Automation in Construction 145 (2023) 104632

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

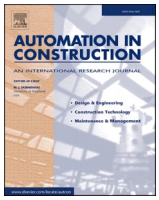

Image-based geometric digital twinning for stone masonry elements

B.G. Pantoja-Rosero [a], S. Saloustros [a], R. Achanta [b], K. Beyer [a,*]

[a] Earthquake Engineering and Structural Dynamics Laboratory (EESD), EPFL, 1015 Lausanne, Switzerland
[b] Swiss Data Science Center (SDSC), EPFL and ETH Zurich, 1015 Lausanne, Switzerland

ARTICLE INFO

Keywords:
Digital twins
Stone masonry
Computational modeling
Structure from motion
3D registration
Image analysis

ABSTRACT

We present an image-based pipeline for generating geometrical digital twins (GDTs) of stone masonry elements with detail down to the stone level. For this purpose, we acquire RGB images of the individual stones and of the wall during the construction phase. In our framework, we use structure from motion (SfM) to first generate 3D source and destination models, which are then registered to form the GDT through non-linear least squares and 2D point feature correspondences detected on the SfM images. This method contrasts with traditional techniques that register point clouds using 3D point descriptors. Because of the robustness of image feature descriptors, we found that using 2D instead of 3D point features facilitates the automation of the GDT generation. To benchmark our algorithm, we compared the results through an Euclidean–distance-based proposed metric with a known 3D textured model from which images were synthetically generated. We show the robustness and feasibility of our method for full size elements, wherein GDTs were generated for dry-stone and stone-mortar systems. This study allows researchers to produce accurate representations of the 3D geometry of walls built for experimental research, reducing therefore uncertainties related to the stone size, shape and arrangement to a minimum when comparing 3D numerical simulations of these walls to experimental results. Codes and data sets are publicly available (https://github.com/eesd-epfl/stone_masonry_GDT and https://doi.org/10.5281/zenodo.7266587).

1. Introduction

Many existing buildings, including those of significant cultural value, are made of stone masonry. This type of building is among the most vulnerable for various static and dynamic load cases, such as differential settlements, traffic loads, and earthquake loads [1–4]. To improve the ability of these buildings to withstand these loads, a good understanding of their response to the various load cases is necessary. The behaviour of stone masonry walls depends not only on the static and kinematic boundary conditions and on the material properties of the mortar and the stones but also on the size and shapes of the stones and on the arrangement of the stones in the wall [5–10]; in the following we refer to this stone layout as the microstructure of the wall.

For engineering practice applications, key engineering quantities of stone masonry walls (*i.e.* stiffness, strength, and deformation capacity) are estimated either from new tests (in situ or laboratory) or using empirical data from large-scale tests available in the literature [11–16]. On the other hand, some national codes (e.g. in Italy [17] and Switzerland [18]) adopt a classification for stone masonry according to their microstructures and propose ranges of values for strength and stiffness (Fig. 1). The new generation of Eurocodes will adopt the same approach. While this approach is suitable for today's engineering practice, in research, a more fundamental approach to treating the effect of microstructures is sometimes warranted. This applies, for example, to research that aims at developing detailed or simplified micro-models of stone masonry [19]. In these models the stones are represented by their actual shape and mortar joints are modelled explicitly, for example through solid elements, contact elements and/or sets of spring elements. With the advancement of numerical simulation tools and the increase in computational power, numerical simulations of stone masonry walls that represent stones and mortar explicitly become more and more feasible and several such simulation approaches exists or are currently developed (see for a 2D example Fig. 2).

When validating such numerical models against laboratory tests, it would be useful to have as input the exact geometry of the microstructure of the wall as it was tested (Fig. 3) rather than only the assignment to one of the classes (Fig. 1). Such a representation of the as-built 3D microstructure down to the level of the individual stone would reduce

* Corresponding author.
E-mail addresses: bryan.pantojarosero@epfl.ch (B.G. Pantoja-Rosero), savvas.saloustros@epfl.ch (S. Saloustros), radhakrishna.achanta@epfl.ch (R. Achanta), katrin.beyer@epfl.ch (K. Beyer).

https://doi.org/10.1016/j.autcon.2022.104632
Received 4 July 2022; Received in revised form 18 October 2022; Accepted 19 October 2022
Available online 3 November 2022

uncertainties resulting from the microstructure to a minimum. The goal of this paper is to present such a method for creating the geometric digital twins (GDTs) of the as-built microstructure of stone masonry walls.

While our work focuses on constructing GDTs for specimens built for laboratory testing purposes, i.e., for specimens for which we can also record the construction process, works that treat the reconstruction of microstructure from sensing techniques that can be applied after the construction of the wall was completed, already exist. If the microstructure of the masonry is regular, manually controlled computer-aided design can be used to define dimensions and shapes of the stones and assemble these to a microstructure. For irregular microstructures, however, new approaches are required. For 2D models, the microstructure can be derived from the outer faces of the walls. Work in recent years has aimed to enhance the accuracy of model geometry through the investigation of numerous non-contact sensing techniques. Kassotakis et al. [22] present an extensive literature review on state-of-the-art techniques for surveying masonry structures using laser scanning (LiDAR) and image collection techniques for obtaining accurate representations of the geometry of existent structures. This geometrical information can be used as an input for various modeling strategies of masonry structures, such as continuum [1,23–31] and block-based approaches [32–35]. While computer vision approaches have been developed in recent years to provide detailed geometry at the stone unit level of the external faces of masonry walls [36–38], it is still not possible to acquire the internal (not visible from the exterior) geometry of these structural elements. Instead, this internal morphology can only be estimated using non-destructive techniques, such as ground penetrating radar and sonic tomography [7,28,29,39–43]. However, these non-destructive techniques yield at present not sufficient information for reconstructing the internal microstructure in terms of a 3D geometrical model. 3D microstructures for single leaf masonry can be based on 2D geometries in combination with extrusion algorithms to generate 3D models [10]. For creating multiple-leaf masonry microstructures, our 3D microstructure generator can be used to generate artificial microstructures [44] resembling those of the stone masonry classes defined in Fig. 1. Such microstructures are suitable for numerical sensitivity studies on the effect of the microstructure on the mechanical response of the stone masonry wall. However, this 3D microstructure generator always generates artificial microstructures and not microstructures of as-built walls.

To complement existing and ongoing work on the geometric digital twinning of stone masonry walls, in this paper we make a contribution towards geometric digital twinning the microstructure of a stone masonry wall *during its construction process*. Such pipeline is useful for the geometric digital twinning of new stone masonry walls and we see immediate applications in research when comparing numerical simulations to experimental results. In a first study, the geometric digital twinning was based on laser scanning and a manual procedure for the reconstruction. This proved, however, as very time-consuming, relied on expensive laser-scanning equipment and the result was also dependent on the person performing the registration tasks. In this paper, we overcome these limitations by presenting a method for creating geometric digital twins (GDTs) of stone masonry walls down to the level of the individual stone of real stone masonry walls. We generate this GDTs from RGB images that are taken of the individual stones before the construction of the wall and from images taken during the construction of the wall.

Our research focuses on developing GDTs for laboratory-built stone masonry specimens so that highly accurate 3D geometries can be employed for numerical simulations via finite or discrete element analysis (FEA or DEA). The results of these mechanical simulations can be compared with those of the real model tested in the laboratory, resulting in a greater understanding of the behavior of existing structures and the creation of new lineaments for the construction of new structures. As stated previously, the current approach for numerical simulations is inaccurate due to the subjectivity present in the typology selection of the elements and the paucity of experimental data. This is the motivation for this study.

In this paper, we describe this method for geometrical digital twinning of stone masonry walls from RGB images and apply it to case studies. In Section 2, we state the problem, provide a general description of our methodology, and emphasize our research contribution. In Section 3, we describe the components of the methodology, including SfM, image feature detection, description and matching, and the nonlinear least squares method with its formulation to our registration problem. Additionally, we explain some extra considerations and capabilities of our codes, such as the use of the RANSAC algorithm to reduce the influence of outlier data and the use of extra views and features for registration. Following this, in Section 4, we validate our methodology on synthetic image data generated by a computer graphics software using a known 3D textured-mesh model. To show the effectiveness of our methodology for generating GDTs, we perform further experiments on real and typical dry stone and stone-mortar masonry elements. Finally, we present the conclusions of our work and the outlook to future developments.

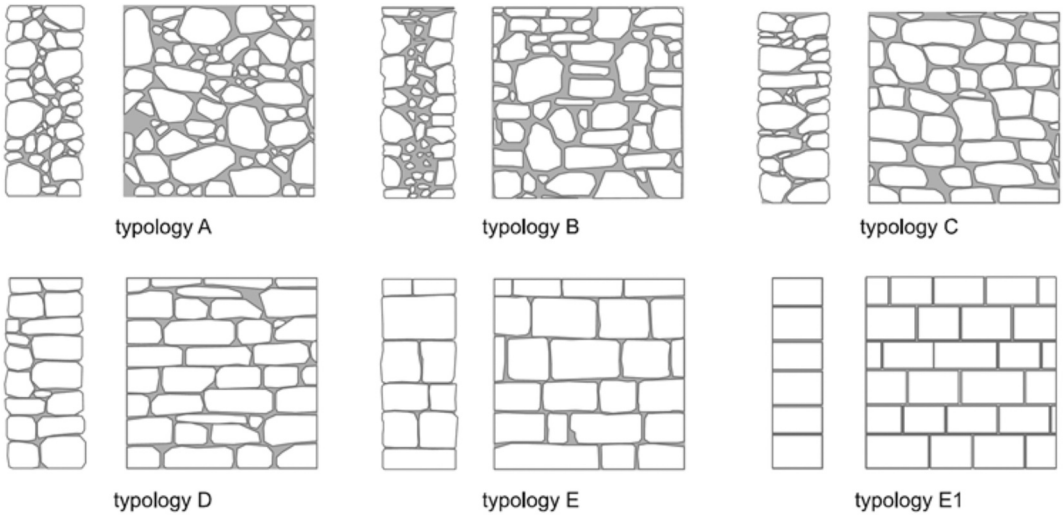

Fig. 1. Classification of stone masonry walls into six discrete classes according to [17] (figure from [16]).

2. Problem statement

The objective of this work is to develop a method for generating GDTs of new stone masonry elements from images taken during construction. A method for doing this can impact the design of new structures in two ways. First, due to the lack of methods for generating GDTs of newly constructed irregular stone masonry walls, it is currently impossible to directly compare between experimental and numerical results of stone masonry structures. Thus for walls built and tested in the laboratory, our method could validate the numerical models used to simulate the response of stone masonry structures. Second, this method can be the basis for the future automated construction of walls using either natural stones or demolition waste, making the development of a procedure for automated GDT generation for stone masonry elements an important step towards the digital age. Furthermore, the methodology proposed here both advances current research in the field of stone masonry structures as well as provides ready-to-use tools for the digitalization of the construction industry. The primary objective of the research presented here is the first aspect, i.e., the generation of highly detailed geometries that serve as input for numerical analyses applicable to laboratory-built stone masonry specimens. This will allow for a greater understanding of the behavior of these types of structures by comparing and validating the numerical modeling and experimental test results.

A digital twin (DT) is an accurate digital representation of a physical object that contains variety of information depending on the type and application [45]. Here, we focus on generating geometric digital twins (GDTs) [46,47] that contain detailed information of the geometry of real stone masonry structural elements. Recent breakthroughs in artificial intelligence, including computer vision and machine learning, make it possible to automate the generation of a GDT of a physical element. For example, a desired real object can be reconstructed from multiple view images together with well-established techniques such as structure-from-motion (SfM) and multiple-view stereopsis [48–50]. In general, these photogrammetry methods extract repeated features from images of the object, which are later used to recover the camera poses (rotation and camera center position) to project those features to the 3D space as point clouds [51,52]. Further post-processing of the images and point clouds, such as the generation of depth maps, can densify the point clouds, which can later be used to generate textured meshes.

In this context, we would like to automate the process of generating GDTs of stone masonry elements using images taken during their construction, with particular emphasis on models for laboratory-built components. In this work, we propose an image-based algorithm that uses SfM information (e.g., images, camera poses, structure) to register 3D models. In this algorithm, we generate 3D models for each individual stone, which we register to a 3D model of the stone masonry element through a non-linear least squares algorithm that uses 2D feature detection-description on images employed in the SfM pipeline. This leads to a GDT of an element that has, for the first time, a level of detail down to the single stones. This level of detail makes the models generated from our pipeline suitable for use in mechanical analysis methodologies, such as finite element analysis (FEA) and discrete element analysis (DEA).

3. Methodology

Our objective is to develop a method for the automated generation of a GDT for stone masonry components that contains detail down to the level of a single stone. For this, we combine computer vision and machine learning techniques, specifically SfM, 2D feature detection, and non-linear least squares, to generate 3D models of stones and walls that can be registered to generate the desired detailed GDT for stone masonry elements.

Fig. 4 represents the pipeline of our methodology. This pipeline starts by generating 3D SfM models as point clouds for each stone S_i, for each wall layer during construction L_j, and for the final built wall W (with integer values for $i \in [0, N]$ and $j \in [0, M]$, where N and M are the number of stones and layers in the wall model, respectively). Later, we use the 2D image features employed to generate each of those SfM models to match reference images used to generate a source model (*src*, this could be a single stone or a wall layer model) with those used to generate a destination model (*dst*, this could be a wall layer or the final wall model). This helps locate correspondences between the *src* and *dst* models in 3D (X_{src}, X_{dst}), which are used to find a transformation matrix ($T_{src-dst}$) that allows their registration. We find transformation matrices to register each stone model S_i to its correspondent wall layer model L_j (T_{SL}) and each wall layer model L_j to the final wall model W (T_{LW}). With these matrices, we can finally transform each stone model S_i to register it in the correct position on the final wall model W to generate the desired GDT as $T_{LW}T_{SL}S_i$. In the next subsections, we detail the important components of this pipeline.

3.1. Structure from motion

The core of our methodology is the SfM technique, as it can both generate 3D models as well as produce information about 2D image features, which facilitates the production of the detailed GDTs for stone masonry components. SfM is a 3D reconstruction technique that employs multiple-view geometry, which was defined by Hartley et al. [52] as the branch of computer vision that uses various image views of a scene to answer three questions: 1) How is the same point visualized and constrained across different views (hereafter referred as *correspondence geometry*)? 2) What is the 3D camera pose (i.e., location and rotation) associated with each view (hereafter referred as *motion*)? 3) What is the 3D position of a point observed in different views (hereafter referred as *structure)?*

In multiple-view geometry, each of the views is associated with a

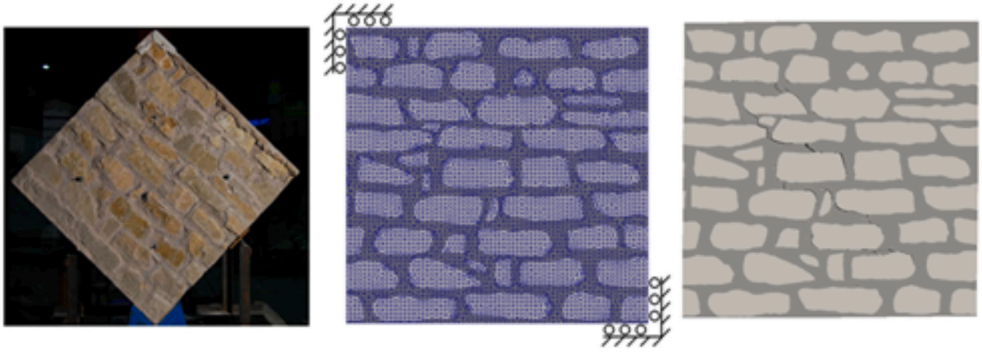

Fig. 2. 2D numerical simulation of a stone masonry panel subjected to diagonal compression loading where stone and mortar are represented explicitly [8] using the open-source software Akantu [20,21].

projection matrix **P** that contains extrinsic (i.e., camera pose) and intrinsic (e.g., focal length) parameters of the camera. This matrix can project a 3D point **X** to the corresponding image as $\mathbf{x} = \mathbf{PX}$. The same point can be projected to another image view using the camera matrix $\mathbf{P}'$ as $\mathbf{x}' = \mathbf{P}'\mathbf{X}'$. In these two views, **x** and $\mathbf{x}'$ are termed correspondences, as they are points in two view images that correspond to the same 3D point **X**. This is used in a classical pipeline for 3D reconstruction from multiple-view geometry, whereby the point correspondences are identified from images and used to generate both motion (i.e., camera poses) and structure (i.e., 3D point clouds) [51]. To generate 3D SfM models, various software have been developed over the last years, including Meshroom© [53], an open source photogrammetry library that we use in our pipeline. Fig. 4 presents examples of point clouds generated using Meshroom©.

3.2. 2D feature detection, description, and 3D point matching

As mentioned previously, multiple-view geometry solves for three interrelated aspects: *correspondence geometry*, *motion*, and *structure*[52]. The *structure* is formed in 3D space by triangulating the information related to the correspondences between views. This triangulation is performed with the *motion*. With this in mind, the starting point of multiple-view geometry is the identification of correspondences between images from two views. These correspondences are two points **x** and $\mathbf{x}'$ in the two image views that correspond to the same 3D point **X**. To find the correspondences in two views, it is necessary to detect features (keypoints) that would possibly form a correspondence. One common solution here is to use points in the images where the intensity gradient of the pixels is high [54], which can be done using various methodologies developed for this purpose based on differentiation, learning, or a gradient (e.g., [55–57]. To form correspondences, each keypoint is described using descriptor vectors, and descriptor vectors of keypoints are compared between two image views. When descriptor vectors meet certain criteria of similarity, it signals a correspondence between a pair of keypoints (e.g., [56,58–61]).

Each model of the stone S_i, of the layer L_j, and of the wall *W* generated with SfM contains information about 2D image feature points **x** (Figs. 5a and 6a) and their associated 3D point **X** (Figs. 5b and 6b). Our methodology uses those 2D features and their descriptions between the source *src* and the destination *dst* model to find 3D correspondences $(\mathbf{X}_{src}, \mathbf{X}_{dst})$. Figs. 5 and 6 illustrate how 3D correspondences can be found by directly employing the information provided by the SfM framework. In these two examples, the *src* (stone model S_i for Fig. 5 and layer model L_j for 6) and *dst* models (layer model L_j for Fig. 5 and wall model *W* for 6) are presented together with one of the image views used to generate the 3D point clouds. The red points $\mathbf{x}_{view}$ (Figs. 5a and 6a) correspond to the fraction of the 3D point cloud $\mathbf{X}_{view}$ (Figs. 5b and 6b) that was contributed by the image view during the reconstruction of the 3D model **X**. Then, for the *src* and *dst* models, we have 2D-3D correspondences $(\mathbf{x}_{src}, \mathbf{X}_{src})$ and $(\mathbf{x}_{dst}, \mathbf{X}_{dst})$, respectively. As the image related to the *src* model contains an object that also is present in the image related to the *dst* model, some of the *src* features $(\mathbf{x}_{src})$ should correspond to *dst* features $(\mathbf{x}_{dst})$. To find the $(\mathbf{x}_{src}, \mathbf{x}_{dst})$ correspondences, we followed the matching process of 2D features described by Lowe [57], which computes Euclidean distances between feature descriptors for $\mathbf{x}_{src}$ and $\mathbf{x}_{dst}$. Then, matched features (correspondences) are defined as those with the smallest value for this distance. According to Lowe [57], one feature point of the source model k (x_{ksrc}) corresponds with a feature point of the destination model m (x_{mdst}) when the distance between their descriptors is equal to the minimum distance and is less than 75% of the second smallest distance between the descriptors of point k and all point descriptors in the destination model. The colored points in Figs. 5c and 6c link the correspondences $(\mathbf{x}_{src}, \mathbf{x}_{dst})$ between the reference images of the *src* and *dst* models. As a result of this and our knowledge of the 2D-3D correspondences $(\mathbf{x}_{src}, \mathbf{X}_{src})$ and $(\mathbf{x}_{dst}, \mathbf{X}_{dst})$, matching the 2D features allows us to find 3D correspondences $(\mathbf{X}_{src}, \mathbf{X}_{dst})$ between models. These correspondences will facilitate the registration between the *src* and *dst* models through a matrix that transforms $\mathbf{X}_{src}$ into $\mathbf{X}_{dst}$ using the non-linear least squares algorithm. It is noted that outliers in the correspondences (matched points that do not actually represent the same point in the 3D model) can affect the results. To avoid this, we used the RANSAC algorithm, explained in the next sections, together with the non-linear least squares algorithm.

3.3. Non-linear least squares algorithm

The non-linear least squares algorithm fits data to a nonlinear model by optimizing a loss function [62]. To do this, *m* parameters $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_m]$ of a model $f(x;\boldsymbol{\beta})$ are found with the goal of minimizing the mean squared error (MSE) that represents the loss function $L(x;\boldsymbol{\beta})$ as:

$$L(\boldsymbol{\beta}) = \frac{1}{2}\sum_{1}^{N} r_i^2(\boldsymbol{\beta}), \tag{1}$$

(a) (b)

Fig. 3. Stone masonry walls: (a) Multi-leaf stone masonry wall built for laboratory testing and (b) corresponding test setup [15].

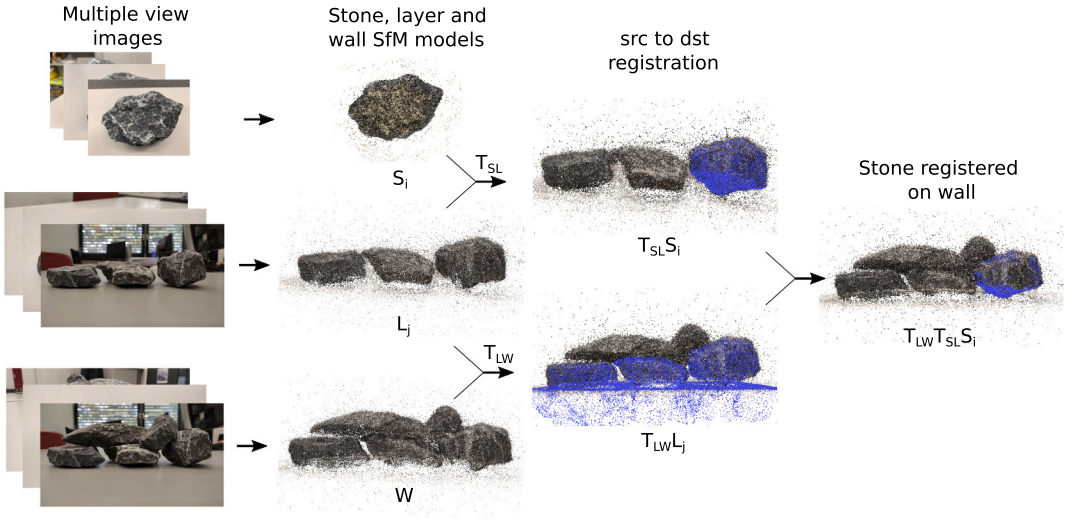

Fig. 4. Pipeline for generating geometrical digital twins (GDTs) of stone masonry elements. From left to right: acquiring images; generating SfM models of the stone, layers of stones, and the wall; registering *src* models to *dst* models; registering each stone at the final position in the wall.

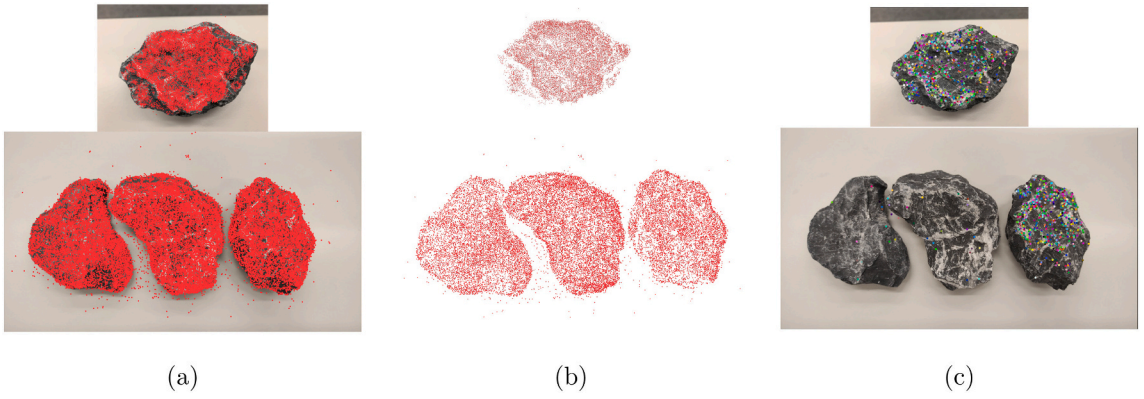

Fig. 5. Feature matching between the reference images for the stone and layer models. Top row: source (*src*) model. Bottom row: destination (*dst*) model. a) Keypoint features detected on the reference image that are part of the point cloud structure. b) Parts of the point cloud structure that were detected on the reference images. c) Matched keypoints between the *src* and *dst* models.

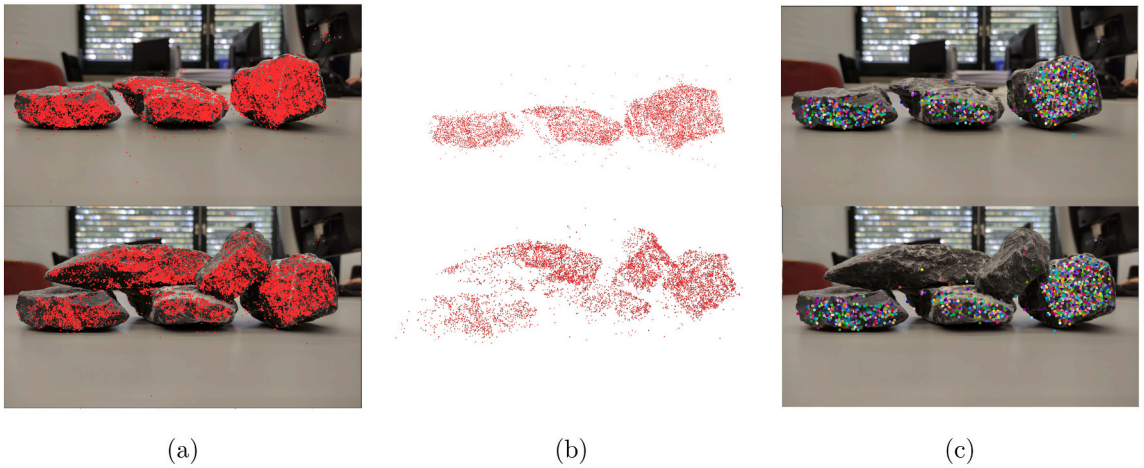

Fig. 6. Feature matching between the reference images for the stone and layer models. Top row: *src* model. Bottom row: *dst* model. a) Keypoint features detected on the reference image that are part of the point cloud structure. b) Parts of the point cloud structure that were detected on the reference images. c) Matched keypoints between the *src* and *dst* models.

where $r_i(\boldsymbol{\beta}) = f(x_i;\boldsymbol{\beta}) - y_i$ is defined as the residual function at N discrete points x_i in the domain, where y_i is the observed data that corresponds to x_i. Formally, the optimization is defined as:

$$\boldsymbol{\beta}_{opt} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} L(\boldsymbol{\beta}), \tag{2}$$

in which the objective is to find the set of optimal parameters $\boldsymbol{\beta}_{opt}$ that minimize $L(\boldsymbol{\beta})$. We solve Eq. (2) through the Gauss–Newton method in which the optimal set of parameters $\boldsymbol{\beta}_{opt}$ is found through changes in the $\boldsymbol{\beta}$ variables following the direction determined by the step $\Delta\boldsymbol{\beta}$, which is obtained iteratively as:

$$\boldsymbol{\beta}_{n+1} = \boldsymbol{\beta}_n - \gamma\Delta\boldsymbol{\beta}, \tag{3}$$

for the iteration $n+1$ with γ as the learning rate controlling the step size, and the step $\Delta\boldsymbol{\beta}$ is given by:

$$\Delta\boldsymbol{\beta} = \mathbf{H}[L(\boldsymbol{\beta})]^{-1}\nabla L(\boldsymbol{\beta}), \tag{4}$$

with $\mathbf{H}[L(\boldsymbol{\beta})]$ and $\nabla L(\boldsymbol{\beta})$ representing the Hessian and the gradient of the loss function, respectively. For a detailed description, refer to Betts et al. [63].

3.4. Solving for point cloud registration

Here we solve the 3D point-set registration problem by transforming one set of points ($\mathbf{X}_{src}$) from a source 3D model (*src*) to fit over another set of points ($\mathbf{X}_{dst}$) from a destination 3D model (*dst*). The sets of points $\mathbf{X}_{src}$ and $\mathbf{X}_{dst}$ are correspondences matched using 2D image features, as described previously. The *src* and *dst* models are represented, respectively, by the sets of points $\mathbf{X}_{src} = [X_0^{(src)}, \ldots, X_N^{(src)}]$ and $\mathbf{X}_{dst} = [X_0^{(dst)}, \ldots, X_n^{(dst)}]$ in $\mathbb{R}^3$ space.

Because the 3D reconstruction provided by SfM is up to scale (i.e., the dimensions of the models are proportional to the real dimensions unless metric information is given in post processing), the transformation necessary to register the *src* to the *dst* model consists of a similarity transformation. In this, the 3D model is assumed to have seven degrees-of-freedom (DOF), three associated translations in perpendicular directions $\mathbf{t} = (t_x, t_y, t_z)^T$, three rotations associated with Euler angles $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)^T$, and one related to scaling s. As the rotation can be represented by a 3x3 matrix that is a function of the rotation angles $\mathbf{R}(\boldsymbol{\theta})$, the objective is to find the parameters $\boldsymbol{\beta} = [\mathbf{t}, \boldsymbol{\theta}, s]$ to transform the *src* point cloud $\mathbf{X}_{src}$ into $\mathbf{X}'_{src}$, such that the Euclidean distance is minimized between the transformed point cloud $\mathbf{X}'_{src}$ and a *dst* point cloud $\mathbf{X}_{dst}$. $\mathbf{X}'_{src}$ in homogeneous coordinates is given by:

$$\mathbf{X}'_{src} = \mathbf{T}(\mathbf{R}, \mathbf{t}, s)\mathbf{X}_{src}, \tag{5}$$

where $\mathbf{T}(\mathbf{R}, \mathbf{t}, s)$ is the 4x4 similarity transformation matrix defined as

$$\mathbf{T}(\mathbf{R}, \mathbf{t}, s) = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{6}$$

It follows that the residual to be optimized is:

$$r(\boldsymbol{\beta}) = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}\mathbf{X}_{src} - \mathbf{X}_{dst}. \tag{7}$$

The transformation matrix $\mathbf{T}$ that determines the registration of the *src* to the *dst* model is found using the nonlinear least squares method. Fig. 7a shows the results after applying the transformation defined by the optimal parameters of the transformation matrix $\mathbf{T}$ to $\mathbf{X}_{src}$. Here, the *src* and *dst* models correspond to 3D point clouds of a single stone S_i and a wall layer L_j, respectively. The results of Fig. 7b can be obtained by applying the same algorithms while considering the *src* model as the wall layer L_j and the *dst* model as the final wall W. Then, combining the transformation matrices of the stone-layer ($\mathbf{T}_{S_iL_j}$) and layer-stone ($\mathbf{T}_{L_jW}$), a transformation matrix can be found to register the stone at its corresponding position in the wall as: $\mathbf{T}_{S_iW} = \mathbf{T}_{L_jW}\mathbf{T}_{S_iL_j}$. Fig. 7c shows the results of the registration of a stone model into the final wall layer.

3.5. RANSAC to avoid outlier influence

The 3D point correspondences ($\mathbf{X}_{src}, \mathbf{X}_{dst}$) can contain a number of outliers (mismatch between *src* and *dst* model features; check Figs. 5c and 6c) because the detection, description, and matching of 2D features are not deterministic procedures. To make the algorithm robust against possible outliers, we use random sample consensus (RANSAC) presented by Fischler et al. [64], which is a simple yet robust methodology that fits models to experimental data while avoiding the consideration of outliers. In essence, the method starts by finding an initial model with the minimum quantity of necessary data n (e.g., if the model is a 2D line, find a model for two points). This data is randomly selected. By establishing a threshold t for a defined error from this model, inliers and outliers can be detected (e.g., the distances from the rest of data to the proposed line). Performed over a previously defined number of iterations k, the generated model is scored based on the quantity of inliers d, and then another minimum data set is selected, a new model is proposed, and a new score is given. The largest score dictates which model is selected (here we score based on the mean error of the inliers). Overall, this procedure renders the methodology suitable for the analysis and interpretation of error-prone data, which is common in computer vision problems. For more details please refer to the original paper wherein Fischler et al. also show a numerical analysis evidencing the robustness of the method [64].

Here, RANSAC is used to find the transformation matrix $\mathbf{T}$ that allows the registration of a *src* into a *dst* model. The RANSAC hyper-parameters we used to obtain a satisfactory performance in our experiments were $n = 4, k = 300, t = 0.1l$ (l: mean of distances among points in $\mathbf{X}_{dst}$) and $d = 0.2w$ (w: number of 3D correspondences).

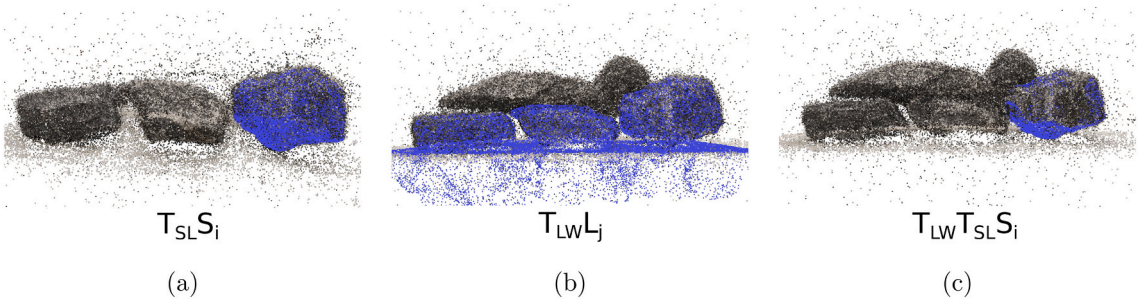

Fig. 7. Registration of *src* to *dst* 3D point cloud models. a) *src*: Stone S_i,*dst*: Layer L_j. b) *src*: Layer L_j,*dst*: Wall W. c) Registration of stone S_i on the wall W by combining the transformation matrices resulting from a) and b).

3.6. Automatic selection of reference images

Our methodology can automatically select the reference images for the *src* and *dst* models used to find the 2D features required for the registration process. To accomplish this, we detect, describe, and match point features between the batch of available images of the *src* and *dst* models. We consider the reference images to be the pair (one from *src* and one from *dst*) with the greatest number of matches. Although this automation works without difficulty, we have enabled an option wherein the user inputs these reference images for faster results, and we plan to speed up the automatic selection of the reference images in the future by implementing parallel programming.

3.7. Adding extra views

Our registration algorithm uses two reference images: one from the *src* model and one from the *dst* model. The features in those images, which were detected and used to generate the SfM model, were then re-used to find 3D point correspondences $(\mathbf{X}_{src}, \mathbf{X}_{dst})$. In case it is required (e.g., there are not enough keypoint correspondences between the initial reference images), we have enabled the use of extra images for the *src* model. As extra images, we selected the *m* (defined by user) extra images that contain the most matched features when compared to the reference image. For this, we again used information gathered from the SfM framework, this time referencing the image matching performed using 2D features. Similarly to the initial reference image, the features from those extra images that were used to generate the 3D point cloud were also used to find extra 3D point correspondences following the procedure explained in previous paragraphs.

3.8. Adding extra features

The input for our algorithm comes from the SfM pipeline. During the SfM computation it is possible to select from various types of point features to be detected and used during the reconstruction process. To increase the number of correspondences $(\mathbf{X}_{src}, \mathbf{X}_{dst})$, our method allows the use of extra keypoint features and descriptors in addition to those of the original SfM framework, specifically SIFT [57], AKAZE [60], ORB [65], FAST [66], and BRIEF [67]. The use of these additional features functions as the rest of the algorithm, wherein we detect, describe, and match new keypoints in the *src* and *dst* reference images to find 2D image correspondences $(\mathbf{x}_{src_{new}}, \mathbf{x}_{dst_{new}})$. Then, these new point correspondences are associated with the 2D features used to generate the 3D point cloud model by SfM to produce extra 3D correspondences $(\mathbf{X}_{src}, \mathbf{X}_{dst})$ as follows:

1. Find the pixel distances between new feature correspondences and the features used to generate the 3D point cloud in SfM ($||\mathbf{x}_{src} - \mathbf{x}_{src_{new}}||$ and $||\mathbf{x}_{dst} - \mathbf{x}_{dst_{new}}||$);
2. If the distance from the previous step is lower than one pixel, the 3D points associated with $\mathbf{x}_{src}$ and $\mathbf{x}_{dst}$ form a correspondence [i.e., $(\mathbf{X}_{src_{new}}, \mathbf{X}_{dst_{new}})$];
3. From $(\mathbf{X}_{src_{new}}, \mathbf{X}_{dst_{new}})$, select as extra correspondences $(\mathbf{X}_{src_{extra}}, \mathbf{X}_{dst_{extra}})$ those that have not been accounted for in the correspondences $(\mathbf{X}_{src}, \mathbf{X}_{dst})$.

These extra 3D correspondences can be concatenated to the initial $(\mathbf{X}_{src}, \mathbf{X}_{dst})$ and then used during the optimization process to find the transformation matrix $\mathbf{T}$ that allows the registration of *src* to *dst* point cloud models. The higher the number of 3D correspondences, the better the accuracy and robustness of our algorithm.

3.9. Geometric digital twins

The objective of our methodology is to create faithful 3D representations of stone masonry elements that reach a level of detail of single stones. To reach the registration results presented in Fig. 8b, the algorithm described in the previous steps is applied to each stone of the

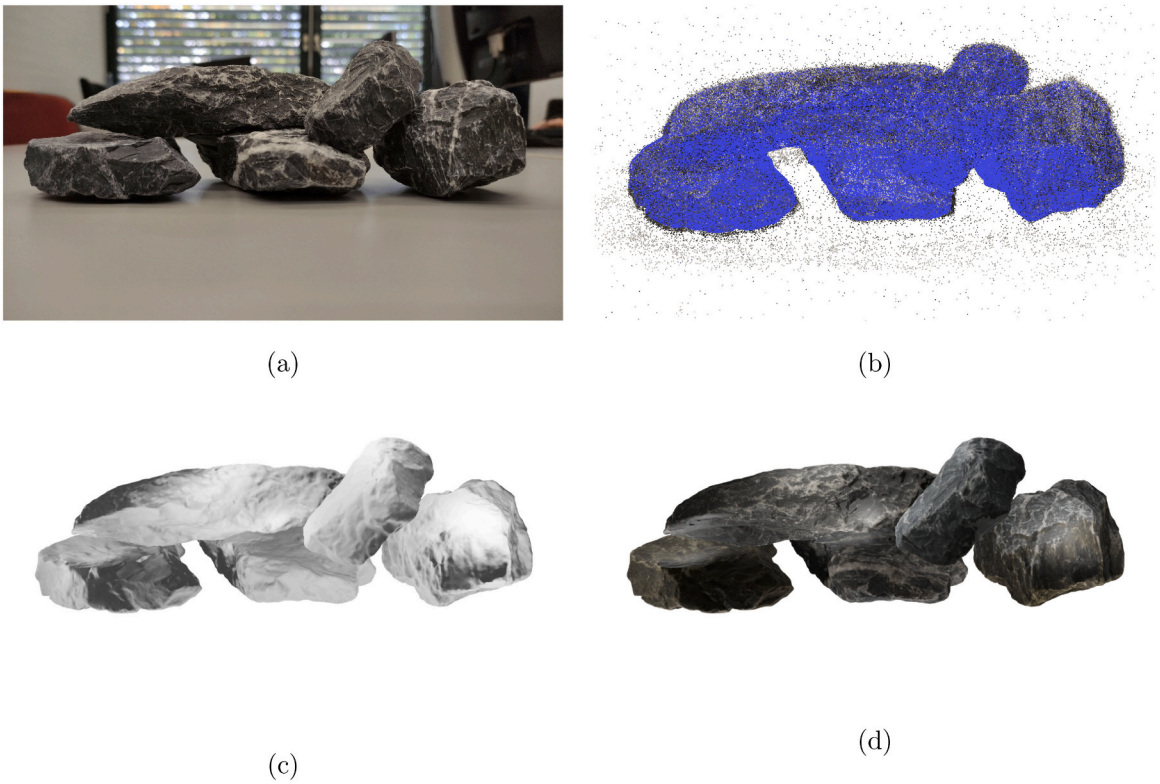

Fig. 8. Digital twinning methodology example: (a) Image view of wall model *W*. (b) Point clouds of stone models S_i registered on the wall model *W*. (c) Digital twin represented as mesh. (d) Digital twin represented as textured mesh.

masonry elements. Taking this even further, obtaining results such as those presented in the Fig. 8c and d requires that the SfM motion models be augmented to generate dense meshes, which can be done through multiple-view stereopsis. The transformations applied to the point cloud models for each stone component can be applied to meshes and textured meshes to reach this final image. The GDTs represented in Fig. 8c and d constitute our final result, and they are capable of generating the geometry for numerical analysis, such as finite element analysis (FEA) or discrete element analysis (DEA).

3.10. Image acquisition device

As it can be laborious to collect the necessary images to generate 3D stone models through SfM and as the input images affect the quality of reconstructions, we propose an imaging device to simplify and standardize image acquisition (Fig. 9). This device consists of a curved robotic arm that rotates around the target object to be reconstructed and takes photos from its five synchronized cameras hung at strategical locations. The camera positions and arm rotation during the image acquisition are selected to guarantee 60–80% overlap between consecutive images, as recommended by Snavely et al. [68]), which can increase the quality of reconstructions. With this device facilitating image collection for the single stone models as a first step, we plan to continue aiming for total automation in data acquisition by developing further tools, including for full stone masonry elements during construction.

3.11. Metrics

No metrics currently exist for evaluating automated GDT generation using photogrammetry. Therefore, we propose a metric to evaluate reconstruction performance by correlating a generated model with the 3D point cloud or mesh ground truths.

3.11.1. Distance of model fidelity (DMF)

Our metric is inspired by the inliers of model fidelity (IMF) metric proposed by Pantoja-Rosero et al. [69], which evaluates the reconstruction of level-of-detail (LOD) models using point clouds by measuring how well the polygonal surface model fits the point cloud. IMF measures the distance from a point (of the ground truth) to the nearest plane (of the assessed model), considering only those inliers that are closer than a defined threshold. The IMF metric score is considered to be the mean distance of a number of inliers.

Our modified version, hereafter called the distance of model fidelity (DMF), differentiates from IMF in two main ways. First, the model to be assessed is the representation of the GDT as a point cloud, and the ground truth can be represented as either mesh or point clouds. Second, instead of considering the quantity of inliers for the score, we compute the mean of the distances of the full model 3D points to the ground truth planes or points.

Thus, the DMF metric can be calculated from the set of points P that represent the GDT, where for each point $p \in P$, we compute its distance to the nearest plane π in the set of planes Π comprising the mesh ground truth,

$$d_p = \min_{\pi \in \Pi} (|\pi^{\mathsf{T}} p|_2)^2. \tag{8}$$

We then define our DMF metric as the average distance from the points to their closest plane,

$$DMF = \frac{1}{|P|} \sum_{p \in P} d_p. \tag{9}$$

Note that if the ground truth is given as a point cloud, d_p in Eq. (8) is the distance between the point $p \in P$ to the closest point of the ground truth model $p_{gt} \in P_{gt}$ as follows:

$$d_p = \min_{p_{gt} \in P_{gt}} \|p_{gt} - p\|_2. \tag{10}$$

In an ideal scenario, where the ground truth and digital twin model perfectly overlap, the DMF score would be zero. When using the score to benchmark methodologies, keep in mind that lower DMF values indicate better performance.

4. Experiments

In this section, we present three examples of the generation of a GDT using our methodology. The first example validates our algorithm using synthetic image data from a known 3D textured-mesh dry stone wall model. The second and third examples use image data from life-sized dry-stone and stone-mortar stacks, which comprise typical stone masonry elements. With these two examples, we demonstrate the viability of the proposed pipeline for generating GDTs of stone masonry walls at infrastructure-level scales.

4.1. Validation: synthetic dry-stone element

In this example, we validated our method for generating a GDT of a 3D textured model. This model is of a dry-stone stack composed of 27 stones in three layers. The synthetic image data was generated according to the indications of our methodology for simulating the construction process in models for stone S_i, layer L_j, and wall W. For this, we used the open-source computer graphics software tool-set Blender©. Images for the different models rendered in this software and used in our methodology are shown in the Fig. 10a. Using the optimal transformation matrix found by our method, Fig. 10b shows the registered point clouds

Fig. 9. Image acquisition device to collect data from objects to be registered.

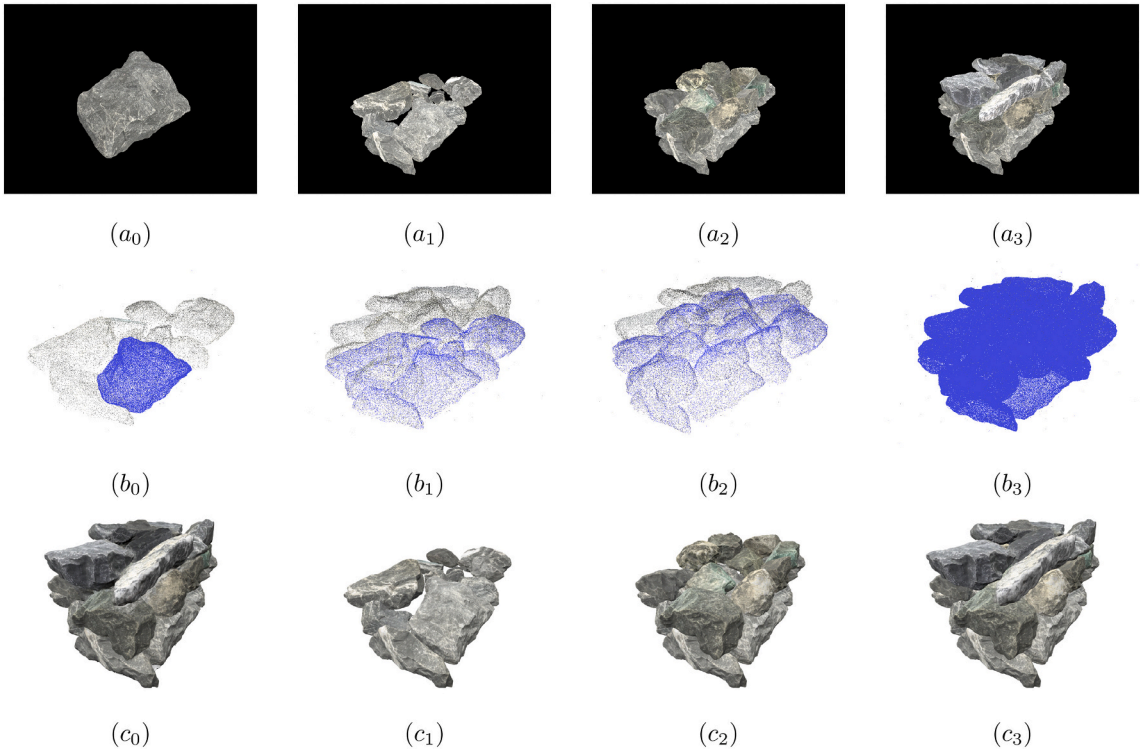

Fig. 10. Generation of a GDT using synthetic image data of a dry-stone element. (a) Examples of the generated synthetic images used as input in the SfM algorithm. (b) *src* SfM models registered on *dst* SfM models; (b_0) Stone model registered on respective layer model; (b_1 - b_2) Layer models registered on wall model; (b_3) All the stone models registered on the wall model. (c) GDT results; (c_0) *dst* textured wall model containing only external information; (c_1 - c_3) GDT containing both external and internal information of each stone component placed at their final position.

generated for how a stone is registered to its respective layer, the registration of the first and second layer to the wall model, and finally the registration of all the stones to the final wall. The results for the GDT model are presented in Fig. 10c. Fig. 10c_0 shows the *dst* textured mesh that contains just external information, and Fig. 10c_1–c_3 show the registered stone models as textured mesh at their corresponding position in the final model for the three layers of the masonry element.

Using our DMF metric, this model scored $DMF = 1.05mm$. This metric represents the mean offset between all of the GDT components and the ground truth. To evaluate the accuracy of this metric, consider that the approximate dimensions of the studied wall are $600mm \times 400mm \times 250mm$ and its bounding box diagonal dimension is $d_w = 763mm$. Similarly, the average of the largest dimension of the bounding box of the stones used in the model is $d_s = 20.5mm$. With these values in mind, our presented *DMF* score corresponds to 0.13% and 5.12% of the largest dimension of the wall and stones, respectively, indicating that our methodology is surprisingly accurate.

It is important to note that we first need to scale both the ground truth and GDT models to real measurements and then align them through a traditional RANSAC global registration and an iterative closest point (ICP) algorithm that is based on 3D point clouds [70]. This of course includes errors in the *DMF* score that are not related to our method, indicating that the method itself is likely even more accurate than reported through the metric.

4.2. Dry stone and mortar-stone stacks

Representing parts of typical stone masonry elements, we next constructed dry-stone and stone-mortar stacks layer by layer using 27 and 25 irregular stones, respectively. Figs. 11a and 12a present images taken from an individual stone and the wall construction process for the two stacks. Figs. 11b and 12b present the registered *src* on the *dst* point cloud models. Finally, Figs. 11c and 12c present the GDTs for the two cases. As the configuration of the elements produced herein constitute the common practices when building stone masonry structures, these results demonstrate the robustness of our methodology and the possibility of scaling this method to complete structures. Similar models were generated using a pipeline based on laser scanning and registration of point clouds with manual intervention. The results reported by the authors in their work indicated that the average time for a stone to be registered on a wall was five minutes. Our algorithm takes an average of 20s to register each stone to the final GDT, quantitatively demonstrating the increased efficiency of our method compared to the state of the art.

5. Conclusions

We present an image-based methodology to generate GDTs for stone masonry elements. Our research focuses on the generation of GDTs of laboratory-built stone masonry specimens. Using an exact geometric representation of the as-built microstructure of the wall permits a proper validation and comparison of numerical and experimental results, which can lead to an increased understanding of the behavior of existing stone masonry structures. For our approach, we assume that the construction process of these walls can be recorded by RGB images that are taken of the individual stones before the construction of the wall and from images taken during the construction of the wall.

To generate these 3D models, we rely on the information required for and produced by photogrammetry pipelines. Our method uses 2D features from reference images automatically selected from the SfM dataset to register 3D GDT models detailed to the level of single stones. The efficient manner in which this is performed substantially decreases the time required for manual user interaction in comparison to current techniques that register 3D models from points generated from laser scanning. In this way, we avoid the use of more complex and expensive devices such as LiDAR, enabling the use of simple imaging devices to generate a faithful GDT.

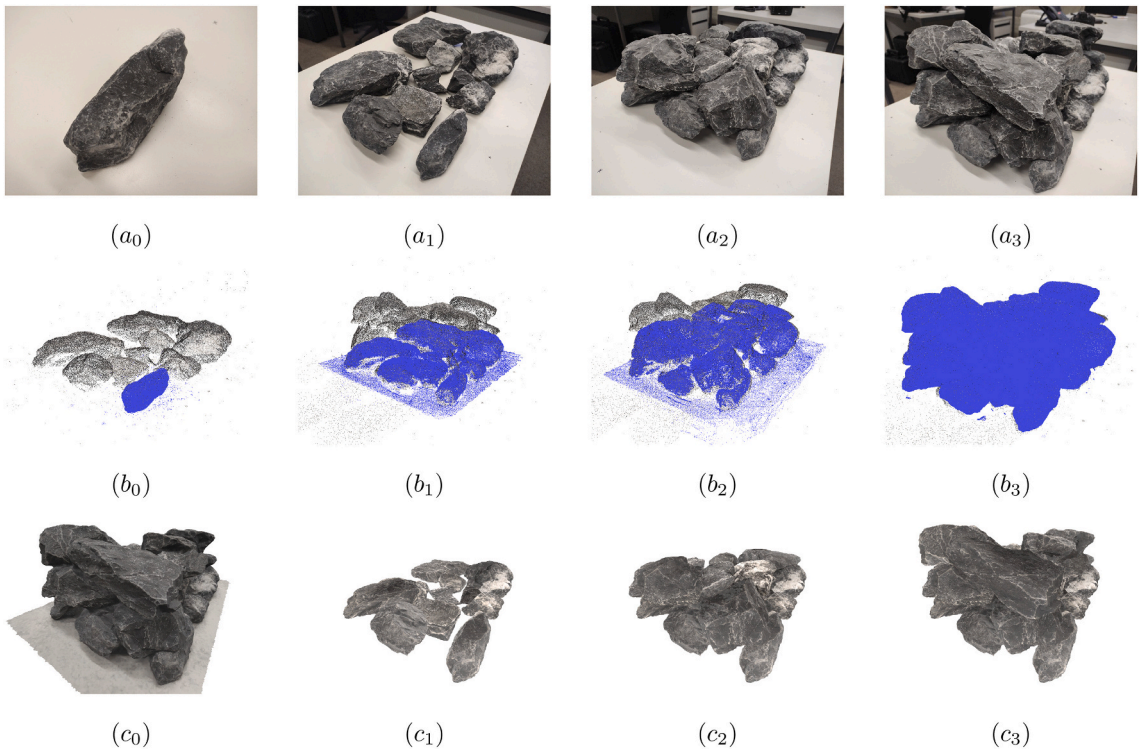

Fig. 11. GDT for a dry-stone element. (a) Images used as input in the SfM algorithm. (b) *src* SfM models registered on *dst* SfM models; (c_0) Stone model registered on respective layer model; (c_1 - c_2) Layer models registered on wall model; (c_3) All the stone models registered on the wall model. (c) GDT results; (c_0) *dst* textured wall model containing only outside information; (c_1 - c_3) GDT displaying internal information of each stone component placed at its final position.

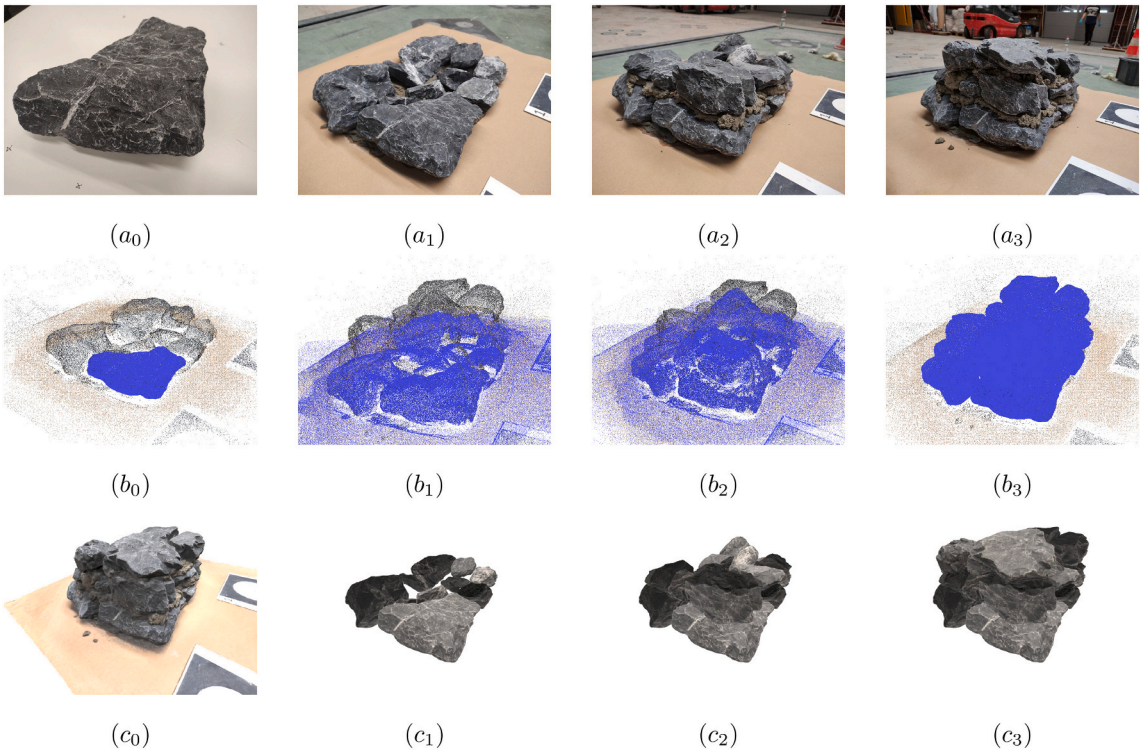

Fig. 12. GDT for a stone-mortar element. (a) Images used as input in the SfM algorithm. (b) *src* SfM models registered on *dst* SfM models; (c_0) Stone model registered on respective layer model; (c_1 - c_2) Layer models registered on wall model; (c_3) All the stone models registered on the wall model. (c) GDT results; (c_0) *dst* textured wall model containing only outside information; (c_1 - c_3) GDT displaying internal information of each stone component placed at its final position.

Although our methodology is presented for stone-masonry elements that undergo testing in experimental campaigns, properly acquired images during the construction process could be used to generate GDTs for entire new-built masonry structures, such as buildings or bridges. This might be computationally expensive using current technology and personal computers, but the continuous development of computational tools and processors will also remove this obstacle in the near future. Along the same lines, the use of the proposed pipeline in real engineering applications will first need to address the issue of efficient data compression and storage.

Even though our examples presented here consisted only of stone masonry materials, our algorithm can easily be adapted and used for structures composed of other materials, such as precast concrete and timber elements. Furthermore, we believe that the presented pipeline can be a valuable tool within a circular construction industry to improve the reuse of residual materials coming from demolition of structures. In particular, our algorithm can generate the geometry of elements built with demolition materials, enabling structural analysis and optimization of the microstructure of the new masonry.

The accuracy of GDT generation depends on the quality of the acquired images and the texture of the elements. The first point can be easily controlled following SfM lineaments (i.e., camera positioning and overlap/redundancy among images), using high quality imaging devices, and controlling light and background. For the second point, the number of detected features on the images drops if the elements are texture-less (intensity of pixels almost constant along the object) and might be not sufficient to generate 3D models and register the source to destination models. Contrary to stones, which have the appropriate texture to work well with our method, steel elements with uniform painting are an example of texture-less elements that might not produce satisfactory results with our method. Nevertheless, it would be still possibly to produce GDTs with the presented method by adding aleatory texture effects through painting before the image acquisition.

For future studies, we plan to enhance the presented pipeline across four main fronts. First, the image acquisition process should be automated to reduce the collection time. For this, we are designing protocols/manuals for users to follow during data collection and are designing imaging devices, such as the one presented in Section 3. The second front relates directly to the algorithm, which should be adapted to the protocols/manuals and devices proposed in the first front. In addition, further development in terms of automation needs to be considered, including parallel computing for the optimal selection of reference images (to perform the 3D correspondences matching based on 2D features) and the optimal selection and detection of models to be registered (i.e., automatically select which stone should be registered in which layer from batches of stone and wall-layer models). Accuracy is also an obvious aspect to improve, which we plan to do by implementing and testing various optimization algorithms in our methodology with the aim of improving the registration. The third front to be considered is the use of the generated models for mechanical analysis. Specifically, we would like to evaluate our algorithm to generate GDTs for laboratory specimens and then perform numerical analyses using our produced geometry to assess and validate experimental results. Finally, we consider that further research is needed for the use of our algorithm in real engineering applications, especially on efficient compression of the data used for the generation and the resultant data after employing our methodology as well as the use and storage of the GDT.

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

Codes and data sets are publicly available.

Acknowledgments

This project is partially funded by the Swiss Data Science Center under grant C18-04 (“Towards an automated post-earthquake damage assessment”). The second author would like to acknowledge funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 896761.

References

[1] S. Saloustros, L. Pelà, P. Roca, J. Portal, Numerical analysis of structural damage in the church of the Poblet monastery, Eng. Fail. Anal. 48 (2015) 41–61, https://doi.org/10.1016/j.engfailanal.2014.10.015. URL:http://linkinghub.elsevier.com/retrieve/pii/S135063071400315X.

[2] A. Drougkas, E. Verstrynge, P. Szekér, G. Heirman, L.E. Bejarano-Urrego, G. Giardina, K. Van Balen, Numerical Modeling of a Church Nave Wall Subjected to Differential Settlements: Soil-Structure Interaction, Time-Dependence and Sensitivity Analysis, Int. J. Archit. Herit. (2019), https://doi.org/10.1080/15583058.2019.1602682.

[3] M. Valente, G. Milani, E. Grande, A. Formisano, Historical masonry building aggregates: advanced numerical insight for an effective seismic assessment on two row housing compounds, Eng. Struct. 190 (February) (2019) 360–379, https://doi.org/10.1016/j.engstruct.2019.04.025.

[4] N. Grillanda, M. Valente, G. Milani, A. Chiozzi, A. Tralli, Advanced numerical strategies for seismic assessment of historical masonry aggregates, Eng. Struct. 212 (March) (2020), 110441, https://doi.org/10.1016/j.engstruct.2020.110441.

[5] A. Borri, M. Corradi, G. Castori, A. De Maria, A method for the analysis and classification of historic masonry, Bull. Earthq. Eng. 13 (9) (2015) 2647–2665, https://doi.org/10.1007/s10518-015-9731-4.

[6] L. Rovero, V. Alecci, J. Mechelli, U. Tonietti, M. De Stefano, Masonry walls with irregular texture of L’Aquila (Italy) seismic area: validation of a method for the evaluation of masonry quality, Mater. Struct. 49 (6) (2016) 2297–2314, https://doi.org/10.1617/s11527-015-0650-2.

[7] F. Lombardi, M. Lualdi, E. Garavaglia, Masonry texture reconstruction for building seismic assessment: practical evaluation and potentials of Ground Penetrating Radar methodology, Constr. Build. Mater. 299 (2021), 124189, https://doi.org/10.1016/j.conbuildmat.2021.124189. URL:https://www.sciencedirect.com/science/article/pii/S0950061821019498.

[8] S. Zhang, S.M. Taheri Mousavi, N. Richart, J.-F. Molinari, K. Beyer, Micro-mechanical finite element modeling of diagonal compression test for historical stone masonry structure, Int. J. Solids Struct. 112 (2017) 122–132, https://doi.org/10.1016/j.ijsolstr.2017.02.014.

[9] S. Tiberti, G. Milani, 2D pixel homogenized limit analysis of non-periodic masonry walls, Comput. Struct. 219 (2019) 16–57, https://doi.org/10.1016/j.compstruc.2019.04.002.

[10] S. Tiberti, G. Milani, 3D voxel homogenized limit analysis of single-leaf non-periodic masonry, Comput. Struct. 229 (2020), 106186, https://doi.org/10.1016/j.compstruc.2019.106186.

[11] M. Kržan, S. Gostič, S. Cattari, V. Bosiljkov, Acquiring reference parameters of masonry for the structural performance analysis of historical buildings, Bull. Earthq. Eng. 13 (1) (2015) 203–236, https://doi.org/10.1007/s10518-014-9686-x.

[12] G. Magenes, A. Penna, A. Galasco, M. da Paré, In-plane cyclic shear tests of undressed double leaf stone masonry panels, in: Proceedings 8th International Masonry Conference (April 2014), 2010, pp. 1–10. URL:https://www.researchgate.net/profile/Guido-Magenes/publication/257333315_In-plane_cyclic_shear_tests_of_undressed_double-leaf_stone_masonry_panels/links/0deec53600cb22062f000000/In-plane-cyclic-shear-tests-of-undressed-double-leaf-stone-masonry-panels.pdf.

[13] G. Magenes, A. Penna, A. Galasco, M. Rota, Experimental characterisation of stone masonry mechanical properties, in: Proceedings of the 8th International Masonry Conference, 2010, pp. 247–256. URL:https://www.researchgate.net/publication/257333264_Experimental_Characterisation_of_Stone_Masonry_Mechanical_Propertieshttps://www.researchgate.net/publication/257333264_Experimental_Characterisation_of_Stone_Masonry_Mechanical_Properties.

[14] M. Godio, F. Vanin, S. Zhang, K. Beyer, Quasi-static shear-compression tests on stone masonry walls with plaster: influence of load history and axial load ratio, Eng. Struct. 192 (April) (2019) 264–278, https://doi.org/10.1016/j.engstruct.2019.04.041.

[15] A. Rezaie, M. Godio, K. Beyer, Experimental investigation of strength, stiffness and drift capacity of rubble stone masonry walls, Constr. Build. Mater. 251 (2020), 118972, https://doi.org/10.1016/j.conbuildmat.2020.118972.

[16] F. Vanin, D. Zaganelli, A. Penna, K. Beyer, Estimates for the stiffness, strength and drift capacity of stone masonry walls based on 123 quasi-static cyclic tests reported in the literature, Bull. Earthq. Eng. 15 (12) (2017) 5435–5479, https://doi.org/10.1007/s10518-017-0188-5.

[17] MIT2019, Ministero delle infrastrutture e dei trasporti, circolare 21 gennaio 2019, n. 7 C.S.LL.PP.: Istruzioni per l'applicazione dell'Aggiornamento delle "Norme tecniche per le costruzioni"di cui al de- creto ministeriale 17 gennaio 2018.

[18] SIA 266/2:2012 Natural stone masonry (2012).

[19] P.B. Lourenço, Computational strategies for masonry structures, Ph.D. thesis, Delft University of Technology, 1996.

[20] M. Vocialta, N. Richart, J.-F. Molinari, 3D dynamic fragmentation with parallel dynamic insertion of cohesive elements, Int. J. Numer. Meth. Eng. 109 (12) (2017) 1655–1678, https://doi.org/10.1002/nme.5339.

[21] N. Richart, J.F. Molinari, Implementation of a parallel fi nite-element library: test case on a non-local continuum damage model, Finite Elem. Anal. Des. 100 (2015) 41–46, https://doi.org/10.1016/j.finel.2015.02.003.

[22] N. Kassotakis, V. Sarhosis, Employing non-contact sensing techniques for improving efficiency and automation in numerical modelling of existing masonry structures: a critical literature review, Structures 32 (March) (2021) 1777–1797, https://doi.org/10.1016/j.istruc.2021.03.111.

[23] I. Lubowiecka, P. Arias, B. Riveiro, M. Solla, 2D pixel homogenized limit analysis of non-periodic masonry walls, Comput. Struct. 89 (17–18) (2011) 1615–1627, https://doi.org/10.1016/j.compstruc.2011.04.016. URL:https://www.sciencedirect.com/science/article/pii/S0045794911001301?via%3Dihub.

[24] T. Hinks, H. Carr, L. Truong-Hong, D.F. Laefer, Point cloud data conversion into solid models via point-based voxelization, J. Surv. Eng. 139 (2) (2013) 72–83, https://doi.org/10.1061/(asce)su.1943-5428.0000097. URL:https://ascelibrary.org/doi/10.1061/%28ASCE%29SU.1943-5428.0000097.

[25] G. Castellazzi, A.M. D'Altri, G. Bitelli, I. Selvaggi, A. Lambertini, From laser scanning to finite element analysis of complex buildings by using a semi-automatic procedure, Sensors 15 (8) (2015) 18360–18380, https://doi.org/10.3390/s150818360. URL:https://www.mdpi.com/1424-8220/15/8/18360.

[26] U. Almac, I.P. Pekmezci, M. Ahunbay, Numerical analysis of historic structural elements using 3D point cloud data, Open Constr. Build. Technol. J. 10 (1) (2016) 233–245, https://doi.org/10.2174/1874836801610010233. URL:https://opencon structionbuildingtechnologyjournal.com/VOLUME/10/PAGE/233/.

[27] G. Castellazzi, A.M. D'Altri, S. de Miranda, F. Ubertini, An innovative numerical modeling strategy for the structural analysis of historical monumental buildings, Eng. Struct. 132 (2017) 229–248, https://doi.org/10.1016/j.engstruct.2016.11.032.

[28] B. Conde, L.F. Ramos, D.V. Oliveira, B. Riveiro, M. Solla, Structural assessment of masonry arch bridges by combination of non-destructive testing techniques and three-dimensional numerical modelling: application to Vilanova bridge, Eng. Struct. 148 (2017) 621–638, https://doi.org/10.1016/j.engstruct.2017.07.011. URL:https://www.sciencedirect.com/science/article/pii/S0141029617306077?via%3Dihub.

[29] L.J. Sánchez-Aparicio, B. Riveiro, D. González-Aguilera, L.F. Ramos, The combination of geomatic approaches and operational modal analysis to improve calibration of finite element models: a case of study in Saint Torcato Church (Guimarães, Portugal), Constr. Build. Mater. 70 (2014) 118–129, https://doi.org/10.1016/j.conbuildmat.2014.07.106. URL:http://www.sciencedirect.com/science/article/pii/S095006181400868X.

[30] R. Rolin, E. Antaluca, J.L. Batoz, F. Lamarque, M. Lejeune, From point cloud data to structural analysis through a geometrical hBIM-oriented model, J. Comput. Cult. Herit. 12 (2) (2019), https://doi.org/10.1145/3242901.

[31] Á. Bautista-De Castro, L.J. Sánchez-Aparicio, P. Carrasco-García, L.F. Ramos, D. González-Aguilera, A multidisciplinary approach to calibrating advanced numerical simulations of masonry arch bridges, Mech. Syst. Signal Process. 129 (2019) 337–365, https://doi.org/10.1016/j.ymssp.2019.04.043. URL:https://www.sciencedirect.com/science/article/pii/S0888327019302778?via%3Dihub.

[32] P. Morer, I. de Arteaga, A. Ortueta, A low-cost photogrammetric methodology to obtain geometrical data of masonry arch bridges, J. Archit. Conserv. 19 (3) (2013) 246–264, https://doi.org/10.1080/13556207.2013.869974.

[33] N. Kassotakis, V. Sarhosis, M.V. Peppa, J. Mills, Quantifying the effect of geometric uncertainty on the structural behaviour of arches developed from direct measurement and Structure-from-Motion (SfM) photogrammetry, Eng. Struct. 230 (September 2020) (2021), 111710, https://doi.org/10.1016/j.engstruct.2020.111710. URL:https://www.sciencedirect.com/science/article/pii/S014102962034311X?via%3Dihub.

[34] B. Riveiro, J.C. Caamaño, P. Arias, E. Sanz, Photogrammetric 3D modelling and mechanical analysis of masonry arches: an approach based on a discontinuous model of voussoirs, Autom. Constr. 20 (4) (2011) 380–388, https://doi.org/10.1016/j.autcon.2010.11.008.

[35] R. Napolitano, M. Hess, B. Glisic, Quantifying the differences in documentation and modeling levels for building pathology and diagnostics, Arch. Comput. Methods Eng. 27 (4) (2020) 1135–1152, https://doi.org/10.1007/s11831-019-09350-y.

[36] Y. Ibrahim, B. Nagy, C. Benedek, Deep learning-based masonry wall image analysis, Remote Sens. 12 (23) (2020) 1–28, https://doi.org/10.3390/rs12233918. URL:https://www.mdpi.com/2072-4292/12/23/3918.

[37] E. Valero, F. Bosché, A. Forster, Automatic segmentation of 3D point clouds of rubble masonry walls, and its application to building surveying, repair and maintenance, Autom. Constr. 96 (August) (2018) 29–39, https://doi.org/10.1016/j.autcon.2018.08.018. URL:https://www.sciencedirect.com/science/article/pii/S0926580517309895?via%3Dihub.

[38] Y. Shen, J. Wang, S. Wei, D. Zheng, V.G. Ferreira, Accurate extraction of brick shapes in masonry walls from dense terrestrial laser scanning point cloud, Measurement 146 (2019) 254–267, https://doi.org/10.1016/j.measurement.2019.05.086.

[39] G. Luchin, L.F. Ramos, M. D'Amato, Sonic tomography for masonry walls characterization, Int. J. Archit. Herit. 14 (4) (2020) 589–604, https://doi.org/10.1080/15583058.2018.1554723.

[40] M.R. Valluzzi, E. Cescatti, G. Cardani, L. Cantini, L. Zanzi, C. Colla, F. Casarin, Calibration of sonic pulse velocity tests for detection of variable conditions in masonry walls, Constr. Build. Mater. 192 (2018) 272–286, https://doi.org/10.1016/j.conbuildmat.2018.10.073.

[41] M.E. Stavroulaki, B. Riveiro, G.A. Drosopoulos, M. Solla, P. Koutsianitis, G. E. Stavroulakis, Modelling and strength evaluation of masonry bridges using terrestrial photogrammetry and finite elements, Adv. Eng. Softw. 101 (2016) 136–148, https://doi.org/10.1016/j.advengsoft.2015.12.007.

[42] E. Cescatti, L. Rosato, M.R. Valluzzi, F. Casarin, An automatic algorithm for the execution and elaboration of sonic pulse velocity tests in direct and tomographic arrangements, in: R. Aguilar, D. Torrealva, S. Moreira, M.A. Pando, L.F. Ramos (Eds.), Structural Analysis of Historical Constructions, Springer, Cham, 2019, pp. 716–724, https://doi.org/10.1007/978-3-319-99441-3_77.

[43] V. Perez-Gracia, O. Caselles, J. Clapes, Ground penetrating radar assessment of historical buildings: the study of the roofs, columns and ground of Santa Maria del Mar, in Barcelona, Mediterranean Microwave Symposium 2015-Janua. doi: 10.1109/MMS.2015.7375496. URL: https://ieeexplore.ieee.org/document/7375496.

[44] M. Shaqfa, K. Beyer, A virtual microstructure generator for 3D stone masonry walls, Eur. J. Mech. A. Solids 96 (2022), 104656, https://doi.org/10.1016/j.euromechsol.2022.104656.

[45] S.H. Khajavi, N.H. Motlagh, A. Jaribion, L.C. Werner, J. Holmstrom, Digital Twin: Vision, benefits, boundaries, and creation for buildings, IEEE Access 7 (2019) 147406–147419, https://doi.org/10.1109/ACCESS.2019.2946515. URL:https://ieeexplore.ieee.org/document/8863491.

[46] R. Lu, I. Brilakis, Generating bridge geometric digital twins from point clouds, in: Proceedings of the 2019 European Conference on Computing in Construction, vol. 1, 2019, pp. 367–376. doi:10.35490/ec3.2019.182. URL:https://ec-3.org/publications/conference/paper/?id=EC32019_182.

[47] E. Agapaki, I. Brilakis, Geometric Digital Twinning of Industrial Facilities: Retrieval of Industrial Shapes, arXiv (2022) 1–65 arXiv:2202.04834, doi:10.48550/arXiv.2202.04834. URL:http://arxiv.org/abs/2202.04834.

[48] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, J.-B. Huang, DeepMVS: Learning multi-view stereopsis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. URL:https://openaccess.thecvf.com/content_cvpr_2018/papers/Huang_DeepMVS_Learning_Multi-View_CVPR_2018_paper.pdf.

[49] M. Jancosek, T. Pajdla, Multi-view reconstruction preserving weakly-supported surfaces, in: Proceedings of Conference on Computer Vision and Pattern Recognition, 2011, pp. 3121–3128 doi:10.1109/CVPR.2011.5995693. URL:https://ieeexplore.ieee.org/document/5995693.

[50] J.L. Schonberger, J.M. Frahm, Structure-from-Motion revisited, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-Decem, 2016, pp. 4104–4113. doi:10.1109/CVPR.2016.445. URL:https://ieeexplore.ieee.org/document/7780814.

[51] R. Szeliski, Computer vision: Algorithms and applications, second ed., Springer Science & Business Media, New York, 2022. URL:https://szeliski.org/Book/.

[52] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, second ed., Cambridge University Press, 2001. URL:https://www.robots.ox.ac.uk/vgg/hzbook/.

[53] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G.D. Lillo, Y. Lanthony, AliceVision Meshroom: an open-source 3D reconstruction pipeline, in: Proceedings of the 12th ACM multimedia systems conference, ACM Press, 2021, pp. 241–247, https://doi.org/10.1145/3458305.3478443.

[54] Y. Li, S. Wang, Q. Tian, X. Ding, A survey of recent advances in visual feature detection, Neurocomputing 149 (PB) (2015) 736–751, https://doi.org/10.1016/j.neucom.2014.08.003. URL:https://www.sciencedirect.com/science/article/pii/S0925231214010121?via%3Dihub.

[55] C. Harris, M. Stephens, A combined corner and edge detector, in: Vision Conference, 1988, pp. 147–151. doi:10.5244/C.2.23. URL:http://www.bmva.org/bmvc/1988/avc-88-023.html.

[56] H. Bay, T. Tuytelaars, L.V. Gool, SURF: Speeded up robust features, in: Computer Vision–ECCV 2006, 2006, pp. 404–417. doi:10.1007/11744023_32. URL:http://link.springer.com/chapter/10.1007/11744023_32.

[57] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110, https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[58] S. Leutenegger, C. Margarita, R.Y. Siegwart., BRISK: Binary robust invariant scalable keypoints, in: International conference on computer vision, 2011, pp. 2548–2555. URL:https://ieeexplore.ieee.org/document/6126542.

[59] P.F. Alcantarilla, A. Bartoli, A.J. Davison, KAZE features, in: European conference on computer vision 7577 LNCS (PART 6), 2012, pp. 214–227. doi:10.1007/978-3-642-33783-3_16. URL:https://link.springer.com/chapter/10.1007/978-3-642-33783-3_16.

[60] P.F. Alcantarilla, J. Nuevo, A. Bartoli, Fast explicit diffusion for accelerated features in nonlinear scale spaces, in: BMVC 2013 - Electronic Proceedings of the British Machine Vision Conference 2013, 2013, pp. 5. doi:10.5244/C.27.13. URL: http://www.bmva.org/bmvc/2013/Papers/paper0013/index.html.

[61] P. Moulon, P. Monasse, R. Marlet, Adaptive structure from motion with a contrario model estimation, in: Asian conference on computer vision, 2012, pp. 257–270 doi: 10.1007/978-3-642-37447-0_20. URL:https://link.springer.com/chapter/10.1007/978-3-642-37447-0_20.

[62] P.C. Hansen, V. Pereyra, G. Scherer, Least squares data fitting with applications, Johns Hopkins University Press, 2013. URL:https://books.google.ch/books?

hl=en&lr=&id=8IrZe3QX0LQC&oi=fnd&pg=PP2&dq=Least+squares+data+fitting+with+applications&ots=itMET7MTCd&sig=vrN310Fd6OVcghHWRwpMkqWHN-4&redir_esc=y#v=onepage&q=Leastsquaresdatafittingwithapplications&f=false.

[63] J.T. Betts, Solving the nonlinear least square problem: application of a general method, J. Optim. Theory Appl. 18 (4) (1976) 469–483, https://doi.org/10.1007/BF00932656. URL:https://link.springer.com/article/10.1007%2FBF00932656.

[64] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395, doi:do. URL:https://dl.acm.org/doi/10.1145/358669.358692.

[65] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2564–2571. doi:10.1109/ICCV.2011.6126544. URL:https://ieeexplore.ieee.org/document/6126544.

[66] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: Computer Vision – ECCV, vol. 3951, LNCS, 2006, pp. 430–443, https://doi.org/10.1007/11744023_34.

[67] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary robust independent elementary features, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6314 LNCS (PART 4), 2010, pp. 778–792. doi:10.1007/978-3-642-15561-1_56. URL:https://link.springer.com/chapter/10.1007/978-3-642-15561-1_56.

[68] N. Snavely, S.M. Seitz, R. Szeliski, Photo Tourism: Exploring photo collections in 3D, ACM Siggraph (2006) 835–846, https://doi.org/10.1145/1179352.1141964.

[69] B. Pantoja-Rosero, R. Achanta, M. Kozinski, P. Fua, F. Perez-Cruz, K. Beyer, Generating LOD3 building models from structure-from-motion and semantic segmentation, Autom. Constr. 141 (2022), 104430, https://doi.org/10.1016/j.autcon.2022.104430.

[70] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM, 2001, pp. 145–152. doi:10.1109/IM.2001.924423. URL:https://ieeexplore.ieee.org/document/924423.