Abstract
The land vehicle-based mobile mapping technology offers an efficient and cost-effective way for collecting georeferenced images along road corridors. These images can be used for many transportation related applications ranging from transportation object inspection to 3D generation. The goal of this research is to automate the process of inspecting transportation objects from collected image sequences. Due to the complexity of this problem, the approach proposed deals with transportation objects with known location information that is derived from an existing location database and with vertical line features, such as street light poles and traffic signs, etc. Numerous tests show that we are able to perform automatic inspection of transportation objects from georeferenced imagery by using the developed approach. Both the technical details as well as the evaluation of the testing results are described in the paper.

Introduction
To maintain an up-dated transportation database, inspectors or surveyors are routinely assigned to go to the field to examine the conditions of road infrastructure, marking out missing objects, and surveying moved objects. Mobile mapping technology, however, has proven to be a fast and cost-effective approach for collecting road related spatial data (Bossier et al., 1991; Schwarz et al., 1993; Novak, 1995; Li, 1997; Tao et al., 1998; Tao, 2000a; Tao and El-Shemy, 2000). The VISAT™ mobile mapping system, originally developed at the University of Calgary in conjunction with the Geofit Inc., Canada, has been in commercial operation since 1996. This system was mainly designed for acquiring high-quality and georeferenced digital image sequences along road corridors (El-Shemy et al., 1998; Li et al., 1994; Schwarz and El-Shemy, 1996; Tao, 1996). Multiple sensors, i.e., Global Positioning System (GPS) receivers, an inertial navigation system (INS) and CCD imaging cameras, have been integrated (see Figure 1). By means of a photogrammetric treatment, 3D coordinates of objects appearing in image sequences can be determined. In the current implementation, eight CCD cameras have been mounted on top of the VISAT™ van. Consequently, the system is able to capture road images with a field of view of 180 degrees. The VISAT™ system offers an absolute accuracy of 0.3 m and a relative accuracy of 0.1 m for object points within a 30-m corridor and at a vehicle speed of 50 to 60 km per hour (El-Shemy, 1996; Tao, 1999; Tao et al., 2001).

With mobile mapping technology, transportation objects can be inspected from captured road images during post-mission processing. Thus, the number of field surveys can be reduced or even eliminated. However, it has been recognized that significant manual work is involved in the process of inspecting transportation objects from a vast amount of collected image sequences. First, the images need to be browsed manually in order to determine whether an object is moving or moved, and then, the location of the object identified needs to be measured manually from a stereo image pair.

The goal of this research is to automate the manual process of inspecting transportation objects using known location information obtained from an existing location database. Because many transportation objects along road corridors have distinctive vertical structures, the approach developed deals with objects with vertical line features, such as street light poles, power-line poles, traffic signs, etc. In this context, the automated inspection process involves (1) verification of the existence of the object and (2) determination of the 3D coordinates of the object location if it exists.

There is other research work focusing on automatic recognition and location of transportation objects from image sequences (Gajdamowica and Ohman, 1998; Geiselmann and Hahn, 1994; Priese et al., 1994). It has been realized that automatic object recognition from imagery is still very restricted. The emphasis of this work is placed on the automatic verification of the object's existence guided by the location information. As a result, autonomous object recognition is eliminated and the reliability of the approach is significantly improved. The test results show that the proposed approach is very promising for operational applications.

Overview of the Approach
The conceptual layout of the approach is shown in Figure 2. It consists of the following three steps: (1) object location prediction, i.e., project the object location in images using the location information obtained from an existing location database; (2) verification of the object's existence, i.e., use feature detection and image matching techniques to verify the object's existence; and (3) determination of the object location, i.e., determine the 3D coordinates of the object using multiple-image matching and update the associated database records. The detailed workflow of the entire process is shown in Figure 3.

Object Location Prediction
The images collected by the VISAT™ mobile mapping system are georeferenced using the GPS/INS integrated positioning method.
Figure 1. The VISAT™ mobile mapping system.

Transportation Objects

Imaging and GPS/INS Georeferencing

Stereo image sequences

Back projection for prediction

Detection and Updating

Transportation Database

Figure 2. Overview of the approach.

Figure 3. The workflow of the automatic inspection process (1 object location prediction; 2 verification of the object existence; 3 determination of the object location).

VISAT™ image capture rate is 0.4 second per image at a vehicle speed of 50 to 60 km per hour. With this capture rate, an object usually appears in three to four consecutive image pairs. We choose the image pair that is closest to the camera for the subsequent processing.

Verification of the Object Existence Using Feature Detection and Matching
If an object is missing or moved, its database record should be updated accordingly. Automatic verification of the existence of the object is a core component in our approach. It involves four steps (see Figure 3): detection of vertical edges, formation of vertical line segments, feature matching of line segments, and object space verification. The above processing is applied to both the left and the right image of a stereo pair in order to improve the reliability of the verification results. If the object cannot be verified by the above process, manual inspection is required to handle the objects missing or moved.

Determination of the Object Location Using Multiple-Image Matching
Once the existence of the object is verified, its 3D coordinates need to be determined in order to update the database records. Multiple-image-based image matching and multiple-point-based photogrammetric intersection are used to derive accurate object coordinates.

Constraints Developed and Employed
Effective use of any available vision constraints is the key to the development of a reliable approach to automatic feature extraction (Tao et al., 2001). The proposed approach was developed based on the following constraints:

- Location Constraint in the Image Domain. Prior information regarding object locations obtained from an existing transportation database is used for predicating the image locations of the
objects. The processing of image windows can then be guided by using this constraint.

- **Georeferencing Constraint.** Image georeferencing information, i.e., the three positional parameters and three attitude parameters, derived from the GPS/INS navigation data is used for direct image-to-space transformation and vice versa.
- **Vertical Line Constraint.** Emphasis is placed on the extraction of objects with vertical line features in images. This constraint reduces the complexity of feature extraction.
- **Constraint from the Stereoscopic and Sequential Imaging Geometry.** The image matching algorithm has been developed by making full use of stereoscopic and sequential imaging geometry.
- **Location Constraint in the Object Space Domain.** The offset between the vehicle trajectory and the location of the extracted object should be neither too far nor too close. This constraint is used to screen out any unqualified features.

**Verification of the Object Existence**

In the VISAT™ system, the cameras have been mounted on top of the van in such a way that the camera baseline is constrained approximately to be parallel to the ground plane and the camera roll angle is set to be very small. As a result, projections of 3D vertical lines are also approximately vertical in images.

The literature regarding extraction of line features from images is very extensive (Basak et al., 1994; Chen and Huang, 1990; Lee and Kweon, 1997; Liu et al., 1990). However, the algorithms vary greatly depending on the types of images and the characteristics of line features to be detected. There is no generic algorithm available that performs well in all environments. Therefore, we have developed a new method that is tailored to our specific needs.

**Detection of Vertical Edges**

An image edge element is comprised of two components, i.e., edge magnitude and edge direction. Because we are more interested in the edges that are vertically oriented, the edge direction information can be a more reliable source. This is mainly due to the fact that the edge magnitude can be likely corrupted by image noise and the other environmental factors, such as shadows. Therefore, edge direction was used as the main evidence for determining edge elements. A three-step algorithm has been developed to detect the edges that are associated with the vertical line features: determination of edge candidates, direction-constrained edge thinning, and determination of vertically oriented edges.

**Determination of Edge Candidates**

Similar to Nevatia and Babu’s strategy (Nevatia and Babu, 1980), the input image is first convolved using the predefined six directional masks (illustrated in Figure 4) coded from 0 to 5. After the convolution, each pixel has six magnitude values and the associated direction codes. A very low threshold of magnitude has been chosen so that all possible edges can be kept at this stage. Pixels whose magnitude values surpass the threshold are accepted as edges. For each edge, the maximum magnitude value and the associated direction code are recorded for the further processing.

**Direction-Constrained Edge Thinning**

The purpose of edge thinning is to retain only those edges whose magnitude is a local maximum. In this algorithm, the obtained direction codes are used to constrain the thinning process. The edge pixel is considered as a thinned edge if the following two conditions are met:

- the edge directions of the two horizontally neighboring pixels are within one unit (45°) of that of the edge pixel (this can be examined using the direction codes), and
- the edge magnitude at the edge pixel is larger than the edge magnitudes of its two neighbors in a direction normal to the direction of the edge pixel.

Each edge pixel is examined by using the above criteria. Edge pixels that do not meet the above conditions are eliminated.

**Formation of Vertical Line Segments**

**Vertical Line Grouping**

Line grouping, being a process of linking isolated edges, has been considered as non-trivial (Basak et al., 1994). Because we are only interested in vertical line features, the vertical line constraint can be enforced; nevertheless, the process is still not straightforward. The primary concern of line grouping is to judge how large a gap between isolated edges should be bridged. A strong linking process can bridge large gaps but may end up with many false line segments, while a weak linking process may not be able to produce desired line segments.

In the system, a model-based algorithm was developed to link edges that are likely related to a vertical feature. An ideal image model of a vertical feature of interest, such as a street light pole, is shown in Figure 5. This vertical feature should have two vertical lines that are parallel to each other with a certain separation. Along a horizontal profile of the vertical feature, there are two edge points within a certain distance and each with an opposite edge direction. Based upon this model, a two-step line grouping algorithm was designed: (1) line grouping to form the primary line segments, and (2) line grouping to form the secondary line segments that are parallel to the corresponding primary line segments.

**Figure 4.** Directional masks coded from 0 to 5 for edge detection.

**Figure 5.** An ideal edge model for a vertical line feature of interest.
The first step is used to form the most reliable and distinct line segments (primary line segments). A conservative bridging strategy (weak linking process) is adopted. The second step is considered as an evidence collection, forming the line segments that are parallel to the associated primary line segments. A strong linking process is, therefore, applied to collect evidences. The algorithm operates as follows (see Figure 6):

(1) Forming primary line segments
(1.a) Scan the edge image, and locate an edge point that has not been processed;
(1.b) Take this edge as a starting point and construct a vertical "bridge" downwards, with a width of three pixels (shown in Figure 6). The length of the bridge is defined by the parameter BL (Bridge Length). BL controls the edge bridging range. In this step, the emphasis is placed on forming distinct line segments. Thus, BL is set to be 15 pixels. It is worth noting that the BL value is empirical based on the tests as well as the image resolution and scale.
(1.c) If an edge point within this bridge has a direction value that is "compatible" with the starting edge point, i.e., the direction difference between these two edge points is less than the predefined tolerance (±15°), then this edge point is recorded as a "compatible edge point." Otherwise, it is an "incompatible edge point."
  Case 1: If the distance between a compatible edge point and the starting point is less than half of BL, link this edge point to the starting point. Then take this edge point as a new starting point and go to (1.b).
  Case 2: If no compatible edge point that meets the above distance criterion exists, but the number of compatible edge points in this bridge is larger than the number of incompatible edge points, link the closest compatible edge point to the starting point. Then take this edge point as a new starting point and go to (1.b).
  Case 3: If neither of the above two cases exists, go to step (1.a).
(1.d) Finally, if the length of the formed line segment surpasses a threshold (50 pixels), it is accepted as a primary line segment and recorded in a line organization file.
Table 1 shows the structure of this file.

(2) Forming secondary line segments
This step is intended to find the secondary line segments that are parallel to the primary line segments. According to the vertical feature model shown in Figure 5, a secondary line segment should lie within a certain distance of a primary line segment of the same vertical feature, and its direction should be opposite to that of the primary line segment. If such a secondary line segment can be found, there is a high possibility that the vertical feature exists. The grouping scheme is similar to the first step except that BL is set to be 30 pixels, and both the direction and distance constraints are added.

(2.a) Scan the surrounding area of a primary line segment with a range of ±10 pixels, and locate an edge point whose direction is opposite to that of the primary line segment, and the distance to the primary line segment is more than three pixels.
(2.b) Take this edge as a starting point and construct a vertical "bridge" downwards, with a width of three pixels.
(2.c) Same as step (1.c).
(2.d) If the secondary line segment found is longer than one-third of the length of its corresponding primary line segment, the field "Parallel" in the line organization file will be set as "True." Otherwise, it is set to "False."

Representation of Line Segments
To facilitate the subsequent analysis, all the primary line segments are recorded in a line organization file. Each line segment has a unique identification number (ID). The relevant parameters of each line segment are computed and recorded. The starting and ending point locations, as well as the length of a line segment are basic parameters. The number of edges refers to the number of compatible edge points of that line segment. Line direction is defined as the averaged direction values of all the compatible edge points of the line segment. The field "Parallel" indicates whether or not a secondary line segment is found. In the field "Stereo," "True" means that corresponding line segments from a stereo image pair are found. The above described process is applied to both the left and right images of a stereo pair.

Feature Matching of Line Segments

Determination of the Candidates for the Corresponding Line Segment
First, the records in the line organization file are sorted based on the order: "Parallel" → "Length" → "Number of edges." High priority is given to the line segment whose "Parallel" is true, length is long, and number of edges is large. Sorting is done only for line segments in the left image. The line segment sorted highest is selected as the master line segment.

Second, the candidates of the corresponding line segment in the right image are chosen based on the direction compatibility. By searching the field "Direction" in the line organization file, a line segment in the right image is considered to be a candidate if its line direction is the same as that of the master line segment within a tolerance ±10°. If no such candidate can

Table 1. Data Structure of the Line Organization File

<table>
<thead>
<tr>
<th>Line ID</th>
<th>Image ID</th>
<th>Xs, Ye, Xe, Ye</th>
<th>Length</th>
<th>Number of edges</th>
<th>Direction</th>
<th>Parallel</th>
<th>Stereo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>110 (left)</td>
<td>357, 201, 359, 266</td>
<td>75</td>
<td>37</td>
<td>3°</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>2</td>
<td>110 (left)</td>
<td>342, 238, 341, 291</td>
<td>53</td>
<td>23</td>
<td>184°</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>3</td>
<td>111 (right)</td>
<td>........</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PHOTOMETRIC ENGINEERING & REMOTE SENSING

1404 December 2001
be found, the current master line segment is disqualified and the next one in the sorted list will be selected as the master line segment. The program will exit if no candidates can be found.

**Line Feature Matching**

Given a master line segment in the left image and one or more candidate line segments in the right image, line feature matching occurs as follows (shown in Figure 7):

1. Generate a set of sampling points along the master line segment at an interval of three pixels.
2. Create a master matching window centered at a sampling point with a size of 15 by 11 pixels.
3. Generate the corresponding epipolar line in the right image and determine the intersection of the epipolar line and the candidate line segments.
4. Create slave matching windows centered at these intersection points with the same 15-by-11-pixels size.
5. Use the weighted cross correlation algorithm to perform image matching between the master window and the search windows. Record the match point whose correlation coefficient value is highest and surpasses the predefined threshold (e.g., 0.7).
6. Repeat from (3.b) until all the sampling points are examined.
7. If the number of match points on a candidate line segment is more than half of the total number of sampling points used, line feature matching is considered to be successful, and the associated record of “Stereo: True/False” in the line organization file is then set to “True.”

Due to the large geometric distortions in mobile mapping imagery, a weighted cross-correlation algorithm was used here (Tao, 2000b). This algorithm places more weight on pixels that are closer to the matching window center. Having this weighting scheme, the algorithm is able to resolve, to some extent, the matching ambiguities caused by geometric distortion as well as the large changes in camera viewpoint.

**Object Space Verification**

Location constraint in object space is applied in this step. After the establishment of the match point pairs, the locations of these points in object space can be computed by using a photogrammetric intersection. The footprint location \((X, Y)\) of the vertical line feature with respect to the ground can be determined by \(X = \sum X_i/N\) and \(Y = \sum Y_i/N\), where \(X_i\) and \(Y_i\) are computed from the photogrammetric intersection of the \(i\)th match pair of the sample points on the vertical line feature and \(N\) is the total number of the match points obtained.

In the **VISAT** system, an object to be measured should lie within 35 meters of the camera in order to meet the accuracy requirement (Tao, 1999). This distance (depth) range is used to screen out the feature whose location is too far from the camera.

Features must be located on the side of the road, i.e., the offset between a feature’s location and the vehicle trajectory should be at least 1 meter. The offset of a feature is computed and used for a further check.

**Determination of the Object Location**

It was concluded in Li et al. (1996) and Tao (2000b) that use of multiple **VISAT** images for photogrammetric intersection can improve the measurement accuracy significantly. For object location determination, the multiple-image matching method using the multi-view epipolar constraint was used. A detailed description of this method can be found in Tao (2000b). Only a brief description is given below.

Shown in Figure 8, for any sample point, for instance, the point \(P_1\), on the line segment in the left image, three epipolar lines, i.e., left-to-right (EPL\(_{\text{lr}}\)), left-to-forward-left (EPL\(_{\text{lfl}}\)), and right-to-forward-left (EPL\(_{\text{rfl}}\)) can be computed. Its corresponding point, \(P_1\), in the right image can be determined by calculating the intersection of the epipolar line (EPL\(_{\text{lr}}\)) and the corresponding line segment, while its corresponding point, \(P_0\) (marked as “O”), in the forward-left image can also be determined by calculating the intersection of the epipolar line EPL\(_{\text{rfl}}\) and EPL\(_{\text{rfl}}\). Next, epipolar-line-based image matching is applied. Image matching is performed between the left image and the forward-left image and a match point, marked as “+,” of \(P_1\) is determined. Similarly, image matching is performed between the right image and the forward-left image to determine the match point, marked as “x,” of \(P_1\). Ideally, these three points, “+,” “x,” and “O,” should be at the same location. Due to the various error sources (georeferencing errors and image matching errors), often this cannot be the case. In the current program, if the distance between any two of them are within 1.5 pixels, the middle location of these two points is considered as the location of the matching point. For this matching, the master window size is set as 17 by 17 pixels and the search window size is set as 25 by 17 pixels. The correlation coefficient threshold is set as 0.70.

The above algorithm is applied for all the sample points. If the number of match points is more than half of the total number of sample points, we say that the corresponding line feature exists. The same process is applied to four consecutive images, i.e., forward and backward image pairs to find match points.

Finally, all the match points obtained are populated into a multi-point-based photogrammetric intersection software (Li
et al., 1996) to calculate the 3D coordinates. It is also realized that a straight-line constrained photogrammetric triangulation can be developed to obtain a more reliable 3D reconstruction of city streets. Most images do not have good contrast because the images were captured in the late afternoon. The street transportation database stored in an AutoCAD system was obtained.

The test image set was collected by the VISAT™ system in the city of Laval, Canada. The road scenes of images are mainly city streets. Most images do not have good contrast because the images were captured in the late afternoon. The street transportation database stored in an AutoCAD system was obtained.

The feature detection was tested first. Figure 9a shows an image with many vertical line features, including a light pole. It can be seen that the light pole does not have good contrast. The result obtained from the edge detection and line formation is given in Figure 9b (only line segments longer than 20 pixels are displayed). Major line features can be detected and, most important, they are grouped and linked very well.

In order to evaluate the overall performance of the approach, 55 objects were manually selected, mainly light poles and traffic signs, from the street database. By using the image georeferencing information, these objects were projected onto the images collected. After manually examining the images, we found that only one object had been removed, and the remaining 54 objects were retained.

The software was executed in a batch mode. After the automated feature detection, 216 line segments were recorded in the line organization file. Among them, 82 line segments were qualified for being master line segments. The automatic stereo matching of the master line segments was then performed. Fifty-seven out of 82 master line segments found their matches. After the object space verification step, 51 out of 57 were retained. After the manual check, 49 objects were found to be correct. This indicates that the overall success rate for this automatic approach can be estimated as 49.54 or 90.7 percent. Examining the remaining cases showed that three of them were caused by occlusions (such as trees or street parking vehicles) and that these objects were barely seen in the images. For the other two cases, stereo matching was not successful. This was mainly due to the large geometric difference in the stereo image pair.

A typical example of the entire process is given in Figures 10a through 10e. Figure 10a shows a stereo image pair. There are four objects of interest (marked in the image), but only object No.1 is examined for the current image pair (within the predicated location window). The other objects will be examined using other image pairs. Figure 10b is the output of edge detection based on six directional masks. The purpose of this step is to detect all the possible edges. As a result, many edges from trees and road boundaries remain in the image. Figure 10c shows the effect of direction-constrained edge thinning. A very large number of undesired edges have been filtered out, while the vertical edges have been kept. The result of the determination of vertically oriented edges is given in Figure 10d. It is clearly seen that edge pixels from trees are largely removed and those on the road boundaries are eliminated very effectively. The result of line formation and line feature matching is shown in Figure 10e. In this case, there are four primary line segments found in the left image. The master line segment is the left-bottom one (marked by 1). The secondary line segments for these primary line segments were all found. One primary line segment, for instance, 1, is in fact the secondary line segment to the other one for instance, 2. Two candidates of the corresponding line segment in the right image were found, namely, 0 and 1. The stereo matching was successful in this case. Object space location of this feature was then determined using the matching line segments 0 and 1. Another example is given in Figure 11 where the detected line features are overlapped on the original images.
Conclusions

An automatic approach to the inspection of vertically structured transportation objects is investigated in this research. The feasibility and the reliability of the proposed approach have been tested using real image data sets. Guided by an existing transportation object database, the proposed approach is capable of inspecting transportation objects from mobile mapping image sequences in an automatic and efficient fashion. The success rate of the developed approach is over 90 percent based on the test data set. This number is very promising because savings in both time and costs would be significant if the majority of manual work could be replaced by an automatic approach.

Automatic inspection is a complicated process. Use of multiple constraints derived from the mobile mapping system is of particular importance to the reliability of the developed approach. Many constraints have been incorporated into this approach, such as a vertical line constraint, a georeferencing constraint, location constraints in both the image and object domains, stereoscopic and sequential imaging geometry constraints, etc. The algorithms involved in this approach have been designed in a way that is tailored to the specific data source, for example, edge detection, edge thinning, line formation, line feature matching, object space verification, etc. It is demonstrated from this research that employing all possible constraints is the most effective way to reduce the complexity of the problem and thus leads to a reliable and successful solution.

However, the developed approach is specific to the given data source and constraints. In the future, more tests are required to improve the reliability and flexibility of the approach for various road scenarios as well as for other mobile mapping image sources.

Acknowledgments

The author would like to thank Drs. R. Li and M.A. Chapman for their great assistance to this research. The financial support from the Canadian Natural Science and Engineering Research Council (NSERC) and private industry sectors are acknowledged. The authors would also like to thank the anonymous reviewers for their constructive suggestions and valuable comments.

References


Figure 10. An example of the entire process. (a) Original image pair overlaid with the prediction windows. (b) Edge detection using six directional masks. (c) Direction-constrained edge thinning. (d) Detection of vertically oriented edges. (e) Line formation and matching (0-master line segment in the left image, 1—the corresponding line segment in the right image).


(Received 14 September 2000; accepted 23 March 2001; revised 27 April 2001)