



Original software publication

Image2DEM: A geometrical digital twin generator for the detailed structural analysis of existing masonry infrastructure stock

Dimitrios Loverdos, Vasilis Sarhosis*

School of Civil Engineering, University of Leeds, LS2 9JT, Leeds, UK



ARTICLE INFO

Article history:

Received 5 October 2022

Received in revised form 17 January 2023

Accepted 19 January 2023

Keywords:

Masonry

Python

Image

Structural analysis

ABSTRACT

Assessing the structural performance of ageing masonry infrastructure is a complex task. Geometric characteristics and the presence of damage in masonry structures may influence greatly their rate of degradation and in-service mechanical response. Therefore, identifying approaches to assess the actual structural condition of these assets is vital. In the last ten years, advances in laser scanning and photogrammetry have started to drastically change the building industry since such techniques are able to capture rapidly and remotely digital records of objects and features in points cloud and image format. However, the direct and automatic exploitation of images for use as geometry in high fidelity models for structural analysis is limited. In this framework, the aim of this paper is to present the development of a software able to fully automate the “scan to structural modelling” procedure for the efficient and accurate structural assessment of ageing masonry infrastructure. “Image2DEM” is based on Python libraries with graphical interface. The images can be captured from DSLR (Digital Single-Lens Reflex) cameras, smartphones, or drones. The image selected is then imported to the programme to detect and extract the masonry micro-geometry. The algorithm provides reliable detection using Artificial Intelligence. Convolutional Neural Networks (CNN) are used to identify the location of masonry units and cracks, with ~96% and ~80% accuracy, respectively. The geometry is extracted in the form of simplified lines to improve efficiency and reduce computational effort. The output is provided in DXF format for compatibility between different programmes. Finally, the geometry extracted is converted to a numerical model for structural analysis. The proposed software has the potential to revolutionize the way we assess existing masonry infrastructure in the future.

© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Motivation and significance

Masonry infrastructure, such as bridges, viaducts and tunnels form a significant part of the UK's critical infrastructure stock; e.g. there are more than 70,000 masonry arch bridges which constitute over 40% of UK's bridge stock [1]. The majority of our masonry infrastructure is ageing, often well beyond 120 years, and showing significant signs of deterioration and damage. Weathering, demands of increasing load-intensity, axle-loads, and factors such as increased frequency of flood events due to climate change have introduced extreme uncertainty in the long-term performance of such infrastructure assets. Also, much of our masonry infrastructure has significant heritage and cultural value (e.g., the Grade II-listed Hungerford Canal Bridge, in Berkshire, England) and the UK has a policy to “retain and repair”, rather than “demolish and replace”. Failure of such infrastructure could lead to direct and indirect costs to the economy and society and hamper rescue and recovery efforts. For example, during the

2009 floods in Cumbria, three masonry arch bridges collapsed while nine were severely damaged, leading to nearly £34 m in repair and replacement costs. The economic and societal impact were even larger, with increased travel time estimated to cost the economy almost £2 m per week. In March 2017, approximately 200 tonnes of rubble fell on to the railway line when a masonry wall collapsed just outside Liverpool Lime Street station, which had the potential to crush or derail a passing train, with disastrous consequences. Therefore, there is an urgent need to better assess the in-service performance of ageing masonry infrastructure stocks, and to provide detailed and accurate data that will better inform maintenance programmes and asset management decisions.

Assessing the structural performance of ageing masonry infrastructure is a complex task. Previous research has clearly demonstrated that the assessment methods currently used by the industry are antiquated and/or over-simplistic [1,2]. For example, for the assessment of masonry arch bridges, the Military Engineering Experimental Establishment (MEXE) method of assessment is still in use; It dates back to the 1940s, has very limited

* Corresponding author.

E-mail address: v.sarhosis@leeds.ac.uk (Vasilis Sarhosis).

predictive capability, and offers little scope for future enhancement. Also, although the primary focus of past research has been into the prediction of structural failure of ageing masonry infrastructure, prediction of the service load above which incremental damage occurs is now a key priority for infrastructure owners, who are under increasing pressure to provide transport networks which are secure and resilient.

Over the last three decades, significant efforts have been devoted to the development of numerical models to represent the complex and non-linear behaviour of masonry structures subjected to external loads [3–6]. Such models range from considering masonry as a continuum (macro-models), to the more detailed ones that consider masonry as an assemblage of units and mortar joints (micro-models; [7–15]). Since ageing masonry infrastructure is typically characterized by low bond strength, cracking is often a result of the de-bonding of the masonry units from the mortar joints. Given the importance of the masonry unit-to-mortar interface on the structural behaviour of aged masonry structures, micro-modelling approaches (i.e., those based on Discrete Element Method) are better suited to simulating their serviceability and load carrying capacity. However, a vital aspect when modelling masonry structures based on the micro-modelling approach is the accuracy in which the geometry of the masonry structure is transferred in the numerical model. So far, the geometry of masonry infrastructure is captured with traditional techniques (e.g., visual inspection and manual surveying methods) which are labour intensive and error prone.

In the last ten years, advances in computer-vision, photogrammetry, and laser-scanning have started to drastically change the building industry. Especially since such techniques are able to capture rapidly and remotely digital records of objects and features in 2d-images [16–20] and point-cloud/3D-mesh formats [21–26]. Even with the use of artificial intelligence for both 2d and 3d environments [26–32]. Although some work has been done in transitioning from point cloud to structural analysis models those are limited to continuum macro-modelling [33–38], or discontinuum macro-modelling [39,40]. Thus, there are still many challenges to overcome, especially regarding the discretization of the numerical models generated. Additionally, a prominent factor in the assessment of masonry infrastructure is the impact of existing pathologies, such as deformations and cracks. According to Heyman [41], geometric changes and existing damage in masonry structures can greatly influence their rate of degradation and in-service mechanical response. The lack of convenient tools that enable image and point cloud data to be readily transformed for use in a structural analysis model has hindered uptake in the engineering community.

In this framework, Loverdos and Sarhosis proposed a workflow to exploit images directly and automatically from ageing masonry infrastructure to generate geometrical digital twins which can be used for the structural assessment, inspection, and documentation [42,43]. The procedure is as follows. Initially, images can be captured using DSLR or a smartphone or from a drone. Any image from any source can be used. Images that include any background (random objects, sky, ground, etc.) are also compatible. Orthorectified images (with equal height/width scale) and good resolution are preferred but are not necessary. Then, the image selected is imported to “image2DEM” software to detect and extract the masonry micro-geometry. The algorithm provides reliable detection using Artificial Intelligence. Convolutional Neural Networks (CNN) are used to identify the location of masonry units and cracks, with ~96% and ~80% accuracy, respectively [27,44,45]. Furthermore, background elements (non-masonry) are filtered out automatically. The geometry is then extracted in the form of simplified lines to improve efficiency and reduce computational effort and a “geometric digital twin” is created. Blocks,

mortar, and cracks are assigned to different layers automatically. The mesh is optionally generated for blocks, mortar, and cracks. The output is provided in DXF format for compatibility between different CAD and BIM environment programmes. Finally, the geometry extracted could be converted to numerical modelling software for the analysis of masonry structures. Furthermore, the mesh generated allows to investigate separation (loss of contact) during the analysis. To enhance simplicity, the elements are allocated to different groups depending on the layers assigned on the CAD file.

2. Software description

The main workflow of the software is simple (Fig. 1). Any image can be used to identify the micro-geometry of masonry (i.e., blocks, cracks, background), using artificial-intelligence and image-processing (Fig. 1; “Detection”). More specifically, blocks and cracks are detected using individual FCNs (Fully Convolutional Networks), while the background and other elements are detected using image-processing. Then, the geometry is extracted to a CAD file for documentation (Fig. 1; “Documentation”). Where each object is assigned to a separate layer automatically, based on the detection method used to generate the binary-image. Finally, the exported geometry is used to generate the numerical model of the structure (Fig. 1; “Analysis”), with a numerical-analysis software (such as UDEC, a discrete-element-method software). The numerical-model is used to evaluate the current state of the structure and estimate the maximum capacity.

The graphical interface was developed to allow the user to easily select the modules that wants to run and allow the modification of the options of the software (Fig. 2). The GUI includes the ability to run all modules together, selected number of modules, or a single module (as seen on the left side of Fig. 2). The options are adjusted as an imported text-file (seen on the right side of Fig. 2). This allows to include many variables on the software that adjust the final output.

Regarding the main-part of the software, there are multiple modules that each have a specific task. Those include functions to load and adjust the image (P1, P2). Detect the micro-geometry of masonry (P5, P7, P9, P11). Improvements to the binary-images using image-processing (P6, P8, P10, P12). Damage-evaluation in terms of their geometrical properties (P13). Feature-extraction of the micro-geometry (P14, P15, P16, P17). And finally, model generation for CAD documentation or analysis (P18). A separate module is used to convert the CAD file to UDEC geometry (“DXF to UDEC”). Although, in most numerical analysis programmes, the DXF file can be used to create the numerical-model. More specifically, all the modules of the programme are described in the table below (Table 1):

Where **P1** and **P2** are relevant to the “Capture” step; **P3–P12** are part of the “Detection” step; **P13–P18** are part of the “Documentation” step; and finally, **P18** and **DXF-to-UDEC** are relevant to the “Analysis” step (see Fig. 1). Those are further explained in the chapters 4–7.

Furthermore, the adjustable-options, provided in the GUI (Fig. 2: right-side), can be used to improve the quality of the final-output. However, the default values are appropriate for almost every case and do not need any adjustment. The only exception are a few basic options, which modify the geometrical-model to meet different needs. Such as the inclusion of damage (cracks), mortar (for detailed micro-modelling), and background (if the masonry does not cover the whole image). For example, if the structure is large, detailed micro-modelling is avoided since the model will be very complex and may cause the analysis to fail. In which case, the user should turn off the definition of mortar (“PO_Use_Mortar=False”) for simplified micro-modelling.

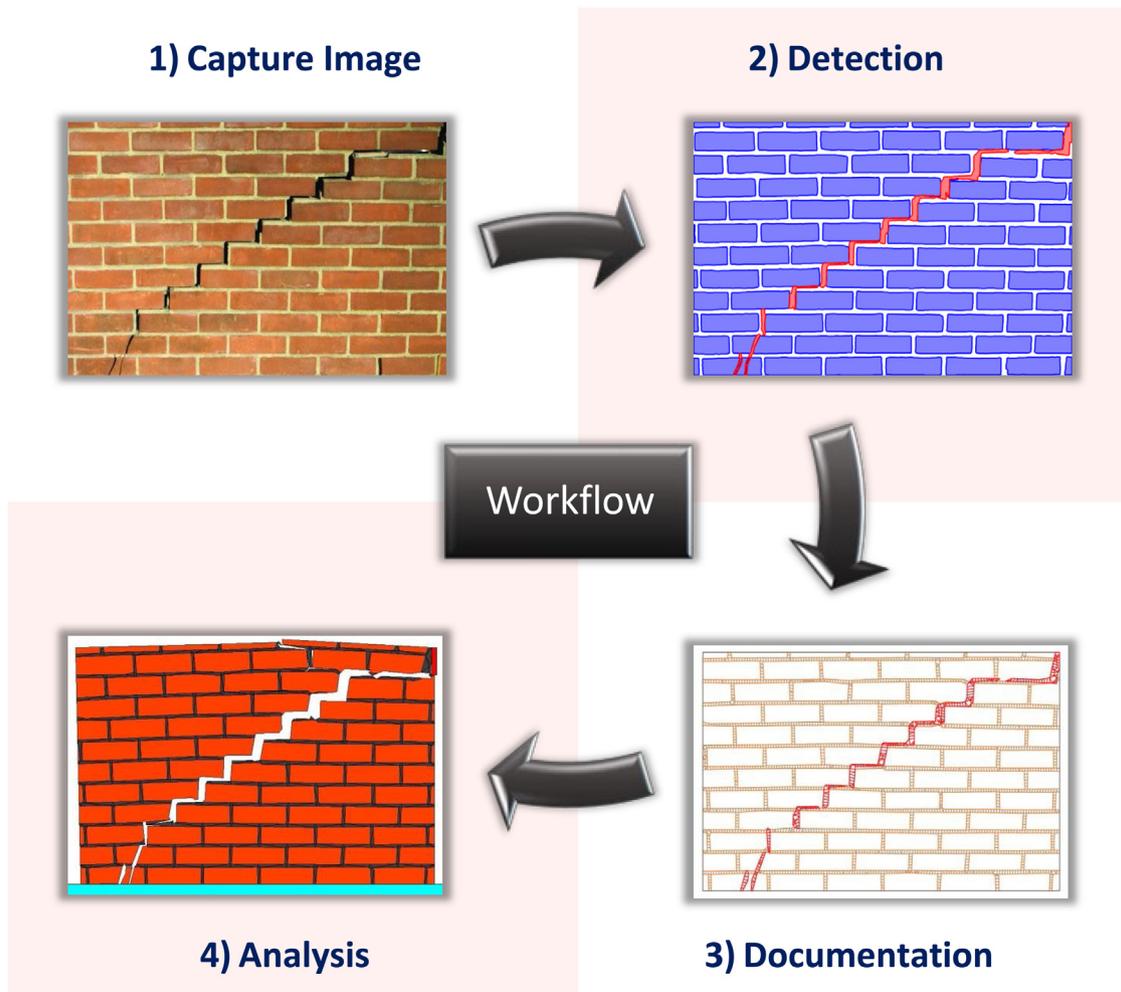


Fig. 1. Workflow of the algorithm.

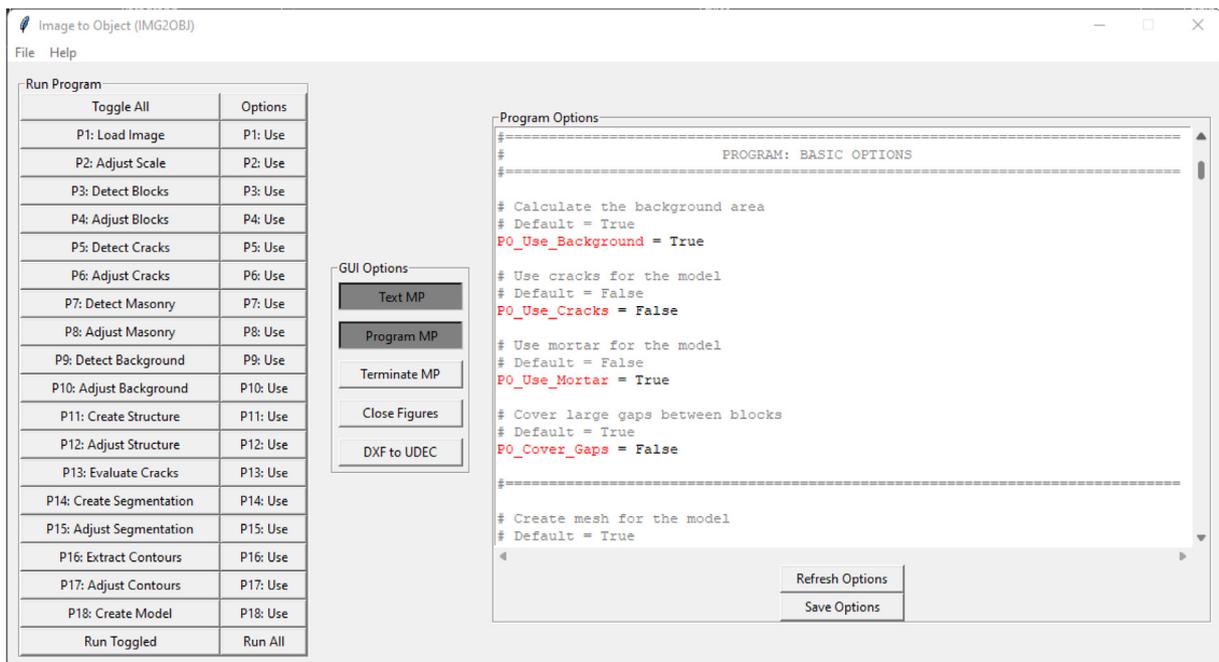


Fig. 2. Graphical Interface of the Software.

Table 1
Separated modules of the programme..

P1	Load Image: Used to select an image with a file-browser.
P2	Adjust Scale: Adjusts the scale of the image programmatically.
P3	Detect Blocks: Detects blocks on image using CNN.
P4	Adjust Blocks: Improves the blocks-mask using image-processing.
P5	Detect Cracks: Detect cracks on image using CNN.
P6	Adjust Cracks: Improves the cracks-mask using image-processing.
P7	Detect Masonry: Detects the overall location of masonry by merging the detected blocks.
P8	Adjust Masonry: Improves the masonry-mask using image-processing.
P9	Detect Background: Identifies the background, either by image-thresholding (for white background), or by inverting the masonry-mask (for undefined background).
P10	Adjust Background: Improves the background-mask using image-processing.
P11	Create Structure: Creates the overall structure by combining all binary-images. It can be used to identify undefined elements, not detected by the block-detection (such as concrete beams).
P12	Adjust Structure: Improves the structure-mask using image-processing.
P13	Evaluate Cracks: Creates a CSV file with the geometric-properties of each isolated-crack (i.e., location, area, length, average-width, and coverage).
P14	Create Segmentation: Applies watershed-segmentation to isolate detected blocks.
P15	Adjust Segmentation: Adjusts the watershed-segmentation to include mortar and damage. Also, applies corrections to the watershed, to ensure the proper geometry-extraction.
P16	Extract Contours: Extracts the micro-geometry of masonry as polylines (using the watershed).
P17	Adjust Contours: Applies line-generalization to the extracted geometry (reducing the number of vertices of the polyline) and filters-out small elements with near zero-area. Furthermore, it adjusts the geometry to improve the general shape of the structure. Additionally, it generates the mesh, optimized for numerical-analysis, to investigate crack-propagation.
P18	Create Model: Creates the DXF file of the geometry of the structure, with every material assigned to individual layer. Furthermore, it provides multiple CSV files with the inner-location of every detected-object, separated by material.
-	DXF-to-UDEC: Convert the AutoCAD file to “fish” commands for analysis using UDEC. Separates materials to different groups (classifications).

3. Step #1: Input image

Initially a representative image, of a masonry-structure, is captured and imported to the software using a browser-window typical to windows-applications (Fig. 2: P1). As it was mentioned earlier, any image can be used with the developed software. However, orthorectified images will provide higher accuracy in the “Documentation” and “Analysis” part of the software, due to equal-scale between the [x, y] axis and due to corrections to image-distortion caused by the camera lens (i.e., barrel effect). Orthorectified images can be produced using photogrammetry software, such as “Context Capture”.

Additionally, the scale of the image is an important aspect of the process for multiple reasons. Firstly, it allows to automatically adjust most variables and thus, minimize user interaction with the GUI. An example of the automated adjustment of a variable is the resize-value of the image, before passing through the CNN networks. But more importantly, it allows the acquisition of the true dimensions of the structure. Those are used for the geometrical-model generation and for the evaluation of the geometric-properties of detected cracks. The scale of the image is acquired programmatically for convenience (Fig. 2: P2). More specifically, the scale is acquired by selecting two points on the input-image and providing the distance between them (Fig. 3).

4. Step #2: Object detection

Blocks on the image are detected using CNN for reliable detection with a validation accuracy equal to 96.86% and validation F1-score of 96.3% (Fig. 2: P3; Fig. 4: a2). Similarly, the cracks are also detected using CNN with a validation F1-Score equal to 79.6% (Fig. 2: P5; Fig. 5: b). Both models were trained with images of typical masonry structures (i.e., not rubble), bonded with mortar.

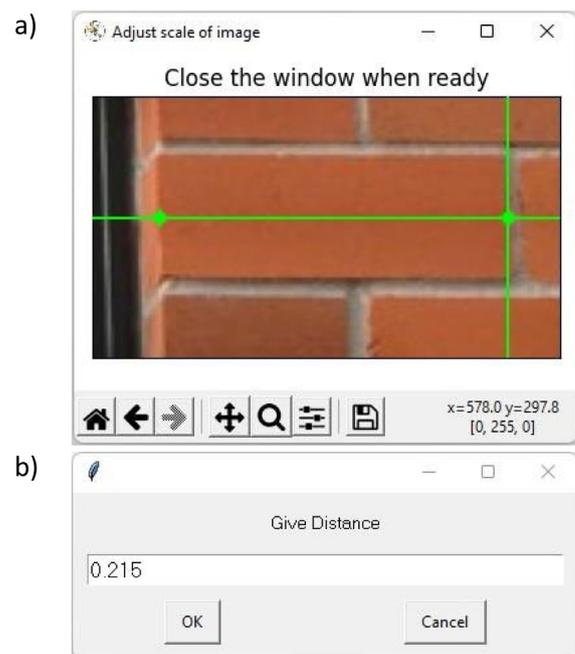


Fig. 3. Image scaling (in meters).

The overall location of masonry is detected automatically from the detected blocks (Fig. 2: P7), using simple image-processing functions (i.e., image-closing to combine the detected-blocks). The background is detected based on user-preference (Fig. 2: P9) and can be acquired either from the masonry-mask (Fig. 4: a3), or the white section of the image (Fig. 4: b3).

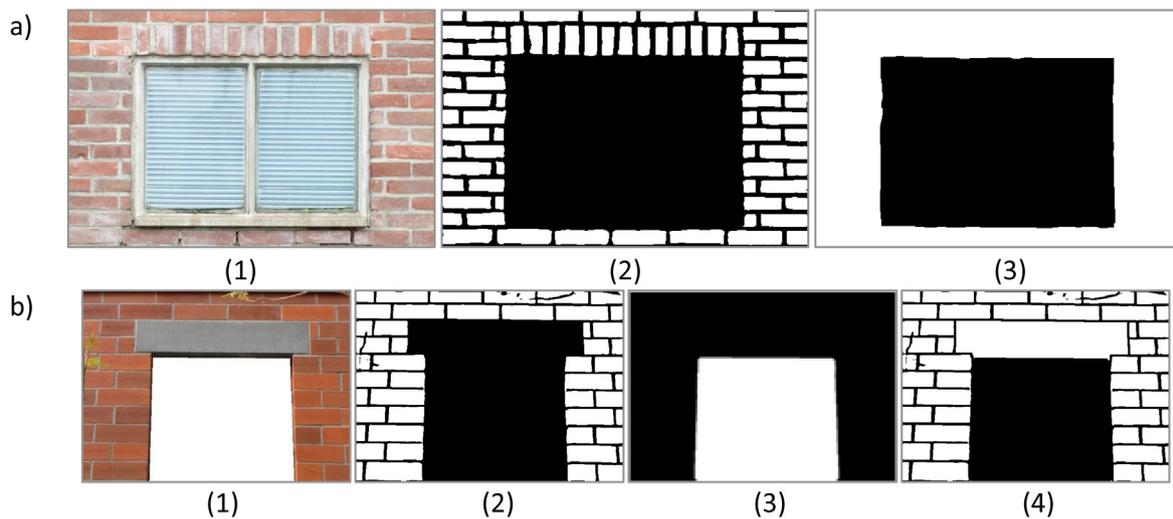


Fig. 4. [a] Detect Background: (1) Original Image; (2) Blocks; (3) Masonry; [b] Detect Other Elements: (1) Original Image; (2) Blocks; (3) Background; (4) Final Structure.

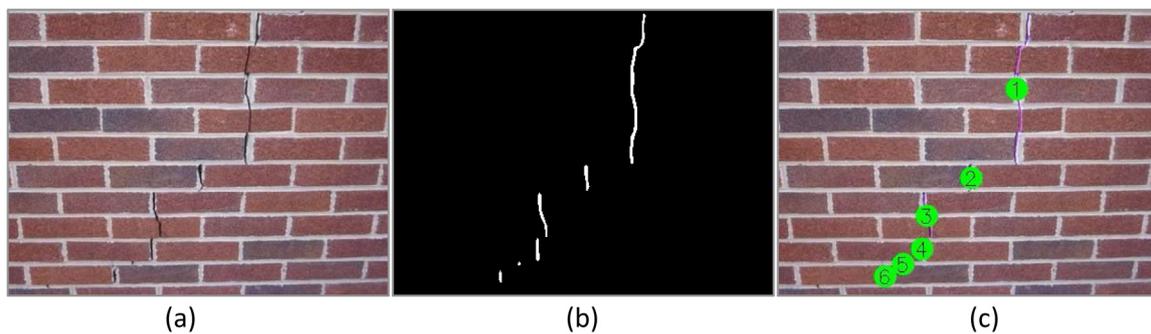


Fig. 5. Crack measurements; (a) Image; (b) Detected cracks; (c) Overlay and labels of cracks.

The structure combines all the binary-images (Fig. 2: P11; blocks, cracks, masonry, background) and has the capability to identify undetected-elements automatically (Fig. 4: b4). More specifically, the areas that does not belong to either background or masonry are assigned as undetected-elements. However, the detection of other structural-elements can only be applied for images with white-background, no-background at all, or a custom-background mask. This is due to the reason that the masonry-mask must be different from the background to identify undetected-elements.

The adjustment of the binary masks is using the same module with different arguments (Fig. 2: P4, P6, P8, P10, P12). They mostly remove small elements from either the background and/or foreground, based on the object-area (converting the number of pixels to scaled-area). Thus, ignoring the extraction of excessively small-elements that are possibly labelled-incorrectly. Those adjustments improve the output considerably. They are applied after each mask-detection to avoid repetition of the detection (Fig. 2: P3, P5, P7, P9, P11). Especially for blocks and cracks since their detection is slower due to the application of a CNN model (Fig. 2: P3, P5).

5. Step #3: Documentation - geometrical model

The detected cracks are used to acquire the geometric properties of each detected-defect (Fig. 2: P13). Each crack is isolated and measured individually using image-processing (Fig. 5). The calculation of the crack metrics is precise, assuming the accu-

Table 2

Geometrical-properties of detected-cracks (the location starts from the top-left side of the image).

Label (No)	Location [xmid,ymid] (pixels)	Area Scaled (mm ²)	Length Scaled (mm)	Width Scaled (mm)	Coverage Cracks (%)	Coverage Masonry (%)
1	[251, 82]	2516	376	7	59.507	0.423
2	[203, 176]	432	52	8	10.211	0.073
3	[156, 216]	700	107	7	16.549	0.118
4	[151, 251]	372	48	8	8.803	0.063
5	[131, 267]	45	3	17	1.056	0.008
6	[112, 280]	164	25	7	3.873	0.028
Total:		4229	607	53	100	0.71

rate output of the crack-detection module. Those metrics can be used to assist engineers with the visual inspection of masonry structures.

The acquired crack-metrics are the location, area, width, length, and coverage (Table 2). The coverage refers to the percentage coverage of the crack-area over the area of all defects or the overall masonry-area. The CSV file includes both scaled (i.e., in milometers) and unscaled (in pixels) values. However, only the scaled values are provided here due to size-limitation of the document width.

The detected geometry of masonry from the binary images is used to generate the geometrical model (Fig. 2: P14-18; Fig. 6). Initially, watershed-segmentation is used to isolate every individual block (Fig. 2: P14). The segmentation is then adjusted to

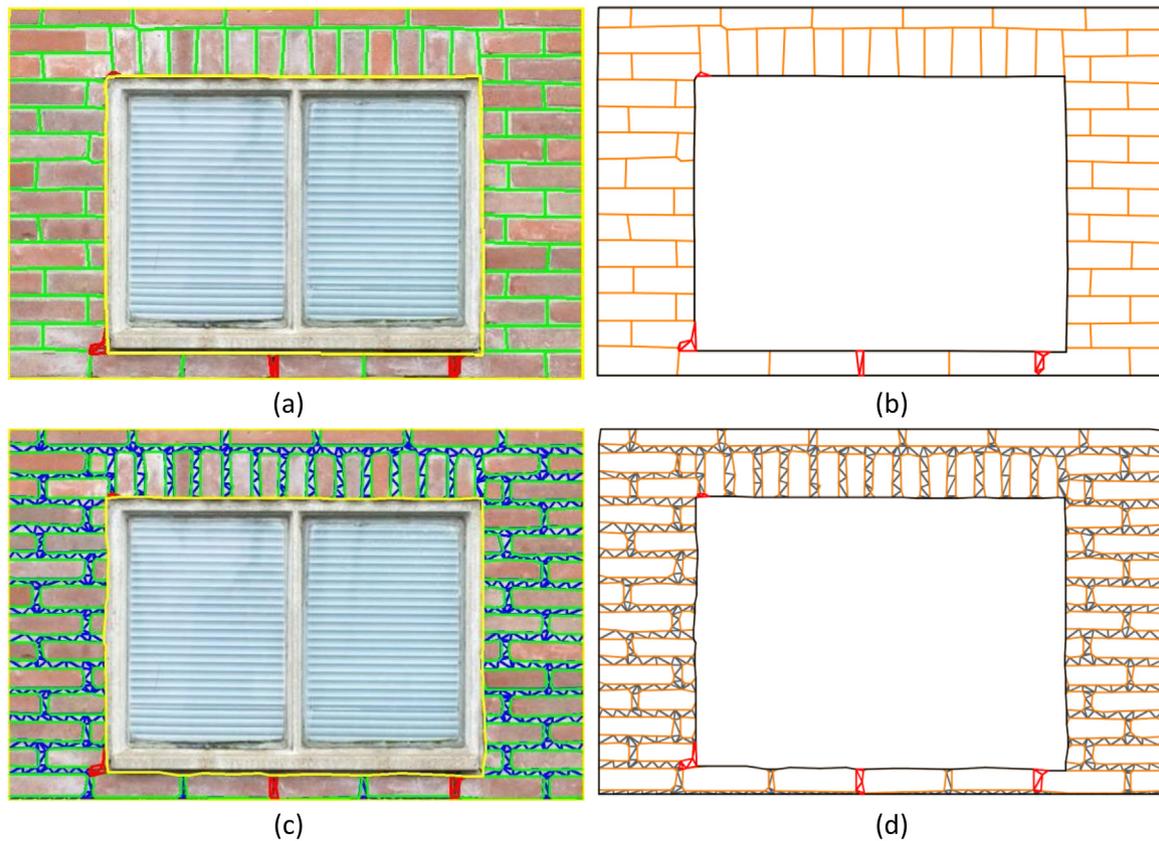


Fig. 6. Geometrical model; (a) Generalized-lines for simplified micro-modelling; (b) AutoCAD drawing (simplified micro-modelling); (c) Generalized-lines for detailed micro-modelling; (d) AutoCAD drawing (detailed micro-modelling).

include mortar and damage, but also to test that every separated-segmentation is assigned a unique-label (Fig. 2: P15). The geometry is then extracted in the form of polylines and is scaled to the real-dimensions for the precise documentation of the structure (Fig. 2: P16). The geometry is also simplified to reduce the number of vertices, so that it retains the minimum number of vertices that best describe each object (Fig. 2: P17), using a generalization algorithm (Fig. 6: a & b). The accuracy of the generalization is adjustable. The mesh is also generated to allow separation between objects and investigate crack-propagation, during the numerical analysis (Fig. 2: P18; Fig. 6: c & d). Finally, the generated geometry is exported to DXF. Both simplified micro-modelling (Fig. 6: b) and detailed micro-modelling (Fig. 6: d) are supported for the model-generation.

6. Step #4: Analysis – numerical model

The final step is the generation of the numerical model of the structure in 2D (Fig. 2: DXF to UDEC; Fig. 8: b; Fig. 10: b). For that the geometry, location, and classification of every object is required to define the model in UDEC. The geometry is defined by the DXF file (Fig. 8: a; Fig. 10: a). The location is defined for every object separately and is considering an inner point within the area enclosed by its individual element. The location is also used to define the class of every object in the numerical model. This is required because in UDEC every individual element is generated by dividing an existing large-block into multiple parts. Thus, the initial classification (of the main block) is irrelevant.

The classification of every object in the numerical-model is made using the inner-location of every object, except for the background where the segmentation-inner location is used instead. The reason why the segmentation-inner-location is used

for the background is because the polyline of the outer-background includes all the other objects as well (blocks, cracks, mortar, mesh-elements, etc.). Thus, using the segmentation location avoids the incorrect classification of an unspecified-object enclosed within the outer-background area.

The geometry extracted is limited to 2D analysis in general. A simple 3D model can be generated under the assumption that every drawn-object has equal depth. Although, automatic generation of 3D models would require a lot of manual-effort to adjust the model for discrete numerical-analysis. Mostly because the inner-materials, block pattern, etc, cannot be detected from image and point-cloud data (i.e., backfill, multi-leaf walls). However, regarding the use of the software for simple documentation, multiple faces can be extracted separately.

7. Output-files and folder-structure

The output folder-structure is divided into multiple-sections. The input-data used during the programme-execution are stored in the “Basic” folder. The user-adjustable options are stored in text-format in the “Data” folder. The “Images” folder includes figures of all the processes, aiming to identify issues with any part of the workflow. Additionally, the folder-structure includes an “Override” folder, which can be used to copy and manually-adjust the binary-images to improve the final-result. More specifically, any image with the keywords “Blocks”, “Cracks”, “Masonry”, “Structure”, will replace the original-input during the programme-execution.

The “Results” folder contains the final-output from the crack-measurements and geometry-extraction (Fig. 2: P13 & P18). Those include the DXF file of the geometry, the CSV file of the crack-measurements, and CSV files of the location of each element



(a)



(b)

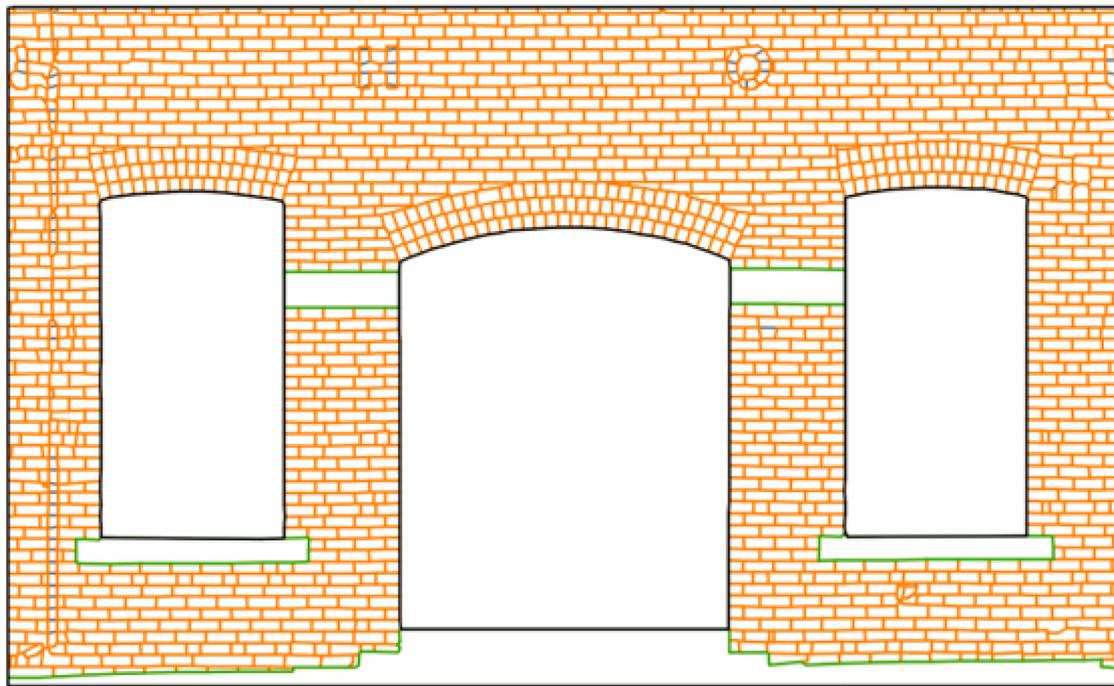
Fig. 7. (a) Sample of x173 images of the “Town House”; (b) Orthorectified image of the Town House, Leeds, UK [43].

(separated by material and detection method). If the companion-programme is used (Fig. 2: “DXF to UDEC”), the “Results” folder will include the TXT files of the complete-geometry in fish-commands (scripting-language of Itasca-software).

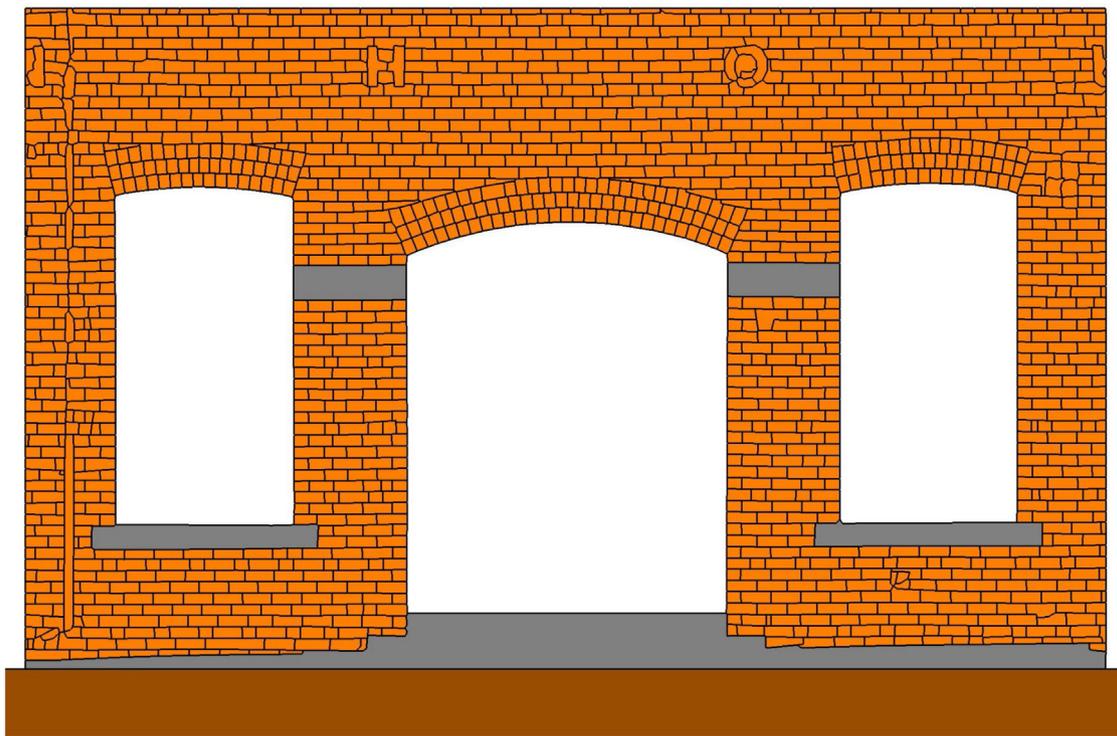
The location of every object is extracted individually per class, in a csv file format. The location-classifications are the following: blocks, block-mesh, mortar, mortar-mesh, damage, damage-mesh, and background. The location-types extracted are: Inner (inner location of the closed-polyline), centre (centroid of the closed-polyline), segmentation (inner location of the watershed-segmentation), and pixel (pixel location in scaled coordinates; for mortar/damage only).

8. Illustrative example #1

The following example is a section of the façade of a brick-work masonry building in the UK (Fig. 7: a). The input is an orthorectified image generated using photogrammetry software (Fig. 7: b). The specific case was selected because it contained different bonding patterns e.g., both arch and regular brick-pattern. The software produced an excellent drawing (Fig. 8: a), except for locations covered with foreground objects. More noticeable near the pipe on the left side of the image. The pipe could be filtered out, within the photogrammetry software, if more images were acquired from the side of the object. In which case, the



(a)



(b)

Fig. 8. (a) Output of software – AutoCAD Drawing: Grey-lines were drawn manually, over problematic areas, to ensure in-plane separation on the horizontal-axis during the numerical-analysis; (b) Numerical model in UDEC (created using the supplementary programme “DXF To UDEC”).

bricks behind the pipe could be reconstructed fully. However, minor corrections can be applied to the drawing directly (Fig. 8: a (grey lines)); or alternatively, by manually-editing the binary-output of the block-detection algorithm (Fig. 2: P4), which is then placed in the “Override” folder to be used instead of the initial-output. Finally, the 2D geometry was then transferred to a structural analysis software e.g., UDEC (Fig. 8: b), to allow the

numerical evaluation of the structure using the discrete element method. The complete procedure took ~ 915 s (for modules P1–P18). The largest amount of time was required by P15 (387 s). The time recorded is for an image-resolution of 4239×2594 pixels; Image-Scale of ~ 0.002 meters/pixel; and 4714 polylines that formed 1577 individual blocks. The computer used is a laptop with i7-9750H CPU, RTX-2060 GPU with 6gb VRAM, and



(a)



(b)

Fig. 9. (a) Sample of $\times 1,217$ images of the “Arch Bridge” used to generate the 3D mesh; (b) Orthorectified image of masonry arch-bridge (laboratory experiment in UoL).

16×2 gb RAM. The computational-time can be largely optimized by allowing multi-core calculations on developed-algorithms used by the software. Currently, most developed-algorithms are only allowed single-core calculations.

9. Illustrative example #2

The following example is a masonry arch-bridge experiment that is being conducted at the laboratory of the University of Leeds (Fig. 9: a). The orthorectified image (Fig. 9: b) was acquired using photogrammetry software, after it produced a proper 3D model (reality-mesh). It can be observed that the stair (presented in the original images; Fig. 9: a) was filtered-out from the 3D mesh (Fig. 9: b), which allows the detection of covered-bricks. Then the developed software was used to generate the AutoCAD drawing (Fig. 10: a) and subsequently, the numerical model in UDEC (Fig. 10: b). The block-detection demonstrates excellent results, especially considering the stained surface of

the bricks (more noticeable in the middle-right side). The numerical model considers the discrete element method, and more specifically simplified micro-modelling, to investigate crack propagation in later stages of the experiment. The complete procedure took ~ 391 s (for modules P1–P18). The time recorded is for an image-resolution of 3840×1899 pixels; Image-Scale of ~ 0.002 meters/pixel; and 2290 polylines that formed 766 individual blocks.

10. Impact

The impact of “Image2DEM” toolkit is to improve the structural analysis of our “as is” existing masonry infrastructure. “Image2DEM” allows a non-expert user to generate the geometry and mesh required for the development of high-fidelity numerical models such as the ones with DEM and FEM starting from images of the structure under consideration. The toolkit considers apart from the segmentation of masonry units, the damages in the structures such as cracks and distortions originated due to

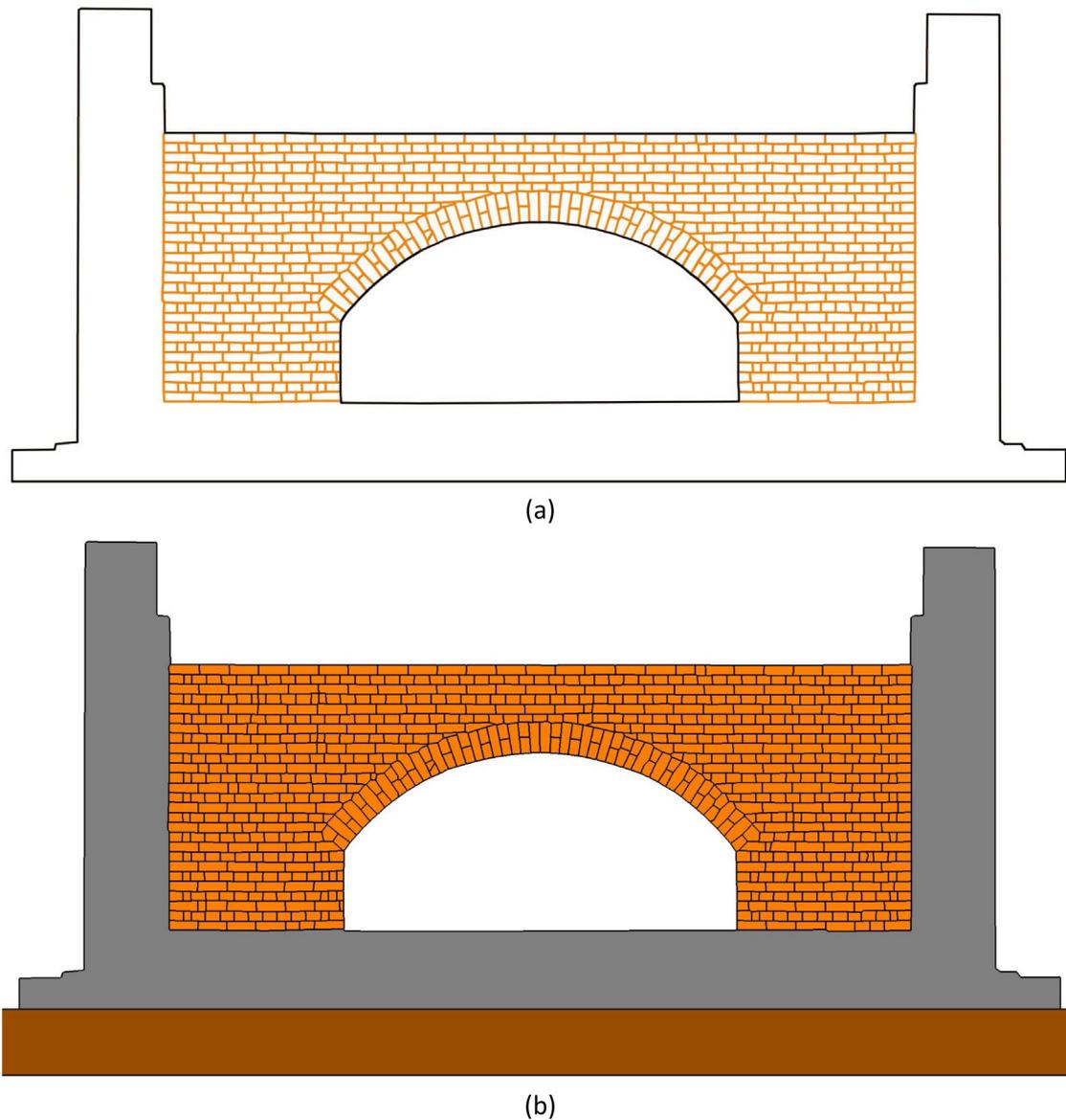


Fig. 10. (a) Output of software – AutoCAD Drawing; (b) Numerical model in UDEC (created using the supplementary programme “DXF To UDEC”).

ground subsidence as well as irregularities or missing bricks. Reliable inspection of infrastructure leads to more informed maintenance schemes, and potentially reduced unnecessary repair and strengthening interventions, which contributes significantly towards the UK’s “Net Zero” strategy [46].

11. Conclusions

The “Image2DEM” software is able to harness current developments in remote surveying methods and couple them with algorithms developed in Python based on Artificial Intelligence and Machine Learning to fully automate the “scan to structural modelling” procedure for the efficient and accurate and detailed structural analysis of our ageing masonry infrastructure stock. According to the method, first, images captured from smartphones or DSLR cameras are uploaded into our “Image2DEM” software. Using computer vision and Artificial Intelligence (AI) techniques, it is possible to detect masonry units (e.g., bricks, blocks) and cracks automatically. The “as is” geometry of the masonry structure generated, can then be extracted in the form of simplified lines (x, y coordinates) in a DXF format. Finally,

DXF files can be used in numerical analysis software for their structural assessment. The ad-hoc graphical tools developed are able to segment individual bricks in a masonry structure and mesh the mortar between them for the structural analysis. This transition from the physical to the digital environment has the potential to gain a better understanding of the “as is” condition of our existing masonry infrastructure and revolutionize the way structural analysis is performed in industry. Using efficient and accurate estimation of the “as is” structural condition of ageing masonry infrastructure, we are able to provide detailed and accurate data that will better inform maintenance programmes and asset management decisions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgements

The project has received partially funding from the EPSRC (EP/T001348/1). "Exploiting the resilience of masonry arch bridge infrastructure: A 3D multi-level modelling framework". The financial support is highly acknowledged.

References

- [1] McKibbins L, Melbourne C, S N, Gaillard C. Masonry arch bridges: condition appraisal and remedial treatment (c656). London: CIRIA; 2006, [Online]. Available: <https://www.ciria.org/ItemDetail?iProductC> [Accessed 1 June 2021].
- [2] Phares BM, Washer GA, Rolander DD, Graybeal BA, Moore M. Routine highway bridge inspection condition documentation accuracy and reliability. *J Bridge Eng* 2004;9(4):403–13. [http://dx.doi.org/10.1061/\(asce\)1084-0702\(2004\)9:4\(403\)](http://dx.doi.org/10.1061/(asce)1084-0702(2004)9:4(403)).
- [3] Lourenço PB. Computational strategies for masonry structures. Delft University Press; 1996, [Online]. Available: <http://www.narcis.nl/publication/RecordID/oa:i:tudelft.nl:uuid:4f5a2c6c-d5b7-4043-9d06-8c0b7b9f1f6f> [Accessed 27 January 2021].
- [4] Lourenço PB. Computational strategies for masonry structures: multi-scale modeling, dynamics, engineering applications and other challenges. In: *Congresso de Métodos Numéricos En Ingeniería*. 2013, p. 451–72, [Online]. Available: <http://repositorium.sdum.uminho.pt/handle/1822/26547>.
- [5] D'Altri AM, Sarhosis V, Milani G, Rots J, Cattari S, Lagomarsino S, Sacco E, Tralli A, Castellazzi G, de Miranda S. Modeling strategies for the computational analysis of unreinforced masonry structures: Review and classification. *Arch Comput Methods Eng* 2020;27(4):1153–85. <http://dx.doi.org/10.1007/s11831-019-09351-x>.
- [6] Asteris PG, Plevris V, Sarhosis V, Papaloizou L, Mohebkah A, Komodromos P. Numerical modeling of historic masonry structures. In: DeMarco B, Austin, Wolfe, Kayla, Henning, Christina, Carbaugh, editors. *Handbook of research on seismic assessment and rehabilitation of historic structures*, no. october. United States of America: Engineering Science Reference (an imprint of IGI Global; 2015, p. 213–56. <http://dx.doi.org/10.4018/978-1-4666-8286-3.ch007>.
- [7] D'Altri AM, de Miranda S, Castellazzi G, Sarhosis V. A 3D detailed micro-model for the in-plane and out-of-plane numerical analysis of masonry panels. *Comput Struct* 2018;206:18–30. <http://dx.doi.org/10.1016/j.compstruc.2018.06.007>.
- [8] Sarhosis V, Lemos JV. A detailed micro-modelling approach for the structural analysis of masonry assemblages. *Comput Struct* 2018;206:66–81. <http://dx.doi.org/10.1016/j.compstruc.2018.06.003>.
- [9] Sarhosis V, Sheng Y. Identification of material parameters for low bond strength masonry. *Eng Struct* 2014;60:100–10. <http://dx.doi.org/10.1016/j.engstruct.2013.12.013>.
- [10] Sarhosis V, Forgács T, Lemos JV. A discrete approach for modelling backfill material in masonry arch bridges. *Comput Struct* 2019;224:106108. <http://dx.doi.org/10.1016/j.compstruc.2019.106108>.
- [11] Segura J, Pelà L, Saloustrós S, Roca P. Experimental and numerical insights on the diagonal compression test for the shear characterisation of masonry. *Constr Build Mater* 2021;287:122964. <http://dx.doi.org/10.1016/j.conbuildmat.2021.122964>.
- [12] Forgács T, Sarhosis V, Bagi K. Influence of construction method on the load bearing capacity of skew masonry arches. *Eng Struct* 2018;168(March):612–27. <http://dx.doi.org/10.1016/j.engstruct.2018.05.005>.
- [13] Erdogmus E, Pulatsu B, Can B, Ozkan K. Analysis of the last standing arch of the roman aqueduct at blaundos. In: *13th North American masonry conference*, no. June. 2019, p. 483–93, [Online]. Available: https://www.researchgate.net/publication/334001391_Analysis_of_the_Last_Standing_Arch_of_the_Roman_Aqueduct_at_Blaundos [Accessed 27 January 2021].
- [14] Sarhosis V, Garrity SW, Sheng Y. Influence of brick–mortar interface on the mechanical behaviour of low bond strength masonry brickwork lintels. *Eng Struct* 2015;88(1):1–11. <http://dx.doi.org/10.1016/j.engstruct.2014.12.014>.
- [15] Sarhosis V, Oliveira DV, Lemos JV, Lourenço PB. The effect of skew angle on the mechanical behaviour of masonry arches. *Mech Res Commun* 2014;61:53–9. <http://dx.doi.org/10.1016/j.mechrescom.2014.07.008>.
- [16] Sithole G. Detection of bricks in a masonry wall. In: *International archives of the photogrammetry, remote sensing and spatial information sciences*, vol. XXXVII (B5). 2008, p. 567–72, [Online]. Available: https://www.isprs.org/proceedings/XXXVII/congress/5_pdf/99.pdf [Accessed 27 January 2021].
- [17] Oses N, Dornaika F, Moujahid A. Image-based delineation and classification of built heritage masonry. *Remote Sens* 2014;6(3):1863–89. <http://dx.doi.org/10.3390/rs6031863>.
- [18] Cluni F, Costarelli D, Minotti AM, Vinti G. Enhancement of thermographic images as tool for structural analysis in earthquake engineering. *NDT E Int* 2015;70(2015):60–72. <http://dx.doi.org/10.1016/j.ndteint.2014.10.001>.
- [19] Brackenbury D, Dejong M. Mapping mortar joints in image textured 3D models to enable automatic damage detection of masonry arch bridges. In: *17th international conference on computing in civil and building engineering*, vol. 173. 2018, p. 530–45, [Online]. Available: <http://programme.exordo.com/icccb2018/delegates/presentation/344/> [Accessed: 27 January 2021].
- [20] Bal İE, Dais D, Smyrou E, Sarhosis V. Novel invisible markers for monitoring cracks on masonry structures. *Constr Build Mater* 2021;300:124013. <http://dx.doi.org/10.1016/j.conbuildmat.2021.124013>.
- [21] Volk R, Stengel J, Schultmann F. Building information modeling (BIM) for existing buildings – Literature review and future needs. *Autom Constr* 2014;38:109–27. <http://dx.doi.org/10.1016/j.autcon.2013.10.023>.
- [22] Andriasyan M, Moyano J, Nieto-Julián JE, Antón D. From point cloud data to building information modelling: An automatic parametric workflow for heritage. *Remote Sens* 2020;12(7):1094. <http://dx.doi.org/10.3390/rs12071094>.
- [23] Bassier M, Vergauwen M. Unsupervised reconstruction of building information modeling wall objects from point cloud data. In: *Automation in construction*, vol. 120. Elsevier B.V.; 2020, 103338. <http://dx.doi.org/10.1016/j.autcon.2020.103338>.
- [24] Barazzetti L, Banfi F, Brumana R, Gusmeroli G, Previtali M, Schiantarelli G. Cloud-to-BIM-to-FEM: Structural simulation with accurate historic BIM from laser scans. *Simul Model Pract Theory* 2015;57:71–87. <http://dx.doi.org/10.1016/j.simpat.2015.06.004>.
- [25] Valero E, Bosché F, Forster A. Automatic segmentation of 3D point clouds of rubble masonry walls, and its application to building surveying, repair and maintenance. *Autom Constr* 2018;96(2017):29–39. <http://dx.doi.org/10.1016/j.autcon.2018.08.018>.
- [26] Valero E, Forster A, Bosché F, Hyslop E, Wilson L, Turmel A. Automated defect detection and classification in ashlar masonry walls using machine learning. *Autom Constr* 2019;106(May):102846. <http://dx.doi.org/10.1016/j.autcon.2019.102846>.
- [27] Dais D, Bal İE, Smyrou E, Sarhosis V. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Autom Constr* 2021;125:103606. <http://dx.doi.org/10.1016/j.autcon.2021.103606>.
- [28] Brackenbury D, Brilakis I, Dejong M. Automated defect detection for masonry arch bridges. In: *International conference on smart infrastructure and construction 2019, ICSIC 2019: driving data-informed decision-making*. (1); 2019, p. 3–10. <http://dx.doi.org/10.1680/icsic.64669.003>.
- [29] Kalfaris R, Wu ZY, Soh K. Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization. *J Comput Civ Eng* 2020;34(3):04020010. [http://dx.doi.org/10.1061/\(asce\)cp.1943-5487.0000890](http://dx.doi.org/10.1061/(asce)cp.1943-5487.0000890).
- [30] Ibrahim Y, Nagy B, Benedek C. Cnn-based watershed marker extraction for brick segmentation in masonry walls. In: *Image analysis and recognition. ICIAR 2019. Lecture notes in computer science*. 2019, p. 11662. http://dx.doi.org/10.1007/978-3-030-27202-9_30.
- [31] Ergün Hatir M, Ince İ. Lithology mapping of stone heritage via state-of-the-art computer vision. *J Build Eng* 2020;34. <http://dx.doi.org/10.1016/j.jobbe.2020.101921>, 101921 Contents, 2021.
- [32] Spencer BF, Hoskere V, Narazaki Y. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering* 2019;5(2):199–222. <http://dx.doi.org/10.1016/j.eng.2018.11.030>.
- [33] Rolin R, Antaluca E, Batoz JL, Lamarque F, Lejeune M. From point cloud data to structural analysis through a geometrical hBIM-oriented model. *J Comput Cultural Heritage* 2019;12(2):1–26. <http://dx.doi.org/10.1145/3242901>.
- [34] Kassotakis N, Sarhosis V. Employing non-contact sensing techniques for improving efficiency and automation in numerical modelling of existing masonry structures: A critical literature review. *Structures* 2021;32:1777–97. <http://dx.doi.org/10.1016/j.istruc.2021.03.111>.
- [35] Bassier M, Hardy G, Bejarano-Urrego L, Droukas A, Verstrynge E, Van Balen K, Vergauwen M. Semi-automated creation of accurate FE meshes of heritage masonry walls from point cloud data. *RILEM Bookseries* 2019;18:305–14. http://dx.doi.org/10.1007/978-3-319-99441-3_32.
- [36] Korumaz M, Betti M, Conti A, Tucci G, Bartoli G, Bonora V, Korumaz AG, Fiorini L. An integrated Terrestrial Laser Scanner (TLS), Deviation Analysis (DA) and Finite Element (FE) approach for health assessment of historical structures. A minaret case study. *Eng Struct* 2017;153(October):224–38. <http://dx.doi.org/10.1016/j.engstruct.2017.10.026>.
- [37] Funari MF, Hajjat AE, Masciotta MG, Oliveira DV, Lourenço PB. A parametric scan-to-FEM framework for the digital twin generation of historic masonry structures. *Sustainability (Switzerland)* 2021;13(19):11088. <http://dx.doi.org/10.3390/su131911088>.

- [38] Pepi C, Cavalagli N, Gusella V, Giofrè M. An integrated approach for the numerical modeling of severely damaged historic structures: Application to a masonry bridge. *Adv Eng Softw* 2020;151:102935. <http://dx.doi.org/10.1016/j.advengsoft.2020.102935>.
- [39] Hinks T, Carr H, Truong-Hong L, Laefer DF. Point cloud data conversion into solid models via point-based voxelization. *J Surv Eng* 2013;139(2):72–83. [http://dx.doi.org/10.1061/\(asce\)su.1943-5428.0000097](http://dx.doi.org/10.1061/(asce)su.1943-5428.0000097).
- [40] Tiberti S, Milani G. 3D voxel homogenized limit analysis of single-leaf non-periodic masonry. *Comput Struct* 2020;229:106186. <http://dx.doi.org/10.1016/j.compstruc.2019.106186>.
- [41] Heyman J. The stone skeleton. *Int J Solids Struct* 1966;2(2):265–79. [http://dx.doi.org/10.1016/0020-7683\(66\)90018-7](http://dx.doi.org/10.1016/0020-7683(66)90018-7), 249–256, IN1–IN4, 257–264, IN5–IN12.
- [42] Loverdos D, Sarhosis V, Adamopoulos E, Drougkas A. An innovative image processing-based framework for the numerical modelling of cracked masonry structures. *Autom Constr* 2021;125:103633. <http://dx.doi.org/10.1016/j.autcon.2021.103633>.
- [43] Loverdos D, Sarhosis V. Geometrical digital twins of masonry structures for documentation and structural assessment using machine learning. *Eng Struct* 2023;275(PA):115256. <http://dx.doi.org/10.1016/j.engstruct.2022.115256>.
- [44] Loverdos D, Sarhosis V. Automation in documentation of ageing masonry infrastructure through image-based techniques and machine learning. In: *Lecture notes in civil engineering*. LNCE, vol. 270, 2023, p. 727–35. http://dx.doi.org/10.1007/978-3-031-07322-9_73.
- [45] Loverdos D, Sarhosis V. Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. *Autom Constr* 2022;140:104389. <http://dx.doi.org/10.1016/j.autcon.2022.104389>.
- [46] BEIS. Net zero strategy: build back greener, no. october. 2021. 2021, [Online]. Available: <https://www.gov.uk/government/publications/net-zero-strategy>.