Secondary Crash Identification on
A Large-scale Highway System

Dongxi Zheng (corresponding author)
Research Assistant
TOPS Laboratory, Department of Civil and Environment Engineering
University of Wisconsin-Madison
1415 Engineering Drive, Room 1249A, Madison, WI, 53706
Phone number: (608)335-0889, E-mail: dzheng3@wisc.edu

Madhav V. Chitturi, PhD
Assistant Researcher
TOPS Laboratory, Department of Civil and Environment Engineering
University of Wisconsin-Madison
B243 Engineering Hall, 1415 Engineering Drive, Madison, WI 53706
Phone number: (608) 890-2439, Fax: (608)262-5199, E-mail: mchitturi@wisc.edu

Andrea R. Bill
Associate Researcher
TOPS Laboratory, Department of Civil and Environment Engineering
University of Wisconsin-Madison
B243 Engineering Hall, 1415 Engineering Drive, Madison, WI 53706
Phone number: (608) 890-3524, Fax: (608)262-5199, E-mail: bill@wisc.edu

David A. Noyce, PhD, PE
Associate Professor
Director, Traffic Operations and Safety (TOPS) Laboratory
Department of Civil and Environment Engineering
1204 Engineering Hall, 1415 Engineering Drive
Madison, WI 53706
Phone number: (608) 265-1882, Fax: (608)262-5199, E-mail: noyce@engr.wisc.edu

Submission date: 11/14/2013
Word count: Text (6371) + 250*Figure (4) + 250*Table (0) = 7371
ABSTRACT

The annual cost of congestion in the United States reportedly exceeds $120 billion (1). Freeway incidents are major sources of non-recurrent congestion and the resulting secondary crashes can prolong the traffic impact and increase the cost. Research on secondary crashes to support statewide transportation system management has been limited. In the current study, a two-phase automated procedure is developed to identify secondary crashes on large scale regional transportation systems. In the first phase, a crash pairing algorithm is developed to extract spatially and temporally near-by crash pairs. The accuracy and efficiency of the algorithm were validated by comparing to an ArcGIS based program. In the second phase, two filters are proposed to reduce the crash pairs for secondary crash identification: the first filter selects crash pairs whose earlier crashes were on mainline highways; the second filter selects crash pairs whose later crashes happened within the dynamic impact areas (i.e., backup queues) of the earlier crashes. Shockwave theory is used to model the dynamic impact of a primary incident. The two-phase procedure uses a linear referencing system for crash localization and can be applied to any regional transportation system with a similar data structure. A case study was conducted on nearly 1,500 miles of freeways in Wisconsin using 2010 data. Among the crash pairs produced by the two-phase procedure, 79 secondary crashes were confirmed via police reports. Preliminary analyses showed that 1) secondary crashes occurred in the same traffic direction as the primary incidents were about three times greater in frequency compared to secondary crashes in the opposing direction, and 2) two-vehicle rear-ends, multiple-vehicle rear-ends, and sideswipes were three major types of secondary crashes (over 85%).
INTRODUCTION

A secondary crash is an undesirable consequence resulting from a primary incident. More formally, according to the Federal Highway Administration (FHWA), “secondary crashes are those that occur with the time of detection of the primary incident where a collision occurs either a) within the incident scene or b) within the queue, including the opposite direction, resulting from the original incident” (2). Existing studies have shown the extended traffic impact and the economic costs of secondary crashes (3–5). Reducing the chances of secondary crashes becomes a major consideration in the dispatch plans of traffic incident management (TIM) agencies (6, 7).

In spite of various findings on secondary crashes, most existing studies were limited by scope. Many studies were conducted on only one or two sample freeways or a short segment of highway; other studies extended the scope to freeways but considered a small regional scale. Only two studies were performed at a large scale that involved statewide highway systems. One of the major reasons for such scope constraints was the challenge of secondary crash identification. In order to identify secondary crashes accurately, most existing studies considered the dynamic features of the traffic impact caused by the primary incidents. Thus, the study scopes were limited to highway facilities where high resolution traffic data were available for dynamic analyses. In addition, modeling the dynamic impact of primary incidents required considerable computational efforts, which for a statewide transportation system could be intolerable or even infeasible. Previous studies considering statewide highway systems did not consider the dynamic impact of primary incidents. In summary, none of the previous studies investigated secondary crashes on a statewide transportation system while considering the dynamic impact of primary incidents.

To fill the above research gap, the current study develops a two-phase automatic procedure. In the first phase, spatially and temporally near-by crash pairs (up to custom static thresholds) are extracted from a large network based on a crash pairing algorithm. The accuracy and the efficiency of this algorithm were validated. In the second phase, two filters are used to select crash pairs that are more likely to be primary-secondary crash pairs. One of the filters utilizes shockwave theory to evaluate the dynamic traffic impact of the primary incidents. At the end of the two-phase procedure, manual review of identified police reports is needed to confirm actual secondary crashes. However, the number of crash reports to review is considerably less.

LITERATURE REVIEW

Secondary crashes have been observed to be one of the notable consequences of freeway incidents. Early in 1970s, Owens conducted an on-the-spot study of traffic incidents on a 21 kilometer (13 mile) stretch of motorway in England during peak hours and found that 32.5% of the observed crashes were related to primary incidents (8). In recent decades, the development of intelligent transportation systems (ITS) has made a variety of transportation data easier to access, which in turn has encouraged researchers to revisit secondary crashes. In earlier studies (3–5, 9–17), an incident was identified as a secondary crash as long as it occurred within a rectangle time-space window originated from another incident. For example, Raub classified an incident as a secondary crash if it happened within 1,600 meters upstream of another incident and no later than 15 minutes after that incident was cleared (9, 10). This method type was called static thresholds in the sense that it considered the spatial impact range of a primary incident to be consistent throughout a certain time period. However, the impact of a traffic incident is typically dynamic with respect to time. Later studies incorporated this fact in secondary crash
identification (18–29). The earliest attempt was made by Moore et al. who classified an incident as a secondary crash only if it fell under the progression curve (i.e., the resulting queue boundary as a function of time based on real-time queue end tracking data) of another incident (5). Curves with a similar concept were generated using the traffic arrival-departure model in other studies (20–22, 24). In fact, these dynamic curves only depicted the moving queue fronts, but did not consider the queue release from the incident location since the onset of incident clearance. To accommodate the releasing front, researchers have used either the speed contour map method or the ASDA model to depict the impact area of an incident (19, 23, 25–29). However, shockwave theory, which can also model the queuing and the releasing dynamics, has not been used in the literature for secondary crash identification.

Research on secondary crashes at the scale of a statewide transportation system has been limited. A majority of the existing studies focused on one or two sample freeways or only a stretch of a highway of which detailed traffic conditions could be obtained through densely deployed traffic detectors, closed circuit traffic cameras (CCTV), or even aircraft based congestion tracking systems (4, 14, 16, 18, 19, 23, 25–28). Some other studies extended the research scope to several freeways or urban arterials within a fully patrolled and ITS assisted district (3, 5, 9, 10, 12, 13, 20–22, 24, 29). Only a few studies were conducted on statewide transportation systems (11, 17). Identifying all spatially and temporally near-by crash pairs from a large highway network, and hence a significant amount of input crashes, could be computationally complex. None of the above studies provided an efficient procedure.

Based on the above literature review, two primary objectives of the current study were set. First, an efficient algorithm to identify all near-by crash pairs (up to custom static thresholds) for a statewide transportation system should be developed. Second, in order to reduce the candidate primary-secondary crash pairs based on the static thresholds, additional filters that incorporate the dynamic feature of primary incident impact should be established.

DATA DESCRIPTION

The WisTransPortal Data Hub of the Traffic Operations and Safety (TOPS) Laboratory at the University of Wisconsin – Madison houses a variety of statewide transportation data prepared and provided by the Wisconsin Department of Transportation (WisDOT) (30, 31). Among these data, the State Trunk Network (STN) data and the crash data are the two primary inputs to the current study. The STN includes the state trunk highways (STH), the US highways (US), the interstates highways (IH), the designated freeways, and the expressways in Wisconsin as of 2012 (32). The crash data cover all reported crashes in Wisconsin since 1994 and are updated monthly. WisDOT provides both the maps (i.e., Esri shapefiles) and the database records of STN and crashes. WisDOT also embeds a linear referencing system in the crash records to allow locating a crash on the STN without using the maps. For the proposed algorithm, the database records with the linear referencing system were used. The maps were used for validation and comparison purposes.

STN Links and Linear Referencing

A traditional way of modeling highway networks is a figure that consists of nodes and directional links. The STN is stored in this manner. Nodes in STN are called reference sites (RS). Each link in the STN starts from one RS (RSfrom) and ends at another RS (RSto). A link represents a highway segment, either mainline or ramp, in one traffic direction with relatively consistent
geometric layout (e.g., number of lanes, lane width, etc.). Figure 1 illustrates how a small portion of a highway network is represented in the STN. Links are displayed as solid arrows with their lengths. RS’s are labeled in circles. An arbitrary location on the STN, for example a crash scene, is determined by a linear coordinate \([\text{link id}, \text{link offset}]\), namely linear referencing. The link id tells in which link the crash occurred and the link offset tells the distance from the link’s RS from to the crash. As of the year 2012, the total number of in-operation links is 33,015 and the total length is 24,903 miles.

![Illustration of the linear referencing system.](image)

**Example distances:**

\[
d(A,B) = (0.8 - 0.15) + 0.2 + (0.3 - 0.05) = 1.1 \text{ miles}
\]

\[
d(B,C) = 0.3 - 0.05 - 0.1 = 0.15 \text{ miles}
\]

**FIGURE 1 Illustration of the linear referencing system.**

**Crash Records**

WisTransPortal stores each reported crash in Wisconsin as a record in the database. Each crash record contains detailed information about the crash, such as a unique identification number, date, time, link id and link offset of the crash location (for linear referencing). Also, each crash record is associated with a document id that links the crash to its police report form MV4000. The MV4000 form provides additional information such as the police investigation with a crash diagram.

**Other Data**

In addition to the STN links, WisTransPortal also stores other highway information. For example, all the routes in STN are stored in a table, each record representing the entire stretch of a highway route and its geographical direction (e.g., US 12 East Bound); virtual mile markers are stored as reference points. Traffic data are also available. WisDOT manages the TRaffic DAta...
System (TRADAS) and the Advanced Transportation Management System (ATMS), with traffic detectors deployed on the STN. WisTransPortal contains information of these TRADAS and ATMS detectors as well as their traffic counts.

FIRST PHASE: CRASH PAIRING ALGORITHM

Given the STN linear referencing system and the crash records, the target of the crash pairing algorithm is to identify all crash pairs \((c_i, c_j)\) that satisfy Formulas 1 and 2. In Formula 2, \(d(c_i, c_j)\) is measured along the STN links by treating the links as bi-directional (see examples in Figure 1). Highway splits, merges, and intersections should be accommodated, which was not addressed by previous studies focusing on individual freeways.

\[
0 \leq t(c_j) - t(c_i) \leq T
\]
\[
d(c_i, c_j) \leq D
\]

where,
\[c_i = \text{Crash } i, \text{ the former crash;}
\]
\[c_j = \text{Crash } j, \text{ the later crash;}
\]
\[t(c) = \text{The time of crash } c \text{ since an early time origin, min;}
\]
\[d(c_i, c_j) = \text{The network distance between crash } c_i \text{ and } c_j, \text{ mile;}
\]
\[T = \text{The time window (threshold), min;}
\]
\[D = \text{The space window (threshold), mile.}
\]

Given the significant sizes of the STN links and the crashes, simple algorithms are either slow or infeasible. One na"ive algorithm is to run Dijkstra’s method repeatedly for every crash. Dijkstra’s method is an iterative approach that finds the shortest path from an origin to every node in a network. Dijkstra’s method can be briefly summarized as follows: All nodes are considered to be infinitely distant from the origin and “unvisited” initially. The method begins from the origin and computes the distances to its neighbors (i.e., nodes with direct connection) and marks the origin as “visited”. In every successive iteration step, the method chooses the closest “unvisited” node to the origin, updates the distances from the origin to that node’s “unvisited” neighbors if the paths become shorter through that node, and marks that node as “visited”. The iteration continues until every node is “visited”. At the end, the distances from the origin to each node are the shortest distances (33). The complexity of Dijkstra’s method is \(O(N^2)\) in respect to \(N\) crashes, where \(N\) is larger than 100,000 for an annual statewide analysis. By repeatedly using Dijkstra’s method for \(N\) crashes, the complexity of the na"ive algorithm becomes \(O(N^3)\), which is not efficient. Another alternative is to use dynamic programming to populate a shortest path matrix between every two crashes. This alternative is infeasible because it not only spends an equivalent amount of computation time as the first algorithm but also requires unacceptable computer memory space (e.g., \(100,000^2 \times 8\) bytes \(\approx 75\) GB) to store the matrix.

The proposed pairing algorithm first analyzes the relationships between links and uses these relationships to derive crash-to-crash distances. For each link, \(l_{ki}\), that contains one or more crashes, the algorithm performs a variant of Dijkstra’s traversal (as will be explained later) and generates the relationships between \(l_{ki}\) and the other links. The distances between crashes are
then calculated based on these relationships. Compared to the first algorithm mentioned above, the number of traversals is bounded by the total number of links no matter how many crashes are analyzed. The pairing algorithm also utilizes the $D$ mile space window to constrain the Dijkstra’s traversal to a relevant portion (normally small) of the STN network. In the following subsections, the concept of a local linear coordinate system is introduced, based on which the relationship between two links can be comprehensively defined. Additionally, the equation to derive crash-to-crash distance from the link-to-link relationship is also given, along with the concept of a candidate link that is used to constrain the Dijkstra’s traversals, the pseudo code of the algorithm with special case explanation, and finally, the validation of this algorithm.

**Local Linear Coordinate System**

A local linear coordinate system (LLCS) is defined for each link, namely a base link, to describe the spatial relationship between any RS and the base link. Let $RS_{\text{base}}$ and $RS_{\text{to}}$ denote the from-reference-site and the to-reference-site of the base link. Under the LLCS, each RS in the network has a two-fold coordinate with the following definitions:

- **Forward (positive) coordinate** ($x^+_{RS}$) = the length of the base link + $d(RS_{\text{to}})$, $d(RS_{\text{from}})$, $d(RS_{\text{to}})$, $d(RS_{\text{from}})$ is the shortest network distance between $RS_{\text{base}}$ and$RS$ in a sub-network without $RS_{\text{from}}$ (and links connected it). If $d(RS_{\text{to}})$, $d(RS_{\text{from}})$ does not exist, $x^+_{RS} = +\infty$. Specially, $x^+_{RS_{\text{base}}} = 0$.

- **Backward (negative) coordinate** ($x^-_{RS}$) = $d(RS_{\text{from}})$, $d(RS_{\text{to}})$, $d(RS_{\text{from}})$, $d(RS_{\text{to}})$ is the shortest network distance between $RS_{\text{base}}$ and $RS$ in a sub-network without $RS_{\text{to}}$ (and links connected to it). If $d(RS_{\text{from}})$, $d(RS_{\text{to}})$ does not exist, $x^-_{RS} = +\infty$. For example, $x^-_{RS_{\text{to}}} = +\infty$.

As an example, in Figure 1, consider $RS_{12}$ under the LLCS of $link_{1,3}$ (as the base link). $x^+_{RS_{12}} = 0.8$ ($link_{3,6}$) + 0.2 ($link_{4,14}$) + 0.3 ($link_{14,13}$) + 0.55 ($link_{13,12}$) = 1.85 miles. $x^-_{RS_{12}} = 0.2$ ($link_{9,3}$) + 0.4 ($link_{10,9}$) + 0.4 ($link_{12,3}$) = 1.0 mile.

A variant of the Dijkstra’s shortest path traversal can be used to calculate the LLCS coordinates of all RS’s on the fly. The traversal is divided into two passes. In the first pass, the Dijkstra’s algorithm starts from $RS_{\text{to}}$ and expands to the rest of the network while ignoring all links connected to $RS_{\text{from}}$. During the traversal, the forward coordinates of all reached RS’s are calculated or updated. Similarly, in the second pass, the Dijkstra’s algorithm starts from $RS_{\text{from}}$ and ignores all links connecting to $RS_{\text{to}}$, filling the backward coordinates of all reached RS’s.

In the context of a LLCS, any link (including the base link) is related to the base link by the LLCS coordinates of its $RS_{\text{from}}$ and $RS_{\text{to}}$. Specifically, let a link to be related to the base link be called a test link and its end RS’s be denoted as $RS_{\text{test}}$ and $RS_{\text{test}}$. Vector $v_{\text{test}} = \begin{bmatrix} x^+_{RS_{\text{from}}}, x^-_{RS_{\text{from}}}, x^+_{RS_{\text{to}}}, x^-_{RS_{\text{to}}} \end{bmatrix}$ is defined as the relationship vector of the test link in the LLCS. With the relationship vector, the network distance between a crash $c_{\text{base}}$ on the base link and a crash $c_{\text{test}}$ on a test link can be easily calculated using Equations 3-7. Since the four coordinates in the relationship vector might result from different routings, there could be four possible crash-
to-crash distances (Equations 4-7) whose geometric meanings are demonstrated in Figure 2. The final crash-to-crash distance should be the smallest possible distance. Besides the distance value, one can also tell if the two crashes were in the same traffic direction. For example, if the final distance is $d^+_F$ (upper right case in Figure 2), the centerline of the resulting route is bolded and the traffic directions of both crashes (green arrows) are on the same side of the centerline, meaning the two crashes (or links) were in the same traffic direction; otherwise, like $d^-_F$ and $d^+_T$, the two crashes were in the opposite traffic directions. Additionally, one can also determine whether $c_{test}$ happened upstream or downstream of $c_{base}$. For instance, $c_{test}$ happened upstream of $c_{base}$ if $d^-_T$ or $d^-_T$ is the final distance (when the test crash direction follows the bolded route); otherwise, $c_{test}$ happened downstream of $c_{base}$ (when the test crash direction departs the bolded route).

$d(c_{base}, c_{test}) = \min(d^+_F, d^+_T, d^-_F, d^-_T)$ \hspace{1cm} (3)

$d^+_F = x^+_{RS^f_{test}} - o_{base} + o_{test}$ \hspace{1cm} (4)

$d^+_T = x^+_{RS^t_{test}} - o_{base} + (l_{test} - o_{test})$ \hspace{1cm} (5)

$d^-_F = x^-_{RS^f_{test}} + o_{base} + o_{test}$ \hspace{1cm} (6)

$d^-_T = x^-_{RS^t_{test}} + o_{base} + (l_{test} - o_{test})$ \hspace{1cm} (7)

where,

$d(c_{base}, c_{test})$ = The network distance between $c_{base}$ and $c_{test}$, mile;

$d^+_F$ = A possible distance via $RS^f_{test}$ forward coordinate, mile;

$d^+_T$ = A possible distance via $RS^t_{test}$ forward coordinate, mile;

$d^-_F$ = A possible distance via $RS^f_{test}$ backward coordinate, mile;

$d^-_T$ = A possible distance via $RS^t_{test}$ backward coordinate, mile;

$x^+_{RS^f_{test}}$ = Forward coordinate of $RS^f_{test}$, mile;

$x^-_{RS^f_{test}}$ = Backward coordinate of $RS^f_{test}$, mile;

$x^+_{RS^t_{test}}$ = Forward coordinate of $RS^t_{test}$, mile;

$x^-_{RS^t_{test}}$ = Backward coordinate of $RS^t_{test}$, mile;

$o_{base}$ = Link offset of $c_{base}$, mile;

$o_{test}$ = Link offset of $c_{test}$, mile;

$l_{test}$ = Length of the test link, mile.
In the previous section, every link is assumed to be tested against the base link. However, given a particular spatial threshold of $D$ miles, a test link too far away from the base link is irrelevant to finding the near-by crash pairs. Only those links whose relationship vectors satisfy a certain condition may contain crashes within $D$ miles of the base link crashes. In fact, the condition is as simple as $\min(x_{RS^+} - l_{base}, x_{RS^-} - l_{base}) \leq D$, where $l_{base}$ is the length of the base link. Links satisfying this condition are called candidate links and form a relatively small and relevant portion of the network (when $D$ is relatively small). The two passes of Dijkstra’s traversal can stop expansion as early as any further RS to be reached has a forward coordinate larger than $l_{base} + D$ and a backward coordinate larger than $D$. Then, all links connected to the already reached RS’s are all the candidate links.

The Algorithm

Below is the pseudo code of the proposed crash pairing algorithm. $L_{base}$ is assumed to be a preprocessed set of links containing at least one crash. The statement “find all candidate links” refers to the preparation of the relationship vectors for all candidate links in the LLCS as described in the above sections. $t(*)$ is the function of getting the time of a crash in minutes since
a consistent time origin. $T$ and $D$ are the static thresholds in minutes and miles, respectively. It should be noted that the recorded time of crash could be slightly different from the time when the crash actually occurred. However, the authors do not expect it to have a significant impact on the results since a large time threshold of 5 hours was used. The statement “calculate $d(c_{base}, c_{cand})$” refers to Equations 3-7.

For each $l_{k_{base}}$ in $L_{base}$:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Find all candidate links of $l_{k_{base}}$ as a set $L_{cand}$;</td>
</tr>
<tr>
<td>2</td>
<td>For each candidate link $l_{k_{cand}}$ in $L_{cand}$:</td>
</tr>
<tr>
<td>3</td>
<td>For each crash $c_{base}$ in $l_{k_{base}}$:</td>
</tr>
<tr>
<td>4</td>
<td>For each crash $c_{cand}$ in $l_{k_{cand}}$:</td>
</tr>
<tr>
<td>5</td>
<td>If $0 \leq t(c_{cand}) - t(c_{base}) \leq T$:</td>
</tr>
<tr>
<td>6</td>
<td>Calculate $d(c_{base}, c_{cand})$;</td>
</tr>
<tr>
<td>7</td>
<td>If $d(c_{base}, c_{cand}) \leq D$:</td>
</tr>
<tr>
<td>8</td>
<td>Add $(c_{base}, c_{cand})$ as a pair in the result;</td>
</tr>
</tbody>
</table>

A special case should be treated differently. As illustrated in Figure 1, the longitudinal distance between crash B and crash C was only 0.15 miles while they occurred on opposite sides of the same highway. However, relying only on the network traversal of links, the resulting distance will go around RS$_{14}$ and be calculated as 0.25 miles. Such unrealistic result is not desirable. In order to overcome this limitation, additional information from the STN were employed. An STN table of route-links is used to aid the links with their physical meanings. Each record in the route-link table tells which highway a link belongs to in what direction. All links on the other side of the same highway are considered candidate links of the base link. When calculating the distance between a crash on the base link and a crash on the other side of the highway, the algorithm calculates the cumulative distances from the two crashes to a far upstream/downstream shared RS on the highway. The difference between these two cumulative distances is considered the distance between these two crashes. Additionally, when a shared RS could not be found, the algorithm further utilize another set of highway reference locations, reference points (RP). Each RP has its on-highway number, on-highway direction, RP number, and RP letter. If two RP’s have the same on-highway number, RP number, and RP letter, they correspond to the same longitudinal position on the highway, even with different on-highway directions. Additionally, each RP, like a crash location, has a linear reference that maps it on to a link. Based on the above input, if two links on opposite sides of the same highway contains RP’s with the same RP number and RP letter, there is a shared longitudinal position between them. Thus, instead of looking for a shared RS, the algorithm looks for a shared longitudinal position based on RP’s.

**Validation**

The pairing algorithm was implemented as a Java program. The program passed several small independent tests (e.g., the entire stretch of a particular highway in STN with crashes of several days) with manual extracted ground truths. In order to further validate the accuracy and the
efficiency of the algorithm, a large scale network was tested. Since the ground truth in the large scale test was infeasible to be manually extracted, a relatively reliable ArcGIS based program was used as a mutual validation reference. The basic idea of the ArcGIS based program is to prepare a network dataset using the STN shapefile and the crash shapefile and use the buffer function of the NetworkAnalyst toolbox to find, for each crash, every other crash that is within a buffer network distance (the spatial threshold) from that crash. The ArcGIS based program was implemented in C++ using the ArcGIS APIs. Due to the unavailability of control over the buffer function of the NetworkAnalyst toolbox, the ArcGIS based program was similar to the naïve algorithm of traversing the network for every pair of crashes, which provided the authors a chance to compare the efficiencies.

Both the pairing algorithm and the ArcGIS based program were tested on 10,922 crashes from a freeway network of about 1,500 total miles in Wisconsin in 2010, with \( D = 10 \) miles and \( T = 5 \) hours. The pairing algorithm yielded 15,901 crash pairs while the ArcGIS based program yielded 13,850 crash pairs. Both systems captured the same 13,594 crash pairs. The ArcGIS based program captured 256 extra pairs, which were later found to be missed by the pairing algorithm due to computer precision problems and did not hurt the validity of the pairing algorithm. The pairing algorithm captured 2,307 extra pairs which were correct output but missed by the ArcGIS based program. In summary, the pairing program correctly identified more crash pairs than the ArcGIS based program. In addition, the ArcGIS based program finished the analysis in two and a half days while the pairing algorithm finished in about 2 hours (30 times faster).

SECOND PHASE: CRASH PAIR FILTERS

For the purpose of identifying secondary crashes, the pairing algorithm produces an initial searching set, which without additional filtering could be too vast to be useful. Two filters are proposed below as to select out crash pairs that are more likely to capture primary-secondary relationship.

**Ramp Filter**

A crash pair resulting from the proposed algorithm will be excluded if its former crash happened on a highway on-ramp or a highway off-ramp. A crash is determined to be on a ramp if the link on which it happened represents a portion or entire segment of a ramp. The rationale behind this filter is that ramp crashes rarely caused secondary crashes. To evaluate this assumption, 85 sample crash pairs whose former crashes happened on ramp were selected and verified. One or two crash pairs were randomly sampled from each 1-hour-5-mile intervals of the 5-hour-10-mile thresholds. Manual review showed that none of the 85 samples contained a primary-secondary pair. Although one crash pair involved two secondary crashes, but they were not related to each other; in addition, these two secondary crashes were captured by the actual primary-secondary pairs whose primary crashes were not on ramps.

**Impact Area Filter**

As mentioned in the literature review, crash pairs resulted from static thresholds could contain false primary-secondary pairs. These false pairs generally have unreasonable combinations of time distance and spatial distance. For example, a candidate pair whose time distance is 0 minutes but the spatial distance is 5 miles is certainly not a primary-secondary crash pair. Since secondary crashes have been recognized to be in the queue caused by the primary
incidents, queue theories were commonly used to establish the time-varying impact area of the primary incidents to identify secondary crashes \((13, 14, 18–29)\). Comparison of various queue estimation methods can be found in more general traffic research papers \((34, 35)\). Based on the literature review, none of the previous secondary crash studies used the shockwave model to estimate the impact area (IA) caused by a primary incident. In the current study, the IA of a crash is defined between two simplified straight shockwave lines, one for the queuing shockwave and the other for the discharging shockwave (Figure 3). Mathematical representation to judge if a crash fell into the IA of another is given in Equations 8 through 10. Traffic flow of the prevailing traffic condition \((q_1)\) is the monthly average hourly traffic volume provided by the TRADAS detectors, with the same day of week and the same hour of day as the former crash. If the later crash happened outside the IA and its parallel portion on the opposite traffic direction of the former crash, the crash pair will be excluded. On the other hand, secondary crashes could happen in the vicinity of the primary incident during its clearance. This type of secondary crashes was typically attributed to the “rubbernecking” effect \((8, 36)\). In order to capture these secondary crashes, a crash pair whose spatial distance (upstream or downstream in either traffic direction) was no larger than 1 mile and whose temporal distance was no large than 1 hour should be reserved even if it does not satisfy the IA requirement.

\[
\begin{align*}
    a_2 \times (t - t_{\text{clearance}}) \leq d & \leq a_1 \times t \quad (8) \\
    a_1 &= (q_2 - q_1)/(k_2 - k_1) \quad (9) \\
    a_2 &= (q_3 - q_2)/(k_3 - k_2) \quad (10)
\end{align*}
\]

where,

\[
\begin{align*}
    t &= \text{The time between the former crash and the later crash, hour;} \\
    t_{\text{clearance}} &= 1 \text{ hour (the simplified crash clearance time);} \\
    d &= \text{The network distance between the two crashes, mile;} \\
    a_1 &= \text{The queuing shockwave speed, mile/hour;} \\
    a_2 &= \text{The release shockwave speed, mile/hour;} \\
    q_1 &= \text{The traffic flow of the prevailing condition, veh/hr/ln;} \\
    k_1 &= \text{The density of the prevailing condition, veh/mile/ln. As a simplification, 65 mile/hr is assumed as the prevailing speed, and } k_1 = q_1/65; \\
    q_2 &= 0 \text{ veh/hr/ln (the traffic flow of the jam condition);} \\
    k_2 &= 352 \text{ veh/mile/ln (the density of the jam condition, assuming 15 feet minimum head to head distance between vehicles);} \\
    q_3 &= 1900 \text{ veh/hr/ln (the traffic flow of the saturated condition);} \\
    k_3 &= 1900/65 \text{veh/mile/ln (the density of the saturated condition).}
\end{align*}
\]
CASE STUDY

A case study was performed on crashes happened on approximately 1,500 miles of freeways in Wisconsin for the year 2010. The layout of these freeways in relation to the entire STN network was illustrated in the upper-right area of Figure 4. Although the case study only used crashes happening on these freeways, the calculation of network distances was not subjected to these freeways, but instead, relied on the entire STN network. A total of 12,513 raw input crashes were retrieved for year 2010, the last five hours of 2009, and the first five hours of 2011. The inclusion of five hours into the previous year and the next year corresponds to the selected 5 hour temporal threshold so crash pairs crossing the new year’s boundary could be captured. The workflow and the resulting reduced data in each step are summarized in the left area of Figure 4.

Before applying the pairing algorithm, the raw input crashes were first reduced based on a focused study scope that excluded inclement weather conditions and deer crashes. In Wisconsin, a large portion of crashes were related to inclement weather during the winter. For example, in January 2010, 1,520 of 3,592 crashes (about 42%) on Wisconsin state trunk highways occurred during or after snow or rain. Some circumstances such as successive run-off-road crashes in snow storms and back-to-back rear-end crashes due to slippery or icy roads were recognized to contribute to secondary crashes. However, weather is out of the control of TIM agencies. Since the current research is focused on secondary crashes that are more likely to be prevented by effective TIM, inclement weather related crashes were not included in this study, but the authors intend to study them separately in the future. For a similar reason, deer crashes (about 20% of all crashes) were excluded from the study. As a result, 7,034 crashes remained as the input to the proposed algorithm.

Conservative thresholds of 10-mile-5-hour were used for the first phase (crash pairing).
The 10-mile-5-hour thresholds are approximately 5 times larger (in each dimension) than most static thresholds used in the literature and supersede all actual temporal-spatial ranges of primary-second pairs found by existing studies (3–5, 9–14, 16–29). Thus, using even larger thresholds is unlikely to include more actual primary-secondary crash pairs. The pairing algorithm generated 8,665 crash pairs (4,231 distinct crashes). The second phase (crash pair filtering) further reduced the number of crash pairs down to 1,012 (88.3% reduction). Up to this point in the analysis, all computations were completed automatically within 2 hours. The resulting 1,012 crash pairs for manual review only contained 1,347 distinct crashes. Compared to the initial input of 7,034 crashes, the review effort was saved by 81%.

Secondary crashes and their corresponding primary incidents were confirmed through manual review of police reports. An estimate of 30 man-hours was used in reviewing the 1,012 candidate crash pairs, averaged to nearly 2 man-minutes per crash pair. Potential employment of ORC (optical character recognition) and artificial intelligence may help to further minimize the reviewing time in the future. A crash was classified as a secondary crash only when its report explicitly referred to a previous crash. This criterion might have resulted in fewer than the actual secondary crashes, but ensured the confidence of further analyses on the resulting secondary crashes. Primary incidents were identified only if they could be matched, by either document number or other key descriptions, to those referred by the secondary crashes. According to these criteria, a total of 79 crash pairs were found to contain secondary crashes. The number of distinct secondary crashes was also 79. Among the 79 pairs, 67 captured the primary incidents.

Preliminary analyses were conducted on the resulting primary-secondary pairs and secondary crashes. Among the 67 primary-secondary pairs, 52 secondary crashes (77.6%) happened in the same traffic direction as the primary crashes and the average spatial and temporal distances were 1,511 feet and 15.7 minutes, respectively; 15 secondary crashes (22.4%) happened in the opposite traffic direction of the primary crashes and the average spatial and temporal distances were 1,264 feet and 18.2 minutes, respectively. Among the 79 secondary crashes, 44 were two-vehicle rear-ends (55.7%), 12 were multiple-vehicle rear-ends (15.2%), 13 were sideswipes (16.5%), 5 were hitting debris (7.3%), 2 were angles (2.5%), 2 involved squad vehicles on primary crash scenes (2.5%), and 1 was losing control (1.3%).
FIGURE 4 Summary of the case study of year 2010.

**Raw input**: All crashes happened on the above 18 freeways (mainline and ramps) in the year of 2010, the last 5 hours of 2009, and the first 5 hours of 2011.

**in IA**: Both the upstream in the former incident's traffic direction and the parallel portion of highway in the opposite traffic direction.

Note: The number of distinct crashes in the parentheses is always smaller than twice of the corresponding number of crash pairs. This is because one crash might be captured in more than one crash pairs. The total number of crash pairs before and after a branching point remains the same. However, since a crash might be involved in two crash pairs belonging to different branches, the sum of the numbers of distinct crashes is always larger than the number of crashes before the branching point.
CONCLUSION, RECOMMENDATION, AND FUTURE WORKS

Secondary crashes are known to prolong the non-recurrent congestion caused by the primary freeway incidents. The benefit of reducing secondary crashes has also been found to exceed the TIM countermeasures such as freeway patrol services. However, research of secondary crashes on large regional transportation systems was limited. The current study contributes to the research community with the following efforts and findings:

- An efficient crash pairing algorithm was developed to extract spatially and temporally near-by (up to custom static thresholds) crash pairs from a large scale regional transportation system. The accuracy and efficiency of this algorithm were validated.
- Two effective filters were proposed to select crash pairs that were more likely to capture primary-secondary relationships. The first filter restricts the primary incidents on mainline highways. The second filter restricts the secondary crashes to be within the dynamic impact areas of the primary incidents. Shockwave theory is first used by the current study to estimate the dynamic impact area of a primary incident.
- A two-phase procedure consisting of the above pairing algorithm and filters automatically narrows down the searching space for secondary crashes in a large regional transportation system. While the procedure is based on the commonly used linear referencing system for crash localization, any transportation system with similar data representation can be analyzed with the procedure. A manual review of the effectively narrowed search space is required.
- A case study for crashes occurring in 2010 on about 1,500 miles of Wisconsin freeways was conducted. From the crash pairs extracted using the two-phase procedure, 79 secondary crashes were confirmed via careful manual review of police reports. Secondary crashes happened in the same traffic direction of the primary incidents were about three times of those occurred in the opposite direction. Two-vehicle rear-ends, multiple-vehicle rear-ends, and sideswipes were three major types of secondary crashes (over 85%). Other crash types, such as hitting debris, angle, losing control, and striking squad vehicles were also observed.

Three major future works are recommended. First, to make the whole workflow of secondary crash identification more automatic, optical character recognition (OCR) and artificial intelligence (AI) might be employed to assist human reviewers in reviewing police reports. Second, more years of data need to be collected to establish a larger sample of secondary crashes for more comprehensive statistical analyses. Last but not least, crashes in inclement weather were not included in this analysis because the objective was to analyze secondary crashes that can be mitigated by TIM strategies. The authors realize that secondary crashes occur in inclement weather and recommend that future studies should examine the impact of weather on secondary crashes.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge support of this study from Paul Keltner of the Bureau of Traffic Operations, Wisconsin Department of Transportation. Special thanks go to the TOPS lab staff, especially Steven Parker, who manages the WisTransPortal and provided technical support.
REFERENCES


