Chapter 23
Augmented Reality for Next Generation Infrastructure Inspections

JoAnn P. Ballor, Oscar L. McClain, Miranda A. Mellor, Alessandro Cattaneo, Troy A. Harden, Philo Shelton, Eric Martinez, Ben Narushof, Fernando Moreu, and David D. L. Mascareñas

Abstract Most infrastructure in America is approaching or has exceeded the intended design life. As infrastructure ages, danger is presented to the citizens and the environment. Disasters like the failure of the Oroville Dam emphasize the importance of preventing failure of infrastructure. Despite Structural Health Monitoring (SHM) emerging as the key discipline to evaluate the integrity of structures, human-based visual inspections remain the dominant technique. Main factors behind the slow adoption of the SHM technology are its high installation and maintenance costs, the low confidence level of decision makers in the technology, and the structural community's familiarity with visual inspections. One major drawback to visual inspection is that it can be highly subjective, relying on a small number of manually recorded data points. Inspectors use tape measures, digital cameras, and human senses. These tools are inadequate to capture and document the state of infrastructure at the resolution needed for tracking damage progression. When working with infrastructure, it is desirable to have high-resolution measurement techniques to detect, localize, and characterize the state of a structure. The present work investigates Augmented Reality (AR) to minimize visual inspection drawbacks. AR is a technology that can superimpose holograms onto the real world. In this work an AR headset is used to enhance the ability of human inspectors to perform infrastructure inspections. The headset features RGB and depth cameras, accelerometers, wireless communication, microphones, and stereo sound. These sensors can capture a high-resolution 3D measurement of the infrastructure being inspected along with RGB photos that can be overlaid on the geometry. The 3D measurement can be used to analyze the state of the structure over time, and track damage progression. The 3D model can also be used as a base to overlay data captured with diverse sensors. The proposed technique will ultimately improve inspectors' ability to assess risk on site, make informed decisions faster, analyze defect growth over time, and create high-resolution documentation of structural inspections that will reduce the variability that occurs. Overall, AR is a promising technique for infrastructure inspection, mapping infrastructure on site, generating models, and providing consistency between inspectors. This technology has considerable potential for improving the quality of infrastructure inspections.

J. P. Ballor
Michigan State University, Materials Engineering, East Lansing, MI, USA
e-mail: ballorj2@msu.edu

O. L. McClain
Southern University and A&M College, Chemistry, Baton Rouge, LA, USA
e-mail: oscar_mcclain_00@subr.edu

M. A. Mellor
University of California San Diego, Structural Engineering, La Jolla, CA, USA
e-mail: mmellor@ucsd.edu

A. Cattaneo (✉) · T. A. Harden
Los Alamos National Laboratory, Applied Engineering Technologies, Los Alamos, NM, USA
e-mail: cattaneo@lanl.gov; harden@lanl.gov

P. Shelton · E. Martinez · B. Narushof
Engineering Division, Public Works Department Los Alamos County, Los Alamos, NM, USA
e-mail: philo.shelton@lacnm.us; eric.martinez@lacnm.us; ben.narushof@lacnm.us

F. Moreu
Department of Civil Engineering, University of New Mexico, Centennial Engineering Center 3056 MSC01 1070, 210 University Blvd NE, Albuquerque, NM, USA
e-mail: fmoreu@unm.edu

D. D. L. Mascareñas
Los Alamos National Laboratory, Engineering Institute, Los Alamos, NM, USA
e-mail: dmascarenas@lanl.gov

R. Barthorpe (ed.), *Model Validation and Uncertainty Quantification, Volume 3*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.1007/978-3-319-74793-4_23

Keywords Augmented reality · Structural inspection · Structural health monitoring · Non-destructive evaluation · Microsoft HoloLens

23.1 Introduction

A large portion of infrastructure in America has exceeded or will exceed its intended design life in the following years [1]. For example, New Mexico is the home of many 100-year-old fracture-critical steel railroad bridges [2]. These bridges present the possibility of failure without regular and detailed inspections [2].

Although human-based inspections are largely adopted, a number of factors are negatively affecting the results obtained with this approach. Variations in training and experience, as well the information inspectors choose to report after an inspection may cause inaccuracies and lack of consistency when the same infrastructure is inspected by different teams [3, 4].

In the field of engineering, there is a push to embrace new technology to help engineers produce the best work possible [5]. Augmented reality, a technology which allows the user to see computer generated images superimposed on the real world, has the potential to minimize inspection inconsistencies. The augmented reality device used in this paper is the Microsoft HoloLens [6]. Augmented reality differs from virtual reality by allowing the user to be able to see the real world with overlaid holographic images (holograms).

This paper will explore augmented reality capabilities in order to create a tool for more consistent inspections. Using the capabilities of the HoloLens, a model of a structure can be created. Inspection data and measurements made by the HoloLens can then be combined with the model. This model can be used by future inspectors, improving consistency of inspections over time. The experimental procedure and results obtained with the current work are presented in the next sections.

23.2 Experimental Procedures

In the field of infrastructure inspection there is a need for improved measurement tools and a standardized system for reporting results. The following apps were developed to address the current needs of the inspection community. The Microsoft HoloLens was used as the augmented reality headset. All the apps presented in the current section were created with the help of the game development platform Unity and Microsoft Visual Studio. The apps created are categorized as either measurement apps or data storage apps. The measurement apps were created for the purpose of improving accuracy and efficiency of inspection measurement, as well as improving measurement reporting. Data storage apps were developed to store information about a structure which can then be reviewed, shared between inspectors, and used in additional data analysis.

23.2.1 Measurement Apps

23.2.1.1 Surface Area App

The surface area measuring app was designed to calculate the surface area of specific regions of sidewalks. The types of surface areas the app would need to measure are horizontal and vertical surface areas, to measure flat sidewalks and curbs. To accomplish these tasks, the app needs to be capable of measuring irregular shapes, such as multi-sided polygons. In order to address these needs, two algorithms were developed to measure horizontal and vertical surface areas. The user can define these surface areas through the placement of holograms to define the edges of the surface areas. The remaining of the current subsection provides details about the implementation of the horizontal and vertical surface area measurement algorithms, as well as a complete description of the user feedback for hologram placement and the output files generated by the app. Limitations of the current surface area measurements algorithms are highlighted.

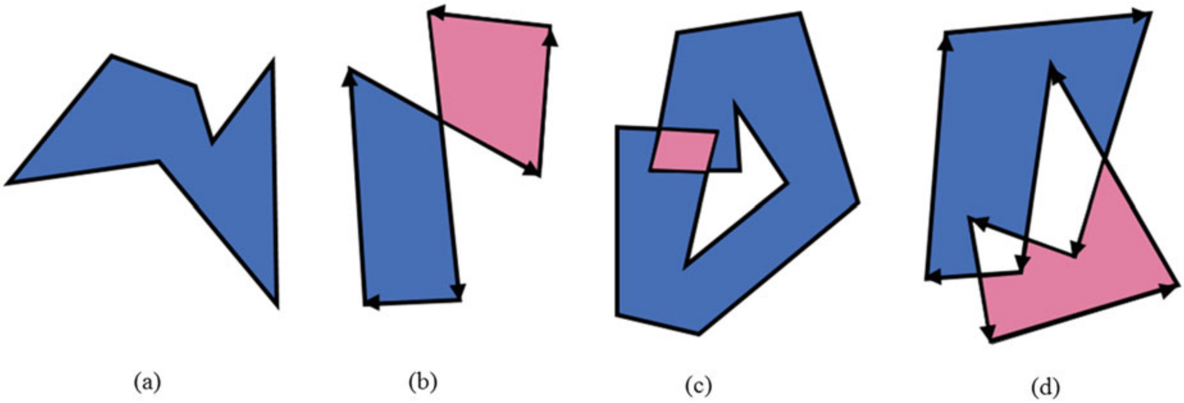

Fig. 23.1 Figure (**a**) is an example of a polygon where the algorithm can calculate the area [7]. Figures (**b**–**d**) are examples of polygons the horizontal algorithm is not able to calculate [7]

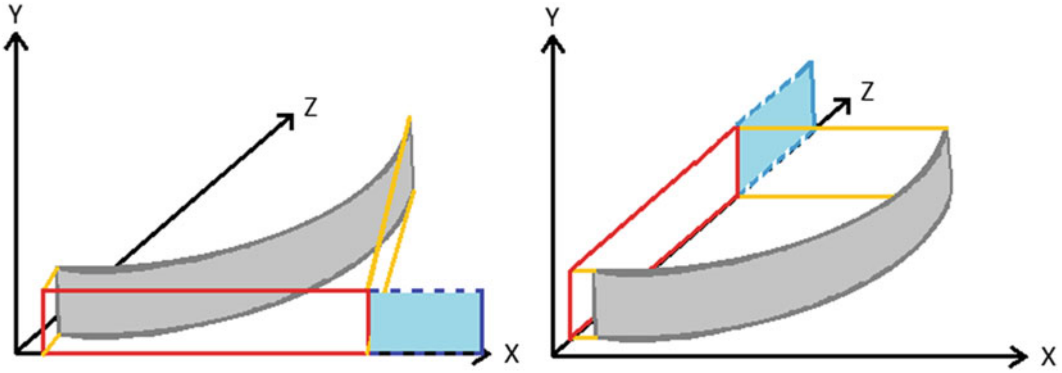

Fig. 23.2 An example of how vertical surface area could be lost to projection error

Horizontal Surface Area Measurement Algorithm

The polygon surface area algorithm by Finley was adapted into C# code to measure the surface area of horizontal polygons [7]. Using the x- and z-coordinates of user-placed holographic markers, the horizontal surface area of the defined polygon can be calculated [7]. Figure 23.1 shows an example of the type of polygon area that can be calculated. X- and z-coordinates were chosen because the HoloLens defines the x-z plane as a horizontal plane. This approach would meet the need to measure horizontal surface areas with irregular shapes. Limitations of this algorithm are that the polygon cannot cross itself, as seen in Fig. 23.1. Also it should be noted that the user-defined surface area is projected onto the x-z plane, ignoring any vertical gradient present. This projection introduces some error into the measurement.

Vertical Surface Area Measurement Algorithm

The horizontal polygon algorithm could not be adapted to a vertical algorithm because it takes coordinate points from 2 planes, so a vertical surface area, such as a curb, would have to be projected on either the x-y or z-y plane. If an angle existed between the curb and the projected plane, significant surface area could be lost. This is illustrated in Fig. 23.2. The vertical algorithm was developed to limit vertical surface area projection error.

The user-placed vertical surface markers consist of “top” and “bottom” markers, which define the top edge and the bottom edge of the vertical surface to be measured. The horizontal distance between top markers is found using the distance formula and the x- and z-components of the vector between two markers, as in Eq. 23.1.

$$d = \sqrt{(x_2 - x_1)^2 + (z_2 - z_1)^2} \tag{23.1}$$

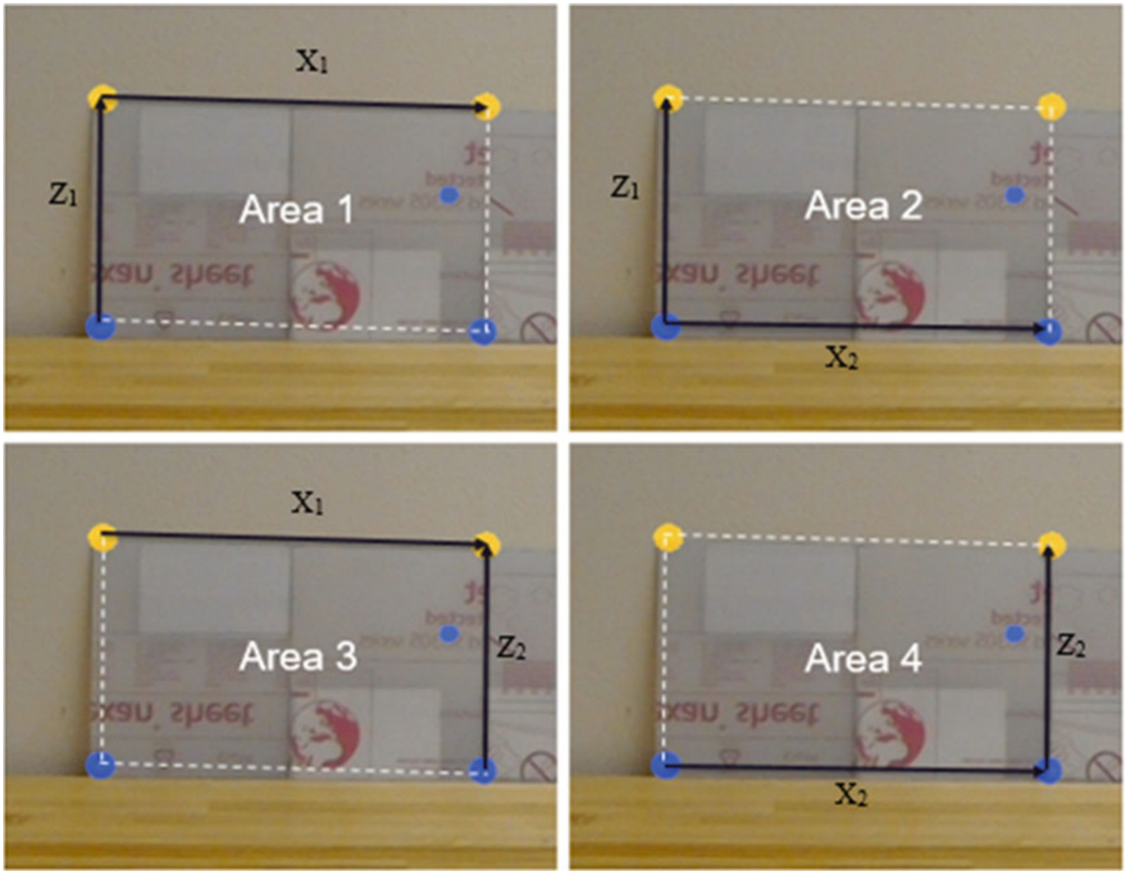

Fig. 23.3 An example of 2 top and 2 bottom markers placed to define a vertical surface area. Also shown are the four areas calculated to get a final average surface area

Similarly, the distance between the bottom markers is computed. The vertical distance between the top and bottom markers is found by taking the y-component of the vector connecting them. With the markers placed as shown in Fig. 23.3, the area of four rectangles defined by the top and bottom markers is averaged to get a final area.

$$Area\ 1 = X_1 * Z_1 \tag{23.2}$$

$$Area\ 2 = X_2 * Z_1 \tag{23.3}$$

$$Area\ 3 = X_1 * Z_2 \tag{23.4}$$

$$Area\ 4 = X_2 * Z_2 \tag{23.5}$$

$$Final\ Area = \frac{Area\ 1 + Area\ 2 + Area\ 3 + Area\ 4}{4} \tag{23.6}$$

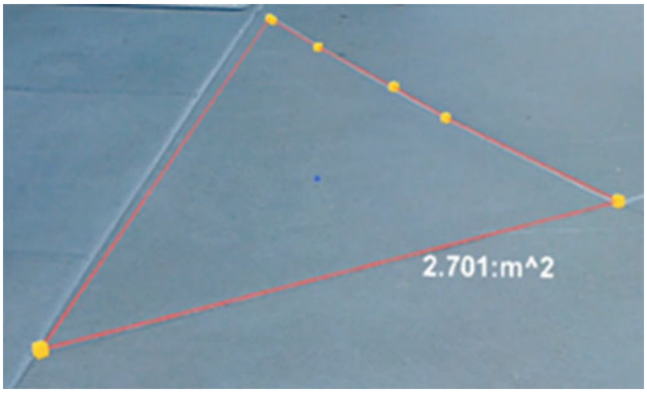

Fig. 23.4 An example of what the user would see when defining a horizontal surface area using the Surface Area App

The surface area measurement algorithm still contains some inherent error because the height of the area is found using only the vector y-component; angles between the surface area and the y-axis are ignored. Also, the algorithm assumes that the user-defined area is a rectangle. This limits the complexity of vertical surface area shapes that can be measured.

User Feedback and Output Files

Three types of markers were created for the surface area app. A yellow cube with side lengths of 3 cm is the marker for defining horizontal surface area. A yellow sphere with a diameter of 3 cm is the marker for defining the top edge of a vertical surface, and a blue sphere with a 3 cm diameter is the marker used for defining the bottom edge of a vertical surface. Visual feedback from both the shape and the color of the marker helps the user keep track of the type of markers he/she has already placed. When calculating surface area, the markers are connected with a red line to provide the user with feedback on the total surface area the app is measuring. The surface area calculated is also instantly displayed in a textbox on screen for the user to view. An example of what the user would see is shown in Fig. 23.4.

Text files with date and time information, marker coordinate positions, and the calculated surface areas are generated automatically for the user. These files allow the user to engage in extra data processing if desired.

23.2.1.2 Depth App

A depth measuring app was developed as a tool for evaluating structural health. The depth of a crack or pothole is one indicator of structural deterioration and damage progression. To properly monitor structural health, damage must be periodically quantified. The depth app aims at allowing the user to efficiently measure and record the magnitude of damage to a structure. Details about the depth measurement algorithm, user feedback and output files are provided hereafter.

Depth Measurement Algorithm

Depth is calculated using 4 user-defined points. The first three markers are placed in order to establish a reference plane for the depth measurement. The last point is placed at the depth that the user wants to measure. The shortest vector is then calculated between the depth marker and the defined plane. The magnitude of that vector is reported to the user as the depth. Figure 23.5 illustrates the three steps taken by the depth measurement algorithm and a screenshot of the scene viewed by the user.

User Feedback and Output Files

Once four markers have been placed the depth will be calculated. The depth is instantly displayed in a textbox on screen for the user to view. Similar to the Surface Area App, text files with date and time information, marker coordinate positions, and the calculated depth are generated automatically for the user. These files allow the user to engage in extra data processing if desired.

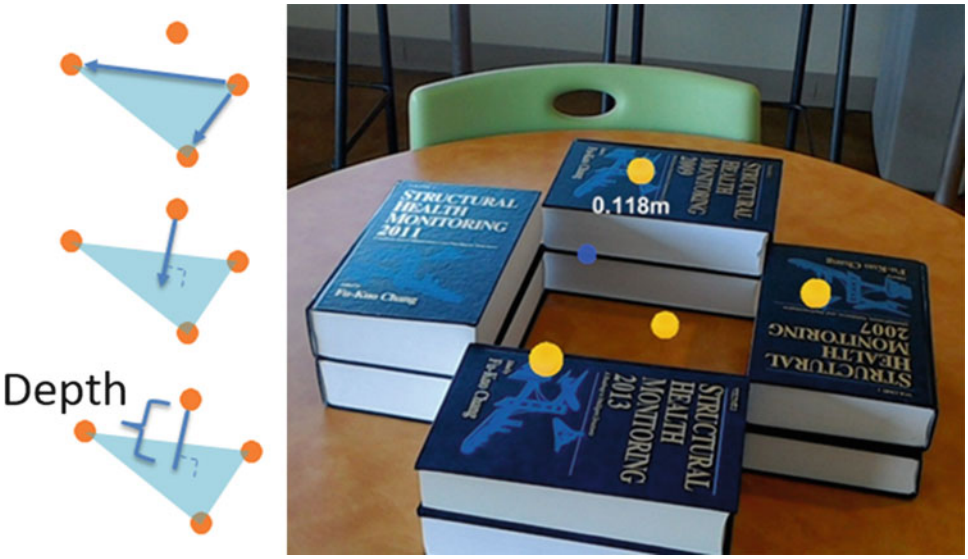
Depth
0.118m
STRUCTURAL HEALTH MONITORING 2011
STRUCTURAL HEALTH MONITORING 2009
STRUCTURAL HEALTH MONITORING 2007
STRUCTURAL HEALTH MONITORING 2013

Fig. 23.5 An illustration of how depth is calculated along with what the user would see when the app is in use

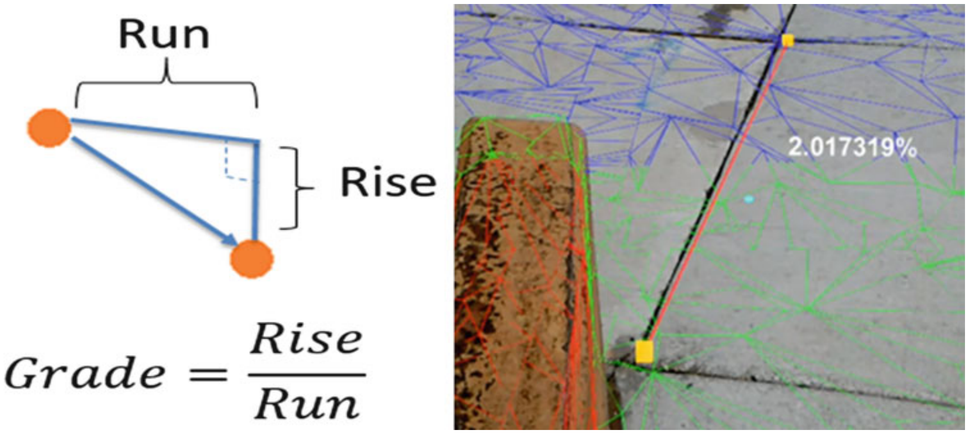

Fig. 23.6 An illustration of how grade is calculated along with what the user would see when the app is in use

23.2.1.3 Grade App

Grade is calculated as the amount of vertical height change over a horizontal distance, which can be calculated as the slope of a line. This is found by creating a ratio between the horizontal and vertical change between set points. The purpose of this app is to allow the user to calculate the grade of a sidewalk to determine if it meets building code. Details about the grade app are provided in the next two subsections.

Grade Algorithm

The grade algorithm works by finding the vector that connects two user-defined points. The slope of the vector is calculated by using the y-component of the vector as the “rise” of the slope. The “run” is calculated by using the x- and z-components of the vector and the distance formula from Eq. 23.1. Grade is then found by dividing the “rise” by the “run”. Since grade is reported as a percentage, the rise over run is then multiplied by 100%. Figure 23.6 illustrates how grade is calculated as well as what the user sees when using the app.

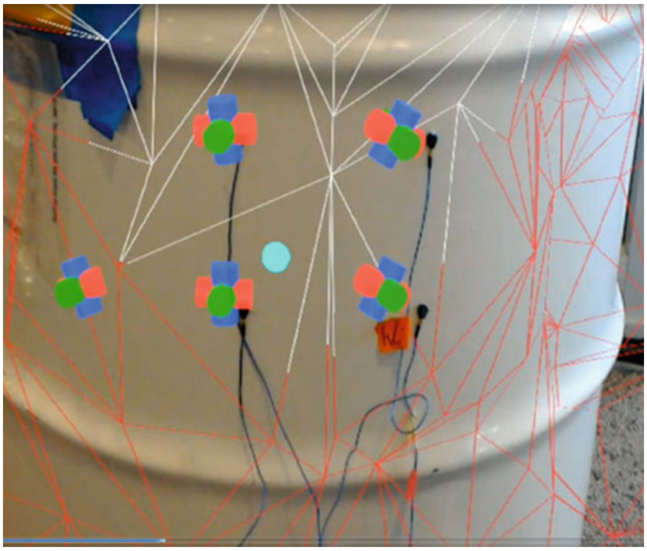

Fig. 23.7 Accelerometer App markers placed on a barrel with accelerometers attached

User Feedback and Output Files

Once two markers have been placed, a grade can then be calculated. The grade is instantly displayed in a textbox on screen for the user to view. Text files with date and time information, marker coordinate positions, and the calculated grade are generated automatically for the user. These files allow the user to engage in extra data processing if desired.

23.2.2 Data Storage Apps

23.2.2.1 Accelerometer Location App

The accelerometer location app is meant to be used as a tool for documenting accelerometer location and orientation on a structure for the purpose of modal analysis. The app enables the user to place a holographic marker onto a structure to permanently record the location and orientation of an accelerometer at its installation. User feedback and output files details follow.

User Feedback and Output Files

The marker created consists of three cylinders which are aligned to form axes as seen in Fig. 23.7. These cylinders are also color coordinated to correspond to the axes defined by Unity. When a marker is placed, the y-axis is automatically aligned to be normal to the surface that it is placed on. Once the user has placed all markers at the desired locations and orientations of each accelerometer on the structure, a text file is created. This text file contains individual marker locations in local coordinates created by the HoloLens. The text file also outputs the orientation vectors for each marker based upon the respective orientation of their axes.

23.3 Future Work

After creating the apps previously mentioned, the team found several areas of structural inspection where augmented reality could also be of use. The two main areas of future potential were found to be modal analysis, particularly accelerometer placement, and file storage of inspection results. With regard to the first point, a model of the structure could be created on the computer, and accelerometers could be placed on the model. The model would then be projected on the real structure using augmented reality, showing the user where and in what orientation the accelerometers should be placed. With regard to the second point, the HoloLens can be programmed to generate world anchor holograms. A world anchor is a type of hologram specific to the HoloLens that can be placed by the user, and, once placed, this hologram remains permanently at this defined location. If the user reopens the app at a later time, the world anchor would appear in the same location where it was saved. As a result of its ability to have a permanent location, world anchors can be used as a tool for information storage.

23.4 Conclusions

Augmented reality has the potential to help structural inspections meet the needs of a constantly growing and developing modern world dependent on its infrastructure. The research presented shows how the game development platform Unity and Microsoft Visual Studio can be successfully used to develop a set of apps to enable infrastructure inspections with a Microsoft HoloLens augmented reality headset.

Current inspections are subject to variability, and are in need of a standardized system for reporting inspection results. An answer to this need can be provided with augmented reality apps created to perform inspection measurements, such as measuring surface area and grade, as well as to store information pertaining to a structure, such as accelerometer placement.

Aside from the apps already created, there is potential for the development of future apps to further explore the measurement and data storage capabilities of augmented reality.

Acknowledgements This project was funded through the Los Alamos Dynamics Summer School program of the Los Alamos National Laboratory Engineering Institute under the direction of Charles Farrar. This work would not have been possible without the help of the Los Alamos County, Sudeep Dasari, Ben Katko, and Brian Bleck.

References

1. ASCE Committee on America's Infrastructure: 2017 Infrastructure Report Card, American Society of Civil Engineers (2017)
2. Daumueller, A.N., Jauregui, D.V., Roach, D.P.: Development of a Structural Health Monitoring System for Life Assesment of Critical Transportation Infrastructure. Sandia National Laboratories, Albuquerque (2012)
3. Washer, G.A., Chang, C.A.: Guideline for Implementing Quality Control and Quality Assurance for Bridge Inspection, June 2009. [Online]. Available: http://onlinepubs.trb.org/onlinepubs/nchrp/docs/NCHRP20-07(252)_FR.pdf. Accessed June 2017
4. Agrawal, A.K., Washer, G.A.: Consistency of the New York State Bridge Inspection Program, Albany (2013)
5. Lab, F.F.: Reflecting on the Future of Design at the IABSE Conference, 2017. [Online]. Available: https://formfindinglab.wordpress.com/2017/05/03/relecting-on-the-future-of-design-at-the-iabse-conference/. Accessed June 2017
6. Microsoft: HoloLens for Research, Microsoft, 2017. [Online]. Available: https://www.microsoft.com/en-us/research/academic-program/hololens-for-research/#. Accessed 5 July 2017
7. Finley, D.R.: Ultra-Easy Polygon Area Algorithm With C Code Sample, 2006. [Online]. Available: http://alienryder.com/polygon_area/. Accessed June 2017