Vehicle Reidentification and Travel Time Estimation On Congested Highway

Gen Li

Kongens Lyngby 2007
IMM-MSC-2007-

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-MSC: ISSN

Preface

This thesis is finished at Informatics Mathematical Modelling, the Technical University of Denmark in fulfillment of the requirements for acquiring the master degree in engineering.

The purpose of this thesis is to build up a new mathematical model for reidentifying vehicles and thereby estimating travel time on congested highways. The essential part of the thesis is the vehicle reidentification, after which several ways of travel time estimation are given.

Gen Li

Lyngby, September 2007

Acknowledgements

I would like to thank my advisor Klaus Kaae Andersen for spending a lot of time reading my hundred-page reports and giving me many useful suggestions and precious advices. I am grateful to my best friends, who have taken very good care of me when I caught a terrible cold right before the deadline of this thesis. And finally, I would like to thank my family for supporting and encouraging me all the time, and for their endless love.

Best wishes to all of you.

For a better tomorrow,

By Gen

Contents

Chapter 1

Introduction

1.1 Problem Description

Travel time from one place to another under congested traffic is always an interesting thing that people concern. Lots of efforts have been made to monitor traffic and estimate travel time. Yet, an easy and economical way of estimating travel time is still of great need nowadays.

Among all the methods for estimating travel time under congested traffic, vehicle reidentification is a very smart way. Once a vehicle detected at one place is reidentified at another place, the travel time of that vehicle is easily estimated by the difference between the arrival times at those two places of detection.

Most highways are equipped with traffic detectors. Those detectors, which are normally installed within every kilometer on the highway, can monitor and record the velocity, time and length of vehicles when they pass by. However, those detectors can only measure information at discrete points but can not provide information of the link between detectors, which means that the vehicles travel time can not be monitored directly. Vehicle reidentification is a way to infer traffic conditions between detectors based on the discrete points information.

1.2 Thesis Achievements

This thesis presents an algorithm to reidentify vehicles and thereby estimate travel time between consecutive detectors on a congested highway, using those information obtained from the already-exist highway traffic detectors. Therefore, there is no need to set up new hardware equipments on the highway, saving a lot of costs.

The purpose of this algorithm is not reidentifying every vehicle on the highway. Actually the algorithm will reidentify 5 out of every 20 vehicles, and this will provide enough information for traffic surveillance.

The way of vehicle reidentification developed by this thesis is different from any of those already-exist methods. It provides good results and it is very easy to implement.

1.3 Thesis Outline

This thesis consists of four main parts, namely

- Data —Analysis of simulated traffic data;
- Methods —Algorithm of vehicle reidentification on highway;
- Results —Outcome of Applying the methods on simulated data. Comparison of travel time estimation between the algorithms from this thesis and the harmonic mean method is included;
- Conclusions

Chapter 2

The Data

Before introducing the methods of reidentifying vehicles and estimating travel time that presented in this thesis, this chapter, will briefly discuss the data that have been used for developing and testing those methods.

2.1 Data Description

The data that have been analyzed are the simulated information of traffic status on a Danish highway. The simulation is done by the popular software VISSIM. It is not the author of this thesis who simulated the data.

2.1.1 Available Information

The data contains the following useful information:

- **Vehicle Length** —Detector measures this with some error, which is normally distributed with mean 0 and standard deviation equal to 1% of the vehicle length[1]. Unit: meter;

- **Vehicle ID** —Only available in simulated data. This is a very important information for testing the performance of the algorithm presented by this thesis;
- **Vehicle velocity** —Velocity when detected. Unit: meter/second;
- **Time of detection** —Time that vehicle arrives at a detector. The first detection starts at 0. Unit: second;
- **Check Point** —Detector at which the vehicle is detected.

All the data are registered as time passes by, containing about 7 continuous hours of information. The data contains 22 detectors, in which 8 of them are going to be used for analysis. Those 8 detectors are No. 15, 16, 19, 20, 8, 9, 21 and 22, forming 6 pairs of consecutive detectors, as is shown by Figure 2.1. Distances between these 6 pairs of consecutive detectors are known.

2.1.2 Measurement Errors

It is unavoidable that data recorded at detectors contain measurement errors(noise). The error of measuring vehicle length, as described in the previous section, is normally distributed with mean 0 and standard deviation equal to 1% of the vehicle length. Equivalently, the accuracy of measuring vehicle length is approximately $me = 3\%$, which means that

$$L(1 - me) \leq LM \leq L(1 + me) \tag{2.1}$$

where L is the true length of a vehicle and LM is the measured length. Equivalently,

$$\frac{LM}{1 + me} \leq L \leq \frac{LM}{1 - me} \tag{2.2}$$

Therefore, two detected vehicles i and j are said to be a **Possible Match** if (2.3) and (2.4) are satisfied at the same time.

$$\frac{LM_i}{1 + me} \leq \frac{LM_j}{1 - me} \tag{2.3}$$

$$\frac{LM_i}{1 - me} \geq \frac{LM_j}{1 + me} \tag{2.4}$$

Since other measurement errors will not affect the vehicle reidentification algorithm presented by this thesis, they are not going to be discussed here.

[1] More detail is described in Section 2.1.2.

2.2 Data Analysis

To make the analysis easy to understand, some basic concepts on a highway are going to be explained first.

For each pair of consecutive detectors, the detector which vehicles pass first is called **Upstream Detector(UD)**, the other one is called **Downstream Detector(DD)**. Vehicles pass the UD with some order. This vehicle sequence order, however, could change when those vehicles arrive at DD, due to lane or position shifts, or whatsoever. The purpose of this section is to investigate the probability of maintaining the same vehicle sequence order between every two consecutive detectors. Probabilities are calculated respectively both from the whole data set, which contains about 7 hours of information, and when the traffic is comparatively congested. The data and its analysis are programmed and processed by the statistical software **R**.

2.2.1 Traffic Flow

The traffic flow status is reflected by the vehicle velocity on the highway. Simply speaking, when the traffic is not heavy, vehicles can drive with high speed, which of course is limited by the traffic regulations, and this is called free flow traffic. On the contrary, when the traffic is very heavy, the driving speed is slower or much slower than that of free flow condition, and this is called the congested flow traffic.

The detected vehicle velocities at those 8 used detectors are plotted in Figures 2.2 to 2.5. And it can be seen that for most of the detectors, during the time period 9000(seconds) to 13000(seconds), the average vehicle speed, which is around 15 units, is lower than other time periods. This period of time is treated as the congested period. Therefore, the aforementioned probability of maintaining the sequence order under the congested traffic is calculated in this period of time.

2.2.2 Large Vehicles

Vehicles have different lengths. Since the vehicle length could be a very important information for vehicle reidentification, the vehicle length shall be analyzed. In this section, the histogram of vehicle length is shown by Figure 2.6 and 2.7. It can be seen that using 6 meters as a classification factor to distinguish small and large vehicles might be a good choice. The reason why the large vehicles shall be distinguished out will be explained in Section 3.4.2.

2.2.3 Probability of Maintaining Sequence Order

For every pair of the aforementioned 6 pairs of consecutive detectors, the probabilities of maintaining the vehicle sequence order are given by Table 2.1 to 2.6.

It is seen that vehicles are likely to change lanes and switch sequence order from one detector to another during the whole simulation period. Even during the period when the traffic is comparatively slow, between time 9000(s) and 13000(s), shifts occur very often, although there are less shifts than other periods. For most pairs of detectors, maintaining the same sequence order with 10 vehicles has a probability around 10% or lower(the probability is about 34% between detectors 16 and 20). The more vehicles in a sequence, the smaller the probability of maintaining the same sequence order will be.

	Whole Period		Congested Period	
No. of vehicles	7092		1689	
No. of vehicles in the sequence	No. of matches	Probability	No. of matches	Probability
1	6000	0.8460	1490	0.8822
2	2848	0.4016	827	0.4896
3	1575	0.2221	539	0.3191
4	961	0.1355	378	0.2238
5	624	0.0880	271	0.1605
6	415	0.0585	200	0.1184
7	281	0.0396	150	0.0888
8	201	0.0283	116	0.0687
9	145	0.0204	87	0.0515
10	103	0.0145	65	0.0385
11	71	0.0100	48	0.0284
12	49	0.0069	34	0.0201
13	33	0.0047	24	0.0142
14	21	0.0030	17	0.0101
15	11	0.0016	10	0.0059
16	5	0.0007	5	0.0030
17	3	0.0004	3	0.0018
18	2	0.0003	2	0.0012
19	1	0.0001	1	0.0006
20	0	0	0	0

Table 2.1: The matching probability from detector 15 to 19, with distance 367 meters.

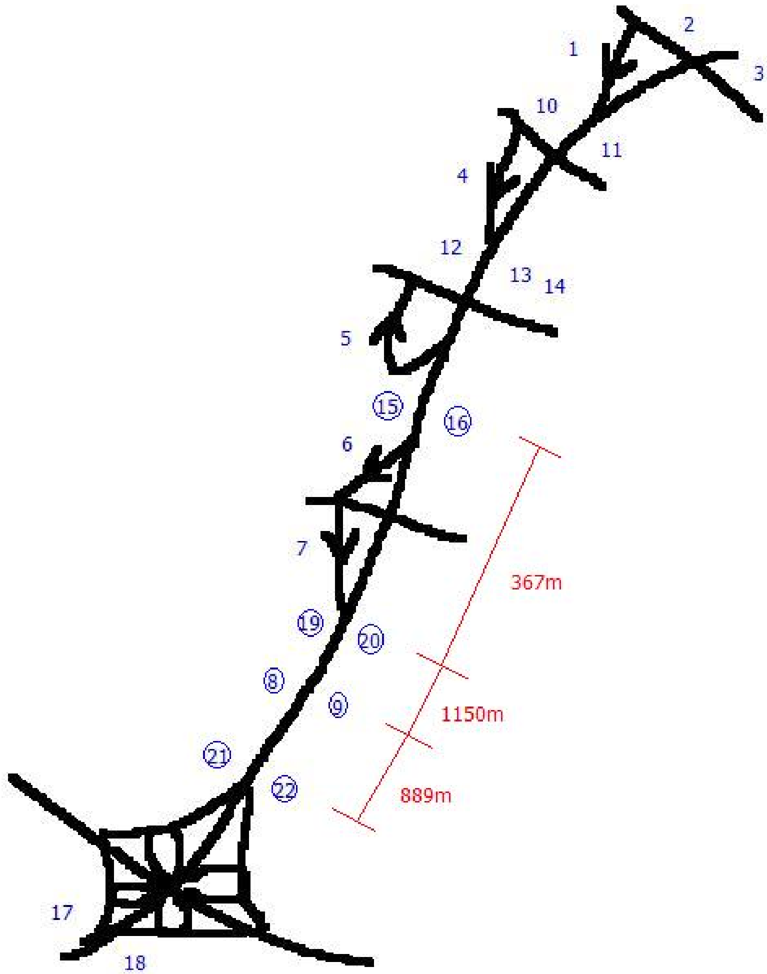

Figure 2.1: The map of the highway on which the simulated data are made. The longest black line divides the highway into two lanes(outer and inner lane). Traffic detectors are marked by blue numbers. Those 8 detectors used by this thesis are marked with blue circles. Distances between them are given by the red lines and letters.

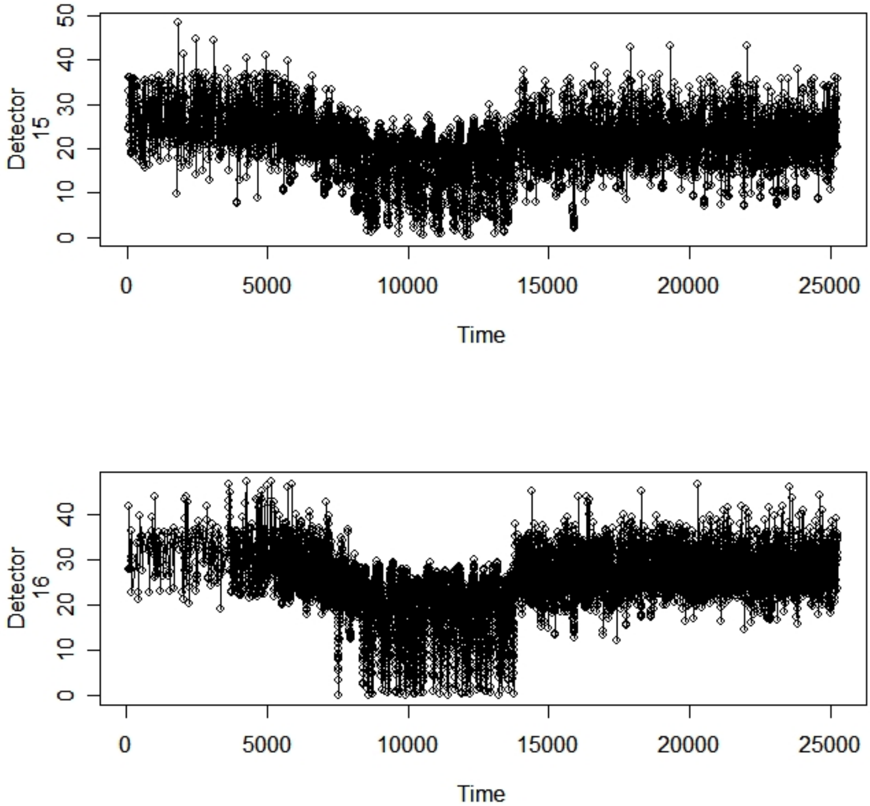

Figure 2.2: Vehicle speed at detectors 15 and 16.

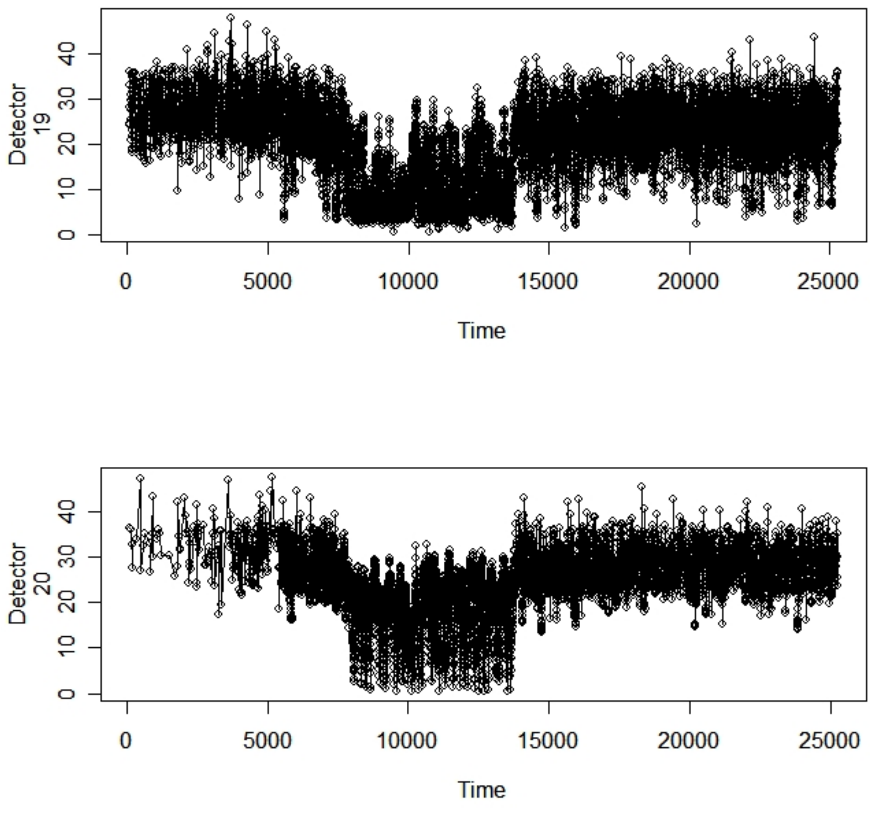

Figure 2.3: Vehicle speed at detectors 19 and 20.

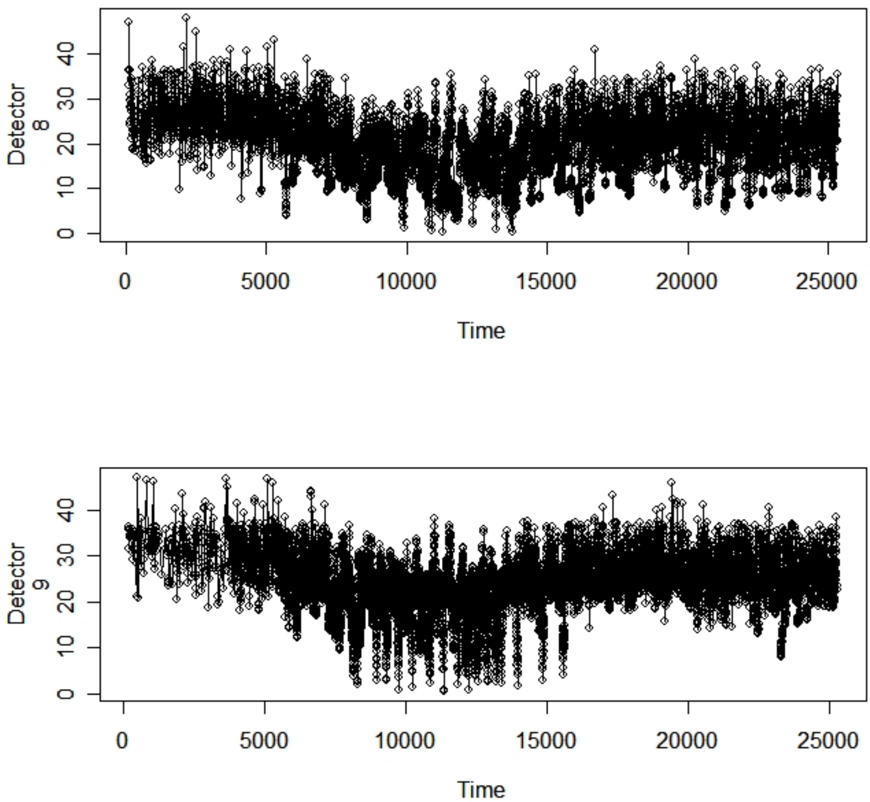

Figure 2.4: Vehicle speed at detectors 8 and 9.

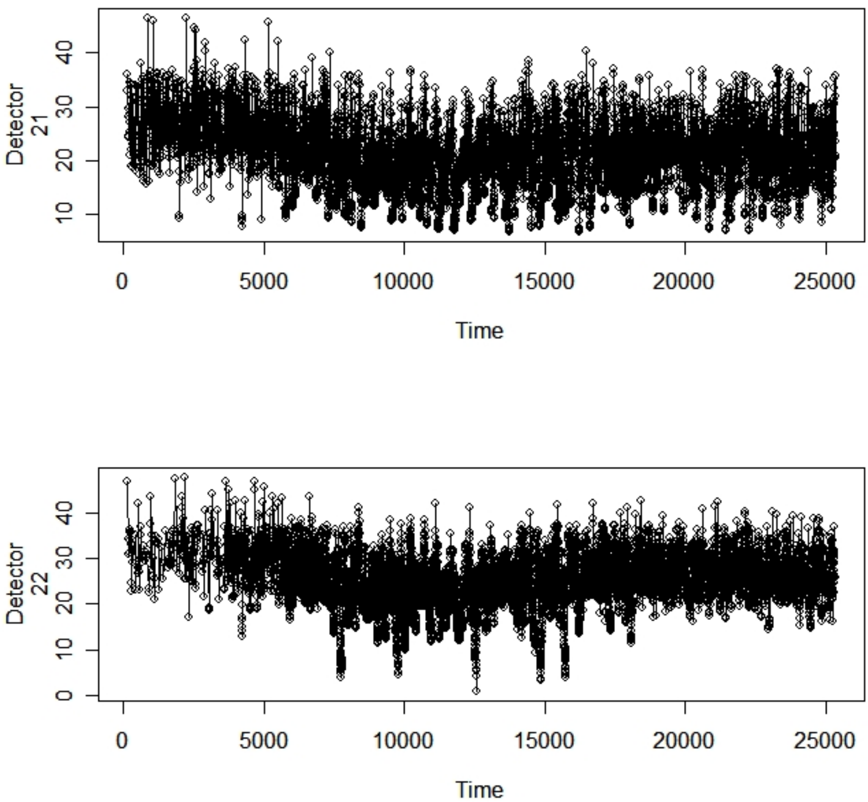

Figure 2.5: Vehicle speed at detectors 21 and 22.

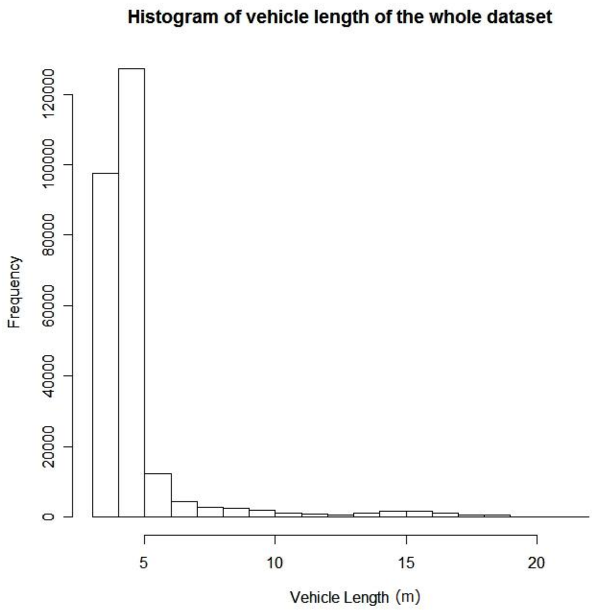

Figure 2.6: Histogram of vehicle length of all data.

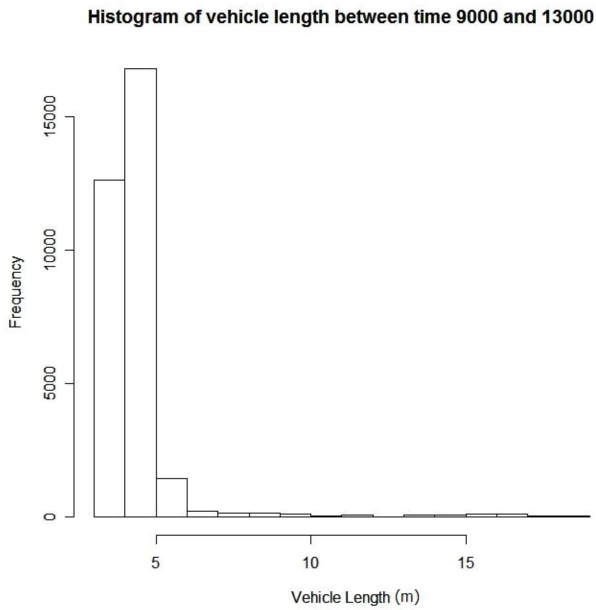

Figure 2.7: Histogram of vehicle length during congested period.

	Whole Period		Congested Period	
No. of vehicles	8234		2010	
No. of vehicles in the sequence	No. of matches	Probability	No. of matches	Probability
1	6144	0.7462	1903	0.9468
2	4617	0.5607	1677	0.8343
3	3699	0.4492	1492	0.7423
4	3046	0.3699	1336	0.6647
5	2552	0.3099	1199	0.5965
6	2170	0.2635	1072	0.5333
7	1861	0.2260	960	0.4776
8	1607	0.1952	859	0.4274
9	1405	0.1706	765	0.3806
10	1243	0.1510	680	0.3383
11	1112	0.1350	609	0.3030
12	1002	0.1217	550	0.2736
13	901	0.1094	496	0.2468
14	812	0.0986	447	0.2224
15	731	0.0888	400	0.1990
16	657	0.0798	357	0.1776
17	592	0.0719	319	0.1587
18	536	0.0651	287	0.1428
19	485	0.0589	259	0.1289
20	442	0.0537	236	0.1174
21	402	0.0488	215	0.1070
22	365	0.0443	197	0.0980
23	330	0.0401	179	0.0891
24	297	0.0361	163	0.0811
25	265	0.0322	147	0.0731
26	235	0.0285	132	0.0657
27	208	0.0253	117	0.0582
28	183	0.0222	103	0.0512
29	160	0.0194	90	0.0448
30	140	0.0170	78	0.0388
31	125	0.0152	68	0.0338
32	112	0.0136	60	0.0299
33	102	0.0124	54	0.0269
34	93	0.0113	48	0.0239
35	85	0.0103	43	0.0214
36	79	0.0096	39	0.0194
37	73	0.0089	35	0.0174
38	67	0.0081	31	0.0154
39	61	0.0074	27	0.0134
40	56	0.0068	24	0.0119
41	51	0.0062	21	0.0104
42	47	0.0057	19	0.0095
43	44	0.0053	18	0.0090
44	41	0.0050	17	0.0085
45	38	0.0046	16	0.0080
46	35	0.0043	15	0.0075
47	32	0.0039	14	0.0070
48	29	0.0035	13	0.0065
49	27	0.0033	12	0.0060
50	25	0.0030	11	0.0055
51	23	0.0028	10	0.0050
52	21	0.0026	9	0.0045
53	19	0.0023	8	0.0040
54	17	0.0021	7	0.0035
55	15	0.0018	6	0.0030
56	13	0.0016	5	0.0025
57	11	0.0013	4	0.0020
58	9	0.0011	3	0.0015
59	7	0.0009	2	0.0010
60	5	0.0006	1	0.0005
61	3	0.0004	0	0
62	2	0.0002	0	0
63	1	0.0001	0	0
64	0	0	0	0

Table 2.2: The matching probability from detector 16 to 20, with distance 367 meters.

	Whole Period		Congested Period	
No. of vehicles	9888		2028	
No. of vehicles in the sequence	No. of matches	Probability	No. of matches	Probability
1	6259	0.6330	1500	0.7396
2	2667	0.2697	1020	0.5030
3	1648	0.1667	748	0.3688
4	1130	0.1143	561	0.2766
5	792	0.0801	410	0.2022
6	567	0.0573	300	0.1479
7	405	0.0410	213	0.1050
8	291	0.0294	153	0.0754
9	210	0.0212	108	0.0533
10	151	0.0153	74	0.0365
11	108	0.0109	51	0.0251
12	76	0.0077	32	0.0158
13	51	0.0052	18	0.0089
14	38	0.0038	12	0.0059
15	27	0.0027	7	0.0035
16	17	0.0017	3	0.0015
17	12	0.0012	2	0.0010
18	7	0.0007	1	0.0005
19	4	0.0004	0	0
20	2	0.0002	0	0
21	1	0.0001	0	0
22	0	0	0	0

Table 2.3: The matching probability from detector 19 to 8, with distance 1150 meters.

	Whole Period		Congested Period	
No. of vehicles	7280		2110	
No. of vehicles in the sequence	No. of matches	Probability	No. of matches	Probability
1	5960	0.8187	1838	0.8711
2	3674	0.5047	1374	0.6512
3	2533	0.3479	1066	0.5052
4	1834	0.2519	852	0.4038
5	1341	0.1842	674	0.3194
6	985	0.1353	525	0.2488
7	734	0.1008	400	0.1896
8	549	0.0754	304	0.1441
9	414	0.0569	229	0.1085
10	316	0.0434	174	0.0825
11	248	0.0341	138	0.0654
12	192	0.0264	105	0.0498
13	148	0.0203	77	0.0365
14	113	0.0155	56	0.0265
15	87	0.0120	41	0.0194
16	67	0.0092	31	0.0147
17	51	0.0070	24	0.0114
18	40	0.0055	20	0.0095
19	32	0.0044	17	0.0081
20	24	0.0033	14	0.0066
21	18	0.0025	11	0.0052
22	12	0.0016	8	0.0038
23	6	0.0008	5	0.0024
24	3	0.0004	3	0.0014
25	1	0.0001	1	0.0005
26	0	0	0	0

Table 2.4: The matching probability from detector 20 to 9, with distance 1150 meters.

	Whole Period		Congested Period	
No. of vehicles	7579		1769	
No. of vehicles in the sequence	No. of matches	Probability	No. of matches	Probability
1	5646	0.7450	1397	0.7897
2	3336	0.4402	1047	0.5919
3	2358	0.3111	813	0.4596
4	1762	0.2325	647	0.3657
5	1354	0.1787	515	0.2911
6	1065	0.1405	415	0.2346
7	857	0.1131	335	0.1894
8	699	0.0922	272	0.1538
9	580	0.0765	225	0.1272
10	478	0.0631	184	0.1040
11	393	0.0519	151	0.0854
12	327	0.0431	127	0.0718
13	271	0.0358	108	0.0611
14	228	0.0301	91	0.0514
15	193	0.0255	78	0.0441
16	165	0.0218	70	0.0396
17	143	0.0189	62	0.0350
18	125	0.0165	55	0.0311
19	108	0.0142	49	0.0277
20	93	0.0123	44	0.0249
21	79	0.0104	40	0.0226
22	68	0.0090	36	0.0204
23	58	0.0077	32	0.0181
24	48	0.0063	28	0.0158
25	40	0.0053	25	0.0141
26	34	0.0045	22	0.0124
27	29	0.0038	19	0.0107
28	25	0.0033	16	0.0090
29	21	0.0028	13	0.0073
30	17	0.0022	10	0.0057
31	13	0.0017	7	0.0040
32	10	0.0013	5	0.0028
33	7	0.0009	3	0.0017
34	5	0.0007	2	0.0011
35	3	0.0004	1	0.0006
36	1	0.0001	0	0
37	0	0	0	0

Table 2.5: The matching probability from detector 8 to 21, with distance 889 meters.

	Whole Period		Congested Period	
No. of vehicles	9589		2394	
No. of vehicles in the sequence	No. of matches	Probability	No. of matches	Probability
1	7539	0.7862	2006	0.8379
2	5098	0.5317	1505	0.6287
3	3785	0.3947	1174	0.4904
4	2904	0.3028	943	0.3939
5	2290	0.2388	774	0.3233
6	1834	0.1913	641	0.2678
7	1503	0.1567	536	0.2239
8	1248	0.1301	454	0.1896
9	1042	0.1087	385	0.1608
10	873	0.0910	323	0.1349
11	735	0.0767	270	0.1128
12	626	0.0653	226	0.0944
13	533	0.0556	190	0.0794
14	451	0.0470	159	0.0664
15	379	0.0395	131	0.0547
16	321	0.0335	109	0.0455
17	270	0.0282	91	0.0380
18	228	0.0238	76	0.0317
19	198	0.0206	63	0.0263
20	174	0.0181	54	0.0226
21	152	0.0159	46	0.0192
22	131	0.0137	38	0.0159
23	110	0.0115	30	0.0125
24	93	0.0097	24	0.0100
25	77	0.0080	19	0.0079
26	62	0.0065	14	0.0058
27	50	0.0052	10	0.0042
28	40	0.0042	7	0.0029
29	32	0.0033	5	0.0021
30	25	0.0026	4	0.0017
31	21	0.0022	3	0.0013
32	17	0.0018	2	0.0008
33	13	0.0014	1	0.0004
34	9	0.0009	0	0
35	6	0.0006	0	0
36	3	0.0003	0	0
37	1	0.0001	0	0
38	0	0	0	0

Table 2.6: The matching probability from detector 9 to 22, with distance 889 meters.

Chapter 3

Methods for Reidentifying Vehicles on highway

In this chapter, a few already-exist methods and the algorithm provide by this thesis for reidentifying vehicles on highways are going to be discussed.

3.1 Normal Existing Methods

Literatures on real time traffic, e.g. travel time estimation and prediction, are very prolific. Among all different approaches of traffic surveillance, vehicle reidentification is a very smart way. So far, the vehicle reidentification study using existing highway detectors is mostly done by Benjamin Coifman and his team. In this part, Coifman's work and some other methods of vehicle reidentification are going to be briefly introduced.

3.1.1 Coifman's Methods

Benjamin Coifman has developed algorithms for reidentifying vehicles between detector stations on highway[1, 2, 3, 4]. When the traffic is uncongested, his algorithm matches vehicles within a time window of reasonable free flow travel times. This part of work is comparatively simple because under free flow traffic, vehicles travels with nearly constant velocity between detector stations. Therefore the point information obtained at each detector are assumed to be representative of extended links spanning detectors. However, in case of congested traffic, this assumption is usually not valid. The following part of this section will mainly introduce his algorithm dealing with congested freeways.

In the congested traffic case, Coifman's algorithm reidentifies measurements from distinct vehicles using existing loop detector infrastructure. The distinct vehicles he used are long vehicles, whose lengths are longer than a threshold based on the 90th percentile length for all the vehicles passing the downstream loop detector. Vehicles shorter than that threshold are not considered.

Coifman's earlier work assumed that platoons of 5-10 vehicles regularly pass both detectors in the same lane. However, that algorithm fails in the presence of frequent lane change vehicles. His later work allows for reidentification even when many vehicles pass only one of the detector stations without being observed at the other one, meaning his new algorithm is more robust to vehicle reordering and unmatchable vehicles that enter or leave a subject lane between detector stations.

In his algorithm, each long vehicle at the downstream detector is considered as primary vehicle with a set of candidate vehicles that are feasible matches at the upstream detector. The upstream vehicle candidates include all vehicles(not only the long ones) and are chosen using two rules: first, to ensure positive travel time a candidate must arrive at the upstream station before the arrival of the primary vehicle at the downstream detector, and second, the total number of candidates shall not exceed the jam density of link, i.e., the storage capacity, n, between the two detectors. Then the length range(due to measurement uncertainty) of the primary vehicle is compared with candidate vehicles present within the n most recent upstream detector arrivals. Possible matches are found out if their length ranges intersect with each other. All possible matches are stored in the so called Travel Time Matrix(TTM), indexed by travel time, where the travel time for each matched pair is obtained by subtracting the arrival time of the candidate match at the upstream detector from the arrival time of the given primary vehicle at the downstream detector. The rows of this TTM correspond to the primary vehicle number, which records the order of the arrivals of the primary vehicles. The indices of the columns of the TTM represent the possible travel times rounded to the nearest integer second. To avoid a very large column size and thereby improve the computational efficiency, the width of TTM is constrained to the travel time corresponding to link velocities falling between 2 mph[1] and 90 mph. The TTM is populated with 0's except for the travel times for each of the given primary vehicle's possible matches, which are given values of 1's. Any row of the TTM can have at most one true match. The false matches are randomly distributed within each row, yielding a low density of possible matches throughout the entire matrix when considering several rows. For the rows where true matches exist, the true matches from consecutive vehicles fall in a small range of columns and increase the density of possible matches above the background level of the false matches, as is shown by Figure 3.1(A). Then the TTM is transformed to the Maximum Density Matrix(MDM) to identify the dense areas, as is shown by Figure 3.1(B). After that the Most Probable Travel Time(MPTT) is figured out, as is shown by Figure 3.1(C).

3.1.2 Some Other Vehicle Reidentification Methods

There are some vehicle reidentification methods[5, 6, 7] that demanding hardware equipments other than the already built highway detectors. Such methods will definitely involves new costs, which could be very high. The purpose of this thesis is to use the current highway detectors to reidentify vehicles, therefore those methods that demanding new hardwares will not be described in details.

3.1.3 Some Other Statistical Measures for Comparing Length of Vehicles

In this part, some existing statistical measures for comparing length of vehicles are briefly introduced. When matching sequences of vehicles, the lengths of vehicles are compared. To do this, four statistical measures could be used: Relative Pattern Score(RPS), Average Pattern Score(APS), Correlation Pattern Score(CPS), and Division Pattern Score(DPS), which are given by

$$RPS = \sum_i 2\frac{x_i - y_i}{x_i + y_i} \tag{3.1}$$

$$APS = \sum_i \frac{x_i - y_i}{n} \tag{3.2}$$

$$CPS = \frac{cov(\underline{x}, \underline{y})}{\sqrt{var(\underline{x}) \cdot var(\underline{y})}} \tag{3.3}$$

$$RPS = \sum_i \frac{x_i}{y_i} \tag{3.4}$$

[1]"mph" stands for miles per hour.

where $\underline{x} = \{x_i\}$ and $\underline{y} = \{y_i\}$ are the vehicle sequences to be matched, $i \in \{1, 2, ..., n\}$. If the sequences match exactly RPS and APS are 0, CPS is 1, and DPS IS n. These scores may be weighted using e.g. a kernel, and then added to get a single score with values ranging from 0 to 4. If this score is close to 4, it means that the sequences match very well using all criteria.

3.2 Methods Provided in this Thesis

The objective of this thesis is to find a way, comparatively easier and faster than those already existing ones, to reidentify vehicles and thereby estimate travel time between consecutive detectors on congested highway. This section will describe the algorithm developed by this thesis.

3.2.1 Basic Ideas

The basic idea of estimating travel time between consecutive detectors is to reidentify vehicle. If a vehicle is detected at DD, and reidentified from the records at UD, then its travel time between these two detectors is estimated by the difference between the arrival times at the two detectors.

For the sake of saving costs, reidentification of vehicles should only use available information from the detectors already built on the highway. Those information consist of vehicle lengths, velocities when detected, time of detections, etc., all with measurement errors, among which only the measurement error of vehicle length will be considered, as already stated in Section 2.1.2.

Given those information, it might be better to reidentify a sequence of vehicles, not a single one. Because

- Information of a single vehicle is very limited, therefore lots of vehicles could have similar information, e.g. lots of vehicles have similar lengths, which consequently make the reidentification difficult;
- A sequence of vehicles always contains more information than a single one does, which will hopefully increase the rate of correct reidentification;
- According to the analysis in the previous chapter, the probability of maintaining the same long vehicles sequence order between two consecutive detectors on the highway is small, which means that if a long sequence of

vehicles is reidentified, it may have a high probability to be a correct reidentification - at least the probability is higher than that of reidentifying a single vehicle.

The following two ways of vehicle reidentification come naturally, of which the second one is used by this thesis.

3.2.1.1 Reidentifying Small Sequence of Vehicles First

Again, according to the previous analysis, probability of maintaining the same short vehicle sequence between two consecutive detectors on the highway is comparatively big. So reidentification of vehicles could focus on small sequence of vehicles. This idea is to first locate, for a small sequence of n vehicles($n = 4$ for instance) at downstream detector(DD), a few possible corresponding sequences(candidates) at the upstream detector(UD). Then, the neighboring vehicles information of those possible sequences are used to decide which one could be the most likely match. This is, for short, called the small to big method, standing for from small sequence to big sequence.

3.2.1.2 Reidentifying Large Sequence of Vehicles First

This idea is like the other way round. The risk of using the small to big method is that it may turn out that there are no matches at UD for a lot of small sequences from DD, which results in a huge waste of computation and lots of wrong reidentifications.

To avoid this, big sequence of vehicles are reidentified first. The idea is to first locate, for a big sequence of N vehicles(the primary sequence) at DD, the Most Possible Corresponding Sequence(MPCS) at UD. It is not demanded that the primary sequence and its MPCS are exactly the same, which in practice is not likely to happen often. On the contrary, it only demands that there are a few vehicles from the two big sequences are the same(their sequence order may have changed though). The MPCS is the sequence that is likely to contain more same vehicles of the primary sequence than other sequences do. Then within the primary sequence and its MPCS, a small sequence of $n(n < N)$ vehicles will be reidentified. This is, for short, called the big to small method, standing for from big sequence to small sequence.

For the same amount of vehicles to be reidentified, there are less big sequences than small ones, meaning that there are less work to do in the big to small method than the other way round. Each big sequence has a MPCS, but not every small sequence has a real match. So the big to small method does not waste as much computation as the other one does.

The method presented by this thesis is based on this idea —**from big to small**. The following of this thesis is going to discus this method in detail.

3.3 Some Concepts

Before explaining the method, several important concepts are going to be introduced in this section.

3.3.1 Possible Match

The concept of a possible match for one vehicle has already been introduced in Section 2.1.2 by equations (2.3) and (2.4).

Two sequences of vehicles are said to be a possible match if they possibly consist of a few same vehicles. For instance, the MPCS defined in Section 3.2.1.2 is a possible match for its primary sequence.

3.3.2 Maximum Correlation Method

The maximum correlation method is the core of the algorithm provided by this thesis. It is comprehensible to believe that a possible match for a sequence of vehicles shares some similar pattern with that sequence. Given those available information from the simulated data, a pattern of a vehicles sequence is mainly described by vehicle lengths in the sequence. Therefore, if two sequences of vehicles are a pair of possible match, their sequence of vehicle lengths shall be correlated. The more correlated, the more possible that the two sequences of vehicles are a pair of match. Based on this reason, the maximum correlation method is used to find the most possible match for vehicle sequences.

Basically speaking, the maximum correlation method is to select a sequence of vehicles, from some sequences, whose vehicle lengths sequence has the largest correlation coefficient with the sequence of vehicles to be matched. Here the elements for calculating the correlation coefficient are the vehicle lengths. Under different situations, the way of using the maximum correlation method varies. Details are explained in the following related sections.

3.3.3 Time Window

Simply speaking, a time window of one vehicle or a sequence of vehicles is the possible time interval in which a true match should lie. A time window can be determined by the maximum and minimum link speed[2] on the highway. Assume the maximum speed allowed on the highway is $Vmax$, and the minimum speed is $Vmin$(this varies according to the real time traffic condition). Let A denote the upstream detector, B denote the downstream one, $v.t_B$ denote the time of detecting a vehicle at B, $dist_{A\rightarrow B}$ denote the distance between A and B, the time window for this vehicle arriving at A is

$$[v.t_B - dist_{A\rightarrow B}/Vmin, \qquad v.t_B - dist_{A\rightarrow B}/Vmax] \tag{3.5}$$

It is noticed that $Vmin$ is a very important factor to control the size of the time window. When it is too small, the time window becomes too big for the maximum correlation method to work. Because the maximum correlation method works on the vehicle lengths sequence, if the time window of a vehicles sequence is too big, there will possibly be a lot of vehicles sequences in that time window sharing similar pattern with the one to be matched with, and there is no guarantee that the true match will be found by the maximum correlation method. In this thesis, time windows are determined according to different situations. It will be explained in detail in the following related sections.

3.4 How The Algorithm Works Under Congested Traffic

In this section, the way of how the algorithm works under congested traffic is explained.

A flowchart of the algorithm presented by this thesis is shown by Figure 3.2. The following sections are going to explain details step by step. Concepts involved will be introduced along the way.

[2]Link speed/velocity: the average speed/velocity of a vehicle travels from one detector to another.

3.4.1 Recording Vehicles At Downstream Detector

The goal of vehicle reidentification is to monitor traffic conditions on road and use this information to make useful estimations. So this job should be done in real time. When applying the method provide in this thesis, the traffic is reidentified every 10 minutes.

For a pair of consecutive detectors on the highway, vehicles pass the downstream detector(denoted by $v.dd$) are recorded from time $t1$ to time $t2$, with $t2 - t1 = 10mins$. Figure 3.3 gives a visual expression of $v.dd$.

3.4.2 Finding Large Vehicles

This section is going to discuss the usage of large vehicles.

3.4.2.1 Reasons of Using Large Vehicles

The reason of finding large vehicles is that:

- Easy to find
- Useful information

Vehicles on the highway differs in their lengths. Most cars share similar lengths, while there are a few large vehicles on the road which appear once in a while. Their lengths are significantly different from small ones. Therefore reidentification of a large vehicle is much easier than for a small one. It is done by simply checking the vehicles length measurements(according to Equation (2.3) and (2.4)) in the corresponding detectors within a suitable time window, in which a true match could possibly be found. The time window of a large vehicle is determined by equation (3.5). The parameter $Vmin$ is set to be $4m/s$ and $Vmax$ is constrained by traffic regulation, setting to be $120km/h$, i.e. $33m/s$.

Similarly, it will be noted that many parameters used in this thesis are established empirically. But they have proven being able to produce acceptable results when the algorithm is implemented. Further, some of the parameters can be adjusted dynamically to satisfy additional requirements, e.g. in Section 3.5.1 a way of adjusting parameter $Vmin$ is given. This is a good characteristic meaning that the algorithm is adjustable to various of situations.

3.4.2.2 Special Situation

In case more than one possible matches is found for a single large vehicle, the neighboring vehicles information are used to decide which match is more possible to be a true match. For example, if in the time window of a large vehicle, more than one possible matches are found according to Equation (2.3) and (2.4), then for each of these possible matches, a vehicles sequence with 5 vehicles before and 5 vehicles after that possible match is taken out. Similarly, the sequence containing the large vehicle to be matched is also taken out. Then the maximum correlation method is used to see which sequence of the possible match has the largest correlation coefficient with the large vehicle's sequence. Here, when calculating the correlation coefficient, the sequences are sorted according to vehicle lengths.

3.4.2.3 Using Large Vehicles Information

Once a most possible match for a large vehicle is found, the travel time of this large vehicle is easily determined. The large vehicle information is used to narrow the time window for other vehicles. When a large vehicle is reidentified, it is high chance that this is a true reidentification. So the travel time of this large vehicle is a very important information. All vehicles that near to that large vehicle are likely to have similar travel times. In this paper, vehicles that enter the same DD within 60 seconds before and after the large vehicle are narrowed according to the large vehicle's travel time, i.e. their travel time are assumed to be less than 1.2 times of the large one's. Details of how to use the large vehicles information to calculate time windows are explained in Section 3.4.5.

3.4.2.4 Removing Possible Fake Matches of Large Vehicles

A wrong reidentification of a large vehicle, which does not often occurs, could affect the whole reidentification badly. So before using the large vehicle's information, wrong reidentification of large vehicles should be removed as many as possible. In this paper, in every period of 10 minutes, those large vehicles reidentified with a travel time larger than 1.4 times or smaller than 0.6 times of the mean large vehicles travel time are removed. This way has proven being able to remove some of the wrong reidentifications when implemented; however, this will also remove some of the correct reidentifications. But after removing, there are still some reidentified large vehicles left, and the information they provide is adequate.

3.4.2.5 The Length of Large Vehicles

As discussed in Section 2.2.2, using 6 meters as a classification factor to distinguish small and large vehicles might be a good choice for most of cases. And this also has proven to be a good choice when the algorithm is implemented. One thing need to be noticed is that for the pair of detectors from 16 to 20, 5 meters shall be used as the classification criteria, otherwise there will be no large vehicles found. And there are not so many vehicles has length longer than 5 meters between these two detectors, so 5 meters is a sound choice for this particular situation.

3.4.3 Locating Big Sequences One By One

As mentioned in Section 3.4.1, the traffic is reidentified every 10 minutes. For a pair of consecutive detectors, at DD, during a 10-minute period, vehicles are recored(denoted by $v.dd$). This step, however, is to divide $v.dd$ into big sequences, one next to each other, all with N vehicles. After division, vehicles left(not enough N vehicles) are neglected. In this thesis, $N = 20$ is used. A big sequence is denoted by $v.ddN$. For the convenience of understanding this step, a visual expression is given by Figure 3.4.

3.4.4 Defining Types of Each Big Sequence

The purpose of defining types for $v.ddN$'s is to calculate time windows accordingly.

According to the large vehicles reidentified in a $v.dd$, different types may be given to different $v.ddN$'s. The idea is just an expansion of the one described in Section 3.4.2.3, namely that vehicles that enter the same DD within 60 seconds before and after[3] the large vehicle are assumed to have travel time less than 1.2 times of the large one's. Because it is reasonable to assume that vehicles near to each other have similar travel times under congested traffic in most of the cases. There can be 3 types for a $v.ddN$:

- Type 1: All vehicles in a $v.ddN$ are covered by one or more large vehicle's interval(s), as is shown by Figure 3.5.

[3] For convenience, this is called **the large vehicle's interval**.

- Type 2: Only some of, not all, vehicles in a $v.ddN$ are covered by one or more large vehicle's interval(s), as is shown by Figure 3.6.
- Type 3: None of the vehicles in a $v.ddN$ are covered by large vehicle's interval.

3.4.5 Computing Time Window For Each Big Sequence

After the type has been decided for a $v.ddN$, its time window can be calculated. First the boundaries of $v.dd$ vehicles travel times are calculated: When large vehicles are reidentified(denoted by $v.dd.large$), their travel times are estimated easily. Then it is assumed that the travel time of all vehicles in this $v.dd$ will not be longer than 1.3 times of the maximum travel time of the large vehicles, nor shorter than 0.8 times of the minimum, i.e.

$$tmin = 0.8min(v.dd.large.tt) \tag{3.6}$$

$$tmax = 1.3max(v.dd.large.tt) \tag{3.7}$$

where $tmin$ and $tmax$ denote the lower and upper bound of vehicles travel times in $v.dd$ respectively, $v.dd.large.tt$ denotes the travel times of large vehicles in that $v.dd$.

The way of calculating time window for different type of $v.ddN$ is described in the following enumerations:

1. For type 1: the time window for this type is calculated as:

   $$[v.ddN.tenter(1) - 1.2cover.tt.max, \quad v.ddN.tenter(N) - tmin] \tag{3.8}$$

   where $v.ddN.tenter(1)$ and $v.ddN.tenter(N)$ denote the time that the first and the last vehicle of a $v.ddN$ that enters the DD first and last respectively, $cover.tt.max$ denotes the maximum travel time of large vehicles whose intervals have covered this $v.ddN$.

2. For type 2: the time window for this type is calculated as:

   $$[v.ddN.tenter(1) - 1.3cover.tt.max, \quad v.ddN.tenter(N) - tmin] \tag{3.9}$$

   The multiplier on $cover.tt.max$ is bigger than that of a type 1 situation, because some vehicles in a type 2 $v.ddN$ are not covered by any large vehicle intervals, and this makes the travel time of those vehicles more uncertain.

3. For type 3: the time window for this type is calculated as:

$$[v.ddN.tenter(1) - tmax, \quad v.ddN.tenter(N) - tmin] \tag{3.10}$$

This time window illustrates more uncertainty of travel time.

3.4.6 Finding Most Possible Corresponding Sequences

This section describes how the big sequences $v.ddN$ are reidentified. As described before, the length of the big sequence $v.ddN$ is $N = 20$.

3.4.6.1 Purpose

The purpose of this step is to find one possible match for a $v.ddN$ with as many same vehicles of that $v.ddN$ as possible. Results of this step will directly affect the next step: the final reidentification of small sequences of vehicles, by which the travel time will be estimated.

3.4.6.2 How to Find

The simplest way to find the MPCS is to use the maximum correlation method. At DD, a big sequence $v.ddN$ is recorded, then the sequence is sorted according to the length of the vehicles detected. At UD, within its time window, all sequences with N vehicles will be tested: First take one sequence $v.udN$ at UD, sort by vehicle length, then calculate the correlation coefficient $cor(sort(v.ddN), sort(v.udN))$. The pair that has the largest correlation will be selected as the result. The reason why the sequence should be sorted is explained in Section 3.4.7.

3.4.6.3 Special Situations

There are mainly two kinds of special situations:

- It is possible that there are more than one pair of sequences which have the same largest correlation. But when the algorithm is implemented, it turns out that this kind of situation rarely occurs, and even if it occurs, those MPCS's found for one $v.ddN$ differ with each other in only one vehicle. And there are not many of them, usually at most two or three in a time window. Therefore just taking any one of them as the MPCS is acceptable.

- It is also possible that within the time window for a $v.ddN$ there are less than N vehicles. Under this situation, all those vehicles in the time window are treated as the MPCS for that $v.ddN$. This kind of situation also occurs rarely.

3.4.7 "To Small": Reidentifying Small Sequence of Vehicles From the Big Sequence

This section explains the core idea of the big to small method. The small vehicle sequence reidentified out of a big sequence $v.ddN$ is denoted as $v.ddn$, in which there are n vehicles. $n = 5$ is used in this thesis.

The way of reidentifying small sequence $v.ddn$ is also based on the maximum correlation method. When a MPCS for a $v.ddN$ is found, all sequences with n vehicles are taken out from that MPCS to compute the correlation coefficients with all n-vehicle sequences from the $v.ddN$. The pair that has the maximum correlation coefficient is selected out as a final match. Of course the elements of correlation are the vehicle lengths. But in this step, when calculating the correlation coefficient, the small sequences are not sorted according to the vehicle lengths, which differs from the way when reidentifying big vehicles sequences. The reasons for this difference are:

- The vehicles of a big sequence (primary sequence) and its possible match (candidate sequence) are not necessary to be exactly the same, and even if there are many vehicles are the same, the order of those vehicles may be different in the primary sequence and its candidate;

- But when a small sequence of n vehicles are reidentified from the big sequence of N vehicles, it is hoped that the small sequence $v.ddn$ and its best possible match share almost the same characteristics: same vehicles and same vehicles order. Since there are only 5 vehicles in a $v.ddn$, this hope is not a dream that never comes true.

3.4.8 Estimation of Travel Time

In this section, different ways of estimating travel time after the vehicles being reidentified are given.

3.4.8.1 Simple Way of Travel Time Estimation

Once a vehicle is reidentified, its travel time is easily estimated by the time difference of entering the UD and DD.

For the purpose of monitoring the traffic on highways, it is not necessary to estimate travel time of every vehicle on the road. Estimating travel time every a few seconds, or even minutes, one or half minute for instance, is enough.

As mentioned in the previous section, for every big sequence $v.ddN$, one small sequence $v.ddn$ is reidentified. This will give enough information.

Based on this, some further work can be done to give travel time estimation in different ways, i.e. Local Polynomial Regression Fitting(LPRF) and Exponentially Weighted Moving Average Method(EWMA).

3.4.8.2 Local Polynomial Regression Fitting

The LPRF will fit a polynomial surface determined by one or more numerical predictors, using local fitting. Since the reidentification algorithm presented by this thesis will reidentify 5 vehicles out of every 20 vehicles, and the time points of reidentifications are random and not continuous, therefore, if a travel time needs to be estimated at some time point where no vehicle is reidentified, a smooth function shall be fitted according to the vehicle reidentifications and thereby provide travel time estimation at any time point.

Before doing LPRF, for every $v.ddn$, a mean arrival time and a mean travel time estimation shall be calculated and used as one of the points to be fitted by LPRF. The LPRF will be done by the R function $loess$, with parameter $span = 0.3$ and $degree = 2$. Parameter $span$ controls the degree of smoothing with a default value 0.75. When its value gets smaller, the degree of smoothing is reduced, but the trend of the travel time will be fitted better. In this thesis, $span = 0.3$ is used. Parameter $degree$ decides the degree of the polynomials to be used. In this thesis, the default value $degree = 2$ is used.

3.4.8.3 EWMA Method

Besides, the exponentially weighted moving average method can be used for travel time estimation.

Inspired by the idea of time series analysis and exponential smoothing method, it is reasonable to put different weights, for instance higher weights on a large vehicle reidentification and reidentifications under a type 1 situation, and lower weights for type 3, etc., on different estimations. To use this method, every small sequence *v.ddn* needs to generate one arrival time and one travel time estimation first, just as what is done before LPRF.

The basic idea of using EWMA is illustrated by

$$\hat{T}_i = (1 - \lambda_i)\hat{T}_{i-1} + \lambda_i T_i \tag{3.11}$$

where T_i denotes the estimated travel time of the i'th *v.ddn*; $\hat{T}_i$ denotes the modified estimation using EWMA method; λ_i denotes the weight put on the estimation of the i'th *v.ddn*.

In this paper, λs are given by the following rules: $\lambda = 0.9$ if it is a large vehicle; when a *v.ddn* is under type 1 situation, $\lambda = 0.4$ if the previous estimation is from a large vehicle and the time point of the previous estimation is less than 15 seconds ago, otherwise $\lambda = 0.7$; $\lambda = 0.6$ if it is a type 2 situation; when a *v.ddn* is under type 3 situation, $\lambda = 0.5$.

The initial value, $\hat{T}_1$, is determined by the following rule: $\hat{T}_1$ is equal to the average of all T_i's, if the first estimation is not a large vehicle nor a type 1 situation; otherwise $\hat{T}_1 = T_1$.

By now the procedure of vehicle reidentification is fully described.

3.5 Estimation of Travel Time Between Several Detectors

The method provided by this thesis, as described previously, is for estimating travel time between two consecutive detectors on a highway. It might be interesting to see if this method could be applied to estimate travel time between several consecutive detectors in one lane.

3.5.1 Two Ways of Estimation

The highway, given by the simulated data, has two lanes. For example, if the travel time from detector 15→19→8→21(inner lane), or from 16→20→9→22(outer lane), is going to be estimated, one could either (1) add the results from each pair of consecutive detectors together, or (2) ignore the middle detectors and use detectors at the start and end as the UD and DD respectively, then apply the method suggested by this thesis.

The following part of this section will briefly describe these two approaches:

1. When the first approach is used, travel time estimations obtained by the LPRF method shall be used. Take the outer lane for instance, first reidentification takes place between detectors 9 and 22 and the travel time is estimated by LPRF. At a time point $t1$, the travel time is estimated as $tt1$, meaning that a vehicle arrived at detector 9 at time $t1 - tt1$ and arrived at detector 22 at $t1$. Then at time point $t1 - tt1$ the travel time from detector 20 to 9 is estimated the same way. This process goes on to the first pair of detectors 16 and 20. Finally, the corresponding travel times of the three parts are added together to obtain one travel time between detector 16 and 22.

2. When the second approach is used, the parameter $Vmin = 4$ for deciding time windows of large vehicles will not work. Actually, according to the simulated data, even during the congested period, the minimum link velocity is much larger than 4m/s, about 20m/s or larger. When the algorithm of this thesis is applied on two consecutive detectors, whose distance in between is not very big, the parameter $Vmin = 4$ can well reidentify large vehicles. But when the algorithm is applied on two detectors whose distance in between is nearly as big as tripled, as the way in the second approach, using $Vmin = 4$ will hardly reidentify any large vehicles correctly, resulting in totally bad reidentifications. Under this circumstance, $Vmin$ is dynamically determined by the harmonic mean velocity of the vehicles detected, i.e. $Vmin = 0.6H(V.detect)$, where $H(\cdot)$ stands for the harmonic mean[4], $V.detect$ stands for the velocities detected by the downstream detector.

[4]Definition of harmonic mean is given in Section 4.2 by Equation (4.1).

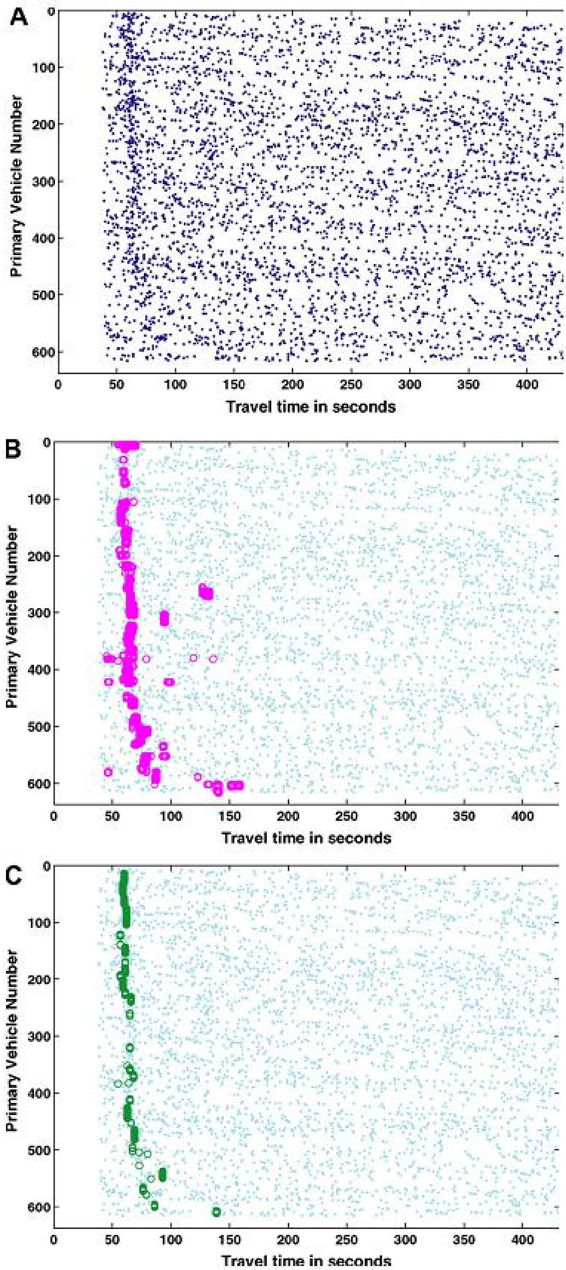

Figure 3.1: (A) Original TTM for a sample data set from two detectors almost one mile apart, dots stands for 1's in the TTM, blank area stands for 0's in TTM, (B) MDM superimposed on the TTM from A, (C) MPTT after finding the unique matches superimposed on the same TTM.

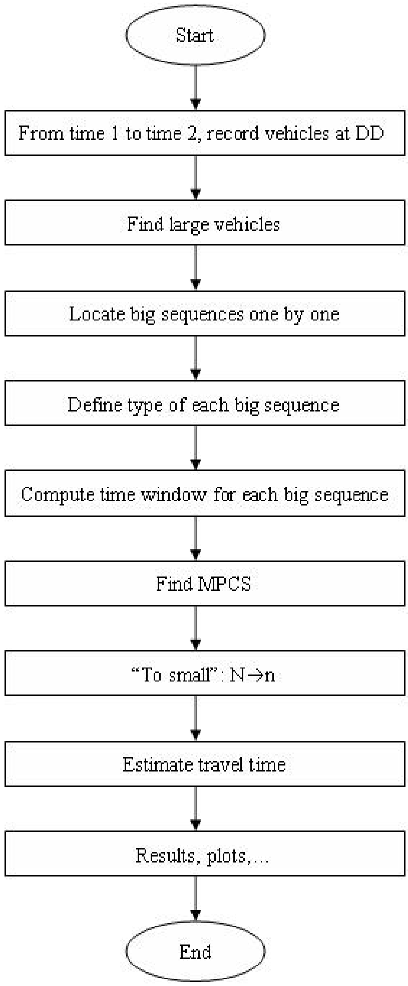

Figure 3.2: Flow chart of the algorithm.

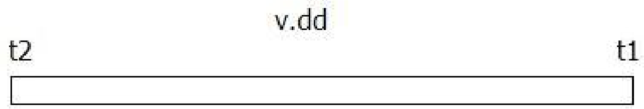

Figure 3.3: A visual expression of vehicles recorded at a DD. It means vehicles $v.dd$ are recorded from time $t1$ to $t2$. The big rectangular means there are a sequence of vehicles.

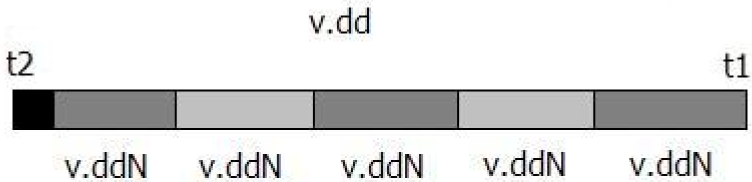

Figure 3.4: A visual expression of dividing $v.dd$ into a series of $v.ddN$'s. The black tail means that there are less than N vehicles left, and they are neglected.

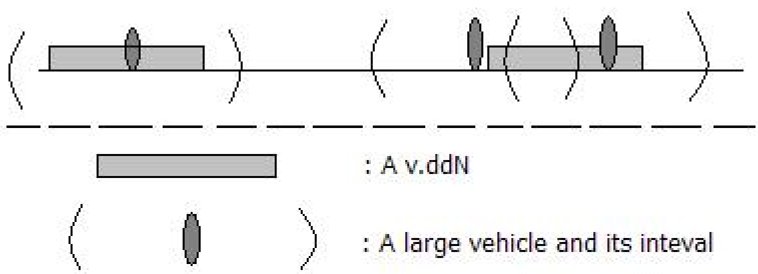

Figure 3.5: A visual expression of a type 1 situation. Legends are denoted below the dashed line.

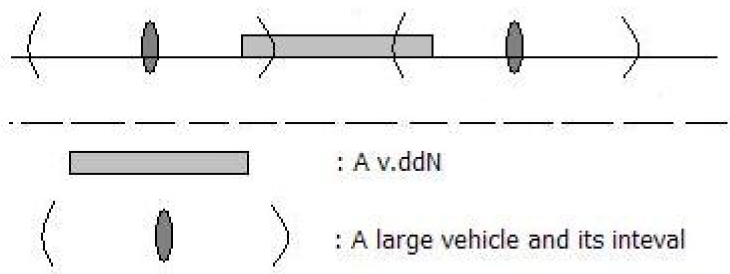

Figure 3.6: A visual expression of a type 2 situation. Legends are denoted below the dashed line.

Chapter 4

Results

In this section, the results of implementing the algorithm on the simulated data are displayed and discussed.

Before showing the results, some notations used by figures are given first:

- Black Circles: The real travel time of vehicles;
- Black Lines: Lines connecting the black circles;
- Green Dots: Vehicles reidentified in type 1 situation;
- Yellow Dots: Vehicles reidentified in type 2 situation
- Red Dots: Vehicles reidentified in type 3 situation;
- Blue Dots: The reidentified large vehicles;
- Pink Dots: The modified travel time estimation by the EWMA method;
- Pink Lines: Lines connecting the pink dots.
- Red Square With Cross Inside: The travel time estimated by the harmonic mean[1].

[1] The harmonic mean is used as a benchmark, which will be described in Section 4.2

4.1 Results Under Congested Time Period

In this section travel time estimated during the congested traffic, i.e. from time 9000(s) to 13000(s), at every pair of consecutive detectors are presented. This section consists of two parts: the first part will show some randomly selected results of ten-minute reidentifications, whose plots are easier and clearer to read than the other part; the second part will show all results during the whole congested time period.

4.1.1 Some Results of Ten-Minute Reidentifications

As mentioned previously, the algorithm is designed for real time traffic surveillance, and implemented every 10 minutes to reidentify vehicles. In this part, one randomly selected ten-minute reidentification for each pair of consecutive detectors is going to be displayed and discussed.

In the statistical tables, the item "Total No. of vehicles downstream" is the total number of vehicles that passed both the upstream and downstream detectors in the corresponding time period; the item "Match rate for sequences" shows the rate of real matches of each big sequence($v.ddN$) and each corresponding small sequence($v.ddn$). Correspondingly, visual results of statistical tables are given by a series of figures. A point in the figures shows a travel time at a time point, for instance, at a time point $t1$, the travel time is estimated as $tt1$, meaning that a vehicle arrived at upstream detector at time $t1 - tt1$ and arrived at downstream detector at $t1$. Each table is corresponded to two figures: one only shows the reidentification results; the other shows results of LPRF and EWMA as well.

It can be seen from the following six statistical tables(4.1 to 4.6) and twelve figures(4.1 to 4.12) that the algorithm for vehicle reidentification works pretty well. The algorithm is designed to reidentify 5 out every 20 vehicles(about 25%), and the outcome shows the correct reidentification rate is mostly above 70%. Especially, the rate of correct large vehicle reidentifications are much higher, mostly above 90%, which means the large vehicle information obtained by this algorithm is very reliable. In the six statistical tables, match rates for most of the big sequences($v.ddN$) are already not low. And based on this, the algorithm searched further to reidentify the corresponding small sequences($v.ddn$), resulting in a higher match rate.

Figures show that simple travel time estimation obtained directly from the vehicle reidentification results is acceptable. Figures with travel time estimations obtained by LPRF and EWMA show that these two ways of travel time estimations have their advantages as well as price. A quick summary of these three ways of travel time estimations is given in Table 4.7.

	Detector 15 to 19(367m)							
Total No. of vehicles downstream	311							
Total No. of sequences $v.ddN/v.ddn$	15							
No. of large vehicles reidentified	16							
No. of large vehicles reidentified correctly	15							
% of correct large vehicle reidentifications	94							
No. of vehicles to be reidentified	75							
No. of correct reidentifications	59							
% of correct reidentification	79							
Match rate(%) for sequences(1 to 8)	1	2	3	4	5	6	7	8
$v.ddN$	85	50	55	50	75	75	90	65
$v.ddn$	100	100	100	100	60	80	80	80
Match rate(%) for sequences(9 to 15)	9	10	11	12	13	14	15	
$v.ddN$	75	65	70	80	85	65	55	
$v.ddn$	100	0	0	100	100	100	80	

Table 4.1: Ten-minute Reidentification results: Detector 15 to 19(367m), from time 10200(s) to time 10800(s). Corresponding visual results are given by Figure 4.1 and 4.2.

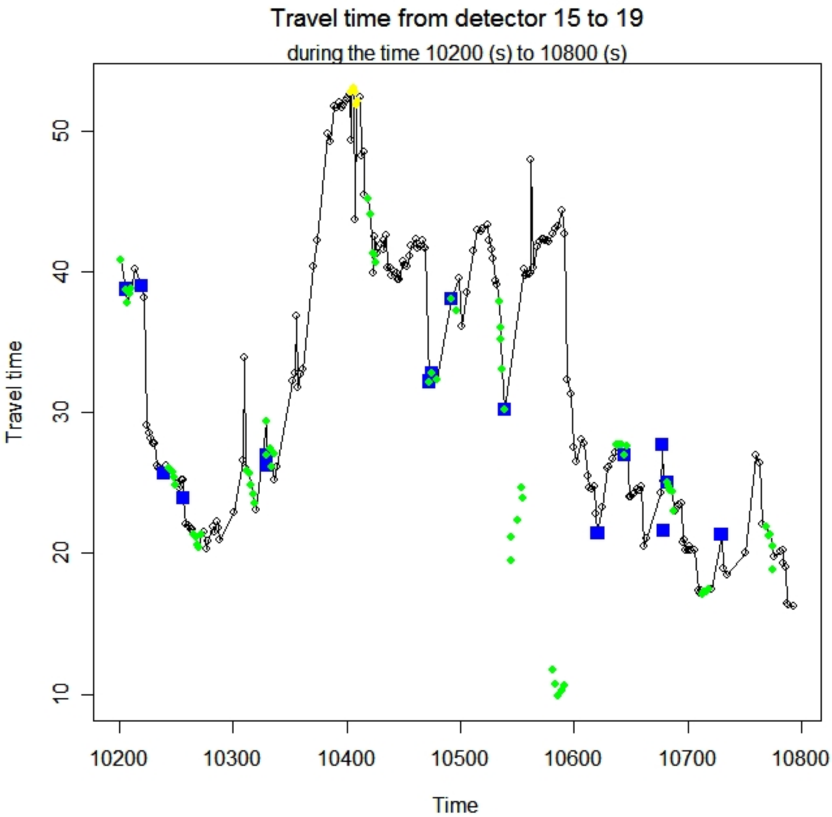

Figure 4.1: Results of vehicle reidentification between detector 15 to 19, from time 10200(s) to time 10800(s). Corresponding statistics are given by Table 4.1.

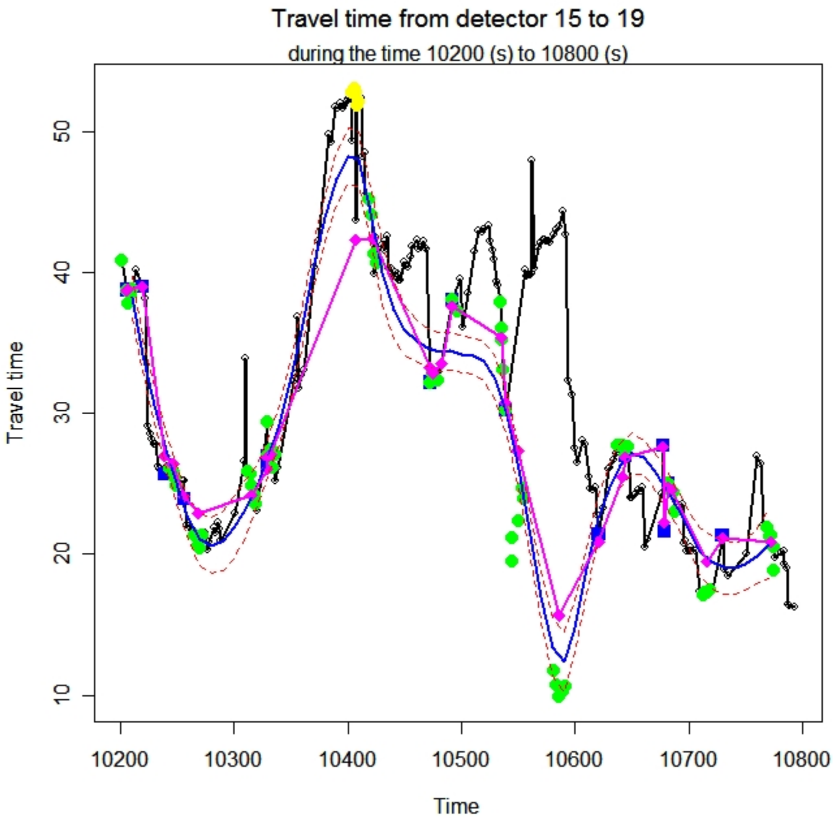

Figure 4.2: Results of vehicle reidentification between detector 15 to 19, from time 10200(s) to time 10800(s), with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve, with dashed red lines showing plus/minus estimated standard errors). Corresponding statistics are given by Table 4.1.

	Detector 16 to 20(367m)							
Total No. of vehicles downstream	314							
Total No. of sequences $v.ddN/v.ddn$	15							
No. of large vehicles reidentified	3							
No. of large vehicles reidentified correctly	3							
% of correct large vehicle reidentifications	100							
No. of vehicles to be reidentified	75							
No. of correct reidentifications	56							
% of correct reidentification	75							
Match rate(%) for sequences(1 to 8)	1	2	3	4	5	6	7	8
$v.ddN$	70	75	55	80	100	70	55	45
$v.ddn$	100	100	100	100	100	100	100	40
Match rate(%) for sequences(9 to 15)	9	10	11	12	13	14	15	
$v.ddN$	95	35	35	0	0	20	25	
$v.ddn$	100	80	100	0	0	0	100	

Table 4.2: Ten-minute Reidentification results: Detector 16 to 20(367m), from time 10200(s) to time 10800(s). Corresponding visual results are given by Figure 4.3 and 4.4.

	Detector 20 to 9(1150m)								
Total No. of vehicles downstream	348								
Total No. of sequences $v.ddN/v.ddn$	17								
No. of large vehicles reidentified	3								
No. of large vehicles reidentified correctly	3								
% of correct large vehicle reidentifications	100								
No. of vehicles to be reidentified	85								
No. of correct reidentifications	66								
% of correct reidentification	78								
Match rate(%) for sequences(1 to 9)	1	2	3	4	5	6	7	8	9
$v.ddN$	50	0	95	45	75	80	50	35	55
$v.ddn$	100	0	100	100	40	100	60	40	100
Match rate(%) for sequences(10 to 17)	10	11	12	13	14	15	16	17	
$v.ddN$	90	65	50	75	75	85	60	0	
$v.ddn$	100	100	80	100	100	100	100	0	

Table 4.3: Ten-minute Reidentification results: Detector 20 to 9(1150m), from time 10800(s) to time 11400(s). Corresponding visual results are given by Figure 4.5 and 4.6.

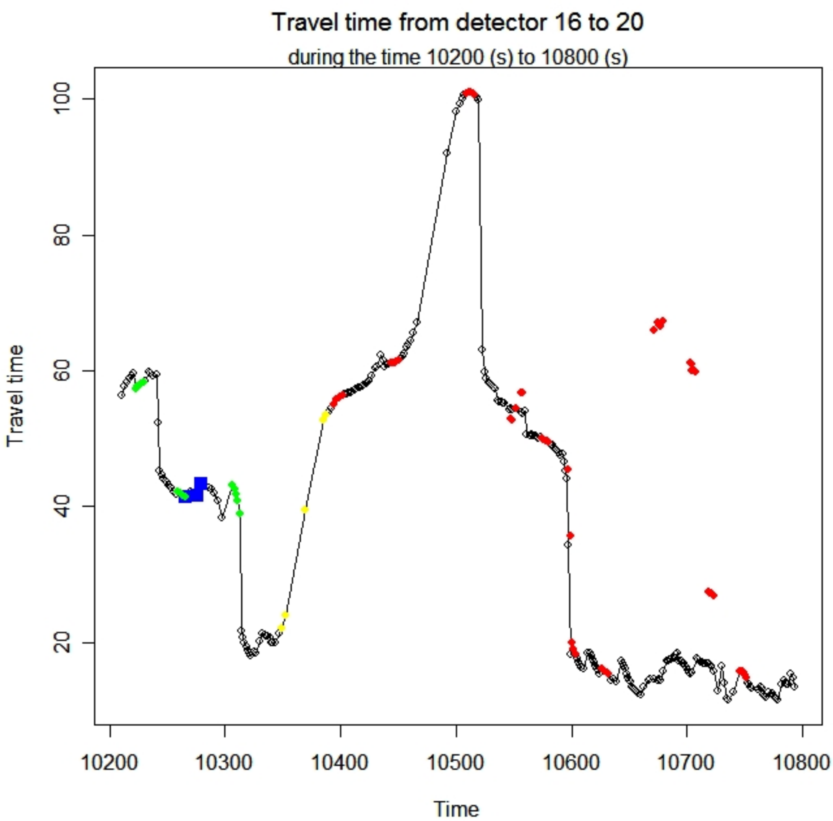

Figure 4.3: Results of vehicle reidentification between detector 16 to 20, from time 10200(s) to time 10800(s). Corresponding statistics are given by Table 4.2.

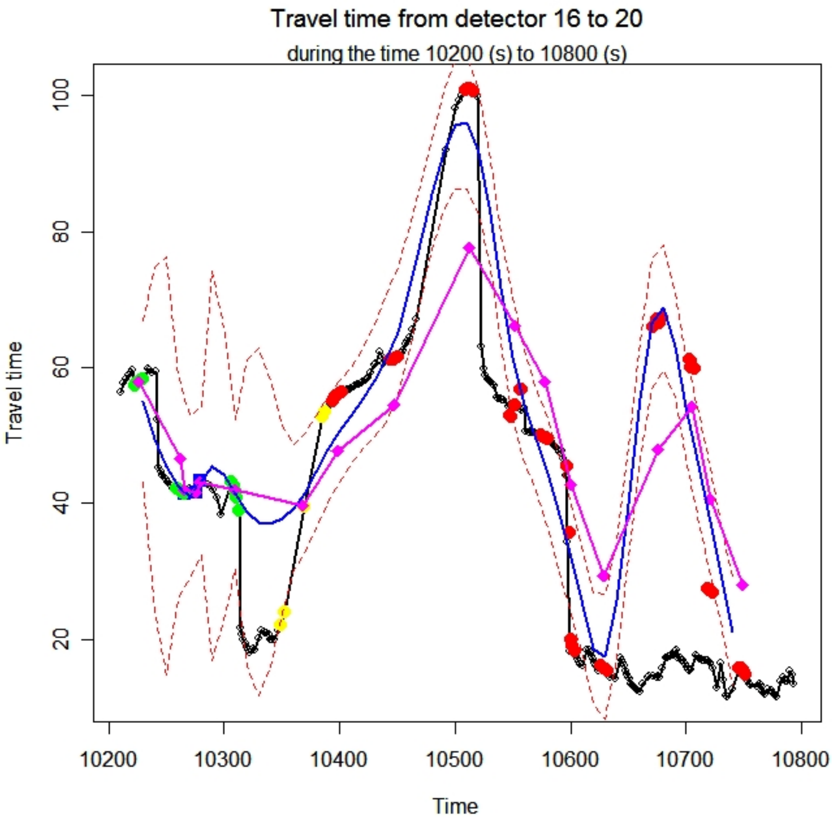

Figure 4.4: Results of vehicle reidentification between detector 16 to 20, from time 10200(s) to time 10800(s), with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve, with dashed red lines showing plus/minus estimated standard errors). Corresponding statistics are given by Table 4.2.

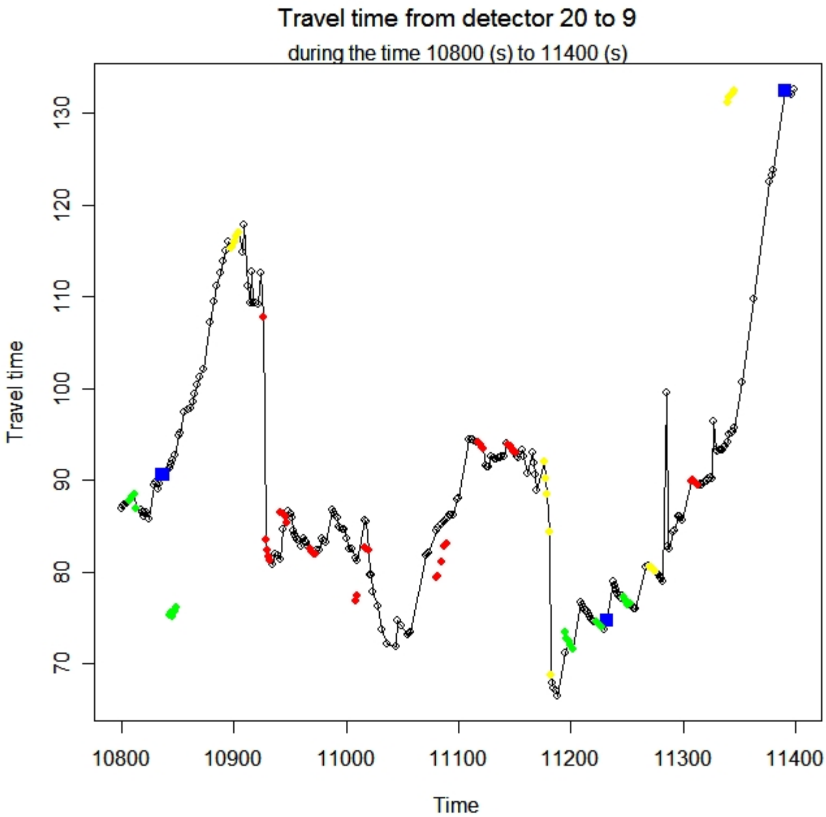

Figure 4.5: Results of vehicle reidentification between detector 20 to 9, from time 10800(s) to time 11400(s). Corresponding statistics are given by Table 4.3.

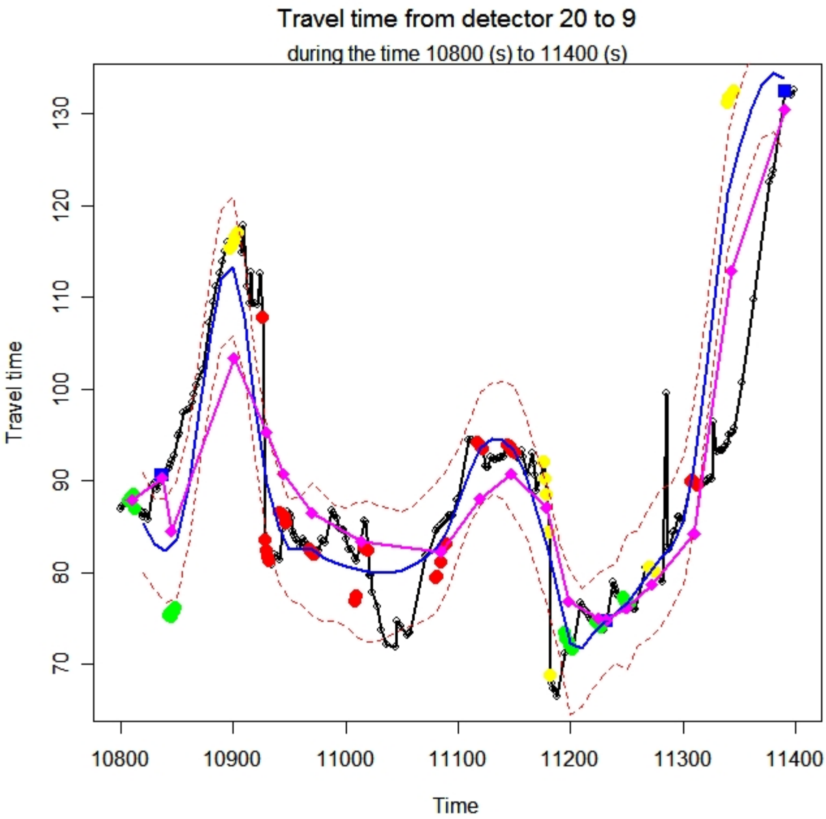

Figure 4.6: Results of vehicle reidentification between detector 20 to 9, from time 10800(s) to time 11400(s), with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve, with dashed red lines showing plus/minus estimated standard errors). Corresponding statistics are given by Table 4.3.

	Detector 19 to 8(1150m)						
Total No. of vehicles downstream	262						
Total No. of sequences $v.ddN/v.ddn$	13						
No. of large vehicles reidentified	18						
No. of large vehicles reidentified correctly	16						
% of correct large vehicle reidentifications	89						
No. of vehicles to be reidentified	65						
No. of correct reidentifications	47						
% of correct reidentification	72						
Match rate(%) for sequences(1 to 7)	1	2	3	4	5	6	7
$v.ddN$	80	70	80	85	75	0	0
$v.ddn$	100	80	100	100	80	0	0
Match rate(%) for sequences(8 to 13)	8	9	10	11	12	13	
$v.ddN$	60	55	85	75	65	70	
$v.ddn$	80	80	100	80	40	100	

Table 4.4: Ten-minute Reidentification results: Detector 19 to 8(1150m), from time 12000(s) to time 12600(s). Corresponding visual results are given by Figure 4.7 and 4.8.

	Detector 9 to 22(889m)								
Total No. of vehicles downstream	365								
Total No. of sequences $v.ddN/v.ddn$	18								
No. of large vehicles reidentified	2								
No. of large vehicles reidentified correctly	2								
% of correct large vehicle reidentifications	100								
No. of vehicles to be reidentified	90								
No. of correct reidentifications	67								
% of correct reidentification	74								
Match rate(%) for sequences(1 to 9)	1	2	3	4	5	6	7	8	9
$v.ddN$	55	85	0	70	75	80	55	85	90
$v.ddn$	100	100	0	100	80	100	0	0	100
Match rate(%) for sequences(10 to 17)	10	11	12	13	14	15	16	17	18
$v.ddN$	60	95	90	55	60	90	20	80	95
$v.ddn$	80	100	100	80	100	100	0	100	100

Table 4.5: Ten-minute Reidentification results: Detector 9 to 22(889m), from time 9600(s) to time 10200(s). Corresponding visual results are given by Figure 4.9 and 4.10.

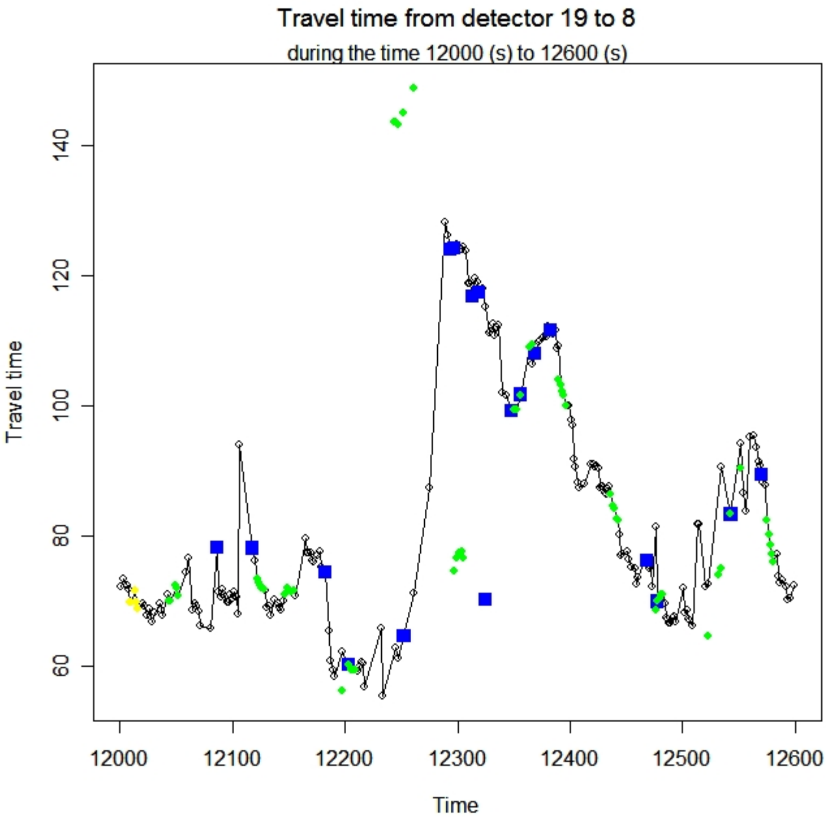

Figure 4.7: Results of vehicle reidentification between detector 19 to 8, from time 12000(s) to time 12600(s). Corresponding statistics are given by Table 4.4.

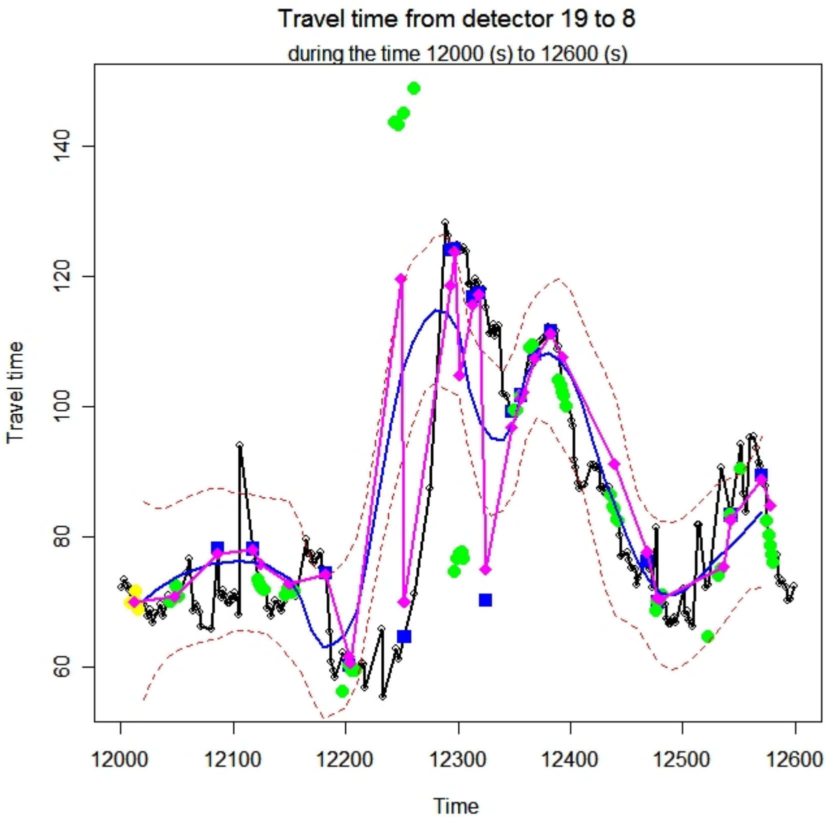

Figure 4.8: Results of vehicle reidentification between detector 19 to 8, from time 12000(s) to time 12600(s), with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve, with dashed red lines showing plus/minus estimated standard errors). Corresponding statistics are given by Table 4.4.

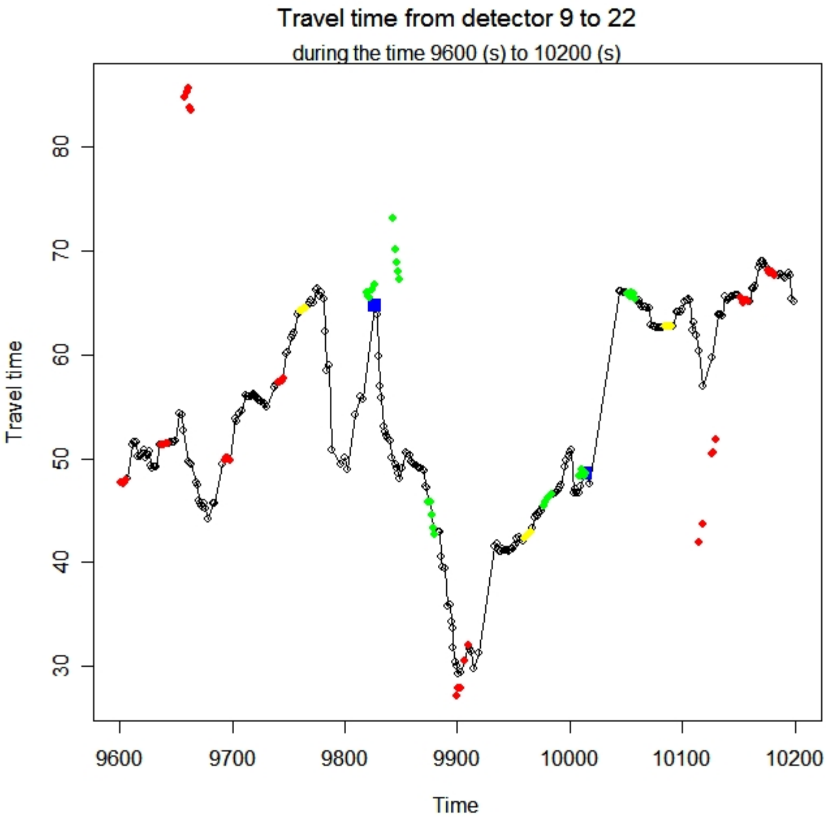

Figure 4.9: Results of vehicle reidentification between detector 9 to 22, from time 9600(s) to time 10200(s). Corresponding statistics are given by Table 4.5.

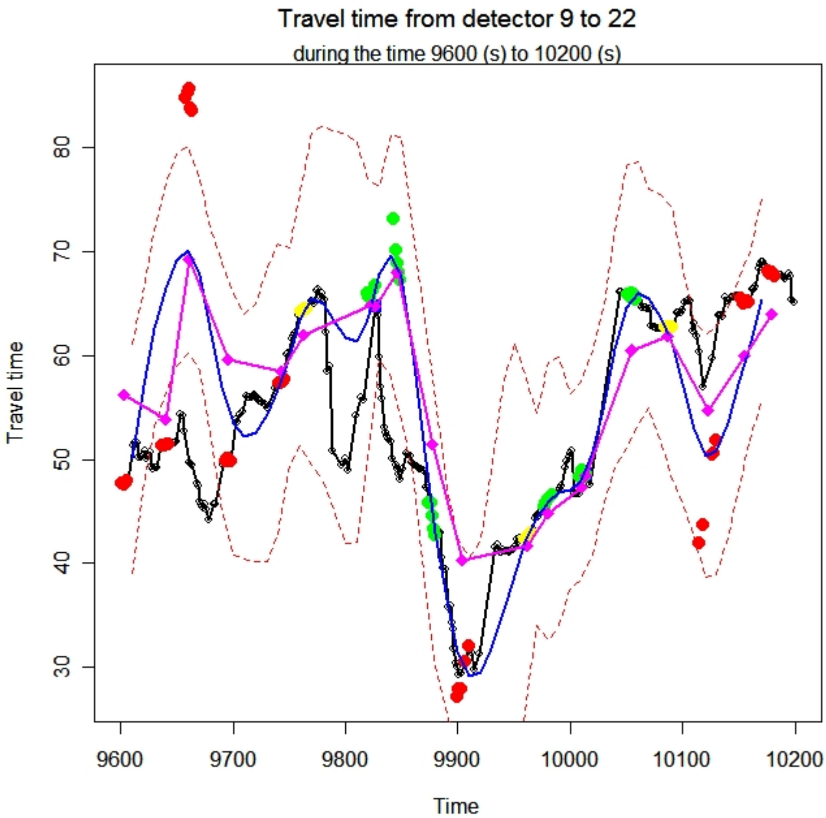

Figure 4.10: Results of vehicle reidentification between detector 9 to 22, from time 9600(s) to time 10200(s), with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve, with dashed red lines showing plus/minus estimated standard errors). Corresponding statistics are given by Table 4.5.

	Detector 8 to 21(889m)						
Total No. of vehicles downstream	277						
Total No. of sequences $v.ddN/v.ddn$	13						
No. of large vehicles reidentified	17						
No. of large vehicles reidentified correctly	16						
% of correct large vehicle reidentifications	94						
No. of vehicles to be reidentified	65						
No. of correct reidentifications	55						
% of correct reidentification	85						
Match rate(%) for sequences(1 to 7)	1	2	3	4	5	6	7
$v.ddN$	85	30	80	75	75	75	80
$v.ddn$	100	0	100	100	80	80	100
Match rate(%) for sequences(8 to 13)	8	9	10	11	12	13	
$v.ddN$	45	35	100	65	90	70	
$v.ddn$	60	100	100	80	100	100	

Table 4.6: Ten-minute Reidentification results: Detector 8 to 21(889m), from time 12000(s) to time 12600(s). Corresponding visual results are given by Figure 4.11 and 4.12.

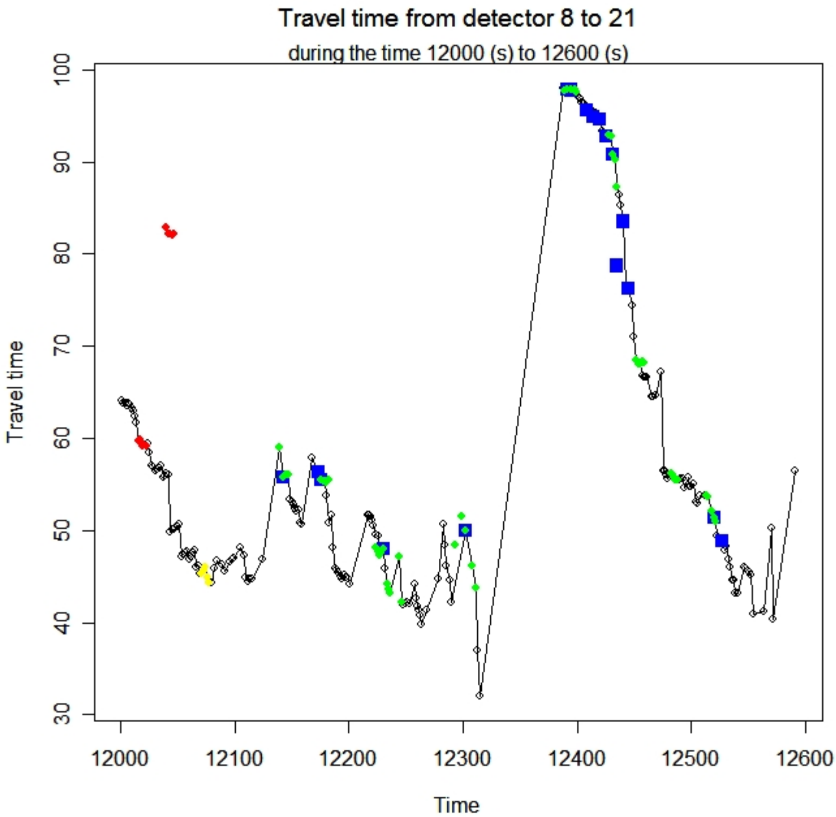

Figure 4.11: Results of vehicle reidentification between detector 8 to 21, from time 12000(s) to time 12600(s). Corresponding statistics are given by Table 4.6.

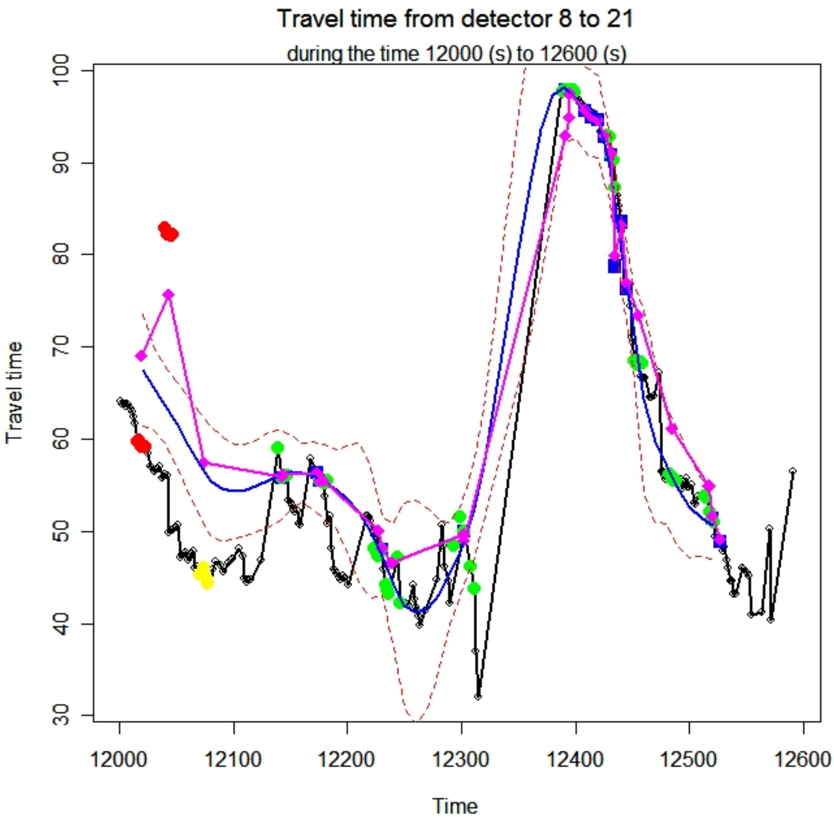

Figure 4.12: Results of vehicle reidentification between detector 8 to 21, from time 12000(s) to time 12600(s), with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve, with dashed red lines showing plus/minus estimated standard errors). Corresponding statistics are given by Table 4.6.

Method	Advantages	Disadvantages
Direct	Simple and quick, does not need further computations	Bad estimation when the reidentification is incorrect
LPRF	Provide travel time estimation in continuous time, and can reflect the travel time trend well most of the time.	Need further computations, and sometimes gives estimation errors even when the reidentification is correct.
EWMA	Provide a way to put different weights on different types of situations, and sometimes reduces estimation errors.	The results depend very much on the way of putting weights; and it sometimes increases the estimation errors.

Table 4.7: A summary of different ways for travel time estimations.

4.1.2 Results For The Whole Congested Time Period

To avoid the randomness that the algorithm only works under those selected time periods in the previous part, in this part, all important results for the whole congested time period are collected in Table 4.8. The corresponding plots are given by Figures 4.13 to 4.18. It can be seen, most of the reidentifications are correct, and most of the wrong reidentifications give travel time estimations that are very close to the real travel times. However, there are reidentifications that give travel time estimations far from true. But the large vehicles reidentified are almost correct. Even if there are wrong large vehicle reidentifications, they lie very close to real travel times. So the large vehicles information obtained by the algorithm is very reliable. Figures 4.19 to 4.24 show the reidentification results together with travel time estimations obtained by EWMA and LPRF.

Case	1	2	3	4	5	6
Detectors	$15 \rightarrow 19$	$16 \rightarrow 20$	$19 \rightarrow 8$	$20 \rightarrow 9$	$8 \rightarrow 21$	$9 \rightarrow 22$
Distance(meters)	367	367	1150	1150	889	889
Total No. of vehicles downstream	2028	2110	1769	2394	1784	2389
Total No. of sequences $v.ddN/v.ddn$	101	105	88	119	89	119
No. of large vehicles reidentified	67	42	91	29	87	25
No. of large vehicles reidentified correctly	66	39	86	28	86	24
% of correct large vehicle reidentifications	99	93	95	97	99	96
No. of vehicles to be reidentified	505	525	440	595	445	595
No. of correct reidentifications	284	454	303	434	349	448
% of correct reidentification	56	86	69	73	78	75

Table 4.8: Reidentification results during congested time period 9000(s) to 13000(s), about 67 minutes.

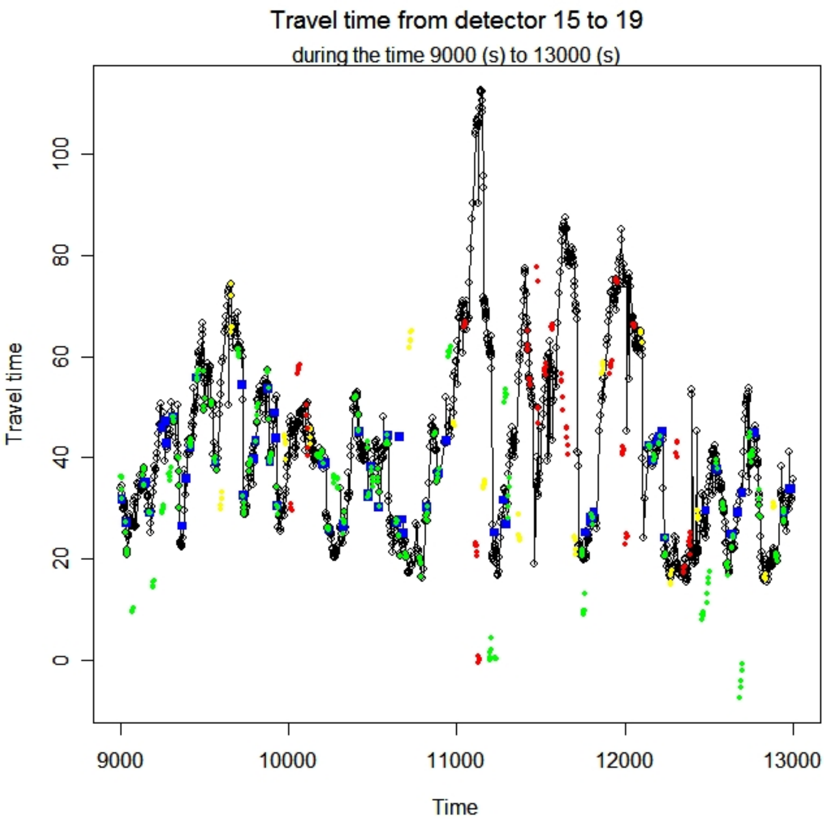

Figure 4.13: Results of vehicle reidentification from detector 15 to 19 during the whole congested period.

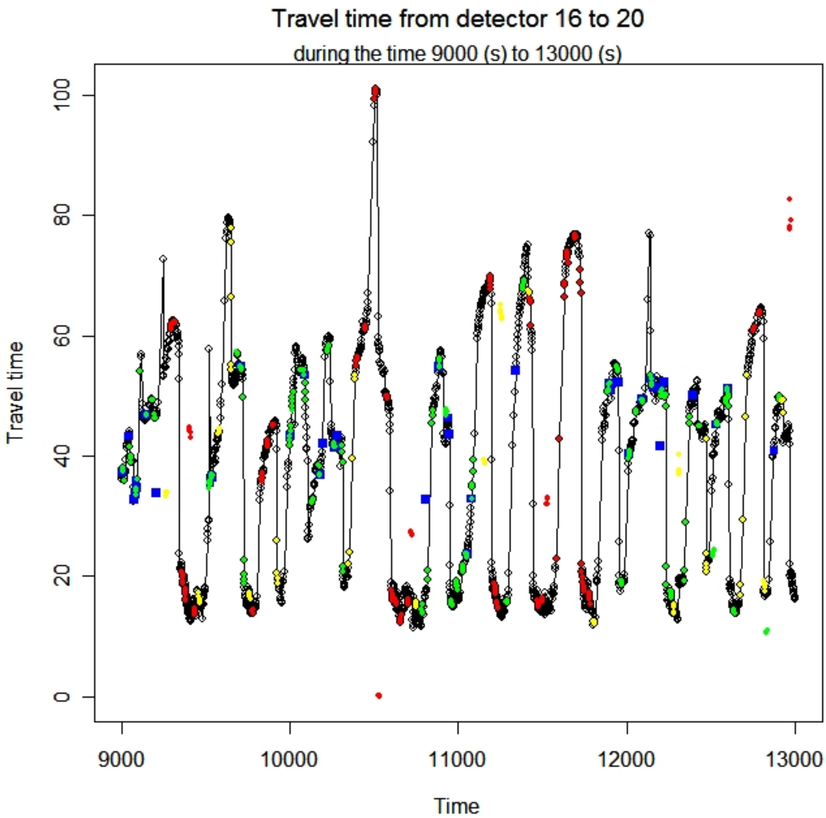

Figure 4.14: Results of vehicle reidentification from detector 16 to 20 during the whole congested period.

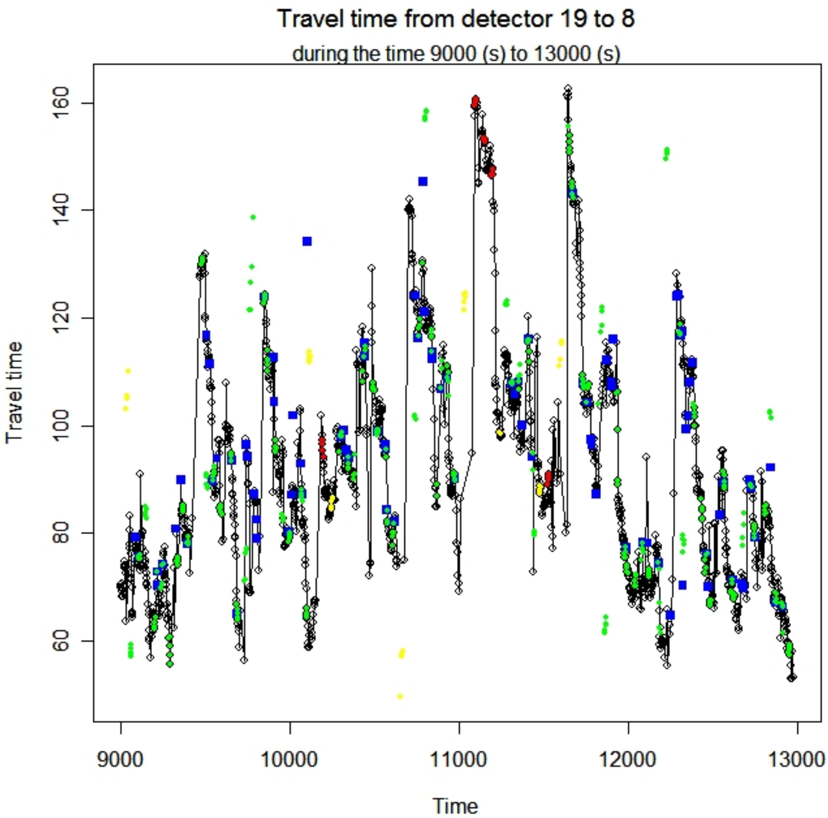

Figure 4.15: Results of vehicle reidentification from detector 19 to 8 during the whole congested period.

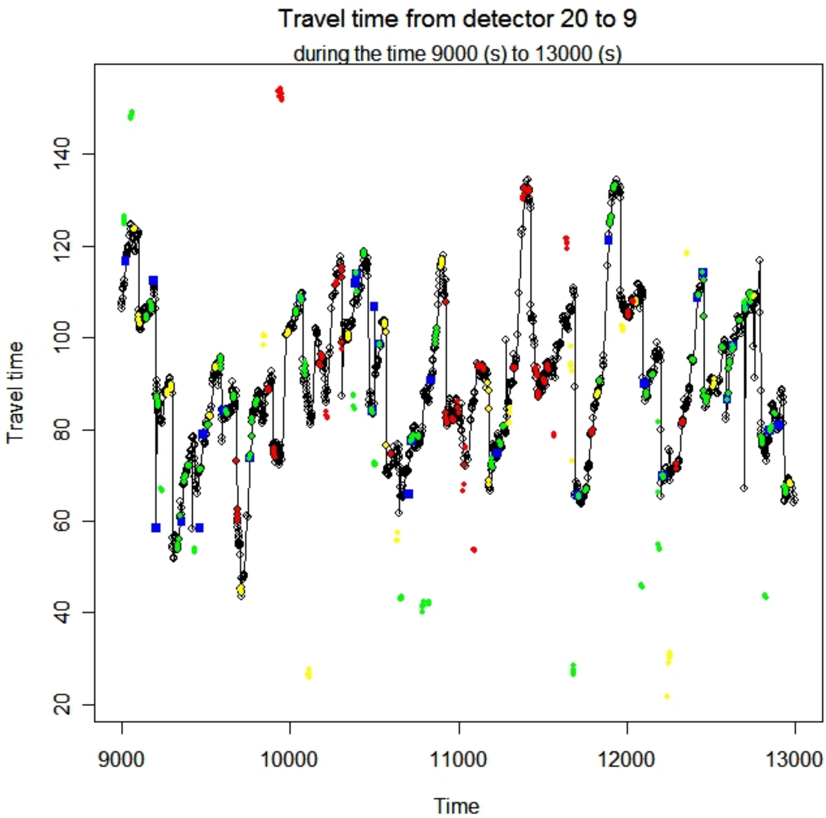

Figure 4.16: Results of vehicle reidentification from detector 20 to 9 during the whole congested period.

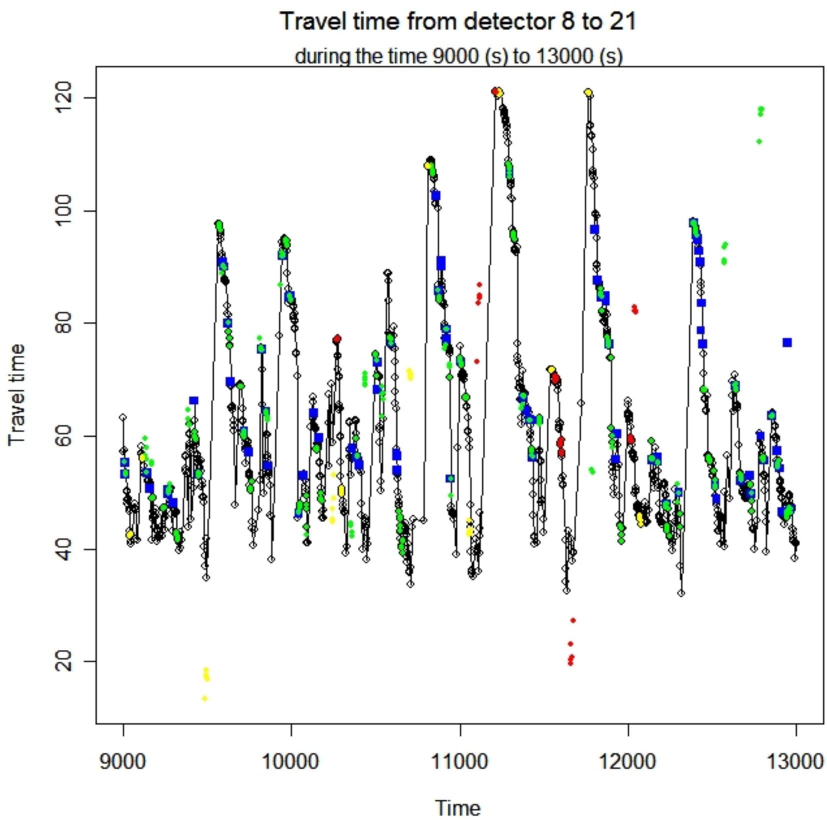

Figure 4.17: Results of vehicle reidentification from detector 8 to 21 during the whole congested period.

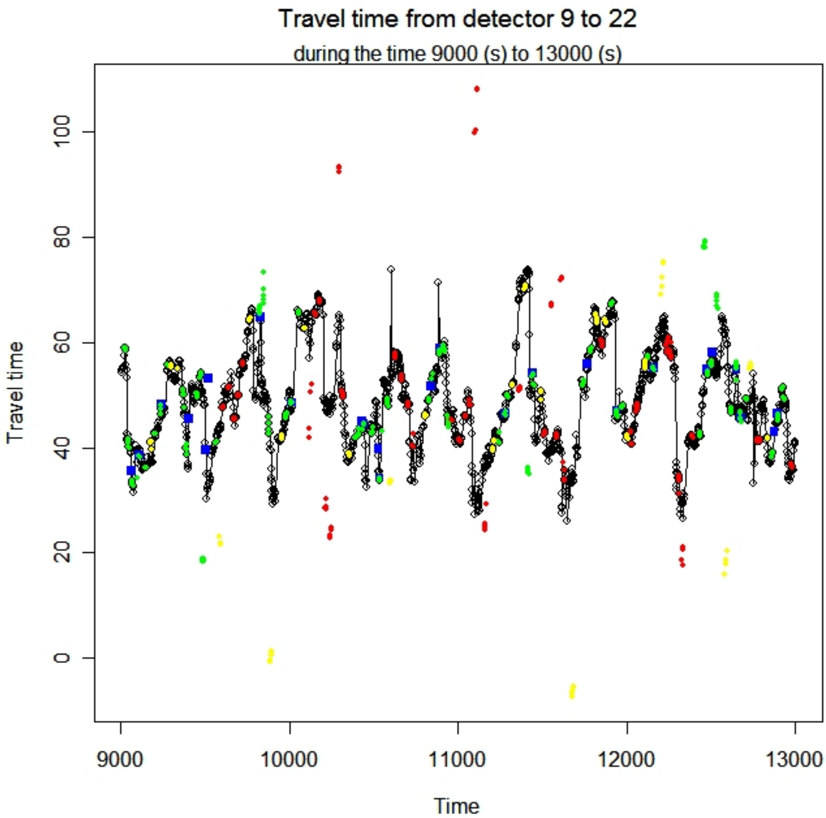

Figure 4.18: Results of vehicle reidentification from detector 9 to 22 during the whole congested period.

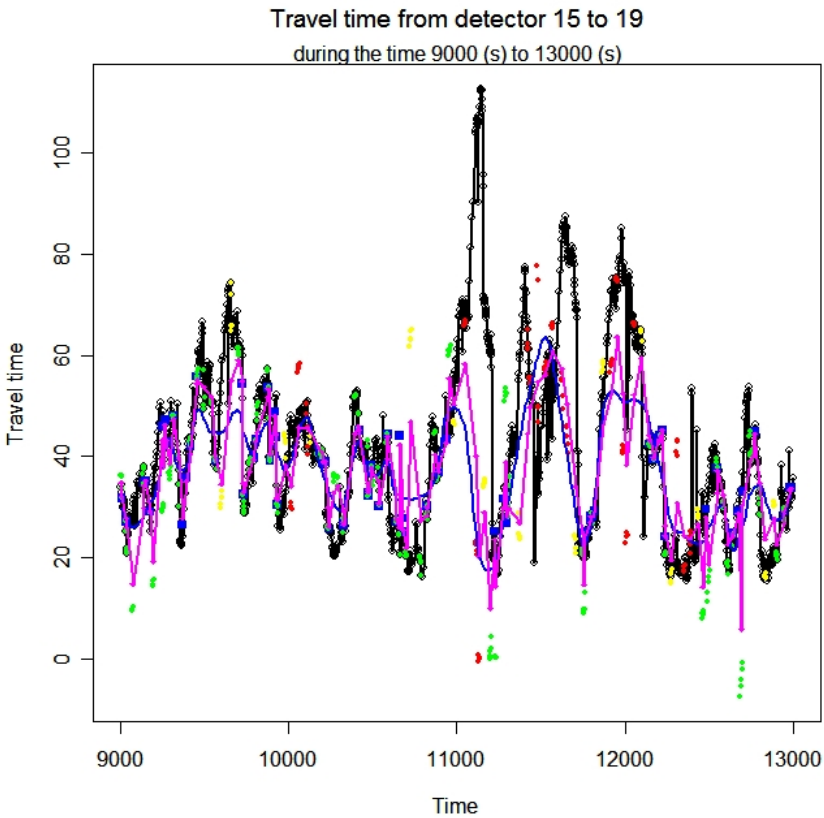

Figure 4.19: Results of vehicle reidentification from detector 15 to 19 during the whole congested period, with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve).

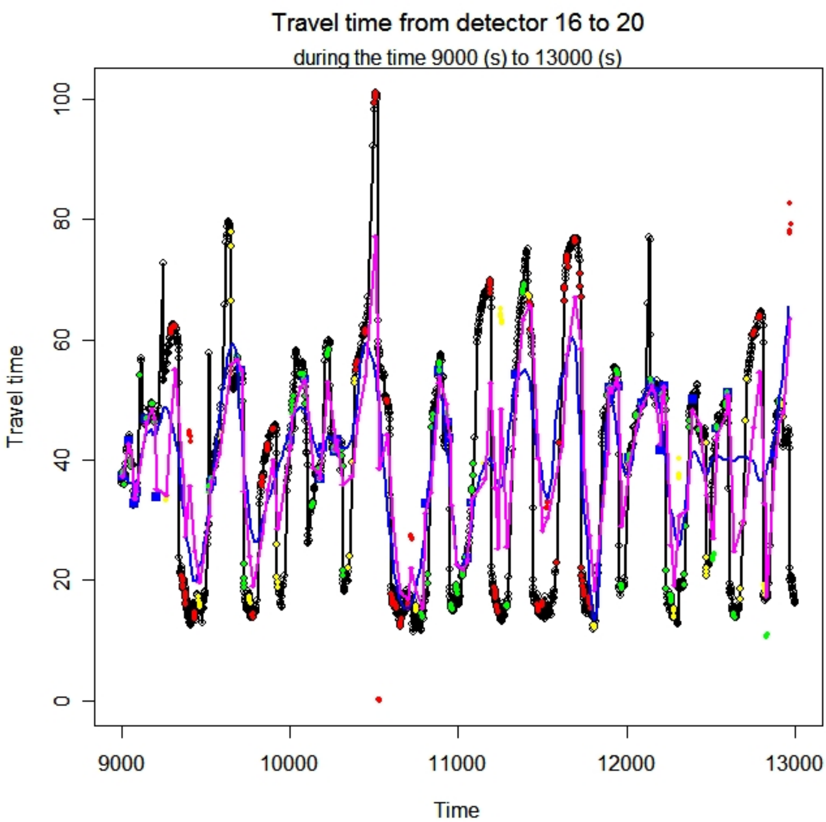

Figure 4.20: Results of vehicle reidentification from detector 16 to 20 during the whole congested period, with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve).

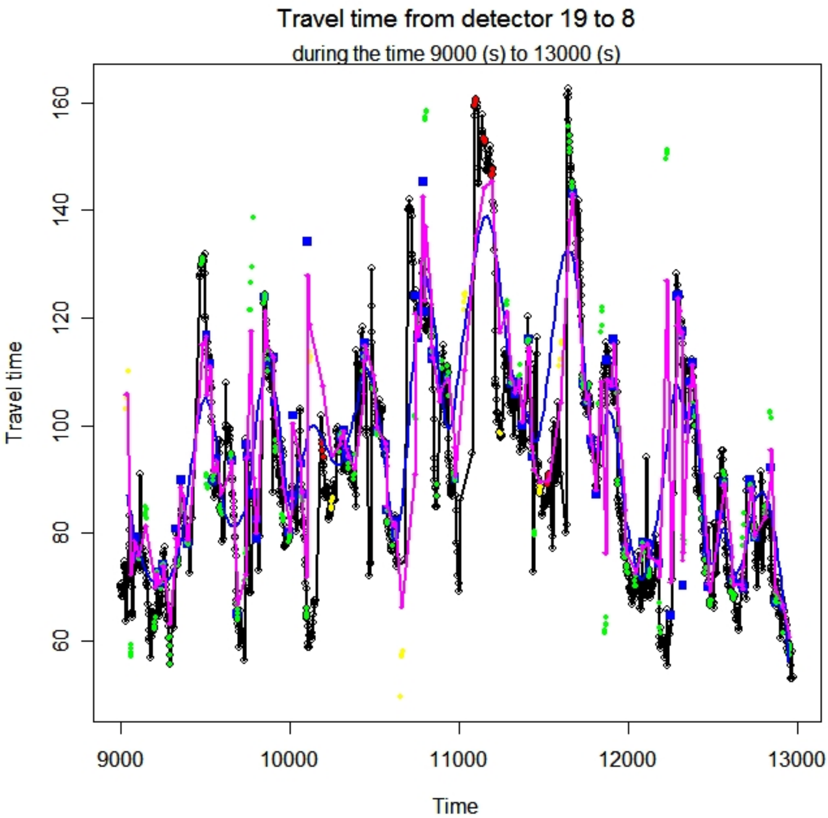

Figure 4.21: Results of vehicle reidentification from detector 19 to 8 during the whole congested period, with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve).

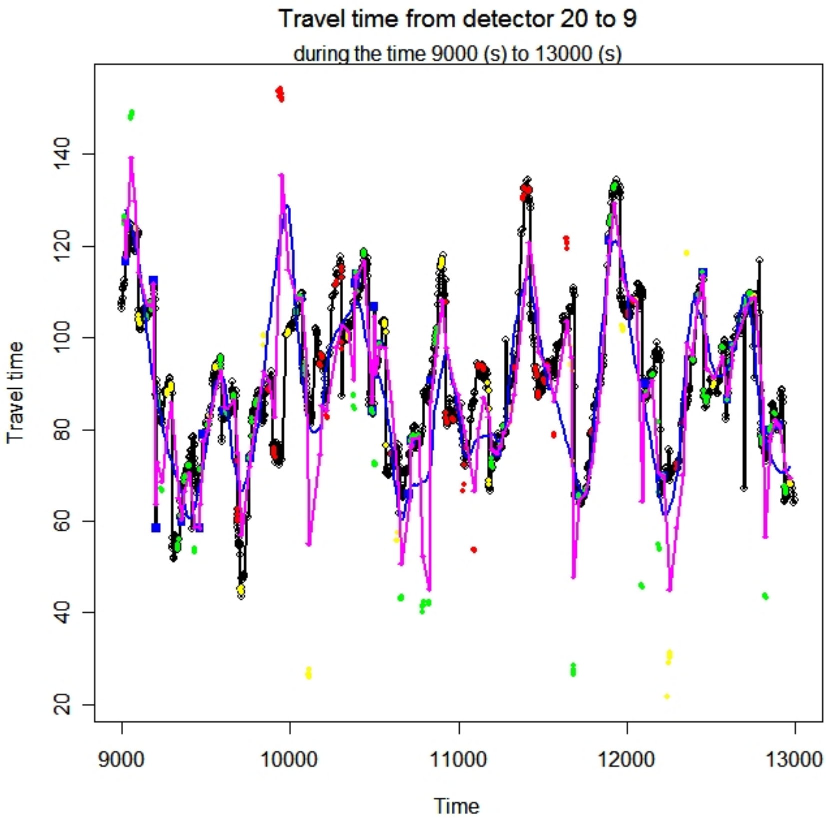

Figure 4.22: Results of vehicle reidentification from detector 20 to 9 during the whole congested period, with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve).

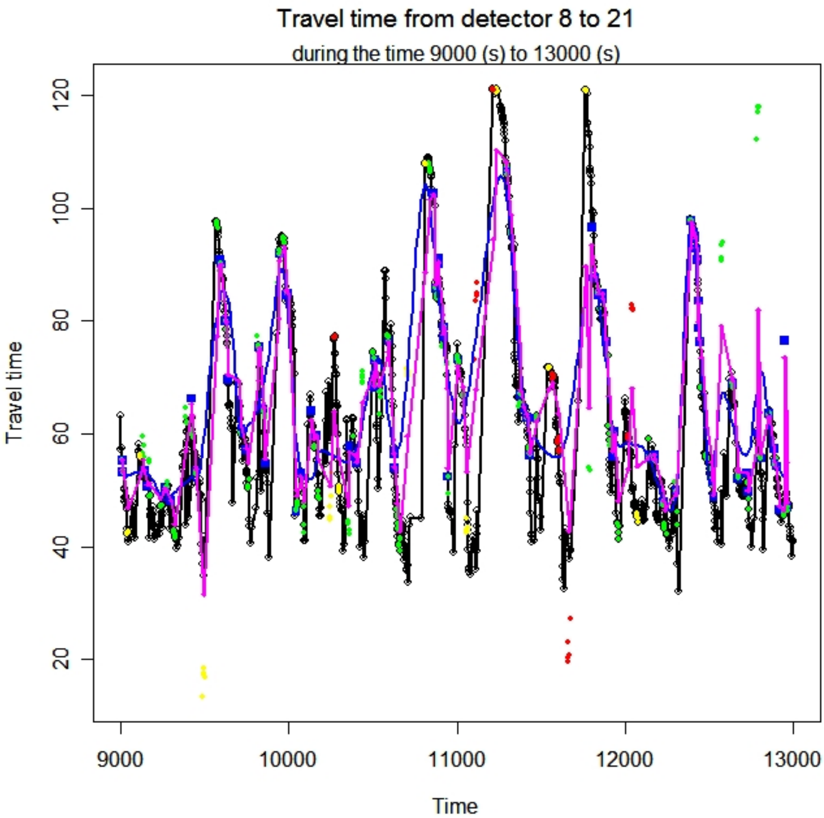

Figure 4.23: Results of vehicle reidentification from detector 8 to 21 during the whole congested period, with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve).

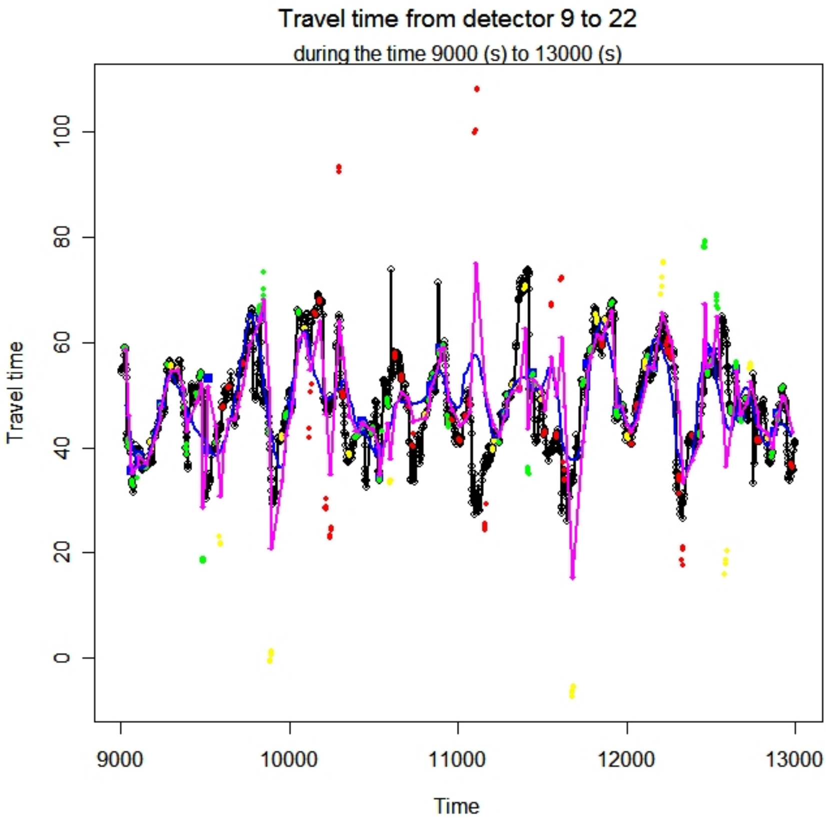

Figure 4.24: Results of vehicle reidentification from detector 9 to 22 during the whole congested period, with travel estimations obtained by EWMA(pink curve) and LPRF(blue curve).

4.2 Results of Benchmark

The LPRF travel estimation provided by this thesis is compared with the harmonic mean method, which is widely used.

4.2.1 Harmonic Mean Method

The harmonic mean for n numbers V_1, V_2, ..., V_n is denoted as:

$$H = \frac{n}{\frac{1}{V_1} + \frac{1}{V_2} + ... + \frac{1}{V_n}} \tag{4.1}$$

The harmonic mean method of estimating travel time is to first calculate the harmonic mean velocity of vehicles in a $v.ddN$, denoted by V_H; second calculate the mean time of detection of those N vehicles as the arrival time of that $v.ddN$, denoted by $v.ddN.te$; then the travel time, denoted by TT, of that $v.ddN$ at the time $v.ddN.te$ is estimated by

$$TT = \frac{Dist}{V_H} \tag{4.2}$$

where $Dist$ is the distance between the two detectors.

4.2.2 Comparison Results

In this section, the LPRF travel time estimation results showed at Section 4.1.1 are compared with the harmonic mean method. Some notations used in figures are explained first:

- Red Square with Cross inside: harmonic mean estimations;
- Red Curve: harmonic mean estimations fitted by LPRF;
- Blue Curve: travel time estimations obtained by the LPRF fitted vehicle reidentifications.

The comparison results are given by Table 4.9 to 4.14 and their corresponding Figures 4.25 to 4.30. The item "Bias" in tables is defined by the mean of travel time estimations minus the mean of real travel time.

It can be seen that, travel time estimation obtained by the LPRF fitted vehicle reidentifications has smaller bias, sum of squared errors and error variance than the harmonic mean method. And from the figures visual results, it can be seen that LPRF curve of the vehicle reidentifications reflects the real travel time trend much better than the harmonic mean curve does.

Detector 15 to 19	Reidentification	Harmonic Mean
Bias	-3.695092	4.087084
Sum of Squared Errors	15821.26	63880.73
Variance of Errors	60.91046	284.5411

Table 4.9: Benchmark results: Detector 15 to 19(367m), from time 10200(s) to time 10800(s). Corresponding visual results are given by Figure 4.25.

Detector 16 to 20	Reidentification	Harmonic Mean
Bias	10.48342	-15.60565
Sum of Squared Errors	93114.39	177989.0
Variance of Errors	271.2638	484.9285

Table 4.10: Benchmark results: Detector 16 to 20(367m), from time 10200(s) to time 10800(s). Corresponding visual results are given by Figure 4.26.

Detector 20 to 9	Reidentification	Harmonic Mean
Bias	0.3733646	-3.823991
Sum of Squared Errors	14161.63	709471.7
Variance of Errors	55.83486	2789.555

Table 4.11: Benchmark results: Detector 20 to 9(1150m), from time 10800(s) to time 11400(s). Corresponding visual results are given by Figure 4.27.

Detector 19 to 8	Reidentification	Harmonic Mean
Bias	0.4189383	-4.902548
Sum of Squared Errors	22931.58	72772.06
Variance of Errors	106.4822	314.3279

Table 4.12: Benchmark results: Detector 19 to 8(1150m), from time 12000(s) to time 12600(s). Corresponding visual results are given by Figure 4.28.

Detector 9 to 22	Reidentification	Harmonic Mean
Bias	2.071180	-5.737136
Sum of Squared Errors	20040.11	57815.32
Variance of Errors	65.52142	168.4177

Table 4.13: Benchmark results: Detector 9 to 22(889m), from time 9600(s) to time 10200(s). Corresponding visual results are given by Figure 4.29.

Detector 8 to 21	Reidentification	Harmonic Mean
Bias	3.170465	-8.72679
Sum of Squared Errors	7522.846	28653.28
Variance of Errors	28.4753	70.39248

Table 4.14: Benchmark results: Detector 8 to 21(889m), from time 12000(s) to time 12600(s). Corresponding visual results are given by Figure 4.30.

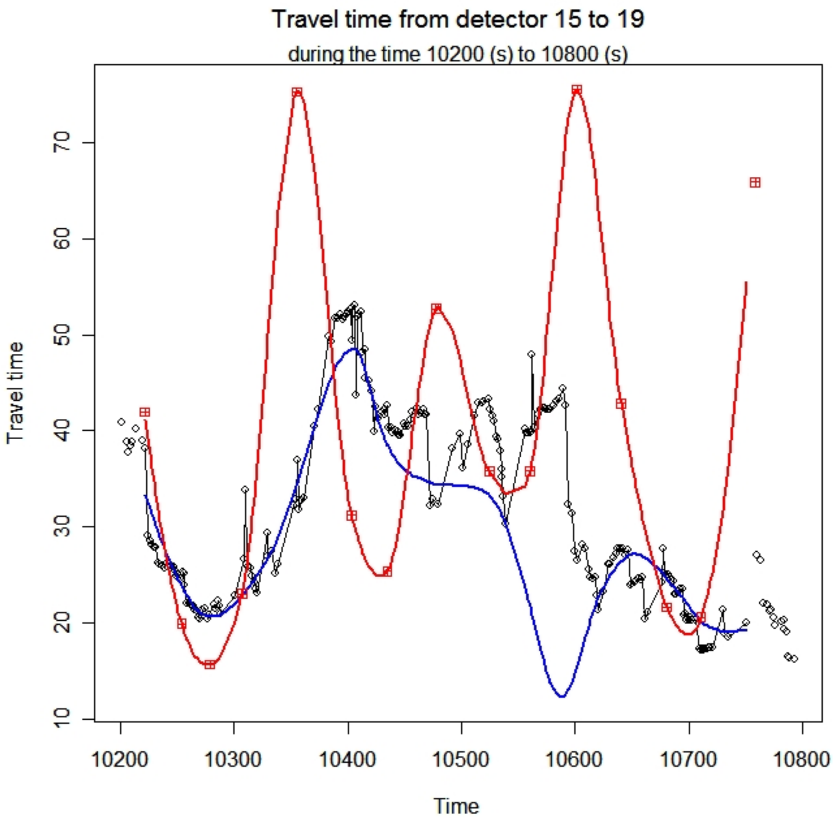

Figure 4.25: Benchmark Results of vehicle reidentification between detector 15 to 19, from time 10200(s) to time 10800(s). Corresponding statistics are given by Table 4.9.

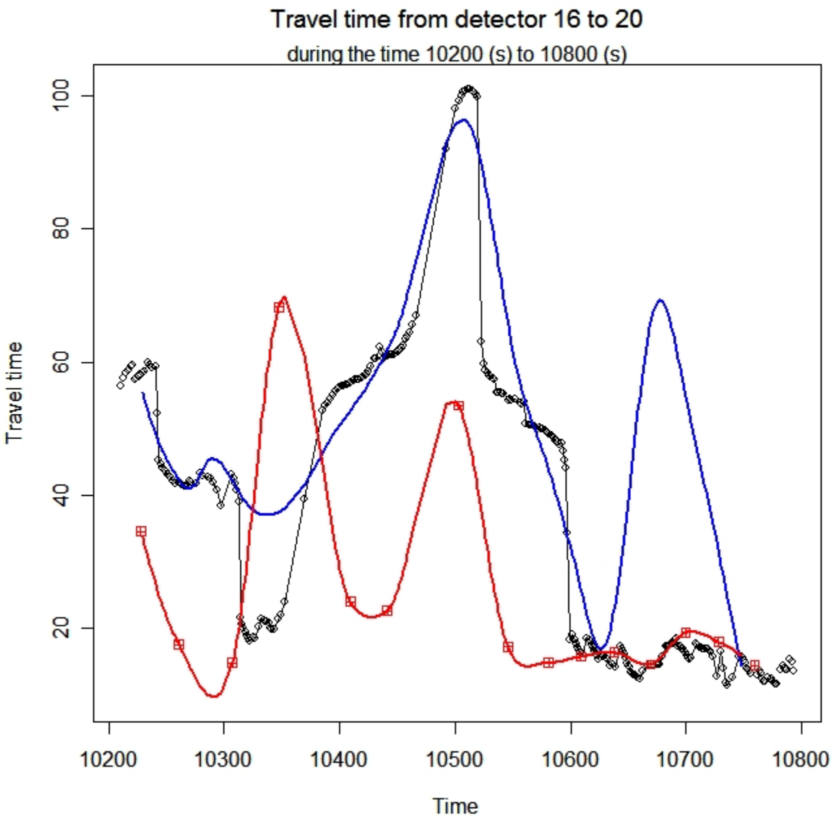

Figure 4.26: Benchmark Results of vehicle reidentification between detector 16 to 20, from time 10200(s) to time 10800(s). Corresponding statistics are given by Table 4.10.

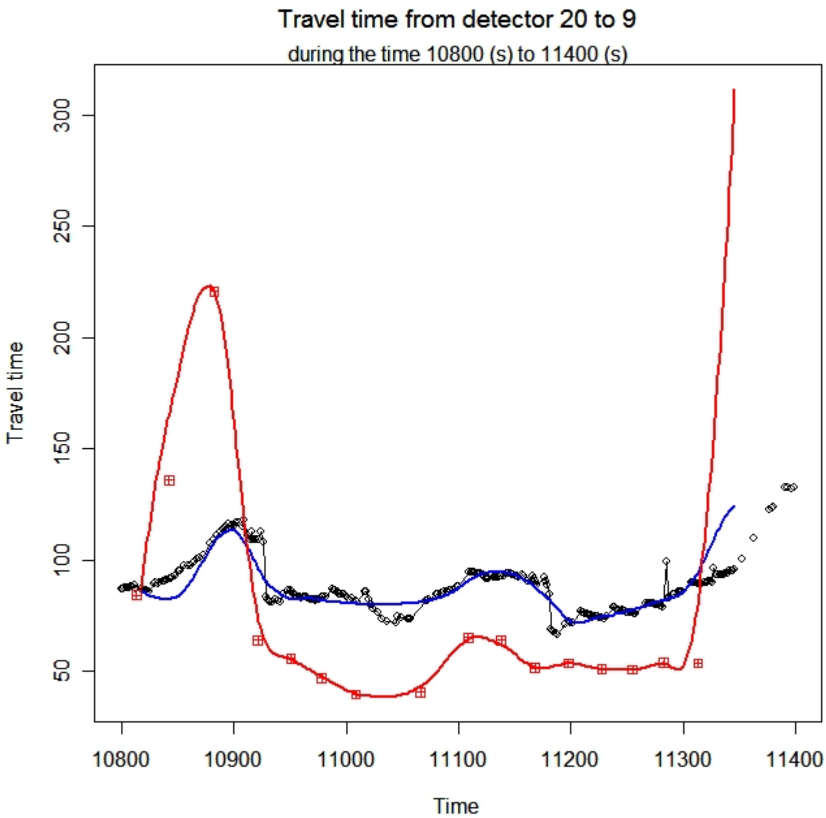

Figure 4.27: Benchmark Results of vehicle reidentification between detector 20 to 9, from time 10800(s) to time 11400(s). Corresponding statistics are given by Table 4.11.

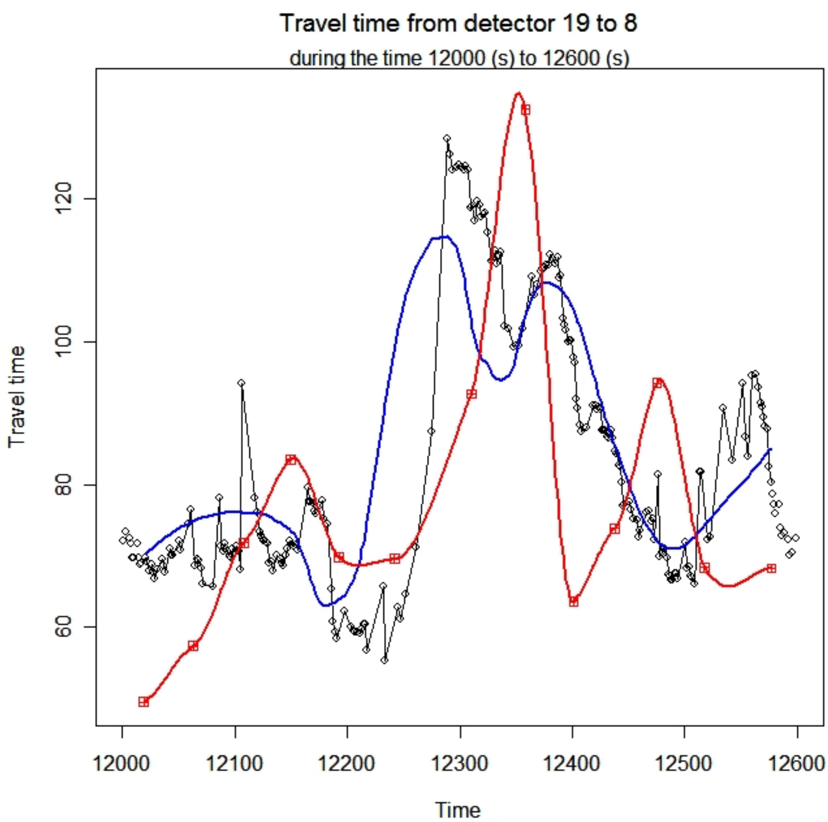

Figure 4.28: Benchmark Results of vehicle reidentification between detector 19 to 8, from time 12000(s) to time 12600(s). Corresponding statistics are given by Table 4.12.

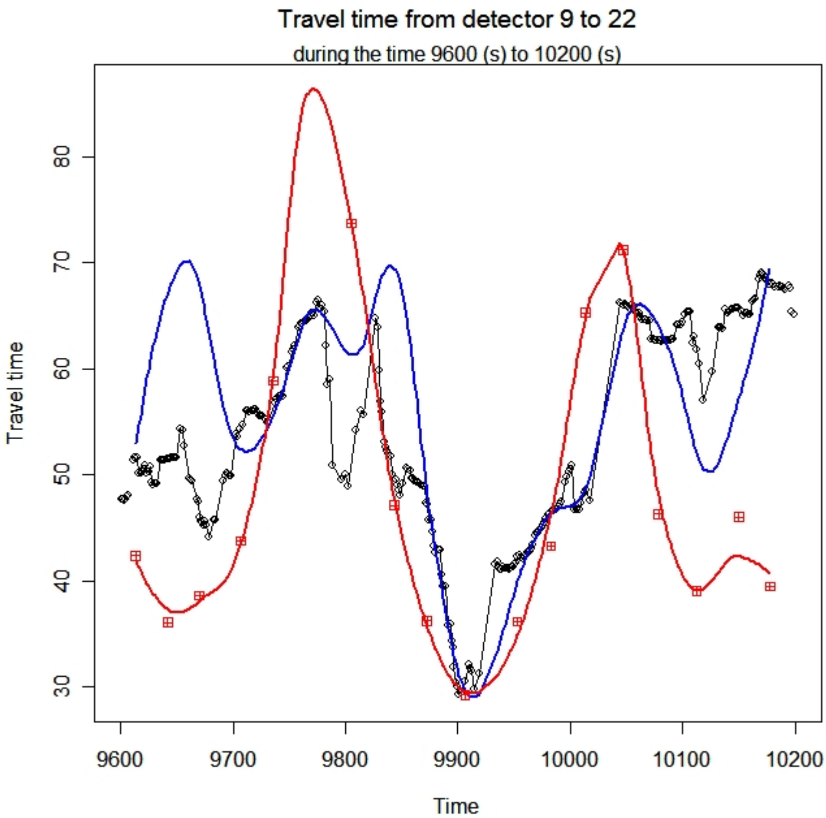

Figure 4.29: Benchmark Results of vehicle reidentification between detector 9 to 22, from time 9600(s) to time 10200(s). Corresponding statistics are given by Table 4.13.

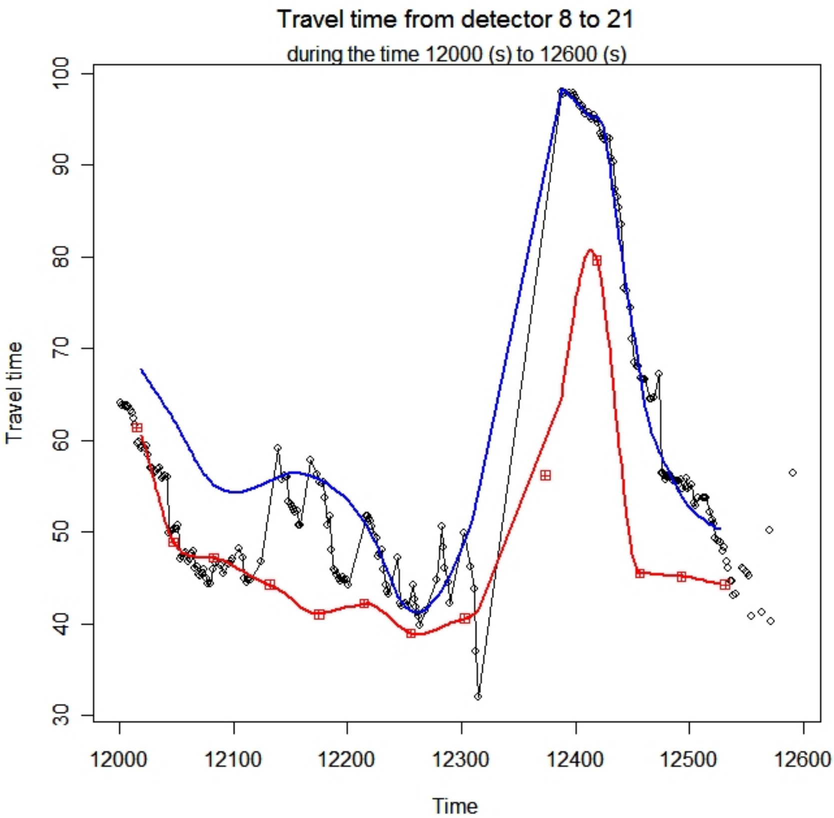

Figure 4.30: Benchmark Results of vehicle reidentification between detector 8 to 21, from time 12000(s) to time 12600(s). Corresponding statistics are given by Table 4.14.

4.3 Results of Estimating Travel Time Between Several Detectors

In this section, results of estimating travel time between several detectors are given. As mentioned in Section 3.5.1, two ways of estimation are used in this thesis.

4.3.1 Results of Approach 1

As mentioned in Section 3.5.1, approach 1 adds the results from each pair of consecutive detectors together. For this approach, only the results fitted by LPRF are given, as is shown by Figure 4.31 and 4.32. In the figures travel time estimations are calculated in a 30 minutes period for both inner lane(detector $15 \rightarrow 21$) and outer lane(detector $16 \rightarrow 22$). The results mostly reflect the trend of real travel times and therefore are acceptable.

4.3.2 Results of Approach 2

As mentioned in Section 3.5.1, approach 2 ignores the middle detectors and use detectors at the start and end as the UD and DD respectively. Results of this approach is given by Table 4.15 and Figure 4.33 and 4.34.

It can been seen that the correct reidentification rate is below 40%. Even though the parameter $Vmin$ is computed dynamically for the time window, as mentioned in Section 3.5.1, the result is still not very reliable. Therefore, it is not recommended to use the algorithm for reidentifying vehicles between several detectors. But the correct reidentification rate for large vehicles is above 80%, which means the reidentification of large vehicles is more reliable.

Case	1	2
Detectors	$15 \rightarrow 21$ (inner lane)	$16 \rightarrow 22$(outer lane)
Distance(meters)	2406	2406
Total No. of vehicles downstream	1784	2389
Total No. of sequences $v.ddN/v.ddn$	89	119
No. of large vehicles reidentified	86	19
No. of large vehicles reidentified correctly	69	18
% of correct large vehicle reidentifications	80	95
No. of vehicles to be reidentified	445	595
No. of correct reidentifications	139	232
% of correct reidentification	31	39

Table 4.15: Reidentification results between several consecutive detectors, during congested time period 9000(s) to 13000(s), about 67 minutes. Corresponding visual results are given by Figure 4.33 and 4.34.

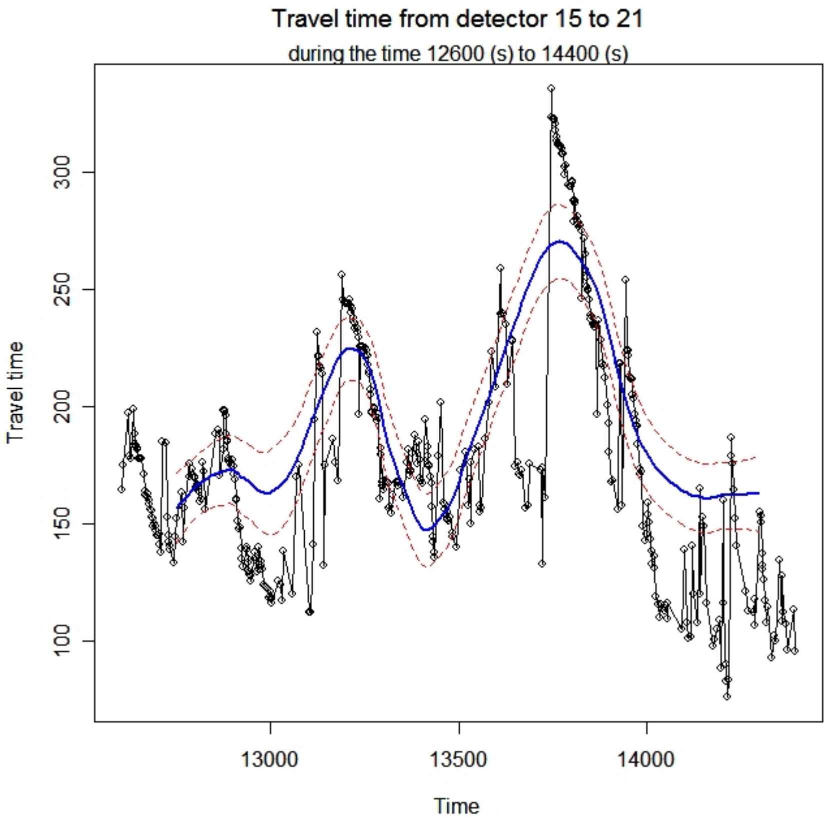

Figure 4.31: Results of approach 1: vehicle reidentification from detector 15 to 21 during time 12600(s) to 14400(s). Blue curve is the LPRF line with red dashed lines showing plus/minus standard errors.

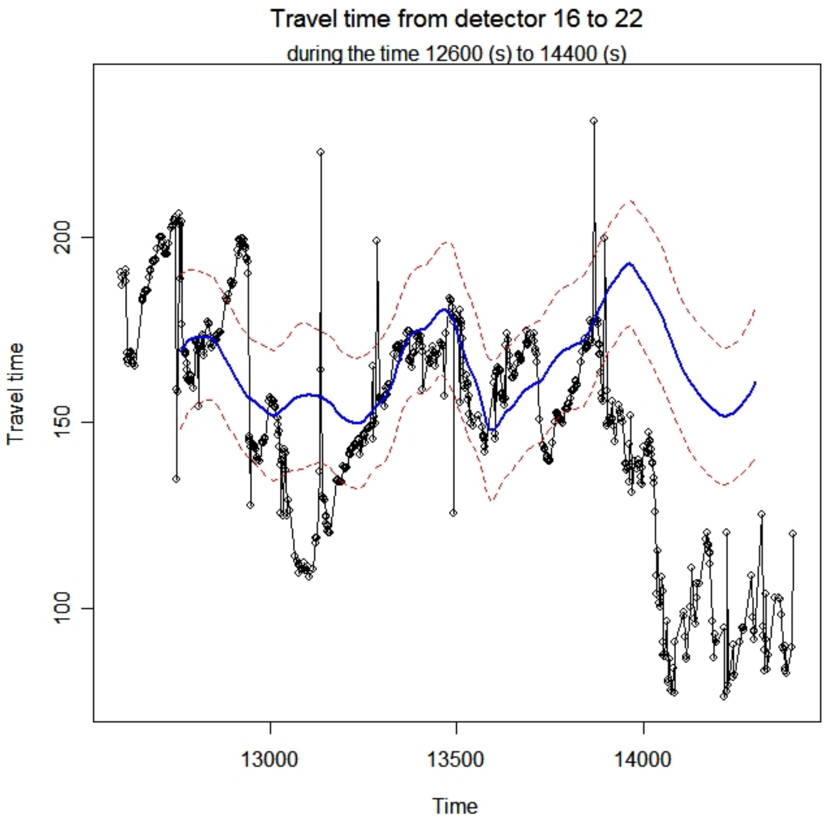

Figure 4.32: Results of approach 1: vehicle reidentification from detector 16 to 22 during time 12600(s) to 14400(s). Blue curve is the LPRF line with red dashed lines showing plus/minus standard errors.

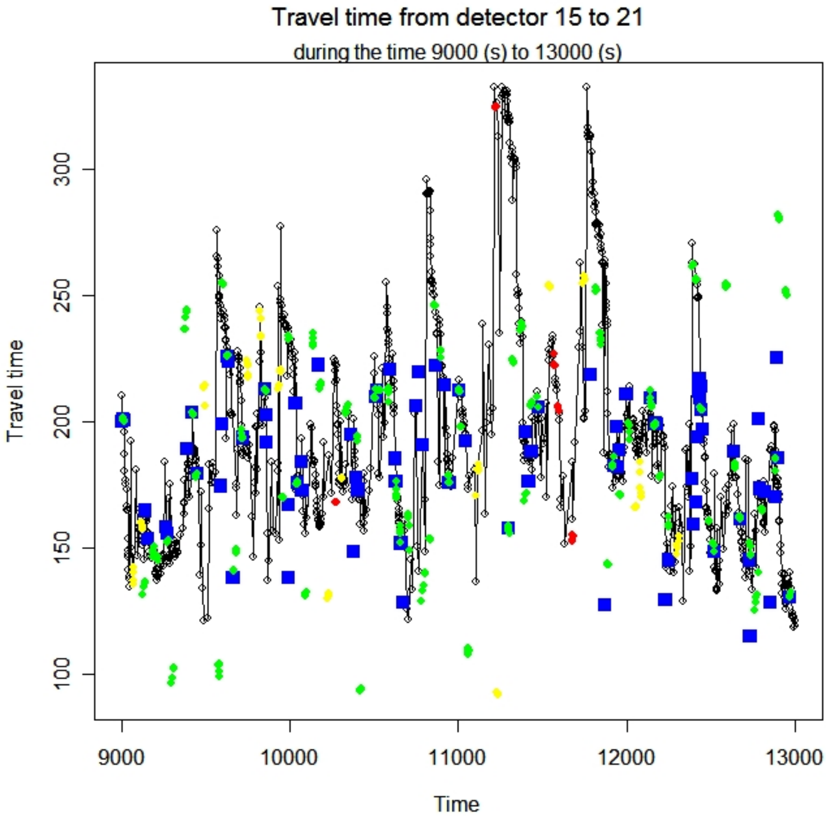

Figure 4.33: Results of approach 2: vehicle reidentification from detector 15 to 21 during time 9000(s) to 13000(s). Corresponding statistics are given by Table 4.15.

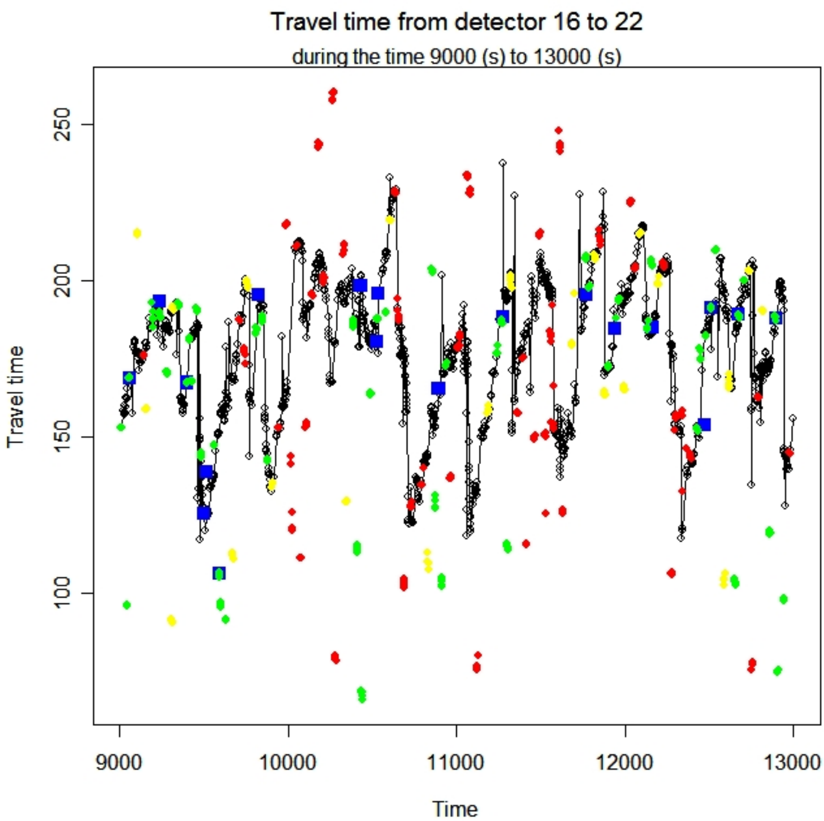

Figure 4.34: Results of approach 2: vehicle reidentification from detector 16 to 22 during time 9000(s) to 13000(s). Corresponding statistics are given by Table 4.15.

Chapter 5

Conclusion

The goal of this thesis is to present an algorithm to reidentify vehicles and thereby estimate travel time between consecutive detectors on a congested highway, using those information obtained from the already-exist highway traffic detectors. Therefore, there is no need to set up new hardware equipments on the highway, saving a lot of costs.

The result of implementing the algorithm has proven that the goal is achieved. It shows the algorithm works well with detectors 367 meters to 1150 meters away from each other. Yet, when the distance is longer than 2000 meters, e.g. between four consecutive detectors, the result is not good. Therefore, it is not recommend to apply the algorithm directly between several detectors on the highway.

The algorithm does not reidentify every vehicle on the highway; actually the algorithm reidentifies 5 out of every 20 vehicles, and this will provide enough information for traffic surveillance.

The way of vehicle reidentification developed by this thesis is different from any of those already-exist methods. It provides good results and it is very easy to implement.

For further applications of the algorithm, those empirically established parameters shall be modified according to different situations.

In the future, establishing parameters dynamically is of great interest to make the algorithm adapted to more situations, e.g. uncongested traffic, longer detector distances, etc..

Appendix A

List of Symbols and Abbreviations

Abbreviation	**Description**	**Definition**
UD	Upstream Detector	5
DD	Downstream Detector	5
TTM	Travel Time Matrix	20
MDM	Maximum Density Matrix	21
MPTT	Most Probable Travel Time	21
RPS	Relative Pattern Score	21
APS	Average Pattern Score	21
CPS	Correlation Pattern Score	21
DPS	Division Pattern Score	21
MPCS	Most Possible Corresponding Sequence	23
LPRF	Local Polynomial Regression Fitting	32
EWMA	Exponentially Weighted Moving Average	32
$v.dd$	Vehicles recorded at DD during a period of time	26
$v.ddN$	A big sequence of N vehicles at DD	28
$v.ddn$	A small sequence of n vehicles reidentified from a $v.ddN$	31

Appendix B

List of Figures

Figures are listed below.

List of Figures

Appendix C

List of Tables

Tables are listed below.

List of Tables

Appendix D

Codes of the Algorithm

The algorithm is coded with *S* language and implemented in software *R*.

The main code of the algorithm together with its benchmark is listed below.

```
# The following code is for vehicle reidentification,
# using the big to small method.

# The LARGE vehicle information is used to narrow the
# possible time window

# The codes are going to find matches during the
# congested period, i.e. 9000~15000, between detector ud1 and dd1

# the counting time unit is 10 mins, which means the travel time
# is estimated every 10 mins


########## functions ##########

## function for taking 2n vehicles around some vehicle
around <- function(v, n, ...){
tf.cp <- tf[tf$CP == v$CP & tf$tenter>0,] # select vehicles at that CP
pos <- which(tf.cp$tenter == v$tenter) # locate position
v.around <- tf.cp[(pos-n):(pos+n),]
return(v.around)
}
##

## function for deciding weights ##
lambda <- function(v, prev.type, t.diff, ...){
if(v$type == 0){
```

```
lambda = 0.9
return(lambda)
}else{
if(v$type == 1){
if(prev.type == 0 & t.diff < 15){
lambda = 0.4
return(lambda)
}else{
lambda = 0.7
return(lambda)
}
}else{
if(v$type == 2){
lambda = 0.6
return(lambda)
}else{# type 3
lambda = 0.5
return(lambda)
}
}
}
}

### function for calculating harmonic means
harmonic <- function(vec,...){
d <- length(vec)
har <- d/sum(1/vec)
return(har)
}
##

###############################

########## parameters #########

# uncomment those detectors when used

# ud1 <- 16 # upstream detector
# dd1 <- 20 # downstream detector
# distance <- 367

# ud1 <- 15 # upstream detector
# dd1 <- 19 # downstream detector
# distance <- 367

# ud1 <- 20 # upstream detector
# dd1 <- 9 # downstream detector
# distance <- 1150

# ud1 <- 19 # upstream detector
# dd1 <- 8 # downstream detector
# distance <- 1150

# ud1 <- 9 # upstream detector
# dd1 <- 22 # downstream detector
# distance <- 889

ud1 <- 8 # upstream detector
dd1 <- 21 # downstream detector
distance <- 889

# print in screen:
cat("\nFrom detector", ud1, "to", dd1, ", Distance:", distance)

# vehicles will be treated as large ones if length is larger than this
```

```
large <- 6

cat("\nLARGE:",large)

vmax <- 120/3.6 # the velocity limit on the freeway(m/s)
# in the time window, for large vehicles
tmin <- distance/vmax
tmax <- distance/4

tx = 15 # tx'th part of time. Change this to obtain other time period.
t1 <- 9000+600*tx
t2 <- t1+600
cat("\nFrom time",t1,"(s) to time",t2,"(s)")
cat("\ntx=",tx)

me <- 0.03 # measurement error proportional to vehicle length

tw.around <- 60 # time interval around a large vehicle,
# in which vehicles are assumed to have similar travel time
cat("\ntw.around large(s):",tw.around)

# use N vehicles as a computing unit
N = 20
n <- c(5,5,5) # small sequence length for type 1,2 and 3
cat("\nn:",n,"\n")

cat("\nN vehicles as a computing unit:",N)

# use this to multiply the travel time used by a large vehicle
multiplier = 1.2
cat("\nMultiplier:",multiplier)
add2 = 0.1
add3 = 0.2
cat("\nUse add2 to give bigger multiplier when type 2:",add2)
cat("\nUse add3 to give bigger multiplier when type 3:",add3)
###############################

#### main matter ####

# at dd1, from time t1 to t2, record the vehicles
v.dd <- tf[tf$tenter>=t1 & tf$tenter<=t2 & tf$CP==dd1 ,
c("CP","tenter","tleave","VehNo","Type","velocity","VehLength","LengthMeasure")]
cat("\nThere are ",dim(v.dd)[1],"vehicles at downstream detector")

v.dd.large <- v.dd[v.dd$LengthMeasure > large,] # large vehicles
v.dd.large.length <- dim(v.dd.large)[1]
cat("\nThere are ",v.dd.large.length,"large vehicles detected at downstream.\n")

v.dd.large.tt <- NULL # travel time for the large vehicle

# LARGE vehicles reidentification
no.match <- NULL # records which large vehicle has no match in upstream
for (l in 1:v.dd.large.length){
v.ud.window <- tf[tf$tenter <= v.dd.large$tenter[l]-tmin &
tf$tenter >= v.dd.large$tenter[l]-tmax & tf$CP == ud1, ]
lmd <- v.dd.large$LengthMeasure[l] # length measured at downstream
pm <- data.frame() # possible match
for (w in 1:dim(v.ud.window)[1]){
lmu <- v.ud.window$LengthMeasure[w] # length measured at upstream
if (lmu/(1+me) <= lmd/(1-me) & lmu/(1-me) >= lmd/(1+me)){
pm <- rbind(pm, v.ud.window[w,])
}
}

howmany <- dim(pm)[1] # how many possible matches
```

```
if (howmany > 1){
# use cor.maxN method to decide which one is better

cor.maxN = 0
cor.unsort = 0
where = 0

# take 10 vehicles around the downstream large
v.dd.large.around <- around(v.dd.large[l,],5)

# find the max cor
v.dd.large.around.length <- sort(v.dd.large.around$LengthMeasure)
for (i in 1:howmany){
v.pm.around <- around(pm[i,],5)
v.pm.around.length <- sort(v.pm.around$LengthMeasure)
temp <- cor(v.dd.large.around.length, v.pm.around.length)
if (temp > cor.maxN){
cor.maxN <- temp
where <- i
  }
}
# then give new single pm
pm <- pm[where,]
howmany = 1
}
if (howmany == 1){
v.dd.large.tt[l] <- v.dd.large$tenter[l]-pm$tenter
}else{# no match
no.match <- c(no.match,l)
}
}
v.dd.large.tt <- subset(v.dd.large.tt, is.na(v.dd.large.tt)==F)
if(length(no.match)>0){
v.dd.large <- v.dd.large[-no.match,]
}

# remove those large v's whose tt is
# longer than 1.4times of the mean large tt
# or shorter than 0.6times of the mean large ttw
out <- which(v.dd.large.tt > 1.4*mean(v.dd.large.tt)
| v.dd.large.tt < 0.6*mean(v.dd.large.tt))

if(length(out)>0){
v.dd.large <- v.dd.large[-out,]
v.dd.large.tt <- v.dd.large.tt[-out]
}

# in the time window, dynamic
tmin <- min(v.dd.large.tt)*0.8
tmax <- max(v.dd.large.tt)*1.3

v.dd.large.length <- dim(v.dd.large)[1]
cat(v.dd.large.length,"of them found possible match.\n")
# print(v.dd.large.tt)

v.dd.large.window <- cbind(v.dd.large$tenter-tw.around, v.dd.large$tenter+tw.around)

# match work: reidentify big sequences v.ddN
jth = 1 # the jth N vehicles in v.dd
match.rateN <- 0  # stores the match rate for v.ddN
match.raten <- 0 # stores the match rate for v.ddn

cor.maxn <- 0 # stores the max correlation

where <- 0 # stores where the max correlation is found
```

```
type = 0 # stores the type of v.ddN
tt.estN = 0 # stores the travel time estimation of a v.ddN
te.estN = 0 # stores the arrival time estimation of a v.ddN
te.ud.estN = 0 # stores the arrival time estimation at UD

tt.estn <- NULL # stores the travel time estimation of a v.ddn
te.estn <- NULL # stores the arrival time estimation of a v.ddn

tt.est.bench <- NULL # stores the travel time estimation of benchmark
te.est.bench <- NULL # stores the arrival time estimation of benchmark

finds <- data.frame() # store the final match found
finds.exp <- data.frame() # store the EWMA results

while (jth*N <= dim(v.dd)[1]){
v.ddN <- v.dd[((jth-1)*N+1):(jth*N),]

wd.tt <- 0 # records the large vehicle's travel time in its window

flag = 0
for (j in 1:N){
# within denotes to which large v's window does this vehicle belong
within <- which(v.ddN$tenter[j]>=v.dd.large.window[,1] &
v.ddN$tenter[j]<=v.dd.large.window[,2])
if (length(within)>0){
# use the biggest window
wd.tt[j] <- max(v.dd.large.tt[within])
}else{
wd.tt[j] = 0
flag = 1
}
}

wd.tt.max <- max(wd.tt)

# #  # maxT is for type 3 time window
# #  maxT = max(v.dd.large.tt)


# if there exists(not all) vehicle (in the N v's)
# that is not in any of the large v's windows, use bigger multiplier
if (wd.tt.max > 0 & flag == 1){
v.ud.window <- tf[tf$tenter<=v.ddN$tenter[N]-tmin
& tf$tenter>=v.ddN$tenter[1]-(multiplier+add2)*wd.tt.max & tf$CP==ud1,
c("CP","tenter","tleave","VehNo","Type","velocity",
"VehLength","LengthMeasure")]
v.ud.window.length <- dim(v.ud.window)[1]
type[jth] = 2
}else{
# if all are within the large v's windows
if(wd.tt.max > 0 & flag == 0) {
v.ud.window <- tf[tf$tenter<=v.ddN$tenter[N]-tmin
& tf$tenter>=v.ddN$tenter[1]-multiplier*wd.tt.max & tf$CP==ud1,
c("CP","tenter","tleave","VehNo","Type","velocity","VehLength","LengthMeasure")]
v.ud.window.length <- dim(v.ud.window)[1]
type[jth] = 1
}else{ # none is in the large v's windows
v.ud.window <- tf[tf$tenter<=v.ddN$tenter[N]-tmin
& tf$tenter>=v.ddN$tenter[1]-tmax & tf$CP==ud1,
c("CP","tenter","tleave","VehNo","Type","velocity","VehLength","LengthMeasure")]
v.ud.window.length <- dim(v.ud.window)[1]
type[jth] = 3
}
}
```

```
  cor.maxN[jth] = 0


# maximum correlation method
v.ddN.vehLength <- v.ddN$LengthMeasure
if(v.ud.window.length > N){
i = 1
while (i+N-1<=v.ud.window.length){
v.ud.vehLength <- v.ud.window$LengthMeasure[i:(i+N-1)]
temp <- cor(sort(v.ud.vehLength),sort(v.ddN.vehLength))
if (temp > cor.maxN[jth]){
cor.maxN[jth] <- temp
where[jth] <- i
  }

i  = i+1
}

MPCS <- v.ud.window[where[jth]:(where[jth]+N-1),]

match.tableN <- table(is.element(MPCS$VehNo, v.ddN$VehNo))
match.rateN[jth] <- match.tableN["TRUE"]/N

# travel time
te.estN[jth] = mean(v.ddN$tenter)
te.ud.estN[jth] = mean(MPCS$tenter)
tt.estN[jth] = te.estN[jth]-te.ud.estN[jth]

# to small
kth = 1 # the kth n vehicles in v.ddN
cor.maxn[jth] = 0
pair <- NULL # store which is the best match

while(kth+n[type[jth]]-1<=N){
v.ddn <- v.ddN[kth:(kth+n[type[jth]]-1),] # the kth n v's in v.ddN

# maximum correlation method
mth = 1 # the mth n v's in MPCS

while(mth+n[type[jth]]-1<=N){
MPCS.m <- MPCS[mth:(mth+n[type[jth]]-1),]
temp <- cor(MPCS.m$LengthMeasure,v.ddn$LengthMeasure)
if (temp > cor.maxn[jth]){
cor.maxn[jth] <- temp
pair <- c(kth,mth)
  }
  mth = mth+1
}

kth = kth+1
}

v.ddn <- v.ddN[pair[1]:(pair[1]+n[type[jth]]-1),]
MPCS.m <- MPCS[pair[2]:(pair[2]+n[type[jth]]-1),]
v.ddn$type <- type[jth]
v.ddn$tt.est <- v.ddn$tenter - MPCS.m$tenter

finds <- rbind(finds,v.ddn)
te.estn <- c(te.estn,mean(v.ddn$tenter))
tt.estn <- c(tt.estn,mean(v.ddn$tt.est))

match.tablen <- table(is.element(MPCS.m$VehNo, v.ddn$VehNo))
match.raten[jth] <- match.tablen["TRUE"]/n[type[jth]]
```

```
}else{ # not enough v's in the window!!!
where[jth] = NA

MPCS <- v.ud.window

match.tableN <- table(is.element(MPCS$VehNo,v.ddN$VehNo))
match.rateN[jth] <- match.tableN["TRUE"]/N

# travel time
te.estN[jth] = mean(v.ddN$tenter)
te.ud.estN[jth] = mean(MPCS$tenter)
tt.estN[jth] = te.estN[jth]-te.ud.estN[jth]

# to small
n.min <- min(n[type[jth]],v.ud.window.length)
kth = 1 # the kth n vehicles in v.ddN
cor.maxn[jth] = 0
pair <- NULL # store which is the best match

while(kth+n.min-1<=N){
v.ddn <- v.ddN[kth:(kth+n.min-1),] # the kth n v's in v.ddN

# maximum correlation method
mth = 1 # the mth n v's in MPCS

while(mth+n.min-1<=v.ud.window.length){
MPCS.m <- MPCS[mth:(mth+n.min-1),]
temp <- cor(MPCS.m$LengthMeasure,v.ddn$LengthMeasure)
if (temp > cor.maxn[jth]){
cor.maxn[jth] <- temp
pair <- c(kth,mth)
  }
  mth = mth+1
}

kth = kth+1
}

v.ddn <- v.ddN[pair[1]:(pair[1]+n.min-1),]
MPCS.m <- MPCS[pair[2]:(pair[2]+n.min-1),]
v.ddn$type <- type[jth]
v.ddn$tt.est <- v.ddn$tenter - MPCS.m$tenter

finds <- rbind(finds,v.ddn)
te.estn <- c(te.estn,mean(v.ddn$tenter))
tt.estn <- c(tt.estn,mean(v.ddn$tt.est))

match.tablen <- table(is.element(MPCS.m$VehNo, v.ddn$VehNo))
match.raten[jth] <- match.tablen["TRUE"]/n[type[jth]]
}

# benchmark
velocity <- v.ddN$velocity
har.mean <- harmonic(velocity)
te.est.bench <- c(te.est.bench, mean(v.ddN$tenter))
tt.est.bench <- c(tt.est.bench, distance/har.mean)

jth = jth+1

}


t.dd <- NULL
tt.dd <- NULL
tf.temp <- tf[tf$CP == ud1 & tf$tenter>0,]
```

```
for (i in 1:dim(v.dd)[1]){
temp <- tf.temp[tf.temp$VehNo == v.dd$VehNo[i],]
if(dim(temp)[1] == 1){
t.dd <- c(t.dd,v.dd$tenter[i])
tt.dd <- c(tt.dd,v.dd$tenter[i]-temp$tenter)
}
}

# EWMA smoothing

finds.exp <- data.frame(cbind(match.rateN,match.raten,type,te.estn,tt.estn))

v.dd.large$match.rateN=0
v.dd.large$match.raten=0
v.dd.large$type=0
v.dd.large$te.estn = v.dd.large$tenter
v.dd.large$tt.estn = v.dd.large.tt

finds.exp <- rbind(finds.exp,v.dd.large[,c('match.rateN','match.raten',
'type','te.estn','tt.estn')])

finds.exp <- finds.exp[order(finds.exp$te.estn),]

finds.exp$tt.exp <- 0


l = 0 # store lambda
# the first one:
if(finds.exp$type[1] == 0 | finds.exp$type[1] == 1 | finds.exp$type[1] == 2){
finds.exp$tt.exp[1] = finds.exp$tt.estn[1]
}else{
finds.exp$tt.exp[1] = mean(finds.exp$tt.estn)
}

# the others
for (i in 2:dim(finds.exp)[1]){
prev.type = finds.exp$type[i-1]
t.diff = finds.exp$te.estn[i]-finds.exp$te.estn[i-1]
l[i] <- lambda(finds.exp[i,], prev.type, t.diff)
finds.exp$tt.exp[i] <- (1-l[i])*finds.exp$tt.exp[i-1]+l[i]*finds.exp$tt.estn[i]
}
tt.exp <- finds.exp$tt.exp

# print(finds.exp)

tt.loess <- loess(finds.exp$tt.estn~finds.exp$te.estn, span=0.3)
tt.predict <- predict(tt.loess,data.frame(te.estn = seq(t1+10,t2-10,10)),se=T)


# some important results to print out
amount <- jth-1 # how many v.ddN or v.ddn
cat("There are ",amount,"groups of v.ddN,",amount*n[1],
"vehicles to be reidentified. \n")

# change values of NA to 0 in match.rateN and match.raten
for (a in 1:amount){
if(is.na(match.rateN[a])){
match.rateN[a] = 0
}
if(is.na(match.raten[a])){
match.raten[a] = 0
}
}

# no. of vehicles got final real matches
```

```
final.match <- sum(n[1]*match.raten)
# overall match rate among all v.ddn
final.match.rate <- final.match/(n[1]*amount)
cat(final.match,"vehicles got correct match. \n")
cat("The overall match rate among all v.ddn is",final.match.rate,". \n")
cat("The overall match rate among all vehicles is",
final.match/dim(v.dd)[1],". \n")
cat("match.rateN:\n")
print(match.rateN*100)
cat("match.raten:\n")
print(match.raten*100)

#### plots ####
graphics.off()
plot(t.dd,tt.dd, col="black", xlab="Time", ylab="Travel time",
ylim=range(finds$tt.est, tt.dd),
main=bquote("Travel time from detector"~.(ud1)~"to"~.(dd1)))
mtext(bquote("during the time"~.(t1)~"(s) to"~.(t2)~"(s)"))
lines(loess(tt.dd~t.dd),col="black")
points(v.dd.large$tenter,v.dd.large.tt, pch=15, col="blue",cex=1.5)

points(finds[finds$type==1,]$tenter,finds[finds$type==1,]$tt.est,
col="green",pch=19,cex=1)
points(finds[finds$type==2,]$tenter,finds[finds$type==2,]$tt.est,
col="yellow",pch=19,cex=1)
points(finds[finds$type==3,]$tenter,finds[finds$type==3,]$tt.est,
col="red",pch=19,cex=1)


# with EWMA and loess
windows()
plot(t.dd,tt.dd, col="black", xlab="Time", ylab="Travel time",
ylim=range(finds$tt.est,tt.dd),
main=bquote("Travel time from detector"~.(ud1)~"to"~.(dd1)))
mtext(bquote("during the time"~.(t1)~"(s) to"~.(t2)~"(s)"))
lines(loess(tt.dd~t.dd),col="black", lwd=2)
points(v.dd.large$tenter,v.dd.large.tt, pch=15, col="blue",cex=1.5)


points(finds[finds$type==1,]$tenter,finds[finds$type==1,]$tt.est,
col="green",pch=19,cex=1.5)
points(finds[finds$type==2,]$tenter,finds[finds$type==2,]$tt.est,
col="yellow",pch=19,cex=1.5)
points(finds[finds$type==3,]$tenter,finds[finds$type==3,]$tt.est,
col="red",pch=19,cex=1.5)

points(finds.exp$te.estn,finds.exp$tt.exp,col=6,pch=18,cex=1.5)


lines(seq(t1+10,t2-10,10),tt.predict$fit, col = 'blue', lwd = 2)
lines(seq(t1+10,t2-10,10),tt.predict$fit+tt.predict$se.fit, col = 'red', lty=2)
lines(seq(t1+10,t2-10,10),tt.predict$fit-tt.predict$se.fit, col = 'red', lty=2)

lines(loess(finds.exp$tt.exp~finds.exp$te.estn),lty=1,col=6,lwd=2)


### benchmark ###
tt.est.bench <- NULL
te.est.bench <- NULL


jth = 1
while (jth*N <= dim(v.dd)[1]){
v.ddN <- v.dd[((jth-1)*N+1):(jth*N),]
velocity <- v.ddN$velocity
```

```
har.mean <- harmonic(velocity)
te.est.bench <- c(te.est.bench, mean(v.ddN$tenter))
tt.est.bench <- c(tt.est.bench, distance/har.mean)

jth = jth + 1
}


# loess predictions
reid.loess <- loess(finds.exp$tt.estn~finds.exp$te.estn, span=0.3)
harm.loess <- loess(tt.est.bench~te.est.bench, span=0.3)

real.te <- t.dd

reid.predict <- predict(reid.loess,data.frame(te.estn = real.te))
harm.predict <- predict(harm.loess, data.frame(te.est.bench = real.te))

# find the index where both reid and harm are not NA
index = which((is.na(harm.predict)+is.na(reid.predict))==0)

real.te <- real.te[index]
# real.predict <- real.predict[index]
real.tt <- tt.dd[index]
reid.predict <- reid.predict[index]
harm.predict <- harm.predict[index]

real.mean <- mean(real.tt)
reid.mean <- mean(reid.predict)
harm.mean <- mean(harm.predict)

reid.bias <- reid.mean-real.mean
harm.bias <- harm.mean-real.mean
cat("bias reid:",reid.bias,"\n")
cat("bias harm:",harm.bias,"\n")

reid.sse <- sum(I(reid.predict-real.tt)^2)
harm.sse <- sum(I(harm.predict-real.tt)^2)
cat("SSE reid:",reid.sse,"\n")
cat("SSE harm:",harm.sse,"\n")

reid.var <- var(reid.predict-real.tt)
harm.var <- var(harm.predict-real.tt)
cat("Var reid:",reid.var,"\n")
cat("Var harm:",harm.var,"\n")

windows()
plot(t.dd,tt.dd, col="black", xlab="Time", ylab="Travel time",
ylim=range(real.tt,reid.predict,harm.predict),
main=bquote("Travel time from detector"~.(ud1)~"to"~.(dd1)))
mtext(bquote("during the time"~.(t1)~"(s) to"~.(t2)~"(s)"))
lines(loess(real.tt~real.te),col="black")

# lines(seq(t1+10,t2-10,1), real.predict, lwd=1)
lines(real.te, reid.predict, col="blue", lwd=2, lty=1)
lines(real.te, harm.predict, col="red", lwd=2, lty=1)

points(te.est.bench,tt.est.bench, pch = 12, col=2)
# lines(te.est.bench,tt.est.bench, lty=3, col=4)
```

Bibliography

[1] Benjamin Coifman. Identifying the onset of congestion rapidly with existing traffic detectors. *TRB 80 Conference*, 2001.

[2] Benjamin Coifman and Michael Cassidy. Vehicle reidentification and travel time measurement on congested freeways. *Published by Elsevier Science Ltd.*, 2001.

[3] Benjamin Coifman and Edgar Ergueta. Improved vehicle reidentification and travel time measurement on congested freeways. *In TRB 81 Conference*, 2002.

[4] Benjamin Coifman and Sivaraman Krishnamurthy. Vehicle reidentification and travel time measurement across freeway junctions using existing detector infrastructure. *Published by Elsevier Science Ltd.*, 2007.

[5] P. Davies, C. Hill, and N. Emmott. Automatic vehicle identification to support driver information systems. *IEEE Vehicle Navigation and Information Systems Conference*, 1989.

[6] T. Huang and S. Russell. Object identification in a bayesian context. *The Fifteenth International Joint Conference on Artificial Intelligence*, 1997.

[7] R. Kuhne and S. Immes. Freeway control systems for using section-related traffic variable detection. *Pacific Rim TransTech Conference*, 1993.