2016

TraSER: A Traffic Signal Event-Based Recorder

Chenhui Liu
Iowa State University

Anuj Sharma
Iowa State University

Edward Smaglik
Northern Arizona University

Sirisha Kothuri
Portland State University, skothuri@pdx.edu

Citation Details
Liu, C., Sharma, A., Smaglik, E., & Kothuri, S. (2016). TraSER: a traffic signal event-based recorder. SoftwareX, 5, 156-162.

Available online at www.sciencedirect.com
ScienceDirect
SoftwareX 5 (2016) 156–162
SoftwareX
www.elsevier.com/locate/softx

Original software publication

TraSER: A traffic signal event-based recorder

Chenhui Liu[a,*], Anuj Sharma[a], Edward Smaglik[b], Sirisha Kothuri[c]

[a] *Iowa State University, Ames, IA, USA*
[b] *Northern Arizona University, Flagstaff, AZ, USA*
[c] *Portland State University, Portland, OR, USA*

Received 28 May 2016; received in revised form 17 August 2016; accepted 14 September 2016

Abstract

In the past decades, the demand for high-resolution event-based traffic signal indication and detector data has increased due to the need for the collection and reporting of performance measures. This paper will first lay a groundwork for why this type of data acquisition is important, followed by the introduction of a new low-cost, user-friendly, high-resolution traffic signal event-based recorder—TraSER, with integrated video. This paper describes TraSER's structure, operating principles, and field applications. TraSER allows researchers to be able to collect high-resolution event-based controller data at signalized intersections easily and conveniently. The paper concludes with a discussion on future expansion of TraSER.

Keywords: Signalized intersection; Event; Traffic signal; TraSER

Code metadata

Current software version	TraSER v.1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00047
Legal Code License	GNU general public license
Computing platforms/operating systems	Windows operating system
Installation requirements & dependencies	Visual C++ Redistributable Packages for Visual Studio 2013
Software code language	C++
If available Link to developer documentation/manual	https://github.com/cliu9/TraSER.git
Support email for questions	cliu9@iastate.edu

1. Motivation

Automated collection of performance measures is needed for better management and operation of modern transportation infrastructure. Shaw conducted a survey of state transportation agencies and metropolitan planning organizations, identifying more than 70 commonly used performance measures [1]. It was found that operational-related data were largely collected by Intelligent Transportation Systems (ITS), and that this data collection was critical for evaluating and improving system operations through performance measures. Pickrell and Neumann illustrated how to integrate the performance measures with the transportation agency decision-making in policy and resource allocation [2].

Performance tracking is also critical for the re-creation of faults and troubleshooting of system failures. Most transportation electronic failure events, such as lamp and detector failures, are highly unpredictable and cannot be easily replicated. The presence of automated data capturing tools can substantially reduce the amount of time and effort required for proac-

* Correspondence to: Department of Civil, Construction, and Environmental Engineering, Iowa State University, Ames, IA, 50010, USA.
E-mail address: cliu9@iastate.edu (C. Liu).

http://dx.doi.org/10.1016/j.softx.2016.09.001

tively identifying a problem or troubleshooting its cause. The biggest challenges for tracking performance measures, as noted by Shaw [1], include the overwhelming procedure of data acquisition and analysis, and meeting the diverse performance measure demand from different audiences. In general, developments in automated data acquisition and traffic monitoring systems have been concentrated in the field of freeway networks [3], with less focus given to signalized intersections and their corresponding systems.

Actors such as policy makers and the public who are not directly involved with the intricacies of traffic signal operations typically use aggregate level performance measures. Critical performance measures such as the degree of saturation, queue discharge characteristics, and sensor faults are required by traffic engineers to troubleshoot their systems. Event based data provides them the time of detection occurrence and phase change events with a high resolution (1/10th of a second) to help them effectively identify the causes of any operational problems. Traffic engineers need tools to ascertain the status of all signal phases and detector status (on/off) over a period of time to evaluate system performance during that duration. In recent years, researchers [3–10] have developed several real-time performance measures using this data to evaluate the level of service, quality of progression, detector faults, and other metrics of performance at signalized intersections. Since state of the art controllers use only digital on or off information for signal phases and detection inputs, event based data can easily be used to reproduce the state of any phase or detector at any point in time over the period which data is collected. Event based data implies that only the signal and detector status and time at which there is a status change is recorded. This significantly increases the efficiency of the collection and storage of digital phase and detector information.

However, most commercially available traffic signal controllers and performance measure systems report aggregated data on items of interest such as volume, occupancy, splits, and others [11–13]. Aggregation is done to varying degrees, with bins of 5, 15, and 60 min typical in the field. This aggregated information is useful for analysis at higher levels, but does not provide enough detail for fault recreation and troubleshooting [4,14,15]. In the last decade, researchers have focused on developing event based data acquisition systems that have the capability of generating high level performance measures as well as enough data resolution for fault recreation, trouble shooting and signal fine tuning [3–5]. This data can later be aggregated for higher level analysis.

Texas Transportation Institute (TTI) developed a Traffic Signal Performance Monitoring System (TSPMS), which included a Traffic Controller Interface Device (CID), a Traffic Signal Event Recorder (TSER), and a Performance Measure Report Generator (PMRG) [5]. Controllers were connected to TSER via CID. Later, based on TSPMS, Liu et al. developed a new high resolution traffic signal data collection system named SMART-SIGNAL (Systematic Monitoring of Arterial Road Traffic Signals) by expanding the data collection from a single intersection to multiple intersections [3,13]. However, these systems were complex and expensive to operate due to the usage of a CID. Rhodes et al. collected event-based signal & detector data and concurrent video stream for evaluating the accuracy of video vehicle detection in Noblesville, Indiana, but gave no details about the data logger technology [9]. The event logging function of ASC/3 controller, which was developed through a joint effort of Econolite and Purdue University, was used to collect event-based signal and detector data in 2007 [4]. A similar system was also developed in Arizona [10]. Sharma et al. also built a data acquisition system in Lincoln, Nebraska in 2008 with the costly commercial Wonderware software [16].

Generally speaking, although high resolution event-based traffic signal and detector data are desired in many aspects, only limited progresses have been made in developing open source data collection tools. The existing tools are not widely used due to either the high cost or complex operation. This study would introduce an open-source, user-friendly, and traffic camera integrated traffic signal event-based recorder (TraSER).

2. Software description

2.1. Theoretical basis

The National Transportation Communications for Intelligent Transportation System Protocol (NTCIP) defines many objects for actuated traffic signal controller (ASC) units using the OBJECT-TYPE macros specified in RFC 1212 [17]. These objects include phase parameters, detector parameters, ring parameters, coordination parameters, and others that can be exchanged between controllers and computers via Simple Transportation Management Protocol (STMP) [18] when computers send specified commands to controllers. TraSER obtains the signal and detector parameters by intercepting the exchanged data. The current version of TraSER can monitor red, yellow, and green indication statuses of 8 vehicle phases, pedestrian crossing signal statuses of 8 phases, and detection statuses of 64 detectors.

2.2. Structure of TraSER

TraSER is a portable application developed with C++ in Visual Studio 2013. It is designed to run on any NTCIP compliant 2070 ATC controller. To utilize TraSER, a controller should be connected to a computer through an Ethernet connection (this can either be a crossover cable connected directly to the controller, or through any IPv4 addressable network). TraSER consists of two modules: the signal and detector data collection module, and the screen capturing module. Fig. 1 presents a programmatic overview of TraSER and Fig. 2 shows the overall graphic user interface (GUI) for the program. Module I collects signal and detector event data from the controller. All the event data are archived in text files by hour of clock time as shown in Table 1 as well as displayed in a GUI (the left form in Fig. 2). Meanwhile, the right four camera images in Fig. 2 show the synchronous traffic stream of four approaches of the intersection. Module II implements a screen capturing utility to synchronously record both the visualized event data and traffic camera data. Details of each module are described in Fig. 1.

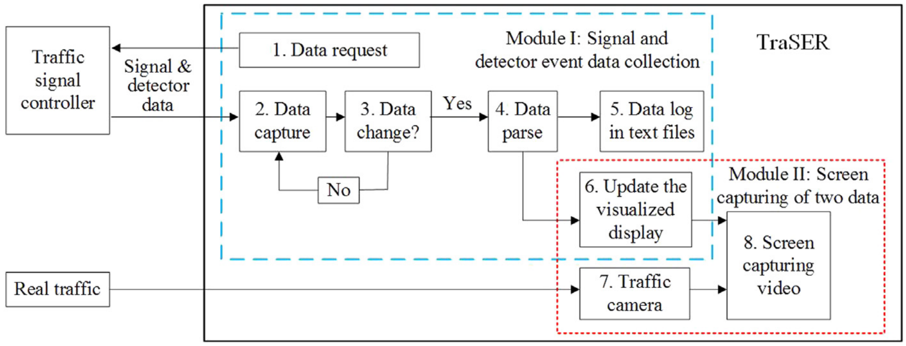

Fig. 1. Data flow diagram for TraSER.

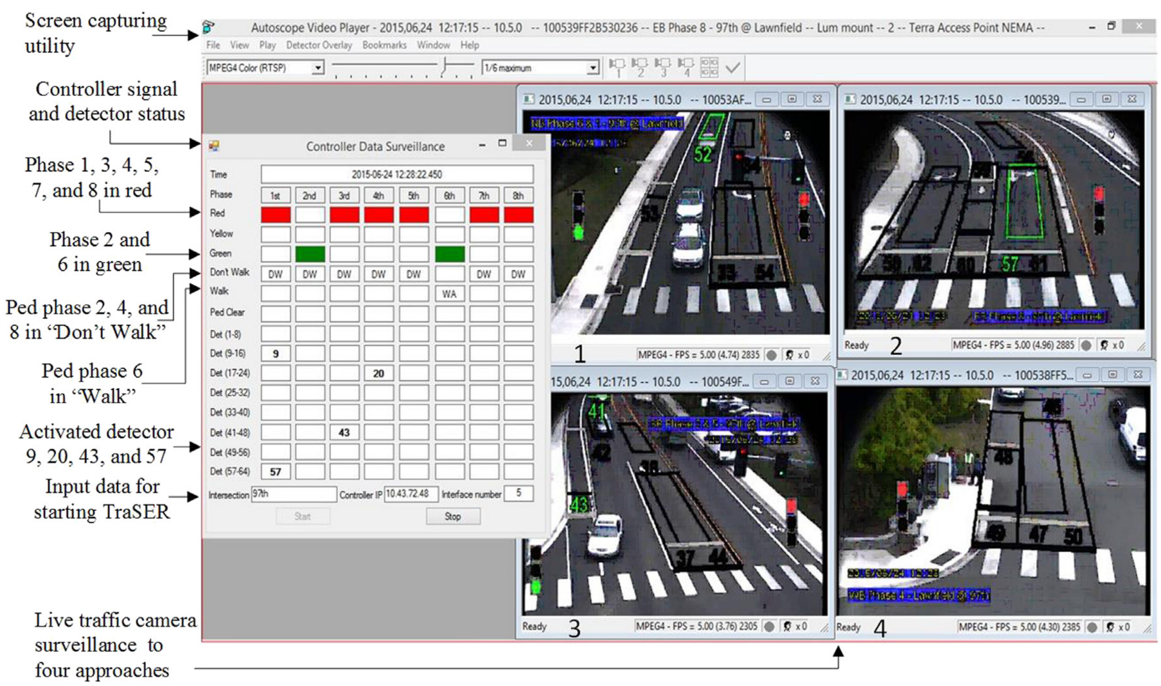

Fig. 2. A screenshot of the running TraSER (Note: there are only 4 pedestrian phases in this example, 2, 4, 6, and 8.)

Table 1
Data log file.

System date	System time	Data type	No.	Status (1, on; 0, off)
6/25/2015	09:47:02.619	R	1	1
6/25/2015	09:47:02.619	G	2	1
6/25/2015	09:47:02.619	G	6	1
6/25/2015	09:47:02.619	DW	3	1
6/25/2015	09:47:02.619	DW	4	1
6/25/2015	09:47:02.619	DET	6	1

2.2.1. *Module I: signal and detector data collection*

Module I is the core of TraSER. From the view of function composition, it consists of two independent functions. Function 1 is responsible for sending data request commands to the controller. Since the controller only responds one time when it receives a data request, Function 1 needs to continuously send the commands to get real-time traffic event data. Function 2 continuously receives, records, and displays the feedback data from the target controller. It includes 5 sub-modules. The pseudo codes of this part are shown in Fig. 3. The data collection frequency is around 60 Hz in lab test when collecting the vehicle and pedestrian signals of 8 phases, and statuses of 64 vehicle detectors.

Referring to Figs. 1 and 3, the roles of 6 sub-modules of Module I are introduced as follows.

- **Data request**: Send data request commands to the controller through the User Datagram Protocol (UDP).
- **Data capture**: Capture the UDP packets containing the requested data sent back from the controller to the computer. This module is developed based on WinPcap, an open source library for packet capture and network analysis on Windows [19].
- **Data change detection**: Compare each new received UDP packet with last record to see if an event has happened, i.e either signal or detector data have changed. If not, the packet would be ignored. Otherwise, TraSER will continue to next step to parse the data.
- **Data parse**: Parse the received data to identify and record the changed items. Meanwhile, the raw hexadecimal UDP data is converted into binary format for analysis.

```
Inputs: the controller IP address and interface number
Outputs: the data record in text and video
Procedure:
Function 1: data request
1. Send data request command
2. If successfully send data request command then
3.      Return to Step 1
4. Else pause 50 milliseconds
5.      Return to Step 1

Function 2: data reception
1. Capture UDP packets on wire
2. For each packet do
3.          If the packet is from the target controller and interface then
4.               If the packet is the 1st one then
5.                    Parse the data from hex format into binary format
6.                    Record the binary data in text files
7.                    Display the binary data on TraSER screen
8.                    End if
9.               Else if new data! = last data then
10.                        Parse the data from hex format into binary format
11.                        Record the binary data in text files
12.                        Update the data on TraSER screen
13.                        End if
14.                  Else discard the packet
15.         Else discard the packet
16. End for
```

Fig. 3. The operating principles of signal and detector data collection of TraSER.

• **Data log**: The binary data are recorded in text files along with the timestamp by hour of clock time as shown in Table 1. The 1st column and 2nd column show the date and time of events, respectively. The 3rd column shows the event types, including vehicle signal, pedestrian signal, and detector call. The vehicle signals include Red (R), Yellow (Y), and Green (G). The pedestrian signals include Don't Walk (DW), Walk (WA), and Pedestrian Clearance (PC). Detector call is abbreviated as "Det". The 4th column shows the serial numbers of signals or detectors. The last column displays the statuses of signals or detectors. A '1' means that a signal or detector call has turned on, and a '0' means that a signal or detector call has turned off. For example, the 1st record in Table 1 shows that vehicle phase 1 became red at 09:47:02.609, 06/25/2015, and the 6th record shows that the detector 6 was activated at 09:47:02.609, 06/25/2015.

• **Data visualization display**: The logged data are concurrently displayed in a form as shown in the left side of Fig. 2, where users are able to directly observe the statuses of traffic signals and detectors in real time. As is shown in Fig. 2, at that moment, phases 2 and 6 were green, and the rest of the phases were red. Pedestrian phase 6 was in "Walk" status. The rest of the pedestrian phases were in "Don't Walk" status. Meanwhile, detectors 9, 20, 43 and 57 were activated.

2.2.2. Module II: screen capturing of controller and traffic camera data

When the controller data collected by Module I are visually displayed on screen, the synchronous traffic camera surveillance is also displayed. Any digital or analog camera surveillance program can be used here. TraSER records the controller and camera data with a third party open source tool—HyCam [20]. HyCam 2 saves the video into Audio–Video Interleaved (AVI) movie files. The video record starts as soon as TraSER is started. The videos can be replayed to corroborate instances observed in the event based data log.

2.3. Characteristics of TraSER

The main characteristics of TraSER include:

• **High sampling rate**: The average sampling rate of 60 Hz is much higher than the requirement of logging event data with a tenth-of-a-second resolution for analyzing performance measures in [21].

• **High time stamp accuracy**: TraSER records the data at millisecond level.

• **Data record in both text and visualized form**.

• **Synchronous video record of traffic camera provides for ground truth of traffic data**.

• **Reliable**: The unidirectional transmission mechanism of signal and detector data from controllers to computers avoids the interruptions of computers to controllers.

• **Easy operation**: TraSER is a portable application with a user-friendly GUI.

• **Low cost**: TraSER is open source software without special requirements for computers.

• **Easy extension of functions**: (1) TraSER can be easily extended to record more parameters, such as ring and coordination parameters, and more phases; (2) TraSER can be extended to have the capacity of setting parameters of controllers, such as placing detector calls. With the set functions, TraSER can be connected to a microscopic transportation simulation software, such as PTV VISSIM [22], and traffic signal controllers to run hardware-in-the-loop (HIL) or software-in-the-loop (SIL) simulations.

A comparison of TraSER and another three event data collection tools mentioned above, i.e., TSER [3,5,13], ASC/3 controller data logger [4,10], and Wonderware [16], are shown in Table 2.

3. Illustrative examples

Oregon Department of Transportation (ODOT) is interested in the performance of non-intrusive vehicle detection sources, such as video and radar, as some of their operational characteristics are desirable. However, the performance of these devices compared to inductive loops varies. To address this, ODOT launched a project to compare the performance of different detection technologies and their impacts on traffic signal operations. Count and occupancy are the two parameters that are widely used in several traffic adaptive controls to manipulate the active timing plan. Part of this project involves field data collection using TraSER, to monitor the operational differences between loop detectors and video detectors surveilling the same area. Fig. 4 shows four pairs of detectors at the intersection of Wilsonville Rd and Town Center Loop in Portland, Oregon. Each pair of detectors consists of one loop detector and one video detector, which are installed at the same location to collect traffic data over the same spatial area. The purple polygons show the detection areas of detectors. TraSER was used to collect the detector data from May 11th to June 24th in 2015. A total of more than 3 million data records were collected without any software failure. The collected detector data were aggregated into 2-minute counts and occupancies, common inputs

Table 2
Comparison of TraSER and other traffic event data collection tools.

Name	TSER	ASC/3 controller data logger	Wonderware	TraSER
Users	[3,5,13]	[4,10]	[16]	NA
Compatible controllers	NEMA TS-1/TS-2 controller	Econolite ASC/3 controller	No Additional Hardware/Software Needed	All 2070 ATC that are NTCIP compliant
Sampling frequency	50–67 Hz	Internal	15 Hz	60 Hz
Time stamp accuracy	Millisecond	10th of a second	Millisecond	Millisecond
Visual display of traffic event data?	No	No	Yes	Yes
Synchronous video record function?	No	No	Yes	Yes
CID needed?	Yes	No	No	No
Open source?	Can be requested	No	No	Yes
Cost	High	Included in newer version of controller (cannot be added to older controllers)	Very high	Low

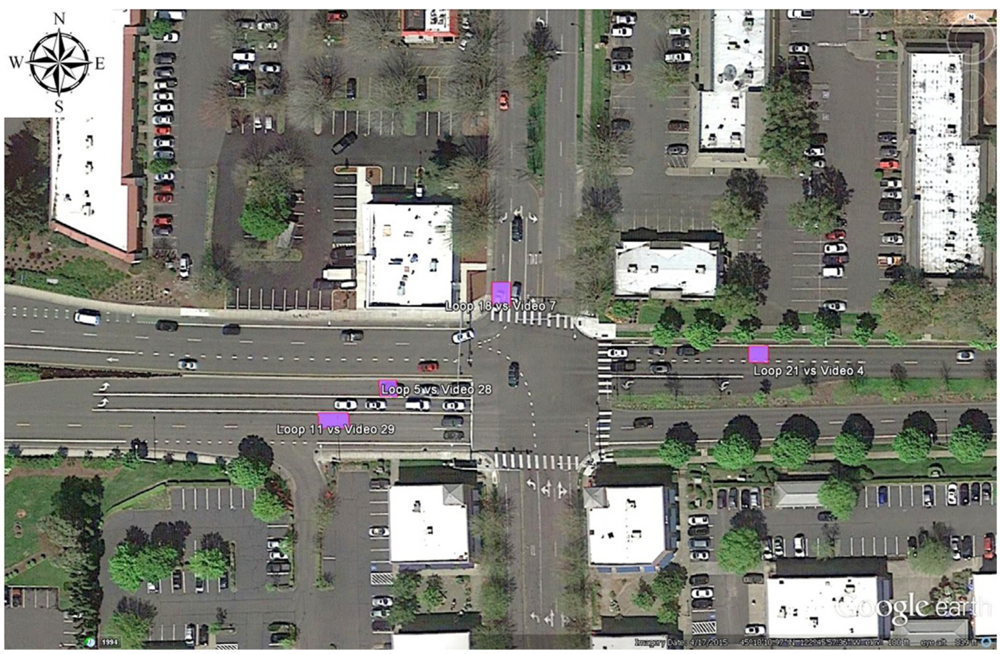

Fig. 4. Layout of the loop detectors and video detectors at the intersection of Wilsonville Rd and Town Center Loop.
Source: "Wilsonville, OR." 45°18′10.97″ N and 122°45′57.36″ W. GOOGLE EARTH. April 17, 2015. Accessed by November 9, 2015.

used by adaptive control algorithms. Fig. 5 shows results of a hypothesis test comparing the counts observed using video and inductive loop detectors. The Kolmogorov–Smirnov Test, where the Null hypothesis is that the compared distributions are the same, is used here, and it can be seen that there is a statistically significant difference in the 2-minute counts observed by video and loop detectors. Similarly, Fig. 6 depicts that there is a statistically significant difference in occupancy observed by video and loop detectors of the same size. It is found that video detectors have smaller counts, but larger occupancies than loop detectors, which is mainly because video detectors often cannot differentiate consecutive vehicles in heavy traffic. Hence this data indicates that there may be a need to adjust the traffic signal timing parameter settings in the traffic signal controller when using video detectors instead of the loop detectors.

4. Conclusion

In the past decades, the demand for high resolution event-based traffic signal and detector data has increased as it is a very useful tool in generating high quality transportation performance measures. However, the availability of low-cost and easy-to-use data collection tools has been lacking. This paper introduces a portable application named TraSER to collect such data. TraSER has many advantages, such as low-cost, high time stamp accuracy, user-friendly operation, synchronous video record of controller and traffic camera data, high reliability, and others. Successful data collection at 4 different sites for 45 days proves reliability of TraSER for uninterrupted long term field data collection.

Currently, TraSER is designed to only collect data. Future versions would add data analysis functions. Another version

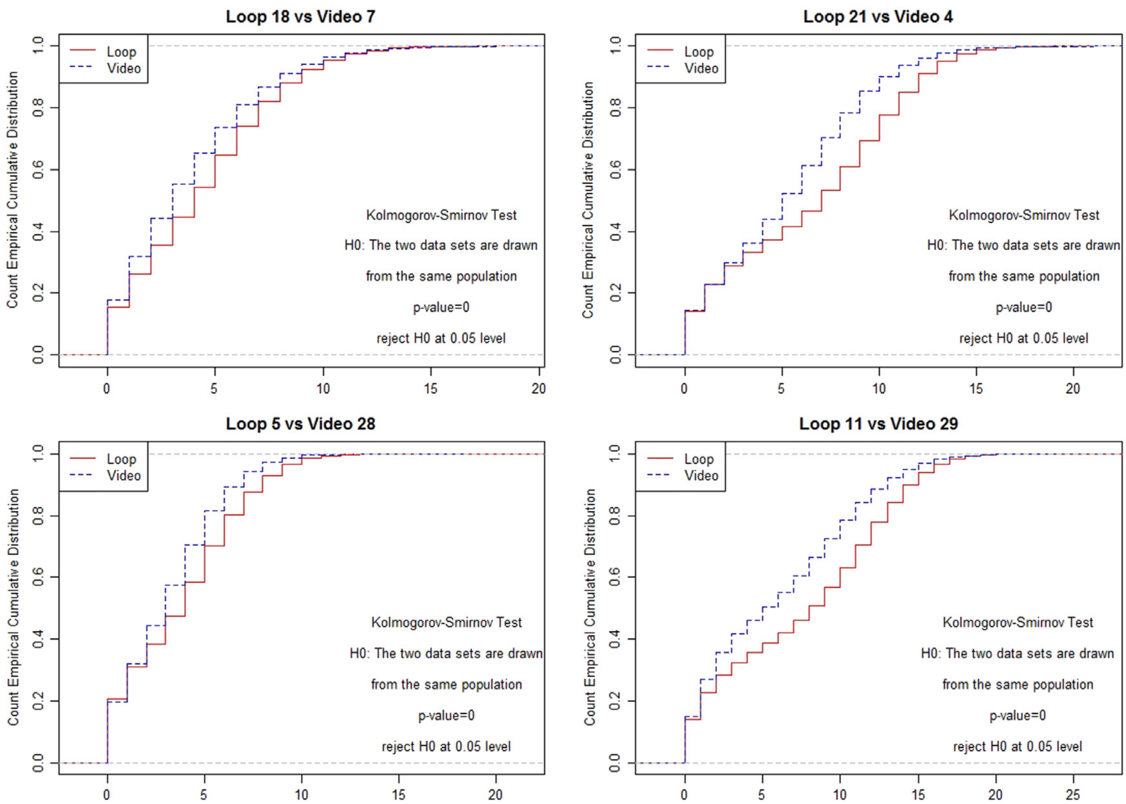

Fig. 5. Kolmogorov–Smirnov test results of the counts.

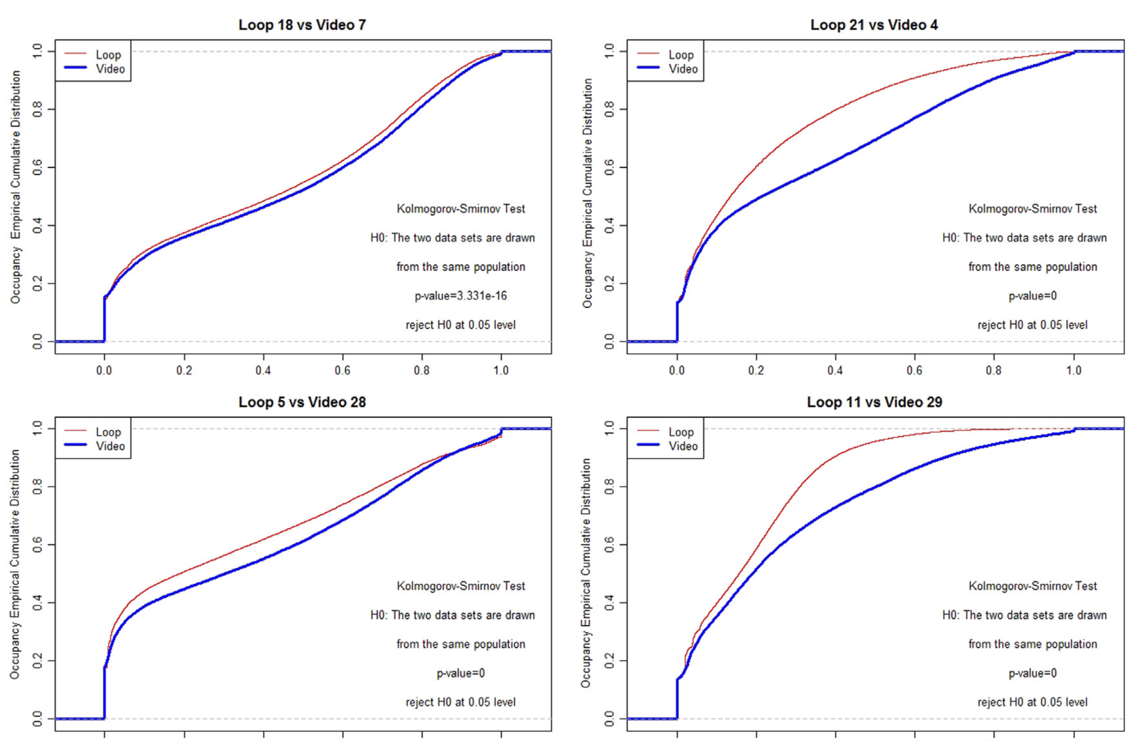

Fig. 6. Kolmogorov–Smirnov test results of the occupancies.

would add the parameter set functions. Combining with microscopic simulation software, such as VISSIM, it can realize HIL and SIL. Additionally, the controller data collection part of TraSER can be used as an independent data collection tool if there is no need for traffic camera record.

Acknowledgments

The authors of this paper would like to thank Mr. Dan Carson, Northwest Signal, for providing help in utilizing NWS software, and the Oregon Department of Transportation for funding this work.

References

[1] Shaw T. NCHRP SYNTHESIS 311: Performance measures of operational effectiveness for highway segments and systems. Transportation Research Board of the National Academies 2003.

[2] Pickrell S, Neumann. L. Use of performance measures in transportation decision making. In: Conference proceedings of performance measures to improve transportation systems and agency operations, Vol. 26, Irvine, CA; 2000. pp. 17–33.

[3] Liu HX, Ma W, Wu X, Hu. H. Development of a real-time arterial performance monitoring system using traffic data available from existing signal systems. Report No. MN/RC 2009-01, Minnesota Department of Transportation, 2008.

[4] Smaglik EJ, Sharma A, Bullock DM, Sturdevant JR, Duncan G. Event-based data collection for generating actuated controller performance measures. J Transport Res Rec 2007;2035:97–106. http://dx.doi.org/10.3141/2035-11.

[5] Balke KN, Charara H, Parker R. Development of a traffic signal performance measurement system (TSPMS). Report No. FHWA/TX-05/0-4422-2. Texas Department of Transportation; 2005.

[6] Sharma A, Bullock DM, a Bonneson J. Input-output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections delay and maximum queue length at signalized intersections. J Transport Res Rec 2007;2035:69–80. http://dx.doi.org/10.3141/2035-08.

[7] Smaglik EJ, Bullock DM, Sharma A. Pilot study on real-time calculation of arrival type for assessment of arterial performance. J Transp Eng 2007; 133(7):415–22. http://dx.doi.org/10.1061/(ASCE)0733-947X(2007)133:7(415).

[8] Day CM, Smaglik EJ, Bullock DM, Sturdevant J. Real-time arterial traffic signal performance measures. Report No. FHWA/IN/JTRP-2008/9, Indiana Department of Transportation; 2008.

[9] Rhodes A, Bullock D, Sturdevant J, Clark Z, Candey D. Evaluation of the accuracy of stop bar video vehicle detection at signalized intersections. J Transport Res Rec 2005;1925(1):134–45. http://dx.doi.org/10.3141/1925-14.

[10] Smaglik EJ, Davis Z, Steele RC, Nau W, Roberts Ca. Supplementing signalized intersection infrastructure to provide automated performance measures with existing video detection equipment. Transport Res Rec: J Transport Res Rec 2011;2192:77–88. http://dx.doi.org/10.3141/2192-07.

[11] Jie L, Van Zuylen HJ. Loop detector data error diagnosing and interpolating with probe vehicle data. In: 93rd annual meeting of transportation research board, Washington, DC, USA; 2014.

[12] Weijermars W, Van Berkum E. Detection of invalid loop detector data in urban areas. J Transport Res Rec 2006;1945(1):82–8. http://dx.doi.org/10.3141/1945-10.

[13] Liu HX, Zheng J, Hu H, Sun J. Research implementation of the SMART SIGNAL system on trunk highway (TH) 13., Report No. MN/RC 2013-06. Minnesota Department of Transportation; 2013.

[14] Dixon M. Development of new actuated signalized intersection performance measurement methodologies using traffic controllers. US Report No. N08-14. Department of Transportation; 2008.

[15] Balke KN, Herrick C. Potential measures of assessing signal timing performance using existing technologies. Report No. FHWA/TX-04/0-4422-1. Texas Department of Transportation; 2004.

[16] Sharma A, Schmitz J. The effects of pedestrian countdown timers on safety and efficiency of operations at signalized intersections. Report No. SPR-P1 (09) P332. Nebraska Department of Roads; 2011.

[17] American Association of State Highway and Transportation Officials (AASHTO), Institute of Transportation Engineers (ITE), and National Electrical Manufactures Association (NEMA). NTCIP 1202 v01.07. 2005.

[18] American Association of State Highway and Transportation Officials (AASHTO), Institute of Transportation Engineers (ITE), and National Electrical Manufactures Association (NEMA). NTCIP 1103 v01.27. 2009.

[19] WinPcap Documentation 4.0.2. http://www.winpcap.org/docs/docs_40_2/html/main.html [23 Feb 2015].

[20] HyperCam™. http://www.hyperionics.com/index.asp [3 March 2015].

[21] Bullock D, et al. Help traffic engineers manage data to make better decisions: automated traffic signal performance measures. ITE J 2014; 84(3):33–9.

[22] PTV VISSIM 7 USER MANUAL, 2014. PTV AG, Karlsruhe, Germany.