WEB-GIS BASED BRIDGE INFORMATION DATABASE VISUALIZATION ANALYTICS AND DISTRIBUTED SENSING FRAMEWORK

by

Yonghong Tong

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Infrastructure and Environmental Systems

Charlotte

2013

Approved by:

____
Dr. Shen-En Chen

____
Dr. Edd Hauser

____
Dr. Aidong Lu

____
Dr. Srinivas Pulugurtha

____
Dr. Don Chen

ABSTRACT

YONGHONG TONG. Web-GIS based bridge information database visualization analytics and distributed sensing framework. (Under the direction of DR. SHEN-EN CHEN)

The national bridge system plays very important role in society operations ensuring mobilities that can sustain social and economic growth. Recent increasingly growing concerns about the safety of existing bridges are shared by highway agencies at all levels of government, including federal, state and municipal. To provide a user-friendly and effective environment and services for accessing and analyzing the National Bridge Inventory (NBI) database, a powerful bridge data management system needs be developed to assist the bridge managers or professionals to manage and maintain effectively and efficiently the national bridge system.

The objective of this research is to develop a Web-GIS (geographic information system) based bridge information database visualization analytics and distributed sensing framework for nation-wide bridge system management. This is accomplished by integrating modern technologies including GIS, Internet, database, remote sensing, visualization, and smartphone technologies. The objectives of this study include: 1) establishment of a system framework for effective use of current available bridge condition data and volunteering sensing data; 2) development of visualization and visual analytic applications appropriate for bridge information; 3) development of user-defined criteria query for decision-making support; and 4) development of a remote sensing database to aid engineers and other professionals in accessing, retrieving and manipulating information from the bridge database. The citizen-based sensors for bridge

monitoring utilize voluntary information-sharing from individuals as a monitoring technique.

The Web-GIS based Bridge Management System (BMS) framework developed in this research allows centralized data collection and data visualization analytics at any place and any time. It is intended as a critical step towards rapid bridge diagnostics using an integrated sensing data approach. Current bridge management is predominantly at state level. Furthermore, by adopting the "citizen sensor" concept, public data can be added into the bridge database as additional information for bridge management.

The outcome of this research is a framework called: "Bridge-WGI." The six critical modules formed the core of the framework, which are: 1) bridge database systems; 2) general bridge information visualization; 3) bridge information analytical visualization; 4) user-defined criteria query; 5) citizen sensing application in bridge monitoring; and 6) remote sensing database application.

The Bridge-WGI framework demonstrates the capabilities of Web-based BMS can be accomplished via the integration of several technologies. These capabilities include: 1) application of volunteering sensing; 2) flexible accessibility via Internet; 3) several advanced visualization of bridge data; 4) bridge data integration; and 5) online user-defined query for decision making support.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to many people for their invaluable guidance, support and encouragement during my Ph.D. study at UNC Charlotte. First of all, I would like to express my sincere appreciation and gratitude to Dr. Shen-En Chen, my mentor and advisor. He kindly mentored me throughout my research and study. It has been a great honor to work with him and to learn from his experience. I would also like to express my sincere appreciation and gratitude to Dr. Edd Hauser for his kind suggestions and encouragement.

I would like to thank Dr. Aidong Lu for her guidance in using visualization techniques and her kindly suggestions. I am indebted to Dr. Srinivas Pulugurtha for his constructive suggestions in GIS applications and to Dr. Don Chen for his advices.

I would also like to thank Dr. David Boyajian for his early guidance and help. I also would like to show my thanks to Dr. David Young, Dr. John Daniels, and Dr. Jy Wu for their supports during my study. Many thanks also go to those professors, administrative staff, and students of Civil and Environmental Engineering Department who have provided assistance throughout my research.

This research is based on a project which is supported by grant number DTOS59-07-H-0005 from the United States Department of Transportation (USDOT), I would like to acknowledge contributions of Mr. Caesar Singh, the Program Manager at USDOT, Mr. Garland Haywood of NCDOT, and other people for supporting my research. I would also like to thank the research project group members, Dr. Kaoshan Dai, Dr. Wanqiu Liu, Andrew Stein, Haitao Bian, Libin Bai, Christopher Watson, Benjamin Smith, and Lu Sun. It was a great time to working together with them.

Last but not least, I wish to express my very special gratitude to my wife and my son, for their continuous support, understanding and encouragement throughout my Ph.D. study. I wish to express my special gratitude to my family in China, especially my mother, for her never-ending love, care, understanding and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AADT	annual average daily traffic
AASHO	American Association of State Highway Officials
AASHTO	American Association of State Highway and Transportation Officials
ADOT	Arizona Department of Transportation
ADT	average daily traffic
ASCE	America Society of Civil Engineers
ASCII	American Standard Code for Information Interchange
BIRM	Bridge Inspector’s Reference Manual
BMS	bridge management system
CRS	commercial remote sensing
CSS	cascading style sheets
CSV	comma separated values
DBMS	database management system
DOT	department of transportation
FHWA	Federal Highway Administration
GIS	geographic information system
GPS	global positioning system
GSS	geography and spatial science
GUI	graphic user interface
IDE	integrated development environment
IRSV	integrated remote sensing and visualization
ISTEA	Intermodal Surface Transportation Efficiency Act

IT	information technology
LaDAR	laser detection and ranging
LiBE	LiDAR bridge evaluation
LiDAR	light detection and ranging
MAPC	multiple axis parallel coordinate
NBI	national bridge inventory
NBIS	National Bridge Inspection Standards
NDE	non-destructive evaluation
NDI	non-destructive inspection
NTI	non-target-intended
RITA	Research Innovative Technology Administration
SDK	software development kit
SDOT	State Department of Transportation
SDSS	spatial decision support system
SHM	structure health monitoring
SI	spatial information
TI	target-intended
UDCQ	user-defined criteria query
UNCC	University of North Carolina at Charlotte
USDOT	U.S. Department of Transportation

CHAPTER 1: INTRODUCTION

1.1 Background

The national bridge system plays very important role in society operations ensuring mobilities that can sustain social and economic growth. In the United States, it is estimated that the total investment needed to improve the nation's infrastructure is about $930 billion over 5 years, while there is a projected shortfall around $549.5 billion (ASCE 2009). Allocating the limited funds to maximize bridge maintenance efficiency is a very important issue to bridge managers. A better management strategy should be introduced to optimize funding allocations in both bridge inspection technology and bridge management.

The sudden collapse of the I-35 Bridge in Minnesota in 2007 increased the awareness of bridge safety issues in the scientific and engineering communities. Abudayyeh et al. (2004) described that more than 70% of in-service bridges were built before 1935, which means these bridges are more than 76 years old, and face heavy deterioration. A recent report from the America Society of Civil Engineers (ASCE) showed that more than 26% of U.S. highway bridges are identified as either structurally deficient or functionally obsolete (ASCE, 2009). Increasing concerns about bridge safety has been expressed by all levels of highway agencies including federal, state and local governments, which translates to a demand for a more efficient and effective maintenance program for the national highway bridge system. For a bridge management program to be effective, it must be able to provide precise interpretation and clear information visualization of the bridge conditions so that the bridge managers can have a clear understanding of the bridge conditions. A strategic maintenance decision-making support system can help the bridge manager to make effective planning, which may include whether the bridge needs to be repaired or be rebuilt and how and when to do the maintenance.

In the United States, there are more than 600,000 bridges that constitute the highway bridge infrastructure. To keep the national bridges safe functionally and structurally, bridge inspection and bridge maintenance play a very important role in bridge management. Bridge management systems (BMS) are designed and implemented to provide efficient and effective maintenance strategies for the national bridge system.

Several weaknesses of existing BMSs can be identified: First, existing bridge inspection is a multi-step process that may take months to accomplish. The entire process, which includes preparing for inspection, conducting on-site inspection, completing the inspection forms, inputting the inspection data, and storing the data in state bridge inventory, may last several days for state agency and 180 days for local agency. After further processing by state DOT staff, the final bridge inventory data are then submitted to the USDOT (Figure 1). This prolonged process may translate into additional costs to the DOT. In a time of cyberspace connectivity, the time period required for data integration in NBI reflects that the process has not benefit from modern IT technology.

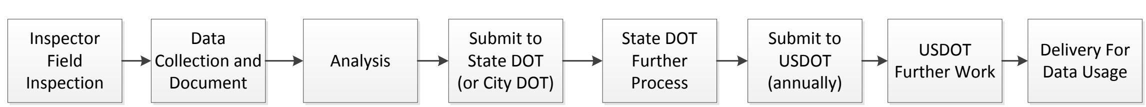


Figure 1: Existing bridge inspection process

Existing BMS software has some additional weaknesses, including 1) limited Web-based application, 2) limited capability to share data information among DOTs, 3) flat graphic user interface, and 4) limited data types. Existing NBI does not include any imagery data such as remote sensing data, photos or any public volunteered information (FHWA 1995).

Furthermore, due to the 1-year or 2-year cycle of inspection and the discrete bridge locations, it is appreciably difficult for DOTs to maintain real-time observations of these bridges. The dispersedly distributed geo-spatial bridge density makes network-level bridge monitoring difficult.

1.2 Problem Statement

Current research will address the above issues mentioned by exploring possible solutions to the following questions:

1). Since existing bridge inspection is based on an one-year or two-year cycle, there is a likelihood damage or incident might occur between the inspection cycle. Can modern IT technology be used to close the data gap between each inspection cycle, thus reduce the cost of repair and increase safety?

2). Current bridge inspection process is a multi-step procedure. It begins when an inspector performs a field inspection and ends when the data is submitted to the USDOT. This process could take several months to complete. Since time is important to ensure the safety of the bridge, is there a possible way to shorten the process?

3). Most current BMSs adopt data in rigid platform media, such as texts, graphics, and images. By adopting emerging techniques, such as computer visualization, can the understanding of the database and bridge conditions for managers be enhanced?

4). Most current BMS software are installed on standalone computers, intranet, or local area network (LAN). There may be drawbacks in software management, for example, to upgrade Pontis, a widely used BMS, a whole package needs to be reinstalled which resulted in extra labor times. In addition, for every update, the data would have to be migrated into a newer version. Hence, can integrating of database and Internet technologies overcome the weakness?

1.3 Purpose and Scope of Research

This purpose of this study is to develop and implement a Web-GIS based bridge information visualization and distributed sensing framework. The Web-GIS based platform allows advanced data visualization and potential future cloud centric applications. The specific objectives of the framework development are:

(1) To provide a better and clearer interpretation for the bridge condition information via visualization techniques;

(2) To provide an assistance module for decision-making support for bridge maintenance;

(3) To provide data entry functionalities for pertinent information including text, images, and documents to expand bridge database;

(4) To provide citizen sensor damage report application; and

(5) To integrate remote sensing data to bridge management system.

1.4 Research Methodology

Considering the advantages of both GIS and Internet, Web-GIS will be used as the main platform for this research. In this development, several programming languages and development technologies will be applied. The NBI bridge database will be the primary database applied in this research. Figure 2 shows the flowchart of the research procedure.

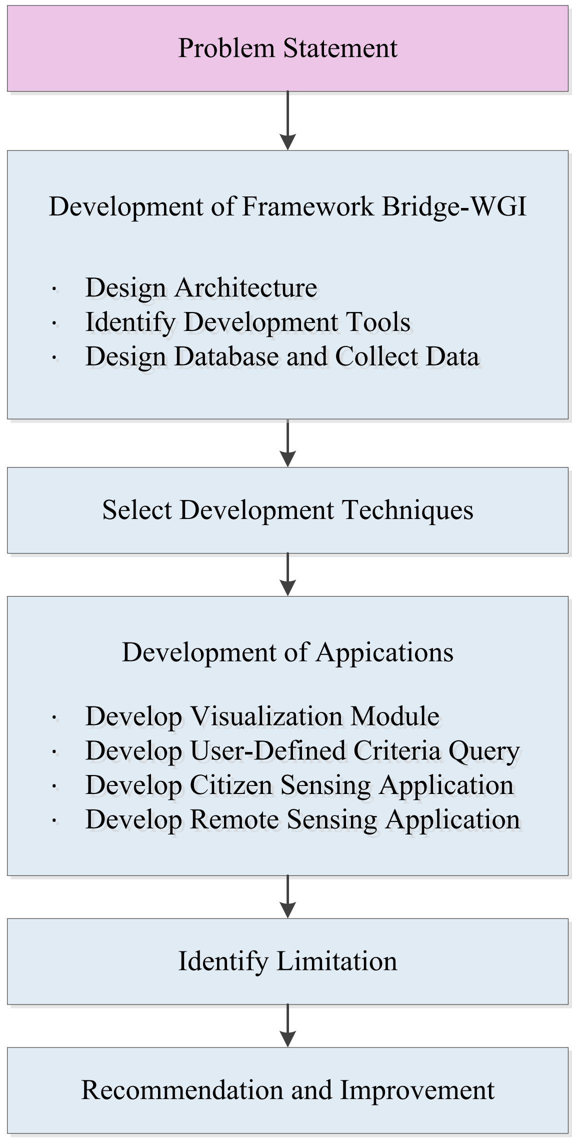


Figure 2: Research process flowchart

From literature reviews, the limitations of available BMSs and suggestions for improvement have been identified (section 1.2). The following sections describe how insights from literatures have been applied to the system framework design.

(1) Design Architecture

The architecture and features of the Web-GIS interactive visualization framework for bridge management system is first to be designed, including the hardware and software selections, database system design and possible other modules.

(2) Identify Development Tools

Well-selected development tools and database system can affect the success of any implementation. Programming languages for Web-based database application including JavaScript, HTML, PHP and Python, and other compiled languages including Java, C# and VB.net, have been recognized and used in current study. The MySQL is a popular database system for Web application. In this design, JavaScript, HTML, PHP and MySQL are mainly applied for the development.

(3) Design Database System

Since the most important component of this framework is database system, a stable and well-designed database system should be established so that future development can easily access this foundation-level component. The database system in this design consists of three parts: 1) bridge data, 2) remote sensing data (LiDAR scan and/or Fly-over aerial image), and 3) the citizen sensing data.

(4) Select Development Techniques

The techniques for this development should be first identified before the implementation. For citizen sensor data collection and bridge condition reporting, mobile computing technology, such as smartphone, will be applied for bridge localization and damage reporting.

(5) Develop Visualization Module

In the visualization module, a map-based visualization interface should be developed; and user can read the related information of a selected bridge. Information visualization analytics should be developed to assist the bridge manager's decision-making process for bridge management. The visualization module displays selective pertinent information about the bridge such as the geospatial location, the average daily traffic, the age of bridge, and the condition rating for deck, superstructure and substructure.

(6) Develop User-Defined Criteria Query Module

A user-defined criteria query (UDCQ) will be adopted, to meet the specific interest of the user. UDCQ methods can provide decision-making support for bridge management, and the bridge manager can choose various bridge parameters to generate user-defined criteria query based on his own knowledge.

(7) Develop Citizen Sensing Application Module

The citizen sensor is suggested as a tool for constructed facility monitoring, which is important to the management of systems where several discrete structures that may be miles away from each other. The sensor module opens the portal for users to upload bridge information via the Internet.

Finally, this research will identify limitations and weaknesses of the Internet-based framework. Acknowledgement of study limitations provides an opportunity to demonstrate the researcher's thought about the research problem.

1.5 Dissertation Organization

Chapter 1 gives an introduction and problem statement for this research. Chapter 2 provides the literature review for this research. In literature review part, technologies used in the research, from National Bridge Inventory, Bridge Management System, GIS, Database, to smartphone technology, will be introduced. Chapter 3 discusses the architecture and structure of the proposed visualization framework, from both hardware configuration and the software development perspectives. Chapter 4 includes three modules: The first module introduces the general bridge information visualization; the second module presents the analytical visualization for bridge database; and the third module is a user-defined criteria query functionality which is developed to assist the bridge manager to do further decision-making for bridge conditional analysis. Several visualization techniques are applied for this visual analytics application. Chapter 5 discusses the citizen sensing application. Chapter 6 introduces the inclusion of remote sensing data into the database. The remote sensing data includes LiDAR scan images and fly-over aerial images. LiDAR scan image can be used to detect damage from the surface of a bridge. Chapter 7 offers discussions on potential contributions and limitations of current system. Chapter 8 presents the conclusions. Chapter 9 presents future study.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

The National Bridge Inventory (NBI) was initiated to monitor bridge operations and safety. NBI provides the national standard in bridge information collection (Chase et al., 1999 and Ahlborn et al., 2010). NBI consists of data extracted from the state bridge management systems (BMS). There are several BMSs, with similar functionalities, being applied to management and maintenance of bridges. It is a fact that the performance and functionality of bridge management system is widely based on the storage and performance of its database (Atzeni et al., 1999).

The proposed research suggests use of modern Internet tools for bridge data management. The development of technologies, such as the Internet, geographic information system (GIS), database system, wireless communication, and computer visualization, will result in more interconnected products, which can improve on bridge management and safety.

2.2 Bridge Inspection and Maintenance

(1) Bridge Inspection History

During the bridge construction boom of the 1950s and 1960s, there were little focus on bridge safety inspection and maintenance. After the collapse of Silver Bridge at Point Pleasant, West Virginia that killed 46 people in 1967, there is a rise in national awareness of bridge safety issues (Rossow, 2012). In 1968, the U.S. Congress required the establishment of a national bridge inspection standard to ensure the safety of the bridge system (Liu, 2010). Since the 1950s, the U.S. transportation system has expanded to become “the largest and most modern highway system in the world” (Roberts and Shepard, 2000).

In 1971, the National Bridge Inspection Standards (NBIS) came into being and national policies are established for bridge inspection and maintenance. The Federal Highway Administration (FHWA) has made considerable efforts to the development of NBI, which helps bridge inspectors inspect and evaluate the national bridges accurately (Rossow, 2012). The NBI database is the most extensive repository of data on highway bridges in the United States. The NBI database contains more than 100 items for each bridge, including basic information such as structure type, structure length, year built, design load, inspection date, year reconstructed, kind of material, and geospatial information such as latitude and longitude, bridge condition rating information such as deck, superstructure and substructure, etc.. Each item has a specific coding structure, which is described by the “Recording and Coding Guide for the Structure Inventory and Appraisal of the National Bridges” (FHWA, 1995). Based on the NBI database, bridges are generally inspected by bridge inspectors following a one-year or two-year cycle. Several months are needed to analyze the collected data and obtain useful information for bridge maintenance.

(2) Bridge Inspection Procedure

Bridge inspection is the first step to determine the present condition of a bridge for evaluating the performance of the bridge. Depending on the particular bridge agency who conducts the bridge inspection, various inspection methods could be used. The Bridge Inspector’s Reference Manual (BIRM), which is sponsored by FHWA, aids the inspector with programs, procedures, and techniques for inspecting and evaluating a variety of in-service highway bridges (Ryan et al. 2006).

Highway bridges are inspected by the state departments of transportation (DOTs) and equivalent transportation authorities. NBI data are collected bridge inspector using functional on-site inspection. To assist inspector in bridge inspection and the identification of inspection problems, the American Association of State Highway and Transportation Officials (AASHTO) provides guidance on bridge inspection: “Guide Manual for Bridge Element Inspection” (AASHTO 2011). Typical routine inspection is visual inspection, which focuses on assessing bridges from both structural and functional aspects, including physical structure conditions, traffic volume and patterns, surrounding environments, etc. Figure 3 shows the bridge inspection procedure proposed by Arizona Department of Transportation (ADOT).

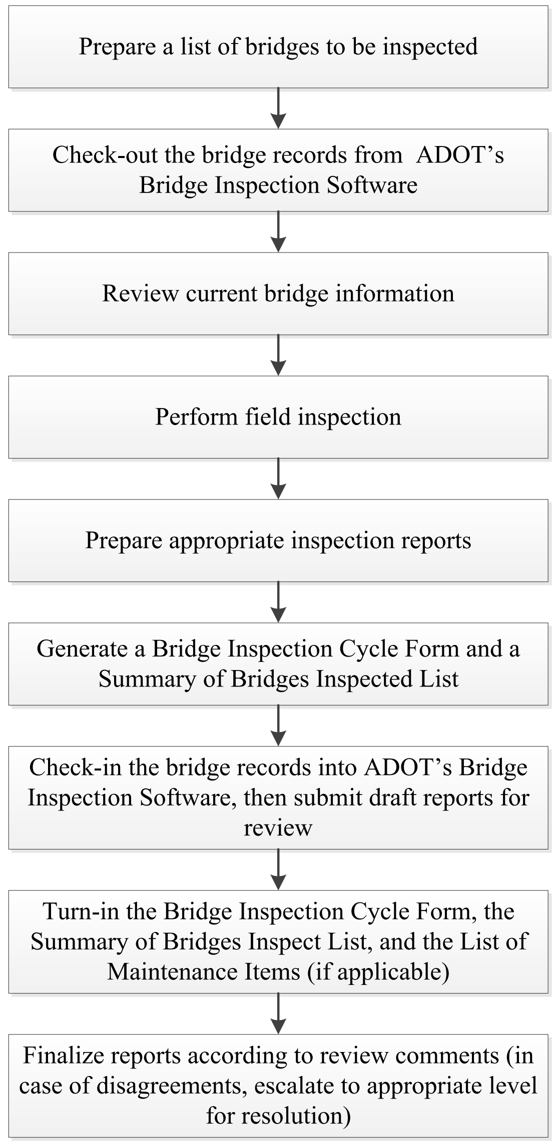


Figure 3: State-level bridge inspection flowchart (Rossow, 2012)

Based on the results of on-site inspections, bridge condition ratings are determined and placed into the state BMS system, which forms the annual basic NBI data. The NBI condition rating system provides a high-level summary of how and where to perform maintenance. After further processing by the state Department of Transportation (DOT), the NBI data report will be submitted to the United States Department of Transportation (USDOT) annually in ASCII file format.

(3) Bridge Condition Rating

To unify recording and coding of bridge information, the FHWA coding guide (FHWA, 1995) suggested a bridge rating for three main elements of a bridge: 1) the deck; 2) the superstructure; and 3) the substructure. Figure 4 illustrates the three main elements of a bridge.

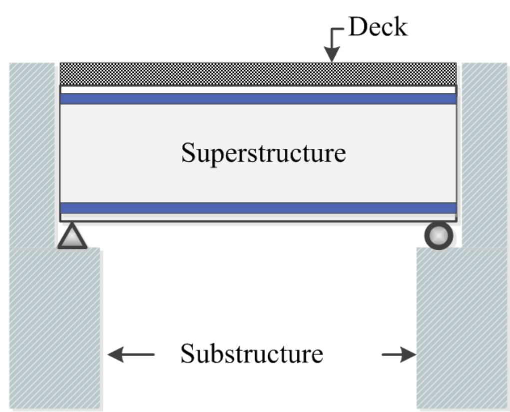


Figure 4: Bridge main elements (Deck, Superstructure and Substructure)

Condition ratings based on field inspection along with written comments, the classified condition rating is one digit number which given by the field inspector, from “0” through “9”. “9” stands for the best condition or a relatively new bridge, while “0” stands for the worst condition, “the failed condition” or being “out of service” (FHWA, 1995). Table 1 shows the code for bridge condition rating. If any of the three elements is rated as a four or less, the bridge is categorized as structurally deficient by federal standards. The structurally deficient rating notifies bridge engineers put more attention to the bridge. Bridge condition rating is used to make important decisions regarding needs assessment and budget allocation (Dabous and Alkasss, 2010).

Table 1: Code for bridge condition rating (FHWA, 1995)

Code	Description
9	EXCELLENT CONDITION
8	VERY GOOD CONDITION - no problems noted.
7	GOOD CONDITION - some minor problems.
6	SATISFACTORY CONDITION - structural elements show some minor deterioration
5	FAIR CONDITION - all primary structural elements are sound but may have minor section loss, cracking, spalling or scour.
4	POOR CONDITION - advanced section loss, deterioration, spalling or scour.
3	SERIOUS CONDITION - loss of section, deterioration, spalling or scour have seriously affected primary structural components Local failures are possible. Fatigue cracks in steel or shear cracks in concrete may be present.
2	CRITICAL CONDITION - advanced deterioration of primary structural elements. Fatigue cracks in steel or shear cracks in concrete may be present or scour may have removed substructure support. Unless closely monitored it may be necessary to close the bridge until corrective action is taken.
1	IMMINENT FAILURE CONDITION - major deterioration or section loss present in critical structural components or obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic but corrective action may put back in light service.
0	FAILED CONDITION - out of service - beyond corrective action.

During inspection, the inspector needs to prepare all needed documents and to review the previous inspections. After the on-site visual inspection, an in-depth inspection should be conducted by the field inspector. To ensure the accuracy and assessment of the condition of the structure, the inspector should use several tools to perform the inspection.

Many sensing technologies have been developed for bridge monitoring, for example ground penetrating radar, infrared thermography, fiber optics embedded sensing, ultrasound, stress wave based methods, LiDAR, etc. (Chase, 1995, Sack and Olson, 1995, Aktan et al. 1997, Washer, 1998, Peeters and Ventura, 2003, Gucunski et al. 2006). Most of those techniques focus on either in-situ placement of sensors or sensors on mobile platforms and can be characterized as target-intended (TI) techniques. TI sensors are typically designed with an intended target in a specific monitoring area and usually consist of supervised monitoring scheme (i.e. with established periodic sampling, passive sampling or continuous monitoring designs). Nondestructive Evaluation (NDE) methods have been employed to complement visual inspection in some state DOTs, but widespread use of NDE technologies are limited.

Although many bridges are currently equipped with online cameras that stream live images of a bridge to the DOTs, most of these cameras are static (fixed in position) and do not have adequate image resolution for structural monitoring purposes. Wireless sensing units have been suggested that can cover a wide area of bridges, however, these sensors are still very much target-centric - they are embedded/mounted in a fixed location on specific bridges to measure specific parameters. Embedded sensors need to be installed accurately in a designed location (Shoukry et al. 2009; Nagayama et al. 2010, Chae et al. 2012). Wireless sensing solution may not be economically feasible for all bridges.

2.3 Bridge Management System (BMS)

(1) Current BMSs

Figure 5 shows a brief history of the bridge management system (BMS) development in the U.S. (Liu, 2010). In the United States, several BMSs have been developed for bridge management, such as Pontis and Bridgit. Currently, Pontis, Bridgit, and state-owned software are three bridge management systems used by different state DOTs. Some states still develop their own BMS software. Pontis and Bridgit are two very well-known BMSs. More than 40 states use Pontis to manage their bridges or bridge database (Jivacate and Najafi, 2003). Pontis is a BMS developed by the FHWA in conjunction with state DOTs and a joint venture consulting firm (She et al., 1999). Bridgit is an initiative carried out by the American National Research Council as a research project jointly sponsored by AASHTO and the FHWA under the National Cooperative Highway Research Program (NCHRP).

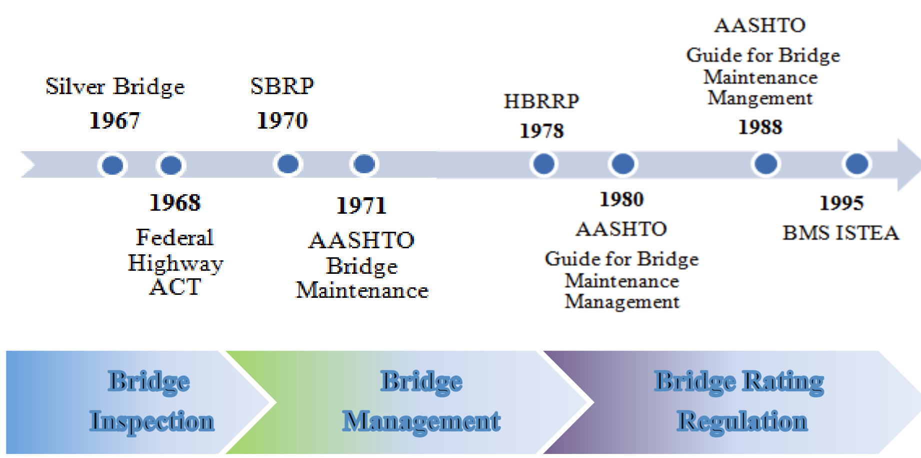


Figure 5: History of bridge management system (Liu, 2010)

Pontis supports the complete bridge management cycle, including bridge inspection and inventory data collection and analysis, recommending an optimal preservation policy, predicting needs and performance measures for bridges. Due to the widespread of the

Pontis applications and its open architecture, among the agencies in the Unites States, most of them customized the system or developed additional applications, such as providing additional data entry forms, reports or other functionalities (Robert et al., 2003).

Bridgit uses a project level-based optimization strategy to provide network-level (within a state agency) recommendations and guidance on how to allocate funds on a bridge network and optimize network performance strategy (Hawk, 1999). The advantage of Bridgit is its ability to define and distinguish between specific protections systems for components while determining feasible options (Elbehairy, 2007).

(2) Recent BMS Developments

Saito (1988) developed a network level bridge management system for the Indiana Department of Highway to manage state-owned bridges. The application included the development of criteria to determine present and future needs, analysis of user costs and impacts of bridge related activities, and the development of methods to set priority for bridge maintenance, rehabilitation, and replacement alternatives.

Chase et al. (1999) suggested several different relational database management approaches and data warehousing techniques to efficiently utilize the NBI bridge information and GIS database. Three different regression methods were applied to model the relationship between bridge condition and deterioration causing factors. She et al. (1999) developed a model to support the development of a GIS-based bridge management system by using a hybrid business and information modeling approach. This may be the earliest model of GIS-based BMS.

Karlaftis et al. (2005) developed a Web-supported national bridge inventory management tool to aid bridge engineer in accessing, retrieving, manipulating, or obtaining information from NBI database. The tool was based on the commonly used software of Microsoft Access and could import text- or spreadsheet-based data.

2.4 Internet Technology and Geographic Information System (GIS)

(1) The Internet Technology

The development of Internet technology makes it possible for data communication among the Web servers and end users (Web 2.0). The information can be shared and transferred from one place to another around the globe with users making choices for access to the geography related information. The integration of GIS and Internet further provides users the flexible accessibility to geospatial information at any time and any place. With the Web-based systems, the response database is continually being updated as the public uses the system and provides inputs (Kingston et al., 2000).

(2) The GIS

Geographic information systems (GIS) have been recognized as an important tool for geospatial data application decision support and planning analysis. GIS technology has been widely implemented in many public and private organizations for decades, because GIS has special features for the storage, retrieval, manipulation, analysis, and display of geographically referenced data (Batty and Xie, 1994). Since map can make better understanding than verbal sentence, GIS provides great capacity for applications about geospatial data.

(3) Web-GIS Applications

In recent years, GIS has begun to appear on the World Wide Web (WWW or Web) ranging from simple demonstrations and references to GIS use, to more complex online GIS and spatial decision support systems. The availability of combining GIS and Internet technologies is becoming a reality in many fields (Kingston et al., 2000; Skarlatidou et al., 2011 and Dragicevic, 2004). Users have flexible accessibility to the bridge information at any time and any place. The system also offers a high degree of flexibility to upload relevant information throughout the user participation process (Kingston et al., 2000).

Most current Web-GIS applications are focused on environmental studies with fewer applications on infrastructure monitoring and management. Sugumaran and Meyer (2004) proposed a Web-based spatial decision support system (SDSS) for environmental planning and management. Shi et al. (2005) presented development of a bridge structural health monitoring and information management system by employing GIS, database and other related technologies. Burdziej (2012) introduced the concept of spatial decision support system that combines network analysis and spatial analysis of accessibility with an interactive Web-based application. Chen et al. (2010) developed an Integrated Remote Sensing and Visualization (IRSV) bridge management system which is proposed to help bridge managers to comprehend voluminous, heterogeneous bridge data from four essential perspectives: geospatial, temporal, relational and per-bridge attributes. An interactive data exploration environment is implemented to help bridge manager evaluate the bridge based on internal factors and/or external factors (Wang et al, 2010).

2.5 Visualization

(1) Computer Visualization

Visualization is the concept that by visual association with amplification of mental processing of contingent data may result in the generation of synergistic correlation revelation. Visualization offers a link between human eyes and computer display to help people identify patterns and to extract insights from large amount of information and can help people to understand and communicate about subjects and ideas of interest. In addition, visualization has been used to communicate ideas, to monitor trends implicit in data, and to explore large volumes of data for hypothesis generation (Zhu and Chen, 2005).

To provide an observer an aggregated representation of available dataset is the goal of data visualization. In addition, another purpose of visualization is to make it capable for human's visual system to understand and explore the dataset. Computer visualization becomes an art and a science that generates images or visual representations from large quantities of dataset (Michalos et al., 2012). Visualization technologies have been applied in many nonscientific contexts, including business, digital libraries and human behavior.

(2) Visualization Techniques

Visualization techniques are used in many research areas for interpreting or presenting large datasets, in the next few years, more powerful practices will be available for implementation (Michalos et al., 2012). Over the past decades, with the development of technologies, visualization techniques have grown up and reached a quite mature level of graphical display. New techniques of data visualization appear, such as parallel coordinates, statistical distribution and flow maps (Liang et al., 2005). Another free program called Protovis, an open-source program, allows user to produce Web-native visualization. Since it is a quick, simple and powerful solution for graphical visualization, Protovis is suggested for data visualization (Michalos et al., 2012). Tatu et al. (2009) introduced scatterplot visualization technique with an application for quality measurement. A current popular JavaScript library named Data-Driven Documents (D3) is introduced for Web-based visualization applications (D3, 2012).

(3) Visualization Applications

Esch et al. (2009) introduced a graphical 3D visualization technique for highway bridge condition ratings; the system is used for loading highway bridges with rating so that engineers can determine which structural components are in need of repair; meanwhile, the related information can be displayed on a graphical user interface.

Zhang et al. (2004) indicated that research on Web-based product information visualization is very significant for supporting the collaborative design and manufacturing at the disperse sites, virtual enterprises, concurrent engineering, Internet-based design and manufacturing. Huang (2003) presented a development of novel interactive visualizations framework for environmental modeling.

2.6 Citizen Sensor

(1) The Concept of Citizen Sensor

The citizen sensor, an individual with social awareness and the capability to capture and share data, usually has the willingness and the desire to share information. The shared data can be personally or socially relevant. Social-relevant data sharing is classified by the availability of sensing tools and the geospatial-temporal position. As Goodchild (2007a) introduced, the power of "humans themselves, each equipped with some working subset of the five senses and with the intelligence to compile and interpret what they sense" is a tremendous geographical resource. However, establishing acceptable standards or metrics of credibility is the key to the effective use of humans as sensors in the specific contexts. Following specialty training, citizen sensors can provide accurate information about their local environments (Goodchild, 2007a and Sheth, 2009). On this regard, citizen sensor relies on two critical technologies: 1) Internet technology that makes the public participation possible from any place with available Internet connection and 2) GIS technology that recognizes the geospatial data of interests.

Volunteer sensors can be considered as a type of non-target-intended (NTI) sensor and represents a critical emerging paradigm to social interactivity and operations due to the empowerment of individuals to capture information, and the availability of online uploads to Internet (Nagarajan et al. 2009). A good example would be the recent Haiti Earthquake that devastated the entire nation: one can easily use online Google Maps™ to view clear shots of damaged structures that are posted and annotated by people in Haiti.

(2) Citizen Sensor Applications

Goodchild (2007b) introduced the application of citizen volunteering information application in Web services, such as Wikimapia (www.wikimapia.org), which is a service operating on similar to Wikipedia and allowing citizens to provide description of places of interest along with geographic coordinates.

Another volunteer performed open source application is OpenStreetMap (OSM, 2012), a collaborative project to create a free editable map of the world. The map is created by using data from portable GPS devices, aerial photography, and other free sources. The map display features a prominent “edit” link. Volunteers collect data by using a handheld GPS unit, a notebook, a digital camera, and a voice recorder.

Christin et al. (2011) described some “participatory” sensing applications as: 1) people-centric sensing applications such as personal health monitoring, calculating environmental impact and exposure to dust particles, monitoring and documenting sport experiences, enhancing social media, and auditing price; 2) environmental-centric sensing applications such as air quality monitoring, noise and ambiance monitoring, thermal column monitoring, and road traffic monitoring, etc.

(3) Mobile Application Development

With the development of technologies, mobile phone becomes a device capable of computing, sensing, and communication. The advances in mobile phone technology coupled with volunteers make it feasible to accomplish large-scale sensing (Burke et al., 2006). The key idea behind citizen sensing is to empower ordinary citizens to collect and share sensed data from their surrounding environments using their mobile phones. Mobile phones, especially smartphone, are functioned as sophisticated sensors. The cameras can be used as video and image sensors. The microphone can be used as an acoustic sensor, and the embedded GPS receivers can provide location information (Christin et al., 2011).

The increasingly comfortable adopting mobile phones impacts our daily lives from many aspects (Estrin, 2010). Citizen sensing can benefit from individuals using mobile phones and cloud services to collect and report data. First, citizen sensor involves very low development cost since the sensing device (mobile phone) and the communication infrastructure (cellular network or WiFi) are already available. Secondly, the volunteer can provide unprecedented spatiotemporal coverage – sensor mobility makes it possible to observe unpredictable events (Christin et al., 2011).

(4) Development Tools

Table 2 shows several software development kits (SDKs) and integrated development environments (IDE's) for mobile device development that are available for mobile application development. Singh and Palmieri (2012) presented the comparison of cross-platform mobile development tools. Popular development languages include C++, C, Java, JavaScript, CSS, HTML, Objective C, Visual Basic, and C#. For current research, Basic4Adroid development platform is selected.

Table 2: Different development tools for mobile application (Mobile, 2012)

Platform	Programming language	Cross-platform deployment	Development tool cost
Airplay SDK (now Marmalade)	C, C++	All native: Android, BlackBerry, BREW, iOS (iPhone), Maemo, Palm/webOS, Samsung bada, Symbian, Windows Mobile 6.x and desktop, OSX	Commercial licenses available
AppFurnace	JavaScript	Android and iOS	Free to use, test and demo. Costs to publish an app.
Application Craft	JavaScript, HTML5, CSS	All platforms: Android, iOS, Blackberry, Windows Mobile, Bada, WebOS, Symbian. Mobile, Desktop and Tablet	Free to use a limited version with advertisements in the application
Appception	HTML5, CSS, JavaScript	Android, iOS	Free and Commercial licenses available
Aqua	C, C++, JavaScript	Android, BlackBerry Playbook, iOS, Palm/webOS, Samsung bada, Windows Mobile 6.x, Windows Desktop	Free & commercial licenses available
Basic4android	Visual Basic	Android	Commercial licenses available
Celsius	Java	Java ME, Android, BlackBerry, iPhone, Symbian, Windows Mobile	Commercial licenses available
CloudPact	HTML, CSS, JavaScript	Android, BlackBerry, iPhone, Windows Mobile	Free limited trial and Enterprise plans available
DragonFireSDK	C, C++	iPhone, iPad, iPod Touch	Commercial licenses available
FeedHenry	HTML, CSS, JavaScript	Apple iPhone & iPad, Android, Windows Phone 7, Blackberry, Nokia WRT.	Free, Professional and Enterprise Plans available
iOS SDK	Objective-C	iPhone, iPad, iPod Touch	Apple tools are free for an Intel-based Mac. Simulator testing is free, but installing on a device needs a fee.
LocalClick Partners	Graphical Drag and Drop, HTML, CSS	iOS, Android, HTML5 Web Apps, HTML5 Hybrid Apps	Free & commercial licenses available. White-label & turnkey mobile app business opportunity

2.7 Remote Sensing Technology

Remote sensing technology is defined as the measurement of object properties on the earth’s surface by using data acquired from optical devices. Remote-sensing systems provide repeatable and consistent views of the surface of the objects. Thus, remote sensing has many applications such as environment assessment, environmental monitoring, and agriculture monitoring (Schowengerdt, 2010). For the past fifty years, several commercial remote sensing (CRS) and spatial information (SI) technologies, wide-bandwidth spectral information sensing and imaging, have been developed integrally with satellite/airborne/ground-based surveillance platforms.

Compared to satellite imagery, airborne sensors have the potential of providing images with higher resolutions. InSiteful imagery (2011) provides aerial photography with a high resolution, which can be made faster than most orthophotography. Stone et al. (2004) discussed that several terminologies have been applied to similar technologies including LiDAR (Light Detection and Ranging) and Ladar (Laser Radar).

Short-distance 3D LiDAR scanners can generate dense point clouds of position information that can be used to establish baseline geometric information for bridges and to establish critical dimensional footprints for pre-event and post-event comparisons. For bridge monitoring, Liu (2010) developed a LiDAR based Bridge Evaluation (LiBE) algorithm that utilize LiDAR data for defect detection, clearance measurement, and displacement measurement. Tong et al. (2010) presented 3D LiDAR technologies studies on newly constructed bridges and for bridges that have experienced close range blasting. For close range blast events, the pre-blast and post-blast scans of a bridge can be used to establish blasting induced effects and damage information.

Navalgund et al. (2007) described several remote sensing techniques applications, such as in sustainable agriculture, ocean color and fishery, environmental assessment and monitoring, disaster monitoring and mitigation, weather and climate studies, and community-centric applications including village resource center and tsunami GIS for relief and rehabilitation.

2.8 Summary

In this chapter, reviews of the previous work on NBI and BMS have been presented. The related emerging information technology (IT) and sensing technologies are also discussed. The literature review revealed that the emerging techniques can be assuredly integrated into bridge health monitoring to provide a better solution. To enhance the national bridge system management, the integration of Internet, GIS, visualization, remote sensing, and citizen sensing techniques will makes a better strategy to enhance the current bridge management system. By applying the technologies discussed above, the proposed Bridge-WGI is developed and can enhance capabilities of the existing bridge management system.

CHAPTER 3: SYSTEM CONCEPT AND ARCHITECHTURE

3.1 Introduction

This chapter describes the development of the Web-GIS based interactive information visualization framework (Bridge-WGI) for bridge data management. Bridge-WGI is intended to provide bridge managers bridge data visualization with accessibility to the Internet at any time and at any place. Various features and capabilities for the bridge management system are included in this framework. The system concept and architecture of Bridge-WGI are presented in the following sections.

3.2 System Concept

3.2.1 Concept Model

The NBIS consists of a database of more than 600,000 bridges that constitutes the US highway bridge infrastructure (FHWA, 2011). These bridges are discretely distributed across the national territory. Network level monitoring requires broad spatial sensing techniques and geospatially enable management systems. The system should be capable of integrating both TI and NTI sensor data. Citizen sensor can be one type of NTI sensor for massive data collection. Figure 6 shows a proposed bridge manage monitoring scheme that collects multiple format sensor data.

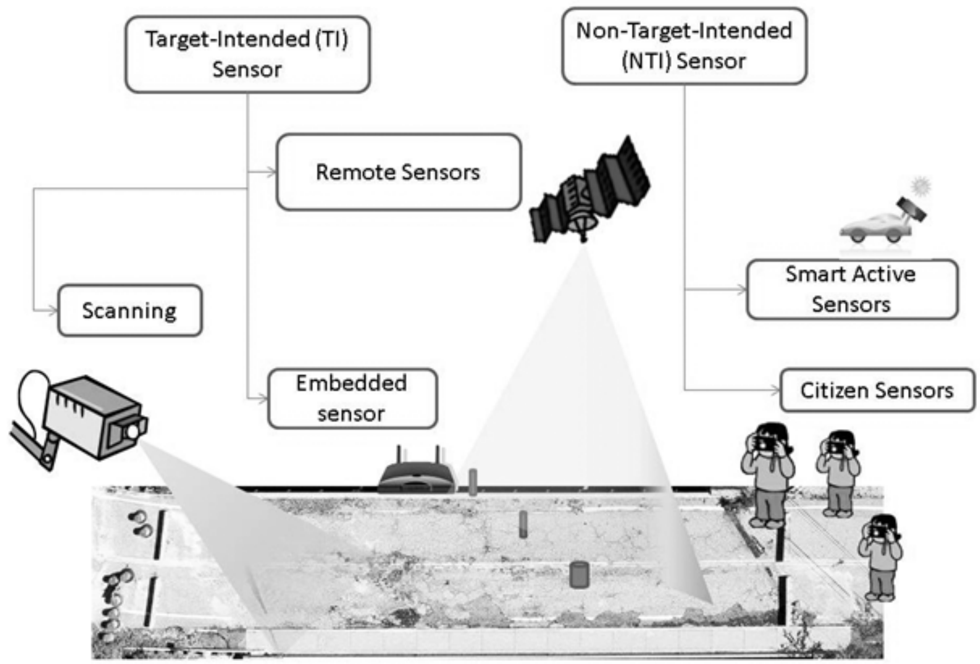


Figure 6: Bridge monitoring using multi-format sensing approaches including citizen sensors

The concept model for Bridge-WGI for bridge inspection and management is shown in Figure 7. By using TI and/or NTI sensing techniques, including the volunteering sensing, the sensing data can be stored into the Web server or data cloud via wired Internet connection or wireless connection. Citizen sensors can collect data and report data via mobile device such as smartphone or tablet. With the flexibility of Internet access, bridge manager or other professionals can use the desktop computer to view the bridge data and conduct related analysis with Bridge-WGI. A system administrator maintains the entire system, installing, supporting and maintaining servers, database administration (MySQL), or other computer systems.

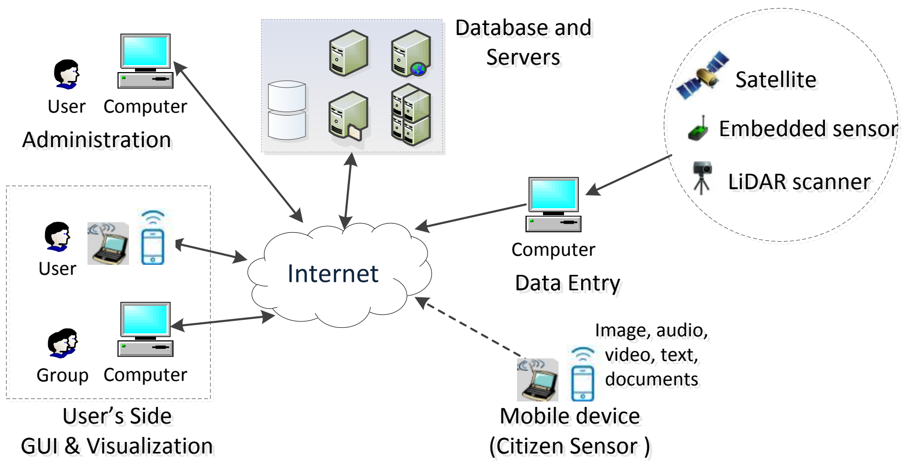


Figure 7: The diagram of concept of bridge inspection and management system

The proposed system should use database technologies such as MySQL to support the bridge data storage and management. Google Map serves as a map server to support the geospatial information mapping. The proposed system includes following capabilities:

a) Display the general information of the bridge condition rating;

b) User collaborative interactions;

c) Interactive data visualization for bridge information;

d) Data entry and document uploading;

e) Citizen sensing data applications; and

f) User-defined criteria query for decision-making support.

3.2.2 Bridge-WGI Framework

The developed system is called Bridge-WGI and its framework consists of four key parts: data collection, data upload, database management, and information visualization (Figure 8). Several enabling technology components: a) citizen sensing data entry and visualization; b) bridge information visualization and analytical visualization; c) user-defined criteria query for decision-making support; and d) remote sensing data entry and visualization, have been developed and integrated.

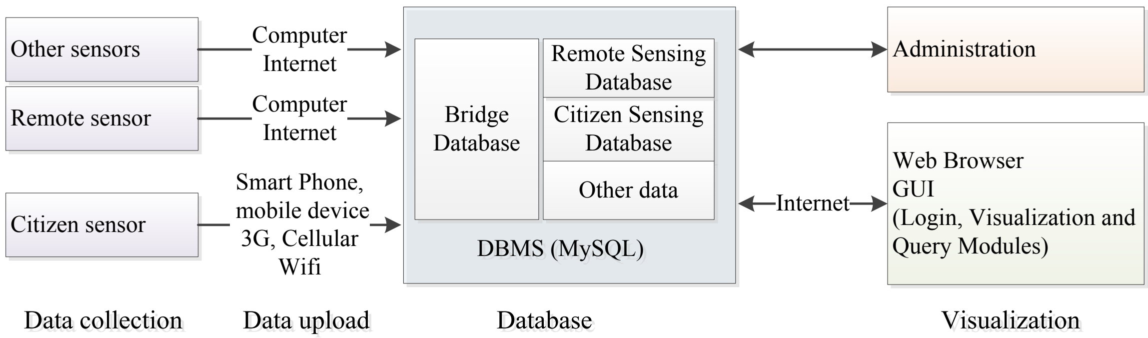


Figure 8: Flowchart for Bridge-WGI system

The database management system (DBMS) is a complex, mission-critical software system (Hellerstein et al., 2007). The DBMS stores and maintains the bridge data which are captured by various sensors. Since several sensing technologies are involved in monitoring the bridge and collecting data, there are many data formats. Bridge-WGI provides the portal for data input and is flexible enough to allow data entry regardless the type of data. Before uploading to database, the raw data should be processed and converted to standard formats consistent with the database.

Bridge-WGI framework includes several analytical functionalities, including general information visualization, interactive analytical visualization, user-defined criteria query, citizen sensor application, and remote sensing data application. Developed with targeted users, Bridge-WGI adopts a client-server system model and is described in the following section.

3.3 Architecture of Bridge-WGI

3.3.1 Client/Server Model for Bridge-WGI

The client-server model for Bridge-WGI is illustrated in Figure 9. The main structure includes two parts: client/user side applications and server side applications. The server side applications provide the resources for the client side application, which includes database management system (DBMS), map server, file system, and Web server. The database system includes three databases: bridge database, remote sensing database, and citizen sensing database. The uploaded volunteering sensing documents are stored in a special file location.

The client side application relies heavily on the visualization design. When a volunteer uploads an image on a computer (client), the computer will send request to the Web server, the image will be uploaded to Web server and stored in the file system by using HTML, PHP and JavaScript techniques. The related information of the image will be appended to the citizen sensing database. When the bridge manager or other professionals view the uploaded information by volunteer sensors from a computer (client), the related information can be retrieved from Web server and displayed by HTML, PHP and JavaScript techniques.

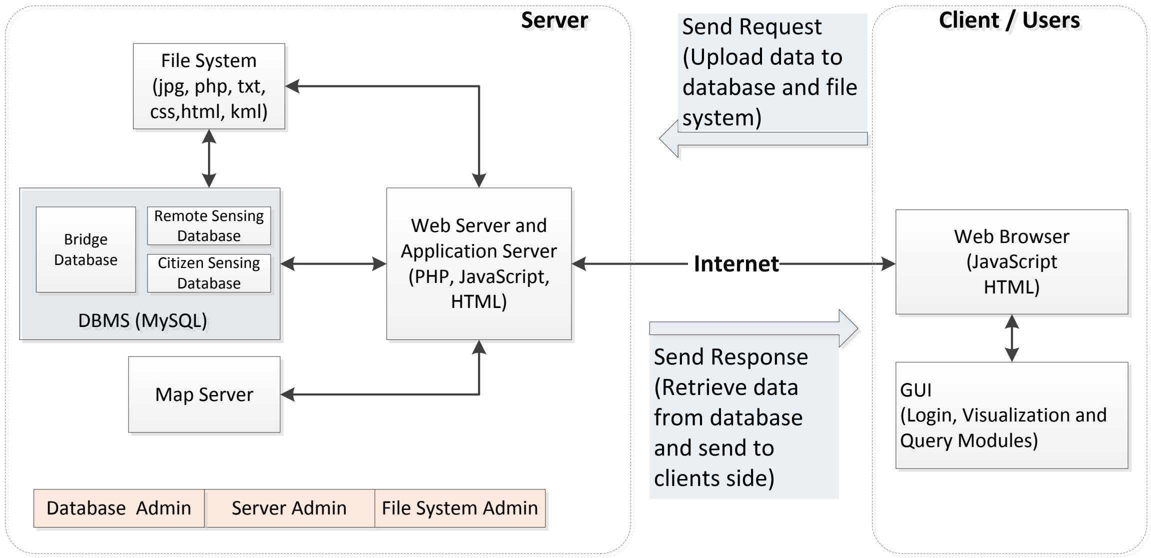


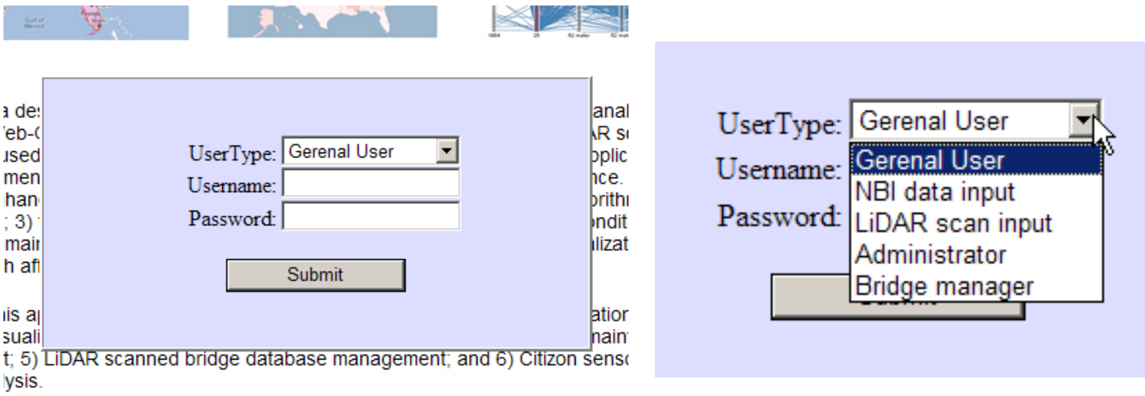
Figure 9: Client-server model for Bridge-WGI

3.3.2 Client / User Side

The clients can be desktops, mobile PCs or mobile electronic products with access to the Internet. There are two parts of this framework, they are user login module and graphic user interface (GUI). The user login module is served as security checking and the main GUI provides the main functionalities of this framework.

3.3.2.1 Login

For security, five-level privileges are designed for Bridge-WGI, including 1) general user (lowest level) who has privilege to view the information of national bridge system and input public data; 2) the bridge inspector has privilege to input bridge inspection data; 3) the bridge manager has privilege to upload the LiDAR scanned data; 4) the administrator can maintain the data which input by other users; and 5) special privilege for bridge manager to conduct the user-defined criteria query (UDCQ) for decision-making support. Only for the general user (citizen sensor) who do not need login, all the other users must enter the user name and password to access the application. Figure 10 shows the login interface.



a) Login interface　　　　b) User-type selection

Figure 10: Login interface

3.3.2.2 Graphic User Interface (GUI)

In Bridge-WGI, the system considers different perspectives and clarity for users. The main functionalities of this application include 1) Homepage; 2) Bridge information display; 3) Bridge condition inspection data entry; 4) LiDAR scan data entry; 5) LiDAR scan and related information display; 6) Citizen sensing data display; 7) Citizen sensing data entry; 8) Bridge data analytical visualization; and 9) UDCQ for decision-making support. Figure 11 illustrates the functionalities of Bridge-WGI. The screenshot of the homepage (http://irsv-test.uncc.edu/ytong1/testnew/) of Bridge-WGI is shown in Figure 12.

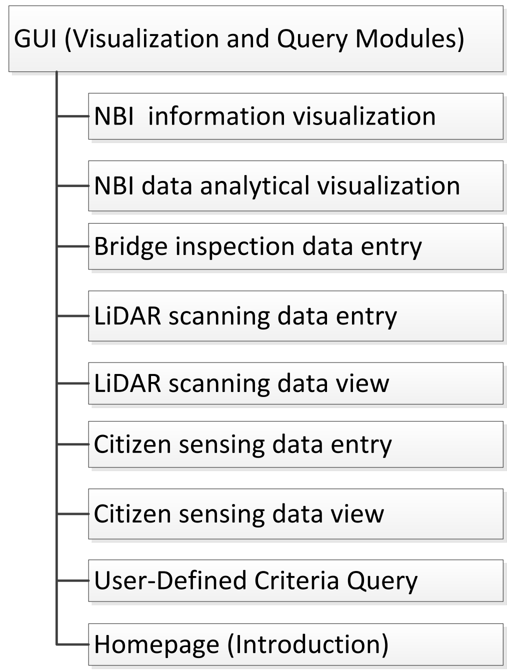


Figure 11: The main functionalities of Bridge-WGI

3.3.3 Server Side Application

In Bridge-WGI, the server side includes database management system (DBMS), map server, file system, Web server and application servers, and UDCQ for bridge maintenance decision-making support.

3.3.3.1 Database Management System (DBMS)

This database management system is implemented on the server side using relational database management system. The relational DBMSs are associated by relation tables that store the bridge data. The database management system consists of bridge database (text and numeric data); LiDAR scan database (text, numeric and imagery data), aerial fly-over image bridge data (numeric, text and imagery data), and citizen sensing database (text and imagery data) (Detail see Section 3.5).

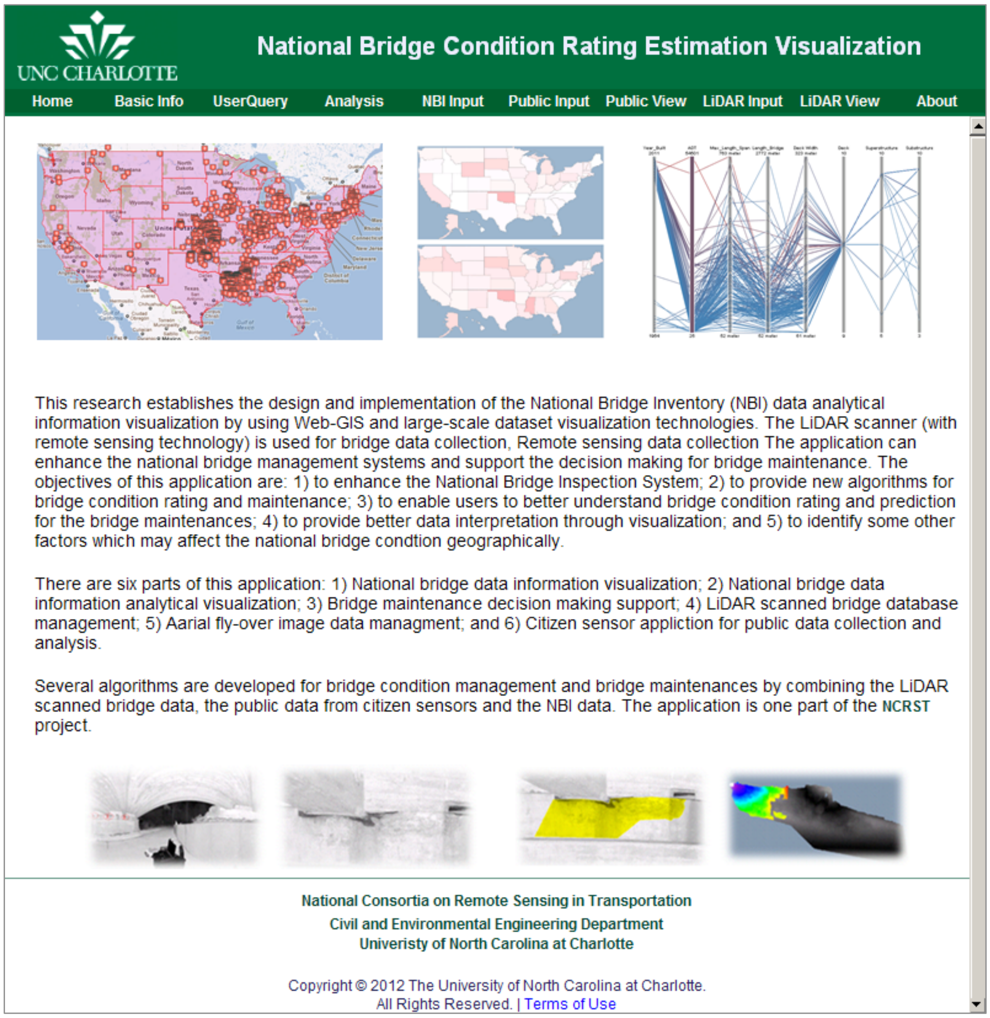


Figure 12: Screenshot of homepage of Bridge-WGI

(Testing website: http://irsv-test.uncc.edu/ytong1/testnew/)

3.3.3.2 Web Server

A Web server can provide a primary function to deliver webpages and the request to clients. A server can deliver HTML documents and other additional contents such as images, cascading style sheets (CSS), and scripts. Web server provides two types of data: static data and dynamic data. The static data is normally stored at an offsite server, while the dynamic data are constructed by programs that execute at the time a request is made. A high-performance Web server can typically deliver several hundred static files per second (Iyengar and Challenger, 1997).

A map server offers a Web map service (WMS), which is a standard protocol for serving geo-referenced map images over the Internet. These map images are generated by a map server using data from a GIS database (OGC, 2012). The specification was developed and first published by the Open Geospatial Consortium in 1999 (Scharl and Tochtermann, 2007). Map server is a development environment for building spatial information (SI) over Internet. Some open source software servers include GeoServer, MapServer, and geozilla.

Bridge-WGI mainly applies Google Map API to support Web map service (GoogleMap, 2012). By using JavaScript programming language to embed Google Map API in the scripts of webpages, the geospatial mapping can be implemented (GoogleMap, 2011).

The framework Bridge-WGI uses MySQL database technology to support bridge data. Google Map service serves as a map server to support the geospatial information. Programs are coded in PHP, JavaScript, and HTML.

3.3.3.3 File System

File system maintains all the files used for this system, such as the webpages (.html, .php, and .css file), geospatial document such as .kml file and some other uploaded bridge data such as images, report documents (Microsoft Word, Excel document, pdf file), LiDAR scanned images, and other document such as compressed file (i.e. zip file). These files reside in specific folders in the server.

3.3.3.4 User-Defined Criteria Query (UDCQ)

To meet the special needs for data analysis, a user-defined criteria query (UDCQ) is developed in Bridge-WGI. UDCQ can provide a mechanism for extending the functionalities the system supports. User-defined query can efficiently support complex database applications, and can provide significant efficiency (Chaudhuri and Shim, 1999).

3.4 System Configuration and Development

The system architecture is built with scripting language on both server side and client side. Google Map API supports the map service on the client side. On the server side, MySQL database is used to manage the database; an Apache server with HTML, PHP and JavaScript, is used for system communication, developing additional system functionality through internal scripting, and establishing dynamic links with external software and data sources. This configuration allows a participant who has access to an Internet connection and a graphic Web browser to interactively and remotely explore Web-GIS system.

3.4.1 System Configuration

In this system, the server hardware is a Dell PowerEdge R210 with 16GB of RAM and 2 x 1TB hard drives. The server hosts two virtual machines named irsv.uncc.edu (production) and irsv-test.uncc.edu (testing). Both virtual machines are configured with 7GB of RAM and 900GB of local storage, which is split into a system drive and a data drive. Each guest runs on a separate physical hard drive.

The virtual machines are operating through the RedHat Enterprise Linux 5.6 which runs the following software:

- Tomcat application server. Tomcat Apache Web server v2.2.3;
- BlazeDS application server with the turnkey configuration;
- MySQL database server v5.0.77;

h The Linux operating system files and configuration is managed by scripting and an automated installation. The manual steps to configure VMware and bootstrap Linux are documented by the IT group.

3.4.2 Web-GIS Platform

Several Web-mapping services are introduced for Web-GIS applications, such as Google Map, Google Earth and Microsoft Virtual Earth. Google Map API is chosen for current development. Google Maps is a Web mapping service application and technology which is provided by Google, and is powered by many map-based services, including the Google Maps website and maps embedded on third-party websites via the Google Maps API (GoogleMap, 2012).

Interactive geospatial visualization has been developed as integral part of the Web-GIS platform and will be presented in Chapter 4.

3.5 Database System Design

3.5.1 Introduction

The database system, which plays an important role in information storing and sharing, is the basis for Bridge-WGI. A well-designed database should have capability to be expanded in the future. Privileges of sharing data among state-, city- and county-level also need to be considered. The database system in this design mainly consists of three parts (Figure 13), including the bridge data database (text and numeric data), remote sensing data (LiDAR scan and fly-over aerial image, text, numeric and imagery data) and the citizen sensing data (text and imagery data). Since the NBI data is in ASCII format, the data need to be converted to a format for MySQL database usage. Some programs for the data conversions need to be developed by using programming languages, such as C++ or Java.

Remote sensing bridge data, including LiDAR scan and fly-over aerial images, are integrated to expand the bridge database. The database system can also include capability to share data with other BMSs.

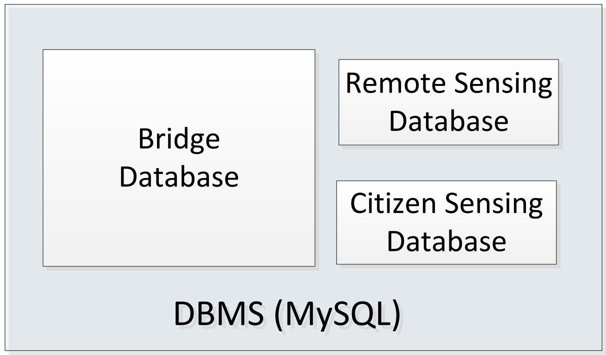


Figure 13: Database systems for Bridge-WGI

3.5.2 Database Structures

In the database, for each bridge, there are more than 100 information fields, including basic information such as structure type, structure length, year built, design load, inspection date, year reconstructed, kind of material, and maintenance responsibility, geospatial information such as latitude and longitude, bridge condition rating information such as deck, superstructure and substructure, etc. The FHWA data structure is based on ASCII format, the detail NBI record format is showed in Appendix B.

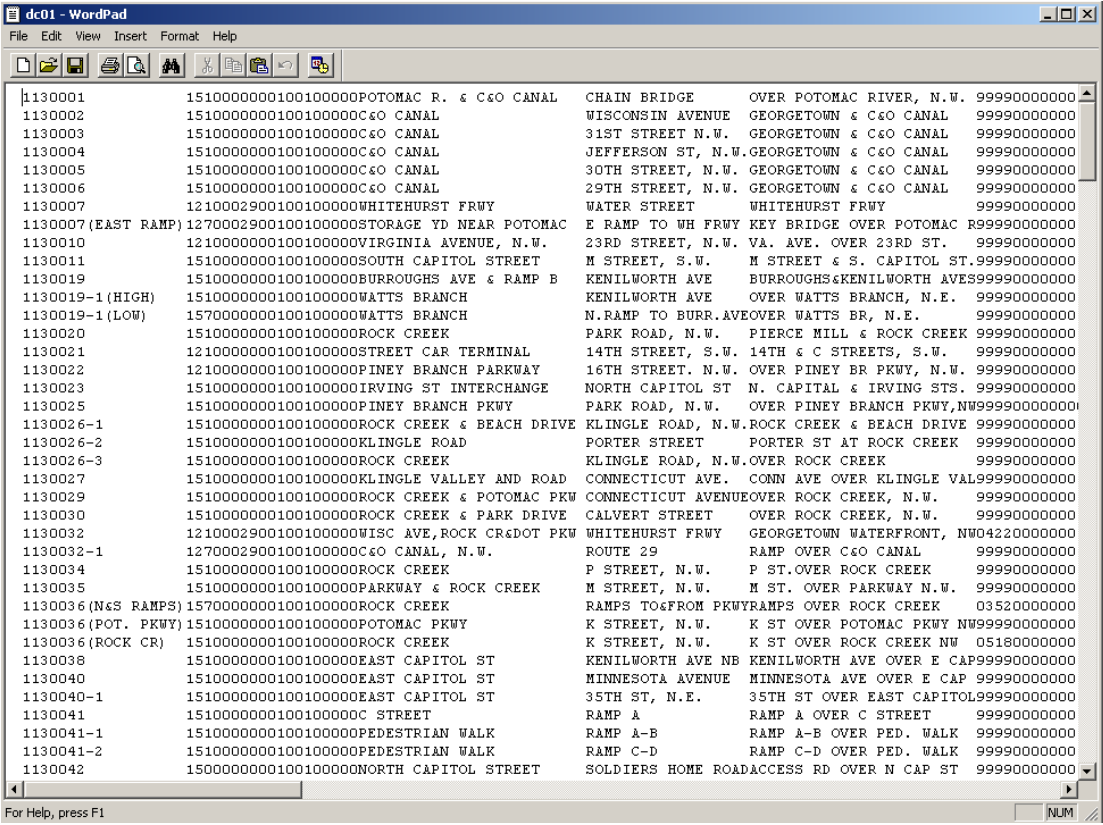
Remote sensing database includes bridge spatial locations (latitude and longitude), bridge ID, city or county of the bridge located, test date, number of uploaded image, numbers of upload documents, and some other information of the LiDAR scan. The citizen sensing database stores the information uploaded by citizen sensor, including bridge ID, upload date, reported information, damage rank and some other related information.

The MySQL database includes eight tables in this database which are “nbi2008”, “nbi2009”, “nbi2010”, “nbi2011”, “remote”, “citizen”, “login”, and “nbiinput”. Among these tables, “nbi2008”, “nbi2009”, “nbi2010” and “nbi2011”, store the NBI bridge database for the year 2008, 2009, 2010 and 2011, respectively. Table “remote” and “citizen” are used for remote sensing data and citizen sensing data. Table “login” stores the username and password for security issues. The “nbiinput” table stores the bridge inspection data of bridge conditions. The detailed structures of these tables are shown in Table 3-C1 through Table 6-C4 in the Appendix C.

3.5.3 Data Collection

3.5.3.1 Data Source

The NBI data is available at USDOT website (http://www.fhwa.dot.gove/bridge/nbi/) for downloading. The data record is available from 1992 to date for all 50 states and Washington DC. The NBI data file is an ASCII format document, each record of bridge has 432 characters to store the information, and the format is shown in Appendix B. A typical NBI ASCII file is shown in Figure 14.



```
dc01 - WordPad
File  Edit  View  Insert  Format  Help

1130001            1510000000100100000POTOMAC R. & C&O CANAL   CHAIN BRIDGE      OVER POTOMAC RIVER, N.W. 99990000000
1130002            1510000000100100000C&O CANAL                WISCONSIN AVENUE  GEORGETOWN & C&O CANAL   99990000000
1130003            1510000000100100000C&O CANAL                31ST STREET N.W.  GEORGETOWN & C&O CANAL   99990000000
1130004            1510000000100100000C&O CANAL                JEFFERSON ST, N.W.GEORGETOWN & C&O CANAL   99990000000
1130005            1510000000100100000C&O CANAL                30TH STREET, N.W. GEORGETOWN & C&O CANAL   99990000000
1130006            1510000000100100000C&O CANAL                29TH STREET, N.W. GEORGETOWN & C&O CANAL   99990000000
1130007            1210002900100100000WHITEHURST FRWY          WATER STREET      WHITEHURST FRWY          99990000000
1130007(EAST RAMP) 1270002900100100000STORAGE YD NEAR POTOMAC  E RAMP TO WH FRWY KEY BRIDGE OVER POTOMAC R99990000000
1130010            1210000000100100000VIRGINIA AVENUE, N.W.    23RD STREET, N.W. VA. AVE. OVER 23RD ST.   99990000000
1130011            1510000000100100000SOUTH CAPITOL STREET     M STREET, S.W.    M STREET & S. CAPITOL ST.99990000000
1130019            1510000000100100000BURROUGHS AVE & RAMP B   KENILWORTH AVE    BURROUGHS&KENILWORTH AVES99990000000
1130019-1(HIGH)    1510000000100100000WATTS BRANCH             KENILWORTH AVE    OVER WATTS BRANCH, N.E.  99990000000
1130019-1(LOW)     1570000000100100000WATTS BRANCH             N.RAMP TO BURR.AVEOVER WATTS BR, N.E.      99990000000
1130020            1510000000100100000ROCK CREEK               PARK ROAD, N.W.   PIERCE MILL & ROCK CREEK 99990000000
1130021            1210000000100100000STREET CAR TERMINAL      14TH STREET, S.W. 14TH & C STREETS, S.W.   99990000000
1130022            1210000000100100000PINEY BRANCH PARKWAY     16TH STREET. N.W. OVER PINEY BR PKWY, N.W. 99990000000
1130023            1510000000100100000IRVING ST INTERCHANGE    NORTH CAPITOL ST  N. CAPITAL & IRVING STS. 99990000000
1130025            1510000000100100000PINEY BRANCH PKWY        PARK ROAD, N.W.   OVER PINEY BRANCH PKWY,NW99990000000
1130026-1          1510000000100100000ROCK CREEK & BEACH DRIVE KLINGLE ROAD, N.W.ROCK CREEK & BEACH DRIVE 99990000000
1130026-2          1510000000100100000KLINGLE ROAD             PORTER STREET     PORTER ST AT ROCK CREEK  99990000000
1130026-3          1510000000100100000ROCK CREEK               KLINGLE ROAD, N.W.OVER ROCK CREEK          99990000000
1130027            1510000000100100000KLINGLE VALLEY AND ROAD  CONNECTICUT AVE.  CONN AVE OVER KLINGLE VAL99990000000
1130029            1510000000100100000ROCK CREEK & POTOMAC PKW CONNECTICUT AVENUEOVER ROCK CREEK, N.W.    99990000000
1130030            1510000000100100000ROCK CREEK & PARK DRIVE  CALVERT STREET    OVER ROCK CREEK, N.W.    99990000000
1130032            1210002900100100000WISC AVE,ROCK CR&DOT PKW WHITEHURST FRWY   GEORGETOWN WATERFRONT, NW04220000000
1130032-1          1270002900100100000C&O CANAL, N.W.          ROUTE 29          RAMP OVER C&O CANAL      99990000000
1130034            1510000000100100000ROCK CREEK               P STREET, N.W.    P ST.OVER ROCK CREEK     99990000000
1130035            1510000000100100000PARKWAY & ROCK CREEK     M STREET, N.W.    M ST. OVER PARKWAY N.W.  99990000000
1130036(N&S RAMPS) 1570000000100100000ROCK CREEK               RAMPS TO&FROM PKWYRAMPS OVER ROCK CREEK    03520000000
1130036(POT. PKWY) 1510000000100100000POTOMAC PKWY             K STREET, N.W.    K ST OVER POTOMAC PKWY NW99990000000
1130036(ROCK CR)   1510000000100100000ROCK CREEK               K STREET, N.W.    K ST OVER ROCK CREEK NW  05180000000
1130038            1510000000100100000EAST CAPITOL ST          KENILWORTH AVE NB KENILWORTH AVE OVER E CAP99990000000
1130040            1510000000100100000EAST CAPITOL ST          MINNESOTA AVENUE  MINNESOTA AVE OVER E CAP 99990000000
1130040-1          1510000000100100000EAST CAPITOL ST          35TH ST, N.E.     35TH ST OVER EAST CAPITOL99990000000
1130041            1510000000100100000C STREET                 RAMP A            RAMP A OVER C STREET     99990000000
1130041-1          1510000000100100000PEDESTRIAN WALK          RAMP A-B          RAMP A-B OVER PED. WALK  99990000000
1130041-2          1510000000100100000PEDESTRIAN WALK          RAMP C-D          RAMP C-D OVER PED. WALK  99990000000
1130042            1500000000100100000NORTH CAPITOL STREET     SOLDIERS HOME ROADACCESS RD OVER N CAP ST  99990000000

For Help, press F1                                                                                    NUM
```

Figure 14: Screenshot for NBI data file with ASCII format

3.5.3.2 NBI-ASCII Format Data Conversion

Since NBI data is based on ASCII format that couldn't be used directly by MySQL database system, further process should be conducted for converting the ASCII format to the suitable format for MySQL application. An application, coded by C++ and named "nbiTran", is developed for data format conversion. By using nbiTran, an ASCII-format file can be converted to a comma separated values (CSV) format file for further MySQL application. PHP is used to upload the CSV format data to MySQL database via the Internet. Figure 15 illustrates the process for the data format conversion. The algorithm for data conversion from ASCII-format to CSV-format is shown in Figure 16.

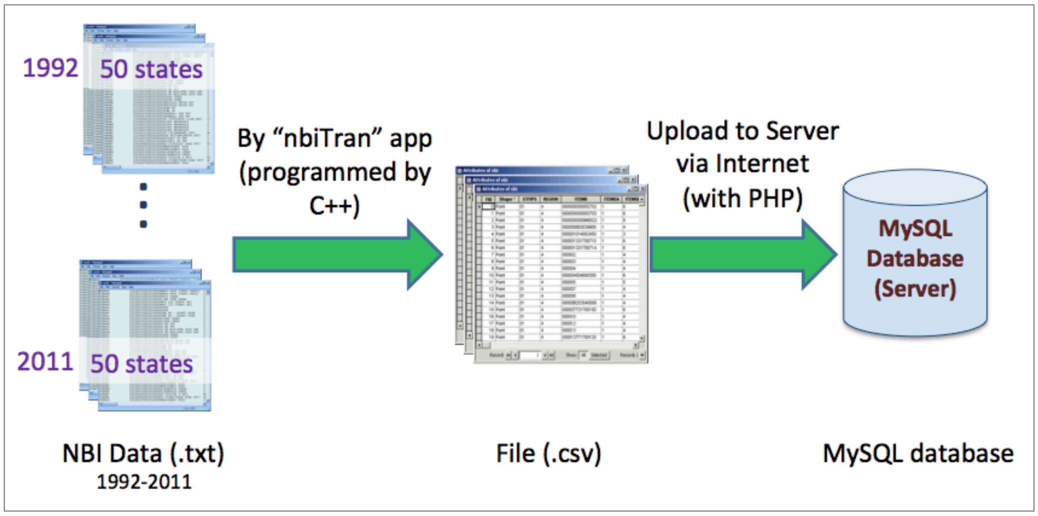


Figure 15: Flowchart for data processing

3.5.3.3 Data Uploading to MySQL Server

PHP is a widely-used open source general-purpose scripting language for Web development (PHP, 2012). PHP is embedded in HTML to make website more dynamic and responsive to user interaction. In this design, a PHP script is used for bridge data uploading to MySQL database system. The PHP script allows user to upload files from the client side browser to Web server.

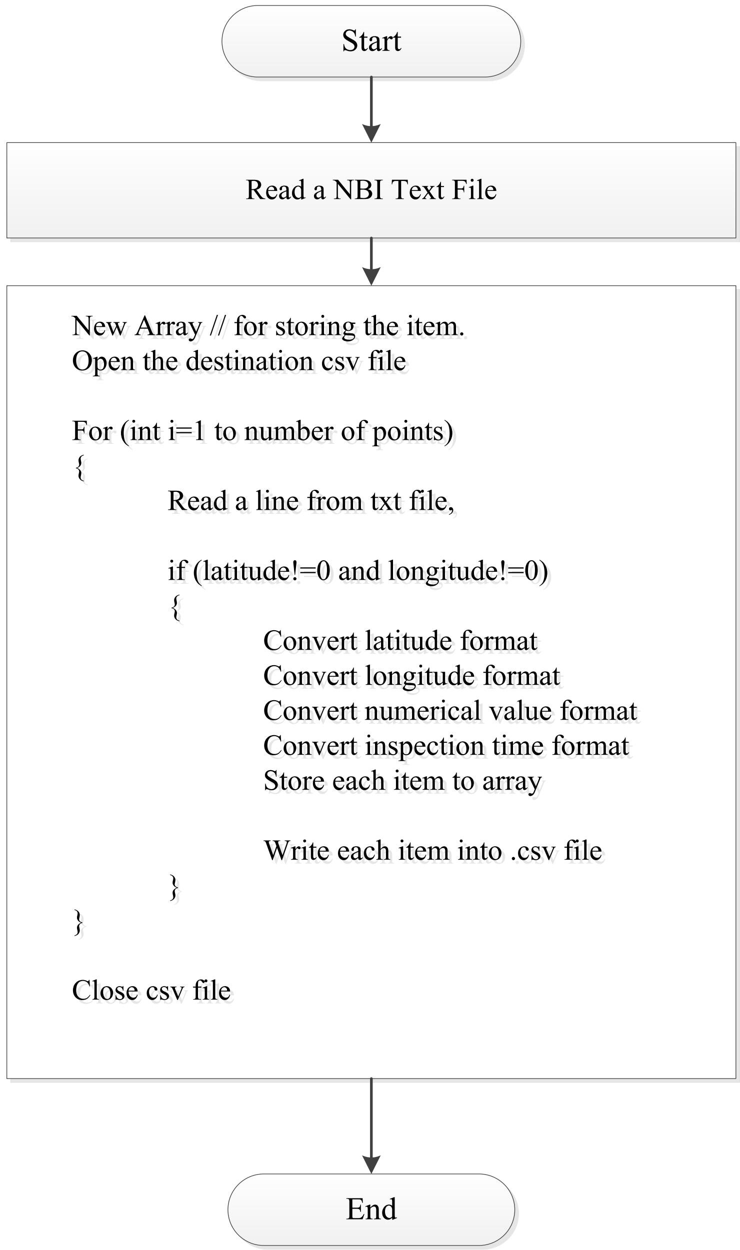


Figure 16: Algorithm for data conversion from text file format to CSV file format

3.6 Summary

The main functionalities and the operation for Bridge-WGI are summarized in Figure 17. There are five modules in Bridge-WBI: 1) general bridge information visualization; 2) analytical visualization for bridge data; 3) user-defined criteria query for decision-making support; 4) citizen sensing data entry and data view and 5) remote sensing data entry and data view. In module 1, bridge manager can search a group of bridges under a specific condition rating of deck, superstructure and substructure to view the bridge distribution, or a bridge searched by the geospatial location or the bridge ID. The general information of the bridge can be viewed (Chapter 4). In module 2, analytical visualization techniques, including map view, parallel coordinates and scatter plot, are applied for bridge manager to do further analysis (Chapter 4). For module 3, the user-defined criteria query application, helps the bridge manager or other professionals to do advanced analysis base on their own knowledge. The procedure includes generating query by using the interface, submitting query and viewing the result (Chapter 4). Module 4, the volunteer sensing data entry and data view applications, supports the volunteers to report bridge damage. Two methods can be applied for damage reporting, one is Web-based application via Internet and the other is smartphone application wirelessly (Chapter 5). Module 5 includes the remote sensing data entry and data view applications (Chapter 6).

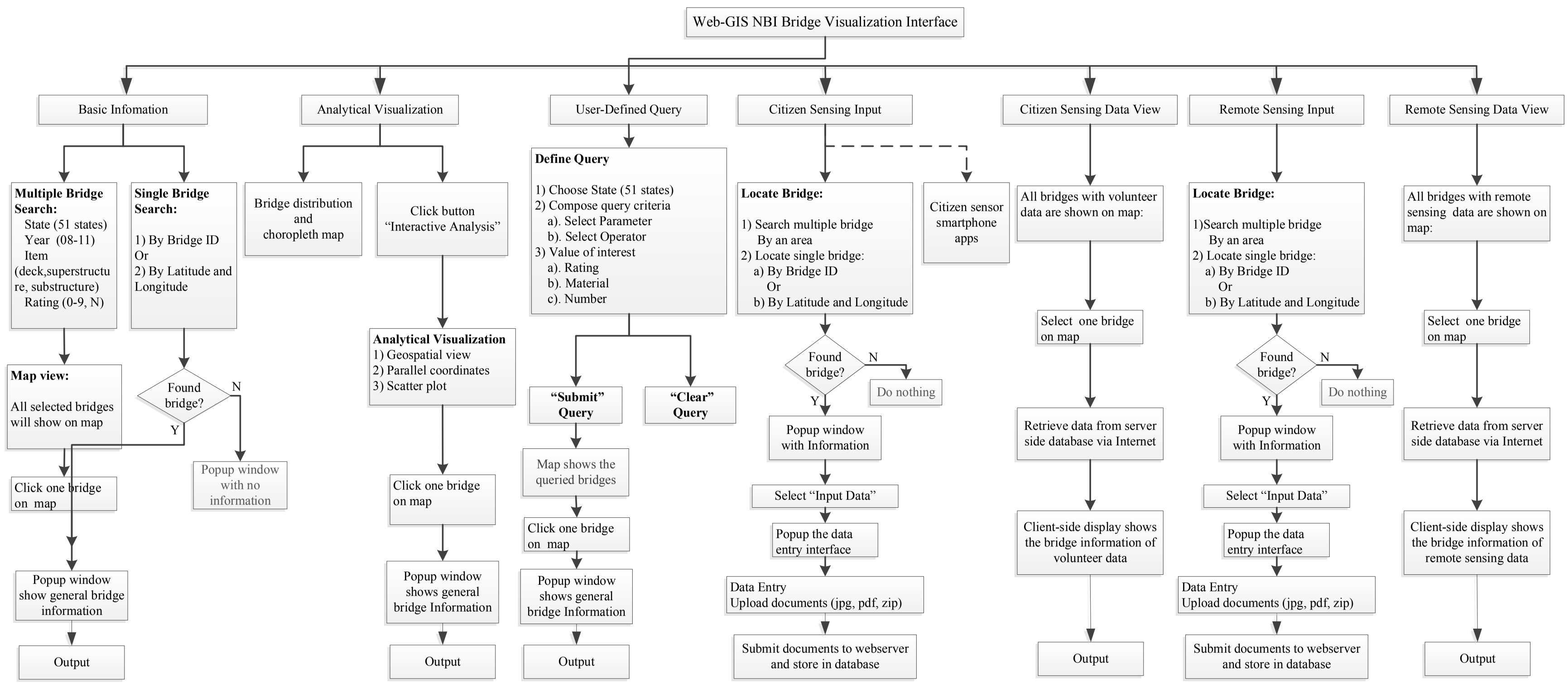


Figure 17: Main functionalities of the visualization framework for Bridge-WGI

CHAPTER 4: BRIDGE INFORMATION VISUALIZATION

4.1 Introduction

In this chapter, the applications of information visualization technologies for NBI bridge database are discussed. The visualization component within Bridge-WGI framework is designed for visualizing the general information of bridge database, such as the information of an individual bridge, the bridge condition in a selected state, or the entire continental U.S. territory including Alaska and Hawaii.

Bridge-WGI enables manager to view information of selected bridges from bridge database including bridge condition rating for deck, superstructure and substructures. In addition, Bridge-WGI includes advanced analytical visualization for the bridge data including parallel-coordinate views. Figure 18 shows the main user interface that the Bridge-WGI displays basic geospatial information allowing user interaction such as selecting a state, year inspected, and bridge item for a bridge. Base map from existing interactive Web-GIS systems such as Google Map, Bing Map or Microsoft Virtual Earth can be adopted within Bridge-WGI. All user interactions such as zoom in/out, street view, satellite view, or map view can be used.

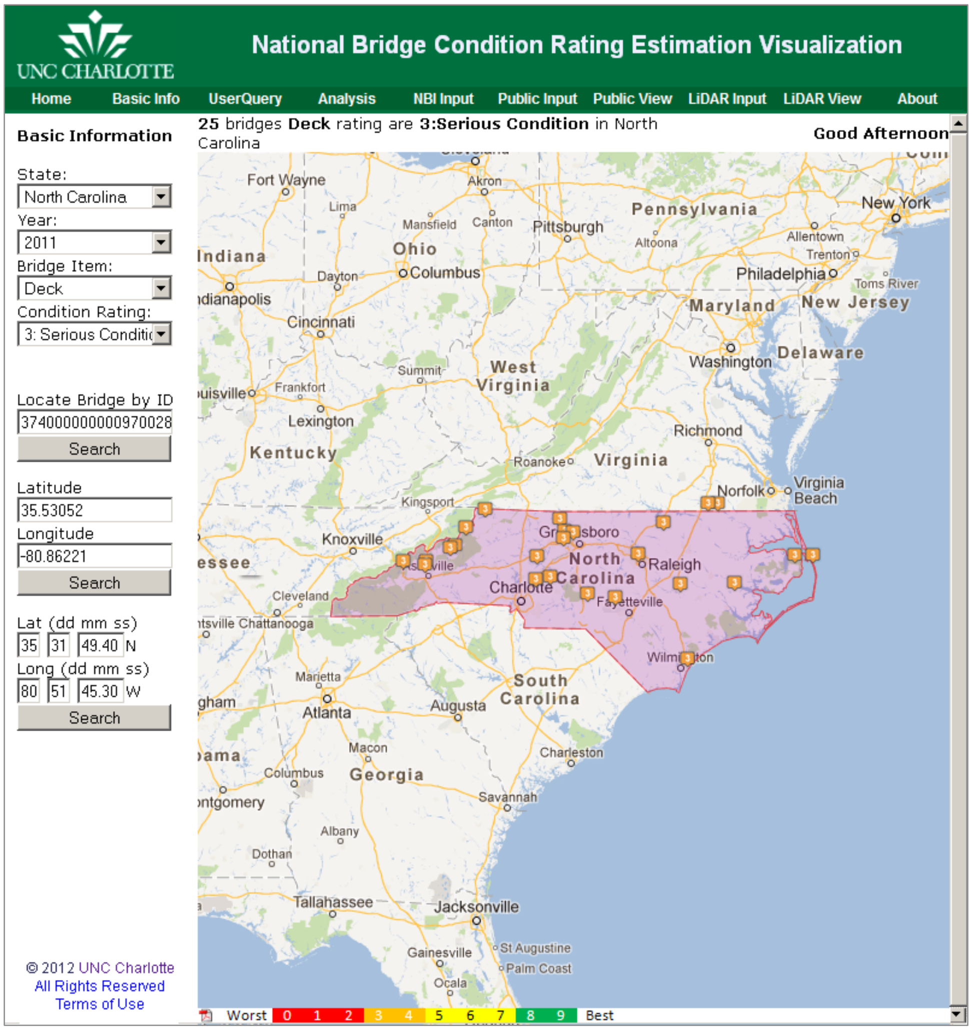


Figure 18: The graphic user interface of bridge information visualization

4.2 Visualization for NBI Bridge Information

The main functionality is to visualize the bridge information in a state level or an individual bridge. The database includes the bridge data from year 2008 to 2011. Users can search bridge by selecting 1) the condition rating of deck, superstructure or substructure; 2) the state; and 3) the year. Figure 19 illustrates the operation for bridge data visualization.

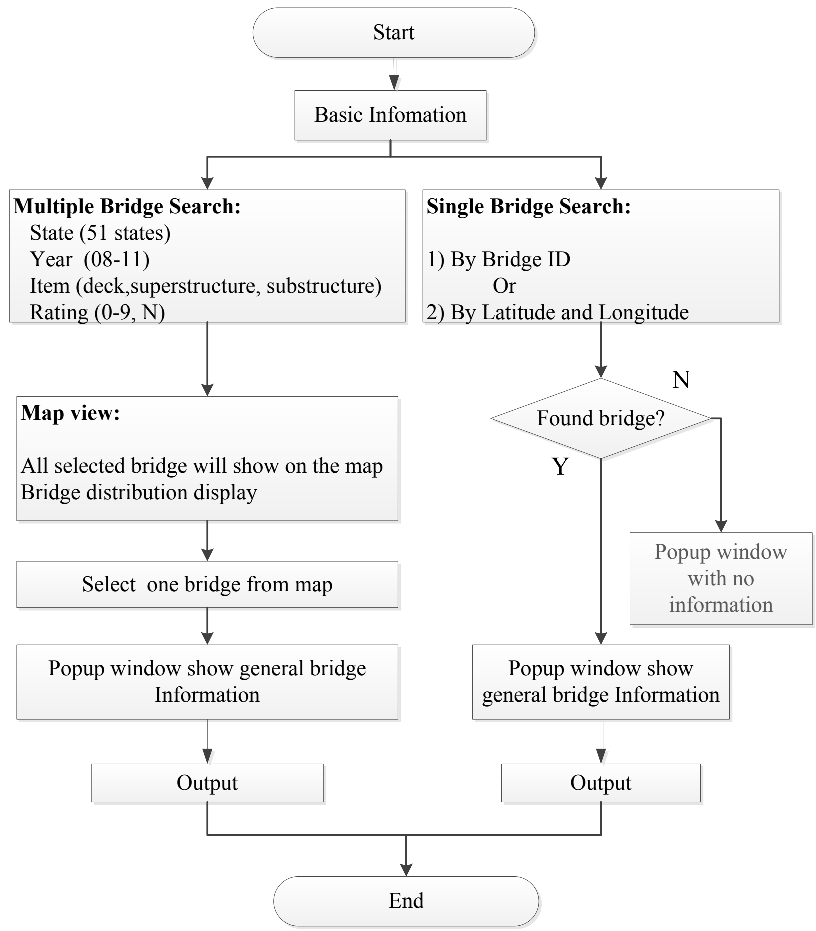
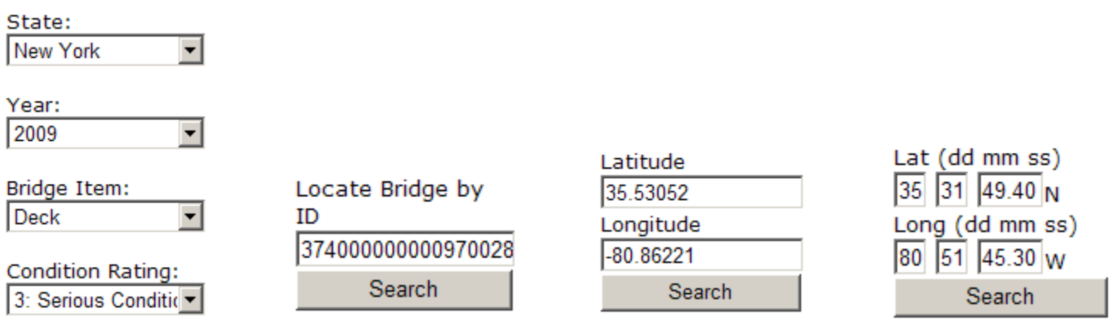


Figure 19: Flowchart for bridge information visualization operation

The bridge information can be viewed from different perspectives, such as allowing user interactivity including selecting the state, year, bridge item for bridge condition display (Figure 18). Moreover, the application allows both state-level and individual bridge information visualization, therefore, this module has two functionalities for the bridge information (Figure 19).

One visualization application is multiple-bridge selection in state-level based on state, year, and condition rating of deck, superstructure and substructure. By using selection tools (Figure 21a), users can choose a bridge element from deck, superstructure or substructure, state, and year to display the bridge under the same condition, Figure 18 shows all the bridges with deck rating of 3 in North Carolina and in the NBI report in year 2011. The bridge icon is color coded based on the condition rating, red for “0” through “2”, orange for “3” and “4”, yellow for “5” through “7”, green for “8” and “9”. Figure 20 illustrates the color scale for the bridge condition rating. By federal standard, if any of the deck, superstructure, and substructure is rated as four or less, the bridge is categorized as structurally deficient.

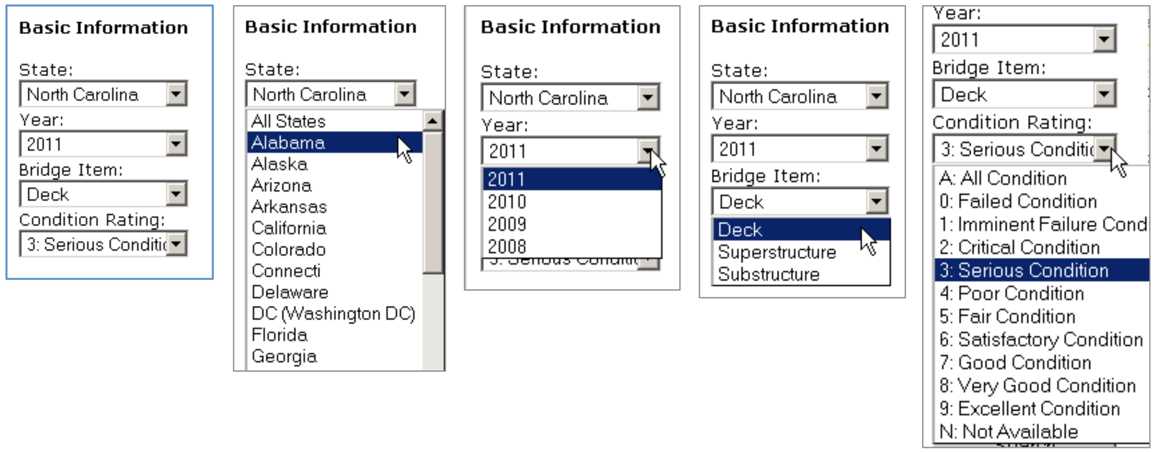
Figure 20: Color scale for bridge condition rating from 0 through 9



a) b) c) d)

Figure 21: Four bridge selection modes

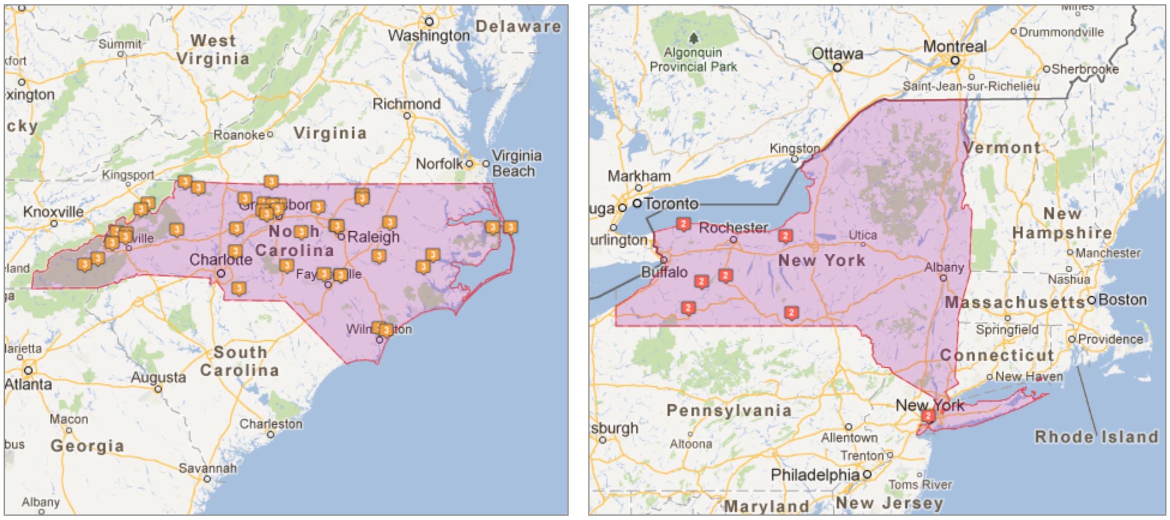
Figure 21 shows the selection modes for bridge information visualization. Figure 22 shows the detail options for multiple bridge selection, including b) state (50 states and Washington D.C.), c) year (from 2008 through 2011), d) bridge item (deck, superstructure and substructure), and e) condition rating (“0” through “9”).



(a) (b) (c) (d) (e)

Figure 22: The options for bridge selection

Based on the state-level multiple bridge-searching, Bridge-WGI allows manager to view the “big picture” of bridge condition distribution at national level or bridges within an individual state. Bridge manager can compare bridge conditions among the states, such as the statistic distribution among the states and the geospatial distribution within the state. The combination of bridge condition rating can be chosen from deck, superstructure and substructure. Figure 23a shows bridges with deck rating of 3 (or in critical condition) in North Carolina in year 2010. Figure 23b shows all the bridges with superstructure condition rating of 2 (or in critical condition) in New York State in 2009.



a) Deck condition rating with 3 (Serious condition) in North Carolina state in 2010.

b) Deck condition rating with 2 (Critical condition) in New York state in 2009

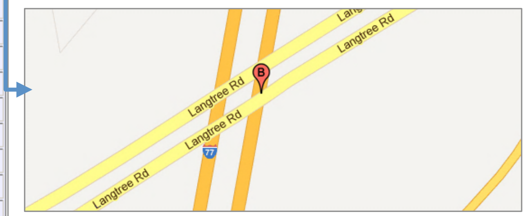
Figure 23: The bridge information displays of North Carolina and New York

The other module is the visualization for an individual bridge (Figure 24). The information includes a map, general information of the bridge, and the bridge condition rating. The basic information of the bridge includes bridge ID, latitude, longitude, year built, year reconstruction, lanes on structure, and average daily traffic (ADT). The map can be shown in three views including street view (Figure 24a), satellite image view (Figure 24b) and map view (Figure 24c). In the bridge condition rating part, the condition rating of deck, superstructure and substructure are shown (Figure 24a). If the bridge is structurally deficient, the condition rating is four or less, the related the bridge element will be highlighted (Figure 25a). In the lower-right corner, there is a printer icon for printing this page. In the bridge information page, there is a pdf icon for generate a PDF document of bridge information for further usage (Figure 25b).

Bridge ID:	374000000000970028
Latitude:	35.530528
Longitude:	-80.862217
Year Built:	2009
Year Reconstructed:	No
Lanes On Structure:	4
Lanes Under Structure:	4
Average Daily Traffic:	4200
Year Of Average Daily Traffic:	2007
Maintenance Responsibility:	State Highway Agency
Kind of Material/Design:	Steel continuous
Type of Design/Construction:	Stringer/Multi-beam or Girder
Number Of Spans In Main Unit:	2
Length Of Maximum Span:	46 meter
Structure Length:	91.7 meter
Deck Width, Out-To-Out:	27.6 meter
Operating Rating:	99.9 metric tons
Inventory Rating:	82.8 metric tons
Structural Evaluation:	9: Superior to present desirable criteria
Bridge Condition Rating (Inspected in 2010)	
Deck:	9:Excellent Condition
Superstructure:	9:Excellent Condition
Substructure:	9:Excellent Condition

a) Specific bridge information

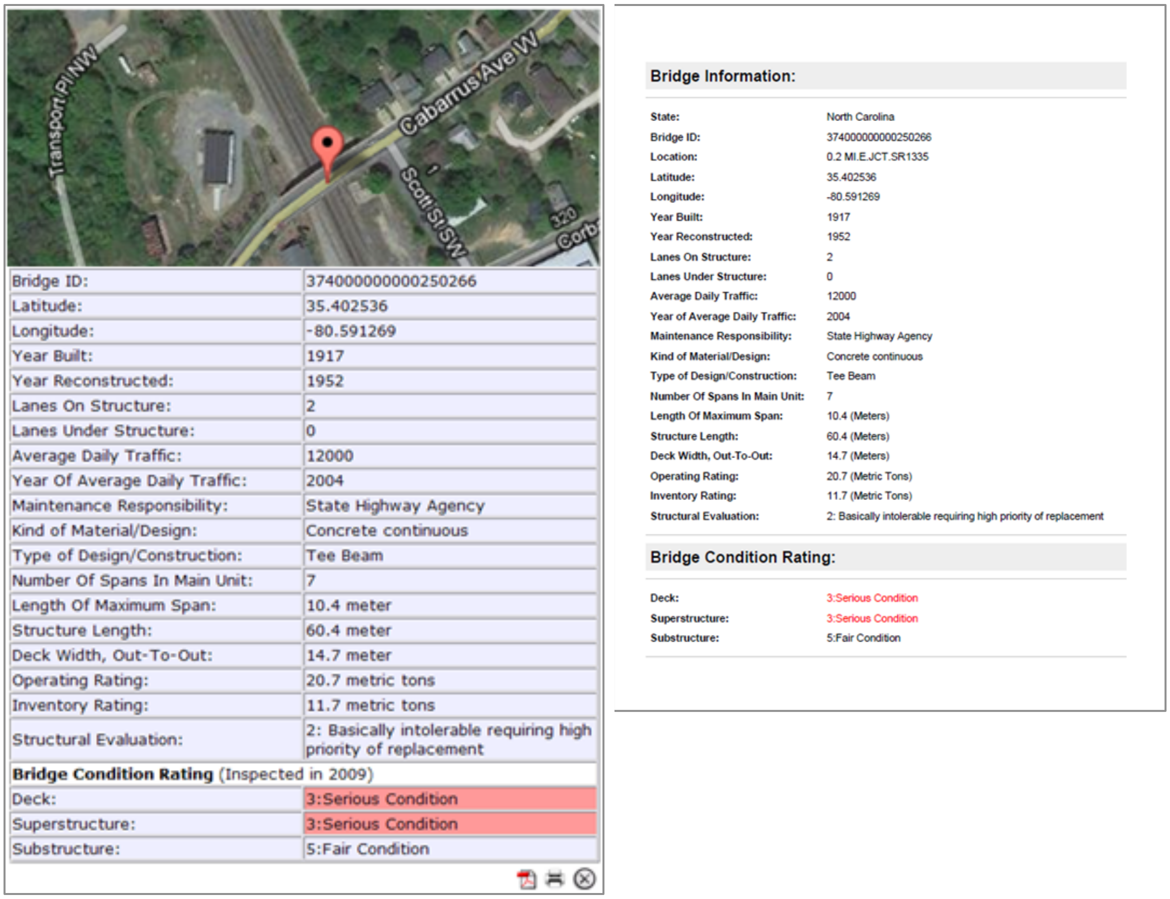
b) Satellite view



c) Map view

Figure 24: Bridge information visualization (NC bridge# 970028).

a) Specific bridge information; b) The satellite view; and c) The map view.



(a) Bridge information (NC Bridge# 250266) (b) PDF file generation

Figure 25: Red background showing condition rating with "4: Poor condition" or less

To make the program user-friendly, the bridge selection tool (Figure 21) is designed for searching a bridge by using geospatial coordinates or other approaches: 1) choose from the map, click a bridge icon to open a new page which can display the information (Figure 24); 2) by bridge ID, in this case, only the identical bridge will be selected (Figure 21b); 3) by using latitude and longitude in decimal degree format (Figure 21c); 4) by using latitude and longitude in degree, minute and second format (Figure 21d). For locating the bridge by latitude and longitude (Figure 21c and 21d), considering a possible difference between the geospatial coordinates which user entered and the geospatial coordinates of the bridge stored in the database, variable of distances can be used.

4.3 Analytical Visualization

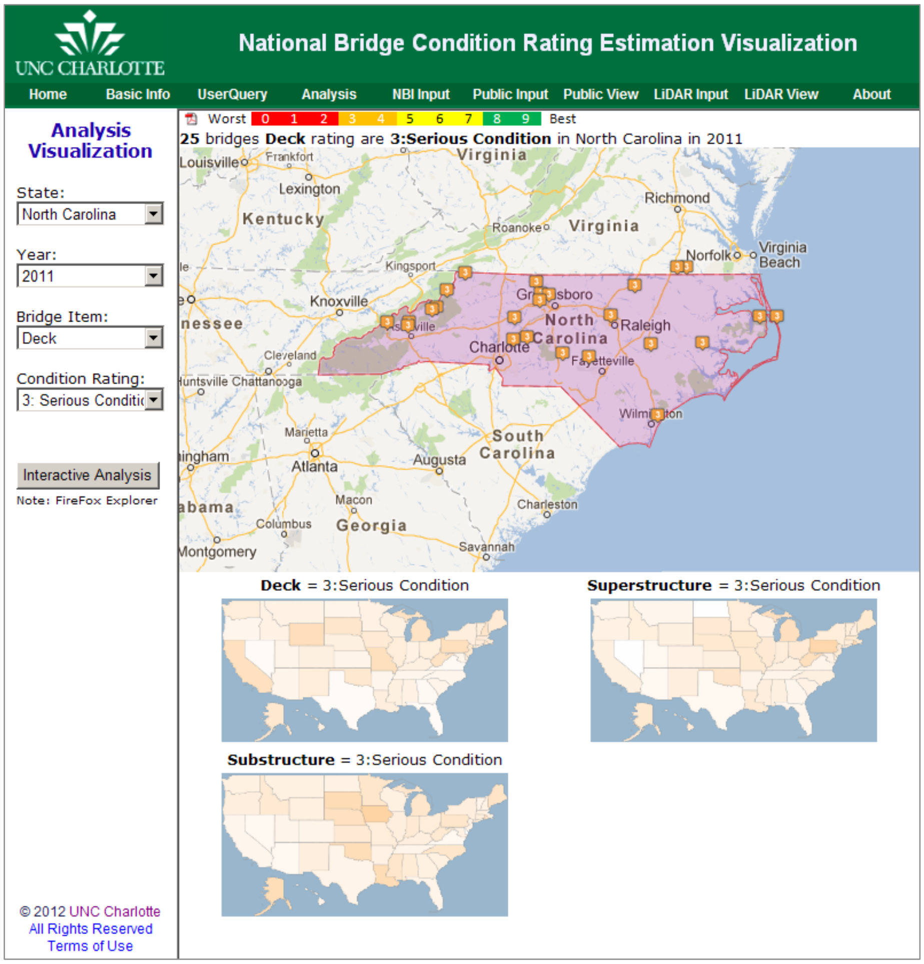


Figure 26: The graphic user interface of information analytical visualization

In Bridge-WGI, the main interface for analytical visualization is shown in Figure 26 and the operation for analytical visualization is illustrated in Figure 27. Analytical visualization can help discover more aspects of the bridge condition which may not be easily figured out from views such as static tabulation. Based on the domain knowledge characterization (bridge management), Wang et al. (2010) identified three critical analytic mode that can be useful in a bridge management application: dynamic geospatial analysis, high dimensional structural analysis and scalable temporal analysis. Following

Wang et al. (2010), the adopted analytical visualization components within Bridge-WGI includes geospatial display, scatter plot and parallel coordinate visualization of bridge data.

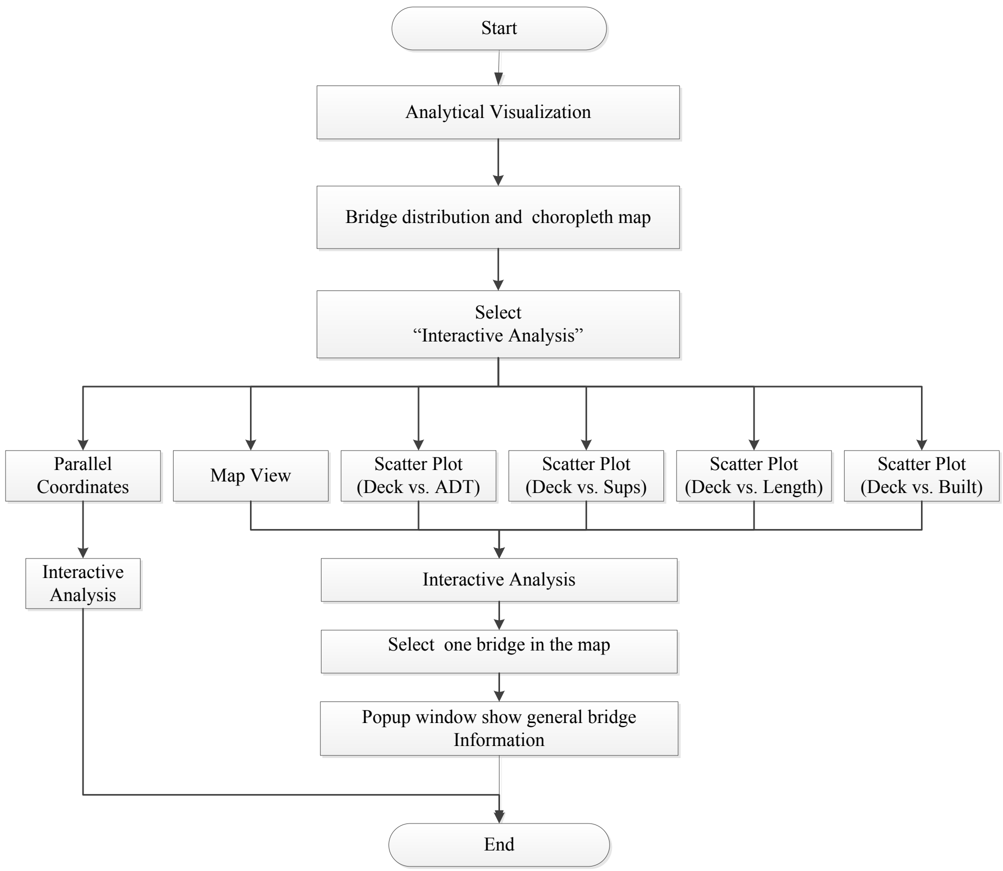


Figure 27: Flowchart for analytical visualization operation

4.3.1 Bridge Condition Rating Based on State Distribution Visualization

Figure 28 shows an example of the national bridge distribution for deck rating with 0 (failed condition). From the map, we can see there are 915 bridges with deck in failed condition nationwide. Most of these bridges are located in states including Kansas (KS), Oklahoma (OK), Louisiana (LA) and Mississippi (MS), while some other states the bridges are in better condition, fox example: North Carolina (NC), Colorado (CO), Maine (ME), Nevada (NV), North Dakota (ND), South Dakota (SD), and Virginia (VA).

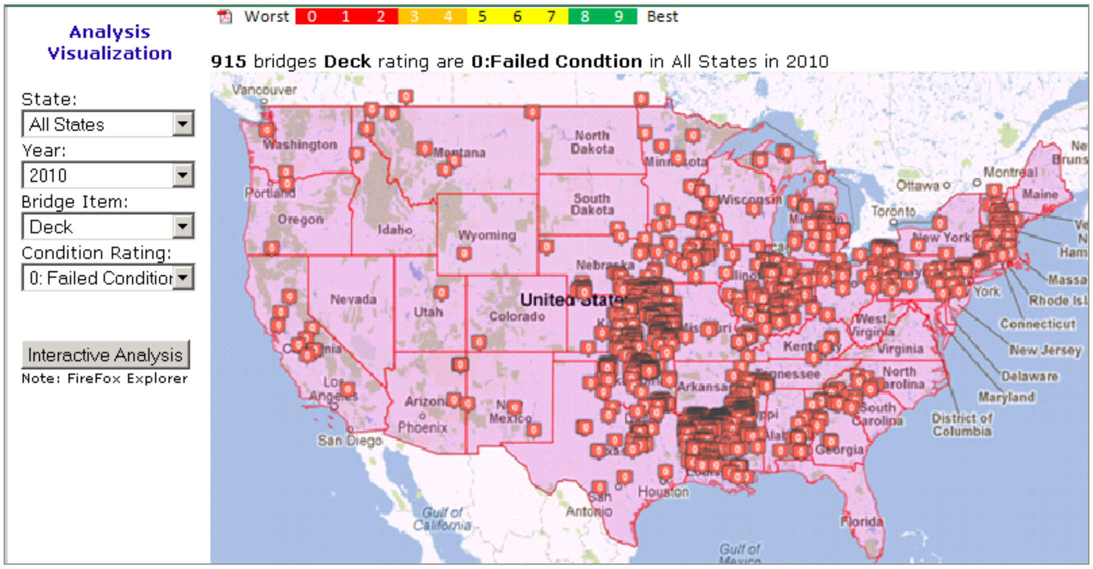


Figure 28: The distribution of bridges with deck condition rating of zero (Failed condition) nationwide in 2010

4.3.2 Parallel Coordinates Interactive Visualization

A multiple axis parallel coordinate (MAPC) view provides high level viewpoints that actually reduce the abstract dimensions of bridge parameters allowing bridge managers to detect and identify causal relationships and trends in data variables. Bridge managers can use their domain expertise to select and simultaneously compare several bridge parameters, physically visualize possible correlations between different bridge attributes and establish mental correlations. Figure 29 shows an interactive view of several bridge parameters. Each straight line linkage between the parallel coordinates represents a single bridge.

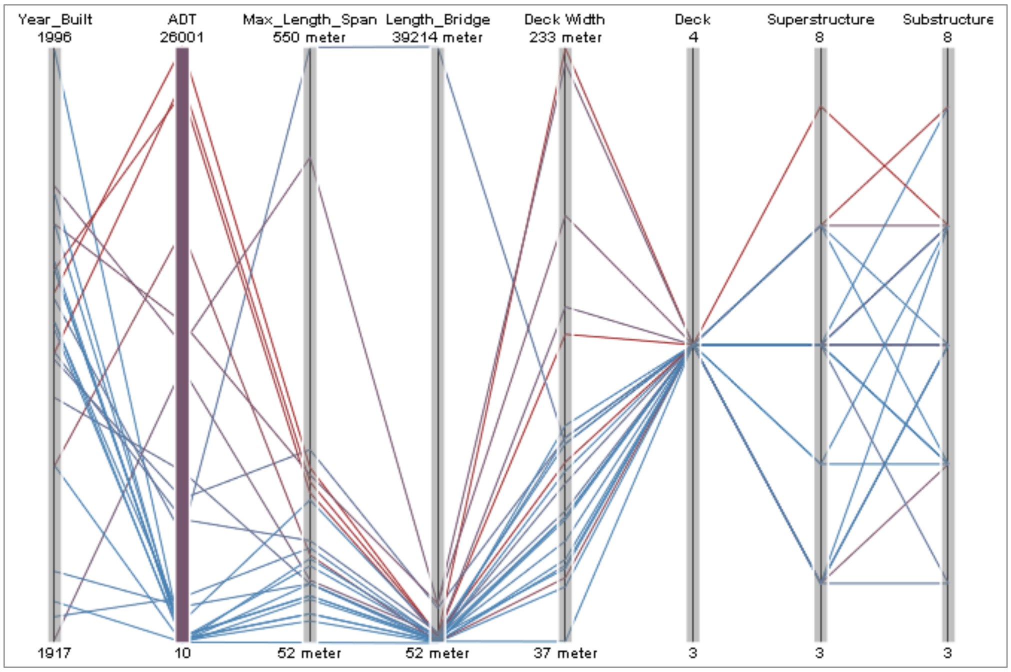


Figure 29: The parallel coordinates visualization for selected bridges (all ADT value from 10 to 26001)

Figure 29 shows the parallel visualization of selected bridges which deck condition rating 3 (serious condition) and an ADT from 10 to 26001 in North Carolina in 2010. If higher ADT is selected, such as ADT from 22362 to 26001, the result will change as shown in Figure 30. This parallel visualization presents that most bridges with higher traffic volume (22362 to 26001 ADT) among selected bridges were built between 1950 and 1970. All these bridges have maximum span length between 170 and 200 meter. No newly-built bridges have high traffic volume within the selection.

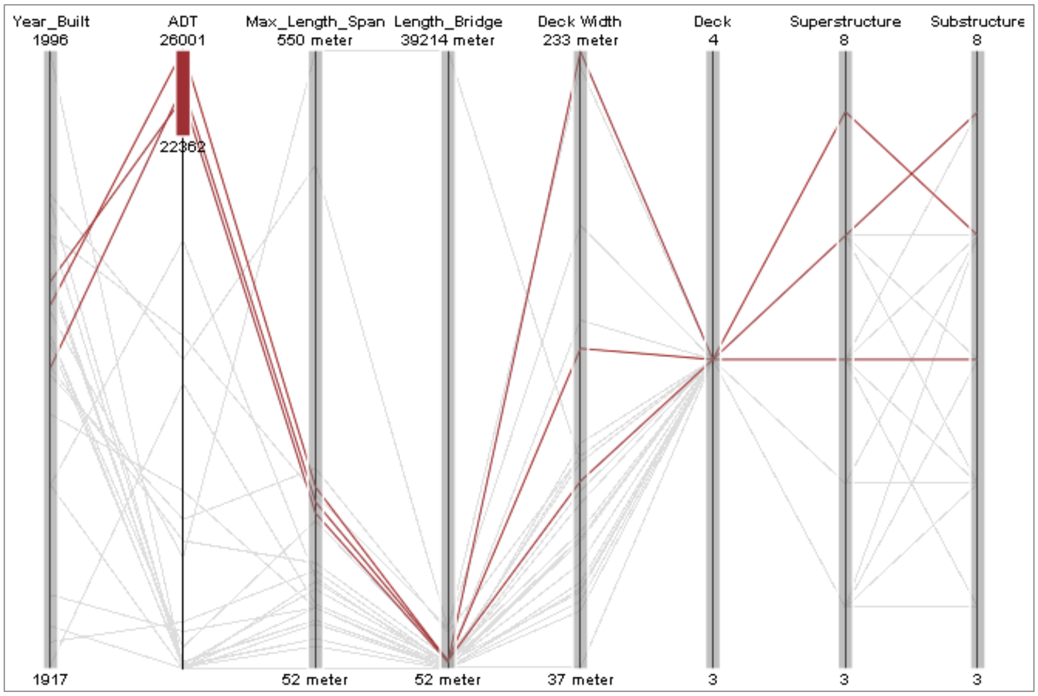


Figure 30: The parallel coordinates visualization for selected bridges, showing ADT from 22362 to 26001

4.3.3 Scatter Plot Interactive Visualization

Multiple scatter plot visualization of bridge data can be implemented including data correlations between different bridge parameters within the bridge database. In this tool, there are mainly two parts (Figure 31), the large area on the left side is the plotted area and the right side is the map for displaying the location of a selected bridge. In the plotted area, the red dots present all the selected bridges. The dot with darker color means there are more bridges overlapped since they have the same value. If the mouse cursor moves over an interested dot (bridge), the information of the specific bridge will display. If an interested bridge is selected in the scatter plot, the bridge location will show up in the right-side map.

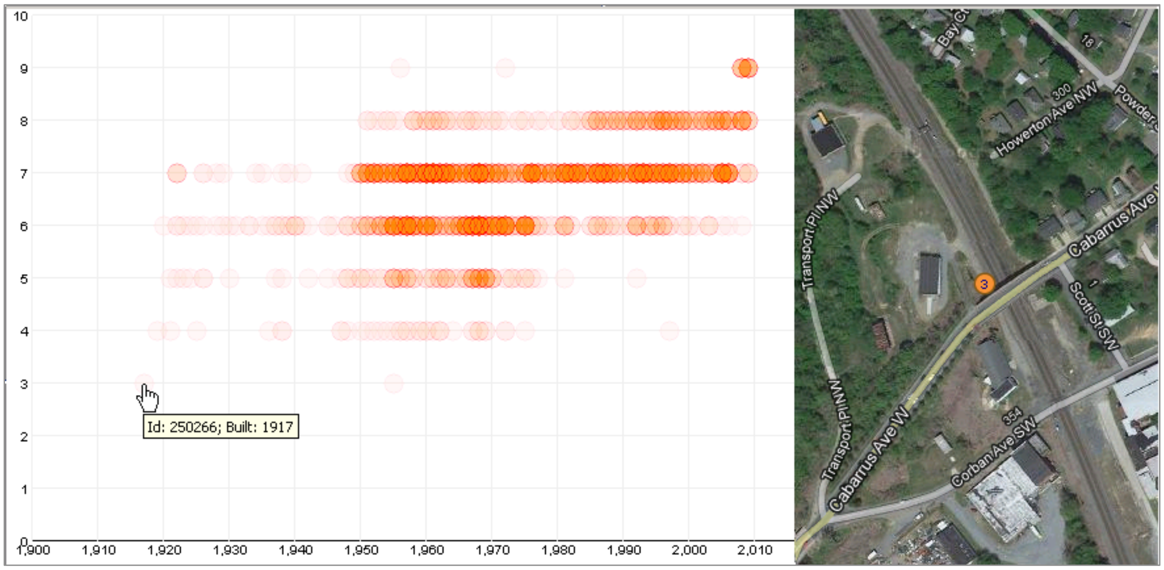
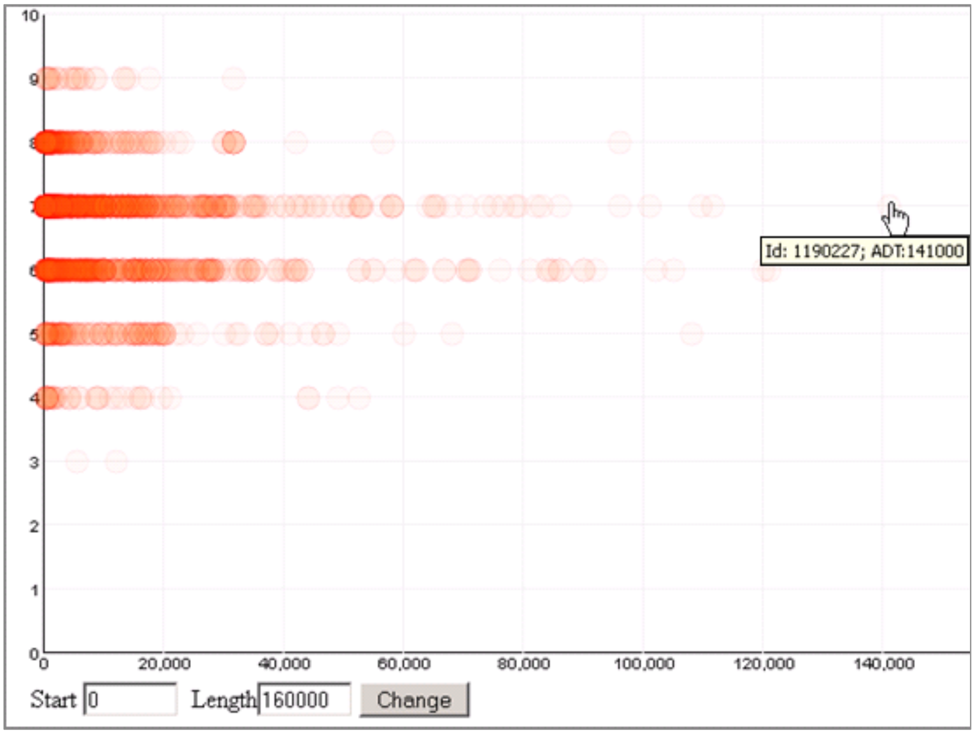
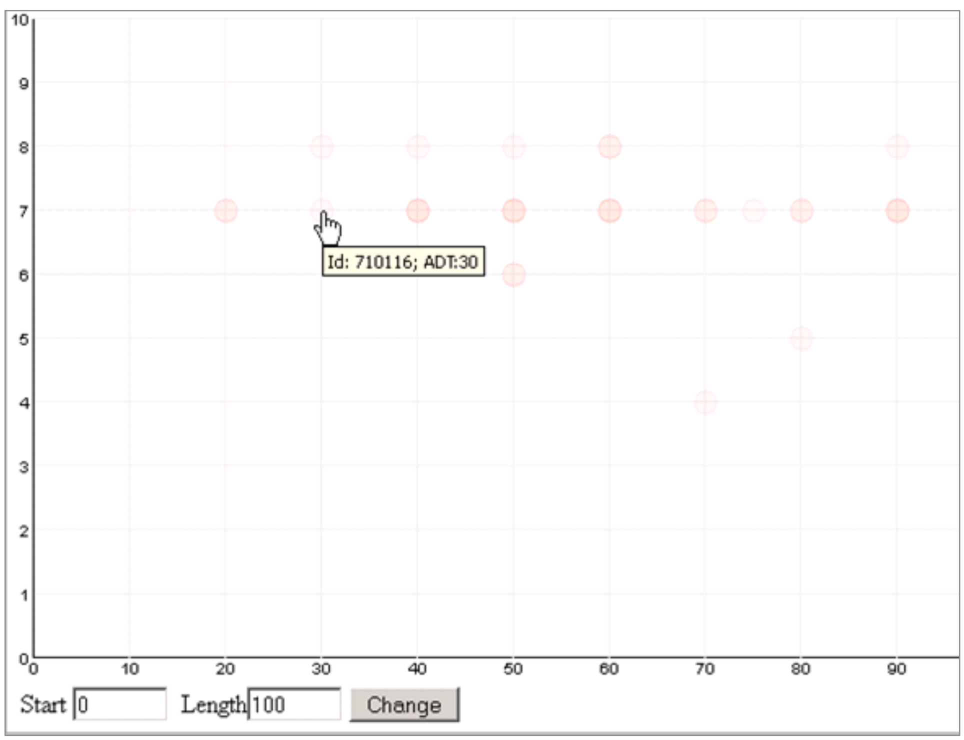


Figure 31: The scatter plot for comparison of deck condition rating and the year built of the bridge

Figure 32a shows the scatter plot correlation between deck rating and average daily traffic (ADT). A critical issue with comparative studies of different bridge parameters is that the scale used may be different; hence, Bridge-WGI allows axes transformation. In Figure 32a, the horizontal axis is the value of ADT which scales from 0 through 160,000, and the vertical axis is the value of deck condition rating from 0 through 9. This interactive visualization also allows user to view a selected fragment of ADT by using input boxes at the bottom. The scatter plot in Figure 32b shows that the horizontal axis scaled from 0 through 100.



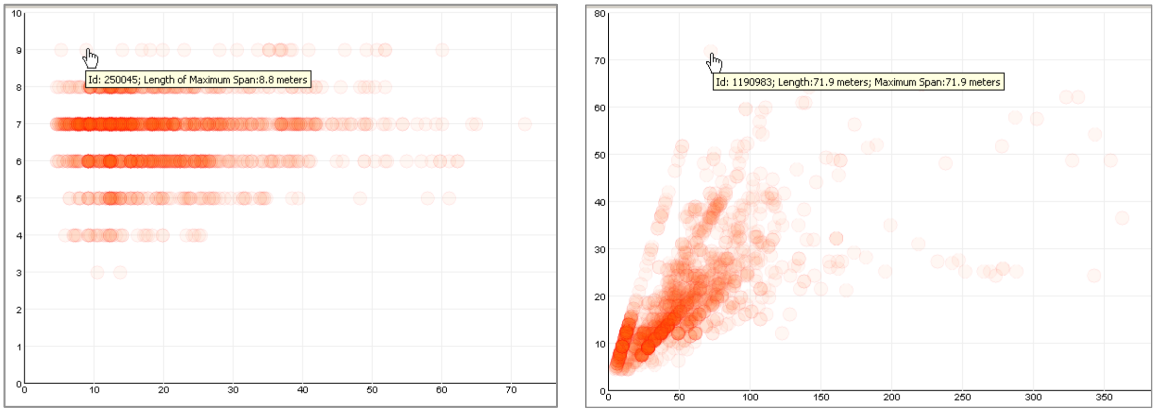
a). ADT from 0 through 160000.



b). ADT less than 100.

Figure 32: The scatter plot of comparison of deck condition rating and ADT of bridges

Figure 33a shows the scatter plot for deck condition versus the maximum span length and Figure 33b shows the scatter plot for maximum span length versus the length of the bridge.



a) Deck Condition vs. Max Span Length b) Max Span Length vs. Length of Bridge

Figure 33: The scatter plot of comparison of deck condition rating vs. maximum span length, and maximum span length vs. length of the bridge

4.3.4 Choropleth Map Visualization

The choropleth map (thematic map) provides an easy way to visualize how a measurement varies across a geographic area (geospatial variability). The technique is used for visualizing the bridge condition distribution cross all states. Single hue progression is used for mapping the bridge condition ratio, which is the number of the bridges with in a specific condition rating value divided by the number of all bridges in the state larger ratio results in darker colors.

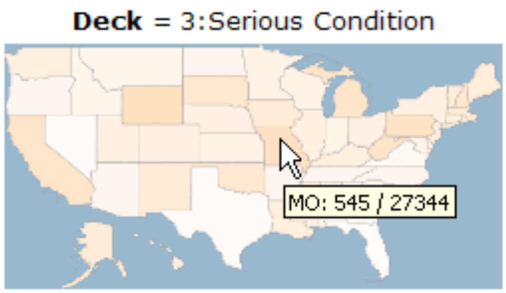
As shown in Figure 34, the choropleth map shows bridge deck with critical condition. While moving the mouse over a state in the map, a ratio OK 716/25829 is shown (Figure 34d), which means that Oklahoma has 716 bridges with critical deck condition and there are 25829 bridges in this state. To compare two states, i.e. between

Oklahoma (darker background) and North Carolina (lighter background), the ratio of Oklahoma is larger than that of North Carolina.

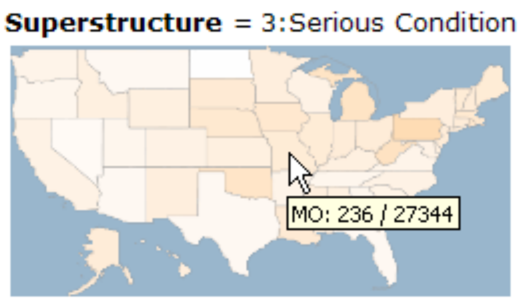
The map-based visualization can answer questions such as what distribution of the bridge condition in the United States and which state has more percentage of failed bridges. For example, Figure 34 shows the bridge condition rating distribution. Figure 34a shows Missouri and Wyoming have darker color which means these two states have more bridges in serious deck conditions.

4.3.5 Pie Chart and Bar Chart Visualization

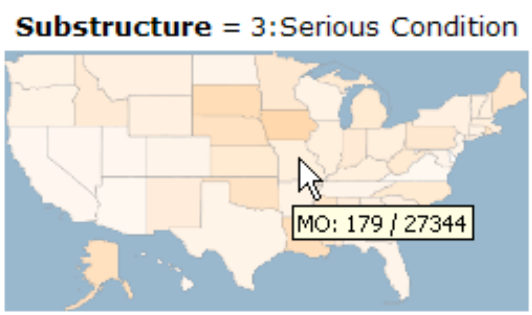
Pie chart and bar chart visualization can provide clear understanding of the statistical visualization for bridge conditions. In Bridge-WGI, pie charts and bar charts are applied for visualizing the bridge data categorized by the condition rating. Bar chart provides the number of the categorized bridges while pie chart provides the percentage of the distribution. Figure 35 shows the deck condition rating distribution nationwide. Pie chart describes the bridge condition rating distribution in each state, the size of circle stands for the total number of bridge in the state, while the sectors stands for the percentage of the categorized the condition rating, which is from “0” through “9”. Pie chart visualization offers the bridge manager a clear presentation of the bridge distribution for each state. For instance, there are more bridges in Texas than that in Florida. The entire bridge condition in Texas is better than those in California, since there are more green color sectors (good condition) in Texas while there are more red color sectors (Bad condition) in California.



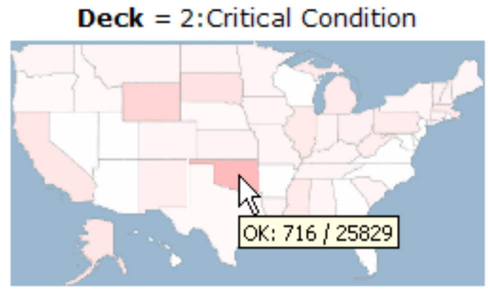
a). the number of bridges which have deck rating condition is 3 (serious condition) , the selected state is Missouri.



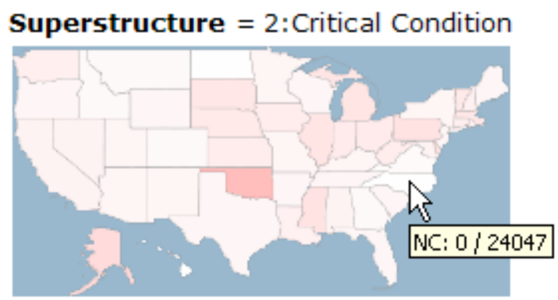
b). the number of bridges which have superstructure rating condition is 3 (serious condition) , the selected state is Missouri.



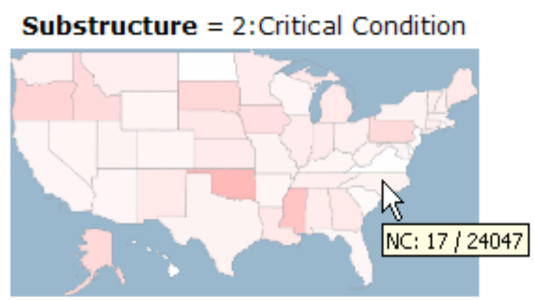
c). the number of bridges which have substructure rating condition is 3 (serious condition), the selected state is Missouri.



d). the number of bridges which have deck rating condition is 2 (critical condition), the selected state is Oklahoma.



e). the number of bridges which have superstructure rating condition is 2 (critical condition), the selected state is North Carolina.



f). the number of bridges which have substructure rating condition is 2 (critical condition), the selected state is North Carolina.

Figure 34: The choropleth map shows the distribution of the bridge rating conditions

Note: 1) The information displayed in map is the number of the bridge with selected condition rating over the number of all the bridges in the state, for example, the notation "OK: 716/25829" in d) means that there are 716 bridges with deck rating of 2 (critical condition) while there are totally 25829 bridges in state Oklahoma (OK). 2) The color indicates the value of the percentage of number of the bridge, darker color shows higher percentage.

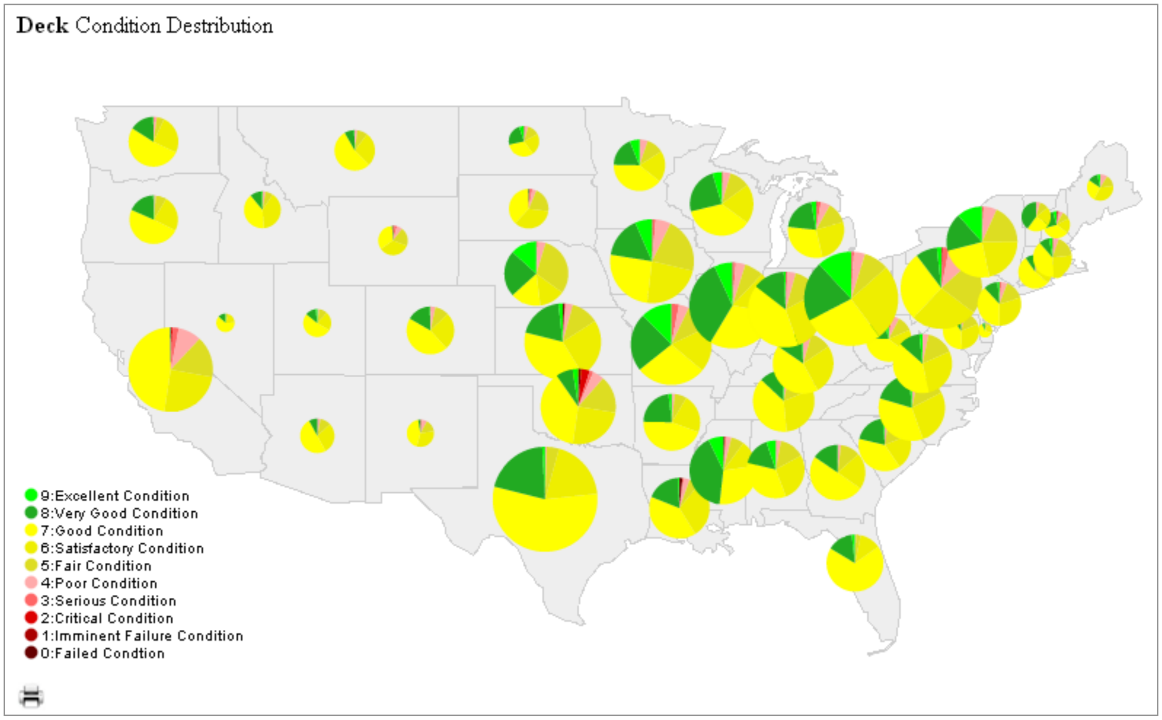


Figure 35: Symbol map with pie chart showing the deck condition rating distribution

A detailed bridge condition rating distribution for a selected state is illustrated in Figure 36. This statistical visualization provides detailed information for bridge condition in a state. The combination of bar chart and pie chart provide a clear statistical visualization to bridge managers.

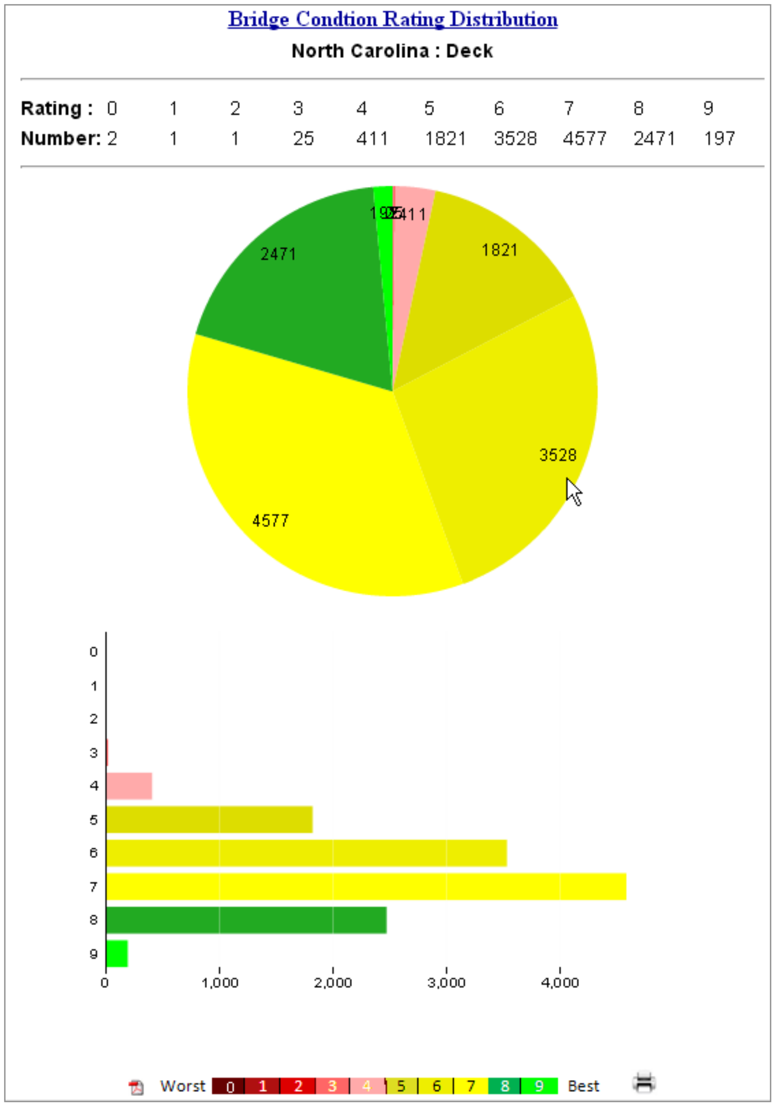


Figure 36: Bar chart and pie chart showing the bridge condition rating distribution for North Carolina

4.4 User-Defined Criteria Query for Decision-Making Support

4.4.1 Bridge Management

In bridge condition analysis, there are many factors that may contribute to bridge deterioration, such as traffic volume, age of the bridge, climate and environmental conditions, structural type and traffic pattern. Bridge manager maybe interested in comparing different bridge attributes or parameters to identify the critical causes for bridge deterioration. Bridge-WGI provides filter-like functions for user query. In this section, a user-defined criteria query (UDCQ) is introduced for the advanced query application.

4.4.2 User-Defined Criteria Query (UDCQ)

The UDCQ is a program-condition-like formula. In a UDCQ, the formulae are generated by using bridge parameters (deck, superstructure, substructure, ADT, etc.), arithmetic operators ("+", "-", "×", "/"), comparison operators (">", "<", "="), and logic operators (AND, OR). Figure 37 shows a screenshot of the UDCQ application interface, the left side is the area for query generation; right side is the area for the map view. The search results will be displayed in map format.

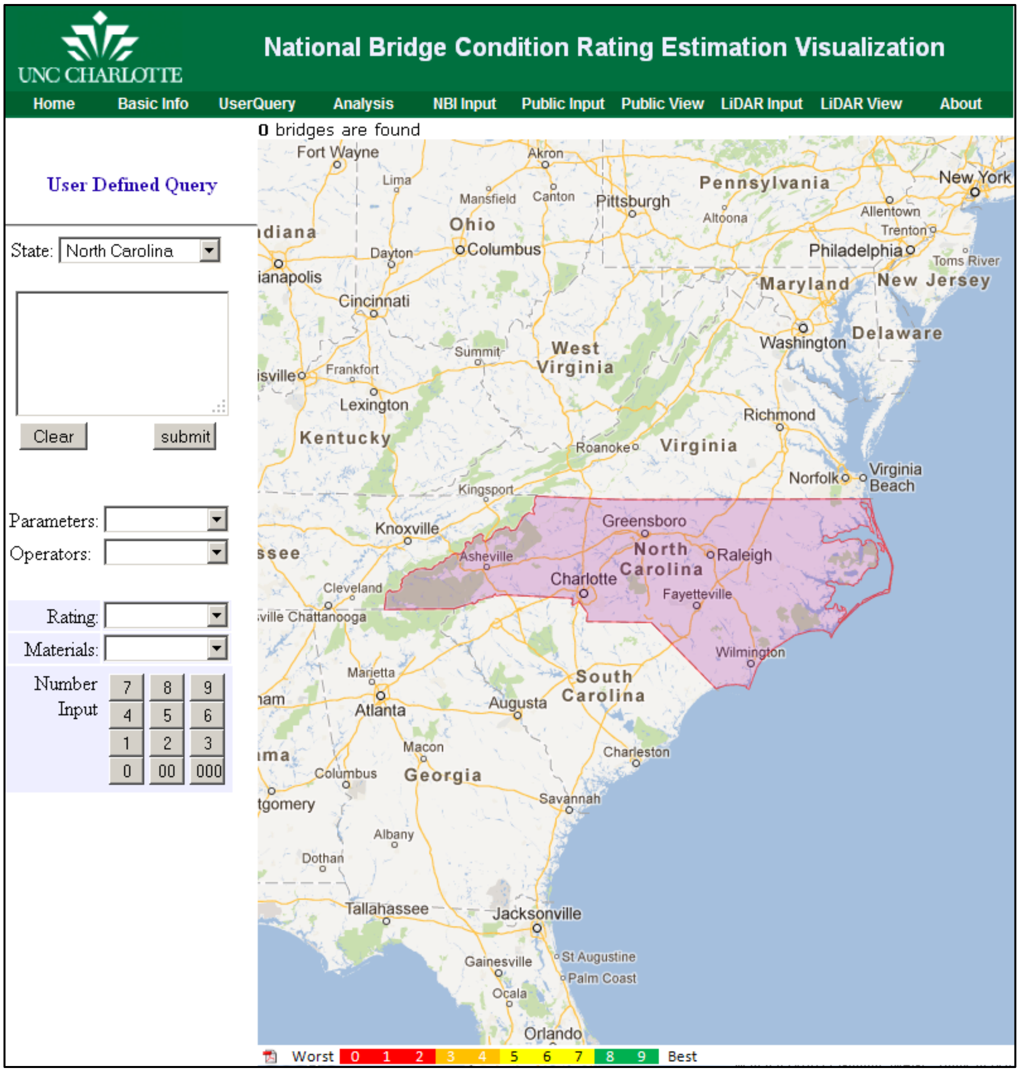


Figure 37: The graphic user interface of user-defined criteria query for decision-making support

Bridges can be grouped together based on different criteria. In addition, user-defined query can efficiently support complex database applications, which can make the applications significantly more efficient (Chaudhuri and Shim, 1999). Figure 38 illustrates the operation for processing a UDCQ.

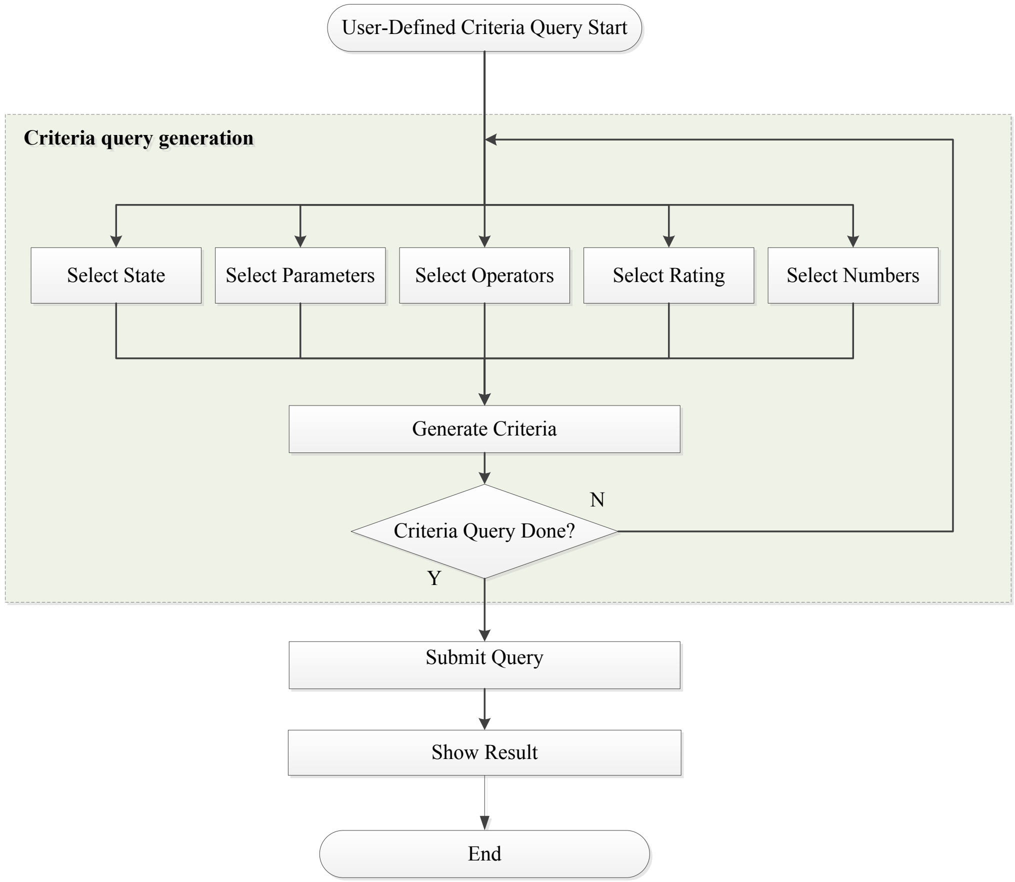


Figure 38: Flowchart for User-Defined Criteria Query (UDCQ) Operation

Figure 39 shows an example of the designed user-defined criteria query application. The user-defined query is about "Search all the bridges among those (1) the deck condition rating is greater than 8 (very good condition) and (2) the superstructure condition rating is greater than 8 (very good condition) and (3) the ADT is greater than 60,000 in North Carolina." The result shows that there are two bridges which meet the criteria.

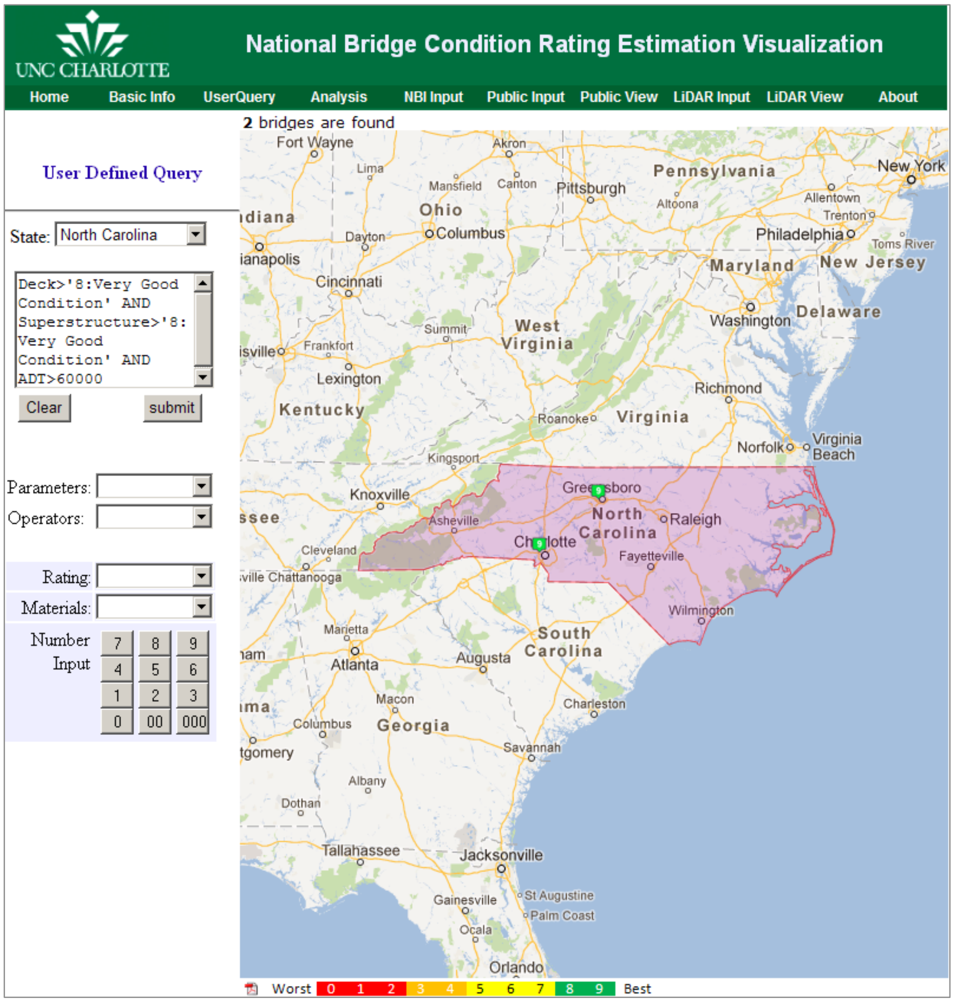


Figure 39: Screenshot of the application example for user-defined criteria query

Figure 40 shows the interface for user-defined query generation. In lower part of Figure 40a, the number keypad is designed for number input for criteria query equation. Figure 40b is the interface of input for state selection, which includes all the 50 states and Washington DC in the United States; Figure 40c shows the input interface for parameters including deck, superstructure, substructure, ADT, latitude, and longitude. Figure 40d shows the operator for making mathematical formula or logical formula. Figure 40e is the values for bridge condition rating which are from “0. Failed Condition” to “9.Excellent Condition”. Figure 40f shows the value for kind of materials of bridge.

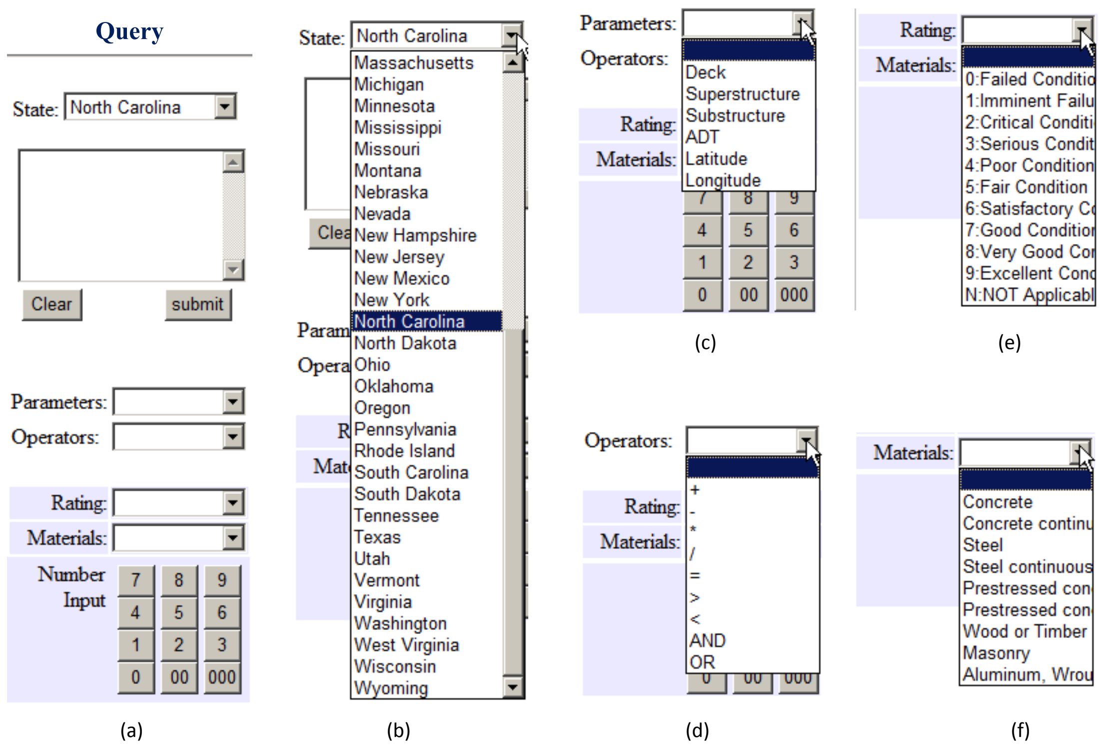


Figure 40: Detail explanation of interface for user-defined criteria query

(a) The main interface for query generation. (b) List box for selection of States (All States in US). (c) List box for selection of parameters. (d) List box for selection of operators. (e) Condition rating value for the deck, superstructure and substructure condition rating. (f) Material values for bridge.

4.5 Summary

Visualization techniques are used for large scale dataset, such as the NBI bridge database. With this visualization system, 1) bridge managers can view the general information of the any interested bridge; 2) bridge managers can generate several knowledge of the bridge damage from various views; 3) interactive analytical visualization make it possible to figure out the correlation between two variables of the bridge dataset; 4) choropleth map assist the bridge managers to get a general idea of the bridge condition distribution all states. Finally, Bridge-WGI includes UDCQ for user query-based visualization selection.

CHAPTER 5: CITIZEN SENSING APPLICATION IN BRIDGE MONITORING

5.1 Introduction

In this chapter, the citizen sensor is suggested as a tool for constructed bridge monitoring, which is useful for the management of systems where several discrete structures that may be miles away from each other, covering large spatial areas and render networked sensing systems challenging. Examples of such systems include bridge structures on the national highway, railway and waterway systems and power transmission and distribution structures. Bridges, in particular, can benefit directly from citizen sensors, since cars and pedestrians pass over them frequently.

Similar to the 911 emergency phone call system, citizens can report imminent danger to a bridge by adopting smartphone technologies and wireless Internet reporting. Due to their wide spatial locations, networked bridge monitoring is difficult to implement. For network-wide application, several different monitoring schemes may be needed to simultaneously capture different schema data to help enhance bridge health state diagnosis, which ultimately will improve overall bridge management (Wang et al., 2010).

This chapter introduces the citizen sensing application in Bridge-WGI. Two applications are developed for citizen sensor report damage. One is a Web-based citizen sensor damage report apps and the other is smartphone citizen sensor apps. The development tools and advantages of using citizen sensing technology are also discussed.

5.2 Citizen Sensor

The objective of a citizen sensor system for bridge monitoring is specifically the motivation of voluntary information sharing amongst citizens to enhance bridge condition monitoring and potential disaster responses. There are three methods to upload the citizen sensing data to the bridge database, including cellular network for smartphone, wired Internet connection for desktop computer or laptops, and WiFi with wireless connection capability for mobile device including tablet, laptop and smartphone. Figure 41 illustrates these three methods as well as their advantages and limitations. Figure 42 illustrates the procedure for volunteer sensing data uploading based on webpage, which can be accessed via wired Internet or WiFi. To report bridge damage, the volunteer first need prepare all the documents to be uploaded, such as imaging, then following the data entry procedure as shown in Figure 42 to upload data. There are two primary steps: 1) locating the bridge and 2) uploading the document to database system.

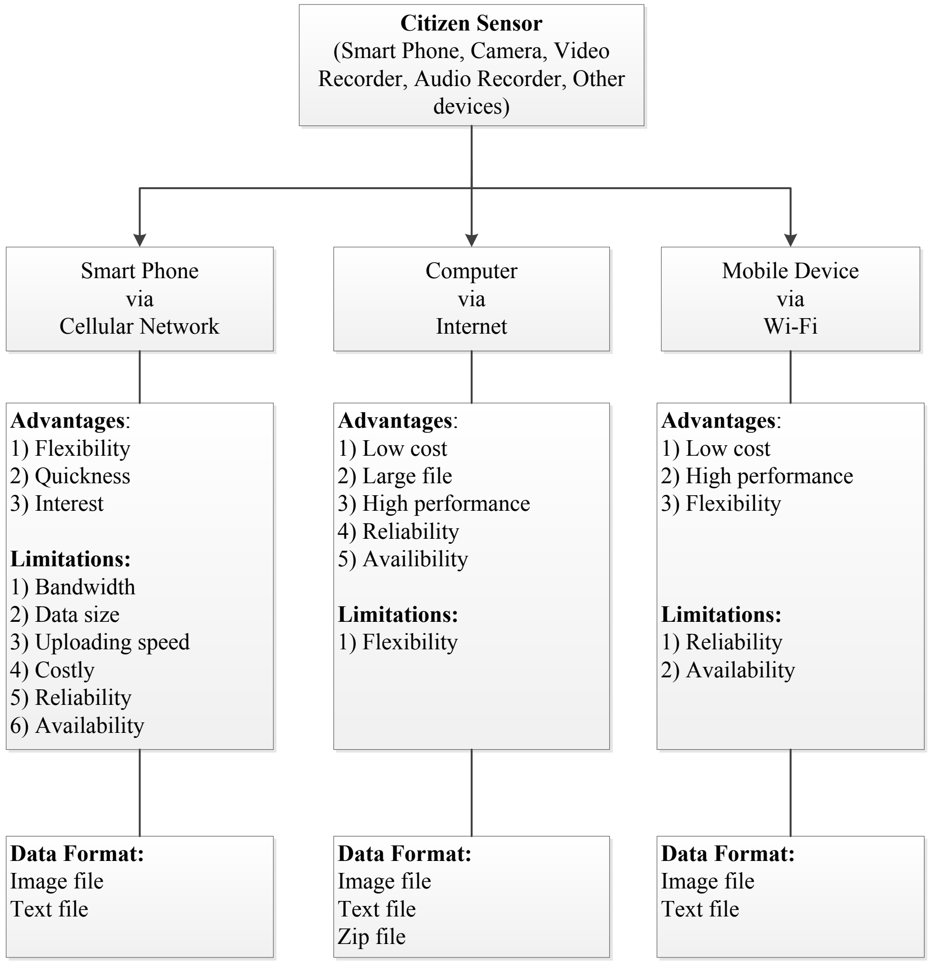


Figure 41: Three methods for citizen sensing data report

5.3 Citizen Sensor Web Applications

5.3.1 Citizen Sensor Web-based Applications

When a volunteer finds an event and wants to upload the information to the bridge database, required information includes the location of the bridge (latitude and longitude), the name of the bridge if available, the date and time of the reporting, the detailed descriptions, and other associated information will be stored in the database. For mobile devices which have geographic position system (GPS) tracking features, the geospatial location of the bridge can be identified. A bridge locator interface is designed for user to locate the bridge for data uploading. Figure 42 illustrates the bridge locating and data uploading.

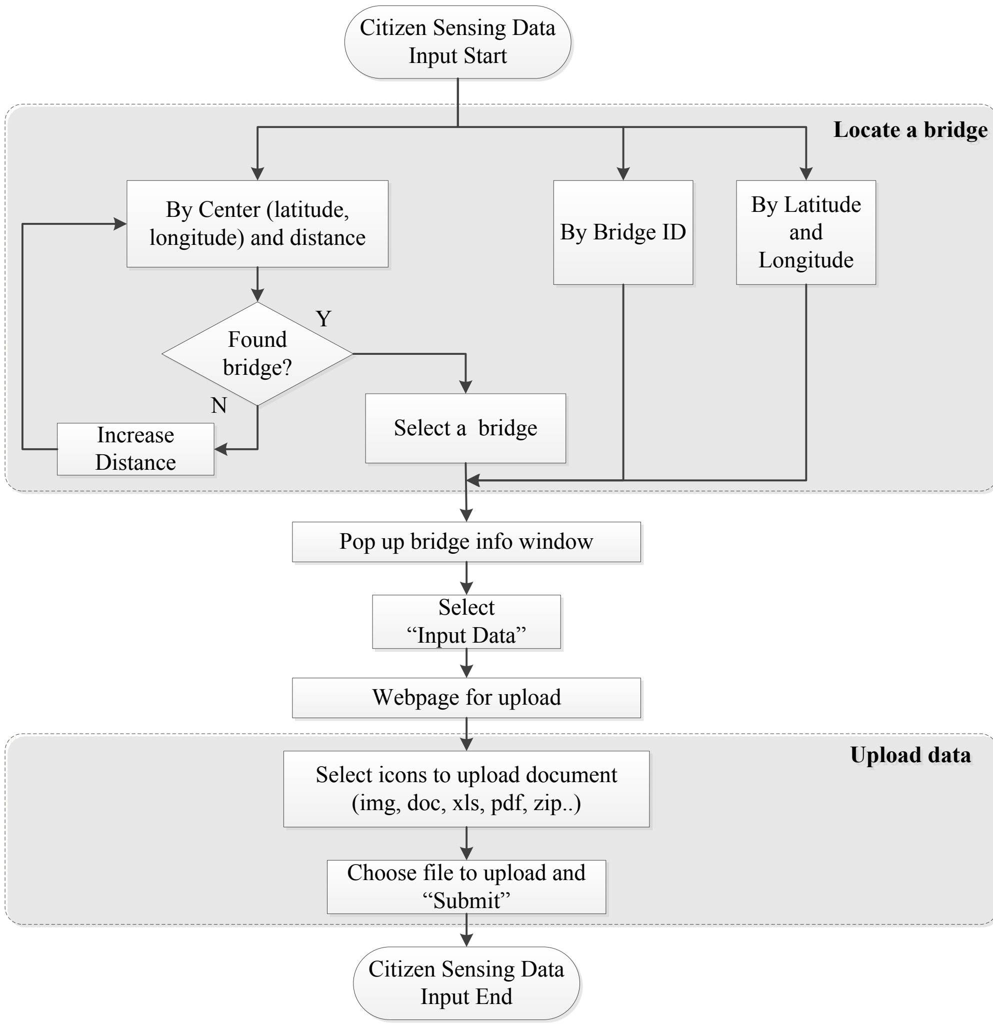
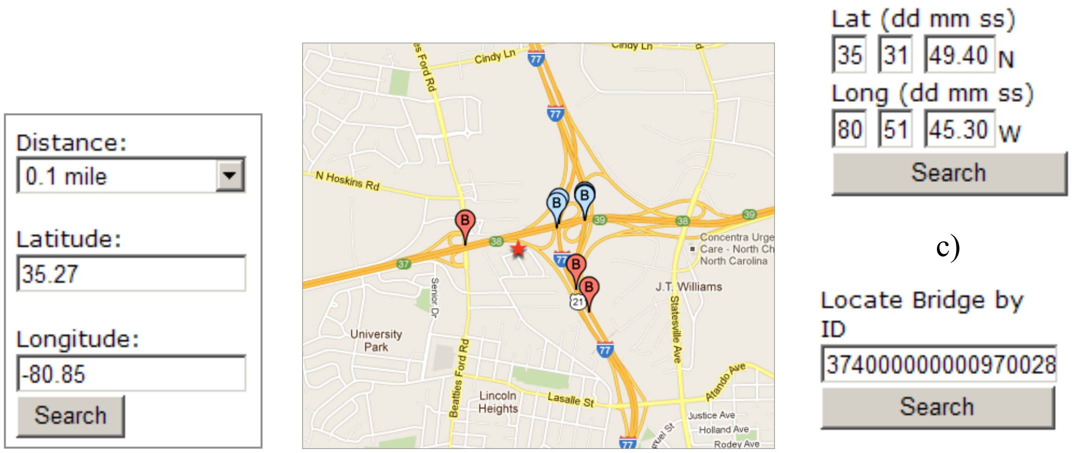


Figure 42: Flowchart for citizen sensing data entry

The system has three options: First method is using latitude and longitude (in decimal format), and a distance (Figure 43a), the map will display all the bridges in the area within a radius (distance) from the center (latitude and longitude) (Figure 43b). The second option for locating a bridge is using the exact latitude and longitude in degree, minute and second format (Figure 43c). The third method is to locate a bridge by using the bridge ID (Figure 43d). Once the bridge located, the bridges are displayed on map.

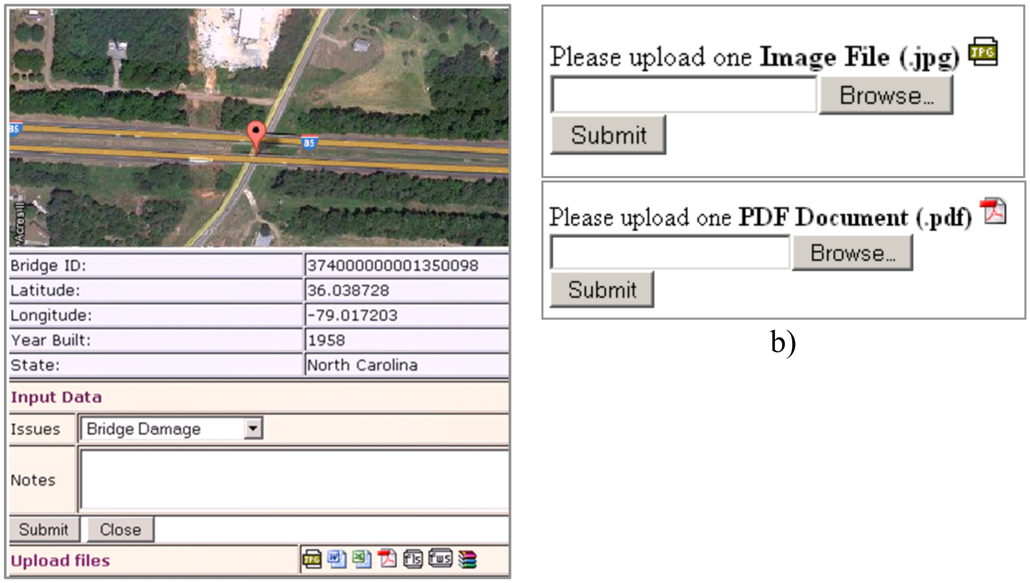


a) b) c) d)

Figure 43: Web-based bridge locator

5.3.2 Data Entry

Once the bridge is located, the related information should be uploaded to the database. There are mainly four parts for the citizen sensing data upload (Figure 44a), including geospatial map view which show the bridge location, general bridge information (bridge ID, latitude, longitude, the year built, and state), condition input (reporting issue, and description). File uploading part provides the functionalities for uploading various documents including images (.img), MS Word document (.doc), MS Excel document (.xls), Adobe document (.pdf), ZIP file (.zip) and LiDAR scan data (.fls and .fws). In the document uploading window, user can select a document to upload (Figure 44b).



a) b)

Figure 44: Public data entry interface (a) and data upload interface (b).

5.4 Citizen Sensor-Smartphone/Tablet Applications

5.4.1 Smartphone App Development

Current Bridge-WGI is complimented with Bridge-Locator, which helps user to upload bridge information via smartphones: Bridge-Locator is developed on Basic4Android platform. Basic4Android is a Visual Basic-based programming language for almost all Android devices. Basic4android (http://basic4ppc.com/) is an IDE designed specifically for android mobile application development. In addition, Basic4Android is a fully featured GUI designer, which supports multiple screens and orientations with WYSIWYG visual design feature. The advantages include outstandingly supporting for database, GPS, Web services, sensors, CSV, and graphics. Furthermore, once a program is compiled, it can be directly deployed on android, requiring no other run time dependencies (Basic4Android, 2011).

By using the development IDE of Basic4Android and Android SDK, Bridge Locator, the application for this distributed sensing, is developed to serve as the “citizen sensor”

(Stein, 2011). Bridge Locator (citizen sensor) data collection function within Bridge-WGI allows citizen sensor to capture/annotate/upload bridge data. Bridge Locator can read in bridge location data from the database systems and locate bridges near the user within a certain radius using the built-in GPS capabilities. The main features of Bridge Locator include user selection of a search radius (in miles), and use the smartphone's 3G/WiFi/GPS signals to locate all the bridges near the user within the search radius. The accuracy of the bridge location is dependent on the device's GPS signal and the cell tower triangulation. The current location can be set to auto-update in the settings every 30 seconds to 2 minutes. This is ideal if the user is driving in a vehicle and wants to automatically see an updated list of bridges without having to press the "Bridges near me" button. Figure 45 shows the operation procedure for smartphone application of Bridge Locator.

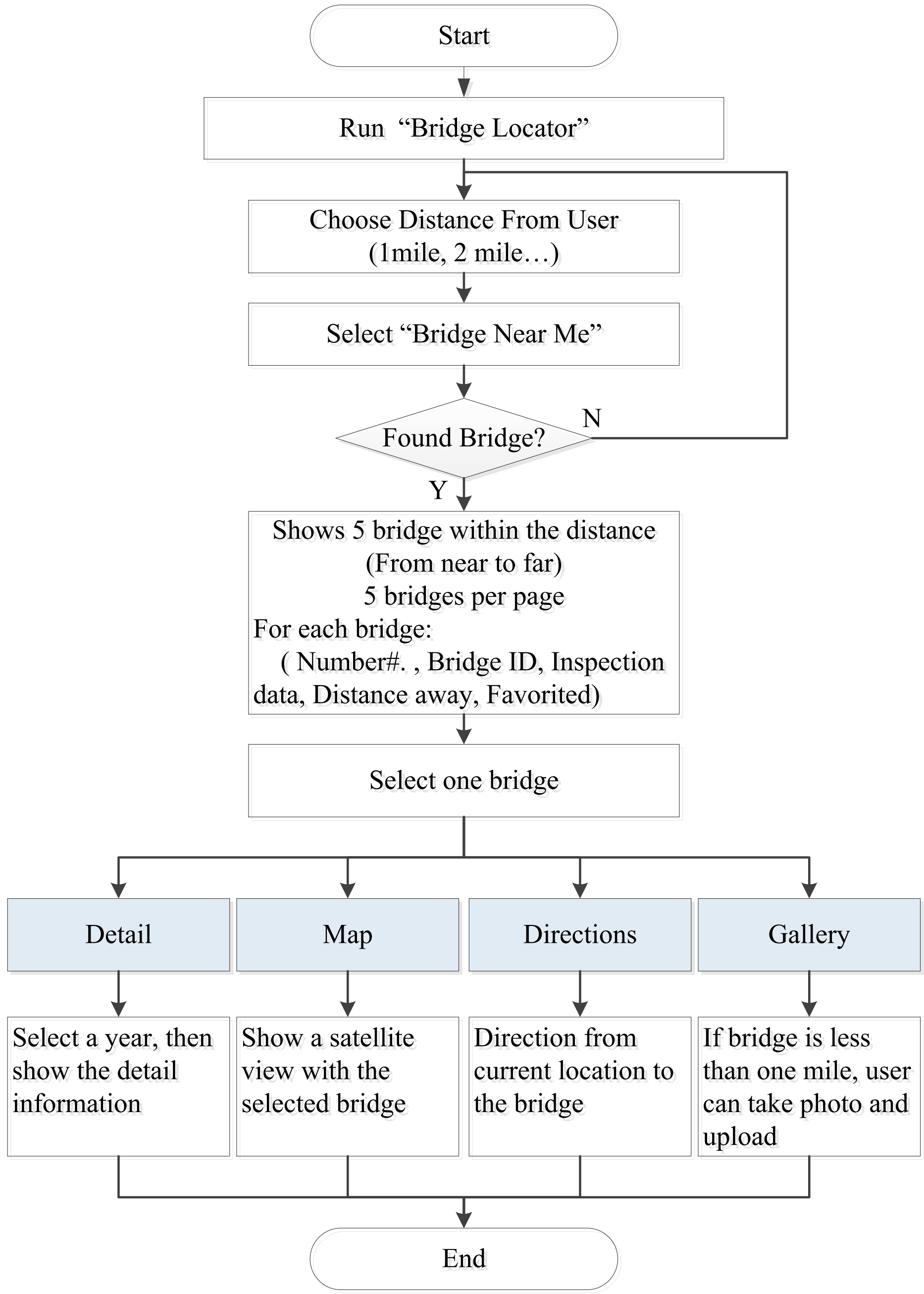


Figure 45: Flow chart for smartphone application (Bridge Locator)

Figure 46 shows a screenshot of Bridge Locator – several functionalities are provided for user inter-activities including options for locating a bridge, listing of bridge location results that allows user to select a specific bridge and the option of changing the scope of radial bridge location (Stein, 2011). The bridges are displayed in a list. The following details are provided: 1) bridge ID, 2) indication whether bridge inspection data is available for the bridge, 3) how far (in miles) the user is from the bridge, point-to-point (“as the crow flies”), and 4) indication whether the bridge is in the user’s favorite list. Bridge Locator utilizes Google Map navigation capabilities; hence, driving directions to specific bridge can be indicated. Bridge location on Google Map in either map view or satellite view is also shown (Figure 47).

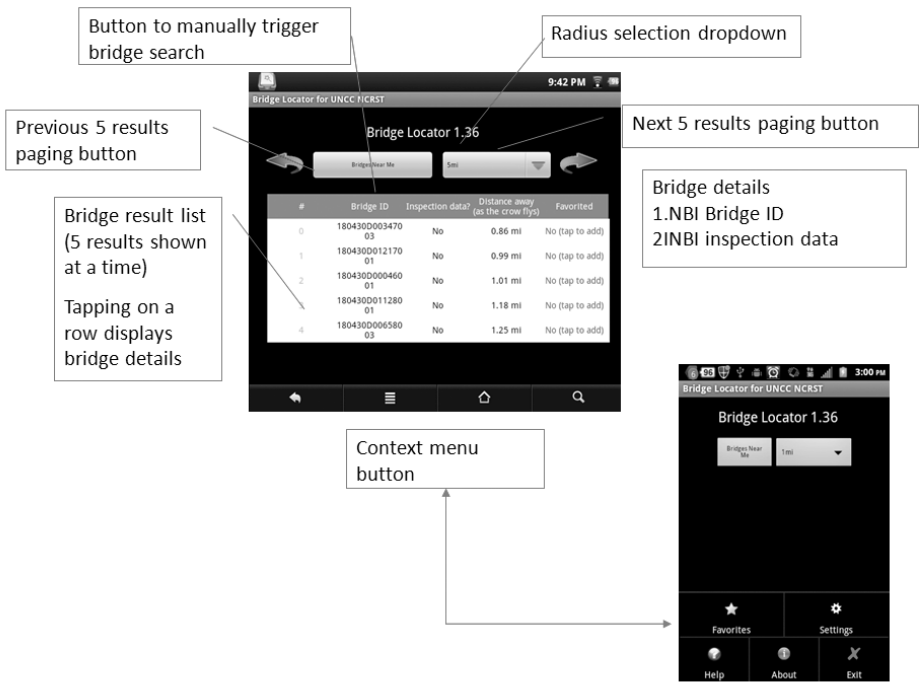


Figure 46: Bridge Locator for locating of the highway bridges near a citizen sensor

To capture data, Bridge Locator automatically verifies if the user is within 1 mile of the selected bridge. If so, Bridge Locator allows the user to capture a picture of the bridge using the phone camera. The image is then submitted to Bridge-WGI for approval (with notification via e-mail). The image is then stored in a photo album for either Web-display or integrated into the bridge database. Figure 48 shows the data collection using the activity function within Bridge Locator. Currently, Bridge Locator does not allow users to directly upload images from the gallery to the bridge database. The upload function is left to the bridge managers or to the database manager to operate.

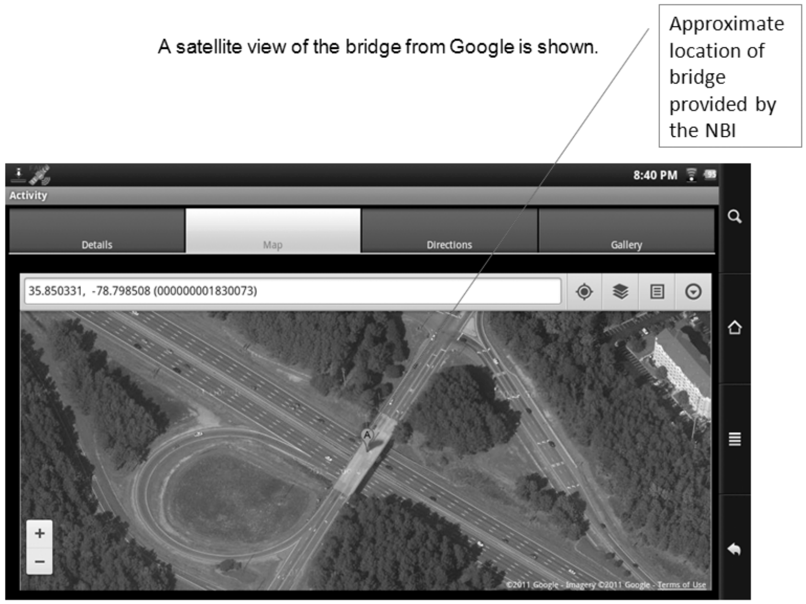


Figure 47: A satellite view of the selected bridge is shown

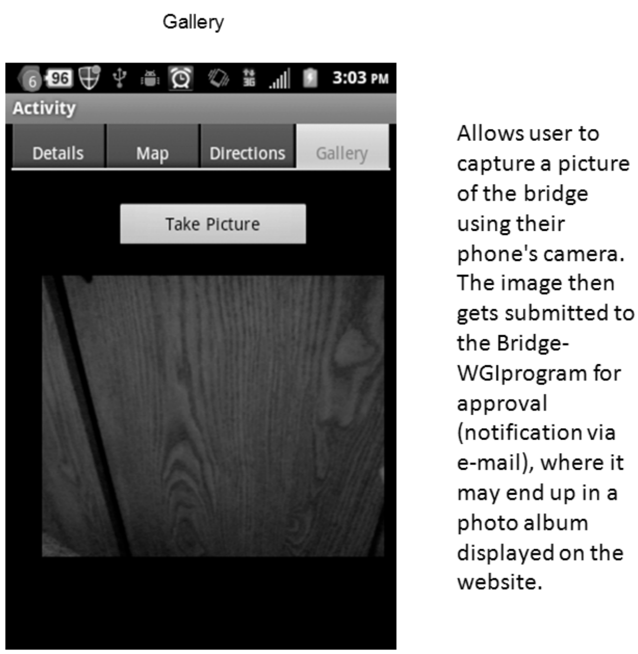


Figure 48: Bridge Locator activity including bridge imaging using camera phone

Bridge Locator has been designed as a firmware, so that users are automatically notified for upgrades and developer news.

5.4.2 Implementation and Test

The Bridge Locator is tested by using Archos 101 Internet tablet, which is a 10.1 inch tablet with 16GB Flash Memory and OS of Android2.2. The test is conducted in the Visualization Lab on UNC Charlotte campus.

To start the application, just click the apps called “Bridge Locator” (Figure 49, the icon in a circle). Figure 50 shows the interface which assists user to select a distance from current location where the user located in order to search all the bridges in this circular area with the radius of the distance (Figure 51). Choosing the “Bridge near me” function can produce the searching result which is shown and sorted by distance (Figure 52). If there are more than 5 bridges are found, the arrow in blue can help to jump to next page (Figure 53). If one bridge is selected, a new interface will popup (Figure 54), there are four tabs captioned with “Details”, “Map”, “Direction”, and “Gallery”. User needs to select an inspection year to show the detail information (Figure 55). The detailed information of the bridge (Figure 56), the map view (Figure 57), the direction from current place to the destined bridge (Figures 57, 58 and 59) and the Gallery (Figure 61). In Figure 60, the location of the user is marked “A” in the map and the bridge is marked “B” in the map. If the bridge is close, the user can take picture (Figure 62), if not, the functionality of taking picture is not applicable.

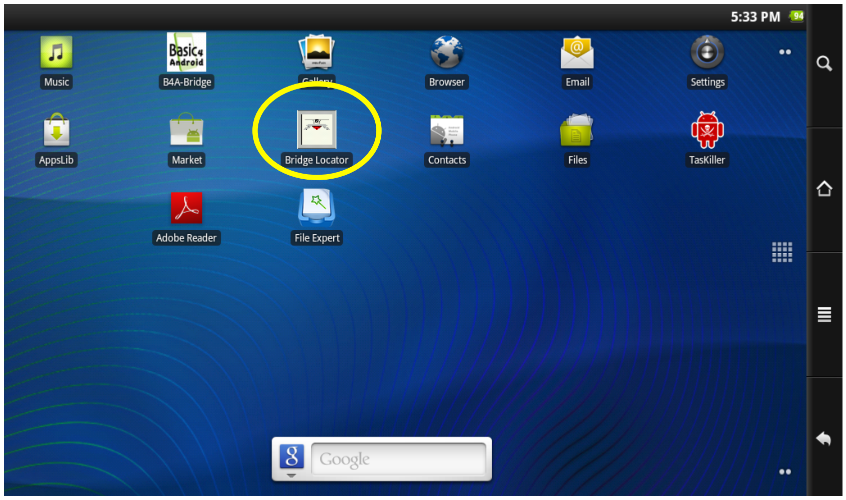


Figure 49: Screenshot for interface of the Archos 101 Internet Tablet

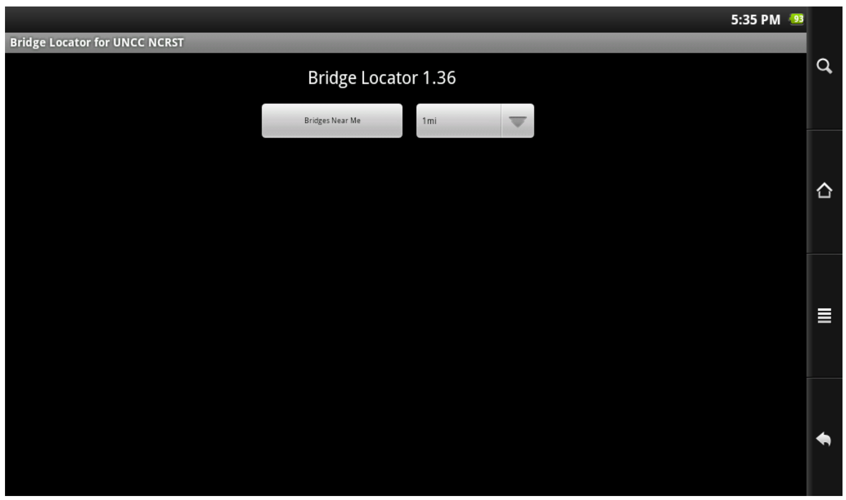


Figure 50: Bridge Locator running interface

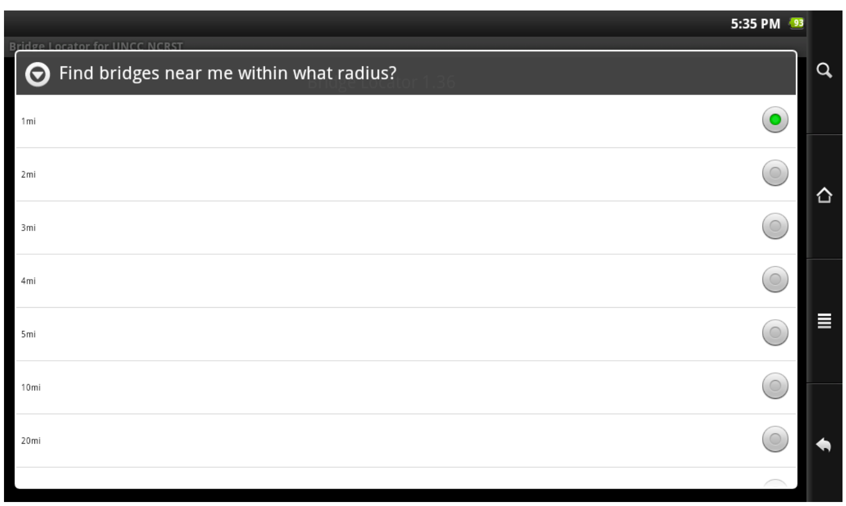


Figure 51: Screenshot for the distance selection

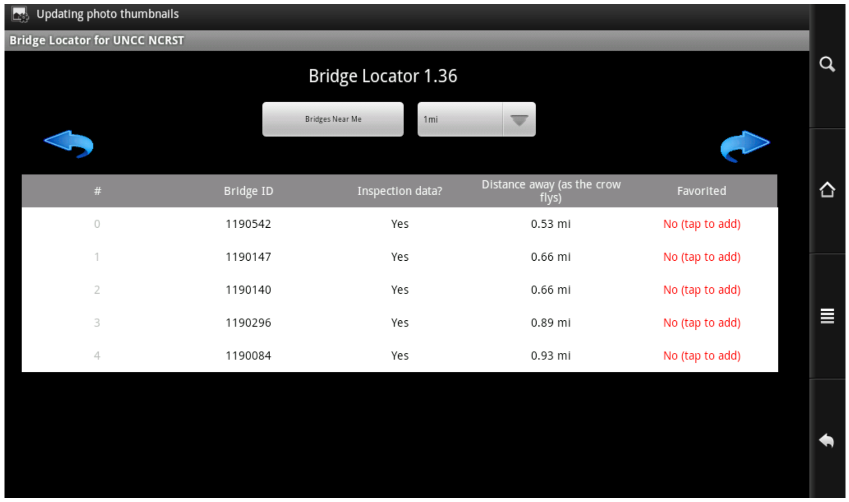


Figure 52: Screenshot shows all the bridges near the user within the distance

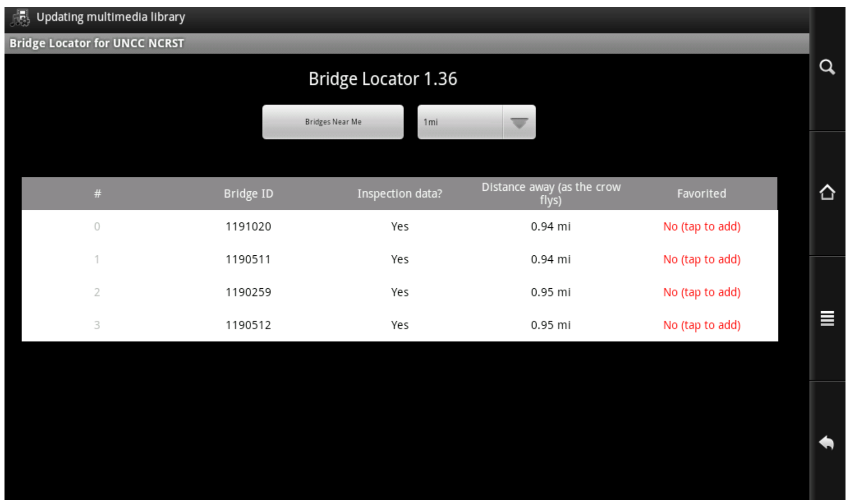


Figure 53: Display more bridges

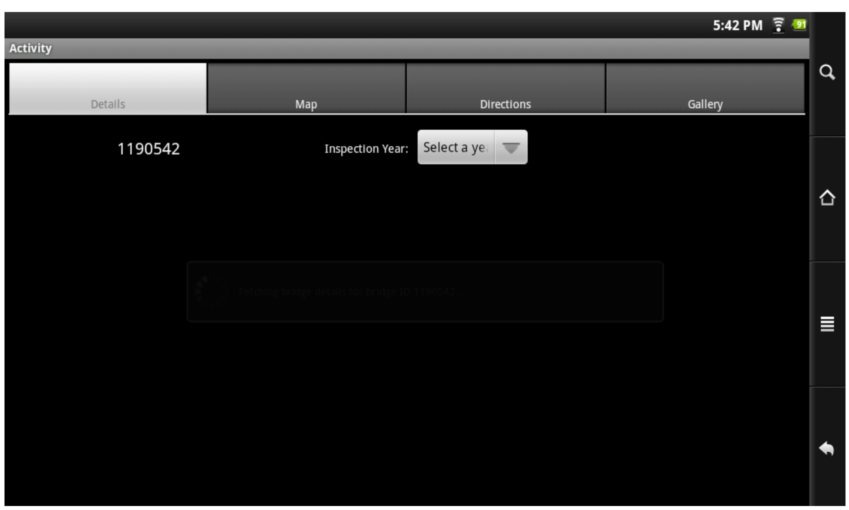


Figure 54: Screenshot shows the selected bridge with ID of 1190542

Figure 55: Screenshot for year selection

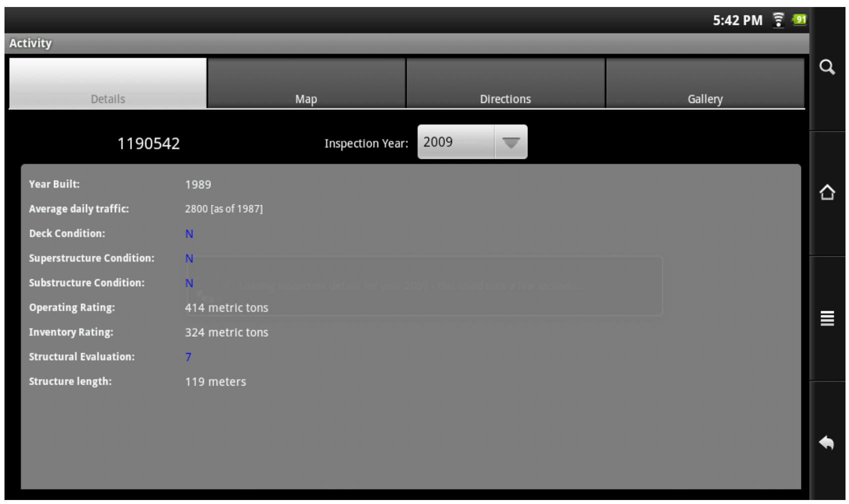


Figure 56: Detail information of the selected bridge

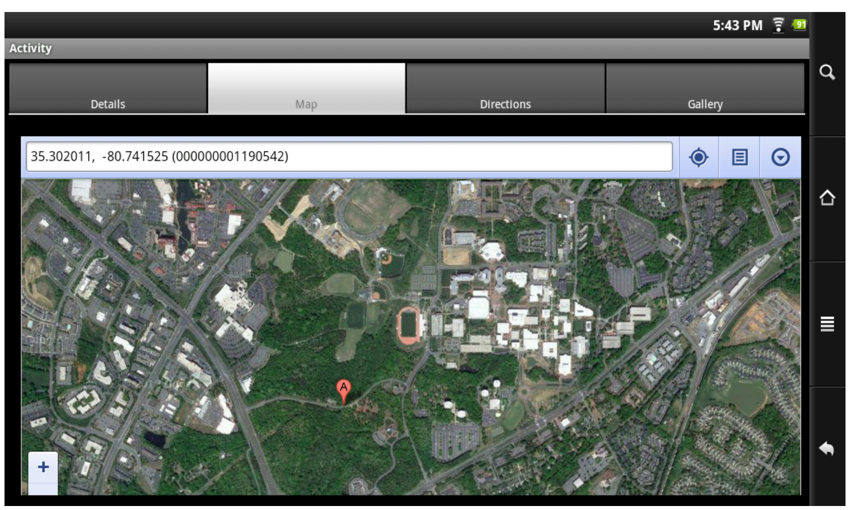


Figure 57: The location of the bridge in map

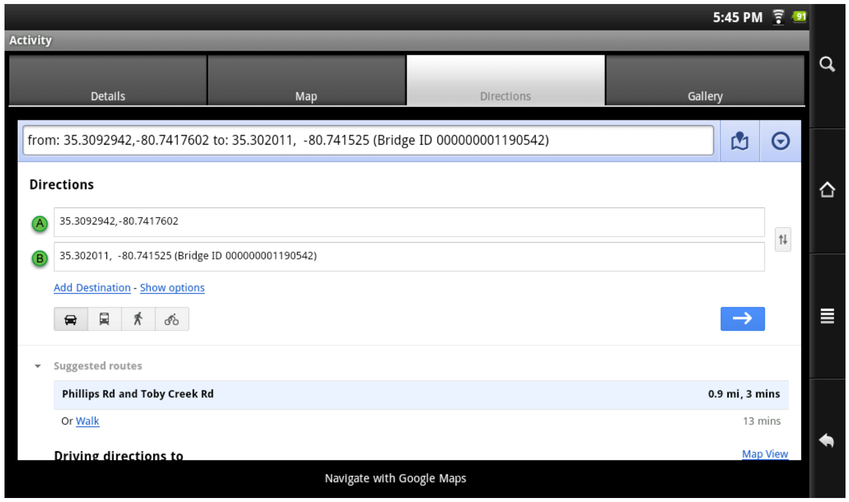


Figure 58: Driving direction from current location to the selected bridge (1)

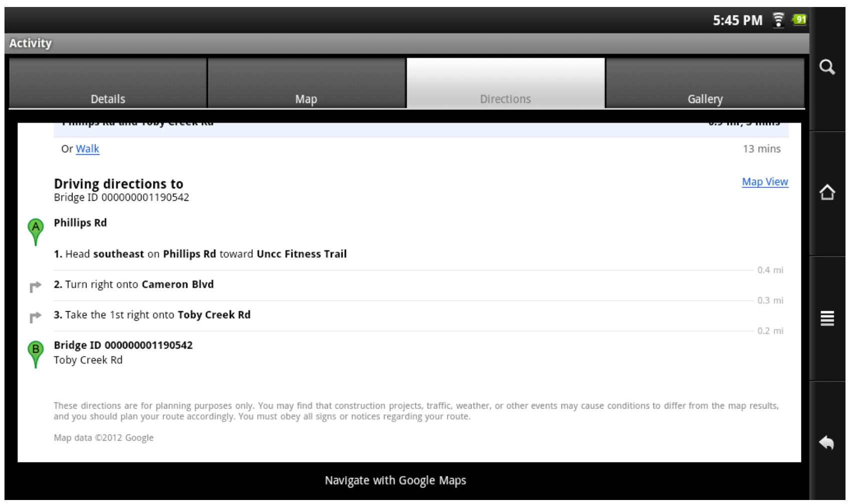


Figure 59: Driving direction from current location to the selected bridge (2)

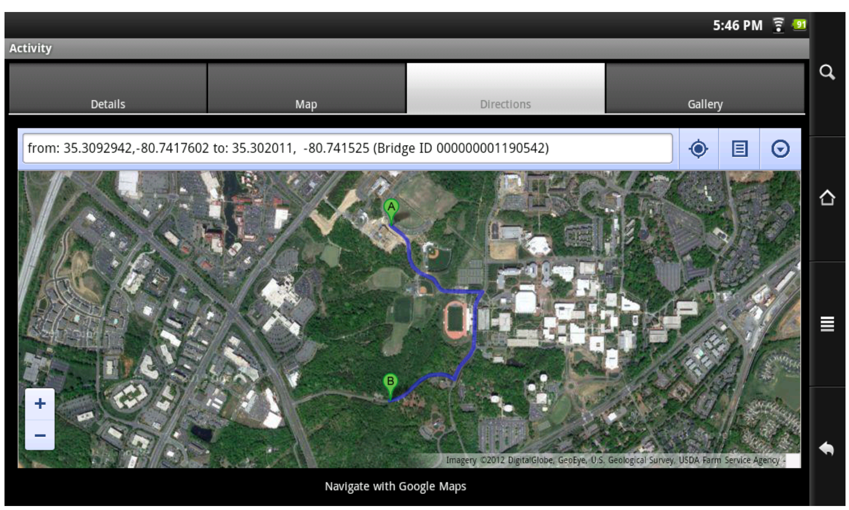


Figure 60: Map view of driving direction

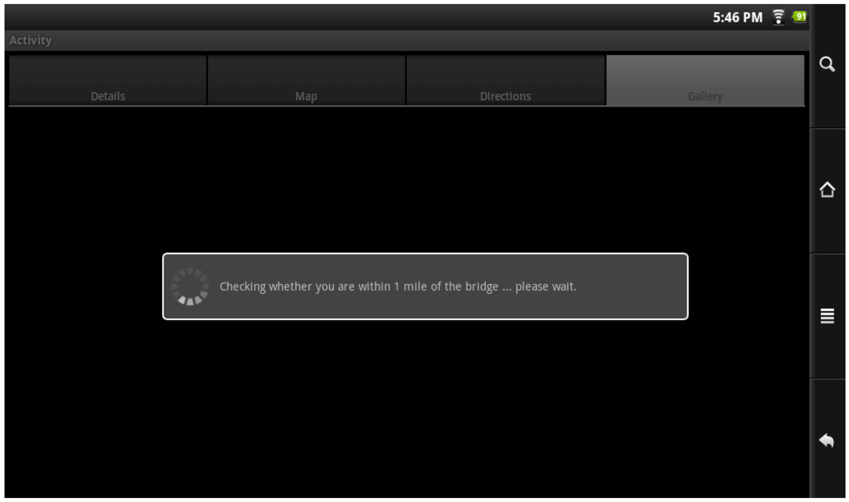


Figure 61: Screenshot for distance checking

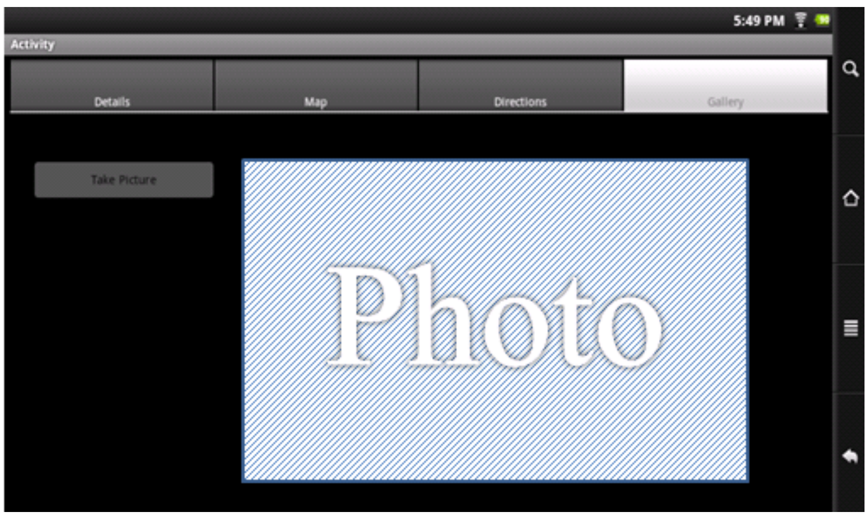


Figure 62: Screenshot for taking photo

5.5 Summary

Citizen sensor applications are presented in this chapter. Citizen sensor application can drastically increase bridge damage data and expand the bridge data. The following key points are highlighted:

- Citizen sensors can collect data and upload data with their willingness; and the shared data can be personal or socially-relevant.
- Web-based citizen sensing data uploading application is developed especially for volunteers who are able to report bridge damage.

- Smartphone citizen sensing application is developed and tested for the volunteers have the access to mobile devices.

CHAPTER 6: BRIDGE REMOTE SENSING DATA INTEGRATION AND VISUALIZATION

6.1 Introduction

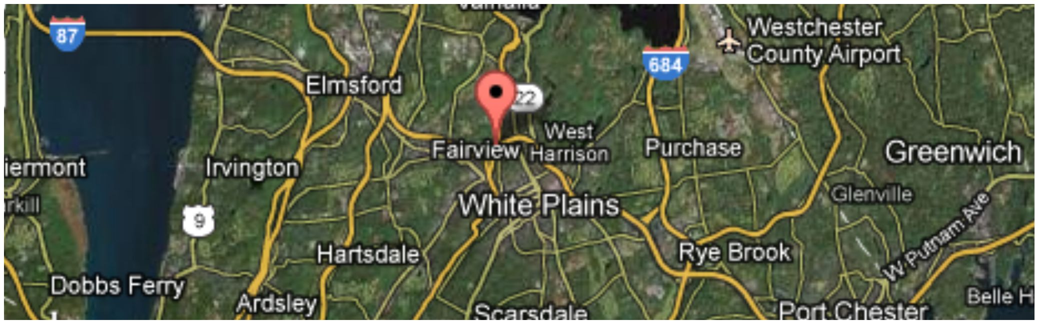
Typically, remote sensing refers to imagery or thematic information taken by airborne and satellite systems. In this research, remote sensing is defined as the sensing technique that collects information of an object, area, or phenomenon from a distance without physically contacting it (Liu, 2010). In this chapter, the integrating current bridge database and the remote sensing data including LiDAR scan and fly-over aerial image into Bridge-WGI is presented.

6.2 Remote Sensing Imagery

In this development, the satellite image, LiDAR scan and aerial photography are discussed, and LiDAR scan and aerial image are implemented in the remote sensing database.

6.2.1 Satellite /Aerial Image

Considering the spatial resolution, satellite data are classified as coarse resolution data and high resolution data. Ranging from dozens of meters to several hundred kilometers, coarse resolution satellite data are mainly used for large scale problems (Glantz et al. 2009). Satellite image is applicable for showing the bridge geospatial location and environment of the bridge. Satellite image provides a very clear geospatial location of a bridge and the highway network. In Bridge-WGI, the Google Map satellite images are used for the map view of the bridges (Figure 63).



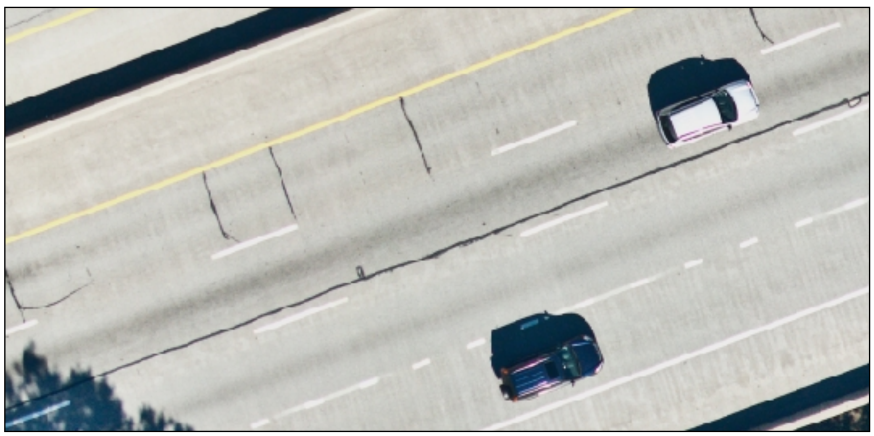
a) The map shows the location of the bridge

b) The close top-view of the bridge in map

Figure 63: The satellite image (NY Bridge # 1044659)

High resolution aerial photography makes it possible to detect the damage on deck surface of highway. This high resolution image is captured by a camera which is mounted on a flying airplane. Figure 64a shows zoom-out view of the aerial fly-over aerial image of the bridge (Bridge Inventory number 1044659 in New York). Figure 64b shows a clear image which is the real size of part of the bridge image in Figure 64a. From this high resolution image, some damages of the bridges on the top of the deck can be identified. Figure 64b shows zoomed high resolution aerial photo and the satellite image (Figure 65) of the same bridge; clearly, the fly-over aerial photo has higher resolution than the satellite image.

a). The high resolution aerial photo.



b). Zoomed in image

Figure 64: The high resolution aerial photo of bridge (NY Bridge # 1044659)

Figure 65: The satellite image of the same area as in Figure 64b.

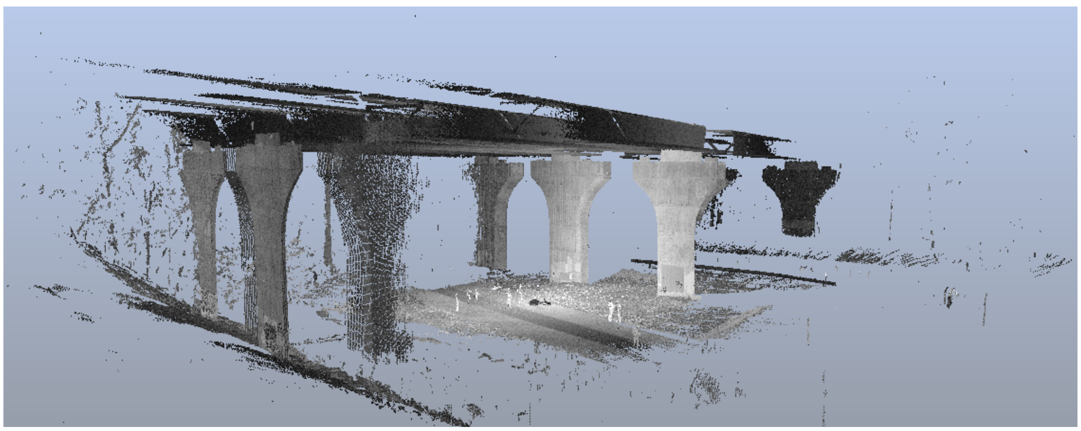
(NY Bridge # 1044659)

6.2.2 LiDAR Scan

LiDAR (Light Detection And Ranging) is a terrestrial 3D laser scanners with a shorter wavelength with an optical remote sensing technology. The LiDAR scanner can collect the surrounding surface information of object by rotating 360 degrees horizontally. The point cloud of the object surrounding surface information can be measured and recorded in a single scan.

Figure 66a shows the LiDAR scan and the Figure 66b shows the 3D scene view of the LiDAR scan (The bridge is located in New York and the bridge ID is 1044659). The two images are generated by FARO SCENE LT (FARO, 2012).

a) LiDAR scan



b) 3D view of LiDAR scan

Figure 66: (a) LiDAR scan of bridge (NY Bridge # 1044659) and (b) 3D scene view

6.3 Implementation of Applying Remote Sensing Data in Bridge-WGI

The Web-GIS based visualization platform integrates the NBI bridge database and remote sensing image data for bridge information visualization and reporting. The goal of integration of remote sensing image with NBI bridge data is to assist national bridge monitoring and management, making it a robust and inclusive data collection system.

Two modules are developed for current remote sensing data application. One application is the remote sensing data uploading to the server database system. The other application is to view the remote sensing data. This remote sensing database can help the bridge manager maintain the remote sensing data.

Since the remote sensing data includes various format, such as image file (fly-over aerial images), LiDAR scan file (the .fls, and .fws document), and possible other document (zip file), and bridge analysis report (doc file), all those files will store in the file system in Web server. In addition, considering the file size of LiDAR scan is up to several hundred megabye (MB), the capacity of PHP uploading and the storage of the file system should be carefully established.

The operation of remote sensing data uploading is illustrated in Figure 67. Before uploading the data, the related bridge first needs to be located. There are three approaches to locate the bridge, including center (latitude and longitude) and radium, bridge ID, or bridge geospatial locations (latitude and longitude). Figure 68 shows the interface for uploading remote sensing data.

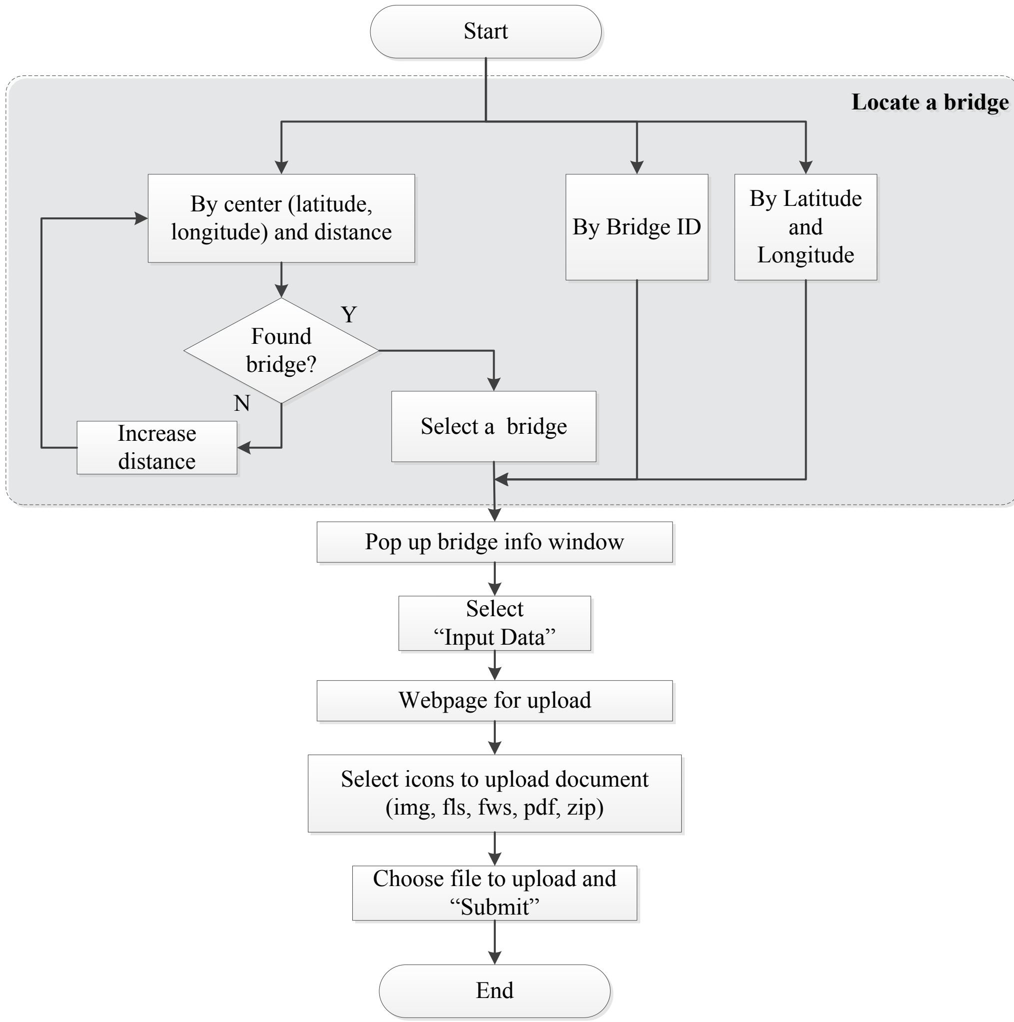
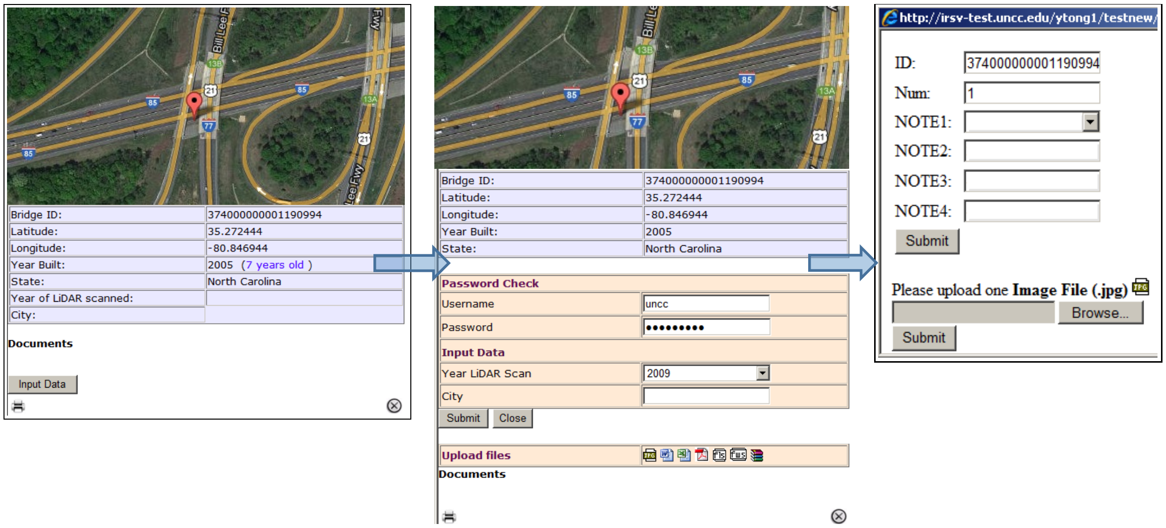


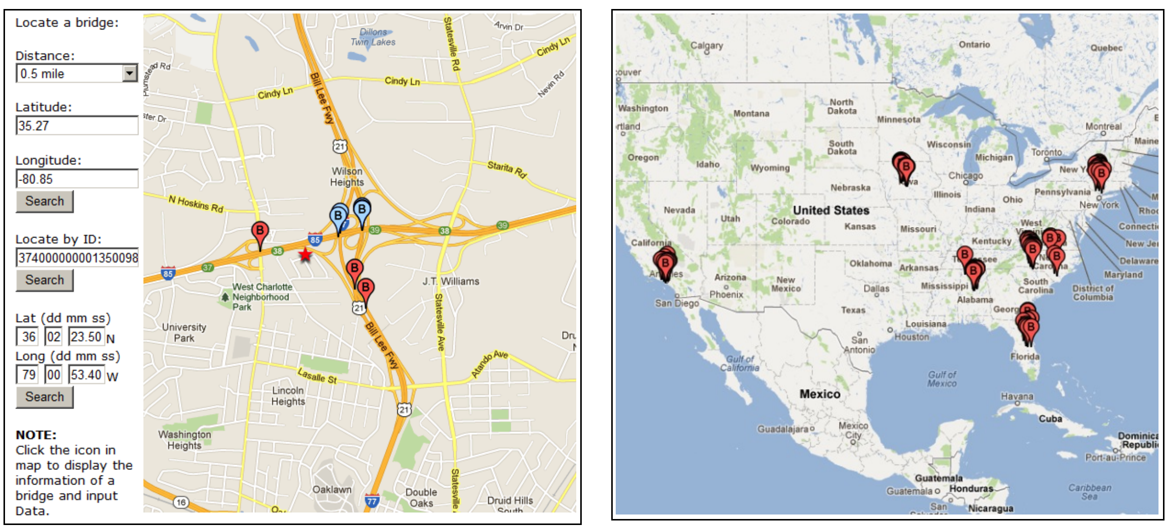
Figure 67: Flowchart for uploading remote sensing data

Figure 69 shows screenshots of main interface of remote sensing data input and data view. Currently the remote sensing data in the database include both LiDAR scan data and fly-over aerial image.



a) Bridge information b) Interface for data input c) Data uploading

Figure 68: Remote sensing data uploading



a) Screenshot for remote sensing data input b) Screenshot for remote sensing data view

Figure 69: Main interface for remote sensing data application

Figure 70 shows the process for viewing the remote sensing data. If a bridge is selected, the information page of this bridge is popped up as shown in Figure 71. The information includes the general information and several remote sensing data, such as the bridge analysis report (the .doc document), the fly-over images, the LiDAR scan data (the .fls, and .fws documents) (Figure 71). By clicking the fly-over image in the bridge information page, the high resolution image will be displayed in a new page (Figure 72). With this high-resolution image, the bridge manager can manipulate the resolution by zoom operation (Figure 72b).

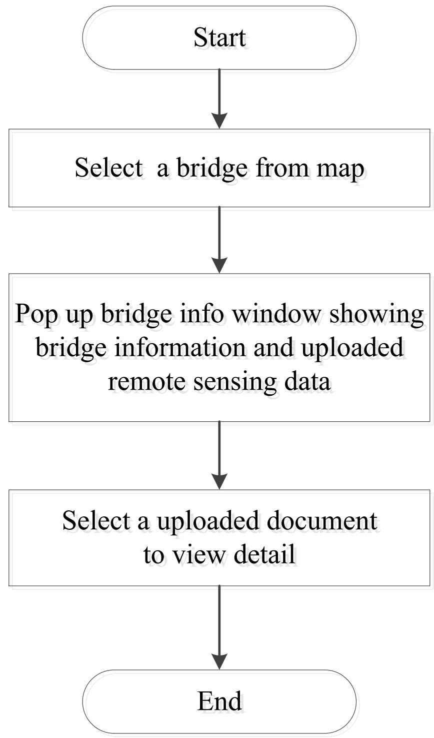


Figure 70: Flowchart for viewing remote sensing data

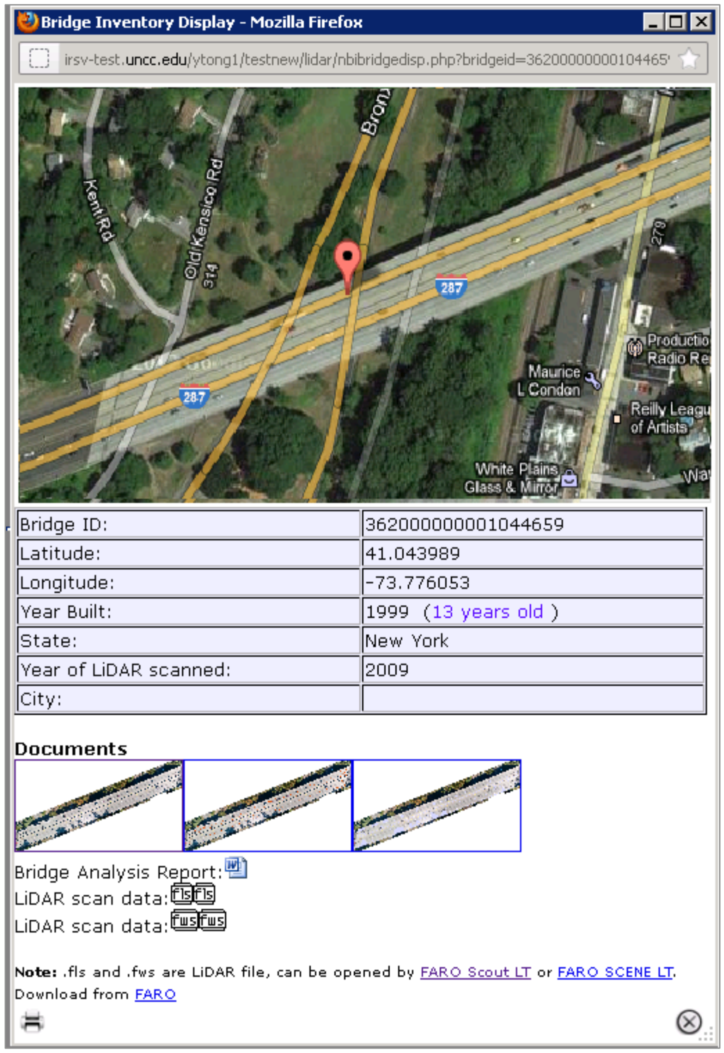


Figure 71: The bridge information with remote sensing data (NY Bridge # 1044659)

a) The high resolution fly-over image of the bridge

b) The real size of the fly-over aerial image

Figure 72: The fly-over image of the bridge (NY Bridge # 1044659)

The bridge analysis report (Figure 71) shows more detailed information of the bridge, which includes the bridge inventory ID and the geographic location, are important parameters for locating the bridge; other bridge information such as the responsible agency and the year built.

Figure 73: The bridge report (NY Bridge # 1044659)

6.4 Development of 3D View Tool for LiDAR Scan

One advantage of using LiDAR scan is that there is a 3D model of the structure (bridge) that can be saved as permanent record for future retrieval. In addition, a series of scans from different positions can be connected into one global coordinate system, such that a full 3D model can be generated for a large structure. Furthermore, LiDAR data processing is significantly simpler than photo image processing. Since LiDAR output is 3D position data, it can be saved in text format and can be preserved longer than image data, hence, it is the preferred tool of analysis for future applications. This flexibility in analysis makes LiDAR an extremely useful tool for analyzing structures (Tong et al., 2010)

A 3D visualization tool for LiDAR scan (LiV3d) is developed to provide 3D interaction such as rotation and zoom function. Based on algorithms of Liv3d, a Web-based 3D visualization tool is suggested for future study. Figure 74 shows the 3D visualization tool development for LiDAR.

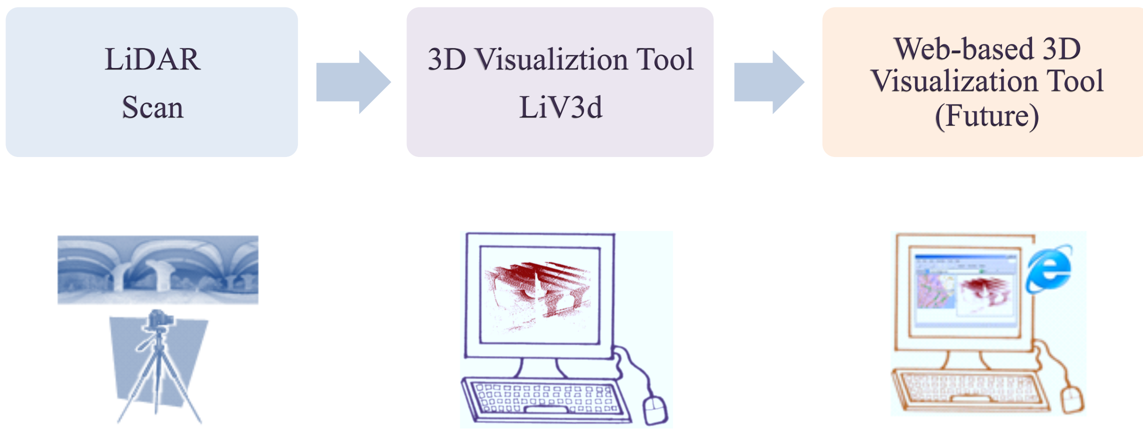


Figure 74: Flowchart for 3D visualization tool development

This proposed 3D visualization tool for LiDAR scan is developed with C++, OpenGL and FLTK programming platform. The main algorithm is as shown in Figure 75.

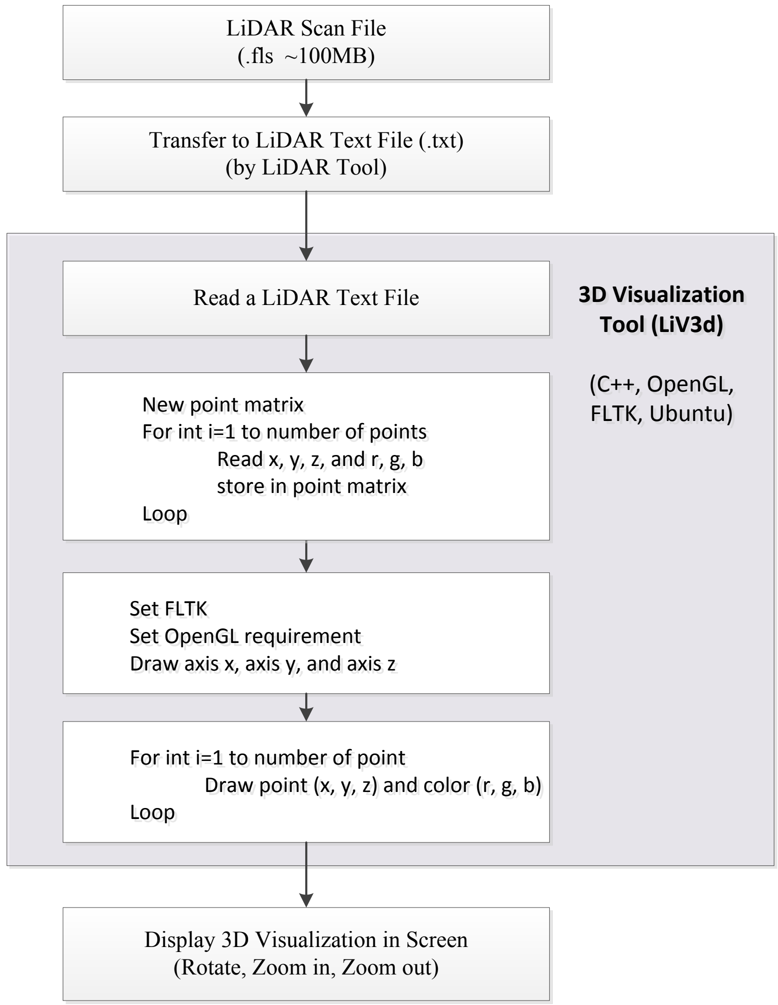
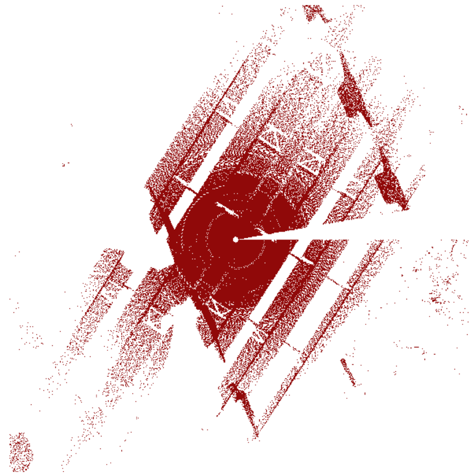


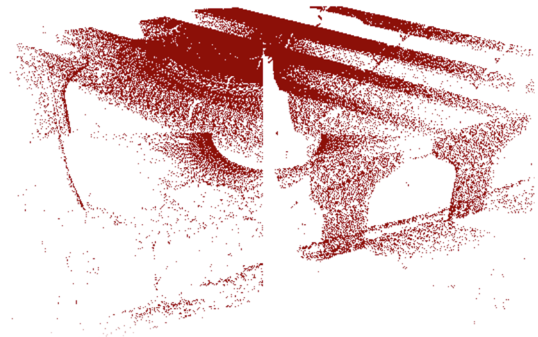
Figure 75: Flowchart for algorithm of 3D visualization tool for LiDAR scan (LiV3d)

The selected bridge in this study is the bridge #1044690 in New York. Figure 63 shows the location of the bridge in Google Map. Figure 66 shows the LiDAR scan, which is taken by LiDAR scanner from 360-angle.

By using the LiV3d, the LiDAR scan file is imported and processed; and the result displays in a graphic user interface. LiV3d has functionalities of rotation, zoom in and zoom out, user can rotate/zoom-in/zoom-out the 3D graph (Figure 76). In Figure 76a, the center of the circles is the location where the LiDAR scanner located when the scan captured. Figure 76a is a top view of this 3D graph and Figure 76b shows the rotated 3D view of the same scan data.



a). Top view



b). Side View

Figure 76: Bridge 3D visualization with LiV3d (NY Bridge # 1044659)

6.5 Summary

Remote sensing database applications are discussed in this chapter. The following key points are highlighted:

- Remote sensing data uploading and data viewing developed,
- 3D interactive visualization tool is developed for viewing the LiDAR scan,
- 3D Web-based visualization tool for LiDAR scan is suggested for future study,

CHAPTER 7: DISCUSSIONS

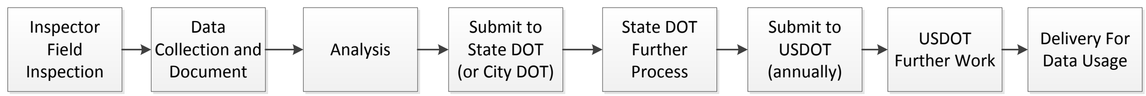
The primary contribution of this research is to demonstrate that a Web-based BMS can integrate several useful functions including: 1) applying voluntary sensing data to increase bridge database and fill the gap of bridge inspection -The extra data can increase both the category and the amount of the NBI bridge condition data drastically; 2) Web-based software offers flexible accessibility at anytime and any place; 3) visualization analytics enhance understanding of the bridge conditions; 4) integrate various bridge data via Internet; and 5) user-defined criteria query for flexible data visualization.

7.1 Benefits

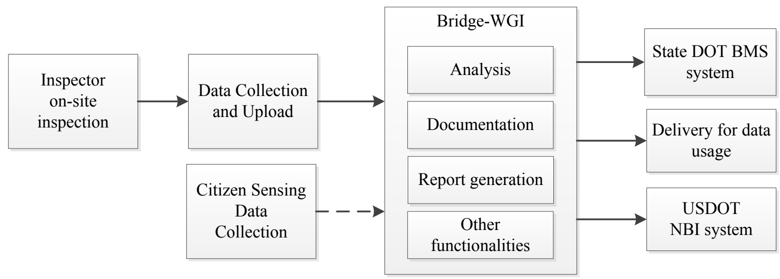
Compare to the conventional procedure of inspection and bridge management (Figure 77a) which is a serial multi-task procedure, Bridge-WGI offers a parallel multi-functionality procedure (Figure 77b) which can improve bridge data management. Other benefits associated with the developed Bridge-WGI framework can be summarized as follows:

(1) Bridge-WGI with availability of Internet access can potentially save both time and cost during bridge inspection and data processing, i.e. data access on portable devices.

(2) Bridge-GWI allows DOTs to share bridge data and do on-line analysis collaboration.



a) Existing procedure for bridge inspection



b) Bridge-WGI procedure for bridge inspection

Figure 77: Comparison between existing procedure and bridge-WGI procedure for bridge inspection

(3) Bridge-WGI decreases the human errors and increases the reliability of bridge data. Since Bridge-WGI makes a one-step entry, the inspector is the first and only person who collects and enters the data, no other person involved in the data entry.

(4) Bridge-WGI can be potentially used for public safety - citizen sensor can report safety related issues to government and obtain safety information from government.

(5) Bridge-WGI is a framework with easy-upgrade feature for future maintenance, such as adding a module, modifying a module, eliminating a module, or upgrading an interface.

7.2 Limitations

Citizen sensing data can expand bridge database and help bridge manager to improve the inspection capability. However, several limitations should be considered, such as capability of citizen sensor and security issues. Since the volunteers are not as professional as the bridge inspectors, the citizen sensor data may be limited in accuracy and relevancy. Therefore the citizen sensing data can at best play a reference role. In addition, education and training may be needed for improving the sensing capability of citizen sensor to make the sensing data more accurate.

Lack of Internet service will be a challenge when volunteers try to report damages. A possible solution is to design a functionality to store the damage information and the volunteer can do the report later when Internet is available.

Generally speaking, database security is not a problem because MySQL database is capable of multiple level of security. Current Bridge-WGI provides two levels of security: The first level, allows several privileges for different user privileges to access the application. For example, the public (citizen sensor) may be granted limited privileges to view partial data on the database. At the second level, each application that accesses the bridge data has application controls that restrict database actions.

The application of citizen sensor can improve the performance of bridge management when critical damage occurs. However, if volunteer sensor reports wrong information, extra work and additional expenses may be associated. Data mining techniques may be helpful in sieving reliable data from citizen sensing data.

Since the file size of remote sensing data and citizen sensing data is large (up to several hundred MB), it is a challenge to perform online computing and analysis in a Web-based application. For the visualization, rendering speed is a challenge; the processing speed mostly depends on the processor of the computer.

7.3 Implementation Considerations

Bridge safety is an important issue for the society; bridge management and bridge maintenance play very important roles for bridge safety. Therefore, both the quantitative and the qualitative of bridge condition data are critical for bridge management.

Citizen sensor represents a critical adaptation of the geography and spatial science (GSS) technologies that have extensive applications in infrastructure monitoring. Almost every recent disaster event that has occurred in the US over the last five years, including the World Trade Center and Pentagon terrorist attacks, hurricane Katrina and more recently, the Louisiana Oil Spill, have been geo-referenced, spatially-analyzed, and displayed using aerial imaging and thematic satellite imaging. Notably, the images of the migration of hurricane Katrina have been used in great detailed temporal demonstrations in geo-referenced contexts (Scharroo et al. 2005).

Coupled with the availability of cell phone cameras and easy-access Internet, one can now view disaster pictures of every major event in the world beyond national boundaries. The willingness to do so is clearly there, as is the technology to integrate their inputs.

Regardless of organized or non-organized citizen sensor systems, the key issues are the accuracy and relevancy of the collected data. Organized citizen sensor should allow feedbacks to help improve on sensor accuracy.

Appropriate citizen sensing application not only can increase safety and decrease cost, but also can help the public to understand national bridge system and increase their social responsibility and awareness of infrastructure and environment issues. With proper training and organization, citizen sensing applications provide great opportunities for the public to act their responsibility for infrastructure and environmental safety.

The central idea of Bridge-WGI is the adoption of Internet technologies, which makes it possible to view bridge data at any time and at any place; this flexibility in accessing bridge data helps the bridge managers to maintain bridge with the potentially parallel and shared processing of bridge data. With the advances in Web video technology, it is envisioned that bridge mangers and inspectors can have real time video conferencing about specific bridge problems and shared data visualization while the inspectors are in the field.

Visualization technologies have the potential to enhance the understanding of the bridge condition and allow users to discover more insightful information. By integrating the remote sensing data, more analytical factors can be considered for bridge safety analysis.

However, this also implies the necessity of investments into IT hardware. As discussed in earlier chapters, the inclusion of imagery and audio-video data can result in exponential increase in bridge data. Hence, Web engines, disk space, Internet connection portals, high-density video monitors, are the hardwares that need to be invested and maintained. Such investment may be difficult for some DOTs and DOTs need to justify the expenditure against potential user cost savings and increased safety level. With the increasing costs to maintain the national bridge infrastructure ($930 billion dollars over 5 years, ASCE 2009), this investment can be easily justified.

Finally, Bridge-WGI only represents the tip of the iceberg for modern Internet technology to implement infrastructure management: There is the futuristic potential for the virtualization of BMS, so that bridge management can automate bridge life cycle cost projection for every alternate maintenance decision.

7.4 Data Accuracy Issues

The fundamental issue with the data in this research is accuracy. Accuracy is the closeness of the results between the observed value and the true value. For current study, there are two key types of data accuracy. One is the positional accuracy and the other is attribute accuracy (Buckey, 2012). For positional accuracy, especially for the bridge geospatial location, there is possible error between the measured geospatial information (latitude and longitude) and the true ground location. Some errors are very easy to be detected, such as the geospatial location may be out of the boundary of the United States. Some errors may not be easily figured out. These errors may be caused during data entry either by human or by automated process (software error). The geospatial view using Web-GIS can validate bridge location data.

The second error is related to the actual bridge information (attribute). The data accuracy issue is more challenging to identify. Source of the error may be due to the inspector making errors during data entry. For instance, the deck condition should be rated as “9”, while the inspector may enter a “6” by typing the wrong key. Algorithms for detecting the error and increasing the data accuracy need to be developed, for example, regression model can be developed to detect data errors.

In this research, the accuracy of citizen sensing data is very important. Since the citizen sensors are not professionals, some of the data may have high possibility of inaccuracy, such as the description of the damage is not precise, the uploaded picture may be completely useless. The system should have the capability to take action to eliminate the data or improve the data. Data validation algorithms should be developed for detecting the citizen sensing data.

CHAPTER 8: CONCLUSIONS

US National Bridge Inventory consists of near 600,000 bridges spreading over the country. To enhance the management of these bridges, by integrating Internet, GIS, visualization, remote sensing and citizen sensing techniques makes it possible to enhance current bridge management and maintenance via the proposed Bridge-WGI framework. Bridge-WGI represents the establishment of a Web-GIS based database with analytical visualization and distributed sensing capabilities.

The developed framework Bridge-WGI consist of 6 modules: 1) database systems, 2) bridge general information visualization, 3) bridge data analytical visualization, 4) user-defined criteria query for decision-making support, 5) citizen sensor application, and 6) remote sensing data application. The conclusions of this dissertation are summarized as follows:

(1). The Bridge-WGI is developed to enhance bridge management and bridge maintenance. The framework provides a multi-functionality application which can assist bridge managers and engineers to access the bridge information with centralized framework that contains the national bridge data and other bridge information.

(2). The Bridge-WGI consists of three databases: The primary database for this framework contains all the national bridge historical data for all 50 states and Washington, D.C.; the second database is designed for citizen sensing data which collected and uploaded by volunteers; the third database is for collected remote sensing data including the LiDAR scan data and fly-over aerial images.

(3). Internet and GIS technologies are integrated into Bridge-WGI to implement a more efficient and more effective approach to represent the geospatial data visualization. Internet technology makes it possible to view bridge information at any time and any place. GIS technology has the capability to display the bridge by geographic location and further geospatial analysis can be performed on the bridge data.

(4). Visualization technology provides a new way to interpret the bridge condition information and the distribution of the bridge condition rating at different scale of interests (nationwide or statewide). The applied analytical visualization technologies can make better presentations of bridge data so that bridge managers can obtain a clear knowledge of the bridge conditions. Analytical visualization technologies allow managers to generate observations about the bridge conditions. Several analytics visualization techniques are applied such as parallel coordinates, scatter plot, and choropleth map.

(5). User-defined criteria query module provides alternative approach to existing bridge evaluation. Higher-level analysts can use this tool to generate complex criteria query for bridge analysis.

(6). Citizen sensor application module provides the system for volunteers (citizen sensors) to report bridge damage with their willingness. Two types of application including Web-based apps and smartphone apps have been developed. Citizen sensing data will expand the bridge database drastically and can potentially fill in the data gap in annual bridge inspection cycles.

(7). Remote sensing application module provides the platform for professionals to maintain the remote sensing data. Remote sensing data, including LiDAR scans and fly-over aerial images, expands the NBI bridge data and can be integrated with bridge database for further analysis. Bridge-WGI can assist bridge managers to conduct bridge management and maintenance with remote sensing data. A visualization tool for LiDAR scan is been developed.

(8). Bridge-WGI necessarily means a critical investment from the DOT, which can be leveraged by the potential savings in improving bridge conditions and reduced maintenance/repair expenditures.

CHAPTER 9: RECOMMENDATION FOR FUTURE STUDIES

The developed framework Bridge-WGI has the potential to enhance the national bridge infrastructure management. Future extensions of the conceptual model can provide further process support mechanisms. Considering the challenge issues and the feasibility of flexible upgrading, to make Bridge-WGI a powerful framework for bridge management, the future research topics are projected as following.

(1). More precise decision-making models can be integrated on this framework. Some technologies such as data mining, predictive analytics, and artificial intelligence (AI) could be applied for further research on bridge management.

(2). Damage detection module should be created to detect and localize the damage from imagery data. Algorithms and tools could be developed to automate the processes to retrieve damage information from remote sensing data and citizen sensing data.

(3). Web-based 3D visualization tool for LiDAR scan can be developed and integrated into Bridge-WGI.

(4). Web-based bridge damage detection algorithms for remote sensing image, the functionalities should include damage detection, damage localization, damage counting, and damage visualization. The related damage information can be integrated with the bridge data to assist bridge management and maintenance.

(5). Algorithms and tools should be developed to detect poor or irrelevant information from massive citizen sensing data.

(6). Apply additional computer vision technology to detect damage from bridge imagery data.

Bridge-WGI has the capability to be flexibly upgraded and modified with current technologies; however, with the development of technologies, several emerging technologies should be considered in the further research, including (1) the effects of graphic fluidity and visualization effect on current Bridge-WGI and citizen sensor integration; (2) since Windows 8 is an open shared OS platform for tablet, smartphone and PC, the impact to Bridge-WGI of Window 8 should be considered; (3) how to update the system to a newer version of operation system such as Window8 and new technologies such as HTML5 with less work should be considered; (4) Future applications in high performance computing technology such as cloud computing should be considered; (5) consideration of upgrade Bridge-WGI to any potential future directions of bridge management software; and (6) case-based reasoning techniques should be considered to use in bridge safety problem solving.

REFERENCES

AASHTO. 2011. AASHTO Guide Manual for Bridge Element Inspection. AASHTO, Washington, D.C..

Abudayyeh, O., Bataineh, M.A., Abdel-Qader, I. 2004. An Imaging Data Model for Concrete Bridge Inspection. Advances in Engineering Software 35(8-9): 473-480.

Ahlborn, T. M., Shuchman, R., Sutter, L. L., Brooks, C. N., Harris, D. K., Burns, J. W., Endsley, K. A., Evans, D. C., Vaghefi, K., and Oats, R. C.. 2010. The State-of-the Practice of Modern Structural Health Monitoring for Bridges: A Comprehensive Review. Michigan Tech Research Institute.

Aktan, A.E., Farhey, D.N., Helmicki, A.J., Brown, D.L., Hunt, V.J., Lee, K.L. and Levi, A. 1997. Structural Identification for Condition Assessment: Experimental Arts, Journal of Structural Engineering, 123(12):1674-1686.

ASCE. 2009. 2009 Report Card for America's Infrastructure. American Society of Civil Engineers. <http://www.infrastructurereportcard.org/fact-sheet/bridges> (Accessed Aug. 20, 2012).

Atzeni, P., Ceri, S., Paraboschi, S., and Torlone, R. 1999. Database Systems. New York: McGraw-Hill.

Basic4Android. 2011. Basic4android: An IDE for Android", <http://slapstart.com/2011/04/basic4android/>. (Accessed Aug. 20, 2012).

Batty, M. and Xie Y. 1994. Urban Analysis in a GIS Environment: Population Density Modeling Using ARC/INFO. Spatial Analysis and GIS, edited by A.S. Fotheringham and P.A. Rogerson, Taylor and Francis, London, 189-219.

Buckey, J. D. 2012. BGIS introduction to GIS, <http://bgis.sanbi.org/gis-primer/>. (Accessed Oct. 10, 2012).

Burke, J., Estrin, D., Hansen,M., Parker, A., Ramanathan,N., Reddy, S. and Srivastava, M. 2006. Participatory Sensing. In World Sensor Web Workshop. Boulder, Colorado.

Burdziej, J. 2012. A Web-Based Spatial Decision Support System for Accessibility Analysis—Concepts and Methods. Applied Geomatics. 4(2): 75-84.

Chae, M. J., Yoo, H. S., Kim, J. Y. and Cho, M. Y. 2012. Development of a Wireless Sensor Network System for Suspension Bridge Health Monitoring. Automation in Construction, 21: 237-252.

Chase, S. B. 1995. NDE for Steel Bridges. Civil Engineering. 65(5): 49-51.

Chase, S. B., Small, E. P. and Nutakor, C. 1999. An In-Depth Analysis of the National Bridge Inventory Database Utilizing Data Mining, GIS and Advanced Statistical Methods Transportation Research Circular No.498: Proceedings of the 8th International Bridge Management Conference (1999), Denver, CO:C-6/1-C-6/17.

Chaudhuri, S. and Shim, K. 1999. Optimization of Queries with User-Defined Predicates. ATM transactions on Database Systems. 24(2): 177-228.

Chen, S. E. Rice, C. Boyle, C. and Hauser, E. 2011. Small Format Aerial Photography for Highway Bridge Monitoring. ASCE Journal of Performance of Constructed Facilities.25(2): 105-112.

Chen, S. E., Wang, X., Vatcha, R., Dou, W., Lee, S., Chang, R., Hauser, E., Tolone, W. L. and Ribarsky.W. 2010. IRSV, an Integrated Visual Analytics Bridge Management System. TRB 89th Annual Meeting Compendium, Transportation Research Board, Washington, D.C.

Christin, D., Reinhardt, A., Kanhere, S. S., and Hollick, M. 2011. A Survey on Privacy in Mobile Participatory Sensing Applications. Journal of Systems and Software. 84:1928-1946.

D3, 2012. Data-Driven Documents. <http://d3js.org/> (Accessed Aug. 25, 2012)

Dabous, A. and Alkass, S. 2010. A Stochastic Method for Condition Rating of Concrete Bridges. ASCE, Construction Research Congress 2010: Innovation for Reshaping Construction Practice. 558-567.

Dragicevic, S. 2004. The potential of Web-Based GIS. Journal of Geographical Systems, 6(2): 79–81.

Elbehairy, H. 2007. Bridge Management System with Integrated Life Cycle Cost Optimization. Ph.D. Dissertation. University of Waterloo, Ontario, Canada.

Esch, G., Scott, M. and Zhang, E. 2009. Graphical 3D Visualization of Highway Bridge Ratings. Journal of Computing in Civil Engineering. 23(6):355-362.

Estrin, D. 2010. Participatory Sensing: Applications and Architecture. Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, San Francisco, CA: 3–4.

FARO 2012. FARO Laser Scanner Software. <http://www.faro.com/focus/us/software>. (Accessed Feb. 12, 2012).

FHWA. 1995. Recording and Coding Guide for the Structure Inventory and Appraisal of the National Bridges, Report FHWA-PD-96-001, U.S. Dept. of Transportation, Washington, D.C.

FHWA. 2011. Bridge Programs NBI Data Tables of Frequently Requested NBI Information. Federal Highway Administration. < http://www.fhwa.dot.gov/bridge/fc.cfm > (Accessed Jan. 15, 2012).

Gastineau, A., Johnson, T. and Schultz, A. E. 2009. Bridge Health Monitoring and Inspections - A Survey of Methods, Transportation Research, Technical report MN/RC2009-29. Minnesota Department of Transportation.

Glantz, P., Nilsson, E. D. and Hoyningen-Huene, W. 2009. Estimating a Relationship between Aerosol Optical Thickness and Surface Wind Speed over the Ocean. Atmospheric Research. 92(1): 58-68.

GoodChild, M. F. 2007a. Citizens as Sensors: The World of Volunteered Geography. GeoJournal. 69(4):211–221.

GooDChild, M. F. 2007b. Citizens as Sensors: Web 2.0 and the Volunteering of Geographic Information. GeoFocus Editorial, 7: 8-10.

GoogleMap. 2011. Google MapAPI. <http://code.google.com/apis/maps/documentation/javascript>. (Accessed Dec. 22, 2011).

GoogleMap, 2012. Google Maps. <http://en.wikipedia.org/wiki/Google_Maps> (Accessed Jan. 01, 2012).

Gucunski, N., Consolazio, G. R. and Maher, A. 2006. Concrete Bridge Deck Delamination Detection by Integrated Ultrasonic Methods. International Journal of Material and Product Technology, Special Issue on NDT and Failure Preventive Technology. 26(1/2):19-34.

Hawk, H. 1999. BRIDGIT: User-friendly Approach to Bridge Management. Eighth Transportation Research Board Conference on Bridge Management, TRB Transportation Research Circular 498, Denver, CO:E-7/1-E7/15.

Hellerstein, J. M., Stonebraker, M. and Hamilton, J. 2007. Architecture of a Database System, Foundations and Trends in Databases, 1(2):141-259.

Huang, B. 2003. Web-Based Dynamic and Interactive Environmental Visualization Computers, Environment and Urban Systems. 27(6):623-636.

InSiteful Imagery. 2011. InSiteful Imagery. <http://www.insitefulimagery.com/about.htm> (Accessed Dec. 20, 2011).

Iyengar, A. and Challenger, J. 1997. Improving Web Server Performance by Caching Dynamic Data, Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems. Monterey, California. 5-5.

Jivacate, I. and Najafi, F. T. 2003. The Current Status of Bridge Management Systems. Proceedings of Canadian Society for Civil Engineering Annual Conference, Moncton, NB, Canada: 1622-1631.

Karlaftis, M., Kepaptsoglou, K. and Lambropoulos, S. 2005. A Web-Supported National Bridge Inventory Management Tool. Public Works Management & Policy. 9(3):248-258.

Kingston, R., Carver, S., Evans, A. and Turton, I. 2000. Web-Based Public Participation Geographical Information Systems: an Aid to Local Environmental Decision-Making. Computers, Environment and Urban Systems, 24(2):109-125.

Liang, K., Monger, P. and Couchman, H. 2005. Interactive Parallel Visualization of Large Particle Datasets. Parallel Computing. 31(2): 243-260.

Liu, W. 2010. Terrestrial LiDAR-bases Bridge Evaluation. Ph.D. dissertation. University of North Carolina at Charlotte, North Carolina.

Michalos, M., Tselenti, P. and Nalmpantis, S. L. 2012. Visualization Techniques for Large Datasets. Journal of Engineering Science and Technology Review. 5(1): 72-76.

Mobile. 2012. Mobile Application Development . <http://en.wikipedia.org/wiki/Mobile_application_development>. (Accessed, Aug. 25, 2012).

Nagarajan, M., Gomadam, K., Sheth, A. P., Ranabahu, A., Mutharaju, R. and Jadhav, A. 2009. Spatio-Temporal-Thematic Analysis of Citizen Sensor Data: Challenges and Experiences. Tenth International Conference on Web Information Systems Engineering, PoznaD, Poland, 539-553.

Nagayama, T., Moinzadeh, P., Mechitov, K., Ushita, M., Makihata, N., Ieiri, M., Agha, G., Spensor, B. F., Fujino, Y. and Seo, J.W. 2010. Reliable Multi-Hop Communication for Structural Health Monitoring, Smart Structures and Systems, 6(5): 481-504.

Navalgund, R. R., Jayaraman, V., and Roy, P. S. 2007. Remote Sensing Applications: An Overview. Current Science. 93(12): 1747–1766.

OGC. 2012. Web Map Service. Open Geospatial Consortium. <http://www.opengeospatial.org/standards/wms> (Accessed Jan. 20, 2012).

OSM. 2012. OpenStreetMap. <http://en.wikipedia.org/wiki/OpenStreetMap>. (Accessed Aug. 20, 2012).

Peeters, B. and Ventura, C. E. 2003. Comparative Study of Modal Analysis Techniques for Bridge Dynamic Characteristics, Mechanical Systems and Signal Processing. 17(5):965-988.

PHP. 2012. What is PHP? <http://www.php.net/>. (Accessed, Feb. 20, 2012).

Roberts, J. E., and Shepard, D. 2000. Bridge Management for the 21st Century, Transportation Research Record. 1696:197-203.

Robert, W. E., Marshall, A. R., Shepard, R. W. and Aldayuz, J. 2003. Pontis Bridge Management System State of the Practice in Implementation and Development. 9th International Bridge Management Conference, Transportation Research Board, Orlando, FL. 49-60.

Rossow, M. 2012. Overview of Bridge Inspection Programs (BIRM). Course No: S01-001 <http://www.cedengineering.com/upload/Bridge%20Inspection%20Programs.pdf> (Accessed Aug. 16, 2012).

Ryan, T. W., Hartle, R. A., Mann, J. E. and Danovich, L. J. 2006. Bridge Inspector's Reference Manual. FHWA NHI 03-001.

Sack, D. A. and Olson, L. D. 1995. Advanced NDT Methods for Evaluating Concrete Bridges and Other Structures, NDT and E International, 28(6):349-357.

Saito, M. 1988. Development of a statewide network level bridge management system. PhD Dissertation, Purdue University, West Lafayette, IN.

Scharl, A. and Tochtermann, K. 2007. The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society. Springer, London.

Scharroo, R., Smith, W. H. F. and Lillibridge, J. L. 2005. Satellite Altimetry and the Intensification of Hurricane Katrina. EOS, Transactions American Geophysical Union, 86(40):366.

Schowengerdt, R. A. 2010. Remote Sensing: Models and Methods for Image Processing 3rd edition, Elsevier Inc.

She, T. H., Aouad, G. and Sarshar, M. 1999. A Geographic Information System (GIS)–Based Bridge Management System. Computer-Aided Civil and Infrastructure Engineering, 14(6): 417–427.

Sheth, A. 2009. Citizen Sensing, Social Signals, and Enriching Human Experience. IEEE Internet Computing. 13:87–92.

Shi, W. Z., Ko, J. M., Zhao, C. Y., and Chen, L. 2005. A Bridge Structural Health Monitoring Information System Based on GIS and DBMS. Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, SPIE, Bellingham, WA, 1012-1020.

Shoukry, S. N., Riad, M. Y. and William, G. W. 2009. Long Term Sensor-Based Monitoring of an LRFD Designed Steel Girder Bridge. Engineering Structures, 31(12), 2954-2965.

Singh, I. and Palmieri, M. 2012. Comparison of Cross-platform Mobile Development Tools <http://www.idt.mdh.se/kurser/ct3340/ht12/MINICONFERENCE/FinalPapers/ircse11_submission_16.pdf> (Accessed Aug. 16, 2012).

Skarlatidou, A, Haklay M., and Cheng T. 2011. Trust in Web GIS: The Role of The Trustee Attributes in the Design of Trustworthy Web GIS Applications. International Journal of GIScience. 25(12):1913–1930.

Stein, A. 2011. Bridge Locator Development, University of North Carolina at Charlotte, Charlotte, NC.

Stone, W. C., Juberts, M., Dagalakis, N., Stone, J. and Gorman, J. 2004. Performance Analysis of Next-Generation LADAR for Manufacturing, Construction, and Mobility, Report NISTIR 7117, National Institute of Standards and Testing, U.S. Department of Commerce.

Tatu, A., Albuquerque, G., Eisemann, M., Schneidewind, J., Theisel, H., Magnor, M. and Keim, D. 2009. Combining Automated Analysis and Visualization Techniques for Effective Exploration of High Dimensional Data. Proceedings of IEEE Symposium on Visual Analytics Science and Technology. Washington: IEEE Computer Society :59-66.

Tong, Y., Bian, H., Watson, C., Dai, K., Liu, W., and Chen, S. E., 2010. LiDAR Scan Applications for Bridge Condition Validation Post-Blast Load and Construction. NDE/NDT for Highways and Bridges: Structural Materials Technology SMT 2010, New York, NY.

Wang, X., Dou, W., Chen, S. E., Ribarsky, W. and Chang, R. 2010. An Interactive Visual Analytics System for Bridge Management. Computer Graphics Forum. 29(3):1033-1042.

Washer, G. A. 1998. Developments for the Non-Destructive Evaluation of Highway Bridges in the USA, NDT & E International, 31(4): 245-249.

Zhang, S., Shen, W. and Ghenniwa, H. 2004. A Review of Internet-Based Product Information Sharing and Visualization. Computers in Industry. 54(1):1-15.

Zhu, B. and Chen, H. 2005. Information Visualization. Annual Review of Information Science and Technology. 39(1): 139-177.

APPENDIX A: NBIS CODE FOR BRIDGE CONDITION RATING (DECK, SUPERSTRUCTURE AND SUBSTRUCTURE)

Code	Description
9	EXCELLENT CONDITION
8	VERY GOOD CONDITION - no problems noted.
7	GOOD CONDITION - some minor problems.
6	SATISFACTORY CONDITION - structural elements show some minor deterioration
5	FAIR CONDITION - all primary structural elements are sound but may have minor section loss, cracking, spalling or scour.
4	POOR CONDITION - advanced section loss, deterioration, spalling or scour.
3	SERIOUS CONDITION - loss of section, deterioration, spalling or scour have seriously affected primary structural components Local failures are possible. Fatigue cracks in steel or shear. cracks in concrete may be present.
2	CRITICAL CONDITION - advanced deterioration of primary structural elements. Fatigue cracks in steel or shear cracks in concrete may be present or scour may have removed substructure support. Unless closely monitored it may be necessary to close the bridge until corrective action is taken.
1	IMMINENT FAILURE CONDITION - major deterioration or section loss present in critical structural components or obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic but corrective action may put back in light service.
0	FAILED CONDITION - out of service - beyond corrective action.

APPENDIX B: NBI RECORD FORMAT

(http://www.fhwa.dot.gov/bridge/nbi/format.cfm)

With the conversion to metric and the addition of new items it is required to expand the size of the NBI record to 432 characters.

ITEM NO	ITEM NAME	ITEM POSITION
1	State Code	1-3
8	Structure Number	4-18
5	Inventory Route	19 - 27
5A	Record Type	19
5B	Route Signing Prefix	20
5C	Designated Level of Service	21
5D	Route Number	22 - 26
5E	Directional Suffix	27
2	Highway Agency District	28 - 29
3	County (Parish) Code	30 - 32
4	Place Code	33 - 37
6	Features Intersected	38 - 62
6A	Features Intersected	38 - 61
6B	Critical Facility Indicator	62
7	Facility Carried By Structure	63 - 80
9	Location	81 - 105
10	Inventory Rte, Min Vert Clearance	106 - 109
11	Kilometerpoint	110 - 116
12	Base Highway Network	117
13	Inventory Route, Subroute Number	118 - 129
13A	LRS Inventory Route	118 - 127
13B	Subroute Number	128 - 129
16	Latitude	130 - 137
17	Longitude	138 - 146
19	Bypass/Detour Length	147 - 149
20	Toll	150
21	Maintenance Responsibility	151 - 152
22	Owner	153 - 154
26	Functional Class Of Inventory Rte.	155 - 156
27	Year Built	157 - 160
28	Lanes On/Under Structure	161 - 164
28A	Lanes On Structure	161 - 162
28B	Lanes Under Structure	163 - 164
29	Average Daily Traffic	165 - 170
30	Year Of Average Daily Traffic	171 - 174
31	Design Load	175

ITEM NO	ITEM NAME	ITEM POSITION
32	Approach Roadway Width	176 - 179
33	Bridge Median	180
34	Skew	181 - 182
35	Structure Flared	183
36	Traffic Safety Features	184 - 187
36A	Bridge Railings	184
36B	Transitions	185
36C	Approach Guardrail	186
36D	Approach Guardrail Ends	187
37	Historical significance	188
38	Navigation Control	189
39	Navigation Vertical Clearance	190 - 193
40	Navigation Horizontal Clearance	194 - 198
41	Structure Open/Posted/Closed	199
42	Type Of Service	200 - 201
42A	Type of Service On Bridge	200
42B	Type of Service Under Bridge	201
43	Structure Type, Main	202 - 204
43A	Kind of Material/Design	202
43B	Type of Design/Construction	203 - 204
44	Structure Type, Approach Spans	205 - 207
44A	Kind of Material/Design	205
44B	Type of Design/Construction	206 - 207
45	Number Of Spans In Main Unit	208 - 210
46	Number Of Approach Spans	211 - 214
47	Inventory Rte Total Horz Clearance	215 - 217
48	Length Of Maximum Span	218 - 222
49	Structure Length	223 - 228
50	Curb/Sidewalk Widths	229 - 234
50A	Left Curb/Sidewalk Width	229 - 231
50B	Right Curb/Sidewalk Width	232 - 234
51	Bridge Roadway Width Curb-To-Curb	235 - 238
52	Deck Width, Out-To-Out	239 - 242
53	Min Vert Clear Over Bridge Roadway	243 - 246
54	Minimum Vertical Underclearance	247 - 251
54A	Reference Feature	247
54B	Minimum Vertical Underclearance	248 - 251
55	Min Lateral Underclear On Right	252 - 255
55A	Reference Feature	252
55B	Minimum Lateral Underclearance	253 - 255
56	Min Lateral Underclear On Left	256 - 258
58	Deck	259
59	Superstructure	260

ITEM NO	ITEM NAME	ITEM POSITION
60	Substructure	261
61	Channel/Channel Protection	262
62	Culverts	263
63	Method Used To Determine Operating Rating	264
64	Operating Rating	265 - 267
65	Method Used To Determine Inventory Rating	268
66	Inventory Rating	269 - 271
67	Structural Evaluation	272
68	Deck Geometry	273
69	Underclear, Vertical & Horizontal	274
70	Bridge Posting	275
71	Waterway Adequacy	276
72	Approach Roadway Alignment	277
75	Type of Work	278 - 280
75A	Type of Work Proposed	278 - 279
75B	Work Done By	280
76	Length Of Structure Improvement	281 - 286
90	Inspection Date	287 - 290
91	Designated Inspection Frequency	291 - 292
92	Critical Feature Inspection	293 - 301
92A	Fracture Critical Details	293 - 295
92B	Underwater Inspection	296 - 298
92C	Other Special Inspection	299 - 301
93	Critical Feature Inspection Dates	302 - 313
93A	Fracture Critical Details Date	302 - 305
93B	Underwater Inspection Date	306 - 309
93C	Other Special Inspection Date	310 - 313
94	Bridge Improvement Cost	314 - 319
95	Roadway Improvement Cost	320 - 325
96	Total Project Cost	326 - 331
97	Year Of Improvement Cost Estimate	332 - 335
98	Border Bridge	336 - 340
98A	Neighboring State Code	336 - 338
98B	Percent Responsibility	339 - 340
99	Border Bridge Structure Number	341 - 355
100	STRAHNET Highway Designation	356
101	Parallel Structure Designation	357
102	Direction Of Traffic	358
103	Temporary Structure Designation	359
104	Highway System Of Inventory Route	360
105	Federal Lands Highways	361
106	Year Reconstructed	362 - 365
107	Deck Structure Type	366

ITEM NO	ITEM NAME	ITEM POSITION
108	Wearing Surface/Protective System	367 - 369
108A	Type of Wearing Surface	367
108B	Type of Membrane	368
108C	Deck Protection	369
109	AVERAGE DAILY TRUCK TRAFFIC	370 - 371
110	DESIGNATED NATIONAL NETWORK	372
111	PIER/ABUTMENT PROTECTION	373
112	NBIS BRIDGE LENGTH	374
113	SCOUR CRITICAL BRIDGES	375
114	FUTURE AVERAGE DAILY TRAFFIC	376 - 381
115	YEAR OF FUTURE AVG DAILY TRAFFIC	382 - 385
116	MINIMUM NAVIGATION VERTICAL CLEARANCE VERTICAL LIFT BRIDGE	386 - 389
	FEDERAL AGENCY INDICATOR	391
	Washington Headquarters Use	392 - 426
	STATUS	427
n/a	Asterisk Field in SR	428
SR	SUFFICIENCY RATING (select from last 4 positions only)	429 - 432

Status field:

1 = Structurally Deficient;
2 = Functionally Obsolete;
0 = Not Deficient;
N = Not Applicable

APPENDIX C: DATABASE TABLE STRUCTURE

Table 3-C1: Structure for tables "nbi2008", "nbi2009", "nbi2010", and "nbi2011".

Field	Type	Null	Key	Default	Extra
row_id	int(11)	NO		0	
stfips	char(2)	YES		NULL	
region	char(1)	YES		NULL	
item8	char(15)	YES		NULL	
item5a	char(1)	YES		NULL	
item5b	char(1)	YES		NULL	
item5c	char(1)	YES		NULL	
item5d	char(5)	YES		NULL	
item5e	char(1)	YES		NULL	
item2	char(2)	YES		NULL	
item3	char(3)	YES		NULL	
item4	char(5)	YES		NULL	
item6a	char(24)	YES		NULL	
item6b	char(1)	YES		NULL	
item7	char(18)	YES		NULL	
item9	char(25)	YES		NULL	
item10	char(4)	YES		NULL	
item11	char(7)	YES		NULL	
item12	char(1)	YES		NULL	
item13a	char(10)	YES		NULL	
item13b	char(2)	YES		NULL	
item16	char(8)	YES		NULL	
item17	char(9)	YES		NULL	
item19	char(3)	YES		NULL	
item20	char(1)	YES		NULL	
item21	char(2)	YES		NULL	
item22	char(2)	YES		NULL	
item26	char(2)	YES		NULL	
item27	char(4)	YES		NULL	
item28a	char(2)	YES		NULL	
item28b	char(2)	YES		NULL	
item29	char(6)	YES		NULL	
item30	char(4)	YES		NULL	
item31	char(1)	YES		NULL	
item32	char(4)	YES		NULL	
item33	char(1)	YES		NULL	

Field	Type	Null	Key	Default	Extra
item34	char(2)	YES		NULL	
item35	char(1)	YES		NULL	
item36a	char(1)	YES		NULL	
item36b	char(1)	YES		NULL	
item36c	char(1)	YES		NULL	
item36d	char(1)	YES		NULL	
item37	char(1)	YES		NULL	
item38	char(1)	YES		NULL	
item39	char(4)	YES		NULL	
item40	char(5)	YES		NULL	
item41	char(1)	YES		NULL	
item42a	char(1)	YES		NULL	
item42b	char(1)	YES		NULL	
item43a	char(1)	YES		NULL	
item43b	char(2)	YES		NULL	
item44a	char(1)	YES		NULL	
item44b	char(2)	YES		NULL	
item45	char(3)	YES		NULL	
item46	char(4)	YES		NULL	
item47	char(3)	YES		NULL	
item48	char(5)	YES		NULL	
item49	char(6)	YES		NULL	
item50a	char(3)	YES		NULL	
item50b	char(3)	YES		NULL	
item51	char(4)	YES		NULL	
item52	char(4)	YES		NULL	
item53	char(4)	YES		NULL	
item54a	char(1)	YES		NULL	
item54b	char(4)	YES		NULL	
item55a	char(1)	YES		NULL	
item55b	char(3)	YES		NULL	
item56	char(3)	YES		NULL	
item58	char(1)	YES		NULL	
item59	char(1)	YES		NULL	
item60	char(1)	YES		NULL	
item61	char(1)	YES		NULL	
item62	char(1)	YES		NULL	
item63	char(1)	YES		NULL	
item64	char(3)	YES		NULL	
item65	char(1)	YES		NULL	

Field	Type	Null	Key	Default	Extra
item66	char(3)	YES		NULL	
item67	char(1)	YES		NULL	
item68	char(1)	YES		NULL	
item69	char(1)	YES		NULL	
item70	char(1)	YES		NULL	
item71	char(1)	YES		NULL	
item72	char(1)	YES		NULL	
item75a	char(2)	YES		NULL	
item75b	char(1)	YES		NULL	
item76	char(6)	YES		NULL	
item90	char(4)	YES		NULL	
item91	char(2)	YES		NULL	
item92a	char(3)	YES		NULL	
item92b	char(3)	YES		NULL	
item92c	char(3)	YES		NULL	
item93a	char(4)	YES		NULL	
item93b	char(4)	YES		NULL	
item93c	char(4)	YES		NULL	
item94	char(6)	YES		NULL	
item95	char(6)	YES		NULL	
item96	char(6)	YES		NULL	
item97	char(4)	YES		NULL	
item98a	char(3)	YES		NULL	
item98b	char(2)	YES		NULL	
item99	char(15)	YES		NULL	
item100	char(1)	YES		NULL	
item101	char(1)	YES		NULL	
item102	char(1)	YES		NULL	
item103	char(1)	YES		NULL	
item104	char(1)	YES		NULL	
item105	char(1)	YES		NULL	
item106	char(4)	YES		NULL	
item107	char(1)	YES		NULL	
tem108a	char(1)	YES		NULL	
item108b	char(1)	YES		NULL	
item108c	char(1)	YES		NULL	
item109	char(2)	YES		NULL	
item110	char(1)	YES		NULL	
item111	char(1)	YES		NULL	
item112	char(1)	YES		NULL	

Field	Type	Null	Key	Default	Extra
item113	char(1)	YES		NULL	
item114	char(6)	YES		NULL	
item115	char(4)	YES		NULL	
item116	char(4)	YES		NULL	
funded	char(1)	YES		NULL	
federal	char(1)	YES		NULL	
wo	char(17)	YES		NULL	
dt	char(2)	YES		NULL	
wo_2	char(16)	YES		NULL	
stat	char(1)	YES		NULL	
sr1	char(1)	YES		NULL	
sr2	char(4)	YES		NULL	
status	char(2)	YES		NULL	
date	char(4)	YES		NULL	
longdd	double	YES		NULL	
latdd	double	YES		NULL	
stpostal	char(2)	YES		NULL	
version	char(2)	YES		NULL	
done	tinyint(4)	YES		0	

Table 4-C2: Structure for table "remote" and "citizen"

Field	Type	Null	Key	Default	Extra
id	char(18)	YES	PRI	NULL	
jpgnum	tinyint(4)	YES		NULL	
zipnum	tinyint(4)	YES		NULL	
docnum	tinyint(4)	YES		NULL	
xlsnum	tinyint(4)	YES		NULL	
pdfnum	tinyint(4)	YES		NULL	
flsnum	tinyint(4)	YES		NULL	
testyear	year(4)	YES		NULL	
city	archar(20)	YES		NULL	
fwsnum	tinyint(4)	YES		NULL	

Table 5-C3: Structure for table "nbiinput"

Field	Type	Null	Key	Default	Extra
id	char(18)	YES	PRI	NULL	
stfips	char(2)	YES		NULL	
region	char(1)	YES		NULL	
item8	char(15)	YES		NULL	
item58	char(1)	YES		NULL	
item59	char(1)	YES		NULL	
item60	char(1)	YES		NULL	
item90	char(4)	YES		NULL	
latdd	double	YES		NULL	
longdd	double	YES		NULL	
done	int(11)	YES		NULL	

Table 6-C4: Structure for table "login".

Field	Type	Null	Key	Default	Extra
username	varchar(10)	NO	PRI		
password	varchar(10)	YES		NULL	

APPENDIX D: SYSTEM ENVIRONMENT CONFIGURATION

MySQL Version: 5.0.77

```
mysql> show variables like '%version%';
+-------------------------+-----------------------------------------------------------+
| Variable_name           | Value                                                     |
+-------------------------+-----------------------------------------------------------+
| protocol_version        | 10                                                        |
| version                 | 5.0.77                                                    |
| version_bdb             | Sleepycat Software: Berkeley DB 4.1.24: (January 29, 2009) |
| version_comment         | Source distribution                                       |
| version_compile_machine | x86_64                                                    |
| version_compile_os      | redhat-linux-gnu                                          |
+-------------------------+-----------------------------------------------------------+
6 rows in set (0.00 sec)
```

| Variable_name | Value
| protocol_version | 10
| version | 5.0.77
| version_bdb | Sleepycat Software: Berkeley DB 4.1.24: (January 29, 2009)
| version_comment | Source distribution
| version_compile_machine | x86_64
| version_compile_os | redhat-linux-gnu

PHP Version 5.3.6

PHP Version 5.3.6

System	Linux coe-web3 2.6.18-194.11.4.el5 #1 SMP Fri Sep 17 04:57:05 EDT 2010 x86_64
Build Date	Apr 25 2011 10:40:29
Configure Command	'./configure' '--build=x86_64-redhat-linux-gnu' '--host=x86_64-redhat-linux-gnu' '--target=x86_64-redh without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-pcr mysql=shared,/usr' '--with-mysqli=shared,/usr/lib64/mysql/mysql_config' '--enable-dom=shared' '--w mssql=shared,/usr' '--enable-sysvmsg=shared' '--enable-sysvshm=shared' '--enable-sysvsem=sha
Server API	CGI/FastCGI

Linux System: Linux coe-web3 2.6.18-194

System	Linux coe-web3 2.6.18-194.11.4.el5 #1 SMP Fri Sep 17 04:57:05 EDT 2010 x86_64
Build Date	Apr 25 2011 10:40:29

APPENDIX E: LIST OF THE SOFTWARE PROGRAM

nbibasic.php

```
/*  This code is developed for bridge information visualization
     Developed by Yonghong Tong. Date: 10/08/2012.
     Advisor : Shen-En Chen
 */

<!DOCTYPE html>

<html>
<head>
<title>Bridge </title>
<meta name="viewport" content="initial-scale=1.0,user-scalable=no" />
<style type="text/css">
    html {height:100%}
    body {height:100%; margin:0px; padding:0px;font-family:verdana,arial,sans-serif;font-size:10pt;}
    #map_canvas {height:97%;width:100%}
</style>

<script type="text/javascript"
    src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript" src="http://coe.uncc.edu/~ytong1/tongbridge.js"></script>
<script type="text/javascript">

    var map,map_cavus,img;
    var lat_c0=29.97515;  //29.43515
    var lng_c0=-88.52533; //-89.16533;
    var zoomnum=8;

    var firstON=0;
    var secondON=0;
    var markersArray=[];
    var markersArray2=[];
    var markshow=0;
    var marker;
    var MArray=[];
    var step=50,step1=50,step2=70;
    var MInit=[];
    var play=1;
    var daynight="n";
    var poly;
    var viewlat1,viewlng1,viewlat2,viewlng2,viewlatc,viewlngc;
    var num=0;
    var viewbox=[];
    var viewtrace;
     var geoLayer;
    var z;
    var bounds,bound1,bound2;
    var cTime,cHH,cMM,cSS;
    var pickedpoint;
    var speedline=[],speedlineid=[];
```

```
var currentTime = new Date();
var hrs = currentTime.getHours();

var hideshow="show";

var clk="zoom";

function hideframe()
{
		if (hideshow=="show")
		{
				hideshow="hide";
				parent.cols="0,*";
				document.getElementById("hide").value="Show";
		}
		else
		{
				hideshow="show";
				top.frameOil.cols = "150,*";
				document.getElementById("hide").value="Hide";
		}
}


function timer1(){
    cTime= new Date();
    cHH = cTime.getHours();
    cMM = cTime.getMinutes();
    cSS = cTime.getSeconds();

    var strTime="";
    if (cHH<10) strTime=strTime+"0"+cHH+":";
      else strTime=strTime+cHH+":";

    if (cMM<10) strTime=strTime+"0"+cMM;
      else strTime=strTime+cMM;

    if (cSS<10) strTime=strTime+":0"+cSS;
           else strTime=strTime+":"+cSS;

		strTime=strTime+"     ";

		document.all.timer.innerHTML = strTime;
		setTimeout("timer1();",1000);
}


function greeting(){
		var welcome;
		if (hrs>=0 && hrs<12)
				welcome=" Good Morning ";
		else if (hrs>=12 && hrs<19)
			welcome=" Good Afternoon ";
		else
				welcome=" Good Evening ";
```

```
        document.write(welcome);
}

function rating(n)
{
        switch (n)
        {
        case "A":{str="A:All Condition";break;}
        case "0":{str="0:Failed Condtion";break;}
        case "1":{str="1:Imminent Failure Condition";break;}
        case "2":{str="2:Critical Condition";break;}
        case "3":{str="3:Serious Condition";break;}
        case "4":{str="4:Poor Condition";break;}
        case "5":{str="5:Fair Condition";break;}
        case "6":{str="6:Satisfactory Condition";break;}
        case "7":{str="7:Good Condition";break;}
        case "8":{str="8:Very Good Condition";break;}
        case "9":{str="9:Excellent Condition";break;}
        case "N":{str="N:Not Available";break;}

        }
        return(str);
}



function rate(n)
{
        switch (n)
        {
        case "A":{str="A:All Condition";break;}
        case "0":{str="0:Failed Condtion";break;}
        case "1":{str="1:Imminent Failure Condition";break;}
        case "2":{str="2:Critical Condition";break;}
        case "3":{str="3:Serious Condition";break;}
        case "4":{str="4:Poor Condition";break;}
        case "5":{str="5:Fair Condition";break;}
        case "6":{str="6:Satisfactory Condition";break;}
        case "7":{str="7:Good Condition";break;}
        case "8":{str="8:Very Good Condition";break;}
        case "9":{str="9:Excellent Condition";break;}
        case "N":{str="N:Not Available";break;}

        }
        document.write(str);
}


function nbiest_init(id,lat,lng,lat_c,lng_c,part,state,year)
{
        var latlng=new google.maps.LatLng(lat_c,lng_c);
        var myOptions ={
                zoom:7,
                center:latlng,
                mapTypeId:google.maps.MapTypeId.ROADMAP//HYBRID
        };
```

```
			var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);

			var kmlurl="http://coe.uncc.edu/~ytong1/nbi/kml418/"+state+".kml";
			var stateLayer = new google.maps.KmlLayer(kmlurl);
			stateLayer.setMap(map);

			for (var i=0;i<lat.length;i++)
			{
				img = "http://coe.uncc.edu/~ytong1/images/rate0"+part[i]+".png";
				latlng=new google.maps.LatLng(lat[i],lng[i]);
				var		marker		=		new
google.maps.Marker({map:map,title:"ID:"+id[i]+";"+rating(part[i]),position:latlng,icon:img});

				attachmsg(marker, id[i], year);
			}
		}

		function attachmsg(marker, number, yr) {
			 google.maps.event.addListener(marker, 'click', function() {

window.open('nbidispbridge.php?bridgeid='+number+'&year='+yr,'open','menubar=no,width=520,height=8
00,left=0,top=0,scrollbars=1,toolbar=no');
			 });
		}

		function test(var1,var2)
		{
			for (var i=0; i<var1.length;i++)
				document.write(var1[i]+'<br>');
		}

	</script></head>
	<body
onload="nbiest_init(javaid,javalat,javalng,lat_center,lng_center,javapart,javastate,javayear);timer1();">
	<?php

	$ratenum=$_POST['ratenum'];
	$part=$_POST['part'];
	$state=$_POST['state'];
	$start=$_POST['start'];
	$str_age=$_POST['age'];
	$year=$_POST['year'];
	$str_age="0";

	if ($start!="1")
	{
	    $state="374";
	    $part="item58";
	    $ratenum="3";
	    $year="2011";
	}

	$bridgepart=$part;
	$age=(int)$str_age;

	include "connect.php";
```

```
$deckarray=array('deck','deck1','deck2','deck3','deck4','deck5','deck6','deck7','deck8','deck9');
$supsarray=array('sups','sups1','sups2','sups3','sups4','sups5','sups6','sups7','sups8','sups9');
$subsarray=array('subs','subs1','subs2','subs3','subs4','subs5','subs6','subs7','subs8','subs9');

$tbl="nbi".$year;

if ($state!="000")
  {
    if ($ratenum !="A")
    {

        $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng,item58 as deck,item59 as sups,item60 as subs FROM $tbl where concat(stfips,region)='$state' and $part='$ratenum' and latdd!=0");

    }

    else
    {

        $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng,item58 as deck,item59 as sups,item60 as subs FROM $tbl where concat(stfips,region)='$state' and latdd!=0 order by $part");

    }

    $result_maxlat=mysql_query("SELECT MAX(latdd) FROM $tbl where latdd!=0 and concat(stfips,region)='$state' ");
    $result_minlat=mysql_query("SELECT MIN(latdd) FROM $tbl where latdd!=0 and concat(stfips,region)='$state' ");
    $result_maxlng=mysql_query("SELECT MAX(longdd) FROM $tbl where latdd!=0 and concat(stfips,region)='$state' ");
    $result_minlng=mysql_query("SELECT MIN(longdd) FROM $tbl where latdd!=0 and concat(stfips,region)='$state' ");
  }
else
  {
    if ($ratenum !="A")
    {
        $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng,item58 as deck,item59 as sups,item60 as subs FROM $tbl where $part='$ratenum' and latdd!=0");

    }
    else
    {
        $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng,item58 as deck,item59 as sups,item60 as subs FROM $tbl where $part!='' and latdd!=0 order by $part");
    }

    $result_maxlat=mysql_query("SELECT MAX(latdd) FROM $tbl where latdd!=0");
    $result_minlat=mysql_query("SELECT MIN(latdd) FROM $tbl where latdd!=0");
    $result_maxlng=mysql_query("SELECT MAX(longdd) FROM $tbl where latdd!=0");
    $result_minlng=mysql_query("SELECT MIN(longdd) FROM $tbl where latdd!=0");
  }
```

```
        $maxlat=mysql_fetch_array($result_maxlat);
        $minlat=mysql_fetch_array($result_minlat);
        $maxlng=mysql_fetch_array($result_maxlng);
        $minlng=mysql_fetch_array($result_minlng);

        $lat_center=($maxlat[0]+$minlat[0])/2;
        $lng_center=($maxlng[0]+$minlng[0])/2;

        $id_array=array();
        $lat_array=array();
        $lng_array=array();
        $part_array=array();

        $a=count($lat_array);
        while ($row=mysql_fetch_array($result))
        {
            $id=$row[0];
            $deck=$row[3];
            $sups=$row[4];
            $subs=$row[5];
            $lat =$row[1];
            $lng =$row[2];

            array_push($id_array,$id);
            array_push($lat_array,$lat);
            array_push($lng_array,$lng);
            if ($part=='item58') array_push($part_array,$deck);
            if ($part=='item59') array_push($part_array,$sups);
            if ($part=='item60') array_push($part_array,$subs);

         //echo $id.",".$deck.",".$sups.",".$subs.",".$lat.",".$lng."<br>";
        }
        echo "<table id='tbltitle' height='10px' width='100%' border='0' cellpadding='0' cellspacing='0'><tr>
<td align='left'>";

        echo "<b>".count($lat_array)."</b> bridges <b><script>item('".$bridgepart."')</script></b> rating
are <b><script> rate('".$ratenum."');</script></b> in <script>stfips('".$state."');</script>"; //" in the year
".$year;

        ?>

        <script>
         // </td><td width='50px'><input type='image' src='http://coe.uncc.edu/~ytong1/images/export.jpg'
border='0' alt='export' onclick='location.href="<?php echo $myfile2;?>";')>
        </script>


        </td><td width='60px' align='right'><b><div id='timer'></div></b></td><td width='130px'
align='right'><b><script type='text/javascript'>greeting();</script>
        </td></tr></table>

        <script type="text/javascript">
            var javaid=[];
            var javalat=[];
            var javalng=[];
```

```
    var javapart=[];

    var lat_center;
    var lng_center;
    var ratenum;

    var javastate;
    var javayear;

    lat_center=<? echo $lat_center;?>;
    lng_center=<? echo $lng_center;?>;

    javastate=stfips2('<? echo $state; ?>');
    javayear=<? echo $year; ?>;

    <?
     for ($i=0;$i<count($lat_array);$i++)
       {
              echo  'javaid['.$i.']="'.$id_array[$i].'";';
              echo 'javalat['.$i.']="'.$lat_array[$i].'";';//'Jan';
              echo 'javalng['.$i.']="'.$lng_array[$i].'";';
              echo 'javapart['.$i.']="'.$part_array[$i].'";';
       }

    ?>

</script>
<div id="map_canvas"></div>

<?
    mysql_close($conn);
?>

</body>
</html>
```

nbiana.php

```
/*   This code is developed for bridge information visualization
      Developed by Yonghong Tong. Date: 10/08/2012.
      Advisor : Shen-En Chen
 */

<!DOCTYPE html>
<html>
<head>

<title>Bridge Inventory </title>
<meta name="viewport" content="initial-scale=1.0,user-scalable=no" />
<style type="text/css">
    html {height:100%}
    body {height:100%; margin:0px; padding:0px;font-family:verdana,arial,sans-serif;font-size:10pt}
    #map_canvas {height:50%}
</style>

<script type="text/javascript"
    src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript" src="http://coe.uncc.edu/~ytong1/tongbridge.js"></script>

<script type="text/javascript">
    function rating(n)
    {
            switch (n)
            {
            case "0":{str="0:Failed Condtion";break;}
            case "1":{str="1:Imminent Failure Condition";break;}
            case "2":{str="2:Critical Condition";break;}
            case "3":{str="3:Serious Condition";break;}
            case "4":{str="4:Poor Condition";break;}
            case "5":{str="5:Fair Condition";break;}
            case "6":{str="6:Satisfactory Condition";break;}
            case "7":{str="7:Good Condition";break;}
            case "8":{str="8:Very Good Condition";break;}
            case "9":{str="9:Excellent Condition";break;}
            case "N":{str="N:Not Applicable";break;}
            }
            return (str);
    }

    function init(lat,lng,lat_c,lng_c,deck)
    {
            var latlng=new google.maps.LatLng(lat_c,lng_c);
            var myOptions ={
                    zoom:7,
                    center:latlng,
                    mapTypeId:google.maps.MapTypeId.ROADMAP//HYBRID
            };
            var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);

            var stateLayer = new google.maps.KmlLayer('http://coe.uncc.edu/~ytong1/nbi/NC.kml');
```

```
		stateLayer.setMap(map);

		for (var i=0;i<lat.length;i++)
		{
			img = "http://coe.uncc.edu/~ytong1/images/rate0"+deck[i]+".png";
			latlng=new google.maps.LatLng(lat[i],lng[i]);
			var marker = new
google.maps.Marker({map:map,title:rating(deck[i]),position:latlng,icon:img});
		}
	}


	function init(lat,lng,lat_c,lng_c,deck,state)
	{
		var latlng=new google.maps.LatLng(lat_c,lng_c);
		var myOptions ={
			zoom:7,
			center:latlng,
			mapTypeId:google.maps.MapTypeId.ROADMAP//HYBRID
		};
		var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);

		var kmlurl="http://coe.uncc.edu/~ytong1/nbi/kml418/"+state+".kml";
		var stateLayer = new google.maps.KmlLayer(kmlurl);
		stateLayer.setMap(map);

		for (var i=0;i<lat.length;i++)
		{
			img = "http://coe.uncc.edu/~ytong1/images/rate0"+deck[i]+".png";
			latlng=new google.maps.LatLng(lat[i],lng[i]);
			var marker = new
google.maps.Marker({map:map,title:rating(deck[i]),position:latlng,icon:img});
		}
	}

	function nbiest_init(id,lat,lng,lat_c,lng_c,part,state,year)
	{
		var latlng=new google.maps.LatLng(lat_c,lng_c);
		var myOptions ={
			zoom:7,
			center:latlng,
			mapTypeId:google.maps.MapTypeId.ROADMAP//HYBRID
		};
		var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);

		var kmlurl="http://coe.uncc.edu/~ytong1/nbi/kml418/"+state+".kml";
		var stateLayer = new google.maps.KmlLayer(kmlurl);
		stateLayer.setMap(map);

		for (var i=0;i<lat.length;i++)
		{
			img = "http://coe.uncc.edu/~ytong1/images/rate0"+part[i]+".png";
			latlng=new google.maps.LatLng(lat[i],lng[i]);
			var marker = new
google.maps.Marker({map:map,title:"ID:"+id[i]+";"+rating(part[i]),position:latlng,icon:img});
```

```
                        attachmsg(marker, id[i], year);
                    }
            }

            function attachmsg(marker, number, yr) {
                    google.maps.event.addListener(marker, 'click', function() {

window.open('nbidispbridge.php?bridgeid='+number+'&year='+yr,'open','menubar=no,width=520,height=8
00,left=0,top=0,scrollbars=1,toolbar=no');
                    });
            }


            function test(var1,var2)
            {
                    for (var i=0; i<var1.length;i++)
                        document.write(var1[i]+'<br>');
            }

        </script>
        <body  onload="nbiest_init(javaid,javalat,javalng,lat_center,lng_center,javapart,javastate,javayear);">

        <?php
        $year=$_POST['year'];
        $ratenum=$_POST['ratenum'];
        $part=$_POST['part'];
        $state=$_POST['state'];
        $start=$_POST['start'];

        if ($start!="1")
        {
            $state="374";
            $part="item58";
            $ratenum="3";
            $year="2011";
        }

        $bridgepart=$part;
        $age=(int)$str_age;

        include "connect.php";

        $deckarray=array('deck','deck1','deck2','deck3','deck4','deck5','deck6','deck7','deck8','deck9');
        $supsarray=array('sups','sups1','sups2','sups3','sups4','sups5','sups6','sups7','sups8','sups9');
        $subsarray=array('subs','subs1','subs2','subs3','subs4','subs5','subs6','subs7','subs8','subs9');

        $state_array=array();
        $total_array=array();

        $deck_array=array();
        $sups_array=array();
        $subs_array=array();

        $tbl="nbi".$year;
```

```
	$resultmap=mysql_query("SELECT COUNT(item58) as total,concat(stfips,region) as state FROM $tbl WHERE latdd!=0 GROUP BY concat(stfips,region)");
	while ($rowmap=mysql_fetch_array($resultmap))
	{
	    $total=$rowmap[0];
	    $stateno=$rowmap[1];

	    array_push($total_array,$total);
	    array_push($state_array,$stateno);
	}

	$temp=0;
	$resultmap=mysql_query("SELECT COUNT(item58) as deck,concat(stfips,region) as state  FROM $tbl WHERE item58='$ratenum' AND latdd!=0 GROUP BY concat(stfips,region)");
	while ($rowdeck=mysql_fetch_array($resultmap))
	{
	    $deck=$rowdeck[0];
	    $stateno=$rowdeck[1];
	    while ($state_array[$temp]<$stateno)
	    {
	              array_push($deck_array,"0");
	              $temp++;
	    }
	    if ($state_array[$temp]==$stateno)
	    {
	              array_push($deck_array,$deck);
	              $temp++;
	    }

	}


	$temp=0;
	$resultmap=mysql_query("SELECT COUNT(item59) as sups,concat(stfips,region) as state  FROM $tbl WHERE item59='$ratenum' AND latdd!=0 GROUP BY concat(stfips,region)");
	while ($rowsups=mysql_fetch_array($resultmap))
	{
	    $sups=$rowsups[0];
	    $stateno=$rowsups[1];
	    while ($state_array[$temp]<$stateno)
	    {
	              array_push($sups_array,"0");
	              $temp++;
	    }
	    if ($state_array[$temp]==$stateno)
	    {
	              array_push($sups_array,$sups);
	              $temp++;
	    }

	}



	$temp=0;
```

```
$resultmap=mysql_query("SELECT COUNT(item60) as subs,concat(stfips,region) as state FROM $tbl WHERE item60='$ratenum' AND latdd!=0 GROUP BY concat(stfips,region)");
while ($rowsubs=mysql_fetch_array($resultmap))
{
    $subs=$rowsubs[0];
    $stateno=$rowsubs[1];
    while ($state_array[$temp]<$stateno)
    {
        array_push($subs_array,"0");
        $temp++;
    }
    if ($state_array[$temp]==$stateno)
    {
        array_push($subs_array,$subs);
        $temp++;
    }

}

if ($state!="000")
  {
    if ($ratenum !="A")
    {

        $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng,item58 as deck,item59 as sups,item60 as subs FROM $tbl where concat(stfips,region)='$state' and $part='$ratenum' and latdd!=0");

    }

    else
    {

        $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng,item58 as deck,item59 as sups,item60 as subs FROM $tbl where concat(stfips,region)='$state' and latdd!=0 order by $part");

    }

    $result_maxlat=mysql_query("SELECT MAX(latdd) FROM $tbl where latdd!=0 and concat(stfips,region)='$state' ");
    $result_minlat=mysql_query("SELECT MIN(latdd) FROM $tbl where latdd!=0 and concat(stfips,region)='$state' ");
    $result_maxlng=mysql_query("SELECT MAX(longdd) FROM $tbl where latdd!=0 and concat(stfips,region)='$state' ");
    $result_minlng=mysql_query("SELECT MIN(longdd) FROM $tbl where latdd!=0 and concat(stfips,region)='$state' ");
  }
else
  {
    if ($ratenum !="A")
    {
        $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng,item58 as deck,item59 as sups,item60 as subs FROM $tbl where $part='$ratenum' and latdd!=0");

    }
```

```
        else
        {
                $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng,item58 as deck,item59 as sups,item60 as subs FROM $tbl where $part!='' and latdd!=0 order by $part");
        }

        $result_maxlat=mysql_query("SELECT MAX(latdd) FROM $tbl where latdd!=0");
        $result_minlat=mysql_query("SELECT MIN(latdd) FROM $tbl where latdd!=0");
        $result_maxlng=mysql_query("SELECT MAX(longdd) FROM $tbl where latdd!=0");
        $result_minlng=mysql_query("SELECT MIN(longdd) FROM $tbl where latdd!=0");
      }



    $maxlat=mysql_fetch_array($result_maxlat);
    $minlat=mysql_fetch_array($result_minlat);
    $maxlng=mysql_fetch_array($result_maxlng);
    $minlng=mysql_fetch_array($result_minlng);

    $lat_center=($maxlat[0]+$minlat[0])/2;
    $lng_center=($maxlng[0]+$minlng[0])/2;

    $id_array=array();
    $lat_array=array();
    $lng_array=array();
    $part_array=array();


    $a=count($lat_array);
    while ($row=mysql_fetch_array($result))
    {
        $id=$row[0];
        $deck=$row[3];
        $sups=$row[4];
        $subs=$row[5];
        $lat =$row[1];
        $lng =$row[2];

        array_push($id_array,$id);
        array_push($lat_array,$lat);
        array_push($lng_array,$lng);
        if ($part=='item58') array_push($part_array,$deck);
        if ($part=='item59') array_push($part_array,$sups);
        if ($part=='item60') array_push($part_array,$subs);

    }

    echo " ";
    echo "<a target='_blank' href='http://www.fhwa.dot.gov/bridge/mtguide.pdf'><img src='http://coe.uncc.edu/~ytong1/images/pdf.gif' height='12' border='0' title='Recording and Coding Guide'></a>";
    echo "<img src='http://coe.uncc.edu/~ytong1/images/scale.bmp'  title='Bridge Condition Rating' usemap='#rate' border='0'/> ";
    echo "<map name='rate'>";
```

```
echo "<area shape='rect' coords='43,1,68,15' title='0:Failed Condtion'/>";
echo "<area shape='rect' coords='68,1,93,15' title='1:Imminent Failure Condition' />";
echo "<area shape='rect' coords='93,1,118,15' title='2:Critical Condition' />";
echo "<area shape='rect' coords='118,1,143,15' title='3:Serious Condition'/>";
echo "<area shape='rect' coords='143,1,168,15' title='4:Poor Condition' />";
echo "<area shape='rect' coords='168,1,193,15' title='5:Fair Condition' />";
echo "<area shape='rect' coords='193,1,218,15' title='6:Satisfactory Condition'/>";
echo "<area shape='rect' coords='218,1,243,15' title='7:Good Condition'/>";
echo "<area shape='rect' coords='243,1,268,15' title='8:Very Good Condition' />";
echo "<area shape='rect' coords='268,1,293,15' title='9:Excellent Condition' />";
echo "</map>";

echo "<br><b>".count($lat_array)."</b> bridges <b><script>item('".$bridgepart."')</script></b>
rating are <b><script> rate('".$ratenum."');</script></b> in <script>stfips('".$state."');</script> in ".$year;
?>

<script type="text/javascript">
    var javaid=[];
    var javalat=[];
    var javalng=[];
    var javapart=[];

    var lat_center;
    var lng_center;
    var ratenum;

    var javastate;
    var javayear;


    lat_center=<? echo $lat_center;?>;
    lng_center=<? echo $lng_center;?>;

    javastate=stfips2('<? echo $state; ?>');
    javayear=<? echo $year; ?>;

    <?
     for ($i=0;$i<count($lat_array);$i++)
       {
              echo  'javaid['.$i.']="'.$id_array[$i].'";';
              echo 'javalat['.$i.']="'.$lat_array[$i].'";';//'Jan';
              echo 'javalng['.$i.']="'.$lng_array[$i].'";';
              echo 'javapart['.$i.']="'.$part_array[$i].'";';
       }

    ?>

</script>
<div id="map_canvas"></div>

<?
    $deckvalue=array();
    $supsvalue=array();
    $subsvalue=array();

    $deckrate=array();
```

```
    $supsrate=array();
    $subsrate=array();

while ($row=mysql_fetch_array($resultmap))
{
    $numtotal=$row['total'];
    $numdeck=$row['deck'];
    $numsups=$row['sups'];
    $numsubs=$row['subs'];

    $decktemp=sqrt($numdeck/$numtotal);
    $supstemp=sqrt($numsups/$numtotal);
    $substemp=sqrt($numsubs/$numtotal);

    array_push($deckrate,$decktemp);
    array_push($supsrate,$supstemp);
    array_push($subsrate,$substemp);

    $strdeck=" "."$numdeck"." / "."$numtotal";
    $strsups=" "."$numsups"." / "."$numtotal";
    $strsubs=" "."$numsubs"." / "."$numtotal";

    array_push($deckvalue,$strdeck);
    array_push($supsvalue,$strsups);
    array_push($subsvalue,$strsubs);

}


    for ($i=0; $i<52; $i++){
            $decktemp=sqrt($deck_array[$i]/$total_array[$i]);
            $supstemp=sqrt($sups_array[$i]/$total_array[$i]);
            $substemp=sqrt($subs_array[$i]/$total_array[$i]);

            array_push($deckrate,$decktemp);
            array_push($supsrate,$supstemp);
            array_push($subsrate,$substemp);

            $strdeck=" "."$deck_array[$i]"." / "."$total_array[$i]";
            $strsups=" "."$sups_array[$i]"." / "."$total_array[$i]";
            $strsubs=" "."$subs_array[$i]"." / "."$total_array[$i]";

            array_push($deckvalue,$strdeck);
            array_push($supsvalue,$strsups);
            array_push($subsvalue,$strsubs);
    }


    switch($ratenum)
    {
            case 0:
            case 1:
            case 2:
                    $chcodeck="ff0000";
                    break;
            case 3:
```

```
            case 4:
                $chcodeck="ff8800";
                break;
            case 5:
            case 6:
            case 7:
                $chcodeck="ffdd00";
                break;
            case 8:
            case 9:
                $chcodeck="00ff00";
                break;
        }

    ?>

    <?
        mysql_close($conn);
    ?>

    <? //****************************************************************** Deck  *?>
    <table align='center' width="100%">
    <tr><td width="50%" align="center"><b>Deck</b> = <script> rate('<?echo
$ratenum;?>')</script></td>
    <td width="50%" align="center"><b>Superstructure</b> = <script> rate('<?echo
$ratenum;?>')</script></td></tr>

    <tr><td align='center'>

    <img src="http://chart.apis.google.com/chart?
    chf=bg,s,99B6CC
    &chs=250x125
    &cht=t
    &chco=ffffff,ffffff,<? echo $chcodeck; ?>
    &chld=ALAKAZARCACOCTDCDEFLGAHIIDILINIAKSKYLAMEMDMAMIMNMSMOMTNE
NVNHNJNMNYNCNDOHOKORPARISCSDTNTXUTVTVAWAWVWIWY
    &chd=t:<? for ($i=0; $i<50; $i++) echo $deckrate[$i].","; echo $deckrate[50];?>
    &chds=0,0.6
    &chtm=usa"
    width="250" height="125" alt=""
    usemap='#mapdeck'border='0'/>

    <map name='mapdeck'>
    <area shape='poly' coords='158,71,157,76,157,93,161,94,161,90,172,90,169,71' title='AL:<? echo
$deckvalue[0]?>' />
    <area shape='rect' coords='38,92,53,112' title='AK:<? echo $deckvalue[1]?>' />
    <area shape='rect' coords='45,62,67,88' title='AZ:<? echo $deckvalue[2]?>' />
    <area shape='poly' coords='130,63,151,63,144,76,144,79,132,79,132,77,129,77' title='AR:<? echo
$deckvalue[3]?>' />
    <area shape='poly' coords='1,36,2,50,32,82,43,82,43,69,20,50,20,36' title='CA:<? echo
$deckvalue[4]?>' />
    <area shape='rect' coords='67,41,98,60' title='CO:<? echo $deckvalue[5]?>' />
    <area shape='poly' coords='221,40,229,40,228,36,221,36' title='CT:<? echo $deckvalue[6]?>' />
    <area shape='poly' coords='213,54,215,51,211,47,211,53' title='DE:<? echo $deckvalue[7]?>' />
    <area shape='poly' coords='161,90,161,93,189,120,193,115,187,91' title='FL:<? echo
$deckvalue[9]?>' />
```

```
<area shape='poly' coords='170,70,172,91,187,91,189,85,178,70' title='GA:<? echo $deckvalue[10]?>' />
<area shape='poly' coords='64,98,77,106,85,116,91,112,84,103,66,96' title='HI:<? echo $deckvalue[11]?>' />
<area shape='poly' coords='33,36,58,36,58,23,50,23,38,7,37,2,33,2' title='ID:<? echo $deckvalue[12]?>' />
<area shape='poly' coords='152,61,158,58,160,52,160,37,158,34,146,34,149,37,142,43,143,47' title='IL:<? echo $deckvalue[13]?>' />
<area shape='poly' coords='158,58,160,52,161,38,172,38,172,51,167,55' title='IN:<? echo $deckvalue[14]?>' />
<area shape='poly' coords='121,28,125,43,144,43,148,36,145,28' title='IA:<? echo $deckvalue[15]?>' />
<area shape='rect' coords='98,46,129,60' title='KS:<? echo $deckvalue[16]?>' />
<area shape='poly' coords='152,64,152,60,160,56,166,55,172,51,182,53,184,58,178,62' title='KY:<? echo $deckvalue[17]?>' />
<area shape='poly' coords='132,80,134,95,148,99,155,99,155,94,150,91,143,89,143,86,146,84,145,79' title='LA:<? echo $deckvalue[18]?>' />
<area shape='poly' coords='232,21,233,30,242,27,249,21,246,17,246,11,20,10,236,16' title='ME:<? echo $deckvalue[19]?>' />
<area shape='poly' coords='212,57,213,54,212,53,194,47,202,50,207,54' title='MD:<? echo $deckvalue[20]?>' />
<area shape='poly' coords='221,32,221,36,231,36,233,39,237,39,237,36,233,35,232,32' title='MA:<? echo $deckvalue[21]?>' />
<area shape='poly' coords='141,12,163,20,159,33,162,37,179,37,184,27,157,5' title='MI:<? echo $deckvalue[22]?>' />
<area shape='poly' coords='119,2,122,28,144,28,138,23,138,16,151,6,128,1' title='MN:<? echo $deckvalue[23]?>' />
<area shape='poly' coords='150,91,143,89,143,86,146,84,145,74,149,70,158,70,156,93,151,93' title='MS:<? echo $deckvalue[24]?>' />
<area shape='poly' coords='125,43,130,63,151,63,144,43' title='MO:<? echo $deckvalue[25]?>' />
<area shape='poly' coords='38,2,38,7,51,23,58,22,89,21,89,2' title='MT:<? echo $deckvalue[26]?>' />
<area shape='poly' coords='89,30,89,41,97,41,97,45,125,45,119,32,111,31' title='NE:<? echo $deckvalue[27]?>' />
<area shape='poly' coords='20,36,20,51,42,70,43,65,46,65,46,36' title='NV:<? echo $deckvalue[28]?>' />
<area shape='poly' coords='233,32,233,30,232,20,230,21,226,28,226,32' title='NH:<? echo $deckvalue[29]?>' />
<area shape='poly' coords='215,51,218,48,219,41,216,39,213,40,214,46,211,48' title='NJ:<? echo $deckvalue[30]?>' />
<area shape='rect' coords='67,61,93,85' title='NM:<? echo $deckvalue[31]?>' />
<area shape='poly' coords='195,36,214,36,220,42,227,41,221,40,222,21,215,21,207,28,197,28,193,35' title='NY:<? echo $deckvalue[32]?>' />
<area shape='poly' coords='178,70,188,70,200,76,213,68,210,62,186,62,174,70' title='NC:<? echo $deckvalue[33]?>' />
<area shape='rect' coords='89,2,120,16' title='ND:<? echo $deckvalue[34]?>' />
<area shape='poly' coords='172,51,182,53,191,47,191,34,182,37,172,37' title='OH:<? echo $deckvalue[35]?>' />
<area shape='rect' coords='106,61,129,75' title='OK:<? echo $deckvalue[36]?>' />
<area shape='rect' coords='1,18,33,36' title='OR:<? echo $deckvalue[37]?>' />
<area shape='rect' coords='191,36,214,47' title='PA:<? echo $deckvalue[38]?>' />
<area shape='poly' coords='229,40,232,40,232,38,229,37' title='RI:<? echo $deckvalue[39]?>' />
<area shape='poly' coords='189,85,178,70,188,70,200,76' title='SC:<? echo $deckvalue[40]?>' />
<area shape='rect' coords='89,16,121,31' title='SD:<? echo $deckvalue[41]?>' />
<area shape='poly' coords='187,63,151,63,149,69,174,69' title='TN:<? echo $deckvalue[42]?>' />
```

```
<area shape='poly' coords='78,85,92,99,119,114,132,96,131,76,107,73,107,62,93,62,93,85' title='TX:<? echo $deckvalue[43]?>' />
<area shape='rect' coords='46,37,67,61' title='UT:<? echo $deckvalue[44]?>' />
<area shape='poly' coords='230,21,226,28,226,32,222,32,222,21' title='VT:<? echo $deckvalue[45]?>' />
<area shape='poly' coords='210,62,178,62,184,58,188,59,192,57,194,53,202,49,206.52,205,54,209,55,214,55' title='VA:<? echo $deckvalue[46]?>' />
<area shape='rect' coords='0,0,33,16' title='WA:<? echo $deckvalue[47]?>' />
<area shape='poly' coords='188,59,192,57,194,53,202,49,191,47,182,54,186,59' title='WV:<? echo $deckvalue[48]?>' />
<area shape='poly' coords='144,28,138,23,138,16,141,12,163,20,159,33,146,33' title='WI:<? echo $deckvalue[49]?>' />
<area shape='rect' coords='59,21,89,41' title='WY:<? echo $deckvalue[50]?>' />
</map>
</td>

<? //************************************************************ Superstructure ?>
<td align='center'>
<img src="http://chart.apis.google.com/chart?
chf=bg,s,99B6CC
&chs=250x125
&cht=t
&chco=FFFFFF,FFFFFF,<? echo $chcodeck; ?>
&chld=ALAKAZARCACOCTDCDEFLGAHIIDILINIAKSKYLAMEMDMAMIMNMSMOMTNE
NVNHNJNMNYNCNDOHOKORPARISCSDTNTXUTVTVAWAWVWIWY
&chd=t:<? for ($i=0; $i<50; $i++) echo $supsrate[$i].","; echo $supsrate[50];?>
&chds=0,0.6
&chtm=usa" width="250" height="125" alt=""
usemap='#mapsups'border='0'/>

<map name='mapsups'>
<area shape='poly' coords='158,71,157,76,157,93,161,94,161,90,172,90,169,71' title='AL:<? echo $supsvalue[0]?>' />
<area shape='rect' coords='38,92,53,112' title='AK:<? echo $supsvalue[1]?>' />
<area shape='rect' coords='45,62,67,88' title='AZ:<? echo $supsvalue[2]?>' />
<area shape='poly' coords='130,63,151,63,144,76,144,79,132,79,132,77,129,77' title='AR:<? echo $supsvalue[3]?>' />
<area shape='poly' coords='1,36,2,50,32,82,43,82,43,69,20,50,20,36' title='CA:<? echo $supsvalue[4]?>' />
<area shape='rect' coords='67,41,98,60' title='CO:<? echo $supsvalue[5]?>' />
<area shape='poly' coords='221,40,229,40,228,36,221,36' title='CT:<? echo $supsvalue[6]?>' />
<area shape='poly' coords='213,54,215,51,211,47,211,53' title='DE:<? echo $supsvalue[7]?>' />
<area shape='poly' coords='161,90,161,93,189,120,193,115,187,91' title='FL:<? echo $supsvalue[9]?>' />
<area shape='poly' coords='170,70,172,91,187,91,189,85,178,70' title='GA:<? echo $supsvalue[10]?>' />
<area shape='poly' coords='64,98,77,106,85,116,91,112,84,103,66,96' title='HI:<? echo $supsvalue[11]?>' />
<area shape='poly' coords='33,36,58,36,58,23,50,23,38,7,37,2,33,2' title='ID:<? echo $supsvalue[12]?>' />
<area shape='poly' coords='152,61,158,58,160,52,160,37,158,34,146,34,149,37,142,43,143,47' title='IL:<? echo $supsvalue[13]?>' />
<area shape='poly' coords='158,58,160,52,161,38,172,38,172,51,167,55' title='IN:<? echo $supsvalue[14]?>' />
```

```
<area shape='poly' coords='121,28,125,43,144,43,148,36,145,28' title='IA:<? echo $supsvalue[15]?>' />
<area shape='rect' coords='98,46,129,60' title='KS:<? echo $supsvalue[16]?>' />
<area shape='poly' coords='152,64,152,60,160,56,166,55,172,51,182,53,184,58,178,62' title='KY:<? echo $supsvalue[17]?>' />
<area shape='poly' coords='132,80,134,95,148,99,155,99,155,94,150,91,143,89,143,86,146,84,145,79' title='LA:<? echo $supsvalue[18]?>' />
<area shape='poly' coords='232,21,233,30,242,27,249,21,246,17,246,11,20,10,236,16' title='ME:<? echo $supsvalue[19]?>' />
<area shape='poly' coords='212,57,213,54,212,53,194,47,202,50,207,54' title='MD:<? echo $supsvalue[20]?>' />
<area shape='poly' coords='221,32,221,36,231,36,233,39,237,39,237,36,233,35,232,32' title='MA:<? echo $supsvalue[21]?>' />
<area shape='poly' coords='141,12,163,20,159,33,162,37,179,37,184,27,157,5' title='MI:<? echo $supsvalue[22]?>' />
<area shape='poly' coords='119,2,122,28,144,28,138,23,138,16,151,6,128,1' title='MN:<? echo $supsvalue[23]?>' />
<area shape='poly' coords='150,91,143,89,143,86,146,84,145,74,149,70,158,70,156,93,151,93' title='MS:<? echo $supsvalue[24]?>' />
<area shape='poly' coords='125,43,130,63,151,63,144,43' title='MO:<? echo $supsvalue[25]?>' />
<area shape='poly' coords='38,2,38,7,51,23,58,22,89,21,89,2' title='MT:<? echo $supsvalue[26]?>' />
<area shape='poly' coords='89,30,89,41,97,41,97,45,125,45,119,32,111,31' title='NE:<? echo $supsvalue[27]?>' />
<area shape='poly' coords='20,36,20,51,42,70,43,65,46,65,46,36' title='NV:<? echo $supsvalue[28]?>' />
<area shape='poly' coords='233,32,233,30,232,20,230,21,226,28,226,32' title='NH:<? echo $supsvalue[29]?>' />
<area shape='poly' coords='215,51,218,48,219,41,216,39,213,40,214,46,211,48' title='NJ:<? echo $supsvalue[30]?>' />
<area shape='rect' coords='67,61,93,85' title='NM:<? echo $supsvalue[31]?>' />
<area shape='poly' coords='195,36,214,36,220,42,227,41,221,40,222,21,215,21,207,28,197,28,193,35' title='NY:<? echo $supsvalue[32]?>' />
<area shape='poly' coords='178,70,188,70,200,76,213,68,210,62,186,62,174,70' title='NC:<? echo $supsvalue[33]?>' />
<area shape='rect' coords='89,2,120,16' title='ND:<? echo $supsvalue[34]?>' />
<area shape='poly' coords='172,51,182,53,191,47,191,34,182,37,172,37' title='OH:<? echo $supsvalue[35]?>' />
<area shape='rect' coords='106,61,129,75' title='OK:<? echo $supsvalue[36]?>' />
<area shape='rect' coords='1,18,33,36' title='OR:<? echo $supsvalue[37]?>' />
<area shape='rect' coords='191,36,214,47' title='PA:<? echo $supsvalue[38]?>' />
<area shape='poly' coords='229,40,232,40,232,38,229,37' title='RI:<? echo $supsvalue[39]?>' />
<area shape='poly' coords='189,85,178,70,188,70,200,76' title='SC:<? echo $supsvalue[40]?>' />
<area shape='rect' coords='89,16,121,31' title='SD:<? echo $supsvalue[41]?>' />
<area shape='poly' coords='187,63,151,63,149,69,174,69' title='TN:<? echo $supsvalue[42]?>' />
<area shape='poly' coords='78,85,92,99,119,114,132,96,131,76,107,73,107,62,93,62,93,85' title='TX:<? echo $supsvalue[43]?>' />
<area shape='rect' coords='46,37,67,61' title='UT:<? echo $supsvalue[44]?>' />
<area shape='poly' coords='230,21,226,28,226,32,222,32,222,21' title='VT:<? echo $supsvalue[45]?>' />
<area shape='poly' coords='210,62,178,62,184,58,188,59,192,57,194,53,202,49,206.52,205,54,209,55,214,55' title='VA:<? echo $supsvalue[46]?>' />
<area shape='rect' coords='0,0,33,16' title='WA:<? echo $supsvalue[47]?>' />
```

```
<area shape='poly' coords='188,59,192,57,194,53,202,49,191,47,182,54,186,59' title='WV:<? echo
$supsvalue[48]?>' />
<area shape='poly' coords='144,28,138,23,138,16,141,12,163,20,159,33,146,33' title='WI:<? echo
$supsvalue[49]?>' />
<area shape='rect' coords='59,21,89,41' title='WY:<? echo $supsvalue[50]?>' />
</map>

</td></tr><tr>
<td align="center"><b>Substructure </b>= <script> rate('<?echo
$ratenum;?>')</script></td><td> </td></tr>
<tr>
<? //************************************************************ Substructure ?>
<td align='center'>
<img src="http://chart.apis.google.com/chart?
chf=bg,s,99B6CC
&chs=250x125
&cht=t
&chco=FFFFFF,FFFFFF,<? echo $chcodeck; ?>
&chld=ALAKAZARCACOCTDCDEFLGAHIIDILINIAKSKYLAMEMDMAMIMNMSMOMTNE
NVNHNJNMNYNCNDOHOKORPARISCSDTNTXUTVTVAWAWVWIWY
&chd=t:<? for ($i=0; $i<50; $i++) echo $subsrate[$i].","; echo $subsrate[50];?>
&chds=0,0.6
&chtm=usa" width="250" height="125" alt=""
usemap='#mapsubs'border='0'/>
<map name='mapsubs'>
<area shape='poly' coords='158,71,157,76,157,93,161,94,161,90,172,90,169,71' title='AL:<? echo
$subsvalue[0]?>' />
<area shape='rect' coords='38,92,53,112' title='AK:<? echo $subsvalue[1]?>' />
<area shape='rect' coords='45,62,67,88' title='AZ:<? echo $subsvalue[2]?>' />
<area shape='poly' coords='130,63,151,63,144,76,144,79,132,79,132,77,129,77' title='AR:<? echo
$subsvalue[3]?>' />
<area shape='poly' coords='1,36,2,50,32,82,43,82,43,69,20,50,20,36' title='CA:<? echo
$subsvalue[4]?>' />
<area shape='rect' coords='67,41,98,60' title='CO:<? echo $subsvalue[5]?>' />
<area shape='poly' coords='221,40,229,40,228,36,221,36' title='CT:<? echo $subsvalue[6]?>' />
<area shape='poly' coords='213,54,215,51,211,47,211,53' title='DE:<? echo $subsvalue[7]?>' />
<area shape='poly' coords='161,90,161,93,189,120,193,115,187,91' title='FL:<? echo
$subsvalue[9]?>' />
<area shape='poly' coords='170,70,172,91,187,91,189,85,178,70' title='GA:<? echo $subsvalue[10]?>'
/>
<area shape='poly' coords='64,98,77,106,85,116,91,112,84,103,66,96' title='HI:<? echo
$subsvalue[11]?>' />
<area shape='poly' coords='33,36,58,36,58,23,50,23,38,7,37,2,33,2' title='ID:<? echo
$subsvalue[12]?>' />
<area shape='poly' coords='152,61,158,58,160,52,160,37,158,34,146,34,149,37,142,43,143,47'
title='IL:<? echo $subsvalue[13]?>' />
<area shape='poly' coords='158,58,160,52,161,38,172,38,172,51,167,55' title='IN:<? echo
$subsvalue[14]?>' />
<area shape='poly' coords='121,28,125,43,144,43,148,36,145,28' title='IA:<? echo $subsvalue[15]?>'
/>
<area shape='rect' coords='98,46,129,60' title='KS:<? echo $subsvalue[16]?>' />
<area shape='poly' coords='152,64,152,60,160,56,166,55,172,51,182,53,184,58,178,62' title='KY:<?
echo $subsvalue[17]?>' />
<area shape='poly'
coords='132,80,134,95,148,99,155,99,155,94,150,91,143,89,143,86,146,84,145,79' title='LA:<? echo
$subsvalue[18]?>' />
```

```
<area shape='poly' coords='232,21,233,30,242,27,249,21,246,17,246,11,20,10,236,16' title='ME:<? echo $subsvalue[19]?>' />
<area shape='poly' coords='212,57,213,54,212,53,194,47,202,50,207,54' title='MD:<? echo $subsvalue[20]?>' />
<area shape='poly' coords='221,32,221,36,231,36,233,39,237,39,237,36,233,35,232,32' title='MA:<? echo $subsvalue[21]?>' />
<area shape='poly' coords='141,12,163,20,159,33,162,37,179,37,184,27,157,5' title='MI:<? echo $subsvalue[22]?>' />
<area shape='poly' coords='119,2,122,28,144,28,138,23,138,16,151,6,128,1' title='MN:<? echo $subsvalue[23]?>' />
<area shape='poly' coords='150,91,143,89,143,86,146,84,145,74,149,70,158,70,156,93,151,93' title='MS:<? echo $subsvalue[24]?>' />
<area shape='poly' coords='125,43,130,63,151,63,144,43' title='MO:<? echo $subsvalue[25]?>' />
<area shape='poly' coords='38,2,38,7,51,23,58,22,89,21,89,2' title='MT:<? echo $subsvalue[26]?>' />
<area shape='poly' coords='89,30,89,41,97,41,97,45,125,45,119,32,111,31' title='NE:<? echo $subsvalue[27]?>' />
<area shape='poly' coords='20,36,20,51,42,70,43,65,46,65,46,36' title='NV:<? echo $subsvalue[28]?>' />
<area shape='poly' coords='233,32,233,30,232,20,230,21,226,28,226,32' title='NH:<? echo $subsvalue[29]?>' />
<area shape='poly' coords='215,51,218,48,219,41,216,39,213,40,214,46,211,48' title='NJ:<? echo $subsvalue[30]?>' />
<area shape='rect' coords='67,61,93,85' title='NM:<? echo $subsvalue[31]?>' />
<area shape='poly' coords='195,36,214,36,220,42,227,41,221,40,222,21,215,21,207,28,197,28,193,35' title='NY:<? echo $subsvalue[32]?>' />
<area shape='poly' coords='178,70,188,70,200,76,213,68,210,62,186,62,174,70' title='NC:<? echo $subsvalue[33]?>' />
<area shape='rect' coords='89,2,120,16' title='ND:<? echo $subsvalue[34]?>' />
<area shape='poly' coords='172,51,182,53,191,47,191,34,182,37,172,37' title='OH:<? echo $subsvalue[35]?>' />
<area shape='rect' coords='106,61,129,75' title='OK:<? echo $subsvalue[36]?>' />
<area shape='rect' coords='1,18,33,36' title='OR:<? echo $subsvalue[37]?>' />
<area shape='rect' coords='191,36,214,47' title='PA:<? echo $subsvalue[38]?>' />
<area shape='poly' coords='229,40,232,40,232,38,229,37' title='RI:<? echo $subsvalue[39]?>' />
<area shape='poly' coords='189,85,178,70,188,70,200,76' title='SC:<? echo $subsvalue[40]?>' />
<area shape='rect' coords='89,16,121,31' title='SD:<? echo $subsvalue[41]?>' />
<area shape='poly' coords='187,63,151,63,149,69,174,69' title='TN:<? echo $subsvalue[42]?>' />
<area shape='poly' coords='78,85,92,99,119,114,132,96,131,76,107,73,107,62,93,62,93,85' title='TX:<? echo $subsvalue[43]?>' />
<area shape='rect' coords='46,37,67,61' title='UT:<? echo $subsvalue[44]?>' />
<area shape='poly' coords='230,21,226,28,226,32,222,32,222,21' title='VT:<? echo $subsvalue[45]?>' />
<area shape='poly' coords='210,62,178,62,184,58,188,59,192,57,194,53,202,49,206.52,205,54,209,55,214,55' title='VA:<? echo $subsvalue[46]?>' />
<area shape='rect' coords='0,0,33,16' title='WA:<? echo $subsvalue[47]?>' />
<area shape='poly' coords='188,59,192,57,194,53,202,49,191,47,182,54,186,59' title='WV:<? echo $subsvalue[48]?>' />
<area shape='poly' coords='144,28,138,23,138,16,141,12,163,20,159,33,146,33' title='WI:<? echo $subsvalue[49]?>' />
<area shape='rect' coords='59,21,89,41' title='WY:<? echo $subsvalue[50]?>' />
</map>
</td><tr></table>
</body>
</html>
```

nbiinput.php

```
/*   This code is developed for bridge inpection data entry
      Developed by Yonghong Tong. Date: 10/08/2012.
      Advisor : Shen-En Chen
 */


<!DOCTYPE html>
<html>
<head>

<title>Bridge Inventory </title>
<meta name="viewport" content="initial-scale=1.0,user-scalable=no" />
<style type="text/css">
    html {height:100%}
    body {height:100%; margin:0px; padding:0px;font-family:verdana,arial,sans-serif;font-size:10pt}
    table {font-family:verdana,arial,sans-serif;font-size:10pt}
    #map_canvas {height:200px;width:500px}
</style>

<script type="text/javascript"
    src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript" src="http://coe.uncc.edu/~ytong1/tongbridge.js"></script>

<script type="text/javascript">
    function YY(){
        cTime= new Date();
        var cYY = cTime.getFullYear();
        document.write(cYY);
    }
    function init(id,lat,lng)
    {
            var latlng=new google.maps.LatLng(lat,lng);
            var myOptions ={
                    zoom:17,
                    center:latlng,
                    mapTypeId:google.maps.MapTypeId.HYBRID
            };
            var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);
            var marker = new google.maps.Marker({map:map,title:"ID:"+id,position:latlng});
    }
</script>
<body  onload="init(id,lat,lng)">
<table align="center"><tr><td>
<?php
$id=$_GET['bridgeid'];
$bridgeid=trim($id);

include "connect.php";
$tbl_const="nbi2011";
$tbl_test="nc9210";

$result=mysql_query("SELECT item9,latdd,longdd,item21,item27,item28a,item28b,item29,item30,
```

```
        item43a,item43b,item45,item48,item49,item52,item64,item66,item67,item106,item58,item59,item
60,item90
        FROM $tbl_const where concat(stfips,region,item8)='$id' order by item90 desc limit 1");

    $item=array();
    while ($row=mysql_fetch_array($result))
    {
        for ($i=0;$i<23; $i++)
          array_push($item,$row[$i]);
    }
    ?>


    <b>Bridge ID:<?php echo $id ?></b>

    <table width="500">
    <tr><td><div id="map_canvas"></div></td></tr>
    </table>

    <table border="1"  cellspacing="0" width="500">
    <tr><td width="50%">Location:</td><td><?php echo $item[0]?></td></tr>
    <tr><td>Latitude:</td><td><?php echo $item[1]?></td></tr>
    <tr><td>Longitude:</td><td><?php echo $item[2]?></td></tr>
    <tr><td>Year Built:</td><td><?php echo $item[4]?></td></tr>
    <tr><td>Lanes On Structure:</td><td><script>trimleft0('<?php echo $item[5]?>')</script></td></tr>
    <tr><td>Lanes Under Structure:</td><td><script>trimleft0('<?php echo
$item[6]?>')</script></td></tr>
    <tr><td>Kind of Material/Design:</td><td><script type="text/javascript">kind_material('<?php echo
$item[9]?>');</script></td></tr>
    <tr><td>Type of Design/Construction:</td><td><script type="text/javascript">kind_material('<?php
echo $item[10]?>');</script></td></tr>
    <tr><td>Number Of Spans In Main Unit:</td><td><script>trimleft0('<?php echo
$item[11]?>')</script></td></tr>
    <tr><td>Length Of Maximum Span:</td><td><?php echo $item[12]/10?> meter</td></tr>
    <tr><td>Structure Length:</td><td><?php echo $item[13]/10?> meter</td></tr>
    <tr><td>Deck Width, Out-To-Out:</td><td><?php echo $item[14]/10?> meter</td></tr>
    </table><br>

    <form name="form1" action="nbiinputdata.php" target="_blank" method="post">
    <b>Bridge Condition Rating </b>
    <table border="0" bgcolor="#FFEEEE" cellspacing="0" width="500">

    <tr><td width="50%">Deck Condition Rating:</td><td>
        <select name="deck" width="250" style="width:250px">
                <option value="0">0: Failed Condition</option>
                <option value="1">1: Imminent Failure Condition</option>
                <option value="2">2: Critical Condition</option>
                <option value="3">3: Serious Condition</option>
                <option value="4">4: Poor Condition</option>
                <option value="5" selected>5: Fair Condition</option>
                <option value="6">6: Satisfactory Condition</option>
                <option value="7">7: Good Condition</option>
                <option value="8">8: Very Good Condition</option>
                <option value="9">9: Excellent Condition</option>
                <option value="N">N: Not Applicable</option>
        </select></td></tr>
```

```
<tr><td width="50%">Superstructure Condition Rating:</td><td>
    <select name="sups" width="250" style="width:250px">
            <option value="0">0: Failed Condition</option>
            <option value="1">1: Imminent Failure Condition</option>
            <option value="2">2: Critical Condition</option>
            <option value="3">3: Serious Condition</option>
            <option value="4">4: Poor Condition</option>
            <option value="5" selected>5: Fair Condition</option>
            <option value="6">6: Satisfactory Condition</option>
            <option value="7">7: Good Condition</option>
            <option value="8">8: Very Good Condition</option>
            <option value="9">9: Excellent Condition</option>
            <option value="N">N: Not Applicable</option>
    </select></td></tr>
<tr><td width="50%">Substructure Condition Rating:</td><td>
    <select name="subs" width="250" style="width:250px">
            <option value="0">0: Failed Condition</option>
            <option value="1">1: Imminent Failure Condition</option>
            <option value="2">2: Critical Condition</option>
            <option value="3">3: Serious Condition</option>
            <option value="4">4: Poor Condition</option>
            <option value="5" selected>5: Fair Condition</option>
            <option value="6">6: Satisfactory Condition</option>
            <option value="7">7: Good Condition</option>
            <option value="8">8: Very Good Condition</option>
            <option value="9">9: Excellent Condition</option>
            <option value="N">N: Not Applicable</option>
    </select></td></tr>
<tr><td width="50%">Year of Inspection:</td><td>
    <select name="year" id="year" width="250" style="width:250px">
            <option value="2006">2006</option>
            <option value="2007">2007</option>
            <option value="2008">2008</option>
            <option value="2009">2009</option>
            <option value="2010">2010</option>
            <option value="2011" selected>2011</option>
            <option value="2012">2012</option>
    </select></td></tr>

    <input type="hidden" value="<?php echo $id?>" name="id"/>
    <input type="hidden" value="<?php echo $item[1]?>" name="latdd"/>
    <input type="hidden" value="<?php echo $item[2]?>" name="longdd"/>

</table>
<table width="500px">
    <tr><td align="right" width="100%"><input type="submit" value="Submit" name="submit"
disabled="disabled"></td></tr>
</table>
</form>

<table width="500px"><tr>
<td width="100%" align="right"><input type="image"
src="http://coe.uncc.edu/~ytong1/images/print.jpg" alt="Print this page" onclick="window.print();">
<input type="image" src="http://coe.uncc.edu/~ytong1/images/close6.jpg" alt="Close window"
onclick="window.close()">
</td></tr>
```

```
</table>
<script type="text/javascript">
    var id=<? echo "'".$id."'";?>;
    var lat=<? echo $item[1];?>;
    var lng=<? echo $item[2];?>;
</script>

<?
    mysql_close($conn);
?>

</td></tr></table>
</body>
</html>
```

<u>nbiinput_edit.php</u>

```
/*   This code is developed for bridge data input edit
     Developed by Yonghong Tong. Date: 10/08/2012.
     Advisor : Shen-En Chen
 */

<!DOCTYPE html>
<html>
<head>
<title>Bridge </title>
<meta name="viewport" content="initial-scale=1.0,user-scalable=no" />
<style type="text/css">
    html {height:100%}
    body {height:100%; margin:0px; padding:0px;font-family:verdana,arial,sans-serif;font-size:10pt;}
    #map_canvas {height:97%;width:100%}
</style>

<script type="text/javascript"
    src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript" src="tongbridge.js"></script>
<script type="text/javascript">

    var map,map_cavus,img;
    var lat_c0=29.97515;  //29.43515
    var lng_c0=-88.52533; //-89.16533;
    var zoomnum=8;

    var firstON=0;
    var secondON=0;
    var markersArray=[];
    var markersArray2=[];
    var markshow=0;
    var marker;
    var MArray=[];
    var step=50,step1=50,step2=70;
    var MInit=[];
    var play=1;
    var daynight="n";
    var poly;
    var viewlat1,viewlng1,viewlat2,viewlng2,viewlatc,viewlngc;
    var num=0;
    var viewbox=[];
    var viewtrace;
     var geoLayer;
    var z;
    var bounds,bound1,bound2;
    var cTime,cHH,cMM,cSS;
    var pickedpoint;
    var speedline=[],speedlineid=[];

    var currentTime = new Date();
    var hrs = currentTime.getHours();

    var hideshow="show";
```

```
var clk="zoom";

function hideframe()
{
        if (hideshow=="show")
        {
                hideshow="hide";
//              top.frameOil.cols = "0,*";
                parent.cols="0,*";
                document.getElementById("hide").value="Show";
        }
        else
        {
                hideshow="show";
                top.frameOil.cols = "150,*";
                document.getElementById("hide").value="Hide";
        }
}


function timer1(){
    cTime= new Date();
    cHH = cTime.getHours();
    cMM = cTime.getMinutes();
    cSS = cTime.getSeconds();

    var strTime="";
    if (cHH<10) strTime=strTime+"0"+cHH+":";
      else strTime=strTime+cHH+":";

    if (cMM<10) strTime=strTime+"0"+cMM;
      else strTime=strTime+cMM;

    if (cSS<10) strTime=strTime+":0"+cSS;
          else strTime=strTime+":"+cSS;

        strTime=strTime+"     ";

        document.all.timer.innerHTML = strTime;
        setTimeout("timer1();",1000);
}


function greeting(){
        var welcome;
        if (hrs>=0 && hrs<12)
                welcome=" Good Morning ";
        else if (hrs>=12 && hrs<19)
            welcome=" Good Afternoon ";
        else
                welcome=" Good Evening ";
        document.write(welcome);
}
```

```
function rating(n)
{
	switch (n)
	{
	case "0":{str="0:Failed Condtion";break;}
	case "1":{str="1:Imminent Failure Condition";break;}
	case "2":{str="2:Critical Condition";break;}
	case "3":{str="3:Serious Condition";break;}
	case "4":{str="4:Poor Condition";break;}
	case "5":{str="5:Fair Condition";break;}
	case "6":{str="6:Satisfactory Condition";break;}
	case "7":{str="7:Good Condition";break;}
	case "8":{str="8:Very Good Condition";break;}
	case "9":{str="9:Excellent Condition";break;}
	case "N":{str="N:Not Applicable";break;}
	}
	return (str);
}
function rate(n)
{
	switch (n)
	{
	case "0":{str="0:Failed Condtion";break;}
	case "1":{str="1:Imminent Failure Condition";break;}
	case "2":{str="2:Critical Condition";break;}
	case "3":{str="3:Serious Condition";break;}
	case "4":{str="4:Poor Condition";break;}
	case "5":{str="5:Fair Condition";break;}
	case "6":{str="6:Satisfactory Condition";break;}
	case "7":{str="7:Good Condition";break;}
	case "8":{str="8:Very Good Condition";break;}
	case "9":{str="9:Excellent Condition";break;}
	case "N":{str="N:Not Applicable";break;}
	}
	document.write(str);
}


function init(id,lat,lng,lat_c,lng_c,scan)
{
	var latlng=new google.maps.LatLng(lat_c,lng_c);
	var myOptions ={
		zoom:16,   //5
		center:latlng,
		mapTypeId:google.maps.MapTypeId.HYBRID //ROADMAP
	};
	var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);

	for (var i=0;i<lat.length;i++)
	{
		img = "http://coe.uncc.edu/~ytong1/images/B"+scan[i]+".png";
		latlng=new google.maps.LatLng(lat[i],lng[i]);
		var marker = new google.maps.Marker({map:map,title:"ID:"+id[i]+" Lat:"+lat[i]+" Lng:"+lng[i],position:latlng,icon:img});

		attachmsg2(marker, id[i]);
```

```
				}
				img = "http://coe.uncc.edu/~ytong1/images/redstar_small.png";
				latlng=new google.maps.LatLng(lat_c,lng_c);
				var marker = new google.maps.Marker({map:map,title:"You are here! "+"
Latitude:"+lat_c+" Longitude:"+lng_c,position:latlng,icon:img});

		}

		function init_old(id,lat,lng,lat_c,lng_c,deck,state)
		{
				var latlng=new google.maps.LatLng(lat_c,lng_c);
				var myOptions ={
						zoom:7,
						center:latlng,
						mapTypeId:google.maps.MapTypeId.ROADMAP//HYBRID
				};
				var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);

				var kmlurl="http://coe.uncc.edu/~ytong1/nbi/kml418/"+state+".kml";
				var stateLayer = new google.maps.KmlLayer(kmlurl);
				stateLayer.setMap(map);

				for (var i=0;i<lat.length;i++)
				{
				//		alert(lat[i]);
						img = "http://coe.uncc.edu/~ytong1/images/rate0"+deck[i]+".png";
						latlng=new google.maps.LatLng(lat[i],lng[i]);
						var marker = new
google.maps.Marker({map:map,title:"ID:"+id[i]+";"+rating(deck[i]),position:latlng,icon:img});

						attachmsg(marker, id[i]);
				/*
						google.maps.event.addListener(marker, 'click', function() {
								alert(id[i]);
								alert("hello");
						});
				*/
				}
		}


		function attachmsg(marker, number) {
				google.maps.event.addListener(marker, 'click', function() {

window.open('nbibridgedispedit.php?bridgeid='+number,'open','menubar=no,width=520,height=780,left=0,
top=0,scrollbars=1,toolbar=no');
				});
		}

		function attachmsg2(marker, number) {
				google.maps.event.addListener(marker, 'click', function() {

window.open('nbiinput.php?bridgeid='+number,'open','menubar=no,width=520,height=780,left=0,top=0,sc
rollbars=1,toolbar=no');
				});
		}
```

```
	function test(var1,var2)
	{
//		alert(txt);
		for (var i=0; i<var1.length;i++)
			document.write(var1[i]+'<br>');
	//	document.write(var2[0]);
	}

</script></head>
<body onload="init(javaid,javalat,javalng,lat_center,lng_center,javascan);timer1();">
<?php

$lat0=$_POST['lat0'];
$lng0=$_POST['lng0'];
$delta=$_POST['delta'];

if ($lat0==0)
{
	$lat0=35.27;
	$lng0=-80.85;
	$delta=0.0072;
}

$latddmax=69.150556;
$latddmin=17.951389;
$lngddmax=-65.391667;
$lngddmin=-174.14;

if ($lat0>$latddmax) $lat0=$latddmax;
if ($lat0<$latddmin)  $lat0=$latddmin;
if ($lng0>$lngddmax)		$lng0=$lngddmax;
if ($lng0<$lngddmin)		$lng0=$lngddmin;



//*****  NEW

include "connect.php";
$tbl="nbi";

$result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng  FROM $tbl where sqrt((latdd-$lat0)*(latdd-$lat0)+(longdd-$lng0)*(longdd-$lng0))<$delta");

$result_maxlat=mysql_query("SELECT MAX(latdd) FROM $tbl where abs(latdd-$lat0)<$delta and abs(longdd-$lng0)<$delta");
$result_minlat=mysql_query("SELECT MIN(latdd) FROM $tbl where abs(latdd-$lat0)<$delta and abs(longdd-$lng0)<$delta");
$result_maxlng=mysql_query("SELECT MAX(longdd) FROM $tbl where abs(latdd-$lat0)<$delta and abs(longdd-$lng0)<$delta");
$result_minlng=mysql_query("SELECT MIN(longdd) FROM $tbl where abs(latdd-$lat0)<$delta and abs(longdd-$lng0)<$delta");

//****** NEW end
```

```
$maxlat=mysql_fetch_array($result_maxlat);
$minlat=mysql_fetch_array($result_minlat);
$maxlng=mysql_fetch_array($result_maxlng);
$minlng=mysql_fetch_array($result_minlng);

 $lat_center=$lat0;
 $lng_center=$lng0;

$id_array=array();
$lat_array=array();
$lng_array=array();

$scan_array=array();

$a=count($lat_array);
while ($row=mysql_fetch_array($result))
{
    $id  =$row['id'];
    $lat =$row['lat'];
    $lng =$row['lng'];

    $resultscan=mysql_query("SELECT * FROM uncctest WHERE id='$id'");
    $scan=mysql_num_rows($resultscan);

    array_push($id_array,$id);
    array_push($lat_array,$lat);
    array_push($lng_array,$lng);
    array_push($scan_array,$scan);
}

echo "Total ".count($lat_array)." bridges found.<br>";
?>
<script type="text/javascript">
    var javaid=[];
    var javalat=[];
    var javalng=[];
    var javascan=[];
    var lat_center;
    var lng_center;
    lat_center=<? echo $lat_center;?>;
    lng_center=<? echo $lng_center;?>;
    <?
     for ($i=0;$i<count($lat_array);$i++)
        {
            echo  'javaid['.$i.']="'.$id_array[$i].'";';
            echo 'javalat['.$i.']="'.$lat_array[$i].'";';//'Jan';
            echo 'javalng['.$i.']="'.$lng_array[$i].'";';
            echo 'javascan['.$i.']="'.$scan_array[$i].'";';
        }
    ?>
</script>
<div id="map_canvas"></div>
<?
    mysql_close($conn);
?>
</body></html>
```

nbiinput_edit_dms.php

```
/*  This code is developed for bridge data input edit by latitude and longitude with degree, minute
    and second  format.
    Developed by Yonghong Tong. Date: 10/08/2012.
    Advisor : Shen-En Chen
 */

<!DOCTYPE html>
<html>
<head>
<title>Bridge </title>
<meta name="viewport" content="initial-scale=1.0,user-scalable=no" />
<style type="text/css">
    html {height:100%}
    body {height:100%; margin:0px; padding:0px;font-family:verdana,arial,sans-serif;font-size:10pt;}
    #map_canvas {height:100%;width:100%}
</style>

<script type="text/javascript"
    src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript" src="tongbridge.js"></script>
<script type="text/javascript">

    var map,map_cavus,img;
    var lat_c0=29.97515;  //29.43515
    var lng_c0=-88.52533; //-89.16533;
    var zoomnum=8;

    var firstON=0;
    var secondON=0;
    var markersArray=[];
    var markersArray2=[];
    var markshow=0;
    var marker;
    var MArray=[];
    var step=50,step1=50,step2=70;
    var MInit=[];
    var play=1;
    var daynight="n";
    var poly;
    var viewlat1,viewlng1,viewlat2,viewlng2,viewlatc,viewlngc;
    var num=0;
    var viewbox=[];
    var viewtrace;
    var geoLayer;
    var z;
    var bounds,bound1,bound2;
    var cTime,cHH,cMM,cSS;
    var pickedpoint;
    var speedline=[],speedlineid=[];

    var currentTime = new Date();
    var hrs = currentTime.getHours();
```

```
var hideshow="show";

var clk="zoom";

function hideframe()
{
            if (hideshow=="show")
            {
                        hideshow="hide";
//                      top.frameOil.cols = "0,*";
                        parent.cols="0,*";
                        document.getElementById("hide").value="Show";
            }
            else
            {
                        hideshow="show";
                        top.frameOil.cols = "150,*";
                        document.getElementById("hide").value="Hide";
            }
}


function timer1(){
    cTime= new Date();
    cHH = cTime.getHours();
    cMM = cTime.getMinutes();
    cSS = cTime.getSeconds();

    var strTime="";
    if (cHH<10) strTime=strTime+"0"+cHH+":";
      else strTime=strTime+cHH+":";

    if (cMM<10) strTime=strTime+"0"+cMM;
      else strTime=strTime+cMM;

    if (cSS<10) strTime=strTime+":0"+cSS;
            else strTime=strTime+":"+cSS;

            strTime=strTime+"     ";

            document.all.timer.innerHTML = strTime;
            setTimeout("timer1();",1000);
}


function greeting(){
            var welcome;
            if (hrs>=0 && hrs<12)
                        welcome=" Good Morning ";
            else if (hrs>=12 && hrs<19)
                  welcome=" Good Afternoon ";
            else
                        welcome=" Good Evening ";
            document.write(welcome);
}
```

```
function rating(n)
{
        switch (n)
        {
        case "0":{str="0:Failed Condtion";break;}
        case "1":{str="1:Imminent Failure Condition";break;}
        case "2":{str="2:Critical Condition";break;}
        case "3":{str="3:Serious Condition";break;}
        case "4":{str="4:Poor Condition";break;}
        case "5":{str="5:Fair Condition";break;}
        case "6":{str="6:Satisfactory Condition";break;}
        case "7":{str="7:Good Condition";break;}
        case "8":{str="8:Very Good Condition";break;}
        case "9":{str="9:Excellent Condition";break;}
        case "N":{str="N:Not Applicable";break;}
        }
        return (str);
}
function rate(n)
{
        switch (n)
        {
        case "0":{str="0:Failed Condtion";break;}
        case "1":{str="1:Imminent Failure Condition";break;}
        case "2":{str="2:Critical Condition";break;}
        case "3":{str="3:Serious Condition";break;}
        case "4":{str="4:Poor Condition";break;}
        case "5":{str="5:Fair Condition";break;}
        case "6":{str="6:Satisfactory Condition";break;}
        case "7":{str="7:Good Condition";break;}
        case "8":{str="8:Very Good Condition";break;}
        case "9":{str="9:Excellent Condition";break;}
        case "N":{str="N:Not Applicable";break;}
        }
        document.write(str);
}


function init(id,lat,lng,lat_c,lng_c,scan)
{
        var latlng=new google.maps.LatLng(lat_c,lng_c);
        var myOptions ={
                zoom:16,   //5
                center:latlng,
                mapTypeId:google.maps.MapTypeId.HYBRID //ROADMAP
        };
        var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);

        for (var i=0;i<lat.length;i++)
        {
                img = "http://coe.uncc.edu/~ytong1/images/B"+scan[i]+".png";
                latlng=new google.maps.LatLng(lat[i],lng[i]);
                var marker = new google.maps.Marker({map:map,title:"ID:"+id[i]+"
Lat:"+lat[i]+" Lng:"+lng[i],position:latlng,icon:img});
```

```
                        attachmsg2(marker, id[i]);
                }
                img = "http://coe.uncc.edu/~ytong1/images/redstar_small.png";
                latlng=new google.maps.LatLng(lat_c,lng_c);
                var marker = new google.maps.Marker({map:map,title:"You are here! "+"
Latitude:"+lat_c+" Longitude:"+lng_c,position:latlng,icon:img});

        }

        function init_old(id,lat,lng,lat_c,lng_c,deck,state)
        {
                var latlng=new google.maps.LatLng(lat_c,lng_c);
                var myOptions ={
                        zoom:7,
                        center:latlng,
                        mapTypeId:google.maps.MapTypeId.ROADMAP//HYBRID
                };
                var map=new google.maps.Map(document.getElementById("map_canvas"),myOptions);

                var kmlurl="http://coe.uncc.edu/~ytong1/nbi/kml418/"+state+".kml";
                var stateLayer = new google.maps.KmlLayer(kmlurl);
                stateLayer.setMap(map);

                for (var i=0;i<lat.length;i++)
                {
                        img = "http://coe.uncc.edu/~ytong1/images/rate0"+deck[i]+".png";
                        latlng=new google.maps.LatLng(lat[i],lng[i]);
                        var marker = new
google.maps.Marker({map:map,title:"ID:"+id[i]+";"+rating(deck[i]),position:latlng,icon:img});

                        attachmsg(marker, id[i]);
                }
        }


        function attachmsg(marker, number) {
                 google.maps.event.addListener(marker, 'click', function() {

window.open('nbibridgedispedit.php?bridgeid='+number,'open','menubar=no,width=520,height=780,left=0,
top=0,scrollbars=1,toolbar=no');
                 });
        }

        function attachmsg2(marker, number) {
                 google.maps.event.addListener(marker, 'click', function() {

window.open('nbiinput.php?bridgeid='+number,'open','menubar=no,width=520,height=780,left=0,top=0,sc
rollbars=1,toolbar=no');
                 });
        }


        function test(var1,var2)
        {
                for (var i=0; i<var1.length;i++)
                        document.write(var1[i]+'<br>');
```

```
    }

</script></head>
<body onload="init(javaid,javalat,javalng,lat_center,lng_center,javascan);timer1();">
<?php

$latd=$_POST['latd'];
$latm=$_POST['latm'];
$lats=$_POST['lats'];

$lngd=$_POST['lngd'];
$lngm=$_POST['lngm'];
$lngs=$_POST['lngs'];

$lat0=$latd+$latm/60+$lats/3600;
$lng0=-($lngd+$lngm/60+$lngs/3600);

// echo $lat0;

$delta=0.007;

if ($lat0==0)
{
    $lat0=35.27;
    $lng0=-80.85;
    $delta=0.0072;
}

$latddmax=69.150556;
$latddmin=17.951389;
$lngddmax=-65.391667;
$lngddmin=-174.14;

if ($lat0>$latddmax) $lat0=$latddmax;
if ($lat0<$latddmin)  $lat0=$latddmin;
if ($lng0>$lngddmax)            $lng0=$lngddmax;
if ($lng0<$lngddmin)            $lng0=$lngddmin;

//*****  NEW
include "connect.php";
$tbl="nbi";

    $result=mysql_query("SELECT concat(stfips,region,item8) as id,latdd as lat,longdd as lng  FROM
$tbl where sqrt((latdd-$lat0)*(latdd-$lat0)+(longdd-$lng0)*(longdd-$lng0))<$delta");

    $result_maxlat=mysql_query("SELECT MAX(latdd) FROM $tbl where abs(latdd-$lat0)<$delta and
abs(longdd-$lng0)<$delta");
    $result_minlat=mysql_query("SELECT MIN(latdd) FROM $tbl where abs(latdd-$lat0)<$delta and
abs(longdd-$lng0)<$delta");
    $result_maxlng=mysql_query("SELECT MAX(longdd) FROM $tbl where abs(latdd-$lat0)<$delta
and abs(longdd-$lng0)<$delta");
    $result_minlng=mysql_query("SELECT MIN(longdd) FROM $tbl where abs(latdd-$lat0)<$delta and
abs(longdd-$lng0)<$delta");

//****** NEW end
```

```
$maxlat=mysql_fetch_array($result_maxlat);
$minlat=mysql_fetch_array($result_minlat);
$maxlng=mysql_fetch_array($result_maxlng);
$minlng=mysql_fetch_array($result_minlng);

$lat_center=$lat0;
 $lng_center=$lng0;

$id_array=array();
$lat_array=array();
$lng_array=array();

$scan_array=array();

$a=count($lat_array);
while ($row=mysql_fetch_array($result))
{
    $id  =$row['id'];
    $lat =$row['lat'];
    $lng =$row['lng'];

    $resultscan=mysql_query("SELECT * FROM uncctest WHERE id='$id'");
    $scan=mysql_num_rows($resultscan);

    array_push($id_array,$id);
    array_push($lat_array,$lat);
    array_push($lng_array,$lng);
    array_push($scan_array,$scan);
}
?>
<script type="text/javascript">
    var javaid=[];
    var javalat=[];
    var javalng=[];
    var javascan=[];
    var lat_center;
    var lng_center;
    lat_center=<? echo $lat_center;?>;
    lng_center=<? echo $lng_center;?>;

    <?
     for ($i=0;$i<count($lat_array);$i++)
       {
              echo  'javaid['.$i.']="'.$id_array[$i].'";';
              echo 'javalat['.$i.']="'.$lat_array[$i].'";';//'Jan';
              echo 'javalng['.$i.']="'.$lng_array[$i].'";';
              echo 'javascan['.$i.']="'.$scan_array[$i].'";';
       }
    ?>
</script>
<div id="map_canvas"></div>
<?
    mysql_close($conn);
?>
</body>
</html>
```