Sensitivity analysis of augmented reality-assisted building damage reconnaissance using virtual prototyping

Suyang Dong, Chen Feng, Vineet R. Kamat *

Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor 48109-2125, United States

**A R T I C L E   I N F O**

Article history:
Accepted 9 September 2012
Available online 23 October 2012

Keywords:
Building damage
Earthquake
Reconnaissance
Augmented reality
Line segment detection
Nondestructive evaluation

**A B S T R A C T**

The timely and accurate assessment of the damage sustained by a building during catastrophic events, such as earthquakes or blasts, is critical in determining the building's structural safety and suitability for future occupancy. Among many indicators proposed for measuring structural integrity, especially inelastic deformations, Interstory Drift Ratio (IDR) remains the most trustworthy and robust metric at the story level. In order to calculate IDR, researchers have proposed several nondestructive measurement methods. Most of these methods rely on pre-installed target panels with known geometric shapes or with an emitting light source. Such target panels are difficult to install and maintain over the lifetime of a building. Thus, while such methods are nondestructive, they are not entirely non-contact. This paper proposes an Augmented Reality (AR)-assisted non-contact method for estimating IDR that does not require any pre-installed physical infrastructure on a building. The method identifies corner locations in a damaged building by detecting the intersections between horizontal building baselines and vertical building edges. The horizontal baselines are superimposed on the real structure using an AR algorithm, and the building edges are detected via a Line Segment Detection (LSD) approach. The proposed method is evaluated using a Virtual Prototyping (VP) environment that allows testing of the proposed method in a reconfigurable setting. A sensitivity analysis is also conducted to evaluate the effect of instrumentation errors on the method's practical use. The experimental results demonstrate the potential of the new method to facilitate rapid building damage reconnaissance, and highlight the instrument precision requirements necessary for practical field implementation.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Rapid and accurate evaluation approaches are essential for determining a building’s structural integrity for future occupancy following a major seismic event. The elapsed time could translate into private financial loss or even a public welfare crisis. Current inspection practices usually conform to the ATC-20 post-earthquake safety evaluation field manual and its addendum, which provide procedures and guidelines for making on-site evaluations [1]. Responders such as ATC-20 trained inspectors, structural engineers and other specialists conduct visual inspections and designate affected buildings as green (apparently safe), yellow (limited entry), or red (unsafe) for immediate occupancy [2]. The assessment procedure can vary from minutes to days depending on the purpose of evaluation [3]. However it has been pointed out by researchers [4,5] that this approach is subjective and thus may sometimes suffer from misinterpretation, especially given that building inspectors do not have enough opportunities to conduct building safety assessments and verify their judgments, as earthquakes are infrequent.

Despite the de-facto national standard of the ATC-20 convention, researchers have been proposing quantitative measurement for more effective and reliable assessment of structural hazards. Most of these approaches, especially non-contact, build on the premise that significantly local structural damage manifests itself as translational displacement between consecutive floors, which is called interstory drift [6]. Interstory drift ratio, which is interstory drift divided by the height of the story, is a critical structural performance indicator that correlates the exterior deformation with the internal structural damage. The larger the ratio is, the higher the likelihood of damage. For example, a peak interstory drift ratio larger than 0.025 signals the possibility of serious threat to human safety, and values larger than 0.06 translate to severe damage [7].

This research proposes a new approach for estimating IDR using an Augmented Reality (AR)-assisted non-contact method. AR superimposes computer-generated graphics on top of a real scene, and provides contextual information for decision-making purposes. AR has been shown to have several potential applications in the civil infrastructure domain such as inspection, supervision, and strategizing [8]. AR-assisted building damage detection is a specific type of inspection.

2. Review of previous work

So far the most commonly accepted approach for obtaining IDR is via contact methods, specifically the double integration of acceleration.
This method is most commonly used because of its robustness and widespread availability in the world’s seismically active regions. However, Skolnik and Wallace [9] identified the vulnerability of double integration to nonlinear response. It has been suspected that sparse instrumentation or subjective choices of signal processing filters lead to these problems.

Another school of obtaining IDR is non-contact methods. Wahbeh and Caffrey [10] demonstrated a vision-based approach: tracking an LED reference system with a high fidelity camera. Ji [11] instead applied feature markers as reference points for vision reconstruction. Similar target tracking vision-based approaches have also been studied in [12] and [13]. However, all of them require the pre-installation of a target panel or emitting light source, and such infrastructure is not widely available and is subject to damage during long-term maintenance, since it is located on the exterior of the structure. Fukuda [14] tried to eliminate the use of target panels by using an object recognition algorithm, for instance orientation code matching. They performed comparison experiments by tracking a target panel and existing features on bridges, such as bolts, and achieved satisfactory agreement between the two test sets. However it is not clear whether this approach works in the scenario of monitoring a building’s structure, as building surfaces are usually featureless.

Researchers also utilized terrestrial laser scanning technology in non-contact methods for continuous or periodic structural monitoring [15,16]. In spite of the high accuracy of such systems, the equipment volume and the large collected dataset put these methods at a disadvantage for rapid evaluation scenarios.

Kamat and El-Tawil [5] first proposed the approach of projecting the previously stored building baseline on the real structure, and using a quantitative method to count the pixel offset between the augmented baseline and the building edge. In spite of the stability of this approach, which has been tested in UMA’s Structural Engineering Laboratory with large-scale shear walls, it required a carefully aligned perpendicular line of sight from the camera to the wall for pixel counting. Such orthogonal alignment becomes unrealistic for high-rise buildings, since it demands that both camera and the wall be at the same height.

Dai et al. [17] removed the premise of orthogonality using a photogrammetry-assisted quantification method, which established a projection relation between 2D photo images and the 3D object space. They validated this approach with experiments that were conducted with a two-story reconfigurable aluminum building frame whose edge could be shifted by displacing the connecting bolts. The experimental results were in favor of the adoption of consumer-grade digital cameras and photogrammetry-assisted concepts. However the issue of automatic edge detection and the feasibility of deploying such a method at large scales, for example with high-rise buildings, have not been addressed.

This paper specifically addresses the above limitations and proposes a new algorithm called line segment detector for automating edge extraction, as well as a new computational framework automating the damage detection procedure. To verify the approach’s effectiveness, a synthetic Virtual Prototyping (VP) environment has been designed to profile the detection algorithm’s sensitivity to errors inherent in the used tracking devices.

3. Overview of proposed reconnaissance methodology

Fig. 1 exhibits the schematic overview of measuring earthquake-induced damage being manifested as detectable building facade drift. The previously stored building information is retrieved and superimposed as a baseline wireframe image on the real building structure after damage. Then the sustained damage can be evaluated by comparing the key differences between the augmented baseline and the actual drifting building edge. Fig. 1 also demonstrates a hardware prototype called ARMOR (acronym for Augmented Reality Mobile Operating platform) [35] where the developed application can be potentially deployed. The inspector wears a GPS antenna and a RTK (acronym for Real Time Kinematic) radio that communicates with the RTK base station. Together they can track the inspector’s position up to centimeter-level accuracy. As discussed in Section 5.2, position and orientation tracking accuracy have great influence on the effectiveness of the estimation algorithm. Meanwhile, the estimation procedure and the final results can be shown in the HMD (acronym for Head Mounted Display) in front of the inspector.

The evaluation procedure is further illustrated in Fig. 2. The first step is for the camera to take pictures of the building. The orientation and location information about the camera needs to be recorded for 3D to 2D projection, as well as for 2D to 3D triangulation. The second step is to extract edges in the captured photo frames. A line segment detector extracts the vertical building edge, and an estimation method is used to represent the horizontal edge with the baseline. The last step involves the triangulation of the 3D coordinate at the key location from multiple corresponding 2D intersections between the vertical and horizontal edges. IDR is subsequently computed by comparing the key difference between two consecutive building floors divided by the story height. The accuracy of IDR calculation thus depends on the accuracy of internal and external camera parameters, the accurate detection of the vertical edge, and the estimation of the horizontal edge.

Besides being a quantitative means of providing reliable damage estimation results, the vertical baseline of the building structure is also a qualitative alternative for visual inspection of local damage. By observing the graphical discrepancy between the vertical baseline and the real building edge, the on-site reconnaissance team can approximately but quickly assess how severe the local damage is in the neighborhood of the visual field. In other words, the larger the graphical discrepancy is, the more severe the damage is. Fig. 3(a) and (b) focuses on different key locations of the building but are views taken from the same angle (i.e., direction). The right-bottom window on each image is a zoom-in view of the key location. The two vertical lines in the zoom-in window represent the detected edge and the vertical baseline respectively. The fact that the gap between the detected edge and the vertical baseline on Fig. 3(a) is smaller than that on Fig. 3(b), indicates that the key location on Fig. 3(b) suffers more local damage than that on Fig. 3(a).

4. Technical approach

The objective of this research was to design, demonstrate, and evaluate a new AR-assisted non-contact method for rapidly estimating the IDR in buildings that manifest residual drift from seismic damage. In particular, the research objectives included the verification of the developed algorithms, and the evaluation of the sensitivity of computed drift to measurement errors inherent in the used tracking devices. Access to a damaged high-rise building is rare. Moreover, such a test bed offers no possibility of inducing specific amounts of drift in the building stories for calibration or evaluation purposes.

In addition, an experimental plan conducted to understand the designed algorithm’s sensitivity to ambient conditions and instrument uncertainty requires a controlled test bed environment. In order to demonstrate and evaluate the developed computational framework, this research designed a synthetic 3D environment based on Virtual Prototyping (VP) principles for verifying the developed algorithms, and for conducting the sensitivity analysis. A Virtual Prototype, or digital mock-up, can be defined as a computer simulation of a physical counterpart that can be observed, analyzed, and tested from life-cycle perspectives, such as design and service, as if it were a real physical model. The creation and evaluation of such a Virtual Prototype is known as Virtual Prototyping (VP) [18]. By using a digital model instead of a physical prototype, VP can alleviate several shortcomings in the design and evaluation process.
Virtual Reality (VR) is a related concept and is specifically defined as a computer simulation of a real or imaginary system that enables a user to perform operations on the simulated system, and shows the effects in real time [19]. VR is thus of significant value to VP because it can facilitate the visual understanding of a virtual product during the design and evaluation process [20]. In effect, VR can support the analysis required for demonstrating and evaluating a proposed design by offering the possibility of immersing end-users in the virtual environment to perform specific tasks [21]. VP using VR principles thus emerged as a clear choice for demonstrating and evaluating the proposed computational framework.

4.1. Reconfigurable virtual prototype of seismically damaged building

The simulated VP environment contains a ten-story graphical and structural building model constructed as plans, as shown in Fig. 4. The graphical model is entirely reconfigurable and capable of manifesting any level of internal damage on its façade in the form of residual drift so that the IDR can be extrapolated for each floor. Given the input IDR, the structural macro model predicts the potential for structural collapse and the mode of collapse, should failure occur. The remainder of this paper focuses on the graphical model behavior and the underlying algorithm of extrapolating the IDR.

---

**Fig. 1.** Schematic overview of the proposed AR-assisted assessment methodology.

**Fig. 2.** Major steps to reconstruct the 3D coordinates of key locations on the building.

- **Monitor Camera Pose**
  - Electronic compass measures camera orientation
  - RTK GPS measures camera location

- **Edge Detection**
  - Detect vertical edges using Line Segment Detector
  - Project horizontal baseline to represent horizontal edge

- **3D Corner Coordinate Reconstruction**
  - Vertical edge and horizontal edge intersect on 2D image
  - Triangulate 3D coordinate from multiple 2D observations
Fig. 3. Graphical discrepancy between the vertical baseline and the detected building edge provides hints about the magnitude of the local damage.

Fig. 4. A ten-story graphical building model is constructed as its macro model counterpart.
The residual drift is represented by translating the joints of the wireframe model that have been superimposed with a high-resolution façade texture. The drift is further manifested through the displaced edges on the surface texture that can be extracted using a Line Segment Detector (LSD). Subsequently, the 2D intersections between extracted edges and projected horizontal baselines are used to triangulate the 3D spatial coordinates at key locations on the building.

The 2D image, where extracted edges and baselines are visible, is taken by the OpenGL camera that is set up at specified corners in the vicinity of the building (Fig. 5). At each corner, the camera's orientation (i.e., pitch) is adjusted to take a snapshot of each floor in sequence and project the corresponding horizontal baseline. In reality, the location of the camera may be tracked by a Real-Time Kinematic GPS (RTK-GPS) sensor, and its orientation may be monitored with an electronic compass. In the simulated VP environment, the location and orientation of the camera are known and can be controlled via its software interface. Random errors can thus be introduced to simulate the effects of systemic tracking uncertainty or jitter expected in a field implementation.

4.2. Damage modeling

The drift with uniform distribution is applied on each joint of a building to imitate the structural damage sustained after the disaster. The damage model is justified by the following statistics: the limit on inelastic IDR is commonly limited within 2.5% by building codes, and it is occasionally relaxed to 3% for tall buildings [22]. Given that the height of a building story is on average 3 m to 4 m, the maximum allowable displacement between two consecutive floors is 0.09 m to 0.12 m when using the most relaxed IDR of 3%. The drift of the corner's position is modeled as uniform distribution in both the X and Y directions. The distribution interval is limited within [−0.06 m, 0.06 m], so that the difference between consecutive floors in either the X or Y direction is less than 0.12 m. In the experiment, a reasonable assumption is made that unless the internal columns buckle or collapse, the height of the building remains the same after the damage. Since the column buckling or collapse situation is not modeled in the simulation, the Z value of the corner coordinate does not change.

A high-resolution façade texture was acquired for the modeled building from the Google Warehouse [23]. The texture is taken by a physical camera and rectified into orthogonal perspective so that it can be superimposed directly on the façade of the wireframe model. Each polygon vertex is assigned a 2D texture coordinate, and the associated clipped texture is pasted onto the surface of the wall (Fig. 6). The texture can thus displace with the drifting vertex in the 3D space, with the goal of estimating the vertex deformation through the displaced texture.

4.3. Camera modeling

In order to achieve parity with a practical field implementation, the OpenGL camera in the simulated environment is configured with the specifics of a real digital SLR (Single Lens Reflex) camera that may be used by an inspector to take pictures of a real building. This section describes how the external and internal parameters of the OpenGL camera were modeled to achieve such parity.

4.3.1. External parameters

The external parameters describe the position and orientation of the camera in the world coordinate system. In the OpenGL environment, the origin and unit of the world coordinate is arbitrarily specified by the programmer, and the pose of the camera can be known and error free. However in the real field application, the position and orientation of the camera may be tracked by GPS and 3D electronic compass, respectively, whose measurements are subject to instrument uncertainty.

The Laboratory for Interactive Visualization in Engineering (LIVE) at the University of Michigan, with which the authors are affiliated, is equipped with an RTK (Real Time Kinematics) GPS with a manufacturer-specified accuracy of 2.5 cm ±2 ppm RMS (Root Mean Square) horizontal, and 3.7 cm ±2 ppm vertical (Trimble 2009). The parts per million (ppm) error is dependent on the distance between the base and rover receiver. For example, if the distance is 10 km, a 2 ppm error equals 20 mm. The RTK-GPS yields a measurement reading in seconds once warmed up. Better accuracy can be achieved with higher-ranking RTK equipment and no significant compromise on collecting time. For example, manufacturers report 3 mm ±0.1 ppm RMS horizontal accuracy with Fast GNSS survey [24]. The 3-axis digital compass used in LIVE for outdoor angular measurements measures yaw, pitch, and roll with a resolution of 0.01° as the manufacturer-specified accuracy. The static accuracy for 3 axes is 0.3° (RMS) when the tilting (i.e., pitch and roll) is smaller than 65°. The accuracy is slightly compromised when the tilting range goes beyond 65° [25].

In order to verify the developed algorithms and computational framework in ideal conditions, the simulated experiments are first performed with ground truth position and orientation readings, i.e. assuming perfect tracking of the camera’s position and orientation. Subsequently, to investigate the practicality of the method in the field implementations, the same experiments are conducted with the introduction of the instrument uncertainty in a certain range.

4.3.2. Internal parameters

In the OpenGL camera, the internal parameters can be represented by left, right, top, bottom, near, and far plane values, which form a viewing frustum (Fig. 7). The physical counterpart of the near plane inside the digital camera is the sensor chip. Given the sensor chip parameters of the mainstream off-the-shelf camera and without a loss of generality, the left and right values are set to ±11.15 mm; the bottom and top values are set to ±7.45 mm [26]. The focal length of the camera is approximately equivalent to the near plane distance, and can be adjusted flexibly from 20 mm to 200 mm. The lens with a similar range is off-the-shelf available and economically affordable. In the simulated experiments, usually the maximum focal length is selected for best performance.

Theoretically, the internal parameter errors are induced by the imperfection of the camera lens and its internal mechanics with a consistent effect. There are two major elements of comprising the induced camera’s systematical error, i.e. the lens distortion, an aggregate of the radial distortion and the decentering distortion, and the
approximation of the focal length distance. Unlike the external errors that can be affected by the environmental variables dynamically, like visibility of the sky and dynamic magnetic field, the internal errors are relatively stable and can be systematically compensated by camera calibration beforehand [27].

4.4. Vertical edge detection

Vertical Edge detection of the building wall is the most critical step for locating the key point on the 2D image plane, which happens to be a fundamental problem in image processing and computer vision domains, as well. Many algorithms for edge detection exist and most of them use Canny Edge Detector [28] and Hough Transformation [29] as a benchmark. However, standard algorithms are subject to two main limitations. First, they face threshold dependency. Edge detection algorithms contain a number of adjustable parameters that influence their effectiveness. The tuning of parameters can yield significant overhead for on-site reconnaissance inspectors and compromise assessment efficiency, as well as the detection accuracy. Second, standard algorithms face false positives and negatives. They either detect too many irrelevant small line segments or fail to interpret the desirable line segments. False positives and negatives are highly related to the threshold tuning.

4.4.1. Active contour approach

The authors’ first attempt was to apply the Graph-Cut Based Active Contour (GCBAC) [30]. Traditional active contour is an energy-minimizing spline guided by external forces and influenced by image forces [31]. By introducing the concept of contour neighborhood, GCBAC alleviates the local minima trapping problem suffered by traditional active contour: the energy-minimizing spline could be trapped by objects in their neighborhood with higher gradient zones, which means instead of detecting edge with global minimized energy, the one with local minimized energy turns out to be the converged result (detected edge).

GCBAC requires manual specification of the initial contour and contour neighborhood width, quantities that are arbitrary and subjective. However optimization can be achieved by using the original baseline of the damaged building to numerically calculate both initial contour and neighborhood width. GCBAC works best when the image covers the entire outline of the building that is not applicable in the real applications (Fig. 8). Unfortunately, the coverage of the entire high-rise building surface inevitably results in lower-resolution details. Moreover, frequent partial occlusion from trees and other buildings can compromise detection accuracy.

4.4.2. Line segment detection approach

The second attempt was a linear-time line segment detector that gives accurate results, a controlled number of false detections, and—more importantly—requires no parameter tuning [32]. LSD cumulates the advantages of Burns’ [33] and Moisan’s [34] methods and gracefully hides their drawbacks. The Burns’ algorithm innovatively ignores gradient magnitudes and uses only gradient, which yields a well-localized result. It is linear-time but subject to the threshold problem. The threshold question was thoroughly studied in Desolneux’s algorithm. It is based on a general perception principle that an observed geometric structure is perceptually meaningful when its expectation in noise is less than one. The principle guarantees the lack of false positives and no false negative. Unfortunately, the method is exhaustive and has an $O(N^4)$ complexity. The innovative combination of these two approaches is a linear-time LSD that requires no parameter tuning and gives accurate results.

LSD outperforms GCBAC in searching for localized line segments. However, there are still multiple line segment candidates in the neighborhood of the actual edge of the building wall (Fig. 9). A filter

Fig. 6. Internal structural damage, shift of the vertex, is expressed through the displacement of the texture.

Fig. 7. The near plane of the OpenGL camera is the counterpart of the physical camera image sensor chip, a device that converts an optical image into an electronic signal.
is used to eliminate those line segments whose slope and boundary deviate significantly from the original baseline. In initial attempts, the authors proposed fully automating the edge detection procedure by choosing the one with the closest distance to the original baseline. It will be shown that this approach is problematic and was thus found to be unfeasible. Manual selection, on the other hand, was identified to be much more accurate and can be completed in a few seconds. If the LSD fails to locate the desirable edge, the user can manually transpose the closest line segment to the desirable position with little time overhead.

4.5. Corner detection

4.5.1. Horizontal Edge Detection

Besides the vertical edge detection, the horizontal edge detection also plays an essential role in deciding the 2D coordinate of the drifting corner. If the horizontal frames of windows approximately match with the physical floors separating stories, then the horizontal edge can also be graphically detected by LSD as windows’ bottom frame (Fig. 9). However, since such assumption is not universally true, we choose to numerically project the horizontal baseline that physically separates stories on the damaged building surface to represent the horizontal edge. Such an approach is more generic than the graphical detection. However since a floor is allowed to drift within the XY plane, its horizontal baseline has to be shifted accordingly before it is projected onto the 2D image so as to match the real horizontal edge. If the 2D projected horizontal edge does not strictly match with the real horizontal edge, then a gap is detectable, and it enlarges as the camera moves closer to the building. Furthermore, the drift between the horizontal baseline and the horizontal edge contains both parallel and perpendicular component (in the x and y direction), and the detectable gap is caused exclusively by the perpendicular component. The drift on the z coordinate is not considered here since internal column buckling or collapse is not considered in the damage model.

Unless the drift is known, it is impractical to deterministically position the edge in the XY plane. Therefore, the proposed solution is to exhaustively test all possible drifting configurations, with computation complexity of $\Theta(N^4)$. This happens because iterating through all the possible shift configurations of two endpoints on one line segment costs $\Theta(N^2)$, given that only the perpendicular drift component between the edge and the baseline is considered. The union of two line segments needed in the triangulation has $\Theta(N^4)$ complexity, (Fig. 11(a)) where N is equal to the uniform distribution interval divided by the estimation step. For example, if the uniform distribution interval is $[-0.6, 0.6]$, and the joint position is shifted from $-0.6$ to $0.6$ by $0.1$ at one step, then N is equal to 12.

Furthermore, a simple approximation can reduce the complexity from $\Theta(N^4)$ to $\Theta(N^2)$ without compromising the accuracy. Since the intersection between the baseline and the edge is close to one
endpoint of the line segment, only the \((x, y)\) of that endpoint dominates the intersection accuracy, and the impact of the other end diminishes significantly given the ratio of the drift magnitude over the distance between two endpoints. Therefore the two points on one line segment can share the same tested shifting value with \(\Theta(N)\) complexity, and subsequently the complexity of two line segments decreases to \(\Theta(N^2)\) (Fig. 11(b)).

There are two edges on the adjacent walls intersecting with the edge, and the one with lower slope (absolute value) should be chosen for calculating the 2D intersection. This is because how close the projected baseline is to the actual edge on the 2D image is not only affected by its 3D coordinate, but also the perspective projection. For example, imagine the camera is initially placed at an infinite point, regardless of the displacement between the horizontal baseline and the building edge, their projections overlap on the 2D image. Then the camera is moved toward the building and eventually placed beneath the baseline. In this case, if the camera looks straight up, the gap between their projections on the 2D image is equal to the displacement in the 3D space. Fig. 10 backs up this observation. In general, the less the slope of the edge, the less the gap between the baseline and edge projected on the 2D image. Additionally, if only one side of the building is covered in the image, the edge on the visible side is chosen (Fig. 14(c, d)).

4.5.2. Corner detection algorithm

The next challenge is to select the best estimation from the \(N^2\) candidates in the aforementioned iteration test. Each pair of tested shift \((\Delta x, \Delta y)\) of the baselines corresponds to an estimated 3D corner position \((x', y', z')\). If the actual 3D corner position is \((x, y, z)\), and if the height of the building remains the same after the damage, an intuitive judgment for the confidence of the estimation is \(\min(z-z')\).

A better judgment also takes \((x', y')\) into account. Say the original 3D corner position is \((x_0, y_0, z_0)\), a proper tested shift \((\Delta x, \Delta y)\) should be close to the estimated shift \((x' - x_0, y' - y_0)\). In other words, \((x' - x_0 - \Delta x, y' - y_0 - \Delta y)\) should be minimized.

Based on the hypothesis above, there are two filters proposed for selecting the estimated corner coordinate. The first one minimizes the square root of \((x' - x_0 - \Delta x, y' - y_0 - \Delta y, z' - z_0)\). The second one sets thresholds for \((x' - x_0 - \Delta x, y' - y_0 - \Delta y, z' - z_0)\), and selects the one with the smallest \((|\Delta x|, |\Delta y|)\) among the filtering results. Based on the experiment results, there is no major performance gain of one over the other. The algorithm is described as a flow chart in Fig. 12.

4.5.3. Interstory Drift Ratio calculation

Once the corner’s position is estimated, the calculation of Interstory Drift Ratio for each story is straightforward given its definition, which is
the interstory drift divided by the height of that story. For example, the right image in Fig. 13 shows a zoom-in view of a certain story. The IDR of that story by the side facing the reader can be calculated as \((P_1 - P_2)/h\).

5. Evaluation of experimental results

To understand the best performance that the computational framework can achieve in the ideal situation, this section starts with studying algorithm performance in a series of controlled comparison experiments with ground true camera tracking data. Later on, the experiment is extended to situations where instrumental errors are included to profile algorithm sensitivity.

5.1. Experiment with ground true location and orientation

The goal of this subsection is to test the best performance that the algorithm can achieve with ground true camera pose tracking data. Even given the ground true tracking data, the estimation accuracy can still be affected by many factors. Therefore a series of comparison experiments is conducted to find the influence magnitude of each factor. For each group of comparison, the statistics shows the average, standard deviation, and the maximum of the square root of x, y coordinate error. The minimum—generally smaller than 1 mm—is not included because it is not essential in judging the accuracy. There are 10 stories and four building edges, and thus 40 corner coordinate samples are included in each experiment group.

5.1.1. Observing distance

The purpose of this set of experiments is to understand the impact of observing distance on the estimation accuracy. The observing distance is the projection of the vector between the camera and building corner on the XY plane. Three experiments are conducted with the same internal camera parameters (6 Mega Pixels), damage model (±0.04 m shifting range), and 0.01 m estimation step, except that the camera is moving farther away from the building.

As evidenced in Table 1, the accuracy improves when the camera moves away from the building. As mentioned in Section 4.5.1, in general, the lower the slope, the less the projected error, because increasing distance helps to approximate orthogonal perspective. Therefore, increasing the distance of the camera from the building has the effect of lowering the slope and attenuating the error. Approximate orthogonal perspective can also be achieved if the camera is at the same height of the baseline. This is supported by the fact that the estimation error is always tiny for the first floor, where the height of the floor is similar to that of the camera and the slope is close to zero.

5.1.2. Observing angle

This experiment tries to understand whether the observing angle could affect the accuracy. The observing angle is formed by the line of sight between two cameras. In the first group, two images from two perspectives cover both sides of the building wall (Fig. 14(a, b)). In this case, the observing angle is closer to a right angle. In the second group, one image covers both sides, while the other one covers only one side; in the third group, both images cover only one side of the building wall (Fig. 14(c, d)). In the latter two cases, the observing angle is closer to 180°. All of the other environmental parameters are controlled as follows: 6 Mega Pixels, ±0.04 shifting range, 35 m observing distance, and 0.01 m estimation step.

It has been shown in Table 2 that the accuracy degenerates significantly when covering only one side of the wall. This indicates that the detection error is minimized when the angle formed by two lines is close to a right angle, and magnified when the angle is either acute or obtuse.

5.1.3. Drift interval

Here we want to understand whether the estimation accuracy could be affected by the uniform distribution interval assumed. The three tested intervals are \(\pm 0.04\) m, \(\pm 0.05\) m, and \(\pm 0.06\) m. The camera distance is fixed at 35 m, the image covers both sides of the building wall with 6 Mega Pixels, and the estimation step is 0.01 m.
As evidenced in Table 3, the increase of the drifting range slightly deteriorates the accuracy, especially the standard deviation. The increasing difficulty of estimating the larger gap between the baseline and floor outline is probably responsible for the degeneration of the accuracy. However, as will be shown in Section 5.1.6, the increase of image resolution can mitigate accuracy loss.

5.1.4. Approximate versus accurate
The authors claimed in Section 4.5.1 that the approximate mode can achieve the same accuracy level as the detailed iterative mode, but with less computational expense. This is supported by the following experiment group. The chosen test bed is the same one as in Section 5.1.3. The first group uses the approximated method of $\Theta (N^2)$ complexity, and the second one uses the accurate method of $\Theta (N^4)$ complexity.

Table 4 shows that even though the average error of $\Theta (N^2)$ is slightly smaller, if that, than the average error of $\Theta (N^4)$, the standard deviation and maximum error have more or less the same accuracy level. Therefore, $\Theta (N^2)$ is a good approximation of $\Theta (N^4)$ without accuracy loss but a significant gain in computational time.

5.1.5. Estimation step
As mentioned in Section 4.5.1, the computation cost is decided by the uniform interval and the discrete estimation step. This experiment group tries to decide the optimal estimation step. The controlled experiment condition with $\pm 0.04$ m interval in Section 5.1.3 is chosen as a benchmark here. The compared estimation steps are $0.01$ m, $0.005$ m, and $0.0025$ m.

As evidenced by Table 5, smaller intervals can hardly increase the accuracy, therefore the $0.01$ m interval is recommended for real applications.

5.1.6. Image resolution
Image resolution is one of the most important factors of characterizing a camera. Unfortunately, the resolution of the OpenGL camera is limited by that of the monitor (1900 × 1200) of the Dell Precision M60 laptop that was used in this research. An alternative is to use a telephoto lens. This can be achieved in OpenGL by pushing the near plane farther away without changing its size. For example, pushing the near plane two times away is equivalent to magnifying the resolution by a factor of four.

Table 6 indicates that a higher image resolution apparently helps promote the accuracy of line segment detection, which in turn increases the overall accuracy. Given the statistics from Sections 5.1.4 and 5.1.5, where no improvement is observed, it can be concluded that the accuracy of line segment detection is the bottleneck of the algorithm given ground true tracking data.
the measurement accuracy of 5 mm as it indicates by the left to right arrow. Unfortunately, to the authors’ best knowledge, a state of the art electronic compass cannot satisfy the precision requirement. Most off-the-shelf electronic compasses report uncertainty bigger than 0.1° (RMS), thus suggesting the need for survey-grade line-of-sight tracking methods for monitoring the camera’s orientation.

The third experiment considers comprehensive errors from both location and orientation readings (Fig. 17). It again proves that uncertainty from an electronic compass becomes the critical source of error in the methodology.

### 5.1.7. Automatic versus manual

As mentioned in Section 4.4.2, the automatic detection filters the line segments by their distance to the original vertical baseline. The closest one is preserved. This experiment demonstrates that such a heuristic is problematic. Similar experiment conditions to the one in Section 5.1.3 are chosen as a test bench.

As shown in Table 7, current automatic selection of the line segment detection is not robust enough to achieve the same level of accuracy as the manual selection. However, these observations do not preclude the existence of other possible heuristics.

### 5.2. Experiments with instrument error

The previous section analyzed the algorithm performance with ground true sensor readings. In this subsection, we design three groups of comparison experiments to test the robustness of this method in the presence of instrument errors. The experiments are conducted with the best configuration, as found in the previous section; i.e. the camera of 18 mega pixels is located about 35 m away from the building with its photos covering both sides of the building.

This first experiment assumes ground truth orientation data, and only introduces error to location. In Fig. 15, the Z axis shows the average estimation error with the unit of meter. The altitude RMS axis shows the accuracy response to the change in electronic compass pitch and roll readings uncertainty, and the yaw RMS axis shows the accuracy response to the change in electronic compass yaw reading uncertainty. The result shows that uncertainty on pitch and roll has a more adverse impact on the displacement error than that of the yaw as it indicates by the right to left arrow. Furthermore, a precision of 0.01° (RMS) on all three axes is required to keep the displacement error in the useful range as it indicates by the left to right arrow. Unfortunately, to the authors’ best knowledge, a state of the art electronic compass cannot satisfy the precision requirement. Most off-the-shelf electronic compasses report uncertainty bigger than 0.1° (RMS), thus suggesting the need for survey-grade line-of-sight tracking methods for monitoring the camera’s orientation.

### 6. Conclusion

This paper described a simulated Virtual Prototyping test bed to evaluate the feasibility of deploying an Augmented Reality assisted non-contact building damage reconnaissance method in the field. The research demonstrated the effectiveness of VR-assisted Virtual Prototyping in evaluating and demonstrating new reconnaissance methods where full-scale physical test bed experimentation is impractical. The experimental plan constructed a ten-story graphical building capable of expressing the internal structural damage through the texture displacement on the surface. LSD can detect the shifted building edge on the captured building image, and the final corner coordinate is triangulated through the intersections between

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Sensitivity of drift error to observation distance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average distance</td>
<td>10 m</td>
</tr>
<tr>
<td>Ave Error</td>
<td>0.0079 m</td>
</tr>
<tr>
<td>Stddev</td>
<td>0.0047 m</td>
</tr>
<tr>
<td>Max Error</td>
<td>0.0165 m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Sensitivity of drift error to observing angle.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both cover two sides</td>
<td>One covers two sides</td>
</tr>
<tr>
<td>Ave Error</td>
<td>0.0048 m</td>
</tr>
<tr>
<td>Stddev</td>
<td>0.0029 m</td>
</tr>
<tr>
<td>Max Error</td>
<td>0.0126 m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Sensitivity of drift error to drift interval.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drifting range</td>
<td>[−0.04 m, 0.04 m]</td>
</tr>
<tr>
<td>Ave Error</td>
<td>0.0048 m</td>
</tr>
<tr>
<td>Stddev</td>
<td>0.0029 m</td>
</tr>
<tr>
<td>Max Error</td>
<td>0.0126 m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Sensitivity of drift error to accurate and approximate estimation mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Θ(N2)/Θ(N4)</td>
<td>[−0.04 m, 0.04 m]</td>
</tr>
<tr>
<td>Ave Error</td>
<td>0.0048 m</td>
</tr>
<tr>
<td>Stddev</td>
<td>0.0029 m</td>
</tr>
<tr>
<td>Max Error</td>
<td>0.0126 m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5</th>
<th>Sensitivity of drift error to estimation step.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimation step</td>
<td>0.01 m</td>
</tr>
<tr>
<td>Ave Error</td>
<td>0.0048 m</td>
</tr>
<tr>
<td>Stddev</td>
<td>0.0029 m</td>
</tr>
<tr>
<td>Max Error</td>
<td>0.0126 m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6</th>
<th>Sensitivity of drift error to image resolution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>6 mega pixels</td>
</tr>
<tr>
<td>Ave Error</td>
<td>0.0048 m</td>
</tr>
<tr>
<td>Stddev</td>
<td>0.0029 m</td>
</tr>
<tr>
<td>Max Error</td>
<td>0.0126 m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7</th>
<th>Sensitivity of drift error to manual and automatic detection mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto manual</td>
<td>[−0.04 m, 0.04 m]</td>
</tr>
<tr>
<td>Ave Error</td>
<td>0.0109 m</td>
</tr>
<tr>
<td>Stddev</td>
<td>0.0119 m</td>
</tr>
<tr>
<td>Max Error</td>
<td>0.0583 m</td>
</tr>
</tbody>
</table>
the detected vertical building edge and the projected horizontal baseline.

The experiment results with ground true location and orientation data are satisfactory for damage detection requirements. The results also highlight the conditions for achieving the ideal measurement accuracy, for example observing distance, angle, and image resolution. The experimental results with instrumental errors reveal the bottleneck for the proposed method in the field implementation conditions. While the state of the art RTK-GPS can meet the location accuracy requirement, the electronic compass is not accurate enough to supply qualified measurement data, suggesting that alternative survey-grade orientation measurement methods must be identified to replace electronic compasses. The conducted sensitivity analysis developed a clear matrix revealing the relation between instrument accuracy and accuracy of computed drift, so the proposed method’s practical implementation can evolve with choices made for higher accuracy instruments than the ones tested. The authors acknowledge that the sensitivity matrix developed from the virtual prototyping may have limitations and needs to be further validated in a real environment setting. For example, the dynamic illumination may bring challenges to the edge detection. Furthermore the estimation method assumes ground true geometric building information is available. It is possible that in reality, such information contains uncertainty or is possibly unavailable for older buildings. The current virtual prototyping has not modeled such data uncertainty. The open source code for the virtual prototyping and its sensitivity analysis is available at <http://pathfinder.engin.umich.edu/software.htm>.

Acknowledgments

The presented work has been supported by the United States National Science Foundation (NSF) through Grant CMMI-0726493. The authors gratefully acknowledge NSF’s support. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

References