

Article

Physics-based Graphics Models in 3D Synthetic Environments Enabling Autonomous Vision-Based Structural Inspections

Vedhus Hoskere^{1*}, Yasutaka Narazaki² and Billie F. Spencer Jr.³

^{1*} University of Houston; vhoskere@uh.edu

² ZJU-UIUC Institute, Zhejiang University; narazaki@intl.zju.edu.cn

³ University of Illinois at Urbana-Champaign, bfs@illinois.edu

* Correspondence: vhoskere@uh.edu

Abstract: Manual visual inspections typically conducted after an earthquake are high-risk, subjective, and time-consuming. Delays from inspections often exacerbate the social and economic impact of the disaster on affected communities. Rapid and autonomous inspection using images acquired from unmanned aerial vehicles offer the potential to reduce such delays. Indeed, a vast amount of research has been conducted toward developing automated vision-based methods to assess the health of infrastructure at the component and structure level. Most proposed methods typically rely on images of the damaged structure, but seldom consider how the images were acquired. To achieve autonomous inspections, methods must be evaluated in a comprehensive end-to-end manner, incorporating both data acquisition and data processing. In this paper, we leverage recent advances in computer generated imagery (CGI) to construct a 3D synthetic environment for simulation of post-earthquake inspections that allows for comprehensive evaluation and validation of autonomous inspection strategies. A critical issue is how to simulate and subsequently render the damage in the structure after an earthquake. To this end, a high-fidelity nonlinear finite element model is incorporated in the synthetic environment to provide a representation of earthquake-induced damage; this finite element model, combined with photo-realistic rendering of the damage, is termed herein a physics-based graphics models (PBGM). The 3D synthetic environment with PBGMs provide a comprehensive end-to-end approach for development and validation of autonomous post-earthquake strategies using UAVs, including: (i) simulation of path planning of virtual UAVs and image capture under different environmental conditions; (ii) automatic labeling of captured images, potentially providing an infinite amount of data for training deep neural networks; (iii) availability of the ground truth damage state from the results of the finite-element simulation; and (iv) direct comparison of different approaches to autonomous assessments. Moreover, the synthetic data generated has the potential to be used to augment field datasets. To demonstrate the efficacy of PBGMs, models of reinforced concrete moment-frame buildings with masonry infill walls are examined. The 3D synthetic environment employing PBGMs is shown to provide an effective testbed for development and validation of autonomous vision-based post-earthquake inspections that can serve as an important building block for advancing autonomous data to decision frameworks.

Keywords: Synthetic Environments, Deep learning, Computer Graphics, Post Earthquake Inspections, Physics-based Graphics Models, Damage Detection

1. Introduction

The inspections of structures that are necessary after earthquakes are laborious, high-risk, and subject to human error. Describing the nature of inspections in a post-disaster scenario, the ATC-20 field manual [1] states that post-earthquake safety evaluations of buildings is “grueling work,” resulting in a high level of stress on the volunteer inspectors that may lead to “burn-out.” Entry into damaged structures for inspections poses a serious

safety risk due to the unknown nature of their structural integrity. Additionally, the time-consuming nature of these inspections can exacerbate the social and economic impacts of the disaster on affected communities. For example, immediately after the 2017 Central Mexico Earthquake, hundreds of thousands of citizens had to exit damaged buildings (Figure 1a) and were left with uncertainty about the state of their homes and offices [2]. The initial evaluations conducted by the Civil Engineering Association of took three weeks from the occurrence of the earthquake [3]; at that time, 1210 buildings showed signs of damage but were safe to occupy, 327 buildings were severely damaged and unsafe to occupy, and 427 buildings needed detailed evaluation [4]. Nearly 17,000 people were being housed in camps for weeks [5] while waiting for detailed inspections [6], and hundreds could still be seen still camping out in tents after 6 months (Figure 1b). Similar scenarios may be observed anytime a large earthquake hits a densely populated region [7–9]. To help communities recover swiftly after a disaster and get people back in their homes and businesses as soon as possible, more efficient inspections are required.



Figure 1. (Left) (a) Building damaged during the 2017 Puebla Earthquake. (right) (b) Temporary tents by citizens evacuated from buildings under inspection

Data-acquisition with unmanned aerial vehicles and data processing using deep learning algorithms have shown tremendous potential in advancing the level of autonomy in post-earthquake inspections. A frequently studied problem is the application of machine learning algorithms, with specific focus on the use of deep convolutional neural networks (CNNs) for damage identification after earthquakes. Yeum et al. [10] proposed the use region-based R-CNN for spalling detection in earthquake damaged buildings. Mondal et al. [11], implemented the Faster R-CNN [12] algorithm to compare different network architectures for multi-class detection of damage in earthquake affected buildings. Xu et al. [13] utilized Faster R-CNN for damage identification of cracks, spalling, and exposed rebar in concrete columns. Researchers have also sought to incorporate context of the damage and information from the entire structure to contribute to a structural assessment using deep learning methods. For example, Hoskere, et al. [14,15] proposed the use of deep-learning based semantic segmentation for multiple types of damage and materials. The proposed methodology was extended to the semantic segmentation of scenes, components, and damage in reinforced concrete buildings in [16]. Narazaki et al. [17,18] proposed the use of fully convolutional neural networks to identify bridge components for post-earthquake inspections. Narazaki et al. [19] employed recurrent neural networks with video data to help better understand the structural component context of close up videos during bridge inspections. Gao et al. [20] developed the PEER-Hub dataset incorporating multiple classification challenges for the post-earthquake assessment of buildings. Liang et al. [21] proposed a three-level image-based approach for post-disaster inspection reinforced concrete bridges using deep learning. Dizaji et al. [22] conducted

preliminary research on using 3D data to train a network for defect identification of cracks and spalling on concrete columns. Pan et al. [23] presented a framework to combine performance assessment with repair cost evaluation using deep learning, extending the types of information that can be extracted from image data to aid decision makers. A detailed review of advances in vision based inspection and monitoring can be found in Spencer et al. [24].

While the efficacy of deep learning methods has been demonstrated for autonomous inspection subtasks such as damage identification and component identification, end-to-end validation is rarely conducted, due in part to the rare occurrence of earthquakes and the difficulty in obtaining ground-truth data. For example, the images employed in the studies discussed above are often well-lit close-ups of the damage; how these images could be acquired in the field is not considered.

The significant progress in computer graphics software over the past decade has allowed the creation of photo-realistic images and videos from 3D synthetic environments. The data generated from these graphics models, termed synthetic data, has been used to validate applications like robotic simulation (e.g., Gazebo [25]) and for reinforcement learning in autonomous vehicles (e.g., AirSim [26]). Such synthetic data has also been used for semantic segmentation of urban scenes that have showed promising performance on field-collected data (Ros et al. [27]). Moreover, improving diversity and photorealism of the simulated worlds has helped improve the performance of methods trained on synthetic data and subsequently applied on field data, as demonstrated by recent results in self-driving [28–30]. 3D synthetic environments can thus offer strong potential for end-to-end validation of strategies for automated post-earthquake damage assessment.

While generating damaged structures in 3D synthetic environments, a key challenge is determining how to realistically display the damage on the structure. Hoskerc et al. [31–33] proposed ideas on physics- and heuristics-based damage models to inform the accurate rendering of the damage on structures. Physics-based graphics models (PBGMs) are graphics models in a synthetic environment informed by underlying finite element model(s), for determining damage representation and/or mechanical behavior of structural and non-structural elements. Heuristics-based models utilize rules based on empirical observations for determining damage representations. Narazki et al. [34] utilized heuristics-based models in a 3D synthetic environment to obtain a dataset of damaged RC viaduct images and train a deep neural network for damage detection. Narazaki et al. [35,36] also developed physics-based graphics models for displacement and strain measurement of miter gates and laboratory bridge structures. Zdziebko et al. [37] developed a Physics-based graphics model of a laboratory beam structure for the development of vision-based displacement measurement algorithms. The generated damage textures will be more physically accurate than heuristically generated damage, which is expected to reduce dataset bias in the long run. Research on using PBGMs to generate realistic and physically-based visual damage patterns of structures has not been explored. Incorporating models for visual representation of damage together with finite element models could unlock new possibilities for end-to-end inspection simulation, leading to more intelligent, efficient, rapid and autonomous structural assessments.

This paper develops a comprehensive and automated framework for generating Physics-based Graphics Models (PBGMs) as part of a 3D synthetic environment with the ability to simulate earthquakes and render physically accurate damage. The framework provides novel strategies for the incorporation of non-linear finite element analysis with constitutive models, and graphics models for visual representation of the damage. The proposed framework is applied to generate renderings from twelve reinforced concrete buildings subject to earthquake excitation. The utility of the proposed framework is demonstrated through three applications and experiments, namely, the creation of a large-scale synthetic dataset of earthquake damaged structures with annotations for semantic segmentation, augmenting real data with synthetic data, and comparing the damage state estimation using UAV and ground-based images. The manuscript is organized into the following sections, (i) 3D synthetic environments for inspections, (ii)

implementation of the proposed framework RC buildings, (iii) applications and experiments, (iv) results, and (v) conclusions followed by references.

2. 3D synthetic environments for visual inspections

3D synthetic environments (Figure 3) are defined as modeling software with the ability to simulate object geometries and textures, lighting sources, and cameras. Using synthetic environments, image capture from UAV during an inspection is simulated by rendering images from camera locations following planned flight trajectories. Different flight paths and data acquisition schemes can be evaluated in the synthetic environment for identification of flight parameters like distance from the structure for optimal identification accuracy of both damage and components, flight path for complete coverage, etc. Before such tests can be carried out, a key challenge is to model the structure and environment of interest. In this study, PBGMs are proposed as an effective tool for modeling the structures of interest in 3D synthetic environments. Generation of synthetic data using PBGMs allows for the creation of useful annotated datasets of damaged structures, as any data from algorithmically generated graphics models will be automatically labeled, both at the pixel and image level using damage locations and states implicit in the PBGM. Different conditions, such as ground excitation, lighting, paint colors, dirt, etc. can be simulated, to generate a wider variety of training data robustly representing different realistic environments (Figure 2). The generated data can be used to train a deep network for semantic segmentation, facilitating automation of multiple tasks. As the damage is informed by a finite-element model, the generated data can be used to conduct overall assessments using the ground truth of the structure condition is available. Finally, as the visual representations are linked to the results of the finite element model, they provide one means of developing finite element model updating strategies. Figure 4 lists applications of PBGMs in synthetic environments for various visual inspection tasks. PBGMs and synthetic environments will provide a testbed for vision-algorithms with readily repeatable conditions. Algorithms that are effective in these virtual testbeds will be more likely to work well on field-collected datasets. The developed datasets using can also be used to augment field datasets to enhance accuracy. This section first described a new framework for the generation of physics-based graphics models followed by scene assembly and image rendering in 3D synthetic environments.



Figure 2. (left)(a) Real image of a building damaged after the Puebla Earthquake captured from a UAV (right) (b) synthetic PBGM image of a building with similar layout as (a) subject to the Tabas earthquake with different lighting generated using the proposed approach.

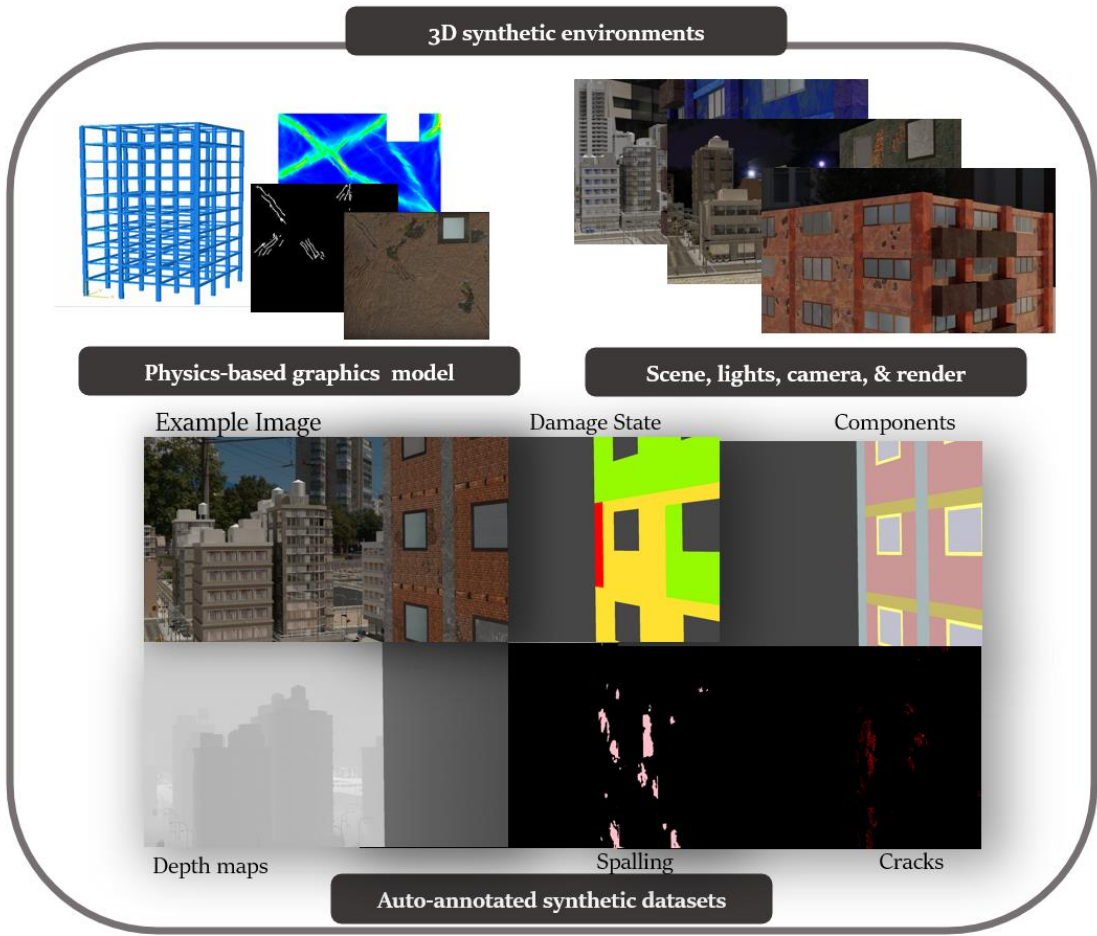


Figure 3. 3D synthetic environments

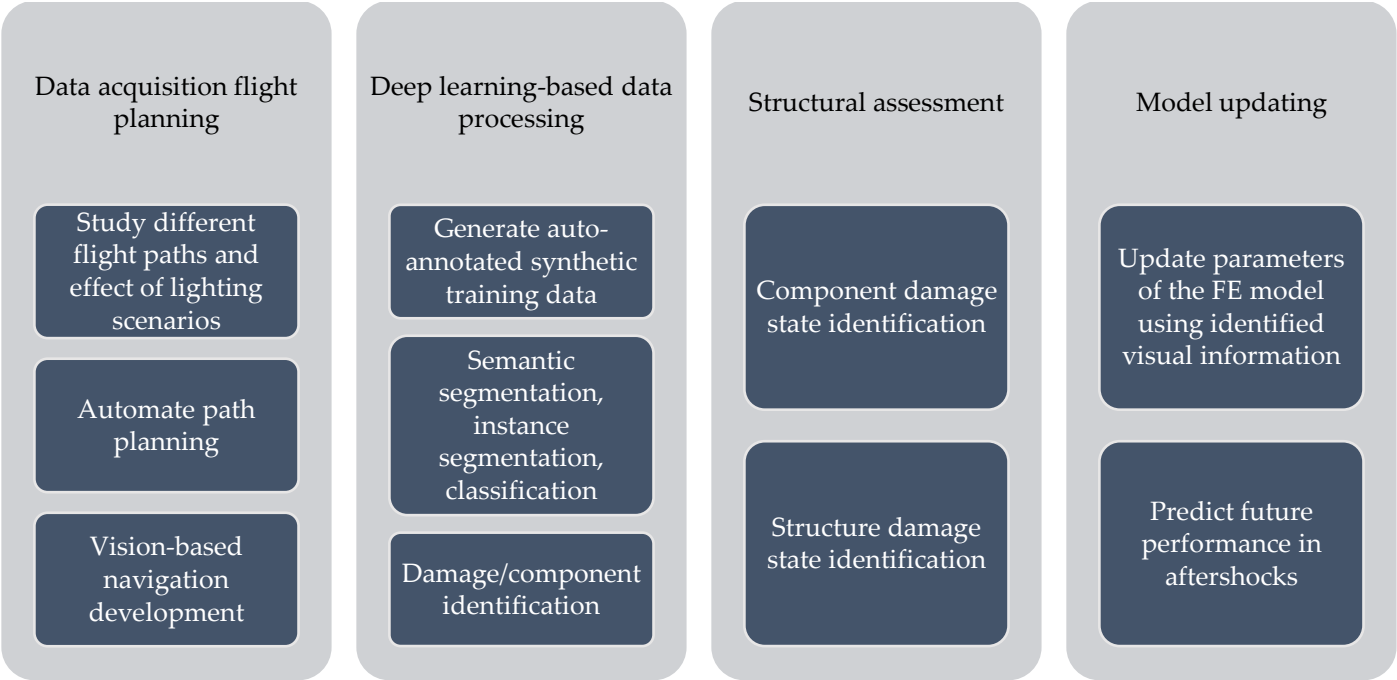


Figure 4. Applications of PBGMs in synthetic environments for autonomous vision-based structural inspections

2.1 Physics-based graphics models

In this section a framework for generation of physics-based graphics models (PBGMs) for inspections is presented. For clarity, the framework is illustrated in a schematic presented in Figure 5 with reinforced concrete buildings with masonry infill walls as the structure type. However, the same procedures may be followed for other structures where the physics can be simulated through finite element models. The framework consists of five steps including, (i) graphics mesh creation, (ii) non-linear finite element analysis, (iii) damage masks generation (iv) damage texture generation, and (v) texture assignment

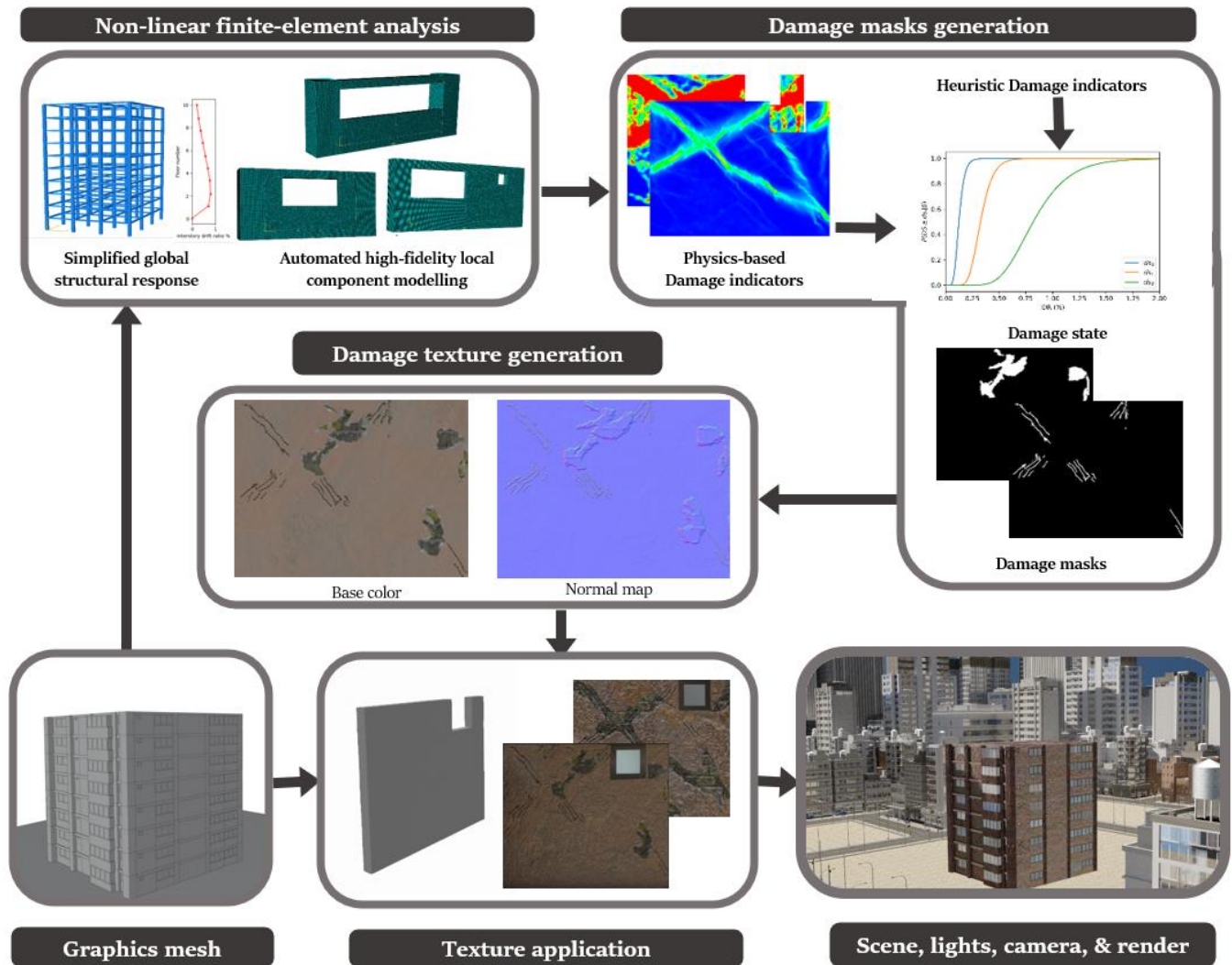


Figure 5. Framework for generation of PBGMs in 3D synthetic environments

2.1.1 Graphics mesh creation

The geometry of the structure of interest in the PBGM is represented by a 3D mesh. The mesh may be created in any 3D creator software. The features of buildings incorporated in the 3D mesh will enable networks trained on synthetic data to learn representations for those features in real images. For structural inspections of buildings, structural components like beams, columns, and shear walls, and non-structural components like infill walls, doors, windows, and balconies, are highly relevant as damage to these components provide visual indicators of structural health. Similar lists can be made for other types of structures to be inspected. All these components should be created programmatically through parameterization, or, as referred to in the field of computer graphics, created "procedurally". Procedural generation of the mesh will allow programmatic

implementation of subsequent steps, thus enabling randomization of both geometry and textures. Randomization has shown to improve the performance for related tasks like robotic perception when learning from synthetic data by Tobin et al. [38] and regarded as an effective way to learn generalizable representations [29].

2.1.2 Non-linear finite element analysis

From the perspective of PBGM generation, non-linear finite element analyses provide valuable insight into the regions in a structure where damage is most likely to occur. The same parameters used to construct the mesh procedurally are used to generate finite element models as well. In the particular case of post-earthquake inspections, a two-step analysis approach is proposed, first obtaining a simplified global response of the structure and then conducting a high-fidelity analysis for the visible components to generate accurate damage patterns. The main pieces of information derived from these analyses are the plastic strain contours, and other damage indicators such as the compression damage index from a concrete damaged plasticity model, which provide direct indicators for cracking and spalling of members – two of the main visual indicators of structural health after an earthquake. As the distribution of plastic strain is not likely to change for small changes in the loading, the number of analyses can be further reduced for large structures with little effect on the final result (i.e., the rendered PBGM) by taking advantage of the fact that components often repeat in a structure (e.g., across floors in a building). The next subsection describes the proposed methodology to identify physics-based damage hotspots using the non-linear analysis.

2.1.3 Damage masks generation

Damage masks are 2D binary images which indicate the presence of damage on component surfaces. Several damage parameters need to be determined before these masks can be generated using the conducted analysis. These parameters relate to the number, size, shape, and location of the damage. Each of the relevant parameters may be determined through, (i) physics-based response, or (ii) defined heuristics. Both these modes come with their own set of merits and demerits. Heuristic methods are the only viable option for many damage cases that are difficult to model (e.g., due to lack of suitable material models or load representations) or for which no empirical data is available. Methods stemming from empirical data are reliable because they are based directly on observations, but identifying good heuristics are challenging. When realizable, physics-based damage masks provide a rigorous approach that link the visual representation to results of finite element analyses, leveraging efforts by researchers in developing state of the art constitutive models. Incorporating the physics enables applications such as estimating structural response (e.g., interstory drift, damage state, etc.), failure mechanisms, and model updating. We first propose a general framework to determine damage parameters and then demonstrate generating masks for common damage types of cracks and spalling using the structural response.

The damage parameters are obtained by Monte Carlo sampling from empirical or heuristic distributions. The first step is to determine the damage state (DS) of the component based on some structural response measure δ . The response measure used may be anything that is sensitive to visual damage. For example, for reinforced concrete buildings with masonry infill, a commonly used damage indicator, the interstory drift may be used as the response measure. The relationship between DS and δ is then modeled through a probability distribution (1). This distribution represents uncertainties in the geometry, method of construction and material properties. The component damage state DS_o is determined by sampling from the distribution as shown in equation 1

$$DS_o \sim P_{\delta}(DS) \quad (1)$$

Various parameters q_{i_o} (e.g., number of cracks, crack width, crack length etc.) are then calculated by sampling from their corresponding distributions representing variation in damage observed given a particular damage state shown in equation (2).

$$q_{i_0} \sim P_{DS_i}(q_i) \quad (2)$$

While it may be possible to estimate the parameters q_{i_0} directly from δ , this two step approach allows for a more intuitive method facilitating the construction of the distributions P_δ and P_{DS_i} based on empirical data. For parameters whose value will vary depending on the location in the member, parameters are further modified by a multiplicative factor derived from the structural response as shown in equation (3).

$$Q_i(X, Y) = q_{i_0} f_i(X, Y) \quad (3)$$

where f is a function of some structural response parameter (e.g., plastic strain) varying in the component. Examples for selecting each of P_δ , P_{DS_i} , and f_i for RC buildings with masonry infill are provided in section 3. The next subsections discuss the generation of masks for cracks and spalls – two common types of defects once the damage parameters have been determined.

Stochastic blobs are amorphous shapes generated to select subregions of generated masks. The plastic strain map \mathbf{E} is normalized to take the form of a probability distribution \mathbf{P} . A center point (x_s, y_s) is obtained by sampling from the distribution. An amorphous blob shaped region \mathbf{S}_b is marked around the center point using a stochastic radius defined as the cumulative sum of a periodic function with random amplitude and phase. The blob generating function takes as input the number of waves along the circumference, w . The precise equations proposed can be found in Figure 8 where $\sim U[a, b]$ represents sampling from a uniform distribution between a and b .



Figure 6. Stochastic blob generation

For each component, the set of crack parameters are $\mathbf{q}_o = \{N, L, W\}$ where N is the number of cracks, L is the length of cracks in pixels, W is the crack widths in pixels. Once these parameters are determined, the following pipeline for generation of crack masks from the plastic strain map provided in Figure 7 can be applied. A gaussian blur is applied, followed by a Canny edge detector [39] to obtain an edge image. The edges are dilated by a factor of W . Finally, to add randomness to each component, a stochastic blob is generated and the intersection of the blob with the dilated edge image is included in the crack mask. This process is repeated N times.

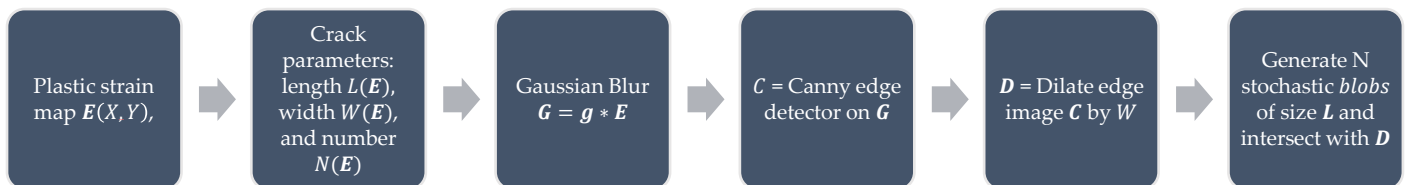


Figure 7. Crack mask generation process

Spalling is another common damage type for reinforced concrete and masonry structures affecting the integrity of components. The damage parameters to be determined here

are $\mathbf{q}_o = \{N_s, R\}$ where N_s is the number of spalled regions, and R is the nominal spall radius. To generate spall masks with these parameters, an area of pixels corresponding to the spall must first be defined. A stochastic blob S_b is generated following the process outlined in Figure 6. In addition to the blob, another region S_E is constructed corresponding to pixels with compression damage greater than the mean compression within the blob. The spall region S is then set as the intersection of S_E and S_b . Rebar is made visible under spalled regions with some probability $P_{DS_i}(q_{rebar})$. The process is illustrated in Figure 8.

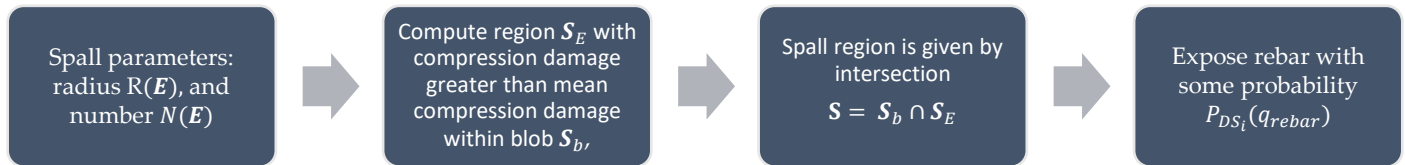


Figure 8. Pipeline for generation of spall masks from plastic strain

2.1.4 Damage Texture Generation

Damage textures are image textures of damaged components. Damage textures need to be generated so as to provide a realistic visual representation of the damaged structure. The following points are discussed to illustrate the process followed in generating damage textures: (i) Bidirectional scattering distribution functions, (ii) material textures, (iii) damage textures, and (iv) annotation textures.

Bidirectional scattering distribution functions: The visual appearance of an object is dependent on how light incident on its surfaces is scattered, transmitted or reflected. In computer graphics the behavior of light incident on a given material is represented by bidirectional scattering distribution function or BSDF [40]. BSDFs can be measured through optical experiments with structured light sources. Based on experiments, researchers have proposed different methods to model BSDFs. A widely implemented model available in many 3D creator software known as the Principled BSDF was proposed by Burely et al. [41] and is a physically-based rendering (PBR) model but with artistically intuitive parameters. Apart from the base color, BSDFs have 10 parameters to describe each pixel including properties of roughness, metallic, specular, anisotropy, etc. Depending on the type of material, several of these may not be applicable, for example a concrete surface may have negligible metallic scattering properties. In addition to these values defining the scattering, the incorporation of surface normal directions at every point play a significant role in accurate renderings. If the surface is modeled at a very small-scale incorporating undulations, then the values of the surface normal can be computed directly from the geometry. However, such detailed surface modeling is seldom feasible and an alternative way to retrieve the same effect is to use a predefined surface normal map.

Material textures: PBR textures encompassing maps with BSDF parameters for base color, roughness, metallicity, etc., and surface normals can be used to adequately represent materials for the purpose of structural inspection simulation. PBR textures for common construction materials incorporating BSDF parameters created through height field photogrammetry are available on websites like CC0textures [42]. A sample image texture of a brick wall rendered from [42] using Blender [43], an open source 3D model creation software is shown in Figure 9. The example incorporates three maps: the base color, a roughness map and the normal map. The roughness changes how the light is reflected, especially near the edges of the bricks and the normal map helps visualize the fine surface undulations and the protrusion of the bricks from the mortar plane. In addition to photogrammetry-based textures, textures can also be procedurally generated in material authoring programs like Substance [44] and provide the ability to create multiple textures with different random seeds. As noted in [29,38] randomization is a crucial means of

enforcing generalization. We utilize both types of PBR maps (photogrammetric and procedural) in the construction of the PBGMs. When multiple layers of materials are present, (e.g. cement plaster over masonry, paint over concrete, etc) maps are selected for each material layer and the displayed layer is selected based on the presence of damage at any given pixel.

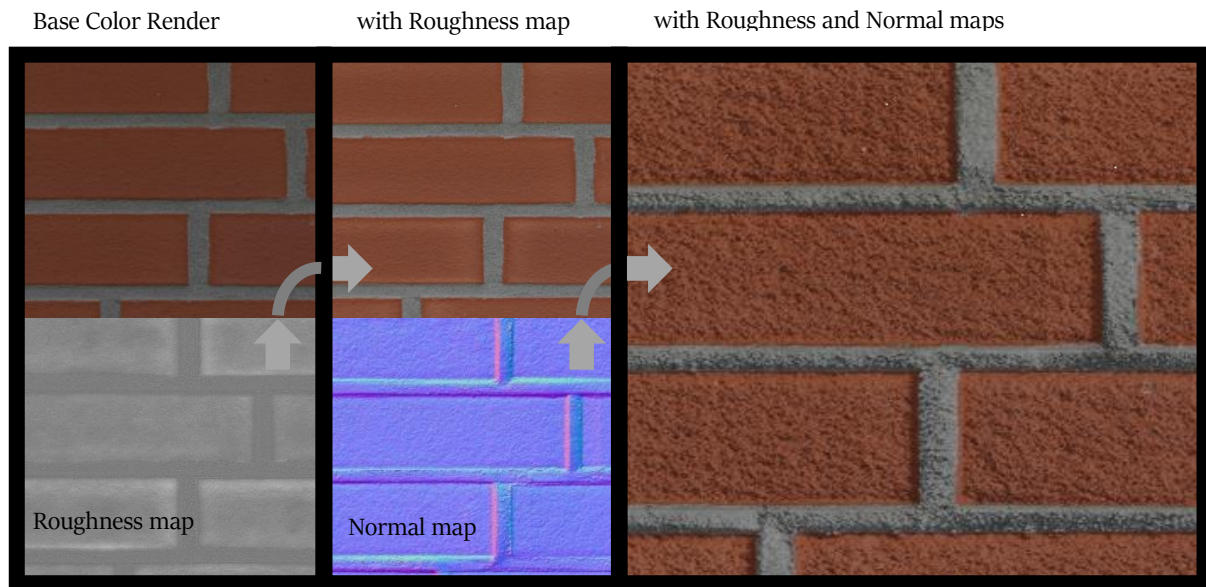


Figure 9. Illustration of PBR texture using base color, roughness and normal maps

Damage textures: The damage textures for the PBGM are obtained using the material textures as the base and modifying the region within the generated physics-based damage masks using opencv-python [45]. The crack is textured by modifying the corresponding surface normal through a bump map. The depth is set as a heuristic function of the plastic strain similar to the width and length and the crack. The spall region is defined by applying a Musgrave [46] texture to create a bump map controlling the variation of depth within the spall region. For reinforced concrete components, rebar is exposed depending on the damage state of the material with some probability p . The rebars are modeled as cylinders with surface variation and a metallic map.

Annotation textures: For deep learning methods, the ground truth synthetic data is rendered by using an emission BSDF. As opposed to the principled BSDF with 10 parameters, an emission BSDF has a single color parameter and acts as a light source. The emission shader is useful for rendering homogenous colors, which is what is required as ground truth for tasks like semantic segmentation. Depending on whether image data or annotation data is being rendered, the appropriate texture types are selected during the rendering process.

2.1.5 Texture application

The generated textures are applied to the components after UV unwrapping the components. For 3D models application of 2D textures requires a correspondence to be created such that 2D surfaces can map to corresponding locations on the 3D surface. This process of “unwrapping” the 3D model is termed UV unwrapping. UV unwrapping is conducted by selecting the edges that are to serve as seams to break up the 3D model. In most programs, once the seams are selected, the resulting 2D surfaces are then arranged to fit within a square surface. The obtained damage masks are also assembled in the same arrangement to create a direct correspondence to the UV map and thus to the 3D model. Other masks like the rebar mask are also arranged in the same way. Here, depending on

the aspect ratio of the component, the arrangement can take on a handful of configurations which are hard coded along with the dimensions of the corresponding component so that the rest of the process can be automated.

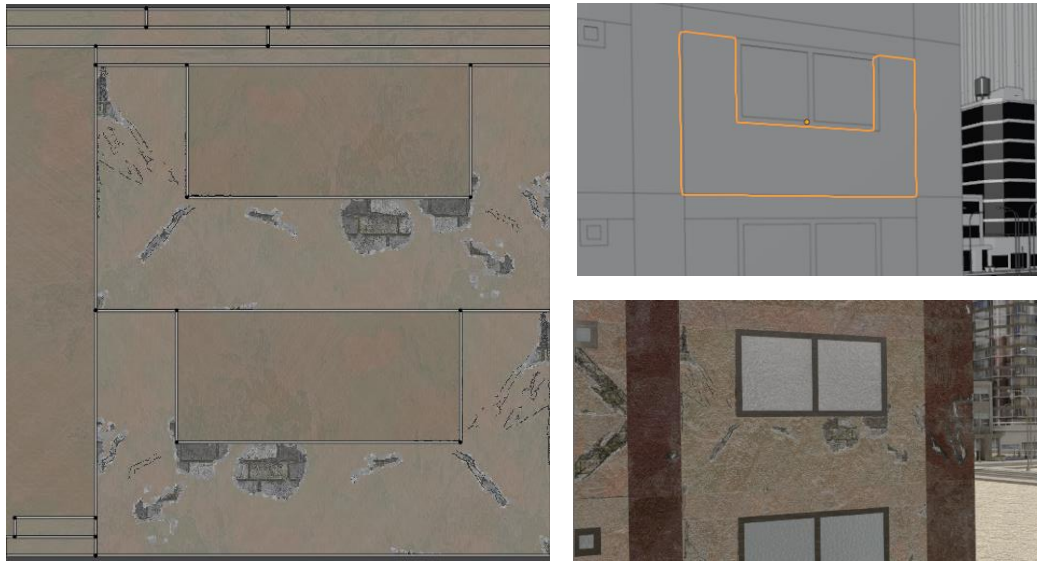


Figure 10. UV unwrapping and texture assignment to a wall

2.2 Scene, Lights, Camera & Rendering

The steps discussed thus far describe the construction of a single PBGM. To obtain photo-realistic images, the background scene also needs to be populated. For post-earthquake building inspections this includes multiple buildings, roads, sidewalks, light poles, electric cables, trees, etc. Randomization of geometry and textures is important towards the ultimate goal of generalizability of deep learning models trained in the synthetic environment. Thus procedural methods are adopted even in the scene assembly for the generation of these items.

The final step is to render the images. There are two modes of rendering commonly available in 3D creator software, namely path tracing and rasterization. Path tracing involves simulating the path of light in the scene and is more computationally expensive than rasterization but is preferred as it produces more photo realistic representations. To render images, a light source and the camera locations and orientations are to be set in the synthetic environment. To simulate realistic outdoor lighting, HDRI maps are used to light the scene [47].

3. Implementation of 3D synthetic environment with RC buildings

The proposed framework was implemented using multiple software applications. 3D model construction was conducted in Blender [43], the finite element analysis was conducted using ABAQUS [48], material authoring using Substance [44], and image processing in python using OpenCV [45]. This section discusses details about the construction of the synthetic environment.

3.1 Details of modeled buildings

3D models were created for 12 different fictional reinforced concrete buildings with masonry infill walls. The layout for these buildings were loosely 3 buildings (shown in Figure 11) that were affected in the Mexico City earthquake in 2017 with some simplifications made for parametric modeling. The buildings were parameterized and different realizations for each of the buildings were constructed with varying dimensions. Photographs of the buildings were obtained from three different sources: datacenterhub [49], google street view [50], and direct photography by the authors.

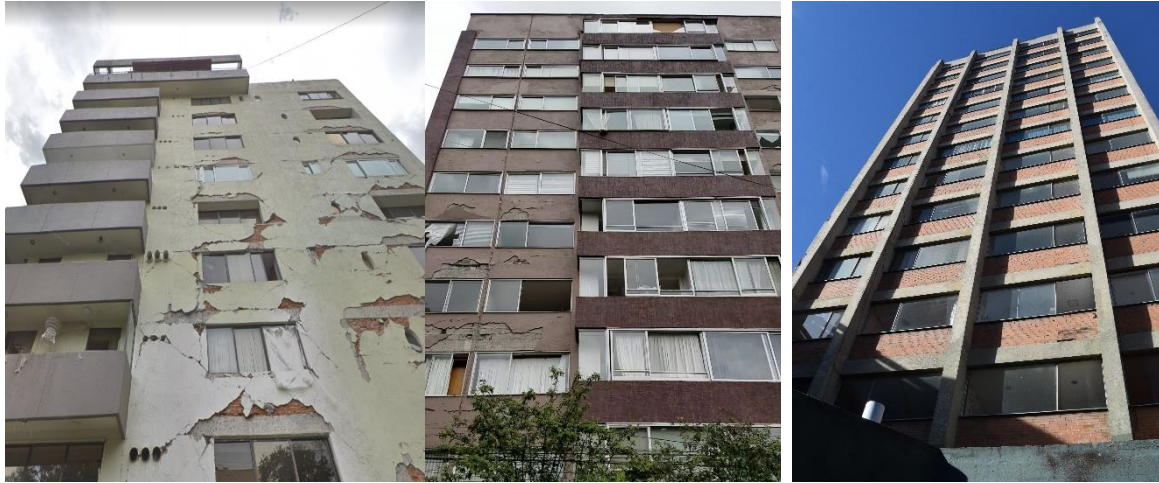


Figure 11. Three reference buildings damaged that suffered damage during the Mexico City earthquake in 2017

The dimensions and layout of the building were parameterized to include dimensions and locations of columns, beams, walls, windows, and balconies. The building properties were stored in a single class object that were used both for finite element model creation and 3D model generation.

3.2 Finite element model

As mentioned in section 2, both the global and local response of the structure are required for the generation of the PBGM. The global analysis of the buildings were conducted using OpenSeesPy [51]. The creation of the mesh was automated based on the building layout parameters developed in the previous section. The structure was modeled using the confined concrete model in OpenSeesPy with the parameters in Table 1.

Table 1. OpenSees model concrete material properties

Material	f'_c MPa (ksi)	f'_{cu} MPa (ksi)	ϵ_{sc0}'	ϵ_{su}'
Concrete01	41.36 (-6)	-34.4 (-5)	-0.004	-0.015

Reinforced concrete sections were created for the members using the patch, section and layer commands in OpenSees. The amount of rebar was set based on ACI 318-14 [52] assuming a c/d ratio of 0.3 in equations (4) and (5), where A_r is the reinforcement area, b, d are the section dimensions of the concrete member, ρ is the rebar ratio, $\beta = 0.85$, f'_c and f_y are the yield strengths of concrete and steel. A concrete cover of 40 mm was set ACI minimum of 1.5 in for beams and columns.

$$A_r = \rho b d \quad (4)$$

$$\rho = 0.85\beta \frac{c}{d} \frac{f'_c}{f_y} \quad (5)$$

The shear reinforcement was assumed to be at a spacing of a maximum of $(100, d/3)$. The shear strength from reinforcement is assumed to be $3\sqrt{f'_c}bd$ and the corresponding rebar area as per ACI 318 are given in equations (6).

$$A_v = \frac{3\sqrt{f'_c}bs}{f_y} \quad (6)$$

The first three global mode shapes of a parametrically generated building is shown in Figure 12

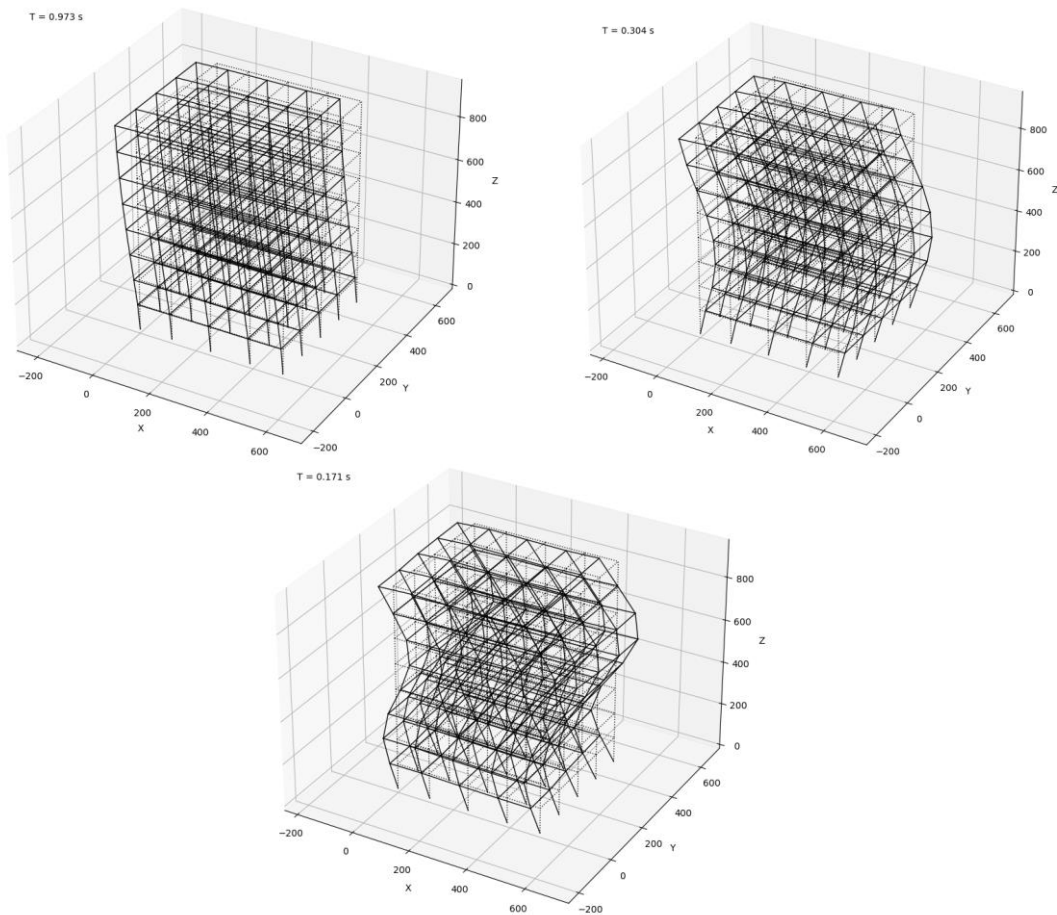


Figure 12. First three horizontal modes of a simulated structure

Each building was subject to the Tabas earthquake with varying intensity from $g/4$ to $g/6$. An example ground motion is shown Figure 13. A full analysis was conducted for the local response of the components using Abaqus. A python script was developed to automate the process of creation of the components of the structure. The components models included the masonry wall and confining columns and beams as shown in Figure 15 all modeled with solid elements. The model also included rebar which was modeled with beam elements with a circular cross-section. The nodal displacements at the corners of components were used as inputs for the detailed local component models. For the concrete and masonry members, the concrete damaged plasticity (CDP) model proposed by [53] was used. The material parameters used for the concrete material was based on values reported in Jankowiak et al. [54] and for the masonry material based on the values reported in Bolhassani et al. [55]. The masonry yield stress for tensile behavior was factored down so that the tensile strength of the masonry was less than that of the concrete. The steel was modeled as a plastic material with yield stress of 200 MPa. The stress-strain curves used are shown in Figure 15.

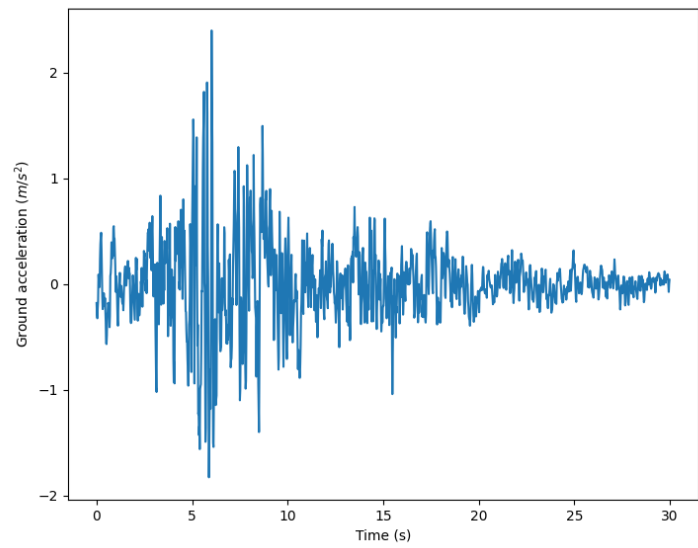


Figure 13. Ground motion input to global model with PGA $g/4$

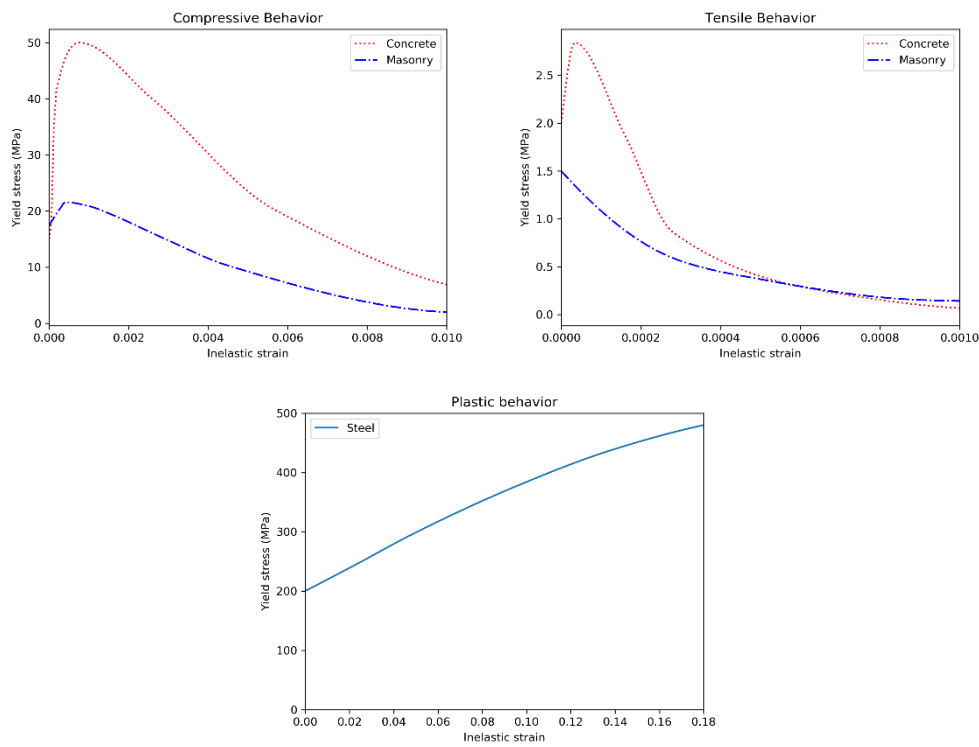


Figure 14. Plastic stress strain curves for concrete, masonry and steel used in the PBCMs

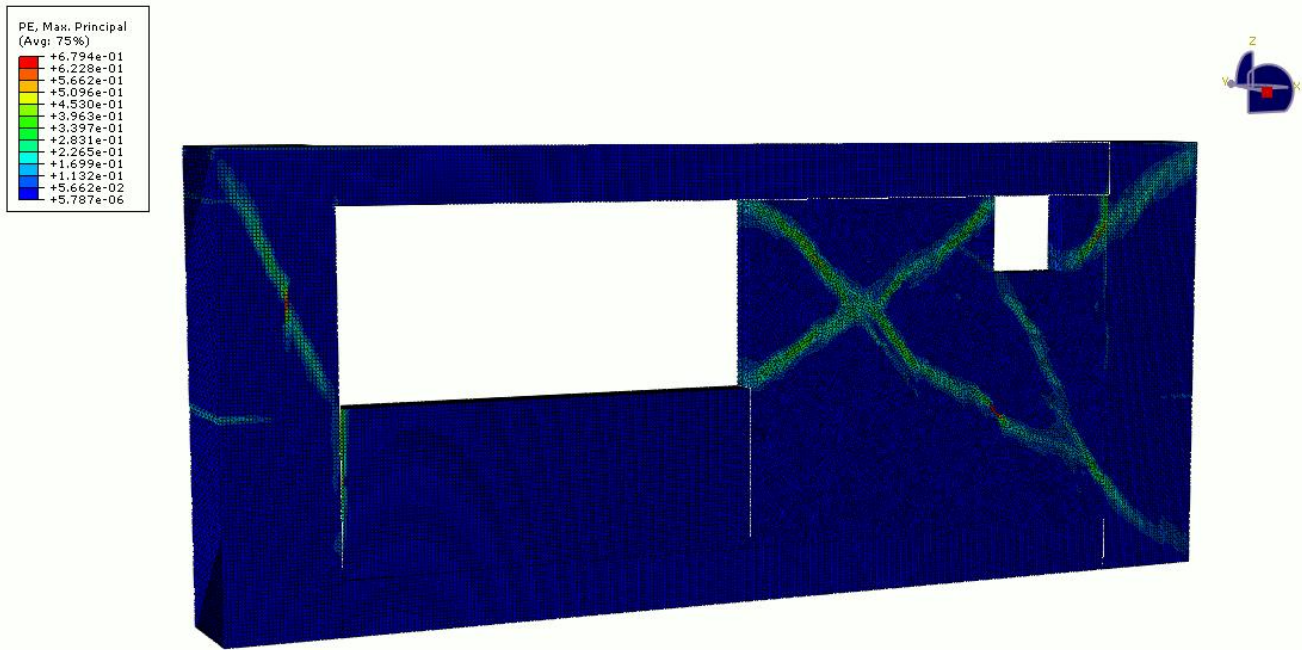


Figure 15. Response of component model in Abaqus including walls with window openings, columns and beams

The rebars are embedded within the concrete members using the embedment interaction option in ABAQUS. The walls are tied to their immediate confining members using the tie constraint in Abaqus. A multi-point constraint is applied to tie the top and bottom surfaces of the beams and columns together. The bottom surface is fixed, and the top surface is subject the interstory drift. The amplitudes are chosen to represent 4 different damage states derived based on values reported by [56].

An Abaqus explicit analysis was run for each unique component and the plastic strains at each of the amplitude levels are stored as an image for input to the texturing process discussed in the next subsection.

3.3 Damage parameters

As mentioned in section 2, the first step in identifying the damage parameters is to determine the damage state of the component. The probability distribution for different damage states given the interstory drift is taken from Chozzi et al. [56] where data from over 150 test on masonry walls subject to in-plane loading were analyzed. A log normal distribution is used to model the conditional probability of exceeding a given damage state as shown in equation (7). $\mu_{ln}(\delta)$ and β represent the central tendency and the dispersion parameters of the cumulative standard normal distribution Φ . The values used for the different damage states are presented in Table 2.

$$P(DS \geq ds_i | IDR = \delta) = 1 - \Phi \left(\frac{\ln(\delta) - \mu_{ln}(\delta)}{\beta} \right) \quad (7)$$

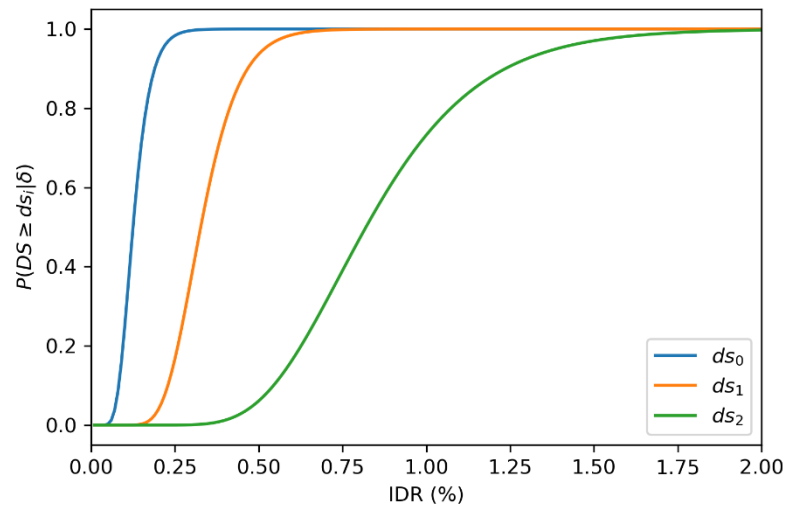


Table 2. Interstory drift ratio for different damage states

Damage description	Low	Moderate	Severe
Interstory drift ratio (IDR) %	0.125	0.25	0.82

Once the damage is determined for the components, the various damage parameters were computed by sampling from their corresponding lognormal distributions. The statistics of the distributions used are provided in Table 3 and the corresponding distributions are plotted in Figure 15. The values for the crack width are based on descriptions of damage states given in Chozzi et al. [56]. The crack length, height and number of cracks for different damage states are approximated based on descriptions given in FEMA 306 [57] based on the component damage classification guides for concrete frames with masonry infill. The spall radius ratio R_s and area A_s has been generalized for both walls and columns based on examples provided in [58]. In the presence of more rigorous experimental data, corresponding distributions may be replaced to better represent the damaged structure.

Table 3. Statistics for damage parameters

Damage Parameter	μ				β			
Damage state	ds_0	ds_1	ds_2	ds_3	ds_0	ds_1	ds_2	ds_3
Crack width W (mm)	0.01	2	3	5	1.5	0.15	0.15	0.2
Crack length ratio L	0.01	0.5	0.85	1	1.1	0.2	0.05	0.05
Number of cracks N_c	0.01	3	6	12	2	0.2	0.15	0.15
Spall radius ratio R_s	0.001	0.01	0.03	0.07	0.5	0.2	0.2	0.15
Spall area ratio A_s	0.01	0.05	0.12	0.25	0.3	0.2	0.15	0.2

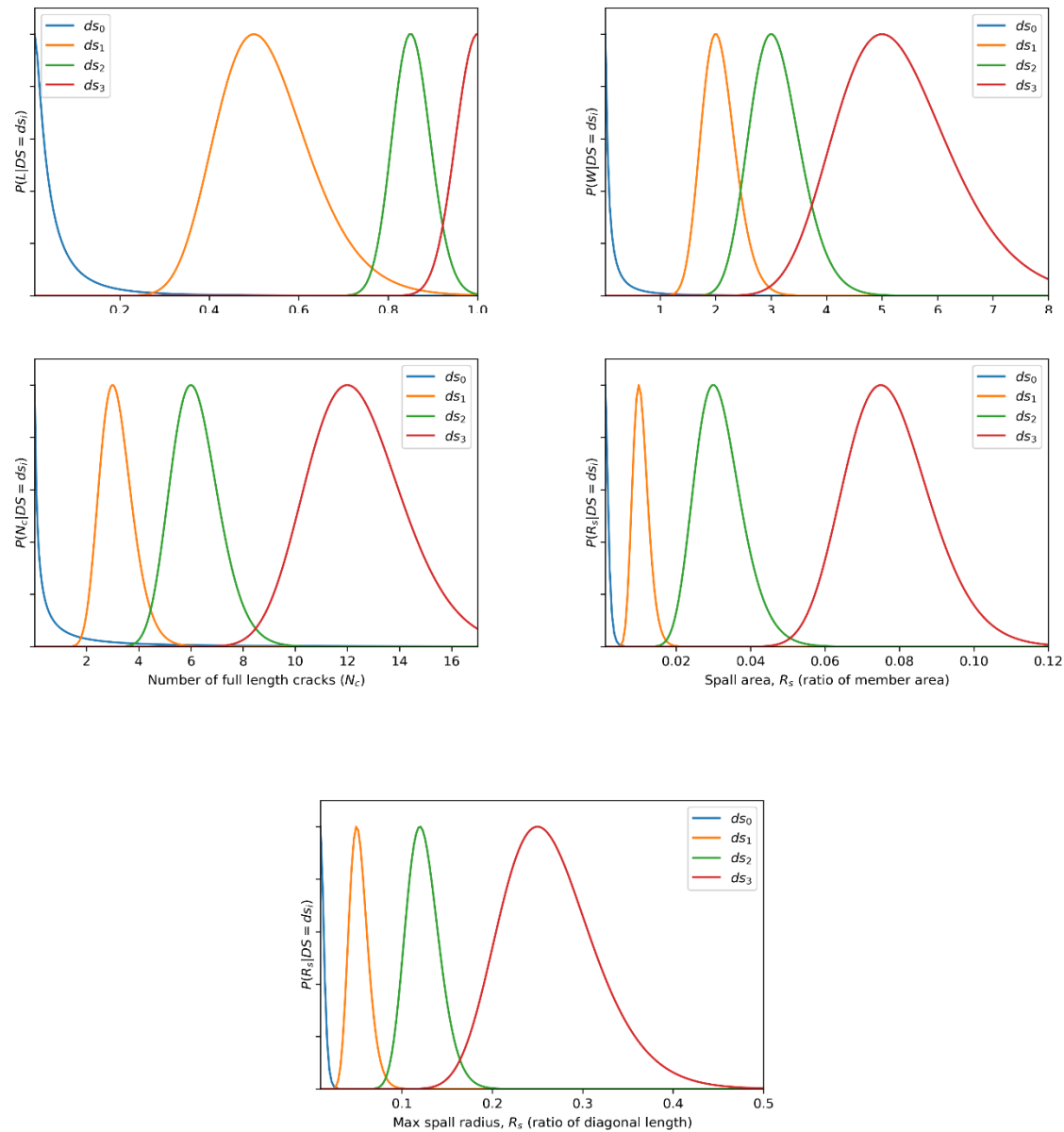


Figure 16. Visualization of damage parameter distributions

3.4 Texture generation

PBR textures are used for all the construction materials. The textures for the paint, walls, beams, and columns were all generated parametrically using Adobe Substance Designer [44]. The visual features parameterized include color properties, amount of dirt, types of dirt, and size and orientation of bricks. For each generated building structure, parameters including the paint color, concrete color, brick size, and brick color are first selected. Then for each component, the parameters are perturbed to provide variability for the components.

3.5 Assembly and lighting

The assembly and construction of the PBGM and synthetic environment are automated using python scripts. In each scene one PBGM building is created. Then, the sidewalks, trees, roads and other buildings are added to complete the scene using the Scene-City Blender plugin. The scene background and lighting was set using HDRI maps

downloaded from [42]. An emission shared was used for the annotations and the images were rendered using the cycles renderer.

4. Applications and experiments

The developed procedure for PBGMs is used to generate synthetic images that can be used for automated visual inspection studies. Three applications and examples are illustrated, (i) QuakeCity Dataset: Large-scale synthetic dataset of earthquake damaged buildings, (ii) Augmenting real data with synthetic data, and (iii) Comparing post-disaster UAV data acquisition with ground camera data acquisition.

4.1 QuakeCity Dataset: Large-scale synthetic dataset of earthquake damaged buildings

Images are rendered from multiple simulated UAV surveys of 11 damaged buildings in a city environment. Each survey replicates a field scenario where a UAV circles the building at different altitudes to cover the entire height, width, and length of the building. Each image captured by the simulated UAV is associated with six different sets of annotations including three damage masks (cracks, spalling, exposed rebar), components, component damage states, and a depth map. The QuakeCity dataset is released as part of the International Competition on Structural Health Monitoring in 2021 for project task 2 [59]. In total, 4,688 images and six annotations per image of size 1920×1080 are included in the dataset, with 3,684 for training, and 1004 for testing.

4.2 Augmenting real data with synthetic data

To reliably train an autonomous visual inspection system, a large amount of training data with damaged structures would be required. Frequently, however, the amount of such training data available is limited. Additionally, careful annotation of available images is also a challenge. In this experiment, we are interested in studying whether the incorporation of synthetic data in cases with limited availability of real data with annotations can help boost the accuracy of networks on unseen real data.

4.2.1 Real image dataset

A dataset for semantic segmentation of real earthquake damaged buildings was developed for the purpose of this study. The images were acquired by the authors after the 2017 Mexico City Earthquake using a DJI Phantom 3, and a Nikon D3300. The images were annotated for the presence of spalling using InstaDam [60]. In total, 150 images of resolution 1920×1080 were annotated as part of the dataset.

4.2.2 Network architecture

A deep network is constructed for semantic segmentation using a ResNet [61] architecture with 45 layers. The details of the encoder part of the architecture are provided in Figure 17. Residual connections involve the summation of the output of prior layers to enforce learning of new information in subsequent layers. These residual connections are used between alternate layers (e.g., Conv0 to Conv2, Conv2 to Conv4, etc.). A rectified linear unit is used as the non-linearity for all layers of the network. The details of the decoder part of the architecture are provided in Model training. The skip connections with 1×1 convolutions described in the previous subsection are taken after the Conv8, Conv20 and Conv32 layers. The network parameters were trained by minimizing the cross-entropy loss function between the predicted softmax probabilities and the corresponding one-hot labels with an L2-regularization weight decay [53]. The incorporation of the weight decay term gives preference to smaller weights and helps tackle overfitting. Batch normalization was applied to address the covariate shift that occurs during training [24], where each feature dimension is shifted by a weighted mean and standard deviation that was learned during training. The percentage of pixels in each of the classes varies significantly. For example, some classes such as cracks have much fewer pixels than spalling or corrosion due to the nature of damage. To balance the frequencies of different classes in the data set and prioritize all classes equally, median class balancing [26] was applied by reweighting each class in the cross-entropy loss. Data augmentation by resizing and cropping was incorporated in order to increase the efficacy and efficiency of training and

prevent issues such as overfitting. The training was conducted using the Adam optimizer [54] implemented in Pytorch [62].

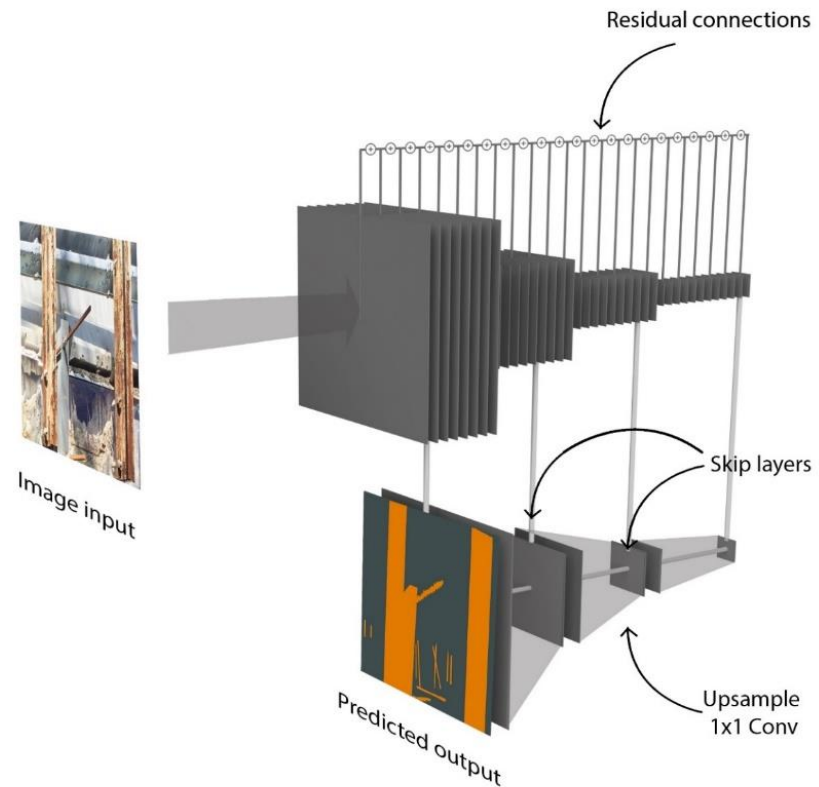


Figure 17. Schematic illustration of feature layers in the proposed FCN

4.2.3 Model training

Eight different models were trained to evaluate the potential role of synthetic data in enhancing the overall performance of the models on real data. The eight models included four pairs of training schemes where each scheme had one model trained purely on real data and another trained on real plus synthetic data. In each pair, the train/test split of real data was varied, starting from 0.2 train + 0.8 test, to 0.8 train + 0.2 test, in increments of 0.2. The same amount of synthetic training data was used in all four schemas, and this included the training images from the QuakeCity dataset (i.e., 3684 images).

4.3 Comparing damage state estimation using UAV and ground-based images

While implementing autonomous visual inspection systems after disasters, a trained model using a dataset conducted prior to the disaster would be used to process new data acquired after the disaster. The quality of the predictions on new data may however vary widely depending on the image acquisition distance. For example, it may not always be possible to have consistent data acquisition modes or distances for various structures of interest. This is especially so in crowded cities where many obstacles are present. To better study the robustness of the trained models, practitioners may want to evaluate the model's performance for different camera distances to see where data gaps are present in the model, or to inform their field acquisition strategies. In such a scenario, using a PBGM would prove very useful, as images could be acquired with different camera paths, and the accuracy of predictions of a fixed trained model can be studied.

In this experiment, we train two different ResNet 45 models to predict component damage states. One model is using only the QuakeCity training dataset and tested on images from another building. Two test sets are prepared, one simulating a UAV camera for data acquisition (UAV-B12), and another simulating a person on a ground collecting images of the structure by pointing the camera forward and upward (Ground-B12).

Together, the datasets are referred to as B12. Another model is trained with the QuakeCity training dataset plus 25% of the images from B12 (QuakeCity+0.25 B12), and evaluated on 75% of the B12 data (0.75 B12). The results of performance on the ground data are reported separately for the UAV and Ground parts of B12.

5. Results

The results from the experiments outlined in the previous section are now presented

5.1 QuakeCity Dataset

Example images of the generated dataset are shown in Figure 18. The images demonstrate the diversity of damaged buildings in the dataset in terms of layout, color, damage level. Images in the scenes are taken from different viewpoints and with different lighting conditions.

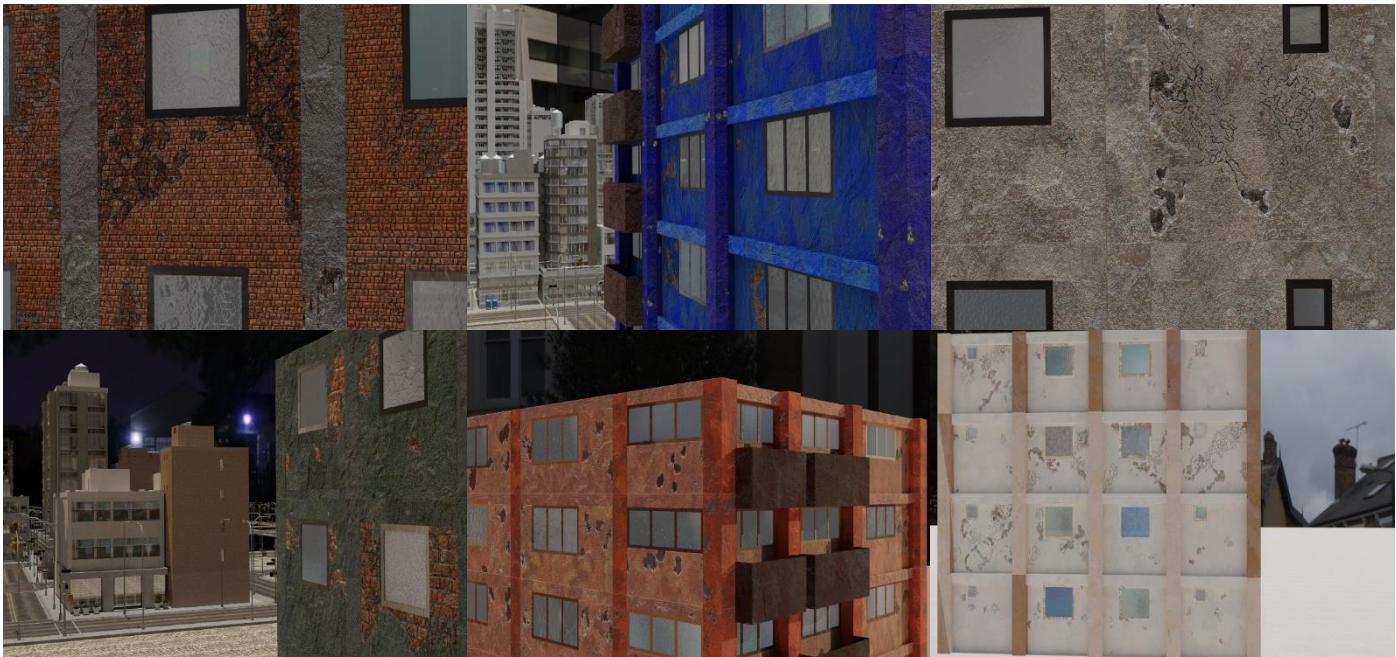


Figure 18. Example images from the QuakeCity Dataset

Figure 19 shows example annotations for an image in the QuakeCity dataset including component damage state, depth map, and component annotations. Figure 20. Shows another image generated with spalling, cracks and rebar annotations for each pixel. The annotation color key for both sets of annotations is provided in Table 4.

Table 4. Annotation color key

Damage state (DS)	DS0	DS 1	DS2	DS3		
Component	Wall	Column	Beam	Window	Frame	Balcony
Damage	Spalling	Crack	Rebar			

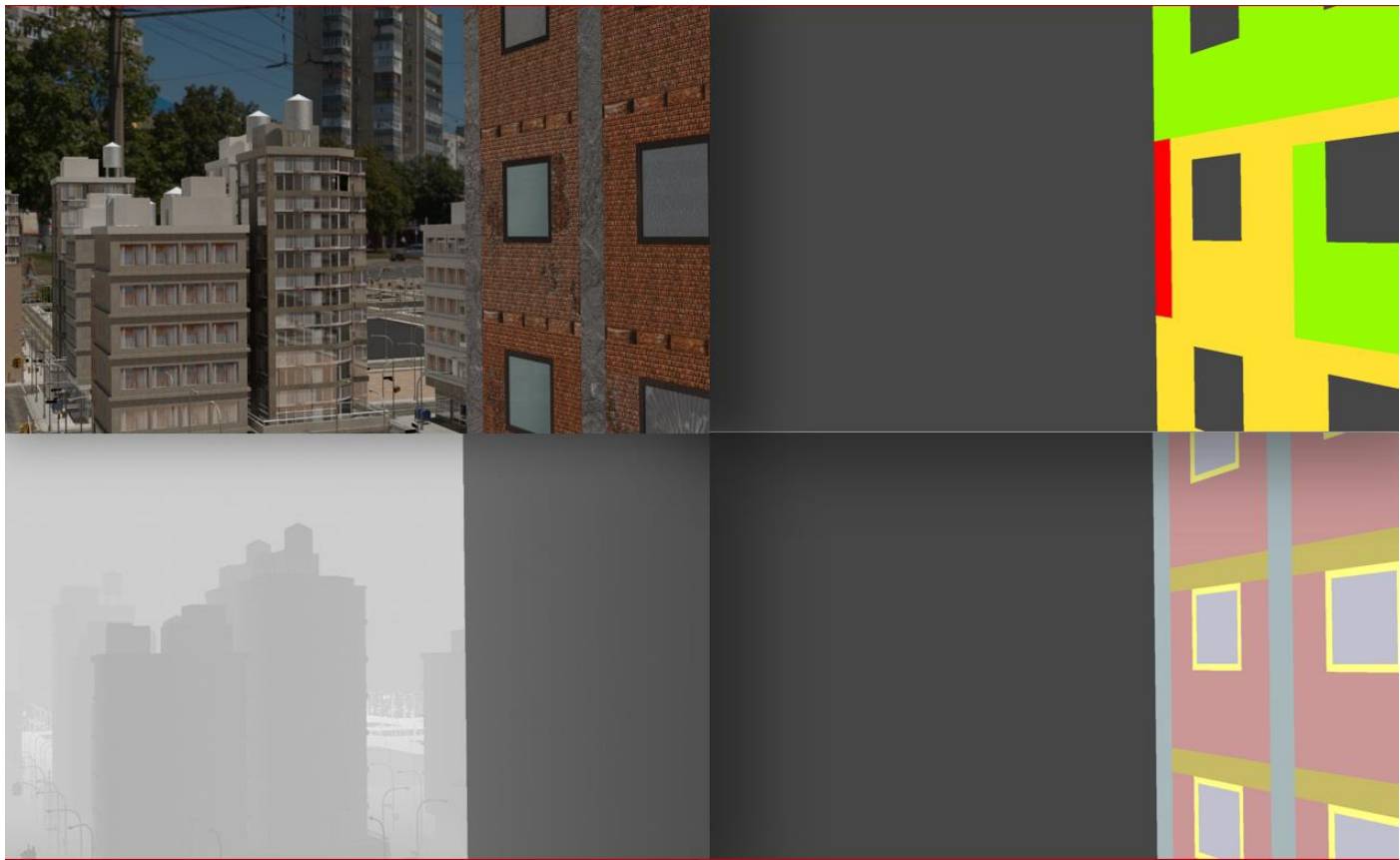


Figure 19. Example annotations (clockwise from top left) Image, Component Damage State, Components, Depth

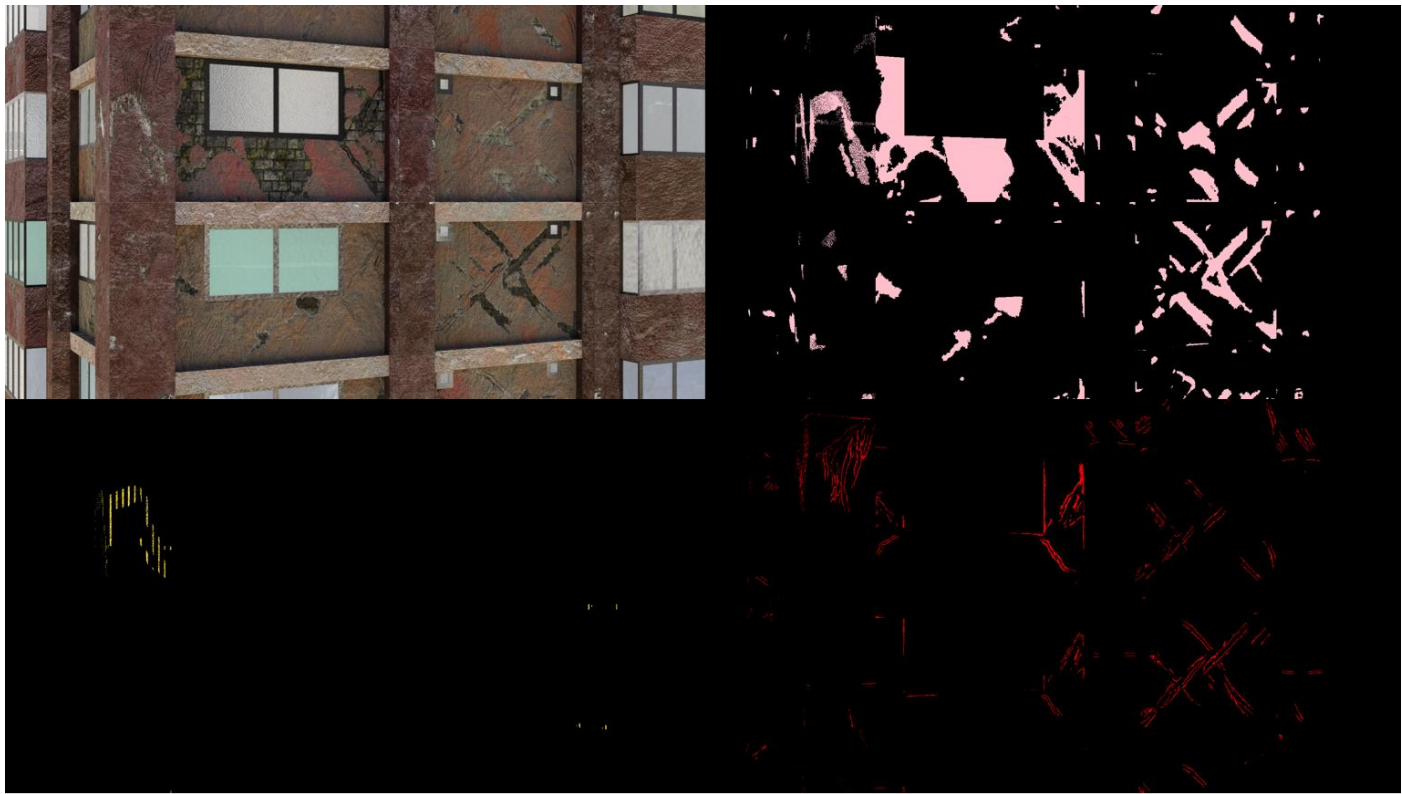


Figure 20. Example annotations (clockwise from top left) Image, Spalling, Cracks, Exposed Rebar

5.2 Augmenting real data with synthetic data

The results from the different models trained are shown in Figure 21 and Figure 22. Figure 21 shows the comparison of test accuracy on 60% of real data while training on 40% of the real images with and without QuakeCity data. While the initial accuracy with only real data is higher than with QuakeCity, after about 75 epochs, it was noticed that there was a significant increase in the performance of the model trained with QuakeCity data. The performance of the model clearly highlights the benefits of using synthetic data to improve the performance of deep learning models on unseen real data.

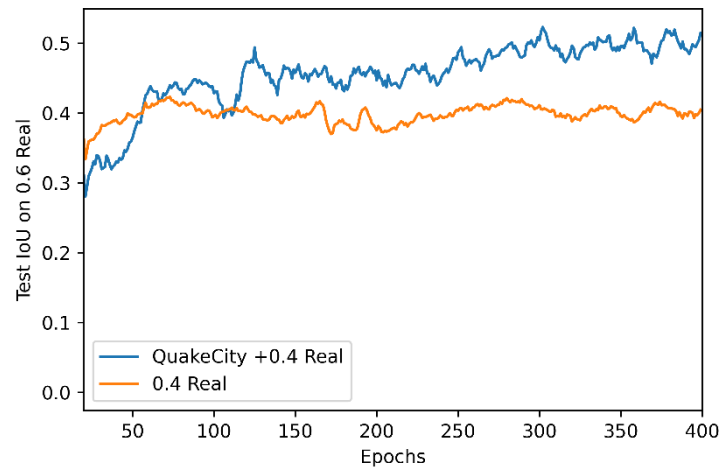


Figure 21. Comparison of test set accuracy on 60% of real data while training on 40% of the real images with and without QuakeCity data

The addition of synthetic data was also shown to improve the performance of the deep neural network for even for varying splits of training and testing data. Figure 22 shows the difference between the two values plotted in Figure 21, for all four models trained. The performance all models trained with the QuakeCity dataset is better than the model without the QuakeCity data after 400 epochs. The improvement in IoU is seen to be as much as 10%. Table 5 shows examples of images where the 0.4 Real model with QuakeCity data performs better than the model without. The quality of the predictions is clearly improved, and the border of the predictions can be seen to be more accurate.

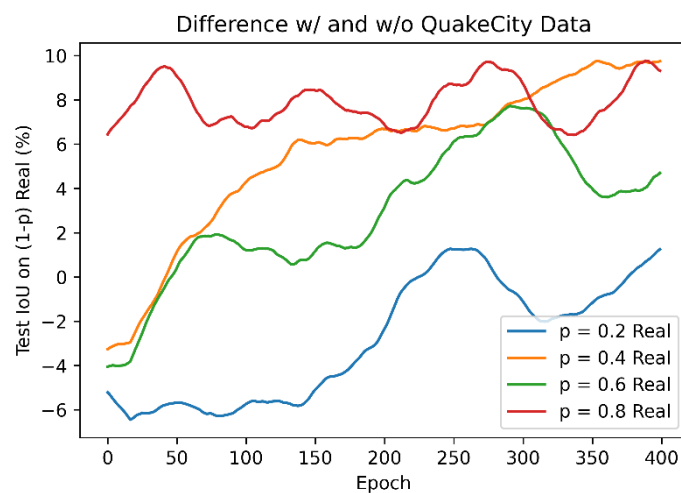












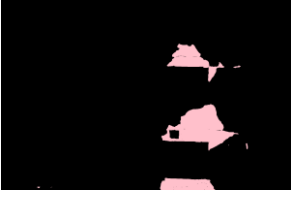





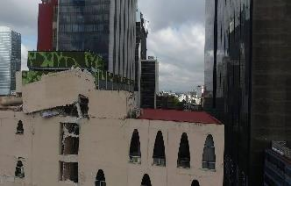





Figure 22. Difference between test accuracy with and without QuakeCity data for varying fractions of real training data

Table 5. Qualitative comparison of results with and without QuakeCity training data

Real Image	Ground truth	Real 0.4	QuakeCity + Real 0.4
			
IoU	N/A	0.40838659	0.84830129
			
IoU	N/A	0.23776706	0.60649584
			
IoU	N/A	0.56585139	0.89356358
			
IoU	N/A	0.3649658	0.65546518
			
IoU	N/A	0.60729252	0.89446025
			
IoU	N/A	0.54610351	0.76673772

5.3 Comparing damage state estimation using UAV and ground-based images

Table 6 shows the test IoU for different damage states for the various models trained. The model trained on the QuakeCity dataset only, which is limited to UAV views performs poorly on Ground B12 images. As a comparison, the performance of the model on 75% of UAV B12 is also shown. With the addition of 25% of B12 to the training dataset, the model performs much better on the remaining 75% of the data and is much closer to the performance on 75% of the UAV B12 set. While the results are along expected lines, the study nonetheless highlights the benefits of using a PBGM for tasks where the value and type of additional information to be incorporated into the network needs to be quantified. Given that there will be some cost associated to incorporating new data into the training dataset, a performance-based approach for data inclusion can be set-up using a PBGM as a reference. Table 7 and Table 8 show examples of predictions for the Ground and UAV B12 test datasets respectively.

Table 6. Comparing damage state estimation using UAV and ground-based images

Train	Test	ds1	ds2	ds3
QuakeCity	Ground B12	0.08	0.49	0.10
QuakeCity + 0.25 B12	0.75 UAV B12	0.40	0.72	0.79
QuakeCity + 0.25 B12	0.75 Ground B12	0.40	0.65	0.71

Table 7. Results for Ground B12 images




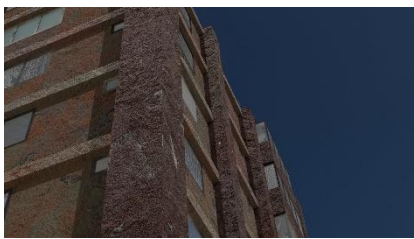
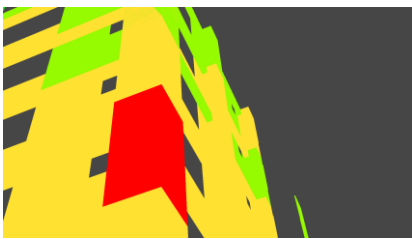
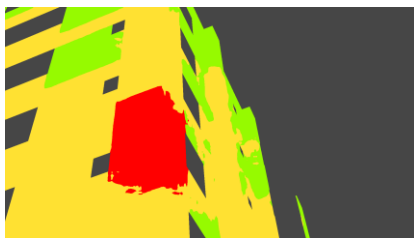

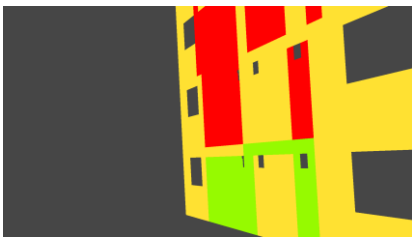
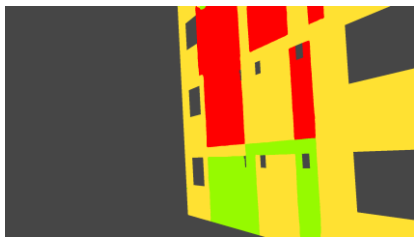
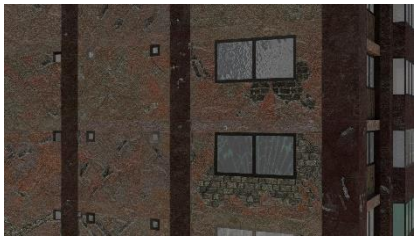
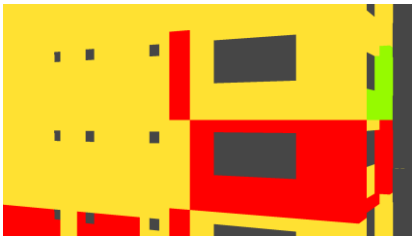
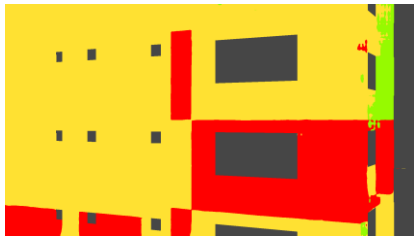




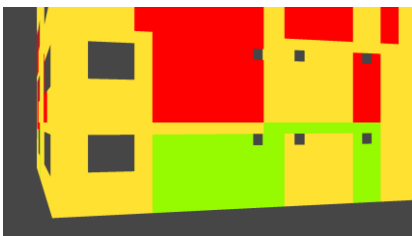
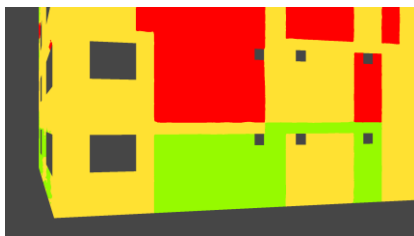
Image from Ground B12	Ground Truth	Inference
		
		
		

Table 8. Results for UAV B12 images

Image from UAV B12	Ground Truth	Inference
		
		
		

6. Conclusions

This paper proposed a framework for generating physics-based graphics models (PBGMs) as part of a 3D synthetic environment that can support development of automated inspection strategies for civil infrastructure. The proposed framework involved combining response of a non-linear finite element model to inform the realistic visual rendering of different damage types. The framework was implemented for eleven reinforced concrete building structures subject to earthquake excitation and the damage types rendered included cracks, spalling, and exposed rebar. Three applications were demonstrated for the proposed framework. First, images were rendered from the damaged structures, pixel-level ground truth was generated for the various damage types, for components, component damage states, and depths. The dataset generated, termed QuakeCity, was released as part of the International Competition on Structural Health Monitoring 2021. The QuakeCity dataset will serve as a benchmark dataset to study the use of deep learning algorithms in automated post-earthquake inspections of building structures. Second, the efficacy of the proposed framework in generating synthetic data to augment real data was demonstrated. It was shown that the performance of models trained with synthetic data and real data performed up to 10 IoU points better than models trained with only real data. Finally, a third experiment was conducted comparing the performance of trained models on ground and UAV-based data. The experiment demonstrated the utility of the proposed framework for studying and quantifying the value of additional information for models trained for visual inspections. The results demonstrate the immense potential of using PBGMs as an end-to-end tool for the development and study of visual inspection systems.

Author Contributions: Conceptualization, VH, YN and BFS; methodology, VH and YN.; software VH.; validation, VH.; resources, BFS and VH.; data curation, VH.; writing—original draft preparation, VH.; writing—review and editing, VH, YN and BFS. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Please refer to suggested Data Availability Statements in section “MDPI Research Data Policies” at <https://www.mdpi.com/ethics>. You might choose to exclude this statement if the study did not report any data.

Acknowledgments: The authors would like to thank Shengyi Wang for his help in annotating the real datasets.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Christopher Rojahn; David R. Bonneville; Nick Delli Quadri; Maryann T Phipps; Richard A Ranous; James E Russell; William E Staehlin; Zan Turner ATC-20-1 Field Manual: Postearthquake Safety Evaluation of Buildings. *Appl. Technol. Counc.* **2005**.
2. Mexico City earthquake: What we know Available online: <https://www.usatoday.com/story/news/world/2017/09/20/what-we-know-mexico-earthquake/684113001/> (accessed on Nov 16, 2020).
3. Eulich, W. Mexico’s Long Wait for Building Inspections Available online: <https://www.usnews.com/news/best-countries/articles/2017-09-27/mexicans-face-a-daunting-wait-for-building-inspections-after-earthquakes> (accessed on Jul 5, 2018).
4. Colegio de Ingenieros Civiles de México (CICM) Resumen preliminar de danos de los inmuebles inspeccionados por las brigadas del CICM del sismo del 19/09/2017 Available online: <https://www.sismosmexico.org/informes> (accessed on Apr 1, 2018).
5. Thousands homeless after Mexico earthquake Available online: <https://www.13abc.com/content/news/Thousands-homeless-after-Mexico-earthquake-447419433.html> (accessed on Nov 16, 2020).
6. 6 months after Mexico quake some still camp outside homes – The Denver Post Available online: <https://www.denverpost.com/2018/03/20/mexico-earthquake-homeless-camps/> (accessed on Nov 16, 2020).
7. 2015 Nepal earthquake: Facts, FAQs, and how to help | World Vision Available online: <https://www.worldvision.org/disaster-relief-news-stories/2015-nepal-earthquake-facts> (accessed on Jun 2, 2020).
8. Italian earthquake: 40 dead and 50,000 homeless - Telegraph Available online: <https://www.telegraph.co.uk/news/worldnews/europe/italy/5113762/Italian-earthquake-40-dead-and-50000-homeless.html> (accessed on Jun 2, 2020).
9. New Zealand earthquake “damaged 100,000 homes” - BBC News Available online: <https://www.bbc.com/news/world-asia-pacific-11191105> (accessed on Jun 2, 2020).
10. Yeum, C.M. Computer vision-based structural assessment exploiting large volumes of images. *Theses Diss. Available from ProQuest* **2016**.
11. Ghosh Mondal, T.; Jahanshahi, M.R.; Wu, R.; Wu, Z.Y. Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance. *Struct. Control Heal. Monit.* **2020**, *27*, doi:10.1002/stc.2507.
12. Ren, S.; He, K.; Gershick, R.; Sun, J.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *39*, 91–99, doi:10.1109/TPAMI.2016.2577031.
13. Xu, Y.; Wei, S.; Bao, Y.; Li, H. Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network. *Struct. Control Heal. Monit.* **2019**, *26*, e2313, doi:10.1002/stc.2313.
14. Hoskere, V.; Narazaki, Y.; Hoang, T.A.; Spencer, B.F. Vision-based Structural Inspection using Multiscale Deep Convolutional Neural Networks. In Proceedings of the 3rd Huixian International Forum on Earthquake Engineering for Young Researchers, University of Illinois, Urbana-Champaign; Urbana, IL, USA, 2017.
15. Hoskere, V.; Narazaki, Y.; Hoang, T.A.; Spencer, Jr., B.F. MaDnet: Multi-task Semantic Segmentation of Multiple types of Structural Materials and Damage in Images of Civil Infrastructure. *J. Civ. Struct. Heal. Monit.* **2020**.
16. Hoskere, V.; Narazaki, Y.; Hoang, T.A.; Spencer Jr., B.F. Towards Automated Post-Earthquake Inspections with Deep

- Learning-based Condition-Aware Models. In Proceedings of the The 7th World Conference on Structural Control and Monitoring, 7WCSCM, July 22-25, 2018, Qingdao, China; 2018.
17. Narazaki, Y.; Hoskere, V.; Hoang, T.A.; Spencer Jr., B.F. Vision-based automated bridge component recognition integrated with high-level scene understanding. In Proceedings of the The 13th International Workshop on Advanced Smart Materials and Smart Structures Technology (ANCRiSST), Tokyo, Japan; 2017.
 18. Narazaki, Y.; Hoskere, V.; Hoang, T.A.; Fujino, Y.; Sakurai, A.; Spencer, B.F. Vision-based automated bridge component recognition with high-level scene consistency. *Comput. Civ. Infrastruct. Eng.* **2019**, mice.12505, doi:10.1111/mice.12505.
 19. Narazaki, Y.; Hoskere, V.; Hoang, T.A.; Spencer Jr., B.F. Automated Bridge Component Recognition using Video Data. In Proceedings of the The 7th World Conference on Structural Control and Monitoring, 7WCSCM; 2018.
 20. Gao Khalid Mosalam, Y.M. *PEER Hub ImageNet (↗-Net): A Large-Scale Multi-Attribute Benchmark Dataset of Structural Images*; 2019;
 21. Liang, X. Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Comput. Civ. Infrastruct. Eng.* **2018**, doi:10.1111/mice.12425.
 22. Shafiei Dizaji, M.; Harris, D. 3D InspectionNet: a deep 3D convolutional neural networks based approach for 3D defect detection on concrete columns. In Proceedings of the Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XIII; Gyekenyesi, A.L., Ed.; SPIE, 2019; Vol. 10971, p. 13.
 23. Pan, X.; Yang, T.Y. Postdisaster image-based damage detection and repair cost estimation of reinforced concrete buildings using dual convolutional neural networks. *Comput. Civ. Infrastruct. Eng.* **2020**, 35, 495–510, doi:10.1111/mice.12549.
 24. Spencer Jr., B.F.; Hoskere, V.; Narazaki, Y. Advances in Computer Vision-based Civil Infrastructure Inspection and Monitoring. *Engineering* **2019**, doi:10.1016/j.ENG.2018.11.030.
 25. Gazebo Available online: <http://gazebo.org/> (accessed on Oct 26, 2020).
 26. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In; 2018; pp. 621–635.
 27. Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A.M. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* **2016**, 3234–3243.
 28. Dosovitskiy, A.; Ros, G.; Codevilla, F.; López, A.; Koltun, V. *CARLA: An Open Urban Driving Simulator*; 2017;
 29. Bewley, A.; Rigley, J.; Liu, Y.; Hawke, J.; Shen, R.; Lam, V.-D.; Kendall, A. *Learning to Drive from Simulation without Real World Labels*;
 30. Richter, S.R.; Darmstadt, T.U.; Anu, Z.H.; Koltun, V. *Playing for Benchmarks*;
 31. Hoskere, V.; Narazaki, Y.; Jr., B.F.S.; Smith, M.D. Deep learning-based damage detection of miter gates using synthetic imagery from computer graphics.
 32. Hoskere, V.; Narazaki, Y.; Spencer Jr., B.F. Learning to Detect Important Visual Changes for Structural Inspections using Physics-based Graphics Models. In Proceedings of the Submitted to the 9th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII-9); 2019.
 33. Hoskere, V. Developing autonomy in structural inspections through computer vision and graphics, University of Illinois at Urbana-Champaign, 2021.
 34. Narazaki, Y.; Hoskere, V.; Yoshida, K.; Spencer, B.F.; Fujino, Y. Synthetic environments for vision-based structural condition assessment of Japanese high-speed railway viaducts. *Mech. Syst. Signal Process.* **2021**, 160, 107850, doi:10.1016/j.YMSSP.2021.107850.
 35. Narazaki, Y.; Hoskere, V.; Eick, B.A.; Smith, M.D.; Spencer, B.F. Vision-based dense displacement and strain estimation of miter gates with the performance evaluation using physics-based graphics models. *Smart Struct. Syst.* **2019**, 24, 709–721, doi:10.12989/SSS.2019.24.6.709.
 36. Narazaki, Y.; Gomez, F.; Hoskere, V.; Smith, M.D.; Spencer, B.F. Efficient development of algorithms for vision-based dense

- 3D displacement measurement using physics-based graphics models. *Submitt. to J. Struct. Heal. Monit.* **2020**.
37. Zdziebko, P.; Holak, K. Synthetic Image Generation Using the Finite Element Method and Blender Graphics Program for Modeling of Vision-Based Measurement Systems. *Sensors* **2021**, Vol. 21, Page 6046 **2021**, 21, 6046, doi:10.3390/S21186046.
 38. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. *Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World*;
 39. Szeliski, R. *Computer Vision*; Texts in Computer Science; Springer London: London, 2011; ISBN 978-1-84882-934-3.
 40. Bartell, F.O.; Dereniak, E.L.; Wolfe, W.L. The Theory And Measurement Of Bidirectional Reflectance Distribution Function (Brdf) And Bidirectional Transmittance Distribution Function (BTDF)</title> In Proceedings of the Radiation Scattering in Optical Systems; Hunt, G.H., Ed.; SPIE, 1981; Vol. 0257, pp. 154–160.
 41. Burley, B.; Disney, W.; Studios, A. *Physically Based Shading at Disney*;
 42. CC0 Textures - Public Domain PBR Materials Available online: <https://cc0textures.com/> (accessed on Jul 11, 2020).
 43. Blender Available online: <https://www.blender.org/> (accessed on Aug 1, 2018).
 44. Adobe Substance by Adobe Available online: <https://www.substance3d.com/> (accessed on May 16, 2019).
 45. OpenCV Available online: <https://opencv.org/> (accessed on Mar 12, 2020).
 46. Musgrave, F.K.; Kolb, C.E.; Mace, R.S. The synthesis and rendering of eroded fractal terrains. In Proceedings of the Proceedings of the 16th annual conference on Computer graphics and interactive techniques - SIGGRAPH '89; ACM Press: New York, New York, USA, 1989; Vol. 23, pp. 41–50.
 47. Reinhard, E.; Heidrich, W.; Pattanaik, S.; Debevec, P.; Ward, G. *High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting*; 2010; ISBN 9780123749147.
 48. © Dassault Systèmes Abaqus 6.14 Online Documentation Available online: <http://abaqus.software.polimi.it/v6.14/index.html> (accessed on Sep 15, 2017).
 49. datacenterhub.org Available online: <https://datacenterhub.org/> (accessed on Aug 6, 2017).
 50. Discover Street View and contribute your own imagery to Google Maps. Available online: <https://www.google.com/streetview/> (accessed on Jul 12, 2020).
 51. The OpenSeesPy Library — OpenSeesPy 3.3.0.0 documentation Available online: <https://openseespydoc.readthedocs.io/en/latest/> (accessed on Sep 27, 2021).
 52. 318-14: Building Code Requirements for Structural Concrete and Commentary Available online: <https://www.concrete.org/store/productdetail.aspx?ItemID=318U14&Language=English> (accessed on Jul 12, 2020).
 53. Lee, J.; Fennes, G.L. Plastic-Damage Model for Cyclic Loading of Concrete Structures. *J. Eng. Mech.* **1998**, 124, 892–900, doi:10.1061/(ASCE)0733-9399(1998)124:8(892).
 54. Jankowiak, T.; Lodygowski, T. Identification of parameters of concrete damage plasticity constitutive model. *Found. Civ. Environ. ...* **2005**, 53–69.
 55. Bolhassani, M.; Hamid, A.A.; Lau, A.C.W.; Moon, F. Simplified micro modeling of partially grouted masonry assemblages. *Constr. Build. Mater.* **2015**, 83, 159–173, doi:10.1016/j.conbuildmat.2015.03.021.
 56. Chiozzi, A.; Miranda, E. Fragility functions for masonry infill walls with in-plane loading. *Earthq. Eng. Struct. Dyn.* **2017**, 46, 2831–2850, doi:10.1002/eqe.2934.
 57. Washington, D.C. *FEMA 306 EVALUATION OF EARTHQUAKE DAMAGED CONCRETE AND MASONRY WALL BUILDINGS Basic Procedures Manual Applied Technology Council (ATC-43 Project) The Partnership for Response and Recovery*; 1998;
 58. Tu, Y.H.; Ao, L.C.; Jean, W.Y. *Study on the Earthquake Damage Evaluation Procedure for RC and Confined Masonry Buildings*;
 59. IC-SHM 2021 – Structures and Artificial Intelligence Lab Available online: <https://sail.cive.uh.edu/ic-shm2021/> (accessed on Oct 17, 2021).
 60. Hoskere, V.; Friedel, D.; Amer, F.; Li, Z.; Yang, W.; Salinamakki, N.; Srikanth, P.; Tang, Y.; Narazaki, Y.; Spencer, Jr., B.F.

InstaDam: A software tool for rapid semi-automated pixel-wise annotation of defects in images.

61. Wu, S.; Zhong, S.; Liu, Y. Deep residual learning for image steganalysis. *Multimed. Tools Appl.* 2017, 1–17.
62. PyTorch Available online: <https://pytorch.org/> (accessed on Oct 18, 2021).