 70$	≥ 80	$>50, \leq 80$
Slow moving traffic	$\geq 30, <80$	≤ 50	$\geq 30, <80$	≤ 60	$\geq 30, <80$	≤ 70	$\geq 30, <80$	≤ 80
Congestion	<30	>50	<30	>60	<30	>70	<30	>80

Table 2: Classification of flow (MARZ, 1999)

These conditions are used to develop a model to predict travel times. A decision flowchart is created to explain the model as shown in Figure 14; each section of the flowchart is explained as well.

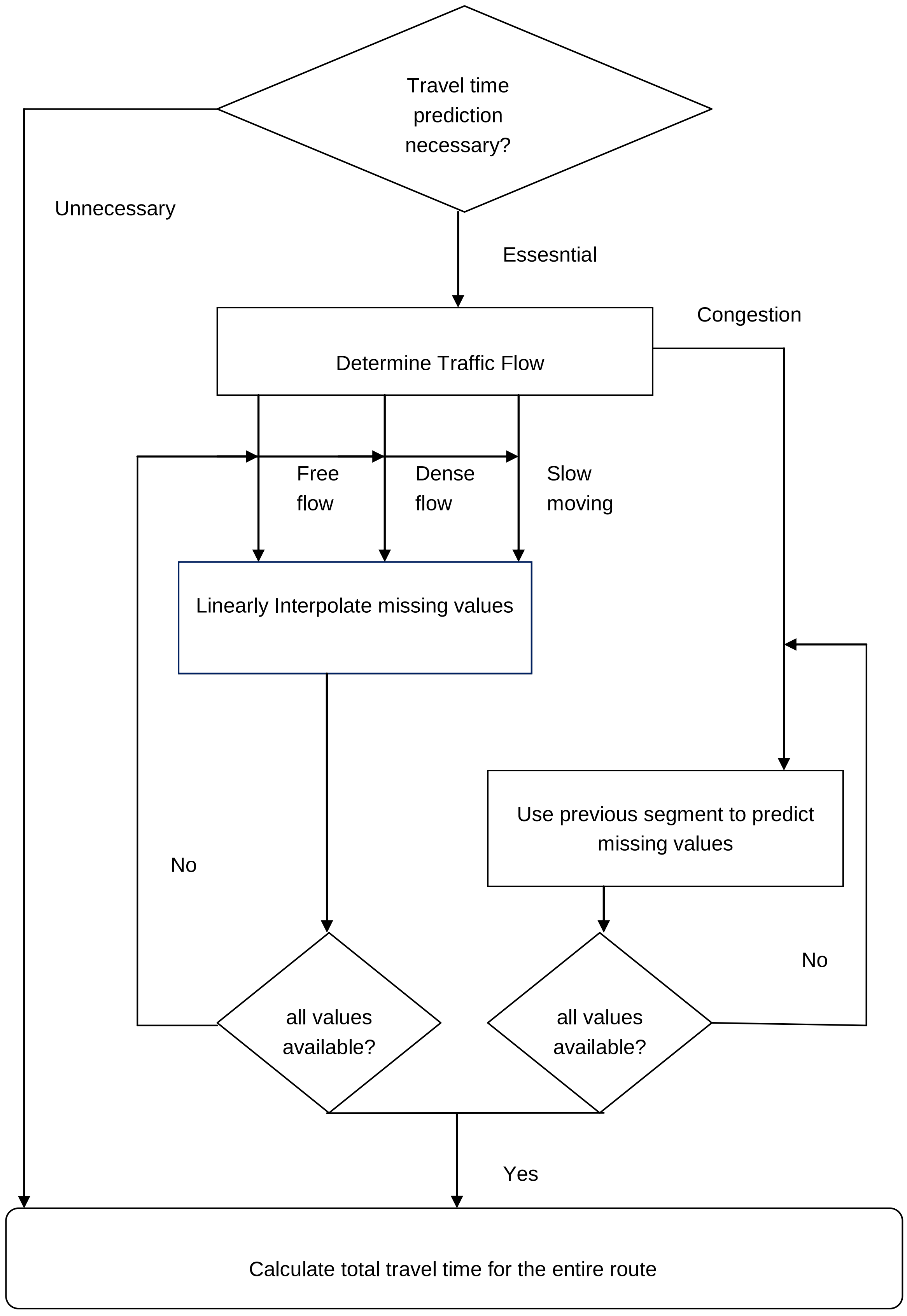

Figure 14: Decision Flowchart for prediction of travel times

Is Travel time prediction necessary?

For this model, prediction is considered unnecessary if travel times are already estimated for every segment of the route. This occurs rarely, only when the flight trajectory passes along every segment of the route with a prior condition that vehicles must be detected in those segments. However the prediction is essential if the values of travel times for various segments are missing which is common in most of the routes. This occurs often because in some cases the segments are too small to contain vehicles in it or if free flow is detected on the route, then it's not necessary that every segment contains vehicles.

Determine Traffic Flow

It is important to identify the traffic states for every segment before starting the prediction. Studies show that the density flow relation is not continuous between free-flow and congested state which can alter the travel times (Kerner, 2004). Traffic states are classified in four scenarios: free flow, dense flow, slow moving traffic, congestion. Conditions used are illustrated in table 2. Depending on the known number of lanes for the segment, conditions are checked one after the other. For the flow to be classified as free, dense or slow moving traffic, all the conditions for velocity V and density D should be valid, whereas for the flow to be classified as congested, any of them that is either conditions for velocity or density should be valid. However in some segments with no detected vehicles, no traffic state is identified. It is not taken into consideration during the identification of traffic state for entire route. Traffic flow for each route is derived by finding the most common traffic state amongst all segments of the route.

Linearly interpolate missing values

If the determined traffic state of the entire route is free flow, dense flow or slow moving traffic, then the missing travel times are statistically interpolated as the traffic dynamics in this case is not as complicated. Interpolation is carried out by considering travel times of the nearby segments by using an in-built function 'interp1' in MATLAB. This command interpolates between data points. It finds values at intermediate points, of a one-dimensional function (in this case travel times) that underlies the data (MATLAB 2012b). Missing values are interpolated based upon the length of the respective segments which varies depending on the centre line coordinates (nodes). Length of segments can range from 50 metres to 300 metres or more.

Use previous segment to predict missing values

If the determined traffic state of the entire route is ‘congested’, then the missing values are simply copied from the previous segment and altered according to the length of current segment. This is done because if the state is congested at that point of time, then the vehicles are moving at the same speed (if they were hypothetically present in the segment) and will require the same time (sec/km) for the current segment as required for the previous segments.

All values available?

If there are still some missing values for segments of a particular route, they are calculated by two different methods. In case of the free flow, dense flow or slow moving traffic, the times are calculated by taking an average of two nearest values. And in case of congested flow, the missing values are copied from the nearest segment. The nearest segment can be before or after the current segment.

Calculate Total Travel time for the entire route

Scaled travel times (veh/km) for every segment of the route is known. Travel times are then unscaled to get the time required to drive through each segment in seconds. Time required to travel the entire route is calculated by aggregating the travel times of each segment. A graph of Travel times along ‘Route 12’ (northbound) from image 27-52 (Figure 9) of Cologne is shown in Figure 15 on the next page to state an example. Another graph of final estimated travel times along northbound route from the Munich database are shown in Figure 15. Due to the congestion on this section of the motorway, Figure 16 looks a lot different from Figure 15. Length of each segment along the route in metres is also illustrated beside the peak of each tower in these graphs.

Travel times are thus obtained for all the generated routes involving both datasets.

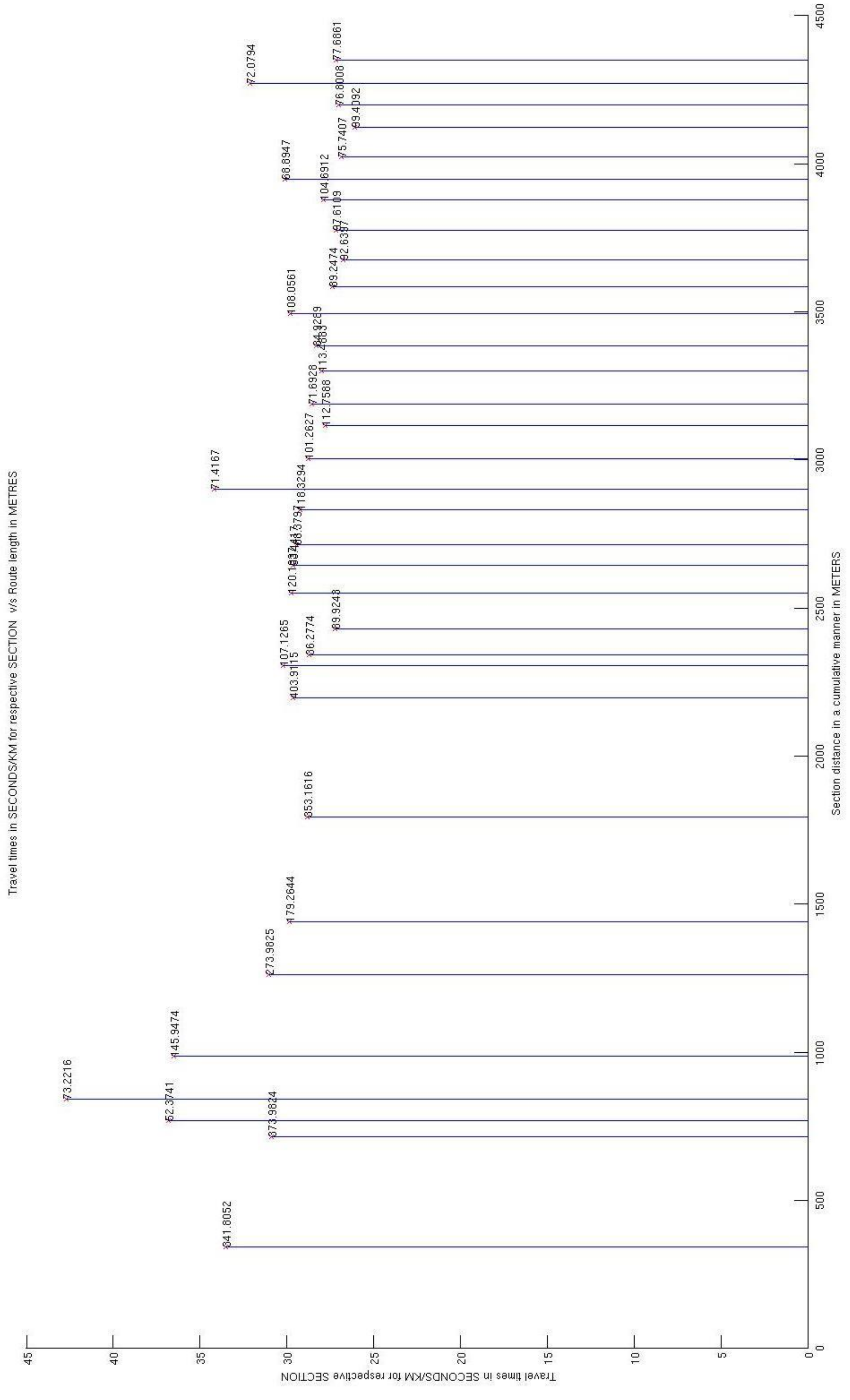

Figure 15: Scaled travel times (sec/km) along the route-Cologne

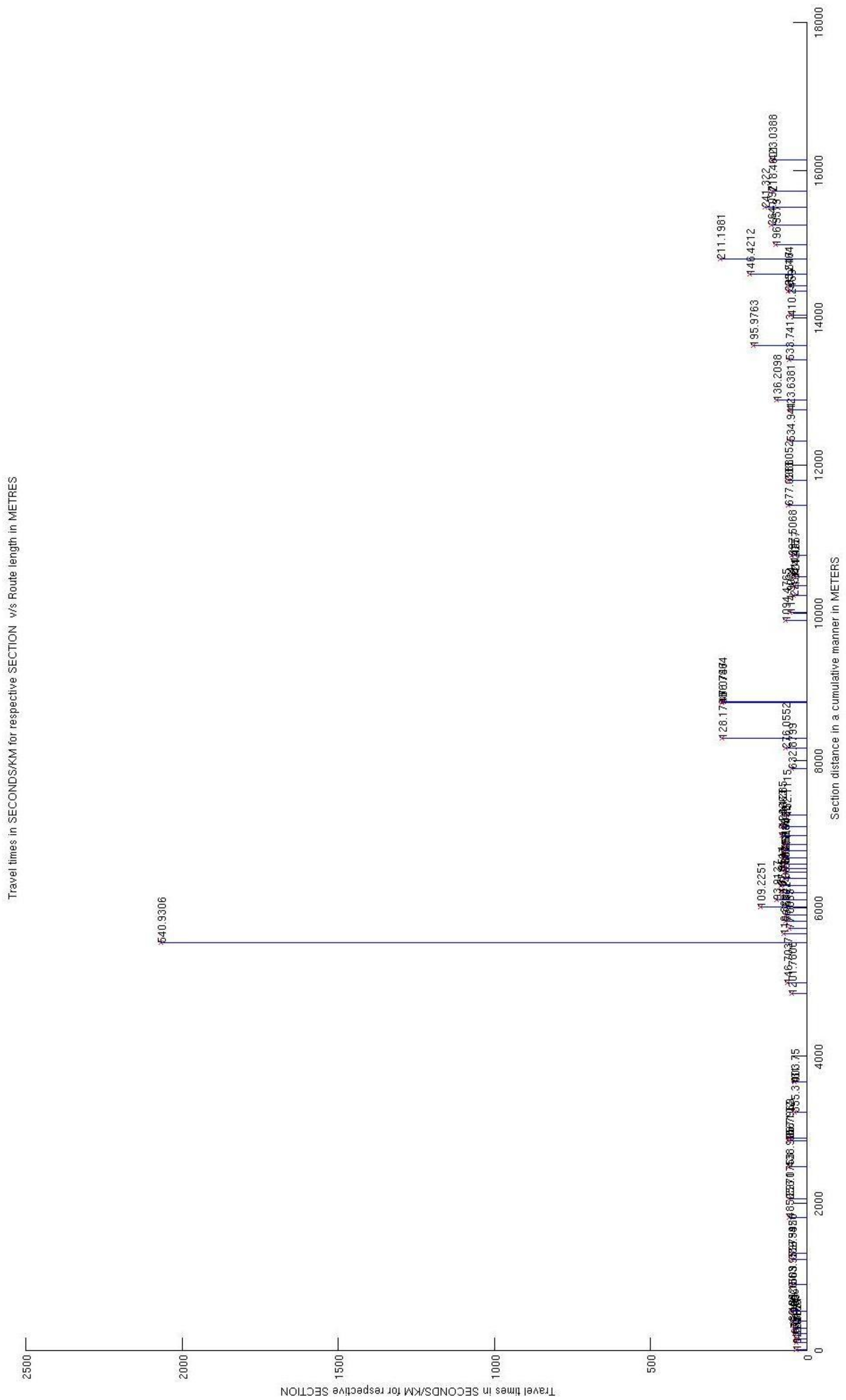

Figure 16: Scaled travel times (sec/km) along the route-Munich

4 ANALYSIS OF GENERATED RESULTS

4.1 Summary of the Results Obtained

With a developed algorithm to estimate travel times automatically on a motorway, results are needed to both validate and provide comparison with the acquired database to prove it's' accuracy. The final results were obtained for both the datasets. Two routes were chosen for comparison from each of the dataset for comparison: route 11 and route 12 (Figure 9) from Cologne; two manually aggregated routes from Munich, one northbound and one southbound along the entire flight path.

4.1.1 Cumulated Distance v/s Time Graphs

The state of traffic on the motorways can also be estimated by looking at these developed graphs which shows travel times along the entire route length. It gives a perception that a vehicle is tracked through the entire route length. The graphs for route 12, route 11 (Cologne); northbound route and southbound route (Munich) are shown in Figure 17, 18, 19 and 20 respectively.

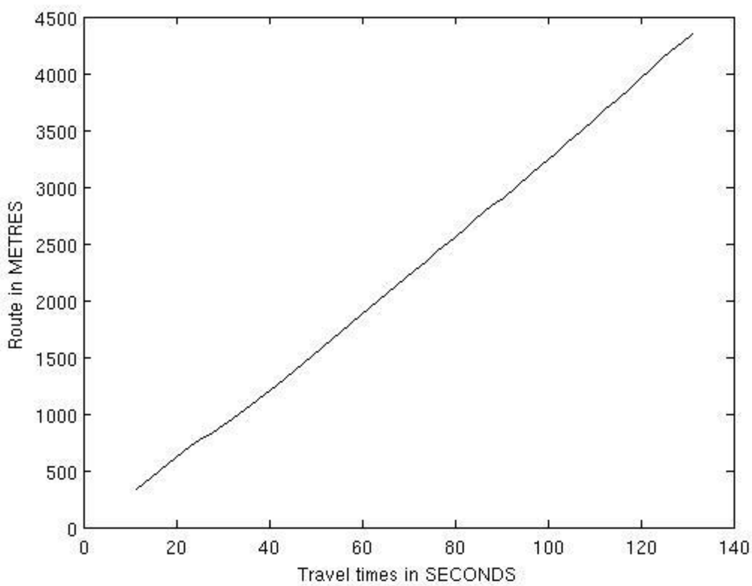

Figure 17: Route 12, northbound, Cologne

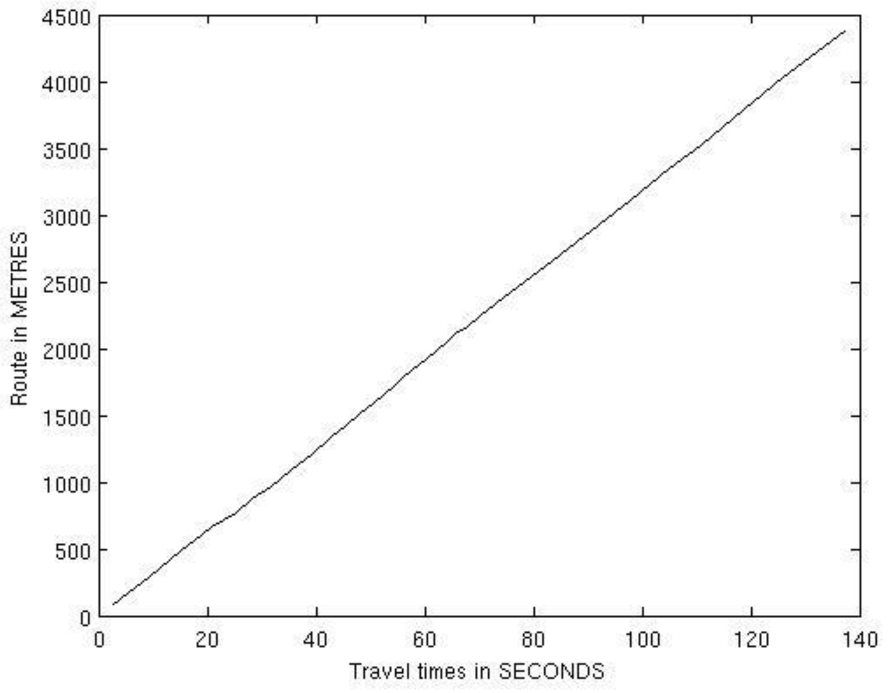

Figure 18: Route 11, southbound, Cologne

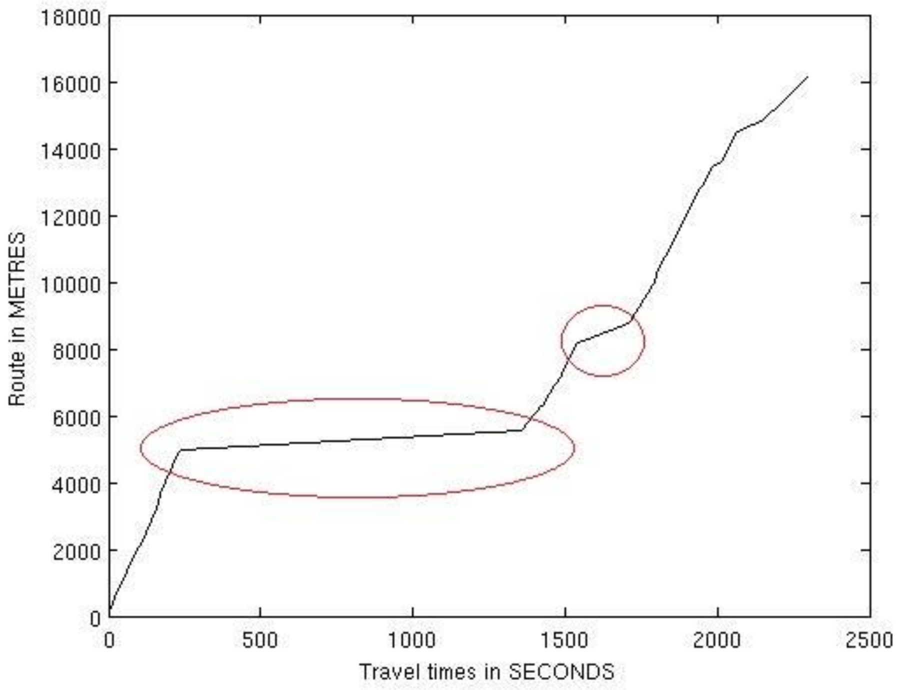

Figure 19: Northbound route, 16 km, Munich

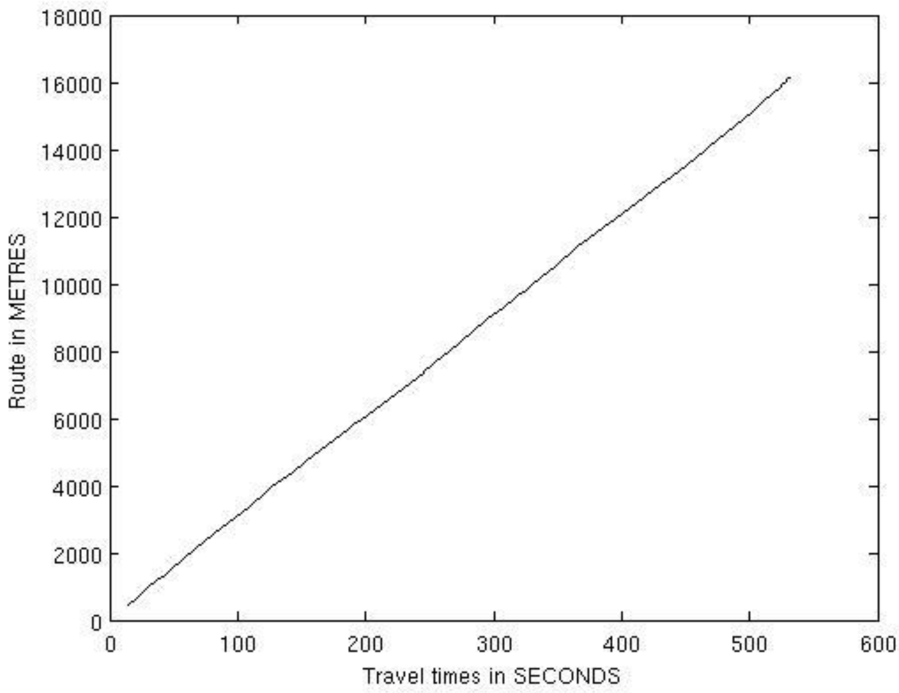

Figure 20: southbound Route, 16 km, Munich

Free flow was expected on the routes near Cologne whereas congestion was expected on northbound direction of the motorway near Munich because of the homebound travellers. The results prove it. Figure 19 highlights the section of the route experiencing halting traffic and long delays which can be used to identify the congested section on the route. And other figures show a smooth straight line depicting smooth traffic flow with low densities.

Travel times estimated by average momentary velocity and average local velocity produce same curve of the graphs. The difference in travel times generated by these two velocities is acknowledged later in this section in Table 3.

4.1.2 Densities and Velocities

Apart from travel times, the algorithm also estimates traffic density and average velocities per segment. But, it doesn't predict these values of missing segments. However, they can be useful in the future for creating a historical database or finding out trends in traffic parameters. A plot of scaled velocities and densities for a congested route and a route with free flowing traffic is shown in Figure 21 and 22 respectively. A diagram of V v/s D is also plotted for both the routes but without 'free velocity' and 'max density'.

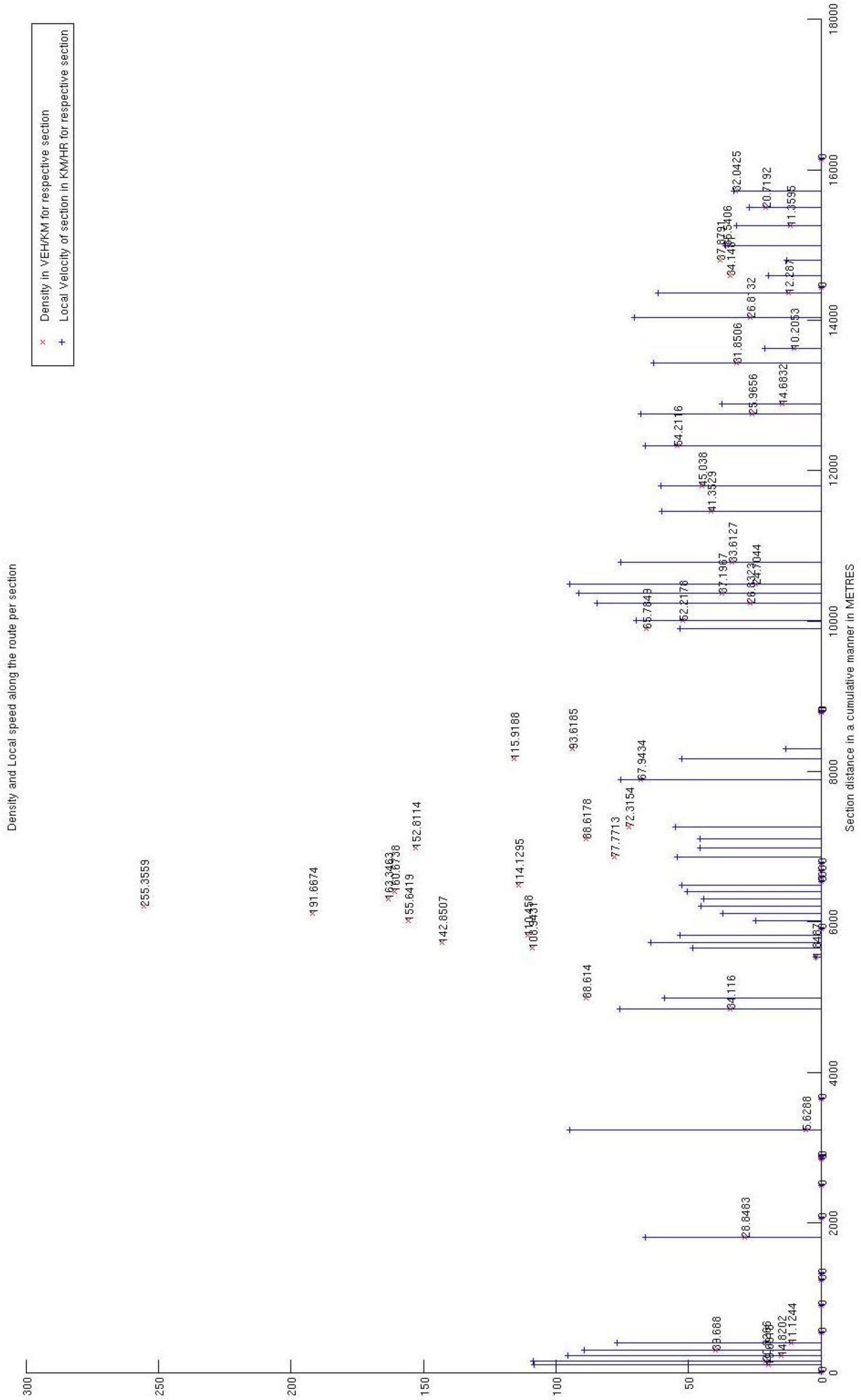

Figure 21: Density and Speeds along congested route, Munich

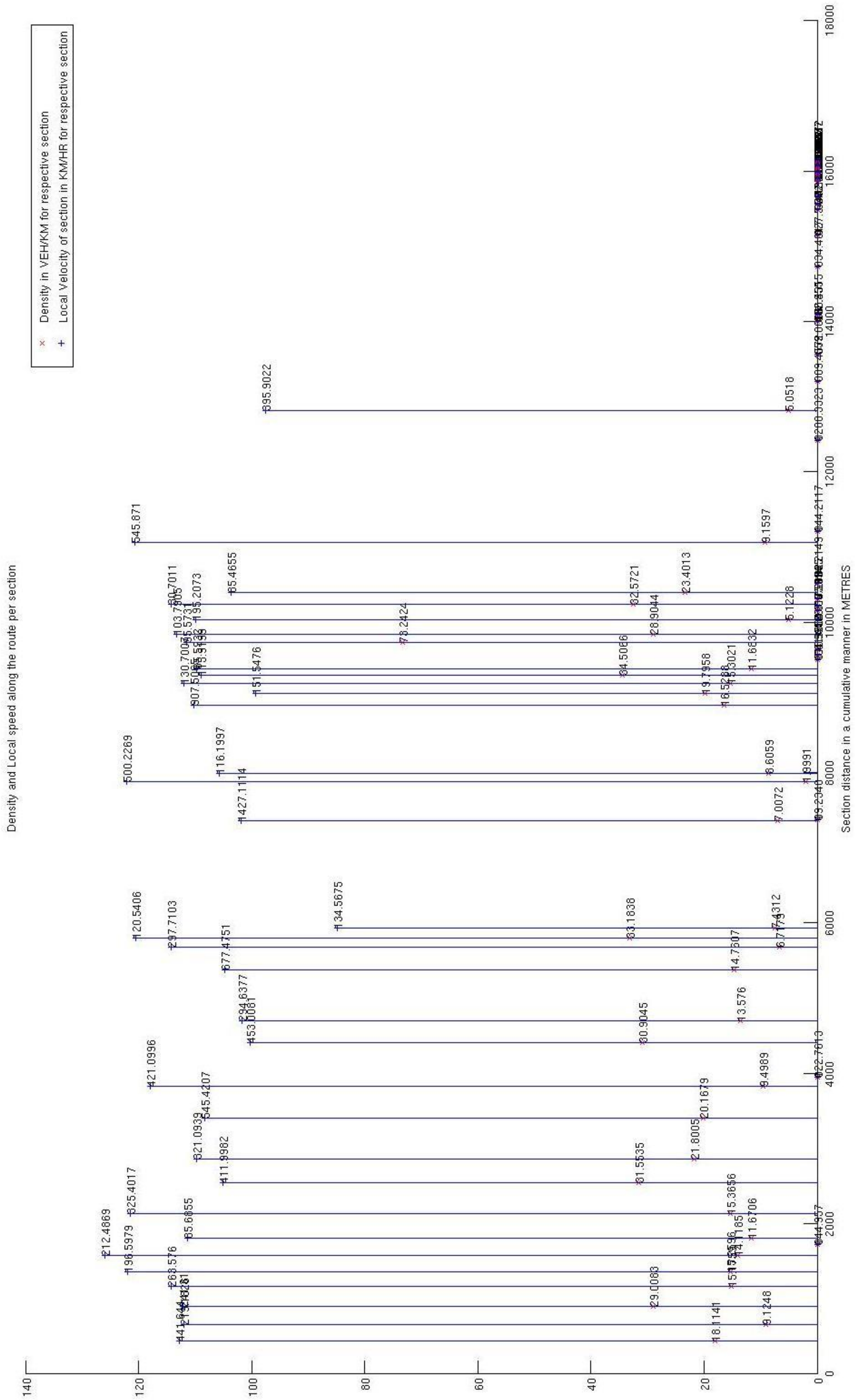

Figure 22: Density and Speeds along route with free flowing traffic, Munich

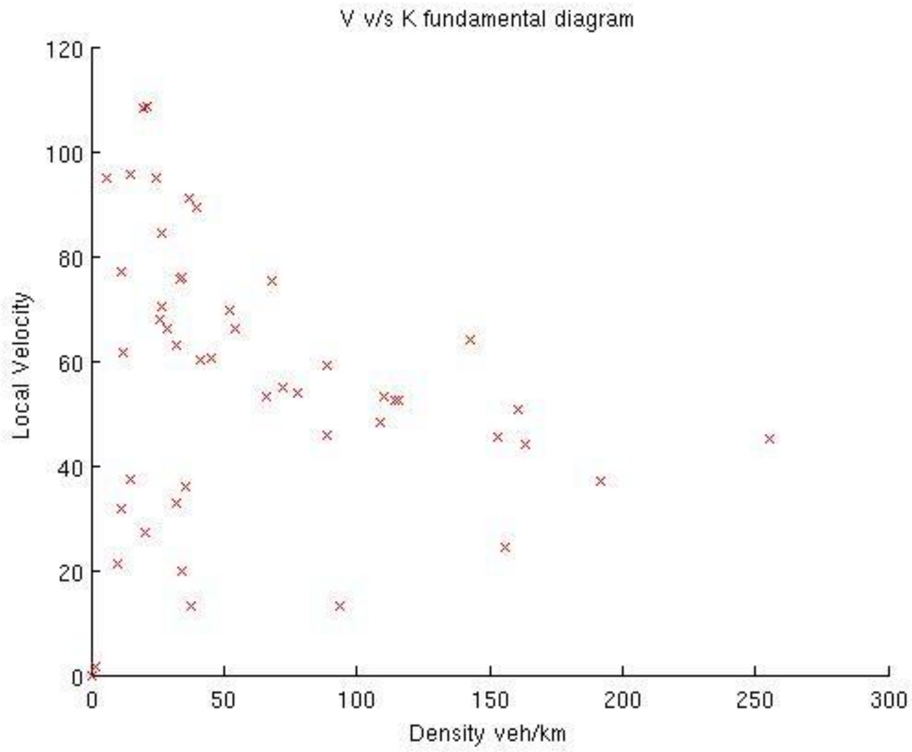

Figure 23: V v/s K Diagram of Traffic flow for Congested Route

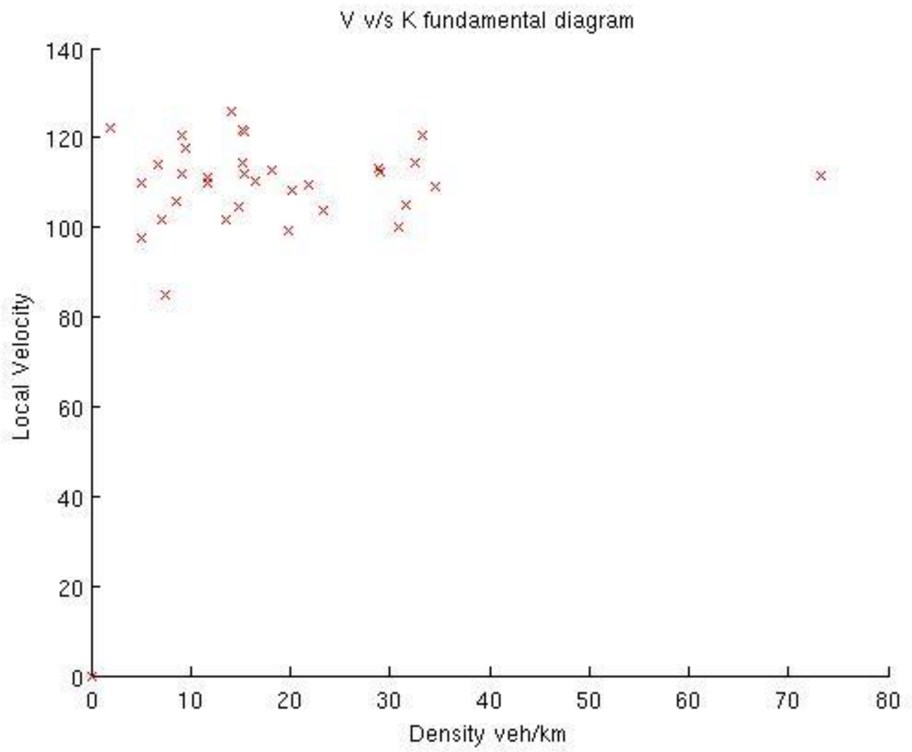

Figure 24: V v/s K Diagram of Traffic Flow for route with free flowing traffic

4.1.3 Summary of Important Results

A summary of the important outputs of the algorithm for comparison with results from other sensors and sources is depicted in Table 3 below. Travel times estimated by using average local as well as momentary velocities are acknowledged in the Table.

Database	Route	Traffic Flow	Route Length (m)	Total Travel time (sec)		
				Locally	Momentarily	Diff
Cologne	11, south	Free Flow	4371.7	137.39	138.86	1.47
	12, north	Free Flow	4349.7	130.97	132.02	1.05
Munich	Northbound	Congestion	16145	2297.7	2577.3	279.6
	Southbound	Free Flow	16148	531.58	535.35	3.77

Table 3: Summary of Results

The difference in estimated total travel time is high during congestion due to the fact that average local velocities differ a lot from average monetary velocities when there are a lot of outliers in the data. These results will be compared with the outputs from real traffic to find out which approach gives a better estimation (Table 4, 5 and 6).

4.2 Comparison of Travel Times

For validating and criticizing the output of an algorithm, a comparison with other means of measurement is required. And for the valid comparison between different means of measurement, the times for data acquisition should match. Thus the database acquired during these below mentioned timeslots are only considered. The Cologne dataset for images 27-52 (route 11 and 12) were acquired between 12:57 and 12:59 German time. And Munich dataset was acquired between 14:01 to 15:11 German time in three overflights. The dataset available for this thesis was acquired by the first overflight between 14:01 and 14:12.

The other sources and sensors for comparison include ground stationary devices like detectors; floating car data, a reference ADAC vehicle and traffic message channel. Database of all these different sources are analysed and unfortunately, it was not possible to compare outputs of the developed algorithm with all means of measurement due to the different acquisition times and/or incompleteness of the acquired information.

4.2.1 Ground Detectors (Inductive loops)

Data from detectors were collected to make an overall comparison of derived travel times and a database was created for comparison with both the datasets. Information extracted from the detectors near Cologne were not analysed whereas Dr. Kurz and his team (2007) had already analysed the information to derive travel times from detectors near Munich.

Cologne

An aerial overview of detectors spread out in Cologne is shown in Figure 25. The detectors which are lying on route 11 and route 12 are circled as they are important for comparison. Information is manually extracted from these particular detectors. Traffic flow (veh/h) and velocities (km/h) registered by these ground sensors are thus known. It is important to know detectors measure local velocities and thus the travel times derived from them should be compared with the travel times estimated by local velocities from algorithm source code.

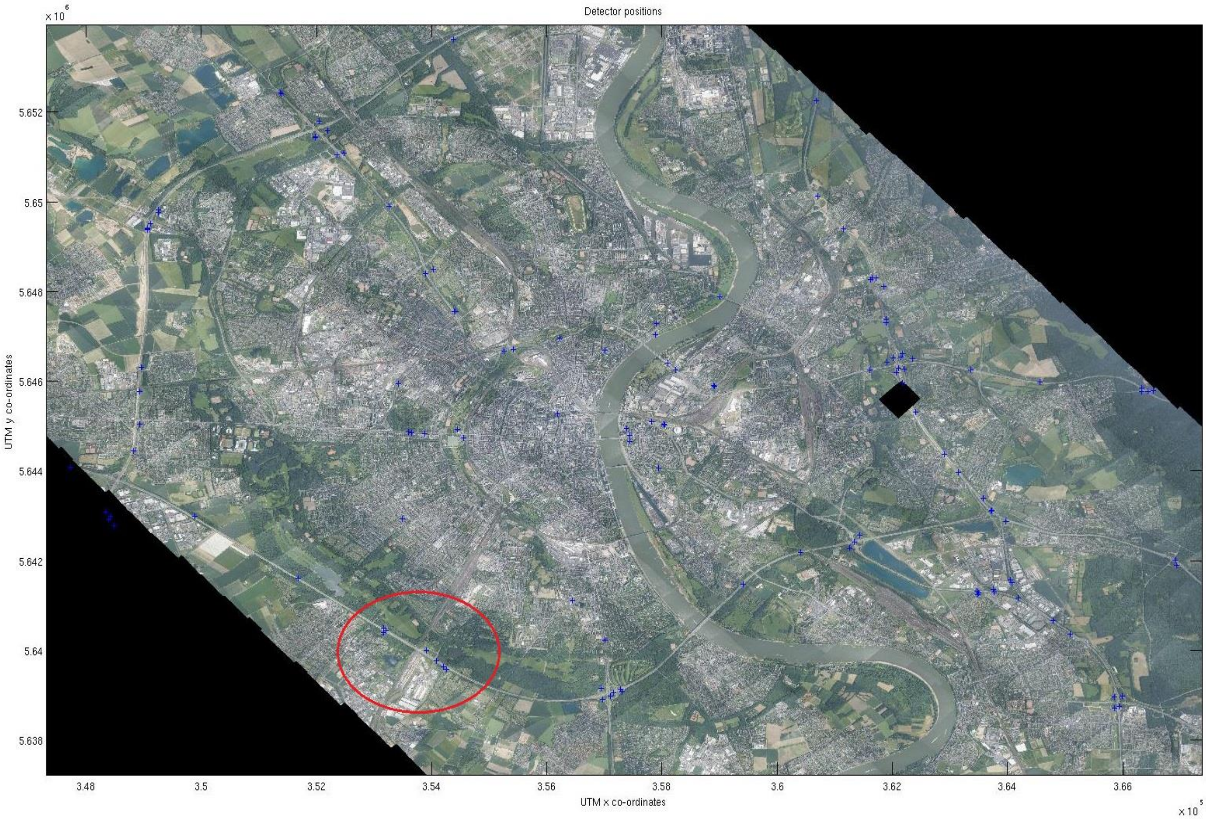

Figure 25: Aerial overview of Detectors in Cologne

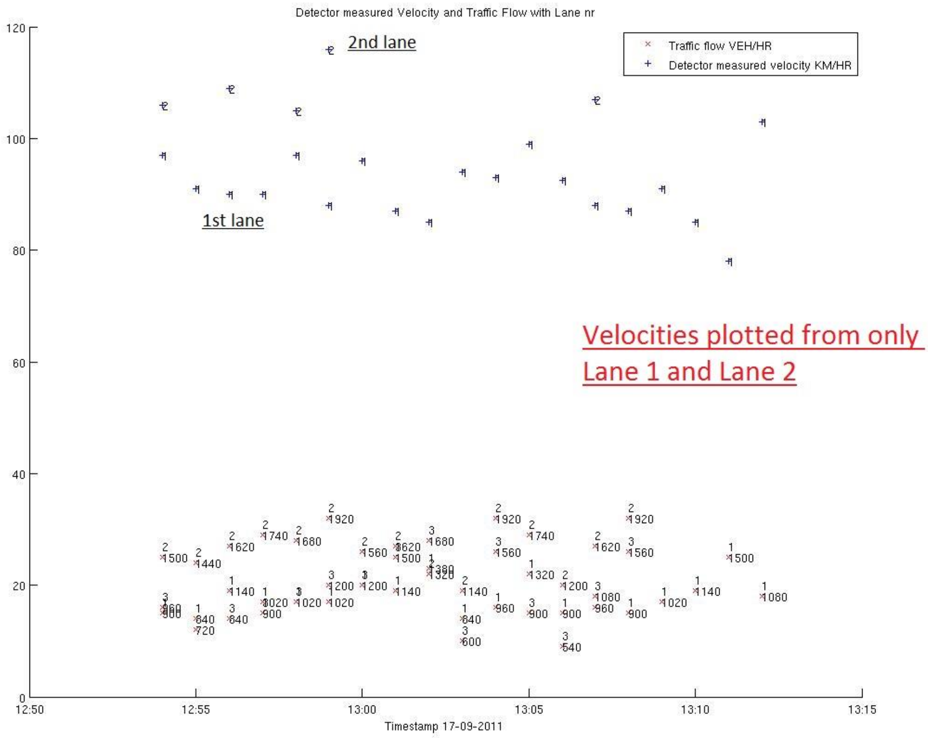

Figure 26: Recorded Velocities and Flow against Time, Route 12 Northbound

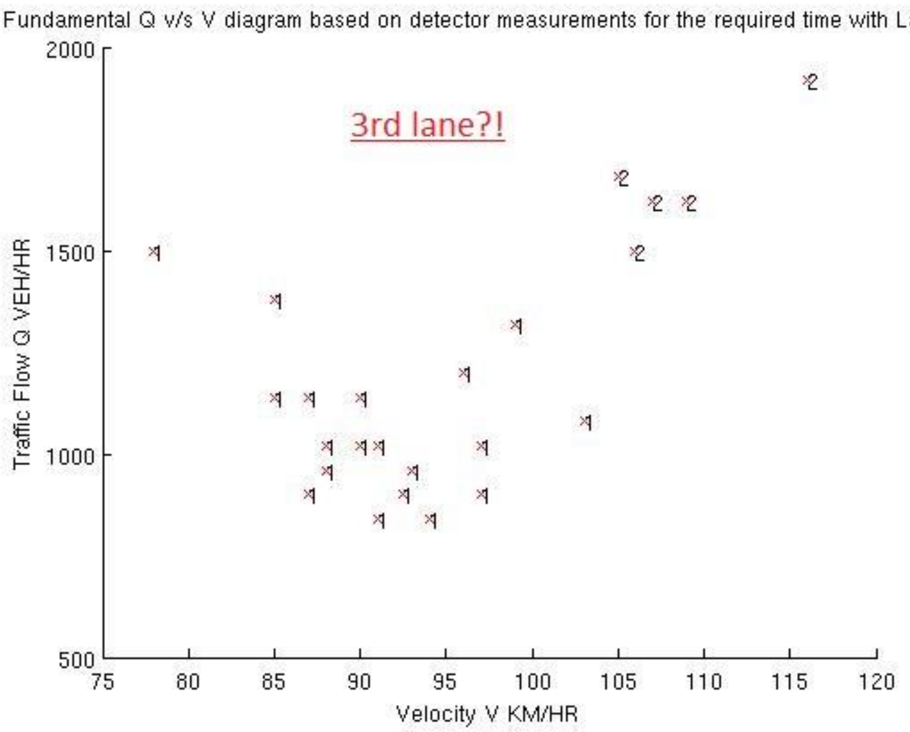

Figure 27: Qv/s V Diagram, Route 12 Northbound

But it was discovered later that detectors lying on route 11 acquired information every 5 minutes and recorded no values. Hence, detectors on route 12 are left for comparison. Traffic Flow and local velocities from these detector measurements are plotted against time in Figure 26. And a diagram of traffic flow against velocities is also illustrated in Figure 27. These graphs are plotted from the information acquired by detectors between 12:54 to 13:12 because they are to be compared with travel times of route 12, whose images were captured between 12:57 to 12:59. Lane numbers are illustrated in these figures for every value plotted.

By having a closer look at the information, it was deduced that values of local velocities for the third lane of the motorway are missing or not recorded by detectors. It was later derived that some values are also not recorded for the second lane of motorway between 12:54 to 13:12. Thus detector data of Cologne is unfit for accurate estimation as well as comparison of travel times.

However, scaled travel times (veh/km) are generated from detectors with the available data by developing a separate algorithm. It is done by dividing distance of unit 1km over the average of recorded local velocities (km/h). Another method is also used to generate travel times- creation of road segment (Section 3.2.1). Polygon is manually created around the detector to detect vehicles and calculate travel time in the same manner as done by the algorithm. This method doesn't involve the use of any data recorded by detectors and thus again proves unfit for accurate comparison. It is just performed for experimental purposes. .

These scaled average travel times (sec/km) from three different methods, each of them involving only local velocities, are then compared as shown in Table 4.

Average scaled travel times estimated by local velocities (sec/km)		
Detectors	**Segment around Detector**	**Image Data**
37.99	31.35	29.94

Table 4: Comparison of Travel Times with Detectors, Cologne, Route 12 Northbound

It can be seen in Table 4 that travel times generated from the detectors have higher values than travel times estimated from Images. This is due to the fact that local velocities recorded by detectors are acquired from the first and second lane and not from the third lane. First lane is the slowest and the third fastest. This difference between velocities of the first and second lane can also be seen in Figure 26 and

Figure 27. As higher values of velocities for the third lane are not recorded, estimated travel times for route 12 tend to be higher.

Munich

All the analysis and extraction of Information from the detectors has already been carried out at German Aerospace Centre (DLR) by Dr. Kurz and his team (2007). Instantaneous travel times were calculated from the data by the following equation:

$$TT(t) = \sum_{i=1}^{n} \frac{l_i}{v_i(t)} \quad (4.1)$$

Where, $TT(t) =$ Travel times for the whole stretch of 16 km at time instant t

$v_i =$ Speeds reported at the detector station i

$l_i =$ Length of the segment assigned to detector station i

Results are made available only for the northbound congested route in Munich and thus used in comparison of travel times as illustrated in Table 5. Traffic messages from the traffic channels contain information regarding congestion lengths which are also acknowledged in the table.

Time	**Traffic message**	**Travel times estimated by local velocities for 16 km strip (min)**	
		Detectors	**Image Data**
14:06	Halting Traffic 14 km	34’	37’57”
14:11	Congestion 7 km		

Table 5: Comparison of Travel Times with Detectors, Munich, Northbound Route

Instantaneous travel times derived from detector data do not completely comply with the reference measurements because speeds at only certain stations are used that are more or less a set of random observations of the complete motorway stretch. Also instantaneous speeds do not reflect the traffic dynamics (in this case: congestion) as they do not consider the actual time, a vehicle requires to pass different stations. (Kurz et al. 2007)

4.2.2 Floating Car Data (FCD)

Floating car data was acquired only in Cologne for comparison of travel times. An aerial overview of the points of acquisition as generated is shown in Figure 28.

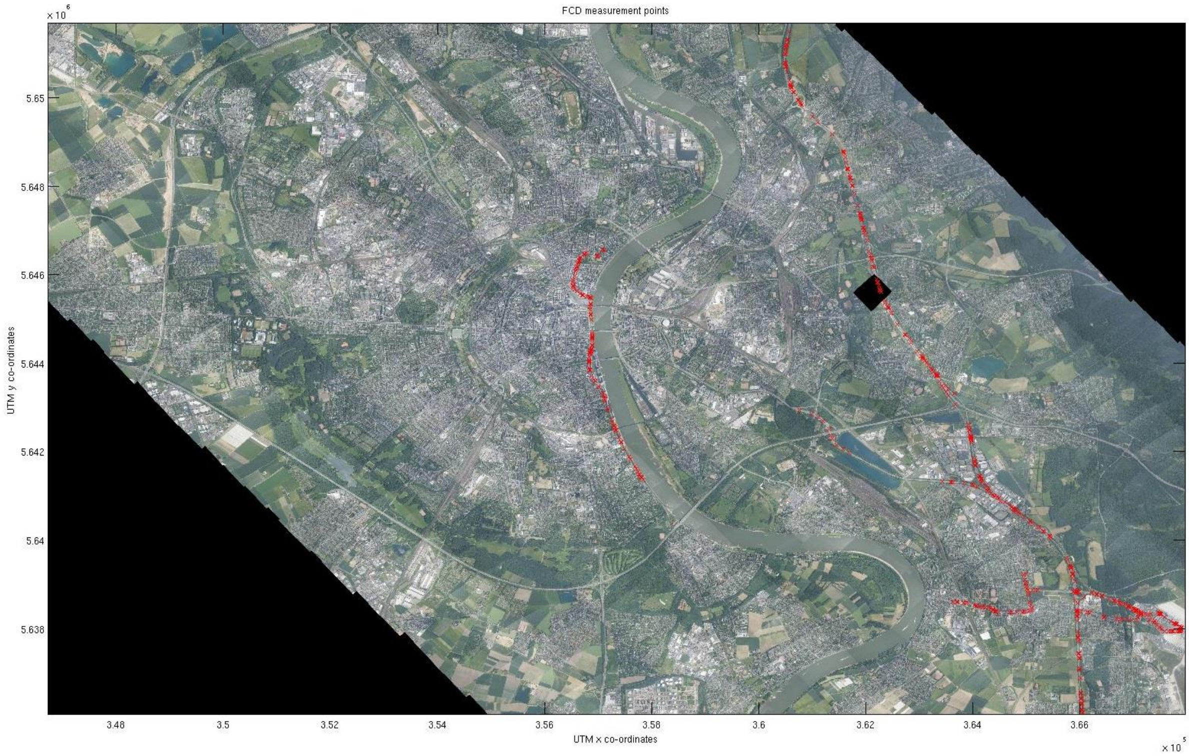

Figure 28: An aerial overview of FCD, Cologne

Information needed for particular routes and their timeslots were unfortunately not available entirely and thus it was not possible to compare travel times from FCD.

4.2.3 ADAC Reference Vehicle

For comparison with the ground truth of travel times, a GPS equipped ADAC reference vehicle conducted runs in northbound and southbound directions and the results were made available by Dr. Kurz (2007). The run started during the exact same time of the day when images were captured between Hofolding and Weyarn near Munich. These runs were not conducted in Cologne. A direct comparison of travel times is illustrated in Table 6.

Direction of Route	Detected Traffic Flow	Estimated Travel Times for 16 km strip (min)		
		Reference Vehicle	Algorithm	
			Locally	Momentarily
Southbound	Free Flow	9’	8’52”	8’55”
Northbound	Congested	35’	37’57”	42’34”

Table 6: Comparison of Travel times with Reference vehicle, Munich

A direct comparison between travel times from the reference vehicle and travel times estimated from images contain systematic errors, as the northbound run takes around half an hour from 14:02 to 14:36 whereas the images were captured in a time span of eleven minutes from 14:01 to 14:12.

5 CONCLUSION

7.1 Summary

The objective of this thesis is to accurately estimate travel times in near real-time situation, based on the traffic data extracted from aerial image time series or sequences.

First, an introduction into the thesis is given explaining its background and motivation. Adopted research approach is discussed and structure of the thesis is put forward giving an idea what to expect from every following chapter. To understand the database generation and Image acquisition process, an overview of the system hardware is given explaining the extraction of traffic data from aerial images. Then a description of various researches made in the field of calculating various traffic parameters from aerial images is presented. However, no concrete research is found regarding accurate estimation of travel times by aerial surveys which made it difficult to develop a concept for this thesis.

Using the information gained, preliminary analysis was performed on the database. Data format and image acquisition times are mentioned in order to understand the formation of algorithm in depth. Then the individual vehicle velocities are derived from known vehicle positions in two consecutive geocoded images by calculating the distance covered over elapsed time. In this context, the next step is to aggregate and consolidate the estimated traffic information. Aggregation is carried out by the creation of routes and road segments from the known centre line coordinates provided by NAVTEQ database. Traffic parameters like velocities are now accumulated in two ways locally and momentarily. Thus average local and momentary velocities are obtained for every segment which is later used in estimating travel times. However for some segments along the route, travel times are not estimated because when the images were captured, vehicles were not present in those segments. The travel times for these particular segments are predicted based on the varying traffic dynamics.

Before comparing these travel times with results from other sources and sensors, a summary of obtained results is presented. Some of the comparisons were not possible due to the incomplete database acquired from other means of measurement. To end the analysis, reasons for difference in the values of results are given with every comparison.

7.2 Conclusions

A few significant conclusions can be drawn by commenting on the overall accuracy of results and functioning of the entire aircraft monitoring system.

The whole processing chain of the system is performed in real-time on board the aircraft and on the ground. The developed algorithm to calculate travel times performs analysis in near real-time as well. This ensures that the results are extremely actual. The developed source code performs calculations for each particular route and all images. Therefore it gives the flexibility to choose a particular route for analysis and developing results.

The overall accuracy of the travel times depends upon the completeness, correctness and quality of the vehicle detection and tracking. Even though automatic detection and tracking algorithms are outside the scope of this thesis, it is important to mention them as they affect the results. The detection quality of the cologne dataset was calculated at 78%. And the completeness of the tracking and detection was calculated at 92% to 97% (which means that 92% to 97% of the vehicles detected are tracked properly) and 90% respectively by Leitloff et al. at DLR (2013). The completeness of this entire system is 68% for Cologne dataset. The system completeness for measuring velocities from induction loops is between 81% and 94%. In Nihan et al. (2002), it is however stated that overestimation occurs occasionally with the dual loop sensors and therefore leads to dropping the system correctness to as low as 53%. Hence the quality of extracted traffic information touches the quality range of induction loops with the advantage that this system provides a higher spatial coverage (Leitloff et al. 2013).

Statiscal analysis generally involves less parameters and thus results in less data error but more specification error. However in this prediction of travel times, statiscal analysis is performed considering the character of free flow. But for the congested conditions, another models could be used to validate results in the process of obtaining better accuracy.

The created road segments detect most of the vehicles from images. This conclusion was achieved by comparing geometrically tracked vehicles northbound or southbound (by their driving directions) and vehicles detected from the created routes by the ‘inpolygon’ function in MATLAB.

The investigations and comparison show that estimation of travel times from aerial image time series is quite accurate. However, a comparison of travel times estimated from image data and from stationary ground detectors such as induction loops is challenging due to the different species of data obtained. And a comparison with a reference vehicle also contains systematic errors because of the different times of acquisition. Another method of comparison against a real reference database, like ANPR (automatic number plate recognition) can also be thought upon in the future to further check the accuracy of produced results.

The results generated by this thesis are still in the evaluation phase but can be made available to the security and emergency organisations to follow the current traffic situation and derive logistics of the desired area.

The further research can focus on predicting travel times more accurately. Use of fuzzy logic can be thought upon in the case of congested traffic flows. However the research is also being done to produce fast and efficient algorithms for the accurate detection and tracking of vehicles.

LIST OF REFERENCES

'A Bird? A Plane? No, It's a Traffic Camera', ITS World 1997, p. 54

Busch, F., Glas, F., Bermann, E., 2004. 'Dispositionssysteme als FCD-Quellen für eine verbesserte Verkehrslagerkonstruktion in Städten – eine überblick', Straßen Verkehrstechnick 09/04

Busch, F., 2011. 'Empirical Data, probability distributions', Lecture notes of Traffic Flow, Technische Universität München, pp. 61-63

Ernst, I., Hetscher, M., Zuev, S., Thiessenhusen, K.-U., Ruhe, M., 2005. 'New approaches for real time traffic data acquisition with airborne systems', In: TRB Transportation Research Board, TRB 2005, Washington

Frassl, M., Lichtenstern, M., Khider, M., Angermann, M., 2010. 'Developing a system for information management in disaster relief methodology and requirements abstract'

Gellert M., Kurz, F., Rosenbaum, D., Meynberg, O., 2013. 'A real-time optical airborne road traffic monitoring system', KEPAF, pp. 645-656

German Aerospace Centre (DLR), 2013. Institute of remote Sensing, Photogrammetry and Image analysis, viewed 1st November 2013, <http://vabene.dlr.de/vabene/3K.htm>

Haralick, R.M., Shanmugan, K., Dinstein, I., 1973. 'Textural features for image classification', IEEE Transactions on Systems, man and Cybernetics 3(6), pp. 610-621

Jordan, T.D., 1965. 'Sky count of Traffic congestion and Demand', In: Traffic Engineering & Control, Vol. 7, No. 5, pp. 312-315

Kerner, B., 2004. 'The Physics of Traffic Empirical Freeway Pattern Features', Engineering applications and Theory, Springer, Berlin

Kurz, F., Charmette, B., Suri, S., Rosenbaum, D., Spangler. M., Leonhardt, A., Bachleitner, M., Stätter, R., Reinartz, P., 2007. 'Automatic traffic monitoring with an airborne wide-angle digital camera system for estimation of travel times', Commission WG III/5

Leitloff, J., Rosenbaum, D., Kurz, F., Meynberg, O., Reinartz, P., 2013. ‘An Operation for Estimating Road Traffic Information from Aerial Images’.

Littleton, B., 1996. ‘Observation Balloon system’, In: National Traffic Data Acquisition Conference (NATDAC ’96), Vol. 1, NM-NATDAC-96, Alliance for Transportation Research, Alburquerque, New Mexico, p. 492

MATLAB version 8.0, 2012b, computer software. The MathWorks Inc., Natick, Massachusetts

MARZ, 1999. ‘Merkblatt für die Ausstattung von Verkehrsrechnerzentralen und Unterzentralen’, Forschungsgesellschaft für Straßen- und Verkehrswesen (FGSV)

Merry, C.J., McCord, M.R., Jafar, Perez, L.A., 1996. ‘Feasibility of Using Simulated Satellite Data Coordinated with Traffic Ground Counts’, , In: National Traffic Data Acquisition Conference (NATDAC ’96), Vol. 1, NM-NATDAC-96, Alliance for Transportation Research, Alburquerque, New Mexico, pp. 494-497

Nihan, N.L., Zhang, X., Wang, Y., 2002. ‘Evaluation of dual-loop data accuracy using video ground truth table: Dual loop error’, Tech. rep., National Technical Information Service , 5285 Port Royal Road, Springfield, VA 22161 USA

Rosenbaum, D., Leitloff, J., Kurz, F., Meynberg, O., Reize, T., 2010. ‘Real-time Image Processing for Road Traffic Data Extraction from Aerial Images’, Commission VII Symposium

Schaefer, R.-P., Thiessenhusen, K.-U., Wagner, P., 2002. ‘A traffic information system by means of real-time floating-car data’, Proceedings of ITS World Congress, October 2002, Chicago, USA

Skycomp, Inc., 1995. ‘Traffic Quality on the OKI Regional Highway System: Eastern corridor Sub-Region’, Ohio-Kentucky-Indiana Regional council of Governments

Steger, C., Ulrich, M., Wiedemann, C., 2008. ‘Machine vision algorithms and applications’, WILEY-VCH Verlag GmbH & Co. KGaA

‘Traffic Surveillance Drone to be Tested’, Aviation Week & Space Technology, March 10, 1997, p. 68

Williams, B., 1997. ‘Existing and emerging techniques for the Automatic collection of Data for Transport Modelling’, Smith System Engineering, United Kingdom

<https://maps.google.de/>, 2013.

LIST OF FIGURES

LIST OF TABLES

APPENDIX: ALGORITHM SOURCE CODE

This appendix contains the source code for estimation of travel times for route 46 (Section 3.2.2 figure 10) from aerial image data (Images 81-107, figure 5). The algorithm is written using MATLAB. It doesn't contain the algorithms written for producing results from detector database used for the comparison of travel times. It also excludes the code for generating graphs and figures.

```
close all
clear all
clc

% Only the important routes for comparison
% All the 1st image out of 3 images of the burst
image_nr=[81,84,87,90,93,96,99,102,105];

for r=1:length(image_nr)
    image_nr1=num2str(image_nr(r));

    if image_nr(r)<100
        im_aux=['/users/shah_mi/la/Data-
cologne/ON00',image_nr1,'/_aux'];
        locationfile=['/users/shah_mi/la/Data-
cologne/ON00',image_nr1,'/car_detected'];
        velocityfile=['/users/shah_mi/la/Data-
cologne/_cpdb1_ON00',image_nr1,'_1'];
    else
        im_aux=['/users/shah_mi/la/Data-
cologne/ON0',image_nr1,'/_aux'];
        locationfile=['/users/shah_mi/la/Data-
cologne/ON0',image_nr1,'/car_detected'];
        velocityfile=['/users/shah_mi/la/Data-
cologne/_cpdb1_ON0',image_nr1,'_1'];
    end

    % Reading data from car_detected file of vehicle detection
    locationraw=fileread(locationfile);
    locationdata=textscan(locationraw,'%d %f %f %f %f %f %f %f %f');

    location = locationdata{1};
    veh_detected(r)=length(location'); %Detecetd Number of cars per
first image of the burst from car_dec

    % Reading Data drom cpdb files for veh tracking
    velocityraw=fileread(velocityfile);
    velocitydata(:,r)=textscan(velocityraw,'%s %s %f %s %f %s %f %s %f
%c %s %f %s %f %s %f %s %f %s %f %s %f %*[^\n]');

    % Number of rows
    corr1=velocitydata{2,r};
```

```
    corr_length(r)=length(corr1');

    % OBject amount (cars tracked) mentioned already in FILES
    veh_tra{r}=velocitydata{2,r}{corr_length(r)-3}; % Object amount
from cpdb tracking file
    true_veh_tracked(r)=str2num(veh_tra{r}); % Tracked Number of cars
from cpdb files for tracking

    t1{r}=velocitydata{2,r}{corr_length(r)-2}; % GPS timestamp1_CPDB
    ti(r)=str2num(t1{r}); % Time stamp of the 1st image in cpdb file
    t2{r}=velocitydata{2,r}{corr_length(r)-1}; % GPS timestamp2_CPDB
    tf(r)=str2num(t2{r}); % Time stamp of the 2nd image in cpbb file
    tdiff(r)=tf(r)-ti(r);

    auxraw=fileread(im_aux);
    auxdata(:,r)=textscan(auxraw,'%s %s %*[^\n]');
    corr2=auxdata{2,r};
    corr_length2(r)=length(corr2');

    % Reading the timestamp of the image from aux file
    timenotei{r}=auxdata{2,r}{corr_length2(r)-14};
    timenote(r)=str2num(timenotei{r});

    %timenotei(81)=558687.579 = 2011-09-17 11:11:27 GMT
    %timenotei(107)=558744.975 = 2011-09-17 11:12:25 GMT
    %So the detector and FCD data to be comapred should be between
these
    %times to get an accurate idea

    utmx1(1:corr_length(r),r)=velocitydata{12,r}; % x co-ord of a
particluar veh in 1st image
    utmy1(1:corr_length(r),r)=velocitydata{14,r}; % y co-ord of a
particular veh in 1st image
    utmx2(1:corr_length(r),r)=velocitydata{16,r}; % x co-ord of the
same veh in 2nd image
    utmy2(1:corr_length(r),r)=velocitydata{18,r}; % y co-ord of the
same veh in 2nd image

    for i=1:true_veh_tracked(r)

        velocity(i,r)=((utmx1(i,r)-utmx2(i,r))^2+(utmy1(i,r)-
utmy2(i,r))^2 )^0.5; % Metre/sec
        velocity(i,r)=velocity(i,r)/tdiff(r)*3.6;  % Km/hr

    end

end

% Consolidation
% Finding the better accurate data to get better accurate results

for r=1:length(image_nr)
    g(r)=percenctile(velocity(:,r),0.95);
end

velocity(velocity==0) = NaN;
```

```
for r=1:length(image_nr)
    f(r)=percenctile(velocity(:,r),0.05);
end

for r=1:length(image_nr)

    findices=find(velocity(:,r)<f(r));
    velocity(findices,r)=NaN; % To make the matrix intact and accuate
results
    utmx1(findices,r)=NaN; % so that those vehicles are also not
detected in inpolygon
    utmy1(findices,r)=NaN;
    utmx2(findices,r)=NaN;
    utmy2(findices,r)=NaN;

    gindices=find(velocity(:,r)>g(r));
    velocity(gindices,r)=NaN;
    utmx1(gindices,r)=NaN;
    utmy1(gindices,r)=NaN;
    utmx2(gindices,r)=NaN;
    utmy2(gindices,r)=NaN;
end


% Reading files

routefile46=['/users/shah_mi/MATLAB/R2012b/bin/Route46.txt'];

routeraw=fileread(routefile46);
routedata=textscan(routeraw,'%d %d %d %f %f');


navteq_id=routedata{1};
node_det=length(navteq_id'); %Nodes detected in route 46

node_utmx(1:node_det)=routedata{4}; % X-coordinate for nodes in route
46
node_utmy(1:node_det)=routedata{5}; % Y-coordinate for nodes in route
46
no_of_lanes(1:node_det)=routedata{3}; % To get sigma and sketch a
perfect Section depending on the number of lanes


% Creating sections or segments(rectangle) based on the formula
for k=1:node_det-1 % nodes detected-1 sections rectanglular

    deltax(k)=node_utmx(k)-node_utmx(k+1);
    deltay(k)=node_utmy(k)-node_utmy(k+1);
    d(k)=(deltax(k)^2+deltay(k)^2)^0.5; %distance or length of the
section for travel times in METRES
    width(k)=no_of_lanes(k)*3.7;

    X(k,1)=((-1*deltay(k)/d(k))*(width(k)/2))+node_utmx(k);
    Y(k,1)=((deltax(k)/d(k))*(width(k)/2))+node_utmy(k);
    X(k,2)=((-1*deltay(k)/d(k))*(-1*width(k)/2))+node_utmx(k);
```

```
    Y(k,2)=((deltax(k)/d(k))*(-1*width(k)/2))+node_utmy(k);

    X(k,3)=((-1*deltay(k)/d(k))*(width(k)/2))+node_utmx(k+1);
    Y(k,3)=((deltax(k)/d(k))*(width(k)/2))+node_utmy(k+1);
    X(k,4)=((-1*deltay(k)/d(k))*(-1*width(k)/2))+node_utmx(k+1);
    Y(k,4)=((deltax(k)/d(k))*(-1*width(k)/2))+node_utmy(k+1);

    % phi(k)=atand(deltax_n(k)/deltay_n(k));
    % sigma(k)=no_of_lanes_n(k)*1.9;

    % X_n(k,1)=(sigma(k)*cosd(phi(k)+90))+node_utmx_n(k); % X coord of
the 1st corner of rectangle
    % Y_n(k,1)=(sigma(k)*sind(phi(k)+90))+node_utmy_n(k); % Y coord of
the 1st corner of rectangle
    % X_n(k,2)=(sigma(k)*cosd(phi(k)-90))+node_utmx_n(k); % X coord of
the 2nd corner of rectangle
    % Y_n(k,2)=(sigma(k)*sind(phi(k)-90))+node_utmy_n(k); % Y coord of
the 2nd corner of rectangle

    % X_n(k,3)=(sigma(k)*cosd(phi(k)+90))+node_utmx_n(k+1); % X coord
of the 3rd corner of rectangle
    % Y_n(k,3)=(sigma(k)*sind(phi(k)+90))+node_utmy_n(k+1); % Y coord
of the 3rd corner of rectangle
    % X_n(k,4)=(sigma(k)*cosd(phi(k)-90))+node_utmx_n(k+1); % X coord
of the 4th corner of rectangle
    % Y_n(k,4)=(sigma(k)*sind(phi(k)-90))+node_utmy_n(k+1); % Y coord
of the 4th corner of rectangle



section(k,:)=[X(k,1),X(k,2),X(k,4),X(k,3),X(k,1),Y(k,1),Y(k,2),Y(k,4),Y
(k,3),Y(k,1)];
    % ARRANGING POINTS IN CLOCKWISE DIRECTION TO FORM A RECTANGLE
PERFORM INPOLYGON
end


% Performing inpolygon: gives 1 if veh detected and 0 if not
image_nr=[81,84,87,90,93,96,99,102,105];
for k=1:node_det-1

    for r=1:length(image_nr) %only the important images for comparison
        xv=[X(k,1) X(k,2) X(k,4) X(k,3) X(k,1)];
        yv=[Y(k,1) Y(k,2) Y(k,4) Y(k,3) Y(k,1)];
        x=utmx1(1:corr_length(r),r);
        y=utmy1(1:corr_length(r),r);

        for p=1:length(x)-4
            zin(p,r,k)=inpolygon(x(p),y(p),double(xv),double(yv));
        end
    end
end

% Initializing the variables
a=size(zin);
vehspeed(a(1),a(2),a(3))=0;
```

```
%  Veh Density and Veh speed classification for the detected veh
through
% inpolygon
for k=1:node_det-1 % k also for sections or segments
    vehnum(k)=0;
    for r=1:length(image_nr) %only the important images for comparison
         x=utmx1(1:corr_length(r),r);
         vehnum_img(r,k)=0;
        for p=1:length(x)-4 % -4 because of NaN values
            if zin(p,r,k)==1
                vehnum_img(r,k)=vehnum_img(r,k)+1; % density
veh/image/section
                vehnum(k)=vehnum(k)+1; % density veh/section
                vehspeed(p,r,k)=velocity(p,r);
            end
        end
    end
end

% Cars tracked by coding or innpolygoN
% TO COMPARE WITH CARS TRACKED GEROMETRICALLY ABOVE
for r=1:length(image_nr)
    calc_veh_route(r)=sum(vehnum_img(r,:));
end

% local and Momentary velocities per image and section

for k=1:node_det-1
    for r=1:length(image_nr)
        localvelo_img(r,k)=0;
        x=utmx1(1:corr_length(r),r);
        for p=1:length(x)-4

localvelo_img(r,k)=localvelo_img(r,k)+((vehspeed(p,r,k)*vehspeed(p,r,k)
)/sum(vehspeed(:,r,k)));

momentaryvelo_img(r,k)=sum(vehspeed(:,r,k))/vehnum_img(r,k);
        end
    end
end

% converting NaN values to 0 for calculating velocities per section
localvelo_img(isnan(localvelo_img))=0;
momentaryvelo_img(isnan(momentaryvelo_img))=0;

%  Divisor for momentary velocity per section
for k=1:node_det-1
   divisor(k)=0;
    for r=1:length(image_nr)
        if momentaryvelo_img(r,k)~=0
            divisor(k)=divisor(k)+1; % Divisor for momentary velocity
        else
            divisor(k)=divisor(k);
        end
    end
```

```
end

% Final Local and Momentary veloctites per section

for k=1:node_det-1
    localvelo(k)=0;
    for r=1:length(image_nr)

localvelo(k)=localvelo(k)+((localvelo_img(r,k)*localvelo_img(r,k))/sum(
localvelo_img(:,k)));
        momentaryvelo(k)=sum(momentaryvelo_img(:,k))/divisor(k);
    end
end


% Travel times per section in seconds
% Velocities in metres/sec
% Length of section in km

for k=1:node_det-1

    momentaryvelo_mpers(k)=1/3.6*momentaryvelo(k);

    localvelo_mpers(k)=1/3.6*localvelo(k); % Velocities in METRES/SEC

    traveltime(k)=d(k)/localvelo_mpers(k); % Travel times per section
in SECONDS

    d_km(k)=d(k)/1000; % Length of section in KM

    tt_km_sec(k)=traveltime(k)/d_km(k); % Times in SECONDS/KM for
respective section

    vehnum_km(k)=vehnum(k)/d_km(k); % Number of VEHICLES/KM for
respective section
end

% Converting NaN values to 0 for calculation
localvelo(isnan(localvelo))=0;
momentaryvelo(isnan(momentaryvelo))=0;


% MARZ conditions for classification
for j=1:k

    if no_of_lanes(j)==3
        if all([localvelo(j)>=80, vehnum_km(j)<=40, vehnum_km(j)>=0])
            flow(j)=1; % free traffic
        elseif all([localvelo(j)>=80, vehnum_km(j)<=70,
vehnum_km(j)>40])
            flow(j)=2; % dense traffic
        elseif all([localvelo(j)<80, localvelo(j)>=30,
vehnum_km(j)<=70])
            flow(j)=3; % slow moving traffic
        elseif all([localvelo(j)==0, vehnum_km(j)==0])
            flow(j)=5; % Not known
```

```
        elseif all([any([localvelo(j)<30, vehnum_km(j)>70]), flow(j-
1)==2])
            flow(j)=3;
        elseif any([localvelo(j)<30, vehnum_km(j)>70])
            flow(j)=4; % congestion
        else flow(j)=5; % Not known
        end

    elseif no_of_lanes(j)==2
        if all([localvelo(j)>=80, vehnum_km(j)<=30, vehnum_km(j)>=0])
            flow(j)=1; % free traffic
        elseif all([localvelo(j)>=80, vehnum_km(j)<=60,
vehnum_km(j)>30])
            flow(j)=2; % dense traffic
        elseif all([localvelo(j)<80, localvelo(j)>=30,
vehnum_km(j)<=60])
            flow(j)=3; % slow moving traffic
        elseif all([localvelo(j)==0, vehnum_km(j)==0])
            flow(j)=5; % Not known
        elseif all([any([localvelo(j)<30, vehnum_km(j)>60]), flow(j-
1)==2])
            flow(j)=3;
        elseif any([localvelo(j)<30, vehnum_km(j)>60])
            flow(j)=4; % congestion
        else flow(j)=5; % Not known
        end

    elseif no_of_lanes(j)==4
        if all([localvelo(j)>=80, vehnum_km(j)<=50, vehnum_km(j)>=0])
            flow(j)=1; % free traffic
        elseif all([localvelo(j)>=80, vehnum_km(j)<=80,
vehnum_km(j)>50])
            flow(j)=2; % dense traffic
        elseif all([localvelo(j)<80, localvelo(j)>=30,
vehnum_km(j)<=80])
            flow(j)=3; % slow moving traffic
        elseif all([localvelo(j)==0, vehnum_km(j)==0])
            flow(j)=5; % Not known
        elseif all([any([localvelo(j)<30, vehnum_km(j)>80]), flow(j-
1)==2])
            flow(j)=3;
        elseif any([localvelo(j)<30, vehnum_km(j)>80])
            flow(j)=4; % congestion
        else flow(j)=5; % Not known
        end

    else
        if all([localvelo(j)>=80, vehnum_km(j)<=20, vehnum_km(j)>=0])
            flow(j)=1; % free traffic
        elseif all([localvelo(j)>=80, vehnum_km(j)<=50,
vehnum_km(j)>20])
            flow(j)=2; % dense traffic
        elseif all([localvelo(j)<80, localvelo(j)>=30,
vehnum_km(j)<=50])
            flow(j)=3; % slow moving traffic
        elseif all([localvelo(j)==0, vehnum_km(j)==0])
```

```
            flow(j)=5; % Not known
        elseif all([any([localvelo(j)<30, vehnum_km(j)>50]), flow(j-
1)==2])
            flow(j)=3;
        elseif any([localvelo(j)<30, vehnum_km(j)>50])
            flow(j)=4; % congestion
        else flow(j)=5; % Not known
        end
   end

end

% Finding the traffic state of the route and travel times for sections
with
% no veh tracked

routeflow=mode(flow);

    if routeflow==1 | routeflow==2 | routeflow==3 | routeflow==5

        % Interpolate Travel times before plotting (flow wihout
congestion)
        nans=isnan(tt_km_sec);

tt_km_sec(nans)=interp1(d_km(~nans),tt_km_sec(~nans),d_km(nans),'linear
');

        tt_km_sec(isnan(tt_km_sec))=0;

        for j=1:k
            if tt_km_sec(j)==0
                tt_km_sec(j)=(tt_km_sec(j-1)+tt_km_sec(j+1))/2.0;
            end
        end

        for j=1:k
            traveltime(j)=tt_km_sec(j)*d_km(j);
        end

        % Take the values from previous segment in case of congestion
    else

        tt_km_sec(isnan(tt_km_sec))=0;

        for j=1:k
            if tt_km_sec(j)==0
                vehnum_km(j)=vehnum_km(j-1);
                tt_km_sec(j)=tt_km_sec(j-1);
            end
        end

        for j=1:k
            traveltime(j)=tt_km_sec(j)*d_km(j);
        end

    end
```

```
    avg_tt_km_s=sum(tt_km_sec(1:k))/k;

% Total time and distance for Route 46
totaltime=sectime(k); % Seconds
totaldis=secdis(k); % Metres
```

ACKNOWLEDGEMENTS

This Thesis was commissioned by Deutsches Zentrum für Luft und Raumfahrt (DLR) Institute of Remote Sensing, and without them this project wouldn't have been possible.

Personally, I would like to thank my boss, Dr. Franz Kurz, for his initiation and supervision throughout this project and for helping me with the paperwork and contracts needed for working at DLR and between DLR and TUM. I would like to thank Dr. Matthias Spangler for the advice and guidance given throughout my thesis. I would especially like to thank him for his generosity in giving me short-notice appointments and clearing all my doubts. Lastly, I would like to thank my friends and co-workers for their support during my time at DLR.

I dedicate this thesis to my family, especially my parents whose love and unfathomable support kept me motivated throughout my studies in Germany.

DECLARATION OF THE MASTER THESIS

I hereby declare that the following master thesis “Accurate Estimation of Travel Times on the Traffic Data Extracted from Aerial Image Time Series” has been written only by the undersigned and without any assistance from third parties.

Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself and I certify that the content of this report is the result of work done by me and has not been submitted for a higher degree to any other University or Institution.

To date, my Master thesis has not been submitted to any other board of examiners in the same or a similar format and has not been published yet.

Munich, 28 November, 2013

Signature of Mihir Shah

Mihir Shah