Detecting Errors and Imputing Missing Data for Single-Loop Surveillance Systems

Chao Chen, Jaimyoung Kwon, John Rice, Alexander Skabardonis, and Pravin Varaiya

Single-loop detectors provide the most abundant source of traffic data in California, but loop data samples are often missing or invalid. A method is described that detects bad data samples and imputes missing or bad samples to form a complete grid of clean data, in real time. The diagnostics algorithm and the imputation algorithm that implement this method are operational on 14,871 loops in six districts of the California Department of Transportation. The diagnostics algorithm detects bad (malfunctioning) single-loop detectors from their volume and occupancy measurements. Its novelty is its use of time series of many samples, instead of basing decisions on single samples, as in previous approaches. The imputation algorithm models the relationship between neighboring loops as linear and uses linear regression to estimate the value of missing or bad samples. This gives a better estimate than previous methods because it uses historical data to learn how pairs of neighboring loops behave. Detection of bad loops and imputation of loop data are important because they allow algorithms that use loop data to perform analysis without requiring them to compensate for missing or incorrect data samples.

Loop detectors are the best source of real-time freeway traffic data today. In California, these detectors cover most urban freeways. But loop detector data contain many holes (missing values) or bad (incorrect) values and require careful cleaning to produce reliable results. Bad or missing samples present problems for any algorithm that uses the data for analysis. Therefore, one must both detect when data are bad and throw them out and then fill holes in the data with imputed values. The goal is to produce a complete grid of reliable data. Analyses that use such a complete data set can be trusted.

Bad data must be detected from the measurements themselves. The problem has been studied by FHWA, the Washington State Department of Transportation, and others. Existing algorithms usually work on the raw 20-s or 30-s data and produce a diagnosis for each sample. But it is difficult to tell if a single 20-s sample is good or bad unless it is very abnormal. Fortunately, loop detectors do not just give random errors—some loops produce reasonable data all the time, while others produce suspect data all the time. By examining a time series of measurements, one can readily distinguish bad behavior from good. The diagnostics algorithm presented here examines a day's worth of samples together, producing convincing results.

Once bad samples are thrown out, the resulting holes in the data must be filled with imputed values. Imputation with time series analysis has been suggested, but these imputations are effective only for short periods of missing data; linear interpolation and neighborhood averages are natural imputation methods, but they do not use all the relevant data that are available. The imputation algorithm presented here estimates values at a detector by using data from its neighbors. The algorithm models each pair of neighbors linearly and fits its parameters on historical data. It is robust and performs better than other methods.

C. Chen and P. Varaiya, Department of Electrical Engineering and Computer Science, and J. Kwon and J. Rice, Department of Statistics, University of California, Berkeley, CA 94720. A. Skabardonis, Institute of Transportation Studies, University of California, Berkeley, CA 94720-1720.

DESCRIPTION OF DATA

The freeway performance measurement system (PeMS) collects, stores, and analyzes data from thousands of loop detectors in six districts of the California Department of Transportation (Caltrans) (Transacct.eecs.berkeley.edu, *1*). The PeMS database currently has 1 terabyte of data online, and it collects more than 2 GB data per day. PeMS uses the data to compute freeway usage and congestion delays, measure and predict travel time, evaluate ramp-metering methods, and validate traffic theories. There are 14,871 mainline loops in the PeMS database from six Caltrans districts. The results presented here are for mainline loops. Each loop reports the volume $q(t)$, the number of vehicles that cross the loop detector during a 30-s interval t, and occupancy $k(t)$, the fraction of this interval during which there is a vehicle above the loop. Each pair of volume and occupancy observations is called a sample. The number of total possible samples in 1 day from mainline loops in PeMS is therefore (14,871 loops) × (2,880 sample per loop per day) = 42 million samples. In reality, however, PeMS never receives all the samples. For example, Los Angeles has a missing sample rate of about 15%. While it is clear when samples are missed, it is harder to tell when a received sample is bad or incorrect. A diagnostics test needs to accept or reject samples on the basis of the assumption of what good and bad samples look like.

EXISTING DATA-RELIABILITY TESTS

Loop data errors have plagued their effective use for a long time. In 1976, Payne et al. identified five types of detector error and presented several methods to detect them from 20-s and 5-min volume and occupancy measurements (*2*). These methods place thresholds on minimum and maximum flow, density, and speed and declare a sample to be invalid if they fail any of the tests. Later, Jacobson et al. defined an "acceptable region" in the k-q plane and declared samples to be good only if they fell inside the region (*3*). This is called the Washington algorithm in this paper. The boundaries of the acceptable region are defined by a set of parameters, which are calibrated from historical data or derived from traffic theory.

Existing detection algorithms (*2–4*) try to catch the errors described by Payne et al. (*2*). For example, chattering and pulse breakup cause

q to be high, so a threshold on q can catch these errors. But some errors cannot be caught this way, such as a detector stuck in the off ($q = 0$, $k = 0$) position. Payne's algorithm would identify this as a bad point, but good detectors will also report (0, 0) when there are no vehicles in the detection period. Eliminating all (0, 0) points introduces a positive bias in the data. On the other hand, the Washington algorithm accepts the (0, 0) point, but doing so makes it unable to detect the stuck type of error. A threshold on occupancy is similarly hard to set. An occupancy value of 0.5 for one 30-s period should not indicate an error, but a large number of 30-s samples with occupancies of 0.5, especially during nonpeak periods, points to a malfunction.

The Washington algorithm was implemented in Matlab and tested on 30-s data from two loops in Los Angeles for 1 day on August 7, 2001. The acceptable region is taken from Jacobson et al. (*3*). The data and their diagnoses are shown in Figure 1. Visually, Loop 1 looks good (Figure 1*b*), and Loop 2 looks bad (Figure 1*d*). Loop 2 looks bad because there are many samples with $k = 70\%$ and $q = 0$ as well as many samples with occupancies that appear too high, even during nonpeak periods, and when Loop 1 shows low occupancy. The Washington algorithm, however, does not make the correct diagnosis. Of 2,875 samples, it declared 1,138 samples to be bad for Loop 1 and 883 bad for Loop 2. In both loops, there were many false alarms. This is because the maximum acceptable slope of q/k was exceeded by many samples in free flow. This suggests that the algorithm is very sensitive to thresholds and needs to be calibrated for California. Calibration is impractical because each loop will need a separate acceptable region, and ground truth would be difficult to get.

There are also false negatives—many samples from Loop 2 appear to be bad because they have high occupancies during off-peak times, but they were not detected by the Washington algorithm. This illustrates a difficulty with the threshold method—the acceptable region has to be very large, because there are many possible traffic states within a 30-s period. On the other hand, much more information can be gained by looking at how a detector behaves over many sample times. This is why Loop 1 is easily recognized as good and Loop 2 as bad by looking at their $k(t)$ plots, and this is a key insight that led to the diagnostics algorithm.

PROPOSED DETECTOR DIAGNOSTICS ALGORITHM

Design

The algorithm for loop-error detection uses the time series of flow and occupancy measurements instead of making a decision based on an individual sample. It is based on the empirical observation that good and bad detectors behave very differently over time. For example, at any given instant, the flow and occupancy at a detector location can have a wide range of values, and one cannot rule out most of them; but over a day, most detectors show a similar pattern—flow and occupancy are high in the rush hours and low late at night. Figures 2*a* and 2*b* show typical 30-s flow and occupancy measurements at Vehicle Detector Station 759531. Most loops have outputs that look like this, but some loops behave quite differently. Figures 2*c* and 2*d* give an example of a bad loop (Vehicle Detector Station 759518). This loop has zero flow and an occupancy value of 0.7 for several hours during the evening peak period—clearly, these values must be incorrect. Four types of abnormal time series behavior were found, and these are given in Table 1. Types 1 and 4 are self-explanatory;

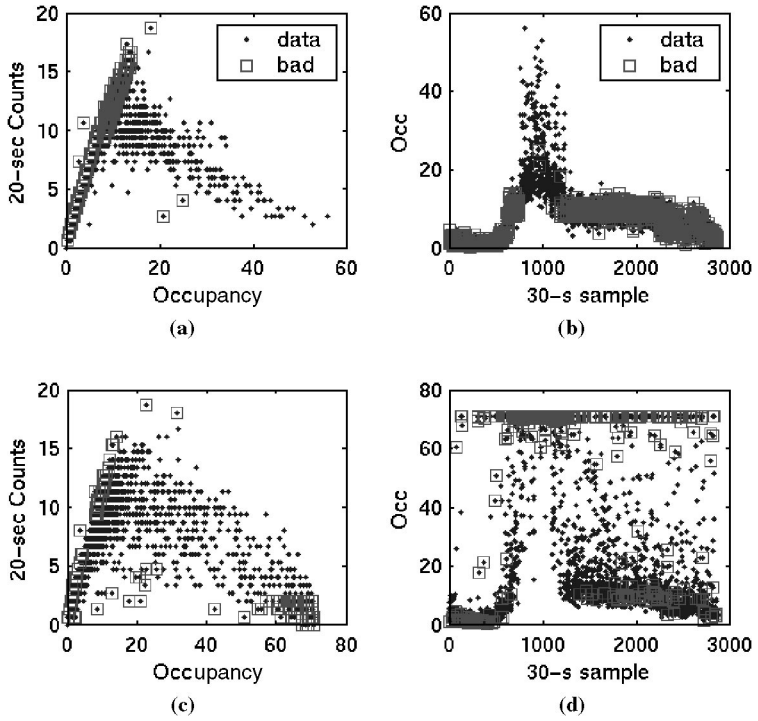

FIGURE 1 Washington algorithm on two loops: Loop 1 (*a*) volume versus occupancy and (*b*) occupancy; Loop 2 (*c*) volume versus occupancy and (*d*) occupancy. Occupancy is in percent. Loops are in Los Angeles, Interstate 5 North, postmile 7.8, Lanes 1 and 2.

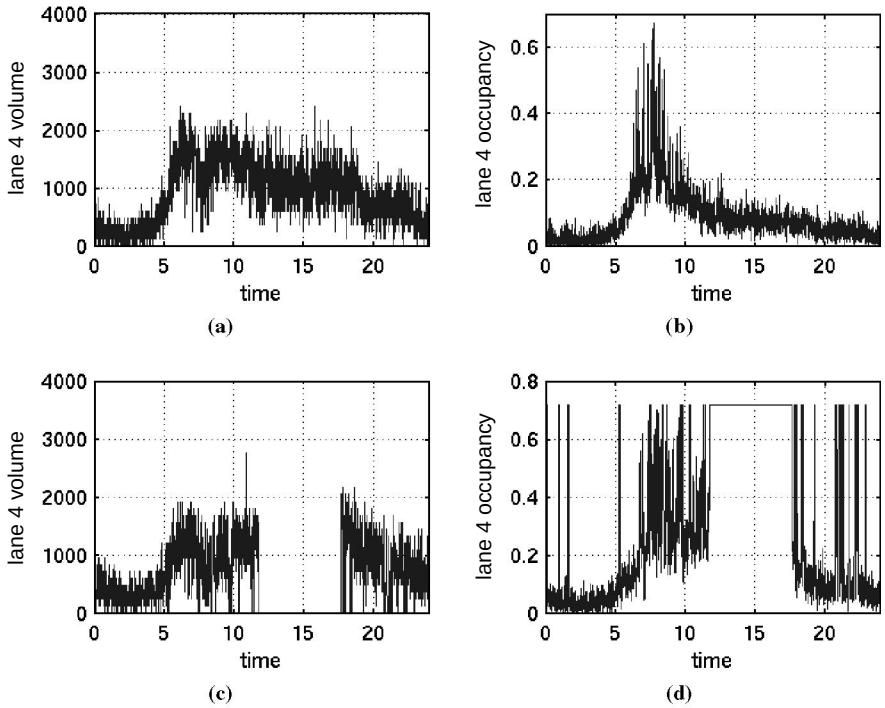

FIGURE 2 (*a*, *c*) Typical and abnormal 30-s flow; (*b*, *d*) occupancy measurements.

Types 2 and 3 are illustrated in Figures 2*c*, 2*d*, and 1*b*. The errors in Table 1 are not mutually exclusive. For example, a loop with all zero occupancy values exhibits both Type 1 and Type 4 errors. A loop is declared bad if it is in any of these categories.

No significant number was found of loops with chatter or pulse breakup, which would produce abnormally high volumes. Therefore, the current form of the detection algorithm does not check for this condition. However, a fifth error type and error check can easily be added to the algorithm to flag loops with consistently high counts.

The daily statistics algorithm (DSA) was developed to recognize error Types 1 through 4. The input to the algorithm is the time series of 30-s measurements $q(d, t)$ and $k(d, t)$, where d is the index of the day and $t = 0, 1, 2, \ldots, 2{,}879$ is the 30-s sample number; the output is the diagnosis $\Delta(d)$ for the dth day: $\Delta(d) = 0$ if the loop is good and $\Delta(d) = 1$ if the loop is bad. In contrast to existing algorithms that operate on each sample, DSA produces one diagnosis for all the samples of a loop on each day.

Only samples between 5:00 a.m. and 10:00 p.m. were used for the diagnostics, because outside this period, it is more difficult to tell the difference between good and bad loops. There are 2,041 30-s samples in this period; therefore, the algorithm is a function of $2{,}041 \times 2 = 4{,}082$ variables. Thus the diagnostic $\Delta(d)$ on day d is a function, $\Delta(d) = f(q(d, a), q(d, a + 1), \ldots, q(d, b), k(d, a), k(d, a + 1), \ldots, k(d, b))$, where $a = 5 \times 120 = 600$ is the sample number at 5:00 a.m. and $b = 22 \times 120 = 2{,}640$ is the last sample number, at 10:00 p.m. To deal with the large number of variables, first reduce them to four statistics, $S_1, \ldots, S_4$, which are appropriate summaries of the time series. Their definitions are given in Table 2, where $S_j(i, d)$ is the jth statistic computed for the ith loop on the dth day. The decision Δ becomes a function of these four variables. For the ith loop and dth day, the decision whether the loop is bad or good is determined according to the rule

$$\Delta_i(d) = \begin{cases} 1 & \text{if } S_1(i, d) > s_1^* \text{ or } S_2(i, d) > s_2^* \text{ or } S_3(i, d) > s_3^* \text{ or } S_4(i, d) < s_4^* \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

TABLE 1 Loop Detector Data Error

Error Type	Description	Likely Cause	Fraction of Loops in District 12
1	Occupancy and flow are mostly zero	Stuck off	5.6%
2	Non-zero occupancy and zero flow, see Figure 2c and 2d	Hanging on	5.5%
3	Very high occupancy, see Figure 1d	Hanging on	9.6%
4	Constant occupancy and flow	Stuck on or off	11.2%
All Errors			16%

TABLE 2 Statistics for Diagnostics

Name	Definition	Description
S_1(i,d)	$\sum_{a\le t\le b} 1(k_i(d,t)=0)$	number of samples that have occupancy = 0
S_2(i,d)	$\sum_{a\le t\le b} 1(k_i(d,t)>0)1(q_i(d,t)=0)$	number of samples that have occupancy > 0 and flow = 0
S_3(i,d)	$\sum_{a\le t\le b} 1(k_i(d,t)>k^*),\ k^*=0.35$	number of samples that have occupancy > k^* (= 0.35)
S_4(i,d)	$(-1)\sum_{x:\,p(x)>0}\hat{p}(x)\log(\hat{p}(x))$, $\hat{p}(x)=\sum_{a\le t\le b}1(k_i(d,t)=x)\Big/\sum_{a\le t\le b}1$	*entropy* of occupancy samples – a well-known measure of the "randomness" of a random variable. If $k_i(d,t)$ is constant in t, for example, its entropy is zero.

where s_j^* are thresholds on each statistic. These four statistics summarize the daily measurements well because they are good indicators of the four types of loop failure given in Table 1. This is seen in the histogram of each statistic displayed in Figure 3. The data were collected from Los Angeles on April 24, 2002. The distribution of each statistic shows two distinct populations. In S_1, for example, there are two peaks, at 0 and 2,041. This shows that there are two groups of loops—one group of about 4,700 loops has very few samples that report zero occupancy, and another group of about 300 reports almost all zeros. The second group is bad, because they have Type 1 error. Since all four distributions are strongly bimodal, Equation 1 is not very sensitive to the thresholds s_j^*, which just have to be able to separate the two peaks in the four histograms in Figure 3. The default thresholds are given in Table 2. The only other parameters for this model are the time ranges and the definition of S_3, where an occupancy threshold of 0.35 is specified. The DSA uses a total of seven parameters, listed in Table 3. They work well in all six Caltrans districts.

Performance

The DSA algorithm is implemented and run on PeMS data. The last column in Table 1 shows the distribution of the four types of error in District 12 (Orange County) for 31 days in October, 2001. Because the ground truth of which detectors are actually bad is not available, the performance of this algorithm must be verified visually. Fortunately, this is easy for most cases, because the time series show distinctly different patterns for good and bad detectors. A visual test was performed on loops in Los Angeles, on data from August 7, 2001. There are 662 loops on Interstate 5 and Interstate 210, of which 142 (21%) were declared to be bad by the algorithm. The plots of occupancy were then checked manually to verify these results. Fourteen loops were found that were declared good, but their plots suggested they could be bad. This suggests a false negative rate of 14/(662 – 142) = 2.7%. There were no false positives. This suggests that the algorithm performs very well.

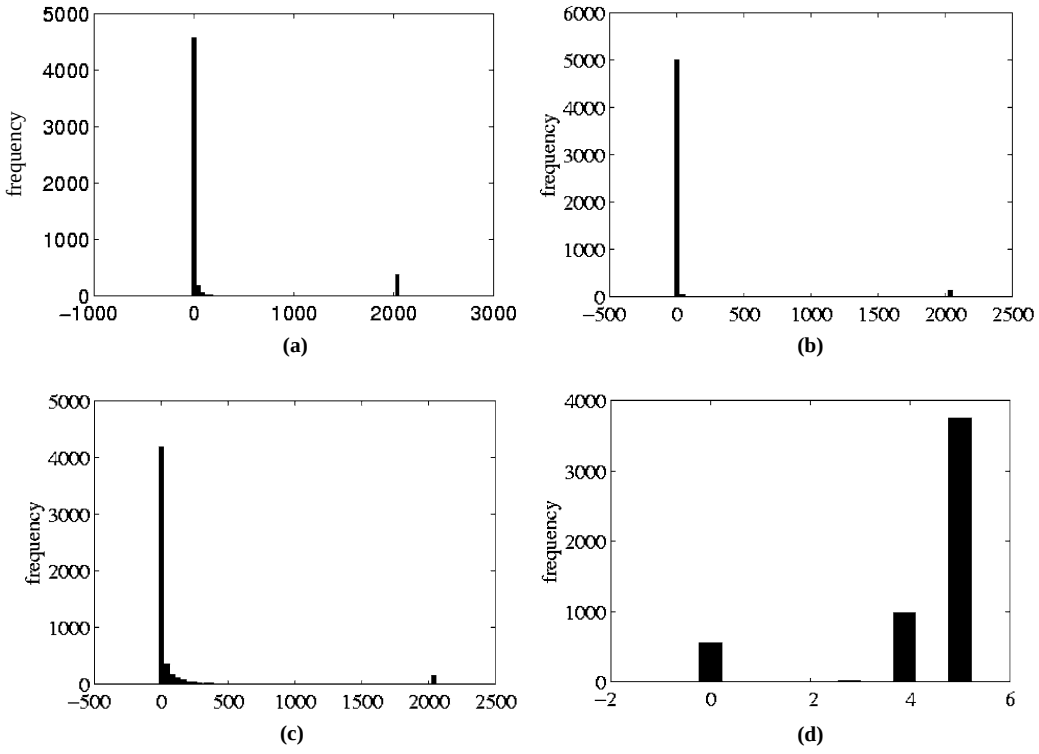

FIGURE 3 Histograms of (*a*) S_1, (*b*) S_2, (*c*) S_3, and (*d*) S_4.

TABLE 3 Parameters of Daily Statistics Algorithm and Default Settings

Parameter	Value
k^*	0.35
s_1^*	1200
s_2^*	50
s_3^*	200
s_4^*	4
a	5 a.m.
b	10 p.m.

Real-Time Operation

The described detection algorithm gives a diagnosis on samples from an entire day, but real-time detection—the validity of each sample as it is received—is also of interest. Therefore, a decision $\hat{\Delta}_i(d, t)$, where d is the current day and t is the current sample time, is wanted. Use the simple approximation

$$\hat{\Delta}_i(d, t) = \Delta_i(d - 1) \tag{2}$$

where Δ_i is defined in Equation 1. Equation 2 has two consequences. First, a loop is declared good or bad for an entire day. As a result, some flexibility is lost because good data from a partially bad loop may be thrown away (this point is discussed in the conclusions section). Second, there is a 1-day lag in the diagnosis, which introduces a small error. The probability of loop failure given the loop status on the previous day was estimated, and Equation 2 was found to be true 98% of the time. Therefore, it is a good approximation.

IMPUTATION OF MISSING AND BAD SAMPLES

Need for Imputation

The measurement of each detector is modeled as either the actual value or an error value, depending on the status Δ:

$$q_{\text{meas},i}(d, t) = q_{\text{real},i}(d, t)[1 - \Delta_i(d)] + \epsilon_i(d, t)\Delta_i(d)$$
$$k_{\text{meas},i}(d, t) = k_{\text{real},i}(d, t)[1 - \Delta_i(d)] + \phi_i(d, t)\Delta_i(d)$$
$$0 \le t \le 2{,}879 \tag{3}$$

where $q_{\text{meas},i}$ and $k_{\text{meas},i}$ are the measured values, $q_{\text{real},i}$ and $k_{\text{real},i}$ are the true values, and ϵ_i and ϕ_i are error values that are independent of $q_{\text{real},i}$ and $k_{\text{real},i}$. An estimate of the loop status was obtained in Equation 2. It says to discard the samples from detectors that are declared bad. This leaves holes in the data, in addition to the originally missing samples. This is a common problem—at each sample time, the user must determine whether the sample is good. An application that analyzes the data must deal with both possibilities.

An approach to missing data is to predict them by using time series analysis. Nihan modeled occupancy and flow time series as autoregressive moving average processes and predicted values in the near future (*5*); Dailey presented a method for prediction from neighbor loops by using a Kalman filter (*6*). In the case here, the errors do not occur randomly but persist for many hours and days. Time series predictions become invalid very quickly and are inappropriate in such situations. An imputation scheme was developed that uses information from good neighbor loops at only the current sample time. This is a natural way to deal with missing data and is used by traditional imputation methods. For example, to find the total volume of a freeway location with four lanes and only three working loops, one may reasonably use the average of the three lanes and multiply it by four. This imputes the missing value by using the average of its neighbors. Linear interpolation is another example. Suppose detector i is bad and is located between detectors j and k, which are good. Let x_i, x_j, x_k be their locations and $x_j < x_i < x_k$; then

$$\hat{q}_i(t) = \frac{(x_i - x_j)q_k(t) + (x_k - x_i)q_j(t)}{x_k - x_j} \tag{4}$$

is the linear interpolation imputation. While these traditional imputation methods are intuitive, they make naive assumptions about the data. The proposed algorithm, on the other hand, models the behavior of neighbor loops better because it uses historical data.

Linear Model of Neighbor Detectors

A linear regression algorithm for imputation is proposed that models the behavior of neighbor loops by using historical data. It was found that occupancies and volumes of detectors in nearby locations are highly correlated. Therefore, measurements from one location can be used to estimate quantities at other locations, and a more accurate estimate can be formed if all the neighboring loops are used in the estimation. Two loops are defined as neighbors if they are in the same location in different lanes or if they are in adjacent locations. Figure 4 shows a typical neighborhood. Both volume and occupancy from neighboring locations are strongly correlated. Figure 5 shows two pairs of neighbors with linearly related flow and occupancies.

Figure 6 plots the distribution of the correlation coefficients between all neighbors in Los Angeles. It shows that most neighbor pairs have high correlations in both flow and occupancy.

The high correlation among neighbor loop measurements means that linear regression is a good way to predict one from the other. It is also easy to implement and fast to run. The following pairwise linear model relates the measurements from neighbor loops:

$$q_i(t) = \alpha_0(i, j) + \alpha_1(i, j)q_j(t) + \text{noise}$$
$$k_i(t) = \beta_0(i, j) + \beta_1(i, j)k_j(t) + \text{noise} \tag{5}$$

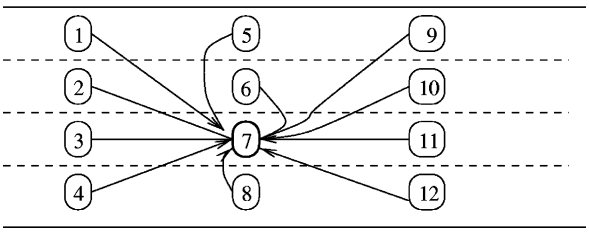

FIGURE 4 Example of neighboring loops.

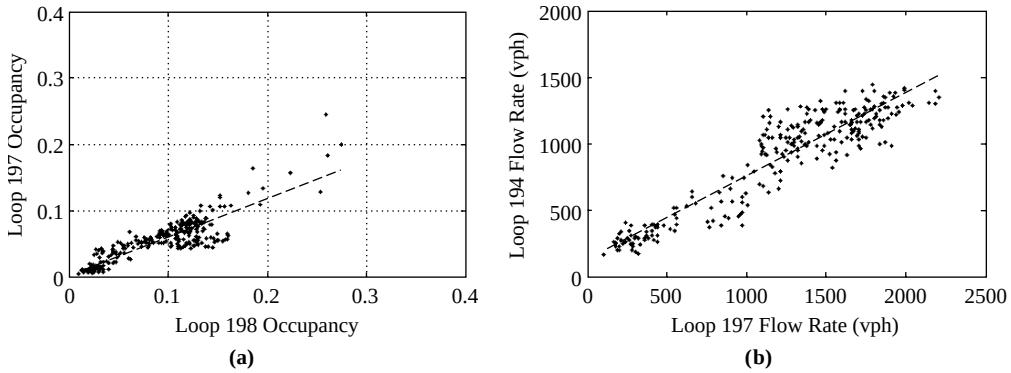

FIGURE 5 Scatter plot of occupancies and flows from two pairs of neighbors.

For each pair of neighbors (i, j), the parameters $\alpha_0(i, j)$, $\alpha_1(i, j)$, $\beta_0(i, j)$, $\beta_1(i, j)$ are estimated by using 5 days of historical data. Let $q_i(t)$, $q_j(t)$, $t = 1, 2, \ldots, n$ be the historical measurements of volume; then

$$\alpha_0(i, j), \alpha_1(i, j) = \arg\max_{\alpha_0' \alpha_1'} \left\{ \frac{1}{n} \sum_{t=1}^{n} \left[q_i(t) - \alpha_0' - \alpha_1' q_j(t) \right]^2 \right\} \tag{6}$$

The parameters for density are fitted the same way. Parameters can be found for all pairs of loops that report data in the historical database, but some loops never report any data. For them, a set of global parameters $\alpha_0^*(\delta, l_1, l_2)$, $\alpha_1^*(\delta, l_1, l_2)$, $\beta_0^*(\delta, l_1, l_2)$, $\beta_1^*(\delta, l_1, l_2)$ is used that generalize the relationship between pairs of loop in different configurations. For each combination of (relative location, lane of Loop 1, lane of Loop 2), the linear model is as follows:

$$\begin{aligned} q_i(t) &= \alpha_0^*(\delta, l_i, l_j) + \alpha_0^*(\delta, l_i, l_j) q_j(t) + \text{noise} \\ k_i(t) &= \beta_0^*(\delta, l_i, l_j) + \beta_1^*(\delta, l_i, l_j) k_j(t) + \text{noise} \end{aligned} \tag{7}$$

where

$\delta = 0$ if i and j are in the same location on the freeway, 1 otherwise;
l_i = lane number of loop i;
l_j = lane number of loop j; and
$l_i, l_j = 1, 2, 3, \ldots, 8$.

The global parameters are fitted to data similar to the local parameters. In Los Angeles, there are 60,760 pairs of neighbors (i, j) for 5,377 loops; in San Bernardino, there are 3,896 pairs for 466 loops. The four parameters for each pair are computed for these two districts and stored in database tables.

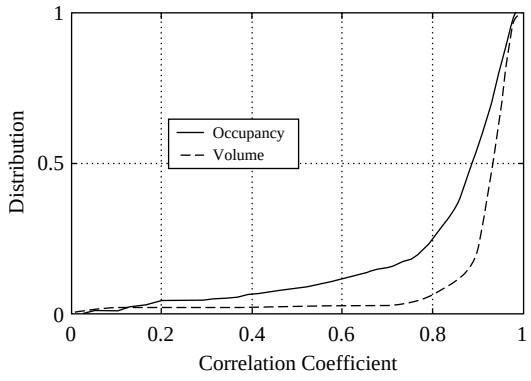

FIGURE 6 Cumulative distribution of correlation coefficients between neighbors.

When values are imputed for loop i by using its neighbors, each neighbor provides an estimate, and the final estimate is taken as the median of the pairwise estimates. Both volume and occupancy imputation are performed the same way. The imputation for volume is

$$\begin{aligned} \hat{q}_{ij}(d, t) &= \alpha_0(i, j) + \alpha_1(i, j) q_j(d, t) \\ \hat{q}_i(d, t) &= \text{median} \left\{ \hat{q}_{ij}(d, t),\ j : \text{neighbor of } i, \hat{\Delta}_j(d, t) = 0 \right\} \end{aligned} \tag{8}$$

Here, $\hat{\Delta}_j(d, t)$, obtained from Equation 2, is the diagnosis of the jth loop—only estimates from good neighbors are used in the imputation. Equation 12 is a way to combine information from multiple neighbors. While this method is suboptimal compared with those with joint probability models, such as multiple regression, it is more robust. Multiple regression models all neighbors jointly, unlike the pairwise model adopted here. Dailey presented an estimation method based on all neighbors jointly (*6*), but here, the pairwise model was chosen for its robustness—it generates an estimate as long as there is one good neighbor. In contrast, multiple regression needs values at each sample time from all the neighbors. Robustness is also increased by use of the median of $\hat{q}_{ij}$ instead of the mean, which is affected by outliers and errors in Δ_j.

After one iteration, the imputation algorithm generates estimates for all the bad loops that have at least one good neighbor. Something remains to be done for the bad loops that do not have good neighbors. A scheme for this has not been chosen, but there are several alternatives. The current implementation simply iterates the imputation process. After the first iteration, a subset of the bad loops is filled with imputed values—these are the loops with good neighbors. In the second iteration, the set of good loops grows to include those that were imputed in the previous iteration, so some of the remaining bad loops now have good neighbors. This process continues until all loops are filled or until all the remaining bad loops do not have any good neighbors. The problem with this method is that the imputation becomes less accurate with each succeeding iteration. Fortunately, most of the bad loops are filled in the first iteration. In District 7 on April 24, 2002, for example, the percentages of filled loops in the first four iterations are 90%, 5%, 1%, and 1%; the entire grid is filled after eight iterations. Another alternative is to use the current imputation only for the first n imputations. After that, if there are still loops without values, another method can be used, such as historical mean. In any case, an alternative imputation scheme is required for sample times when there are no good data for any loop.

TABLE 4 Performance of Imputation

Quantity	Mean	Standard Deviation	Mean Absolute Error	Standard Deviation of Error	Mean Error
Occupancy	0.085	0.061	0.013	0.021	0.001
Volume (vph)	1220	527	132	201	6

Performance

The performance of this algorithm was evaluated for data from April 24, 2002. To run this test, 189 loops were found that are themselves good and also had good neighbors. From each loop i, the measured flows and occupancies $q_i(t)$ and $k_i(t)$ were collected; the algorithm was then run to compute the estimated values $\hat{q}_i(t)$ and $\hat{k}_i(t)$, based on neighbors. From these, the root-mean-square errors for each loop were found; see Table 4. This table shows that the estimates are unbiased, as they should be. The standard deviation of imputation error is small compared to the mean and standard deviation of the measurements. Figure 7 compares the estimated and original values for one loop. They show good agreement.

The performance of the algorithm was also compared against that of linear interpolation. Fifteen triplets of good loops were chosen for this test. Ten of the triplets are loops in the same lane, different locations, while five triplets have their loops in the same location, across three lanes. In each triplet, two loops were used to predict the volume and occupancy of the third loop by using linear interpolation. In every case, the neighborhood method produced a lower error in occupancy estimates; it produced smaller errors in flow estimates in 10 of 15 locations. Overall, the neighborhood method performed better in the mean and median, as expected.

CONCLUSIONS

Algorithms were presented to detect bad loop detectors from their outputs and to impute missing data from neighboring good loops. Existing methods of detection evaluate each 20-s sample to determine if it represents a plausible traffic state, but it was found that there is much more information about how detectors behave over time. The presented algorithm makes diagnoses based on the sequence of measurements from each detector over a whole day. Visually, bad data are much easier to detect when viewed as a time series. The algorithm found almost all the bad detectors that could be found visually.

The imputation algorithm estimates the true values at locations with bad or missing data. This is an important functionality, because almost any algorithm that uses the data needs a complete grid of data. Traditionally, the way to deal with missing data is to interpolate from nearby loops. The presented algorithm performs better than interpolation because it uses historical information on how the measurements from neighbor detectors are related. The volume and occupancy between neighbor loops were modeled linearly, and the linear regression coefficients of each neighbor pair were found from historical data. This algorithm is simple and robust.

There remain many possibilities for improvements to the algorithms described here. The detection algorithm has a time lag. To address this, a truly real-time detection algorithm is being developed that will incorporate neighbor loop measurements as well as the past day's statistics. While the linear model describes most neighbor pairs, some pairs have nonlinear relationships, so a more general model may be better. Another area for improvement is the handling of entire blocks of missing data. The current imputation algorithm needs a large number of good loops to impute the rest, but it does not work if most or all the loops are bad for a sample time. A method is needed for addressing this situation.

Single-loop data diagnostics is an important area of research. While loop detectors are the most abundant source of traffic information, the data are sometimes bad or missing. The algorithms presented construct a complete grid of clean data in real time. They simplify the design of upper-level algorithms and improve the accuracy of analysis based on loop data.

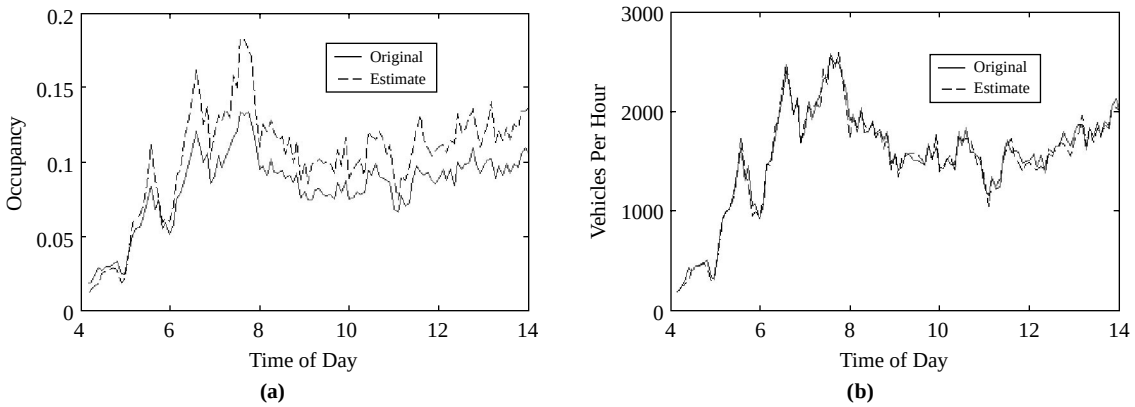

FIGURE 7 Original and estimated (*a*) occupancies and (*b*) flows for a good loop.

ACKNOWLEDGMENTS

This study is part of the PeMS project, which is supported by grants from Caltrans to the California PATH Program. The authors thank the engineers from Caltrans Districts 3, 4, 7, 8, 11, and 12 and from Caltrans headquarters for their encouragement, understanding, and patience.

REFERENCES

1. Chen, C., K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia. Freeway Performance Measurement System: Mining Loop Detector Data. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1748*, TRB, National Research Council, Washington, D.C., 2001, pp. 96–102.
2. Payne, H. J., E. D. Helfenbein, and H. C. Knobel. *Development and Testing of Incident Detection Algorithms.* FHWA-RD-76-20. FHWA, U.S. Department of Transportation, 1976.
3. Jacobson, L. N., N. L. Nihan, and J. D. Bender. Detecting Erroneous Loop Detector Data in a Freeway Traffic Management System. In *Transportation Research Record 1287*, TRB, National Research Council, Washington, D.C., 1990, pp. 151–166.
4. Cleghorn, D., F. L. Hall, and D. Garbuio. Improved Data Screening Techniques for Freeway Traffic Management Systems. In *Transportation Research Record 1320*, TRB, National Research Council, Washington, D.C., 1991, pp. 17–23.
5. Nihan, N. Aid to Determining Freeway Metering Rates and Detecting Loop Errors. *Journal of Transportation Engineering*, Vol. 123, No. 6, 1997, pp. 454–458.
6. Dailey, D. J. *Improved Error Detection for Inductive Loop Sensors.* WA-RD 3001. Washington State Department of Transportation, Olympia, 1993.

Publication of this paper sponsored by Committee on Highway Traffic Monitoring.