Context-Aware Computing Framework for Improved Bridge Inspection

Manu AKULA[1], Atul SANDUR[2], Vineet R. KAMAT[3] and Atul PRAKASH[4]

[1] Ph.D. Candidate, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48109-2125; email: akulaman@umich.edu
[2] M.S.E. Student, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2125; email: athuls@umich.edu
[3] Associate Professor, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48109-2125; email: vkamat@umich.edu
[4] Professor, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2125; email: aprakash@umich.edu

ABSTRACT

Bridge inspections are tedious, time consuming and complex tasks in the field which require highly specific information pertinent to the decisions at hand. Bridge inspectors assess the condition of bridge components based on standard rating guidelines and previous bridge inspection reports. Most bridge inspections are currently conducted manually with little support from Information Technology and the captured data must be manually entered into a computer system for later retrieval. Context-aware computing promises to make inspections more efficient by automatically associating captured data with bridge design data in a database. This paper presents a context-aware computing platform that facilitates bi-directional flow of information and supplements field inspectors with relevant data to support their operations. The implemented context-aware computing framework automatically interprets the spatial-context of the bridge inspector based on the inspector's position and head orientation. The framework also provides methods to map and store the geometric representation of the bridge-inspection elements in a database. To validate the framework, a context-aware computing application was designed. The application coordinates the inspector's spatial-context with the bridge model in the database so that the inspector is provided with data that is relevant to his/her context. The application also facilitates bi-directional communication between the inspector and the bridge inspection database management system. Finally, the characteristics of context-aware computing supported bridge inspection routines are compared against the traditional (manual) approach to bridge inspection routines.

1. IMPORTANCE OF THE RESEARCH

1.1 Current Status of Bridge Inspection Technology

There are 604,460 public highway bridges (that are 20 feet in length or longer) in the United States, as documented in the National Bridge Inventory (NBI), that are subject to the National Bridge Inspection Standards according to the Bureau of Transportation Statistics [Moore, W. H., 2011]. The Michigan Department of Transportation (MDOT) lists 4,397 highway bridges under its jurisdiction, of which

341 are structurally deficient and 899 are functionally obsolete. MDOT regularly inspects thousands of bridges and overpasses on Michigan highways. MDOT employs more than 20 inspectors who have specialized training and work in teams of two. MDOT bridge inspectors evaluate bridges and assign condition ratings using the National Bridge Inspection Rating Scale [MDOT, 2011].

Bridge inspections are currently documented manually – the bridge inspector assesses the condition of a bridge based on standard rating guidelines and previous bridge inspection reports [Farrar, 2008]. The inspector carries the rating guidelines and the previous reports in their paper form along with the current report forms while conducting the inspection. In many cases, the inspector has to come to the site with excessive preparation and plenty of time and effort is wasted in searching, streamlining and retrieving relevant information. Upon returning to the office, after the inspector completes the inspection, the relevant data gathered on the field is entered into a database management system. The bridge inspection methodology, as currently employed in several states, is a time consuming and tedious process providing an opportunity to improve the efficiency of the process using information technology.

1.2 Background of Context-Aware Computing

Context aware computing is defined as the use of environmental characteristics such as a user's location, time, identity, profile and activity to inform the computing device so that it may provide information to the user that is relevant to the current context [Burrell and Gay, 2001]. Context aware computing can potentially enable mobile users in a wide variety of fields to leverage knowledge about various context parameters to ensure that they get highly specific information, pertinent to the decisions at hand. The concept of context-aware information delivery centers around the creation of a user centered, mobile, dynamic (indoor and outdoor) computational work environment which has the ability to deliver relevant information to on-site mobile users by intelligent interpretation of their characteristics in space and time so that they can take more informed decisions.

Context awareness can thus be of great value for civil engineering inspectors, emergency responders, security and military personnel. For example, interpreting the context of civil engineers during post disaster reconnaissance, or while conducting a bridge inspection, can allow bi-directional flow of streamlined information and thereby improve the efficiency of the decision making process. Context-aware applications can be used in providing support to complex, tedious and time consuming tasks. Civil engineers, fire fighters, military personnel and a host of other professionals stand to benefit from context-aware applications as it makes bi-directional flow of information more efficient and relevant based on a mobile user's context [Akula, M., et al., 2011].

Context-aware computing can tremendously reduce the time and effort involved in conducting bridge inspections by facilitating flow of information, in real-time, between a bridge information model and the on-site inspector. Based on the context of the inspector, streamlined data (such as relevant multimedia, applicable sections of rating guidelines, previous inspection ratings and notes for the bridge components in context) can be supplemented to field inspectors to support their operations in real-time. The process of updating the database with appropriate field data can also be similarly automated. The context-aware computing framework is described in section 2 of this paper.

2. CONTEXT-AWARE COMPUTING FRAMEWORK OVERVIEW

2.1 Context-Aware Computing Architecture for Bridge Inspections

The context-aware computing framework for bridge inspections is based on a database management system (DBMS) that provides access to 1) information regarding the geometric models of the bridge components of all the bridges and 2) an inspection database which stores information such as historic inspection data, inspection guides, etc. and has the ability to store future inspection data. The architecture of the context-aware computing framework is centered around a mobile inspector whose position and head orientation are tracked ubiquitously and are updated continuously, as shown in Figure 1. The position and head orientation information is used to determine the spatial-context of the inspector by constructing a geometric model of the spatial region in the inspector's vision. Non-spatial contextual parameters, such as the inspector's profile, task, etc., are coordinated with the list of bridge components/elements to determine the list of components/elements of operational importance. The geometric model of the inspector's spatial-context is coordinated with the geometric models of the bridge components/elements, that are of operational importance, to determine the list of relevant components/elements of the bridge that are in the inspector's spatial-context. The list of elements in context are then prioritized and passed to the application that interfaces with the inspector. The inspector interface application queries the inspection database for relevant information which is streamlined based on the prioritized list of contextual elements and the inspector's input.

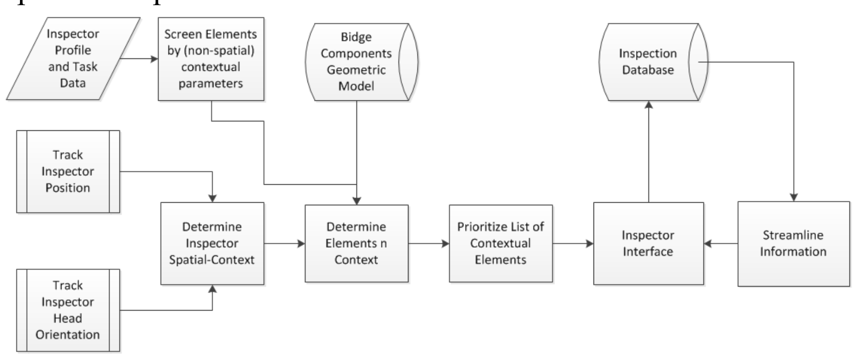
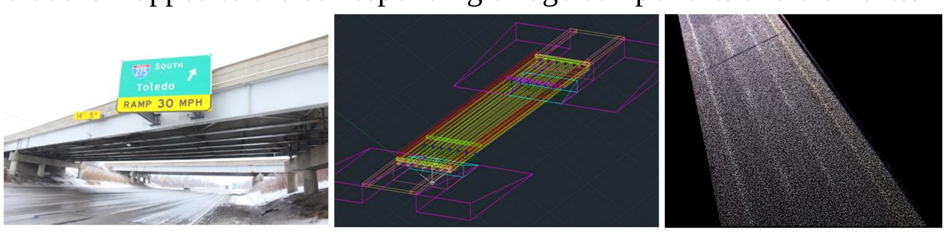


Figure 1: Architecture of the context-aware computing framework

Section 2.2 of this paper describes a method to map and store a geometric model of the bridge components and section 2.3 describes a method to construct the geometry of the inspector's spatial-context. The inspector interface application is designed as a customized application dedicated to the task at hand and allows for bi-directional communication with the inspection database. The bi-directional communication facility allows the inspector to access, edit and add information to the inspection database. The framework allows for the development of context-aware computing applications by drawing on the elements from the aforementioned framework and customizing the inspector interface application. Section 3 of this paper presents a context-aware computing application for improved bridge inspections based on the computing framework presented above.

2.2 Geometric Representation of the Bridge Components

The bridge components and constituent elements are modeled as a point cloud with each point in the point cloud being mapped to the element it represents geometrically. The point cloud can be developed using several methodologies including laser scanning, image based photo reconstruction and CAD model based point cloud development. The bridge selected as the Case Study Bridge (CSB) for this research is the I-275 highway bridge over Telegraph Road (US-24) in Monroe, Michigan, shown in Figure 2 (a). The point cloud was developed and mapped to the constituent bridge elements using the CAD model of the bridge in which the elements are modeled as either prisms or cylinders. As a first step, the CAD model of the bridge is registered to a particular coordinate frame designated as the Bridge Frame of Reference (BFR) which is used as the common coordinate frame in this research. The point cloud, with the desired density, is then created by generating points on the desired planar (or curved) surfaces and associating them with the appropriate element. For example, in the CAD model of the CSB, shown in Figure 2 (b), the bridge deck is modeled as a parallelepiped and the point cloud corresponding to the "deck surface" element is generated on the top surface (parallelogram) of the aforementioned parallelepiped. A photorealistic rendition of the deck surface point cloud is shown in Figure 2 (c). The point cloud generated through this methodology is comprised of points that are mapped to the corresponding bridge components and elements.



(a) (b) (c)

Figure 2: (a) The case study bridge, (b) the CAD model of the bridge and (c) a photorealistic rendition of the point cloud model of the entire deck surface.

The CSB is modeled as a combination of the following constituent components (and corresponding elements) – Deck (3 deck surface, 1 expansion joint, 3 other joint, 6 deck railing and 6 deck sidewalk elements), Superstructure (7 girder and 14 bearing elements), Substructure (2 abutments, 8 piers and 2 slope protection elements) and Approach (2 approach pavements and 4 approach sidewalk elements) components. The point cloud, and the mapping to the bridge components and elements, is stored in a database which can also be used as the inspection database.

2.3 Geometry of the Inspector's Spatial-Context

The context-aware computing framework computes the region of space visible to a mobile inspector as a viewing frustum shaped as a frustum of a cone. The position and head orientation, in terms of roll, pitch and yaw, of the inspector are used to define the line of sight and the region of space that is in the inspector's field of view is computed as shown in Figure 3.

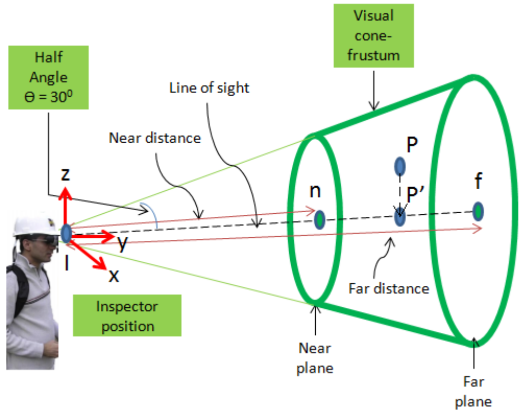


Figure 3: A geometric representation of the mobile inspector's viewing frustum.

The cone is computed using the line of sight, the near and far plane distances and the angular extent (half-angle) of the field of view of the inspector. Objects closer to the inspector than the near plane and beyond the far plane of view are considered to be out of sight and context.

2.4 Identifying Elements in the Inspector's Spatial-Context

To identify the elements in the inspector's spatial-context, the framework processes the point cloud to compute the list of points that are located inside the viewing cone-frustum. A point P, in the point cloud, is considered to be of contextual relevance if - 1) the distance between the inspector's eye and the point as projected on to the line of sight (IP') is greater than the near plane distance (In) and less than the far plane distance (If) and 2) the angle between the vector from inspector's eye to the point (IP) and the line of sight vector (In or If or IP') is less than the angular field of view (half-angle) of the inspector. The bridge components/elements corresponding to the list of points within the viewing cone-frustum are identified as the elements in the inspector's spatial context.

3. IMPROVED BRIDGE INSPECTION APPLICATION

3.1 Improved Bridge Inspection Application Overview

The improved bridge inspection application is designed based on the context-aware computing framework described in section 2.1. The inspector's position is tracked ubiquitously by a hybrid tracking system [Akula et al., 2011] based on integrating the Global Positioning System (GPS) and another infrastructure based positioning system, to track the inspector when the inspector is under the superstructure and the GPS signal is unavailable. The inspector's head orientation is tracked by an electronic compass mounted on the inspector's hard hat and the inspector's spatial-context is constructed as explained in section 2.3. Based on the inspector's spatial-context, the elements in the inspector's context are identified and passed onto the inspector interface application described in section 3.3.

3.2 Improved Bridge Inspection Database

The bridge inspection database stores two facets of bridge information – 1) the geometric information and 2) the maintenance information. The database stores geometric information regarding bridges which are divided into their components (Deck, Substructure, Superstructure, Approach and Culvert) according to NBIS and these components are further subdivided into their constituent elements (such as Deck surface, girders, bearings, abutments, piers, railings, etc.). The research presented in this paper implements the list of elements in the database at two levels – 1) the NBIS guided MDOT inspection element list, as mentioned previously in section 2.2, and 2) the PONTIS element schema.

The database stores bridge inspection information such as ratings, comments, reports and multimedia (pictures, videos, guidelines and relevant documents) which is used by the context-aware application as a repository. The database also allows tagging multimedia with comments regarding observations made by the inspector. The database is stored in a remote server and is queried using wireless 3G networks. The entity-relationship model of the database is shown in Figure 4.

3.3 Inspector Interface Application for Improved Bridge Inspection

The inspector's spatial context parameters are acquired and updated continuously and are used to compute the region of space currently in context. This region of space and the geometry of the bridge elements are checked for intersection to determine the list of elements currently in context which is then passed onto the inspector interface application. The application allows the inspector to select the specific element of interest from the list of elements in context however this selection can be automated by implementing prioritization algorithms on the list of elements in context. After a particular element is selected the application allows the inspector to retrieve historic bridge inspection ratings and the corresponding inspector comments.

The developed application also allows the retrieval and visualization of multimedia information such as pictures, videos, parts of the rating guidelines and other documents relevant to the selected element in tabbed viewing screen space as shown in Figure 5. Based on the historic inspection information and the current status of the bridge elements the inspector evaluates the element, updates the ratings and uploads the supplementary inspection comments/multimedia to the database using the functionality provided in the application.

After the inspector completes evaluating all the elements of the bridge, the application computes and updates the ratings of the components and the bridge using the element ratings. The application also provides the functionality to create and upload bridge inspection reports to the database in the form of a PDF file. The bridge inspection report PDF is compiled in the same format as the MDOT inspection report form with the additional aspect of embedded multimedia at relevant locations to supplement the inspector's observations and to justify the evaluation. The printout of the PDF file resembles a compiled MDOT bridge inspection report. The application has been designed to allow regions on the uploaded images to be tagged with comments and observations made by the inspector. A detailed discussion of the challenges in implementing this application is beyond the scope of this paper.

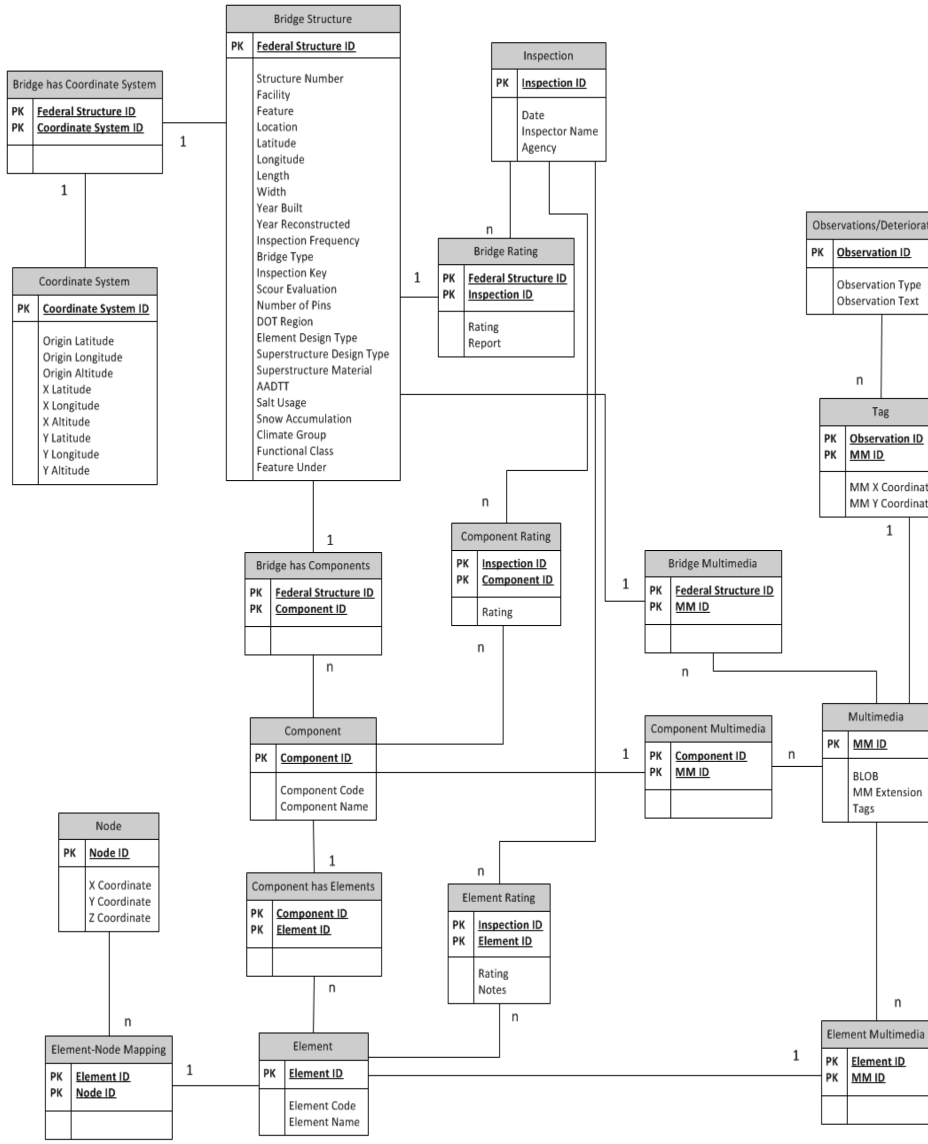


Figure 4: Entity relationship model of the improved bridge inspection database.

4. ACCURACY IN INTERPRETING SPATIAL-CONTEXT

The accuracy in interpreting the list of elements in the user's context is dependent on the accuracy of constructing the inspector's spatial context geometry, which is defined by the spatial parameters of the inspector – the position and the head orientation, and the geometry of the bridge elements. In order to quantify the uncertainty in interpreting the spatial context of the inspector, the authors developed and analyzed a simulation study, presented in section 4.1.

4.1 Simulation Study of Uncertainty in Interpreting Spatial-Context

The simulation study was performed on two geometric data sets where the elements in the database were classified and compiled using 1) the NBIS guided MDOT inspection element list and 2) the PONTIS element schema.

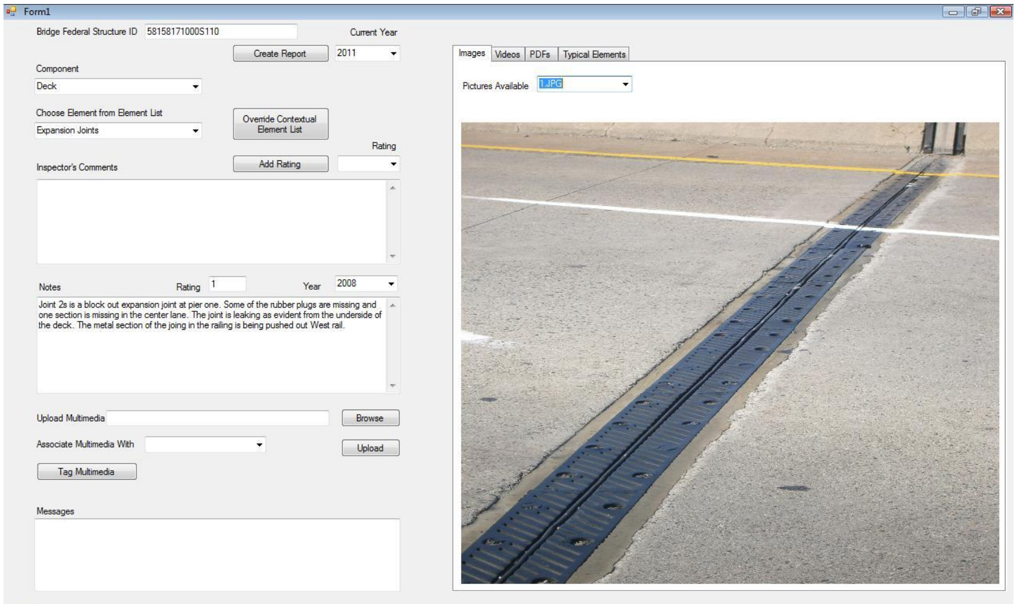


Figure 5: The inspector interface application for improved bridge inspections.

For each of the aforementioned datasets the authors performed simulation experiments for the following combinations of positioning and head orientation tracking technologies:

1) The GPS position is tracked using Real Time Kinematic GPS which is highly accurate [Cobb, 1997]. The typical nominal RMS errors for these RTK-GPS systems is 2.5 centimeters horizontally and 5 centimeters vertically. However, GPS may be unavailable in some areas under the bridge and for such areas the position is tracked by integrating the GPS with an Ultra-Wide Band based positioning technology [Akula et al., 2011]. The positioning system to be used was the DART UWB positioning system with a maximum uncertainty of 50 cm [Khoury and Kamat, 2009]. The head orientation was assumed to be tracked by TCM5 compass which has a manufacturer specified uncertainty of less than 0.5 degrees [PNI Sensor Corporation, 2011].

2) The GPS position is tracked using Wide Area Augmentation System capable Garmin eTrex GPS which has been demonstrated to have RMS uncertainty of 3 m [GPS Information, 2011] which was modeled as a Weibull distribution [Wilson, 2011]. The GPS denied positioning system and the orientation tracking system were assumed to be the same as above. The aforementioned technologies which determine the position and head orientation are summarized in Table 1.

Table 1: Summary of technologies employed to measure spatial parameters.

Technology	Spatial Parameter Measured	Error Characterization	Advantages	Disadvantages
RTK-GPS	GPS Position	RMS error of 5 cm	High accuracy	GPS may not be available under the bridge; Large payload
WAAS capable Garmin eTrex	GPS Position	RMS error of 3 m	Low payload	GPS may not be available under the bridge; Low accuracy
UWB Positioning	Position in GPS denied areas	Maximum error of 50 cm	Good accuracy	Infrastructure must be installed
TCM5 compass	Roll, Pitch and Yaw	Angular RMS error of 0.5 degrees	High accuracy	Behavior may be erratic when close to metalic structures

The simulation study was conducted by using the following methodology:

1) 508 position-head orientation combinations that are encountered during a typical inspection routine of the CSB were documented. The list of elements in spatial context for each of the aforementioned combinations were determined and served as the ground truth list of elements.

2) For each of the 508 combinations the simulation determines the position-orientation combinations had they been acquired using the tracking technologies by sampling errors from the aforementioned probabilistic error distributions and adding them to the ground truth parameters. These newly calculated spatial parameters are used to determine the list of elements in spatial-context corresponding to the list of elements for the ground truth position-orientation combination.

3) The lists of elements acquired from simulating the tracking technologies are compared to the corresponding lists of elements determined by the ground truth data to evaluate two probability metrics – (a) the probability of the tracking technology missing an element present in the ground truth case and (b) the probability of the tracking technology wrongly identifying the list of elements in context, not considering false positive cases (where all objects/ information that are truly in context are identified as in context resulting in additional information but no information loss). The results of the simulation study are summarized in Table 2.

Table 2: Summary of results of the simulation study cases.

Element list schema	Tracking Technology	Probability of missing an element	Probability of identifying the element list wrongly
MDOT - NBIS	RTK-GPS with UWB; TCM5 compass	0.08 %	0.56 %
MDOT - NBIS	WAAS capable Garmin eTrex with UWB; TCM5 compass	1.27 %	6.74 %
PONTIS	RTK-GPS with UWB; TCM5 compass	1.05 %	15.41 %
PONTIS	WAAS capable Garmin eTrex with UWB; TCM5 compass	3.51 %	25.72 %

5 SUMMARY AND CONCLUSIONS

The paper presents a context-aware computing framework to facilitate the development of context-aware bridge inspection applications. The paper also presents a context-aware application for improved bridge inspections. The uncertainty in interpreting the context using the aforementioned framework was evaluated using a simulation case study, which is also presented. The simulation study concludes that the accuracy of interpreting context is dependent on 1) the element classification schema and 2) the tracking technologies used to determine spatial parameters. The MDOT classification schema inferred from NBIS is relatively free of uncertainties compared to the PONTIS classification schema. While uncertainty can be reduced by using high precision tracking technology like RTK-GPS, it has the disadvantage of high payload. This problem is addressed by designing the application to allow the inspector to override the list of elements identified by the framework in order to manually choose the element of interest.

The improved bridge inspection application reduces the time and effort spent on the field and in the office in searching, preparing, retrieving and documenting information compared to the traditional (manual) inspection methodology. The application also allows multimedia mapping in order to record observations with more detail compared to the manual counterpart. The context-aware framework presented in the paper can be used to develop applications that can be used for training bridge inspectors and to help inspectors make more informed evaluations by allowing comparisons between multiple bridge elements across various structures with similar characteristics thus leading to more objective evaluations than manually documented inspections.

ACKNOWLEDGEMENTS

The authors would like to gratefully acknowledge the generous support offered by the U.S. Department of Commerce, National Institute of Standards and Technology (NIST) Technology Innovation Program (TIP) under Cooperative Agreement Number 70NANB9H9008. Additional support was provided by the University of Michigan and the Michigan Department of Transportation (MDOT). The intellectual contributions offered by Prof. Sharada Alampalli and Dr. Mohammed Ettouney are greatly appreciated.

REFERENCES

Akula, M., Dong, S., Kamat, V.R., Ojeda, L., Borrell, A. and Borenstein, J., (2011), “Integration of Infrastructure Based Positioning Systems and Inertial Navigation for Ubiquitous Context-Aware Engineering Applications”, *Advanced Engineering Informatics*, Elsevier, Volume 25 Issue 4, 640-655.

Burrell, J. and Gay, K., (2001), “Collectively defining context in a mobile, networked computing environment”, *Proceedings of the Conference on Human Factors in Computing Systems*, Association for Computing Machinery (ACM), New York, NY, 231–232.

Farrar, M. M., (2008), “The AASHTO Manual for Bridge Evaluation”.

GPS Information (2011), “Garmin's eTrex Legend C - With Horizontal Compass and Antenna” <http://gpsinformation.us/vistacolor/etrexvistacolor.html> (November, 21, 2011)

Khoury, H. M., and Kamat V. R. (2009), "Indoor User Localization for Context-Aware Information Retrieval in Construction Projects", *Automation in Construction*, 18 (4), Elsevier Science, New York, NY, 444-457.

MDOT, (2011), “Highway Bridge Report – October 1, 2011”, Michigan Department of Transportation, Lansing, MI. 1-2.

Moore, W. H. (2011), “Table 1-28: Condition of U.S. Highway Bridges”, *National Transportation Statistics*, Bureau of Transportation Statistics, Washington, D.C.

PNI Sensor Corporation (2011), <http://www.pnicorp.com/products/tcm-legacy> (November, 21, 2011)

Wilson, D. L. (2011), “GPS Horizontal Position Accuracy”, GPS Accuracy Webpage < http://users.erols.com/dlwilson/gpsacc.htm> (November, 21, 2011)