

Portable Loop Fault Diagnosis Tool Development for Control Cabinet Level

by

Xiao-Yun Lu*

*California PATH, ITS, University of California, Berkeley
Richmond Field Station, Bldg 452, 1357 S. 46th Street, Richmond, CA 94804-4648
Tel: 1-510-665 3644, Fax: 1-510-665 3537, Email: xylu@path.berkeley.edu*

ZuWhan Kim

*California PATH, ITS, University of California, Berkeley
Richmond Field Station, Bldg 452, 1357 S. 46th Street, Richmond, CA 94804-4648
Tel: 1-510-665 3586, Fax: 1-510-665 3537, Email: zuwhan@berkeley.edu*

Meng Cao

*PATH, ITS, University of California, Berkeley
Richmond Field Station, Bldg 452, 1357 S. 46th Street, Richmond, CA 94804-4648
Tel: 1-510-6653552, Fax: 1-510-665 3537, Email: mcao@berkeley.edu*

Pravin Varaiya

*Department of EECS, University of California, Berkeley
271M Cory Hall, Berkeley CA 94720-1770
Tel: 1-510-642-5270, Email: variya@eecs.berkeley.edu*

Roberto Horowitz

*Department of ME, University of California, Berkeley
5138 Etcheverry Hall, Berkeley CA 94720-1770
Tel: 1-510- 642-4675, Email: horowitz@me.berkeley.edu*

Joe Palen

*Division of Research and Innovation, California Department of Transportation
1127 O Street, Sacramento CA 95833
Tel: 916-654-8420; Fax: 916-654 9977, Email: Joe.Palen@dot.ca.gov*

For presentation and publication
88th Annual Meeting
Transportation Research Board
Washington, D.C.
July 28, 2008

Words: 5079
Plus 9 Figures (2,250)
Total: 7329

**Corresponding author*

Abstract

This paper presents preliminary results in the research and development of a Portable Loop Fault Detection Tool for use at the control cabinet level. This work is complementary to most previous work focusing on macroscopic faulty loop data. Part of the project is to develop a real-time vision-based multi-lane multi-vehicle tracking algorithm for freeways to use as baseline measurements to compare with the lower level loop signal for direct loop fault detection. The system is primarily developed for both freeways and arterials. It is composed of a mobile trailer, a retractable pole with a video camera mounted on top to look at the suspicious loop detector on the ground, a computer running the vehicle tracking algorithm, and another computer at the control cabinet interfacing with a loop detector card through RS232 serial port. Both computers run IEEE 802.11b wireless for information passing and synchronization. A small data packet of the virtual loop information passes from the trailer computer to the cabinet computer through a UDP protocol. Information from the virtual and the physical loops are then compared which can be viewed on a visual display. Preliminary tests have been conducted and the results are analyzed.

1. INTRODUCTION

A systematic approach to detecting faulty loops is crucial to traffic operation, traveler information, and Corridor Management. Traffic detection systems are widely used for traffic management and control in California. The statewide *sensor system* consists of over 25,000 sensors located on the mainline and ramps, and grouped into 8,000 vehicle detector stations (VDS). Over 90 percent of the sensors use inductive loops. However, loop data are not reliable. The loop data delivered to TMC may contain error created at a point or several points between the loop detector and the TMC database, which presents a great challenge to loop fault detection. To solve this problem, it is necessary to take a systematic approach. This approach is composed of three complementary tasks: (a) loop fault detection; (b) faulty loop data correction/imputation; and (c) loop detection system maintenance. In our previous work [1], we categorized the loop fault detection approaches into three levels: (i) Macroscopic Level: such as the TMC/PeMS; (ii) Mesoscopic Level - a stretch of freeway such as the Berkeley Highway Lab (BHL); and (iii) Microscopic Level at a control cabinet. Different data are available at different system levels. The former two are of a high level and the latter is of a low level. Loop fault detection at the high level is done usually through an analysis of aggregated data. Such approach is indirect, and shortcomings are obvious: (a) data aggregation in time and space would smear the faulty problem; and (b) a communication fault caused by data error/loss make it impossible to isolate the loop fault detection problem. Only the detection at the control cabinet level can directly detect the fault(s) in hardware and software, isolate them from the communication fault, and correct the faults permanently. In [1] we conducted (a) Systematic review of previous loop fault detection and data correction methods; and (b) Systematic classification of possible faults and causes in different levels.

The Problem

As listed in [1], the loop faults to be diagnosed at the microscopic level include hardware, software and installation problems, and loop card faults. Those faults appear as: *mis-assignment, temporary data missing, crosstalk, absence of data or constant for a period of time, broken cable, chattering, broken card, card sensitivity being too high or too low, broken pulse, mismatch of ON/OFF time instances between upstream and downstream loops for dual loop stations.*

This is the only level that one could conduct direct loop fault detection and isolate the loop faults from possible other system faults. Data in this level can either be the ones processed by loop detector card, which will be loop ON/OFF time instances or occupancies; or the raw loop pulse signal before the loop card. The main characteristics of those data are that (a) they do not pass any communication media and thus there is no possibility of communication fault which usually pollutes or looses the data stream; (b) all the raw information is available with a proper interface with the control cabinet; (c) real-time data are available; and (d) most importantly, an independent data source or baseline data could be obtained at this level thus loop fault detection could be conducted by comparing the loop detector reading with these data.

It is noted that only the fault detection at this level can be called direct and can be completely separated from communication error. It is also possible, only at this level, to

identify all the loop faults and their exact causes. In addition, it is practical only at this level that one could use baseline data generated from independent sensor(s) for comparison in loop fault diagnosis.

Document Review

A systematic document review on loop fault detection through faulty data at different aggregation level was conducted in [1]. The characteristic of this approach is to apply various statistical analysis methods to the aggregated loop data to figure out possible faults in the loop detection system. Since this approach is indirect, the faults that can be detected are usually large scale problems such as electric or communication system faults. These cannot tell exactly what the fault is, appearing at which point of which loop detection system. This is one of the limitations to the macroscopic and mesoscopic approaches. Besides, communication faults are tangled with the loop detection system fault. In the rest of the section our document review will be focused on the low level loop fault detection.

Many operating agencies use specialized loop testers to assess the quality of the wiring [2, 3]. However, these tools bypass the controller and the loop sensors. Therefore, they do not analyze the entire detector circuit, nor do they analyze the circuit in operation. To this end, most operating agencies employ simple heuristics, for example, manually checking the loop sensor indicator light when a vehicle passes over a loop. Such tests are typically employed when the loop detectors are installed close to the control cabinet. Many practitioners and some researchers [3, 4, 5] formalized this heuristic by examining if the time series of 30 second average flow and occupancy is within a statistical tolerance.

In [7] Chen and May addressed the fault detection problem for a single loop. They used the number pulses to replace the vehicle counts to verify the loop data. Therefore, if a pulse is broken, it would cause a data problem. They developed an automated loop fault detection system using aggregated data. Their method is to examine the detector *ON/OFF time instant*. Unlike conventional aggregate measures, their approach was sensitive to errors such as "pulse breakups", where a single vehicle registered multiple detections because the sensor output flickered off and back on. This was the main disadvantage to use vehicle count from a single loop for fault detection: one could not isolate other loop fault from the pulse flickering problem.

Low level loop data correction traces back to the Freeway Service Patrol study in 1990s [8, 9]. It looked at the transition time measures in sub-second of dual loop stations with the 20ft distance between upstream and downstream loops. It observed some faults in low level data including:

- missing data;
- a matching error which results in unreasonable occupancy and speed;
- on-time and off-time not always being related;
- no-flow and no-speed but with positive occupancies;
- existing pulses in both up and down streams;

The author mentioned that some causes of the abnormal phenomenon could be explained as a vehicle's changing lane. However, there was no systematic diagnosis in [8] for those faults, nor systematic methods for data correction.

In [10], Skabardonis et al use dual loop information to detect loop fault. This paper focuses on the evaluation of loop sensors and the detection of cross-talk. It was developed for off-line

data analysis but could possibly be used for on-line applications in the future. The operation procedure is summarized in three steps:

- (i) Record a large number of vehicle actuations during free flow traffic;
- (ii) For each vehicle, match actuations between the upstream and downstream loops in the given lane;
- (iii) Take a difference between the matched upstream and downstream on-times and examine the distribution on a lane-by-lane basis. Assuming that the loops are functioning properly, only a small percentage of the differences should be over 1/30 seconds. Otherwise, the “Cross-talk” fault is announced.

The idea of using dual loop speed traps to identify the detector errors is also used in [10]. At free flow, the on-time difference and the off-time difference should be the same if there is no hardware problem. So if they are not the same, there may be a hardware and/or software problem. However, such an assumption is not true for non-free-flow.

Developing a portable tool for systematic and direct loop fault detection and correction from the control cabinet level has not been reported in the literature to our knowledge.

Main Contributions of the Paper

This paper presents a practical work conducted at California PATH for developing a portable tool to be used at the control cabinet level to accurately diagnose any fault(s) of a loop detection system (including loop circuits, loop cards, cable links, etc.), to check the detection accuracy, to adaptively adjust sensitivity of detector card, and to correct the faulty data. To achieve these functionalities at the low level, it is necessary to utilize an independent source as a baseline data to compare against the loop detection system output. Such a comparison also permits an evaluation of the loop system. Since multiple-vehicle tracking technologies using digital video camera on freeways have been well-developed and tested at PATH, it is used as the baseline measurement in the portable tool for the loop fault diagnosis. This paper presents the preliminary development of the system including the hardware, the software, the data communication method, and the algorithm.

The paper is structured in a following way: Section 2 is for the overall system structure; Section 3 is for the algorithm and the system development; section 4 presents some experimental work; and Section 5 is for concluding remarks and future work.

2. PORTABLE LOOP FAULT DETECTION TOOL (PLFDT)

The PLFDT (Figure 1) is designed for systematic loop fault detection at the control cabinet level. A loop detector(s) could be identified as being *suspicious* from a higher level data analysis in TMC/PeMS. The suspicious loops will then be diagnosed further using the portable tool. The tool will enable the operator to use independent stream of traffic measurements for comparing with the suspicious loop detector data. This portable tool will be designed to achieve the following objectives:

- determination of the exact fault type and causes in the detection system
- on-site diagnosis of faults on:
 - mis-assignment
 - malfunctioning such as misfiring

- inappropriate card sensitivity settings
- inductance variation due to temperature and humidity
- broken loop circuit due to
 - improper installation
 - road surface maintenance
 - fatigue
- facilitating on-site detector precision evaluation and calibration.

Overall System Structure of PLFDT

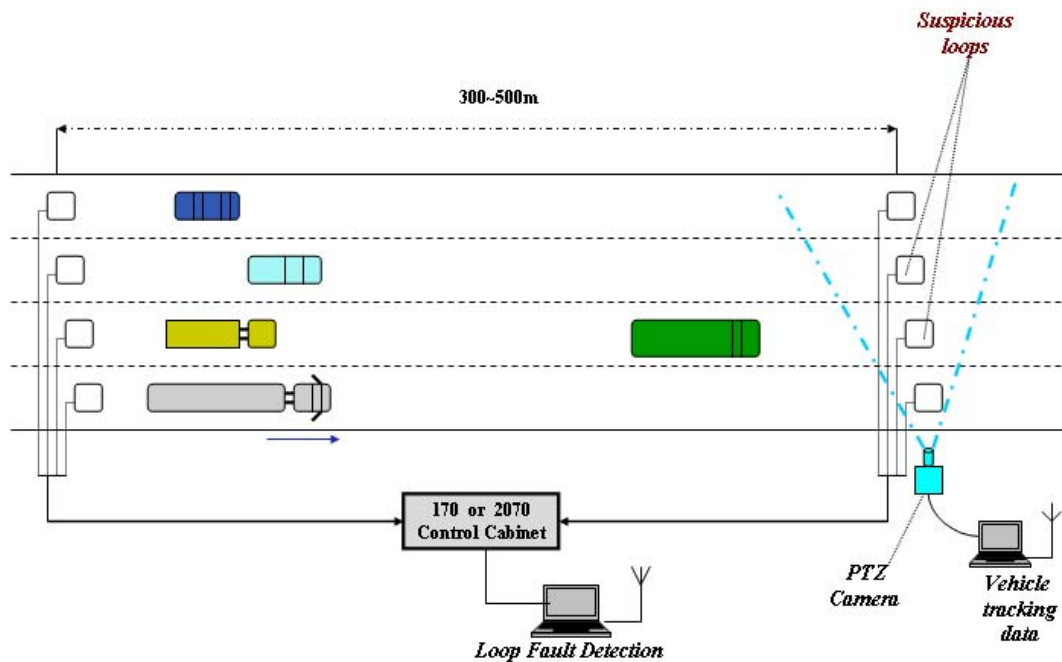


Figure 1. Overall system structure

The system development includes hardware, software and algorithm development. Our hardware setup consists of the following components:

- Mobile trailer which can be towed to the site near the suspicious loop location
- Retractable pole with a PTZ (Pen-Tilt-Zoom) camera mounted on it
- Two laptop computers with the Linux operating system
- *Computer A* to capture video images, process them in real-time, and send out the result via IEEE 802.11b wireless
- *Computer B* to interface with a loop detector card, receive the video processing data from Computer A through IEEE 802.11b wireless, and compare the synchronized signals for loop fault detection;

Mobile Pole for Roadside Video Camera Mounting

A mobile pole for the roadside camera setup has been developed (Figure 2).



Figure 2. Left: Mobile retractable pole; Upper right: PTZ camera on top for looking at the loop and for vehicle tracking to obtain baseline data; Lower right: video computer also running IEEE 802.11b wireless communication using USB port.

The mobile trailer has four retractable folding legs for supporting the platform for leveling and robustness. The Pan-Tilt-Zoom parameters can be controlled using the remote controller or using a control software running under Microsoft Window System through the RS 232 serial port interface. This setup process is necessary for the camera to view the loops on the ground and to display on the computer screen so that a virtual loop can be overlaid on the actual loop.

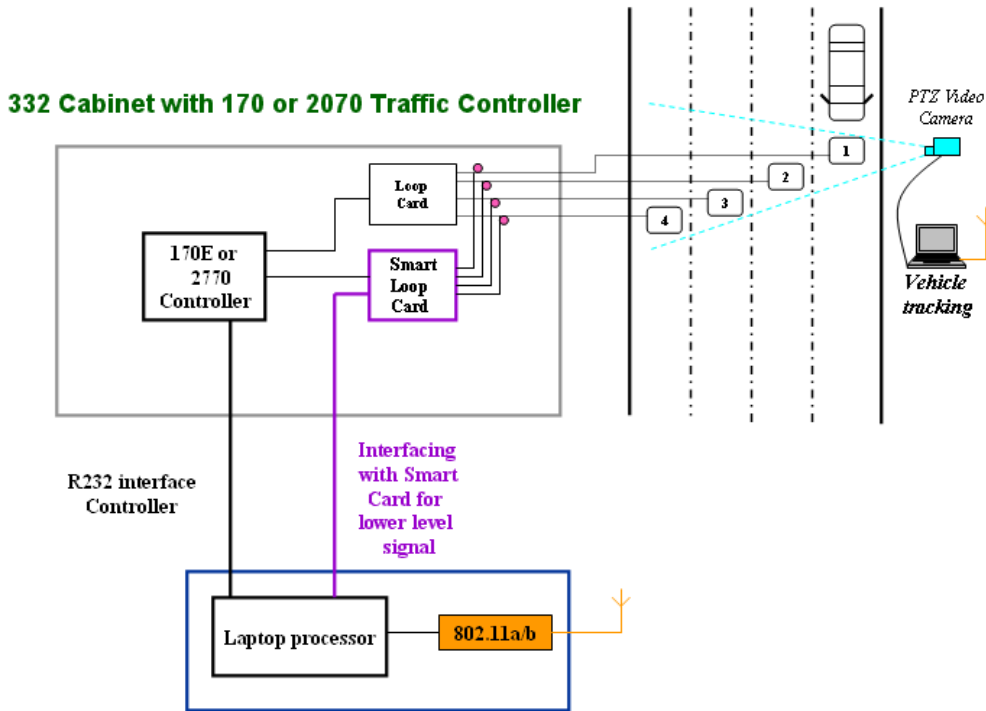


Figure 3. Interfacing with control cabinet and smart card

Interface with Control Cabinet

A loop card receives raw analog signals from each loop circuit, processes them with a physical oscillator and amplifier, and outputs traffic signals. Loop cards can be divided into two types: single-layer and multi-layer output cards (Figure 3). Single-layer output cards have only two outputs -- the vehicle count (volume) and occupancy -- which are results of processing the input signal from the loop circuit. For example, Sarasota GP5 and Reno 222 cards are single-layer output ones that are widely used. There is no direct interface port with these cards. Instead, their signals are directly fed into the controller. The output from the card to the controller is either 1 or 0 without the lower level signal available. The low level signal is more attractive than the binary data for several reasons: (a) it can be used for extracting vehicle signatures for re-identification; (b) it tells if the sensitivity of the card is properly adjusted; (c) it tells if any algorithm in the card has a flaw; and, (d) most importantly, we can remove time delay incurred in the traffic controller. Multi-layer output or *Smart Cards*, such as the 3M Canoga and IST cards, have multi-level output information including the start detection times, the occupancy, the vehicle count, fault status, and even the inductance intensity signals calculated from the frequency. A smart card has a built-in RS232 interface port, and thus lower level signal can be obtained. We chose a smart card, 3M Canoga C922, which is compatible with 332 Traffic Control Cabinets and both 170 and 2070n controllers, for our current development. The update rate for the 3M Canoga C922 card is 13Hz.

3. SOFTWARE DEVELOPMENT

The software has the following three components:

- High precision synchronization of the timers on the two computers through wireless communication
 - Real-time multi-lane and multi-vehicle tracking using the video camera, and
 - Matching signals from the two data streams,
- which are described respectively in this section.

Synchronization of the Two Computers with Wireless Communication

The two data streams (from the loop and the video camera) includes timestamps for matching. Potential faults are diagnosed by comparing the matched data pairs. The video processing data and the loop data are collected with time stamps in two different computers. Therefore, computer system time synchronization is critical.

We use wireless-based (UDP) synchronization tool developed by the California PATH to synchronize the two computers within 1 millisecond difference. The procedure is described as follows:

Step 1: *Computer A* send a signal packet, *MSG1*, to *Computer B* containing its current system time, say *Start_TIME*.

Step 2: When *Computer B* receives the signal packet *MSG1*, it immediately sends back the acknowledge signal packet, *MSG2*, to *Computer A*, which contains:

- a) the *Start_TIME* from the packet it received
- b) the current system time on *Computer B* at the time of receiving the packet from *Computer A*, say *Rcv_TIME*

Step 3: *Computer A* gets the acknowledge message, *MSG2*, and marks the current system time after receiving *MSG2* from *Computer B*. Then, the round trip time *R1* of data passing is calculated by subtracting the *Start_TIME* from the current system time. The clock skew between the two computers are then estimated by comparing *Rcv_TIME* with $(Start_TIME + 0.5 * R1)$.

Step 4: *Computer A* sends a time setting packet to *Computer B* with the clock skew and *Computer B* adjusts its system time accordingly.

The above process is iterated for 100 times and the average round trip time are used to estimate the clock skew. According to our experiment, the resulting clock skew is far less than 1 millisecond. It is a much more accurate and reliable way to synchronize the two computers than other affordable methods, such as using GPS units.

Real-Time Multi-lane Vehicle Tracking Algorithm

We have designed a computer vision system to obtain baseline measurements to compare. The whole system consists of three parts:

- a camera (Canon VC50i pan-tilt-zoom communication camera);
- a moving platform; and
- a Linux-based video processing software

The Canon VC50i camera provides a wide range of view by panning through a broad reach of 200 degree, tilting of 120 degree, as well as a 26x optical zooming. It provides superior camera optics that a good quality of images can be obtained even with challenging illumination conditions, such as strong shadow cast that causes too high image contrast. An

Intel Pentium Core laptop computer equipped with a USB frame grabber was used for video data processing.

There are many commercial vision-based vehicle detection systems (“virtual loop detectors”) also available. However, most virtual loop detectors are based on the background subtraction algorithm. They normally use frontal-view video images to avoid difficulties caused by occlusions and to get better lane positioning. Since our application requires the camera on the roadside, it is difficult to adopt those systems.

A video processing algorithm has been developed to detect and track vehicles in multiple lanes at the same time. The algorithm combines the background subtraction algorithm with the feature tracking and grouping algorithm to better handle the occlusion problem. The developed algorithm is more robust to shadow and occlusions than conventional virtual loop detectors and, thus, better separates between lanes. An example detection result is shown in Figure 4. We see that the upper-left background subtraction result cannot separate the vehicles in multiple lanes but the newly developed algorithm can correctly localize them.

An example placement of a “virtual loop” over the loop mark on the ground is shown in Figure 5. The trajectories of all the vehicles in the image are estimated, and the virtual loop is triggered by analyzing the trajectories.

The software was developed under Linux environment using the OpenCV library. The algorithm runs on a Pentium Core processor (1.83GHz) in real-time at 10 fps. The details of the image processing algorithm are described in [11].

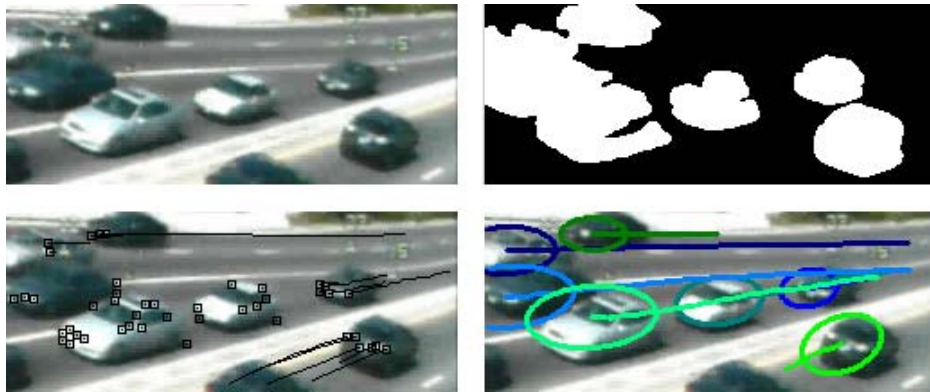


Figure 4: An example feature tracking. Upper-left: the original video image; Upper-right: the 'background subtraction' cue where the four vehicles in the left are detected as one big region; Lower-left: the feature detection and tracking cue; Lower-right: result by combining the background subtraction cue and the feature detection and tracking cue.



Figure 5: Flashing of the virtual loop is triggered when a vehicle trajectory reaches it

The 10Hz update rate is frequent enough to avoid any missing vehicle count due to high vehicle speed. For example, when a vehicle moves at 70mph or 31.3 m/s, and loop length 2m and vehicle length 4m, the total crossing length for a vehicle starting on upstream edge and leave at downstream edge is: $2+4=6m$. Thus the expected duration is $10.0/31.3=0.192s$. If the video camera update rate is 10Hz, there are at least one or two frames of the video where the vehicle is on the virtual loop. In addition, even if a fast vehicle passes through the virtual loop in between the frames, we can still infer vehicle's passage by analyzing the continuous trajectory that the vision algorithm provides.

Comparison of Physical Loop and Virtual Loop

Figure 6 illustrates the system structure developed to graphically monitor and compare the loop information. The instantaneous physical loop information and virtual loop information packets are processed and formatted as follows:

Loop Information Package

```
typedef struct{
    double timestamp;
    double Inductance[Max_Loops];
}Loops_TYPE
```

Virtual Loop information package

```
typedef struct {
    double timestamp;
    double On[Max_Loops];
}Virtual_Loops_TYPE
```

Thus the physical loop packet and virtual loop packet are matched based on the time stamp. However, the packet update rate from the vision system and that for Canoga card are different. The update of information packet for a vehicle over the virtual loop in the vision system is about 10 fps. The maximum update rate from the Canoga card is around 13 Hz. So this is not a one-to-one matching. On the other hand, the messages from both sides could possibly have some delay due to wireless communication or some other unknown reasons. To solve this problem, two First-In-First-Out (FIFO) buffers were built on the cabinet computer. One is used to store virtual loop packets from vision system and the other is used to store the physical loop packets from the Canoga card. With those two buffers, two initially synchronized computers can work independently as long as the data is time stamped. Each packet from the video computer (which has a lower update rate) is matched with the physical loop packet which has the closest time stamp by looking up their buffer. This approach significantly increases the reliability of the system.

Currently, the Loop Information Packet includes: time stamp and inductance. Occupancy can be deduced for a given sensitivity threshold. One of the research topics in the near future is to develop an adaptive sensitivity to address the inductance fluctuation caused temperature and humidity over the loop the road surface.

4. PRELIMINARY EXPERIMENTAL DATA ANALYSIS

Tests have been conducted at the Experimental Intersection in PATH Headquarters, RFS, U. C. Berkeley. A 3M Canoga 922 card was connected to a 322 traffic control cabinet to read the raw loop inductance data directly from the physical loop, as shown in Figure 7-(a). It also transfers these loop information to the Laptop through the RS-232 serial cable as shown in

Figure 7-(b). The 3M Canoga 922 card could read at most two physical loops at the same time.

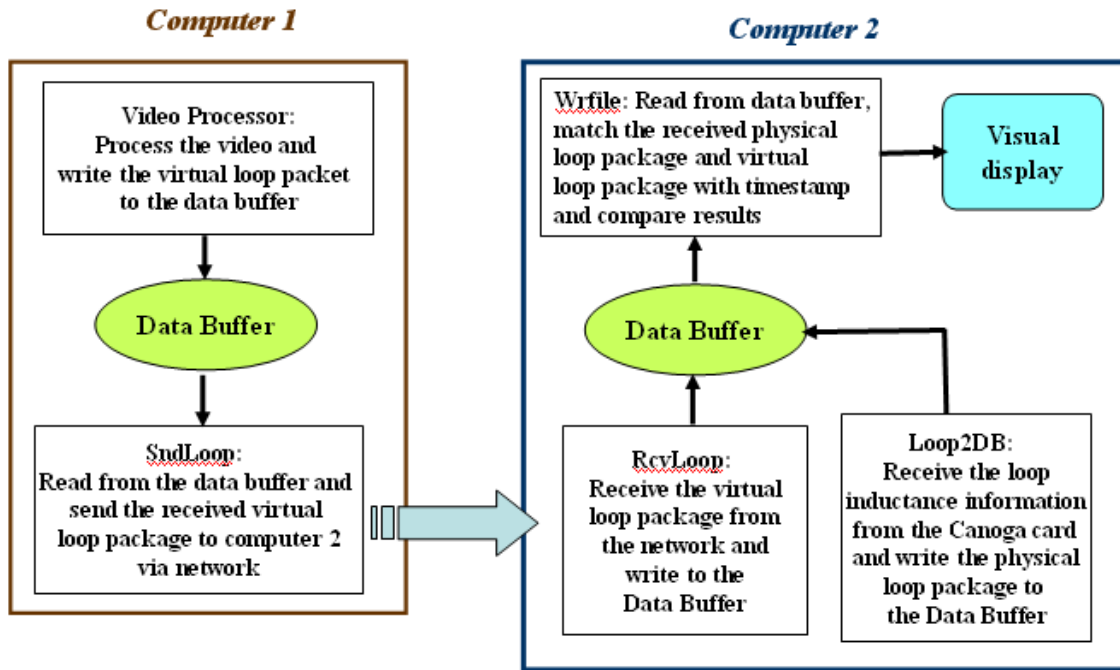


Figure 6. Software structure and interaction for the two computer Laptops

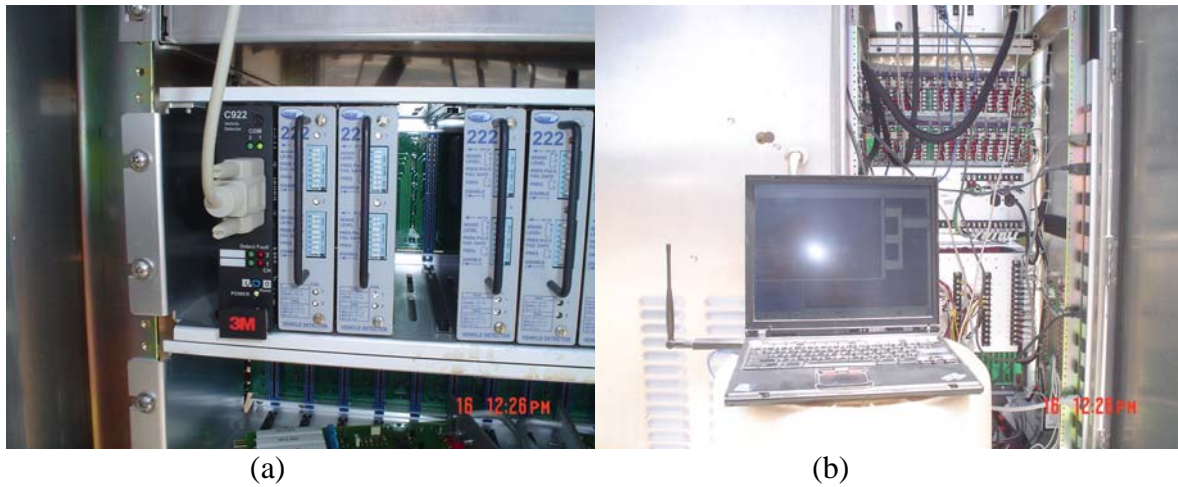


Figure 7. (a) Laptop RS232 serial interface with C922 3M Canoga Card; (b) the laptop also run IEEE802.11b wireless communication with USB port

On the other side, the vision system was set up as illustrated in Figure 2 (a). The camera was mounted on the top of the trailer pole to look downward towards the loop detectors at the RFS test intersection. The camera's intrinsic parameters were estimated by using the Camera Calibration Toolbox for Matlab[®] (http://www.vision.caltech.edu/bouguetj/calib_doc/). The

extrinsic parameters are estimated by a simple external calibration algorithm which uses a single rectangle [12]. USB 300mW WiFi adapter with 9dBi and 5dBi antennas were used for reliable wireless communication with about 800m distance coverage.

Figure 8-(a) and (b) shows the vehicle detection and tracking process. A virtual loop is turned on (as highlighted) when the vehicle ellipse hits the loop rectangle in the world coordinates in the vision system.

We tested the system for around one and half hours at the RFS at UC Berkeley and all of the vehicles passed through the intersection have been detected from both virtual loop system and physical loop system based on the observation. Note that it is a particularly difficult environment for video processing due to heavy moving shadow of trees shaking by a strong wind. The packet buffer described in the previous section makes the synchronization only have an average error of 0.0436 seconds, which means an error of 1.16 meters in space if the vehicle runs at 60 mps. Figure 9 - (a-e) shows the comparison of detections from virtual loops and physical loops when the vehicle is running at different speeds. The red bars represent the virtual loops' on/off information and the blue bars indicate the inductance changes of the physical loops. Each column shows a pair of results for the corresponding virtual and physical loop. The reference origin time in the four sub-figures is exactly the same. The x axis represents the time domain with the number of packets as a unit. Since the vision algorithm works at around 10Hz, each packet is about 0.1s long. In our experiment, two loops were monitored. The exact physical loop size was 2 meters in width by 1.8 meters in length. Considering the vehicle's physical length, the efficient length of a regular sedan is around 6 meters.

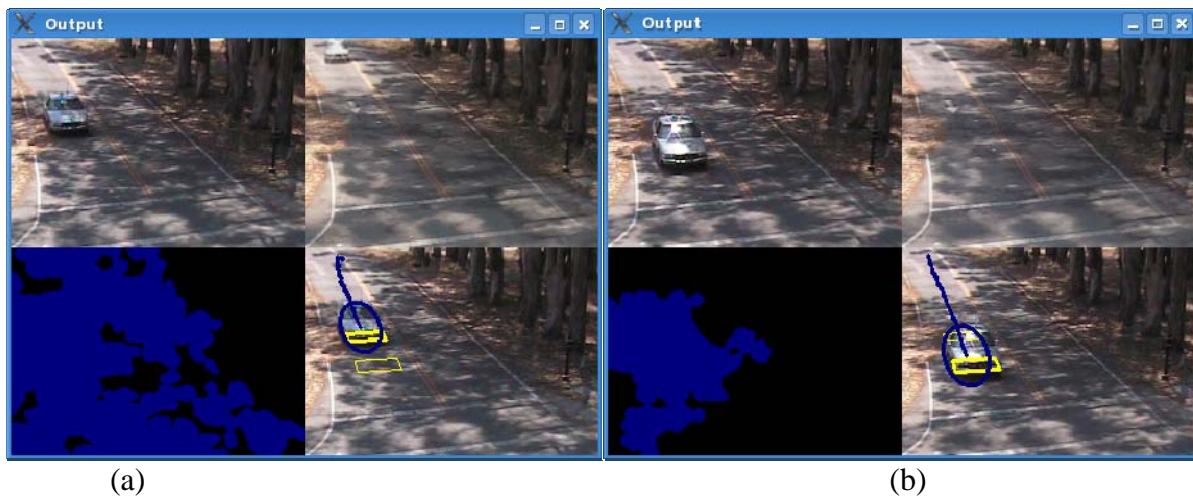


Figure 8. Vehicle tracking with virtual loop matched with real loop on the ground

In Figure 9, we see that at a low speed, such as 5 mph in (a), 15 mph in (b) and 25 mph in (c), the physical loop data is bell-shaped, while at high speed, such as 45 mph in (d) and 50 mph in (e), it is shown as a signal pulse. The vertical is the inductive intensity variation calculated based on variation of the pulse frequency of the loop as a vehicle passing over relative to the inductance of absence of vehicles. It can be observed that as the vehicle speed increases, the occupancy time decreases. At the speed of 50mph, the duration only lasted for two time steps. Even in higher speed, it is expected that, the vehicle over loop can still be

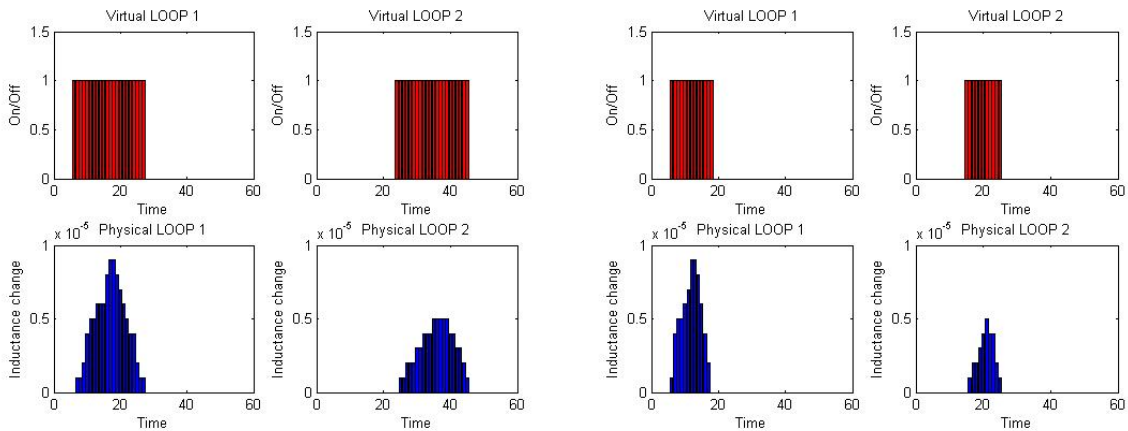
caught due to high frequency pulse signal of the loop circuit. For the vehicle tracking over virtual loop with video, the time instant for virtual loop ON (with a over it) can still be guaranteed due to continuous tracking in advance.

Note that the inductance intensities of the two nearby loops are different even for the same vehicle passing at the same speed. This implies a practical challenge in directly using the inductance data as the only vehicle signature for re-identification over different loops.

5. CONCLUDING REMARKS

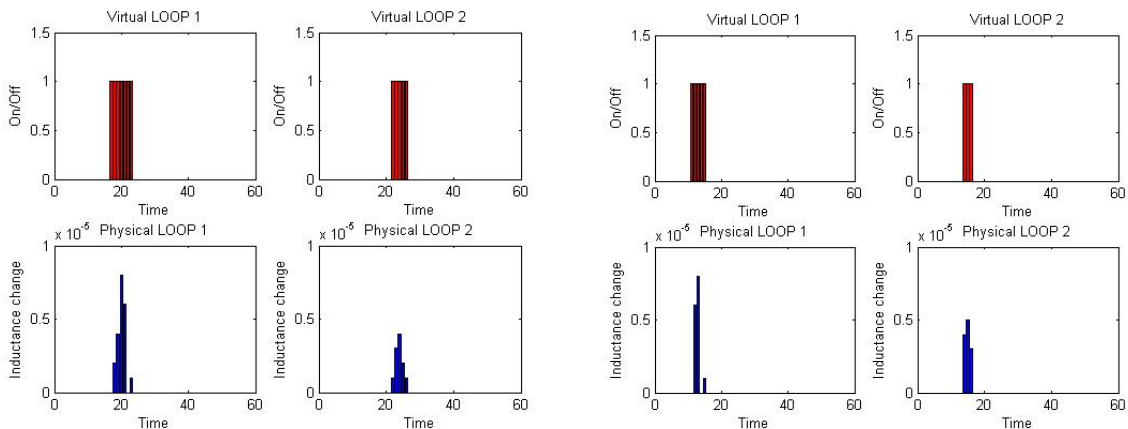
We presented the research and development of a portable tool for systematic loop fault detection at the control cabinet level. Experimental tests up to 50mph of a vehicle speed demonstrated that this concept is feasible in operating in real-time. Continuous tracking of the vehicles from further the upstream of the loop guaranteed that the vehicle is reliably caught over a loop even at a high speed. An effective and reliable synchronization and data communication scheme was presented. Experimental results showed that the matching of the two sensors were reliable -- it had never missed for over 30 tests.

The next step of the research will be in three directions: (a) testing on real freeway (such as the Berkeley Highway Lab test-bed) and interfacing with the 170 controller for practical deployment; (b) testing and improving for multi-lane and multi-vehicle tracking algorithm; and (c) developing algorithm for systematic loop fault detection, data correction, and cleansing at the control cabinet level.



(a) vehicle speed at 5mph

(b) vehicle speed at 15mph



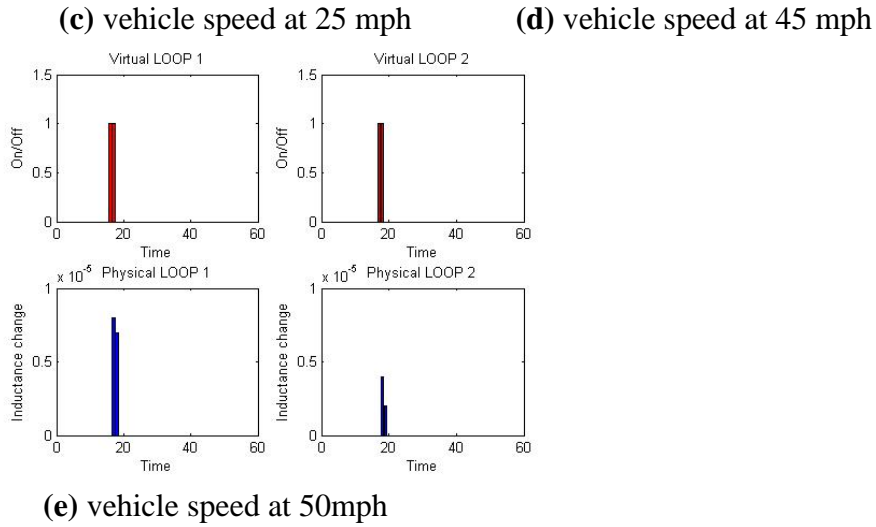


Figure 9 (a-d) Comparison of the virtual loops and physical loops data when the vehicle is running at different speeds. The time duration of the signal in each plot is the duration of the vehicle practically over the loop.

ACKNOWLEDGEMENTS

This work was performed as part of the California PATH Program (TO6327) of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation (Caltrans). The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

Software and hardware support from PATH staff Susan Dickey, Scott Johnston and John Murray, Dave Nelson, John Spring are gratefully acknowledged.

REFERENCES

1. Lu, X. Y., Varaiya, P., Horowitz, R. and Palen, J., Faulty loop data analysis/correction and loop fault detection, accepted to 15th ITS World Congress, November 16-20, 2008, New York
2. Kell, J., Fullerton, I., and Mills, M., Traffic Detector Handbook, Second Edition, Federal Highway Administration, Washington, DC, 1990.
3. Ingram, J., The Inductive Loop Vehicle Detector: Installation Acceptance Criteria and Maintenance Techniques, California Department of Transportation, Sacramento, CA, 1976.
4. Jacobson, L, Nihan, N., and Bender, J., Detecting Erroneous Loop Detector Data in a Freeway Traffic Management System, *Transportation Research Record 1287*, TRB, Washington, DC, 1990, pp 151-166.
5. Cleghorn, D., Hall, F., and Garbuio, D., Improved Data Screening Techniques for Freeway Traffic Management Systems, *Transportation Research Record 1320*, TRB, Washington, DC, 1991, pp 17-31.
6. Nihan, N., Aid to Determining Freeway Metering Rates and Detecting Loop Errors, *Journal of Transportation Engineering*, Vol 123, No 6, ASCE, November/December 1997, pp 454-458.
7. Chen, L., and May, A., Traffic Detector Errors and Diagnostics, *Transportation Research Record 1132*, TRB, Washington, DC, 1987, pp 82-93.
8. Karl Petty, Freeway Service Patrol (FSP) 1.1: The Analysis Software for the FSP Project, California PATH Research Report, UCB-ITS-PRR-95-20
9. Skabardonis, A., Petty, K., Noeimi, H., Rydzewski, D. and Varaiya, P. (1996). I-880 Field Experiment: Data-Base Development and Incident Delay Estimation Procedures, *Transportation Research Record 1554*, TRB, pp 204-212.
10. Coifman, B., Using Dual Loop Speed Traps to Identify Detector Errors, 78th TRB Annual Meeting, Washington, DC., Jan. 1999
11. Kim, Z., Realtime Object Tracking based on Dynamic Feature Grouping with Background Subtraction, Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008.
12. Kim, Z., "Geometry of Vanishing Points and its Application to External Calibration and Real-time Pose Estimation", *Institute of Transportation Studies, Research Reports*, Paper UCB-ITS-RR-2006-5, July 1, 2006.