ARTICLE IN PRESS

European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

An image convolution-based method for the irregular stone packing problem in masonry wall construction

Qianqing Wang *, Bryan German Pantoja-Rosero, Ketson R.M. dos Santos, Katrin Beyer

Earthquake Engineering and Structural Dynamics Laboratory (EESD), EPFL, 1015 Lausanne, Switzerland

ARTICLE INFO

Dataset link: https://github.com/eesd-epfl/StablePacking-2D/

Keywords:
Irregular stones
Stable packing
Dry-stack masonry
Image convolution

ABSTRACT

The use of natural stones as building material can help reducing the carbon footprint of the construction industry. However, their non-uniform shapes makes the construction of stone masonry structures challenging. Therefore, the development of efficient algorithms for the stacking of irregular stones obeying structural and architectonic requirements is essential. In this paper, we propose an image-based method for automating the stacking of non-uniform stones in the construction of 2D load-resistant stone masonry walls. Stone wedging, a traditional technique employed by skilled masons, is implemented to reinforce the stability of stone placements. We use image processing for accelerating the stone selection and placement, and determine the wall's resistance using a variational rigid-block modeling approach. It is demonstrated that the developed method is efficient and robust in challenging conditions. The analysis of the computational performance of the presented method shows that it is suitable for automated construction.

1. Introduction

Construction industry is responsible for 30%–40% of energy consumption as well as for high levels of CO_2 emission (Dabaieh et al., 2020). The major contributions to the emission of CO_2 are the extraction, production, and transportation of raw materials (Koroneos & Dompros, 2007). On the other hand, stones have an extremely low embodied energy (Morel et al., 2001) and the potential to be recycled without reprocessing. However, constructing load-bearing structural elements with natural stones is challenging due to their irregular geometry and the massive shortage of skilled masons (Brehm, 2019).

When the skill level of the craftsman is low, the structural performance of the built structure is drastically reduced (Bothara & Brzev, 2012). To leverage the labor work and to improve its efficiency, the construction of masonry structures can be conducted by robots, where the construction process is calculated by fast computer algorithms used to feed the robots with the most appropriate sequence of actions necessary to achieve a given structural goal. A family of such algorithms are devoted to performing stacking of solid object such as regular-shaped objects (Kollsker & Malaguti, 2021), irregular-shaped objects (Paxton et al., 2022), and natural materials (Furrer et al., 2017). In the context of stone masonry, stone stacking algorithms are developed for arranging the placement of stones during the construction of masonry structures ensuring its structural and architectonic goals. In contrast to the broader category of packing problems (Grandcolas & Pain-Barre, 2021; Leao et al., 2020; Umetani & Murakami, 2022), the stone stacking problem specifically concentrates on achieving structural stability in the assembly and adhering to the principles of traditional masonry craftsmanship.

Different stacking algorithms have been proposed to generate dry-stack masonry walls. Furrer et al. (2017) considered the structural stability in the construction of a stone tower using the relative position of the center of mass of the stones and the support polygon. Recent developments also include the use of dynamic simulations to remove unstable stone positions (Johns et al., 2020; Liu et al., 2021; Wermelinger et al., 2018). However, these works only considered gravity load, and they did not account for lateral loading, which is non-negligible for real-world structures. On the other hand, Liu et al. (2018) and Thangavelu et al. (2018) considered the lateral resistance in their stacking strategies by simulating the shake table test, where the built walls are shaken with an increasing amplitude until the point where the stone displacements reached a given limit. Their method provides a quantitative mean to evaluate the structural performance of built walls; however, it is computationally expensive.

To optimize the load bearing capacity of masonry walls by manipulating the stone layout, a simulation method that is capable to differentiate the mechanical response of masonry walls as a function of the stone layout is needed. Such micro-scale modeling can be achieved by finite element method, distinct element method, and rigid block

* Corresponding author.
E-mail addresses: qianqing.wang@epfl.ch (Q. Wang), bryan.pantojarosero@epfl.ch (B.G. Pantoja-Rosero), ketson.santos@epfl.ch (K.R.M. dos Santos), katrin.beyer@epfl.ch (K. Beyer).

https://doi.org/10.1016/j.ejor.2024.01.037
Received 30 August 2023; Accepted 25 January 2024
Available online 7 February 2024

models (de Felice, 2011; Portioli et al., 2021; Wang et al., 2023; Zhang et al., 2017). As the efficiency of the stacking algorithm is one of our main concerns, we adopt rigid block models to conduct a kinematic analysis due to its computational efficiency (Portioli et al., 2021).

This paper proposes a comprehensive stacking method for the construction of dry stone masonry walls accounting for the rules-of-art practised by masons and structural performance evaluated by numerical simulations. With regard to the state-of-the-art, the developed method introduces three novel features: We use image processing techniques to accelerate the optimization processing and use a kinematic analysis to assess the structural performance of the built wall in terms of its lateral resistance. In addition, stone wedging, a technique employed by skilled masons, is formulated in the algorithm to stabilize stone poses. This approach can facilitate the development of automated construction tools due to its reduced computational cost, robustness, and ability to generate walls consistent with the engineering practice.

In the following, we first introduce in Section 2 the characteristics of dry joint stone masonry walls as well as the state-of-the-art in stacking algorithms. The developed stone stacking algorithm is then presented in Section 3. It includes, the selection of candidate stones, the optimal placement strategy, the stability assessment, and the selection of the best stone for placement. Section 4 introduces the procedure employed in the structural analysis of stone masonry walls based on the limit-state behavior. Benchmarks are presented in Section 5, including examples based on images of real and artificially generated stones. The built walls are compared with reference walls using four evaluation metrics. The sensitivity of the developed method with respect to changes in the sampling method and the image resolution are also discussed in Section 6. The final remarks are presented in Section 7.

2. Dry stone walls

Dry stone masonry walls are unmortared structures typically utilized as retaining walls, boundary walls, and other simple building forms (Mundell et al., 2010). Depending on the local availability of stones and the regional constructive practice, dry stone walls show different styles (Mundell et al., 2010). Despite the regional variation, this type of structural system are commonly characterized by a tight packing, straight course, interlocking, and wedging (Mundell et al., 2010), as seen in Fig. 1. A tight packing of stones in a masonry structure is achieved by placing stones as close as possible and filling voids with small blocks (Vivian, 2014). Professional masons, usually have the freedom to shape the available stones according to the need to achieve a dense stacking. To the extent possible, horizontal layers, referred to as “course”, are composed of stones with uniform height, presenting a straight level appearance (Mundell et al., 2010). When there is a wide variation in stone heights, small stones are packed together to match the height of large stones such that a straight course is obtained for preserving a flat composition of layers (McRaven, 1999). Moreover, each block should ideally be in contact with several other stones (McRaven, 1999; Mundell et al., 2010; Vivian, 2014). This is achieved through offsetting vertical joints to form “interlocking”. Furthermore, small shards of stones can be inserted to fill openings in the face (Vivian, 2014) or to prevent rocking (Mundell et al., 2010), this process is also known as wedging.

These construction principles were developed by professional masons over generations. However, considering the current shortage of skilled masons in the job market, robots (Brehm, 2019) and AR supported construction (Settimi et al., 2023) emerge as alternative for increasing the productive of the construction industry in such challenging scenario. To ensure the quality of the structure constructed by such machines, the mason’s intuition must be translated into computer-readable rules. Table 1 summarizes the aspects that have been considered by the state-of-the-art stone stacking algorithms found in the literature. In addition to these geometrical features, factors such as stable footing, inward sloping, stability under gravity, and stability

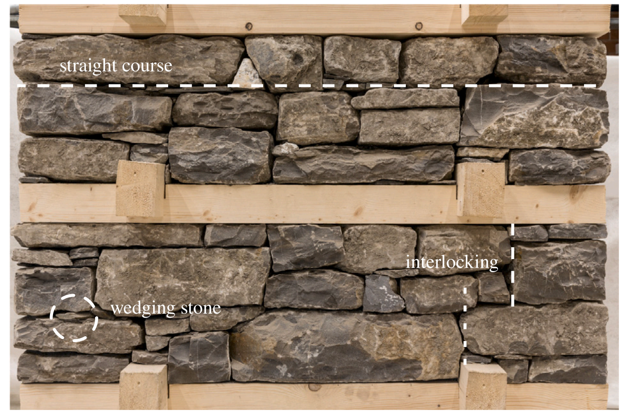


Fig. 1. Geometric features observed from a dry stone wall built by skilled masons (EPFL, 2023).

under lateral loading are also considered in the computational problem to ensure the structural stability.

The algorithms presented in Table 1 can be classified into three categories: (i) model-based method solved by linear scalarization, (ii) model-based method solved by hierarchical filtering, and (iii) model-free method. The model-based method translates the mason’s rules-of-thumb into mathematical formulations and optimization problems. As there are several construction rules, the problem becomes a multi-objective optimization problem and is commonly solved using either linear scalarization (LS) or hierarchical filtering (HF). The former formulates a single-objective optimization problem by summing all objectives linearly (Johns et al., 2020; Lambert & Kennedy, 2012). The latter removes undesirable solutions by forming hierarchical filters from the features (Liu et al., 2021; Thangavelu et al., 2018). The model-free method is based on machine learning techniques, typically referring to reinforcement learning (RL). The method formulates a reward function to train a model (often a neural network) that outputs the placement and order of stones (Liu et al., 2018; Menezes et al., 2021). The reward function presents the desired property of the wall, which is also inspired from masons’ practice.

It is important to highlight that several of these algorithms are developed for 2D problems. For their application in real-world construction scenarios, stones need to maintain a uniform shape along the third dimension, and their third dimension should be comparable. An example of this can be seen with synthetic stones created in the work of Liu et al. (2018), which prevents the out-of-plane rocking of stones. Additionally, the out-of-plane wall failure is not considered in the stability assessments. Our current work also focuses on a simplified 2D version of the stone stacking problem, yet it preserves the crucial aspects, such as geometric and physical constraints.

In this paper, a model-based method is developed for stacking raw stones by combining linear scalarization and hierarchical filtering. Our method is distinguished by its comprehensive integration of masons’ rules-of-art, notably as the pioneering approach to introduce wedging within the stacking process. Additionally, we are one of only two methods that account for the lateral resistance of the constructed wall. Compared to the dynamic simulation method utilized by Thangavelu et al. (2018), the kinematic analysis method we adopted significantly accelerates the determination of maximal resistance, thereby enhancing the efficiency of our approach.

3. Method

This section introduces the proposed method for stacking dry-joint walls. The problem’s scope and primary challenge are outlined in Section 3.1 . A high-level description of the proposed algorithm is provided in Section 3.2. Detailed explanations of the various components of the algorithm can be found in Sections 3.3 to 3.8.

Table 1
Components of objective function (for linear scalarization method), filters (for hierarchical filtering method), and reward function (for reinforcement learning method) in stone stacking algorithms. Approach developed for a 2D or 3D environment and tested with a physical experiment.

Components in formulation	Model-based					Model-free	
	LS + HF	LS		HF		RL	
	Ours	Johns et al. (2020)	Lambert and Kennedy (2012)	Liu et al. (2021)	Thangavelu et al. (2018)	Menezes et al. (2021)	Liu et al. (2018)
Masons' rules-of-art							
Tight packing	×	×	×		×	×	×
Straight course	×			×	×		
Interlocking	×	×		×	×		
Wedging	×						
Placement stability							
Stable footing	×			×	×		
Inward sloping	×	×		×	×		
Gravity	×	×		×	×	×	×
Lateral load	×				×		×
Others							
Physical experiment		×		×	×		×
2D/3D	2D	3D	2D	3D	2D	3D	2D

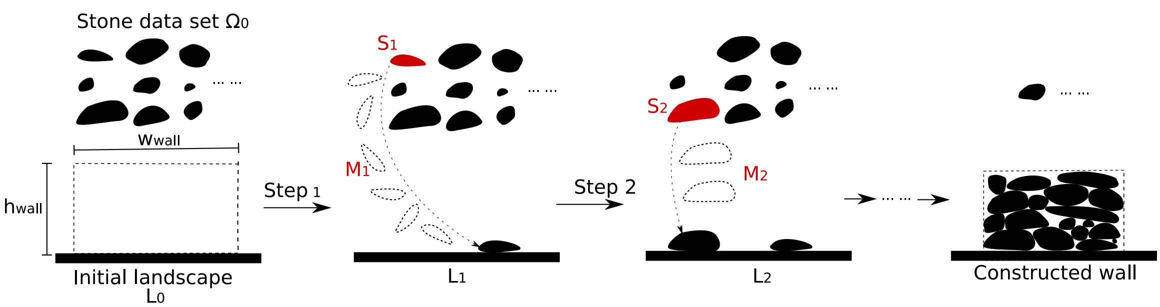


Fig. 2. Stone stacking problem as solving sequential actions $(M_1, S_1), (M_2, S_2), \ldots$.

3.1. Stone stacking problem

Given a stone set Ω_0, an initial landscape L_0, and a desired wall dimension (height h_{wall} and width w_{wall} for a 2D problem, see Fig. 2), the stone stacking problem search a sequence of actions $\{A_1, A_2, \ldots, A_i\}$ for constructing a dry joint masonry wall. Each action is composed by the selection and placement pair (M_i, S_i), where M_i correspond to the transformation of the stone S_i.

This problem has three constraints. The first constraint determines the physical possibility of an action for avoiding, for instance, the overlap of two stones. The second one is focused on the stability of the partially-built wall, where a placed stone $M_i S_i$ should not fall down and the self weight of the newly placed stone should not cause any rocking or lifting of the other placed stones. Finally, the layout of stones should follow the rules of art observed from masons practice (see Section 2).

3.2. Algorithm for dry stone stacking

The main challenge for developing a method for stacking stones and optimizing the structural performance of the constructed wall lies on the fact that the configuration of the wall depends on all the actions taken in the construction process. In this work, we employ a current-best greedy heuristics approach by considering the construction as a Markov chain where each step depends only on the current state of the process and not the past. This assumption was also used in the stacking algorithms developed by Liu et al. (2021) and by Thangavelu et al. (2018). In each step i, the algorithm conducts an action (M_i, S_i) and the landscape is updated from L_i to L_{i+1}.

Fig. 3 present the steps taken by the proposed method for constructing a stone masonry wall. It starts by setting an initial set of Q stones $\Omega_0 = \{S_1, S_2, \ldots, S_Q\}$, the design goals in terms of wall's width w_{wall} and height h_{wall}, and the initial landscape L_0 (see Section 3.3). The first step consists of rotating stones in the initial set to an optimal pose defined in Section 3.6. Then the stones are clustered using the method presented in Section 3.4. Next, a sequential processes of stone selection and placement starts. At each step i, a subset of K candidate stones $\Pi_i = \{S^i_m \mid S_1, \ldots, S^i_m, \ldots, S^i_K\}$ is sampled (see Section 3.4), and for each stone within this subset a trial placement (see Section 3.6) and consequent stabilization (see Section 3.7) are performed. Next, the best stone (S^i_{m*}) is chosen via hierarchical filtering and scalarized multi-objective optimization (see Section 3.8). The selected stone is then placed on the landscape, and the remaining ones are returned to their respective clusters, and the update of the landscape is represented by the following set of matrices

$$L_{i+1} = L_i + M_{m*} S^i_{m*} \tag{1}$$

$$M_{m*} = R_{s,m*} T_{m*} R_{0,m*} \tag{2}$$

where M_{m*} is the transformation to place stone S_{m*}, $m*$ is the index of the selected stone at step i; $R_{s,m*}$ is the rotation matrix obtained with the stabilization algorithm (see Section 3.7), T_{m*} and $R_{0,m*}$ are the translation and rotation matrices determined by the placement algorithm (see Section 3.6). If no candidate can be stably placed, the algorithm can resample candidates from the clusters and start another placement attempt. The number of consecutive failure is denoted as ζ. The construction continues until one of the following conditions are met: (i) the design goal is achieved, (ii) all stones are placed, and (iii) there are more than four consecutive failures. The mechanization of this method is presented in Algorithm 1. A detailed discussion about each component of the stacking algorithm developed herein are presented in the next sections.

3.3. Image-based data set

The stone data set are defined as binary images, where pixels corresponding to the stone domain are non-zero and pixels of the

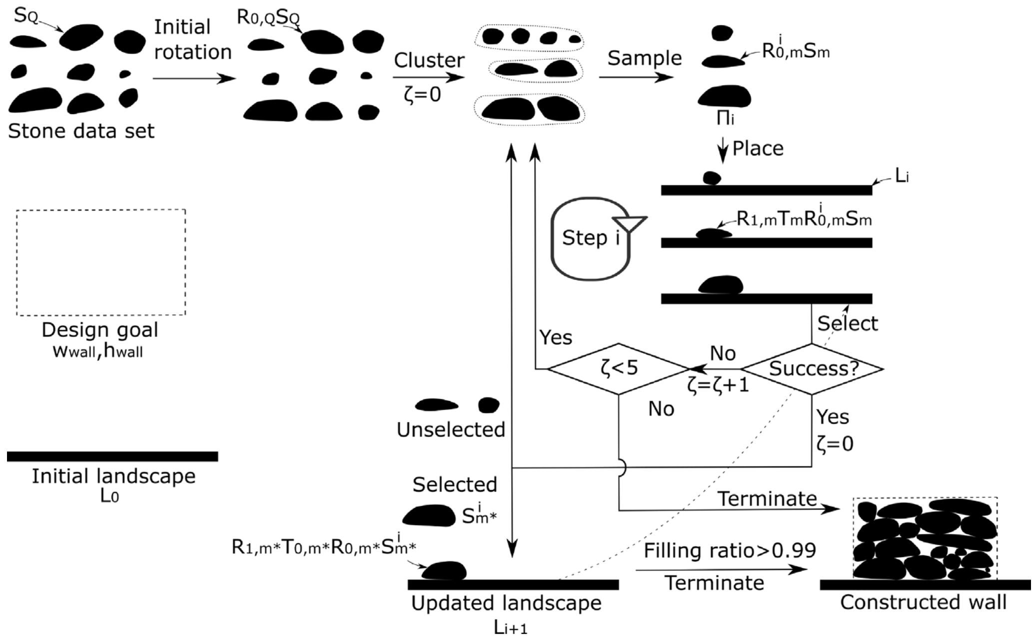
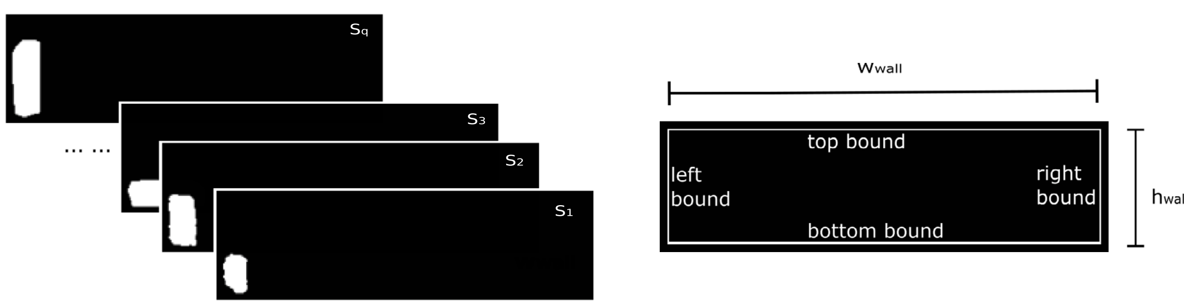


Fig. 3. Proposed stacking algorithm workflow for building dry joint walls.



(a) (b)

Fig. 4. Binary images of (a) stone data set, (b) initial landscape.

background assume the value 0, as seen in Fig. 4a for an example stone data set. The initial landscape, which is also represented by a binary image, defines the width w_{wall} and the height h_{wall} of the planning wall by four bounds with non-zero pixel values (see Fig. 4b). As stones shapes are represented in pixels, we also discretize the transformation space with 1-pixel intervals, and the rotation space with 1-degree intervals.

3.4. Sampling candidate stones

The first step in the construction process is the sampling of a candidate stone set $\{S_1 \dots S_K\}$. Clearly, one can use the whole data set as the candidate set. However, a clustering approach is used herein by identifying the stones by their similar features in order to sample a subset of candidate stones to reduce computational cost.

In this work, the geometrical similarity between two stones is determined by two features. One is the stone eccentricity, which measures how much it deviates from a circle. It is estimated by the eccentricity of an ellipse with similar second-moment, which is defined as the ratio of the focal distance (distance between focal points) over the major axis length. The other feature is the stone size, estimated as the ratio of the area of its bounding box to the largest bounding box area in the stone data set. Next, the agglomerative clustering method (Ward, 1963) is used with a distance threshold of 0.2. Fig. 5 shows an example of a set with 32 stones which is separated into 7 clusters. Then candidate stones are selected at random from each cluster.

3.5. Evaluation indices of an action (M_m, S_m)

In the construction step i, an action consists of selecting and placing a stone S_m* in the current landscape L_i using the transformation M_m. To compare different actions, we propose a set of indices. Among them one can include the shape factor (SF) measuring the deviation of each stone's shape from the ideal rectangular form defined as the ratio of the area of the stone to that of its axis aligned bounding box (Almeida et al., 2016):

$$\mathrm{SF}(M_m, S_m) = \frac{A_{S_m}}{A_{B_{M_m S_m}}}, \tag{3}$$

where A stands for area, B stands for the bounding box which is aligned with the global frame, $M_m S_m$ refers to the transformed pose of stone S_m. If the stone is a completely axis-aligned rectangle, the index attains its highest value of 1, as shown in Fig. 6.

Another index in the local filling ratio (LFR) which assumes the stones should form a densely packed structure. It is defined as

$$\mathrm{LFR}(M_m, S_m) = \frac{A_{mask,fill}}{A_{mask}}, \tag{4}$$

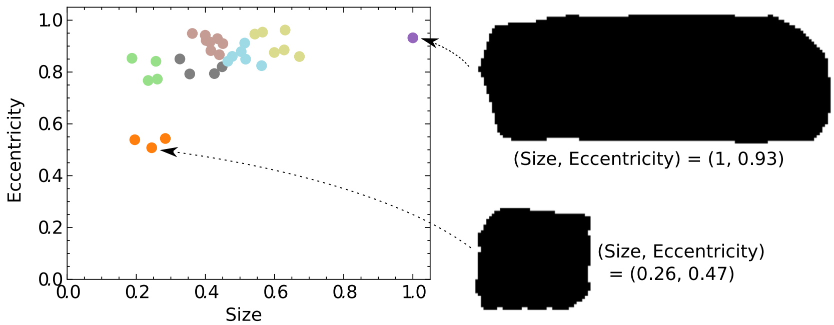


Fig. 5. Clustering result of the stones to build the PR wall (Ward, 1963).

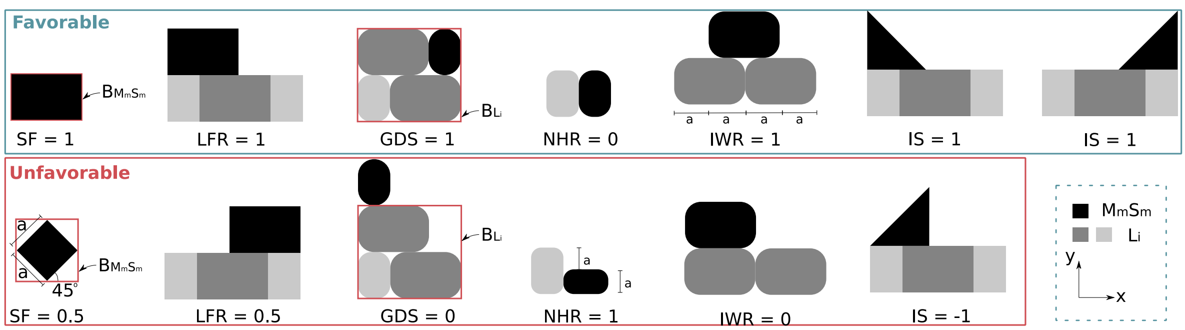


Fig. 6. Favorable and unfavorable cases for the proposed geometric indices.

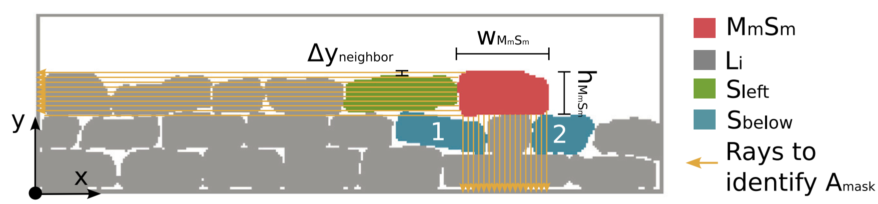


Fig. 7. Illustration of calculating indices of a stone M_mS_m on a landscape L_i.

where A_{mask} represents the area covered by arrows in Fig. 7. Fig. 7 is an example of a landscape L_i in the construction step i. The current stone M_mS_m to be evaluated is in red. The mask to calculate the LFR of the current stone refers to the part of the image that is covered by "rays" pointing from the contour of the current stone to the border of the wall. Here we consider "rays" pointing to the negative direction of x and y, as we assume the stones are packed from bottom to top, and from left to right, and $A_{mask,fill}$ represents the area inside the mask that is occupied by other stones. The index ranges from 0 to 1, where a higher value indicates a denser layout on the left-side of the stone and below the stone. Examples of favorable and unfavorable placement are illustrated in Fig. 6.

The geometrical dice score (GDS) is an index inspired on the formulation of the dice score (Pantoja-Rosero et al., 2022; Sorensen, 1948) to measure how much an action can fill the current partially-built wall. It is written as

$$\mathrm{GDS}(M_m, S_m) = \frac{A_{B_{L_i}}}{A_{M_mS_m}} \times \frac{A_{M_mS_m \cap B_{L_i}}}{A_{M_mS_m \cup B_{L_i}}}, \tag{5}$$

where B_{L_i} is the bounding box of landscape L_i. The index varies between 0 and 1, where a value of 1 indicates that the stone is entirely located within the bounding box of the present landscape, while a value of 0 implies that it is completely outside (Fig. 6).

Further, the neighbor height ratio (NHR) uses the height of the current stone and its neighbors to quantify the straightness of the course. As we assume the stones are placed from left to right, one can use the left neighboring stone (shown in green in Fig. 7) to calculate the height difference $\Delta y_{neighbor}$. The neighbor height ratio is obtained by comparing the height difference with the height of the stone:

$$\mathrm{NHR}(M_m, S_m) = \frac{\Delta y_{neighbor}}{h_{M_mS_m}} \tag{6}$$

$$\Delta y_{neighbor} = \mid \max y_{M_mS_m} - \max y_{left} \mid . \tag{7}$$

Herein, y_{left} refers to the y coordinates of the pixels occupied by the left neighboring stone shown in green in Fig. 7. A straight course is represented by NHR = 0. In the extreme case, where M_mS_m is the first stone of a new course, one can use the difference of the average height of all other stones in Π_i and that of the current stone to approximate this value, which is written as

$$\Delta y_{neighbor} = h_{M_mS_m} - \sum_{q=1}^{Q} h_{S_q}. \tag{8}$$

The interlocking (see Section 2) quantifies how much a stone deviates from the stack bond. Thus, interlocking width ratio (IWR) is defined herein as the ratio of the vertical joint offset distance to the width of the underlying stone, and it is written as follows

$$\mathrm{IWR}(M_m, S_m) = \left(\frac{\|\max(x_{M_mS_m}) - \min(x_{below_2})\|}{w_{below_2}} + \frac{\|\min(x_{M_mS_m}) - \min(x_{below_1})\|}{w_{below_1}} \right), \tag{9}$$

where x is the horizontal coordinate and w is the width of a stone. The index $below_1$ and $below_2$ refer to the following two stones: S_{below_1}

Algorithm 1: Stone stacking algorithm

Input : $\Omega_0 = \{S_q \mid S_1\ S_2\ \dots S_q\ q \le Q\}$, L_0, w_{wall}, h_{wall}
Output: $(A_0, A_1, \dots, A_i)$
begin
  cluster the stone data set Ω_0 ; ▷ Section 3.4
  construction step $i = 0$;
  number of consecutive failures $\zeta = 0$;
  while $i \le Q$ **do**
    sample a candidate stone set Π_i from Ω_i ; ▷ Section 3.4
    for *each stone* $S_m \in \Pi_i$ **do**
      solve the optimal initial pose $R_{0,m}$; ▷ Section 3.6
      solve the optimal translation T_m ; ▷ Section 3.6
      if *placement is stable* **then**
        $R_{s,m} = I$;
      else
        solve the stabilization matrix $R_{s,m}$; ▷ Section 3.7
      end
      $M_m = R_{s,m} T_m R_{0,m}$;
    end
    select the best action $A_i = (M_{m*}, S_{m*})$; ▷ Section 3.8
    if *selection succeed* **then**
      update the current landscape $L_{i+1} = L_i + M_{m*} S_{m*}$;
      update the stone data set $\Omega_{i+1} = \Omega_{i+1} - S_{m*}$;
      $i = i + 1$;
      $\zeta = 0$;
    else
      resample candidate stone set;
      $\zeta = \zeta + 1$;
    end
    if *filling ratio* >0.99 *or* $\zeta \ge 5$ **then**
      construction terminates;
    end
  end
end

is the stone that has its top right corner located closest to the bottom left corner of the current stone's bounding box. S_{below_2} is the stone that has its top left corner located closest to the bottom right corner of the current stone's bounding box. They are shown in blue in Fig. 7. The index IWR varies between 0 and 1, with higher value indicating better interlocking (Fig. 6).

The inward sloping (IS) is another index developed herein. An inward-sloping top surface (large IS) on the boundary is favored by the rules of art of dry stone masonry and is also implemented in other stacking algorithms (Liu et al., 2021). Such characteristic is quantified by the slope measured between the highest points of the left side and of the right side of the stone, and it is written as

$$\mathrm{IS}(M_m, S_m) = \frac{\max_{0 \le x \le x_{c,M_m S_m}} y_{M_m S_m} - \max_{x_{c,M_m S_m} \le x \le w_{wall}} y_{M_m S_m}}{\operatorname{argmax}_{0 \le x \le x_{c,M_m S_m}} y_{M_m S_m} - \operatorname{argmax}_{x_{c,M_m S_m} \le x \le w_{wall}} y_{M_m S_m}} \times \operatorname{sign}(x_{c,M_m S_m} - x_{c,wall}). \tag{10}$$

Here $x_{c,wall}$ refers to the x coordinate of the center of the wall. By comparing the x coordinate of the center of the stone and the center of the wall, we distinguish between stones placed on the left and those on the right. This ensures that a good placement always yield a slope towards the inner part of the wall, as illustrated in Fig. 6.

Further, the limit load coefficient (LLC) is a normalized load multiplier obtained from kinematic analysis (see a detailed explanation in Section 4), which is written as

$$\mathrm{LLC}(M_m, S_m) = \frac{\alpha}{\mu} \tag{11}$$

where α is the load multiplier and μ refers to the coefficient of friction. The normalized feature quantifies the maximal lateral resistance of the structure and ranges between 0 and 1.

3.6. Trial placement of each candidate stone

Once the set of candidate stones is sampled, one can proceed with the trials, where each stone S_m is tested for the best placement on the current landscape. To this end, a novel algorithm is proposed herein for analyzing and rating the transformations to all possible positions on the current landscape L_i. Then, one can determine the transformation M_m maximizing the objective function, as presented in the following optimization problem

$$\max_{M_m} \quad \mathrm{LFR}(M_m, S_m) + H(\mathrm{GDS}(M_m, S_m), \beta) - \frac{x_{c,M_m S_m}}{w_{wall}}, \tag{12a}$$

$$\text{s.t.} \quad M_m = T_m R_{0,m}, \tag{12b}$$

$$(M_m S_m) \odot L_i = \mathbf{0}, \tag{12c}$$

$$R_{0,m} = \operatorname{argmax} \quad \mathrm{SF}(R_{0,m} S_m), \tag{12d}$$

Here $\odot$ refers to the element-wise multiplication between two matrices. The transformation M_m is composed of rotation $R_{0,m}$ and translation T_m matrices. These matrices are applied to S_m, which is a candidate stone, to obtain its transformed configuration $M_m S_m$. Further, $x_{c,M_m S_m}$ denotes the center of the transformed stone and $H(\cdot,\cdot)$ is a nonlinear function given by

$$H(x, \beta) = \begin{cases} 0 & x \le \beta \\ x & x > \beta, \end{cases} \tag{13}$$

In the majority of cases evaluated, we found that using this nonlinear function with a value of β equal to 0.7 resulted in better outcomes compared to solely utilizing GDS or utilizing values of β equal to 0, 0.5, or 0.9. This phenomenon may be attributable to the exclusive contribution of high GDS values in fulfilling the existing landscape.

Two constraints are considered in the formulated optimization problem. First, the constraint in Eq. (12c) determines that the transformed stone $M_m S_m$ should not overlap the landscape L_i. On the other hand, the constraint in Eq. (12d) imposes that the rotation R_{0,S_m} of the stone S_m orient the stone to a pose where SF is maximized.

To narrow down the possible solutions before evaluating all positions, the landscape image is processed beforehand. The objective of this process is to remove impossible placements of the next stone. We first filter out overlapping positions, as the next stone should not overlap with the landscape. We then use a convolution to determine non-overlapping positions by assuming the stone image as the convolution kernel. Fig. 8 shows an example of a landscape and its matrix representation. The stone image, which determines the convolution kernel, is shown as a black rectangle. The resulting non-overlapping positions obtained from convolution are then illustrated as a red shadow in the matrix representation.

Next, we filter out positions where the stone is not in touch with the landscape (i.e. floating positions). To find these contact positions, we dilate the landscape by one pixel, and use the same stone kernel to execute another convolution. The resulting candidate positions are shadowed in blue in the matrix representation (see Fig. 8). With such a dilation-convolution method, we are able to detect positions that allow the stones to be in touch with the previous landscape. The difference of the two convolution results (red and blue in Fig. 8) produce the feasible solutions as represented by the pink color in the landscape. With the reduced solution space, we evaluate the value of the objective function defined in Eq. (12) for all positions. The candidate stone will then be placed on the optimal position shown as the black rectangle placed on the landscape in Fig. 8.

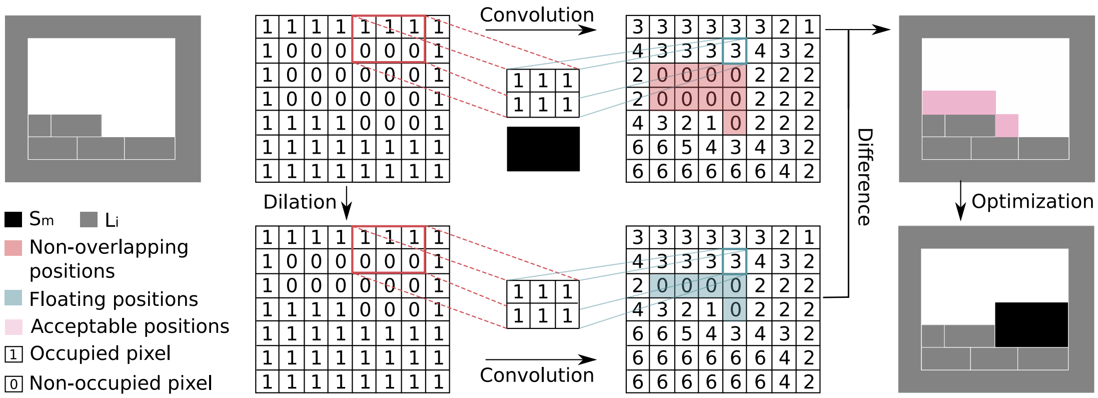


Fig. 8. Image processing steps to determine acceptable positions for placing the stone S_m on the landscape L_i. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.7. Stabilization of each trial placement by rotation or wedging

As observed in Fig. 9a, there is always a possibility that the stone placement is not stable. To avoid such conditions, one stability assessment criterion and two stabilization methods are introduced in this section. In this regard, a stone is considered stable when the vertical projection of its center of mass (CoM) lies within the support polygon, which is defined as the polygon that envelops the contact points between the stone and the landscape (Furrer et al., 2017). In 2D, the support polygon is given by a polyline connecting the contact points A and B (Fig. 9a). To assess the stability of a placement, we first find the contact point between the stone and the landscape. This is achieved by dilating the stone image by one pixel and checking if it overlaps with the landscape. Once the contact points are detected, one can determine the support polyline and examine whether the CoM of the stone (point **C** in Fig. 9a) can be vertically projected on it. If this is the case, the stone placement is considered stable. Otherwise, one of the stabilization methods must be used.

The first method involves rotating the stone around the point **A** with a rotation matrix denoted as $R_{s,m}$. Point **A** is defined as the end of the support polyline that is closer to the Center of Mass (CoM, as shown in Fig. 9a). To determine the direction of rotation, the sign of the cross product between the vectors **AB** and **AC**, i.e., $\mathbf{AB} \times \mathbf{AC}$, is used. The rotation is in the counterclockwise direction if $\mathbf{AB} \times \mathbf{AC}$ is positive, and negative otherwise. The rotation stops when new contact points are detected. The process can be carried out iteratively, with point **A** dynamically changing based on new contact points, until the vertical projection of the CoM (point $\mathbf{C}'$ in Fig. 9b) lies on the new support polyline. The aforementioned approach may fail in cases where the stone overlaps with the landscape before new contact points being identified. Therefore, we propose the second method outlined below.

The second method developed herein, referred to as wedging, is inspired on a technique known as “wedging” which is used by masons. This technique consists of adding small stones, also known as “wedge stones”, below the bigger ones as a mean to fill the void causing the observed instability (shown in Fig. 1). In this method, the supported side, which is covered by the gray square in Fig. 9c, is not considered in the “wedging” process. Therefore, the uncovered part of the stone is expanded until it touched another one. The size of the expansion determines the height h_{wedge} of the wedge stone, and the width of the contact area determines its width w_{wedge}, see Fig. 9c. The shortcoming of this technique is that it requires additional small stones, which might not be available. Thus, in this paper, these methods would work hierarchically such that the rotation-based stabilization has the priority, whereas wedging is employed only when the rotation-based method fails.

3.8. Selection of the best action

Once the transformation of all candidate stones is solved, the algorithm proceeds with the selection of the best action (stone and its transformation) from the set Π_i. To this end, we apply a combination of hierarchical filtering and linear scalarization of multi-objective optimization, shown in Fig. 10. The pseudo code of this algorithm can be found in Appendix A. The algorithm shown in Fig. 10 starts with a matrix of K rows, where each row represents the five indices of one candidate stone. The third column of the matrix indicates the IWR for inner stones, whereas for boundary stones, it indicates their IS. We distinguish between inner stone and boundary stone by comparing the minimum width of stones in the data set with the smallest distance from the stone $M_m S_m$ to the left boundary and the minimum distance from the stone $M_m S_m$ to the right boundary. If the former is larger, the stone $M_m S_m$ is considered as an edge stone and we use IS as the corresponding third index (the third column of the matrix). Otherwise, IWR is assigned to the third index.

The initial step involves filtering out candidates with a LLC less than or equal to zero, which ensures that the placement remains stable under gravity. The resulting set of candidate indices, consisting of K_1 stones, is stored in a new matrix with K_1 rows and five columns. If no stone can pass the first filter, i.e. $K_1 \leq 0$, the construction process ends. Otherwise, the set of K_1 stones are filtered further with the second filter $\mathrm{LLC}_m > \overline{\mathrm{LLC}}$ into a set of K_2 stones. This filter leaves actions whose stability index (LLC) is larger than the average value, which is calculated as

$$\overline{\mathrm{LLC}} = \frac{1}{K_1} \sum_{m=1}^{K_1} \mathrm{LLC}(M_m, S_m). \tag{14}$$

If no stone can pass the filter, i.e. $K_2 \leq 0$, the best stone S_m is chosen as the one whose indices maximize the following objective

$$\max_m \quad \mathrm{LLC}_m, \tag{15a}$$

$$\text{s.t.} \quad m = 0, \dots, K_1. \tag{15b}$$

Otherwise the hierarchical filtering process continues to the third filter that selects stones whose interlocking index (NHR) is larger than the average value over the K_2 stones and obtain a set of K_3 stones. The average value of NHR in this filter can be calculated as

$$\overline{\mathrm{NHR}} = \frac{1}{K_2} \sum_{m=1}^{K_2} \mathrm{NHR}(M_m, S_m). \tag{16}$$

If $K_3 \leq 0$, the best stone S_m is the one that maximize the following objective

$$\max_m \quad \omega_1 \mathrm{LLC}_m + \omega_2 \mathrm{NHR}_m, \tag{17a}$$

$$\text{s.t.} \quad m = 0, \dots, K_2. \tag{17b}$$

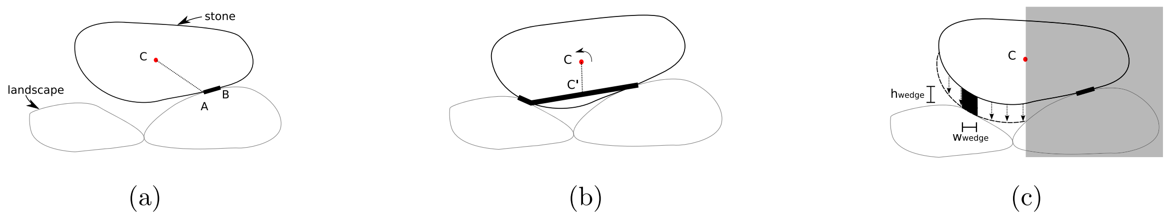


Fig. 9. (a) Unstable initial pose, (b) rotation-based method where the rotation direction is indicated by the arrow, and (c) the wedge method. The wedge stone (in black) located underneath the unsupported part of the stone to make it stable.

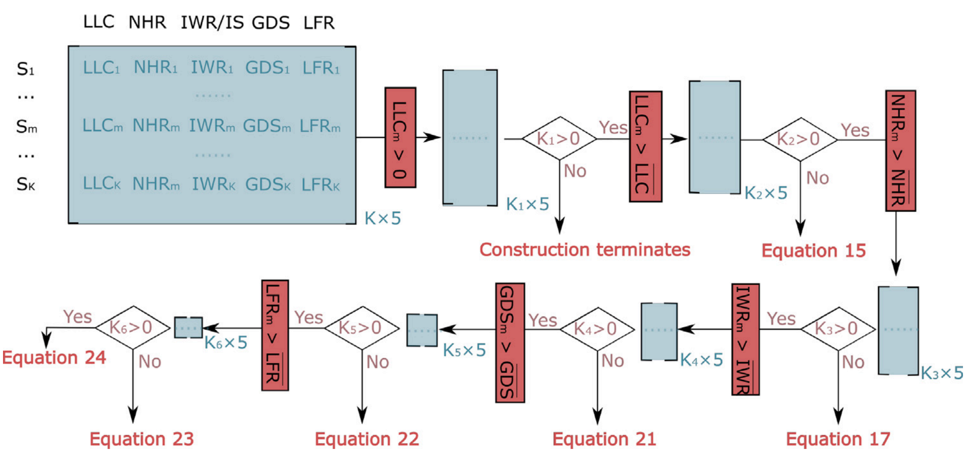


Fig. 10. Selecting the best stone S_{m*}.

Here the ω_1 and ω_2 are two weights for the two indices. This filtering process continues until all filters have been applied. The upcoming filters are on IWR, GDS and LFR sequentially, with the averaged values calculated as

$$\overline{\mathrm{IWR}} = \frac{1}{K_3}\sum_{m=1}^{K_3} \mathrm{IWR}(M_m, S_m), \tag{18a}$$

$$\overline{\mathrm{GDS}} = \frac{1}{K_4}\sum_{m=1}^{K_4} \mathrm{GDS}(M_m, S_m), \tag{18b}$$

$$\overline{\mathrm{LFR}} = \frac{1}{K_5}\sum_{m=1}^{K_5} \mathrm{IWR}(M_m, S_m). \tag{18c}$$

The same type of condition is verified for K_4, K_5 and K_6. If at any point a condition cannot be satisfied, the best stone is selected with the corresponding objective functions as in Eqs. (19)–(21). The whole process is summarized in Fig. 10.

$$\max_m \quad \omega_1 \mathrm{LLC}_m + \omega_2 \mathrm{NHR}_m + \omega_3 \mathrm{IWR}_m, \tag{19a}$$

$$\text{s.t.} \quad m = 0, \ldots, K_3. \tag{19b}$$

$$\max_m \quad \omega_1 \mathrm{LLC}_m + \omega_2 \mathrm{NHR}_m + \omega_3 \mathrm{IWR}_m + \omega_4 \mathrm{GDS}_m, \tag{20a}$$

$$\text{s.t.} \quad m = 0, \ldots, K_4. \tag{20b}$$

$$\max_m \quad \omega_1 \mathrm{LLC}_m + \omega_2 \mathrm{NHR}_m + \omega_3 \mathrm{IWR}_m + \omega_4 \mathrm{GDS}_m + \omega_5 \mathrm{LFR}_m, \tag{21a}$$

$$\text{s.t.} \quad m = 0, \ldots, K_5. \tag{21b}$$

When not specified, we adopt $\omega = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5) = (1, 1, 1, 1, 1)$ in the numerical examples presented in this paper. If all conditions are satisfied and all filters can be applied, the best stone is chosen with the following objective function

$$\max_m \quad \omega_1 \mathrm{LLC}_m + \omega_2 \mathrm{NHR}_m + \omega_3 \mathrm{IWR}_m + \omega_4 \mathrm{GDS}_m + \omega_5 \mathrm{LFR}_m, \tag{22a}$$

$$\text{s.t.} \quad m = 0, \ldots, K_6. \tag{22b}$$

This hierarchical filtering process builds a hierarchy of criteria, removes unsatisfactory actions, and selects from the remaining actions one that maximizes the weighted sum of certain indices. In this work, the filters indicate different criteria for building dry stone masonry wall, namely the stability under gravity, the stability under lateral load, the course straightness, the interlocking for inner stones, the inward sloping for boundary stones, the global filling and the local filling. In Section 5.1, one can find how the indices influence the filtering outcome and how it changes the layout of the built wall.

4. Kinematic analysis

The analysis of the resistance of masonry walls is performed with a kinematic analysis approach. In this analysis, the main goal is to find the maximal admissible load and the corresponding collapse mechanism. In this regard, the lateral resistance of the wall under construction is used to obtain the load factor α used in the evaluation index in Eq. (11). The method used in this paper is the kinematic analysis method proposed by Krabbenhoft et al. (2012) to simulate granular material, which was further adapted by Portioli (2020) and Portioli et al. (2021) to analyze masonry walls (see Section 4.1). A Pic-to-Ri algorithm is developed herein to convert the aggregate image into an input data for the kinematic analysis (see Section 4.2).

4.1. Kinematic analysis using mathematical programming

The numerical model employed in this scientific paper represents the application of the mathematical programming approach introduced by Portioli (2020) for the limit analysis of masonry walls. While Portioli et al.'s implementation is accessible as an executable in Matlab (Portioli, 2021), we reimplement the approach in Python to enhance computational efficiency. Fig. 11 shows a model of two stones modeled as rigid bodies in contact. The internal forces between them occur through contact points that are located at the two ends of the contact interface. In the approach, a four-point contact formulation is employed, which means that each contact interface has four contact points, as shown

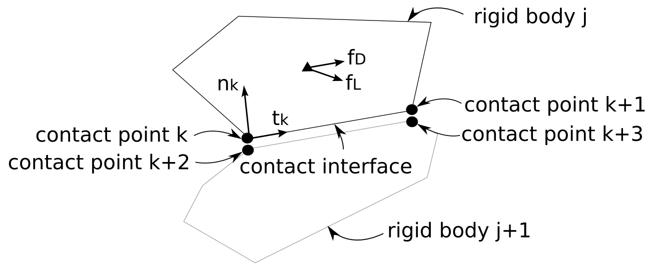


Fig. 11. Rigid block model for kinematic analysis where the positions of the two bodies are shifted for visualization purpose .
Source: Adapted from Portioli (2020).

in Fig. 11. Forces at the contact point k are collected in a vector c, including the normal c_{n_k} and the tangent c_{t_k} forces. The direction of c_{n_k} and c_{t_k} are denoted by two unit vectors n_k and t_k, respectively; as shown in Fig. 11. External loads, including the dead f_D and live f_L loads, are applied in the center of the elements. The problem of solving for the maximal load factor (i.e. load multiplier) is then formulated as a mathematical programming problem as follows (Portioli, 2020):

$$\max \quad \alpha, \tag{23a}$$

$$\text{s.t.} \quad E_k c = f_D + \alpha f_L, \tag{23b}$$

$$c \in \left\{ c_k \in R^2 : \mu c_{n_k} \geq \left\| c_{t_k} \right\|, c_{n_k} \geq 0 \right\}. \tag{23c}$$

This is a maximization problem with two constraints, one on the equilibrium condition (Eq. (23b)) and another on the failure criterion (Eq. (23c)). Eq. (23b) establishes the equilibrium of stones subjected to dead and live loads, with E_k being the equilibrium matrix, whose entries are determined from the location of contact points and element centers. Thus, one can write this matrix as (Portioli, 2020):

$$E_k = -\begin{bmatrix} t_k & n_k \\ (D_{jk} \times t_k)_y & (D_{jk} \times n_k)_y \end{bmatrix}, \tag{24}$$

where D_{jk} is the vector representing a line segment from the center of the element j to the contact point k. The failure at the interfaces (i.e., opening and sliding) is formulated in Eq. (23c). In particular, the failure due to sliding follows the Coulomb friction law, where μ represents the coefficient of friction. One can see that Eq. (23) is a linear optimization problem that can be easily solved using conventional methods such as the simplex method, which is used herein in its dual version (MOSEK, 2024). The kinematic variables, i.e. displacement of the elements, can be obtained from Lagrange multipliers associated to the solution of Eq. (23) (Portioli, 2020).

4.2. Pic-to-Ri: image-based generation of rigid block model

To run the previously described kinematic analysis, one should retrieve the information about the interface and contact points from geometrical models. In this section, a novel method, referred to as “Pic-to-Ri” is introduced herein for generating the rigid block model automatically from images. In this regard, the information about the mass and center of mass of the stones in a wall are recovered from its image. Moreover, the contact coordinates and directions are obtained from the detected interface between bodies. As an example, let us consider an aggregate of three stones with an initial landscape as shown in Fig. 12. Using the stacking algorithm presented in the previous section, we obtain Fig. 12a, where each stone consists of pixels that share the same value, and different stones have different pixel values. The area of each stone is estimated by counting its number of its pixels, and the center of mass (shown as blue triangles in Fig. 12c) is calculated as the geometric center of its pixels. Using convolution in the edge detection, we can detect the contact between stones (see Fig. 12b), where green arrows indicate the normal directions at each point of the detected interface. Next, only the end points of the interface are used to perform the kinematic analysis, where the normal direction (n_k) is estimated as the average of all normal vectors in a given interface (see Fig. 12c).

5. Experiments

The examples presented in this section are used to evaluate the performance of the developed methods using various stone data sets. We first use the topology of real stones extracted from a wall built by masons in the work of Almeida et al. (2021) to demonstrate the construction process. The intrinsic characteristics of the proposed algorithm are revealed by varying the algorithm parameters and sampling methods. Then we apply the algorithm to five sets of stones to verify the robustness and adaptability of the algorithm. To facilitate comparison between different walls, we also propose metrics to evaluate the quality of the dry joint masonry wall. Furthermore, we investigate the computational performance of the developed algorithm and its sensitivity to image resolution.

5.1. Building a dry-joint stone masonry wall

The demonstration consists of building a wall with a set of stones extracted from the image of the wall built with partially regular stones in Almeida et al. (2021). Fig. 13 illustrates the original wall built by masons, where stones are colored by the clustering result. The stones shown in Fig. 13(a) and their arrangement are directly annotated based on the photograph of the constructed wall’s facade. The gaps between the stones arise due to their lack of contact at the facade plane and annotation errors. To reduce the gap, we reposition the stones such that the distance between their centers and the bottom left corner is minimized. The manipulation is performed sequentially, prioritizing stones with smaller distance to the bottom left corner. The stones are also dilated by one pixel for the same reason. The optimized layout is shown in Fig. 13(b). In the following work, we use stones from direct annotation as the initial stone data set for the algorithm, but the evaluation of limit multiplier of the reference wall is conducted based on the optimized wall.

To build a similar wall using our proposed algorithm, we assume the initial landscape to be flat. As a base case, we use the sampling strategy described in Section 3.4. The weights used in the selection algorithm (Fig. 10) are $\omega = (1, 1, 1, 1, 1)$. The constructed wall is shown in Fig. 14, along with the action order denoted by the numbers written on each stone. The colors of the stones show the cluster to which they belong. It can be seen that the wall is generally constructed from left to right. This is imposed by the placement optimization formulation (Eq. (12a)). The right side is also filled as a result of using GDS in Eq. (12a). The wall displays geometric features such as straight course height and interlocking. However, the layout is more regular on the left side of the wall, which shows the difficulty in placing stones on the right boundary. This results from the fact that the right boundary is filled in the final few steps, when the availability of stones has been reduced to the point that the algorithm cannot find alternatives. The load multipliers of the built wall are 0.79 and 0.71 for tilting the table towards the left and right, with the failure mechanism shown in Fig. 15. The vertical displacement of the sliding stone in Fig. 15(a) results from the associative flow rule in the kinematics analysis, which causes artificial dilation in the sliding failure mode (Portioli, 2020).

To investigate the influence of the sampling method on the construction process, we apply three additional sampling strategies that differ from the variant-stone sampling method described in Section 3.4 and the results are shown in Fig. 16. The numbers on the stones indicate the action index, and the colors show the cluster to which the stone belongs to. Fig. 16a is a wall constructed with a random sampling strategy, where the candidate set in each construction steps is a subset of the whole set of available stones sampling from an uniform probability

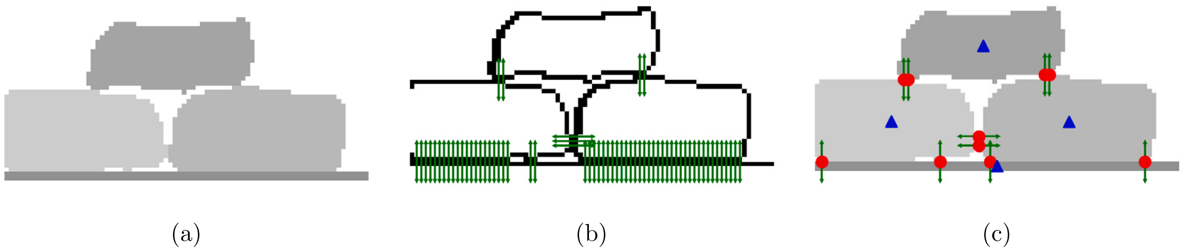


Fig. 12. Pic-to-Ri method: automated rigid block modeling from image data. (a) Segmented image of a constructed wall. (b) Detected contours and interface directions. (c) Rigid bodies and contact points for kinematic analysis.

Fig. 13. A masonry wall built by masons with stones clustered into seven classes. (a) original wall (b) optimized wall with stones repositioned and dilated by one pixel. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

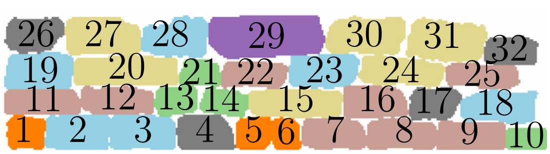


Fig. 14. Dry-stone masonry wall constructed with the algorithm developed herein. The numbers represent the stones placement sequence and the colors denotes their respective cluster. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

distribution. To make a fair comparison between the strategies, we sample 7 candidate stones in each iteration. The wall shown in Fig. 16b was built with a two-fold variant-stone sampling method, where at each construction step, we first use the basic variant-stone sampling method to determine the most appropriate cluster of stones, and then all stones from that cluster are considered as the candidate set from where the best stone is selected from. On the other hand, the wall in Fig. 16c was built with the whole stone data set as candidate set in each construction step, with the placed stones being removed in each step (i.e. full batch method, the candidate set equals the remaining data set).

For all sampling methods, the algorithm is able to complete the construction process by stacking stones to fill the target wall volume. However, the uniform random sampling method yields a more irregular stone layout (Fig. 16a) compared to strategic sampling methods (Fig. 14 and Fig. 16b), especially on the boundaries and at the last stage of the construction process corresponding to the fourth course of the wall. The two-fold variant-stone sampling method (Fig. 16b) is an enhancement to the basic variant-stone sampling method (Fig. 14) and the obtained wall has better geometric features, such as the filling on the right-hand boundary and the straightness of course height. However, this advanced method is twice as computationally expensive as the base method because it requires two sub-iterations at each construction step. The wall built with non-replacing full-batch sampling (Fig. 16c) is better in terms of straight course height, especially in the first three horizontal layers. The last layer is less satisfactory as stones are used up and there is less diversity in the candidate stone set.

5.2. Quantitative evaluation of dry joint stone masonry wall

In this section, we build walls with five different stone data sets. The data sets are obtained through manual segmentation of stones from walls images built either by masons or by a reference stacking algorithm. The original walls are used as reference walls for the result obtained with the algorithm developed herein. The first three data sets are composed by the geometry of stones retrieved from images in the work of Almeida et al. (2021), where skilled masons built three types of stone walls (regular (R), partially regular (PR) and irregular (IR)) in an experimental campaign. The fourth and fifth data sets are extracted from dry joint masonry walls built by a well-trained deep reinforcement learning agent (Liu et al., 2018).

To facilitate the comparison between different configurations, we use the following metrics to evaluate the quality of a stone masonry wall. The stone filling to quantify the filling of the target wall volume by comparing the sum of stone areas with the target wall area (Almeida et al., 2016) as given by

$$F_{SF}\% = \frac{\sum A_{S_i}}{w_{wall} \times h_{wall}}, \tag{25}$$

where A_{S_i} is the stone area S_i, and w_{wall} and h_{wall} are the width and height of the wall. The horizontal arrangement to quantify the straightness of the courses using a horizontal alignment factor (Almeida et al., 2016). The factor is calculated by comparing the average length of n horizontal shortest paths along the joints (i.e., not crossing stones, example of paths are shown in Fig. 17a) to the width of the target wall. This metric is written as

$$F_{AH}\% = \frac{1}{n}\sum_{i=1}^{n}\frac{h_i - w_{wall}}{w_{wall}}, \tag{26}$$

where h_i is the length of the i shortest horizontal path along the joints. A lower value of F_{AH} indicates a straighter course height. Another metric is the vertical arrangement to quantify the interlocking between vertical joints. We compare the average length of n vertical shortest paths along the joints with the height of the wall by estimating the following factor (Almeida et al., 2016)

$$F_{AV}\% = \frac{1}{n}\sum_{i=1}^{n}\frac{v_i - h_{wall}}{h_{wall}}, \tag{27}$$

where v_i is the length of the shortest vertical path i. An example of the shortest vertical paths of a dry-joint wall is shown in Fig. 17b. In this regard, a high value of F_{AV} indicates better interlocking. Further, the lateral resistance is evaluated by simulating the tilting table test using the kinematic analysis described in Section 4. We analyze two

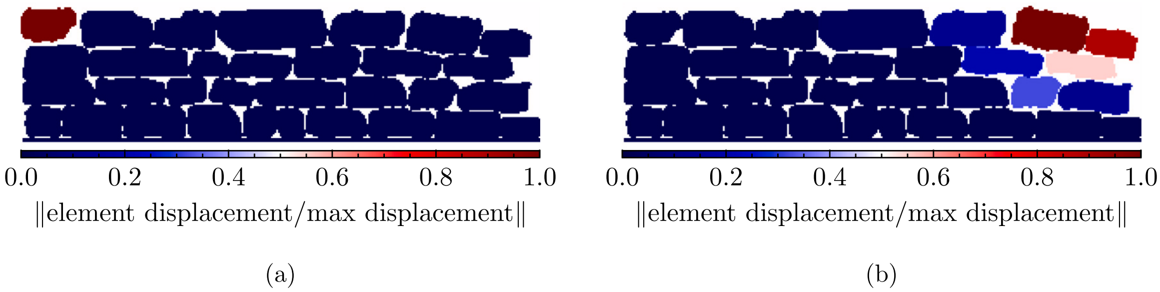


Fig. 15. Failure mechanism of constructed PR wall in a tilting table simulation for a (a) horizontal force direction in the left, and for an (b) horizontal force direction in the right. Elements are colored by the ratio of their displacement to the maximum displacement in the model. The observed vertical displacement of the sliding stone in (a) is a consequence of artificially dilated sliding surfaces when employing the associative flow rule (Portioli, 2020). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

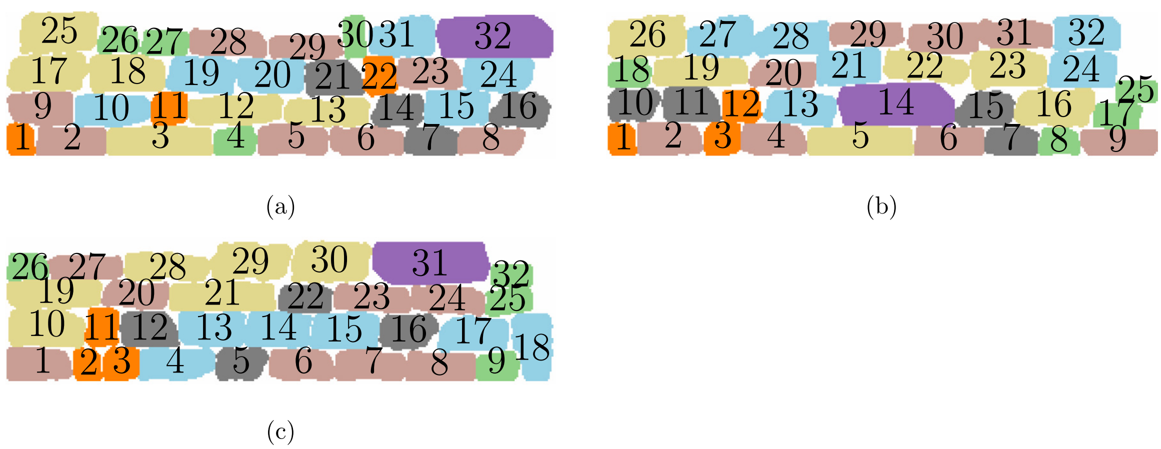


Fig. 16. Constructed wall using different sampling method to form candidate stone set: (a) uniform sample; (b) two-fold sample; (c) full batch sample. The numbers represent the construction sequence, the colors represent the cluster. Action index are denoted at the center of each stone. Stones of the same cluster is in the same color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

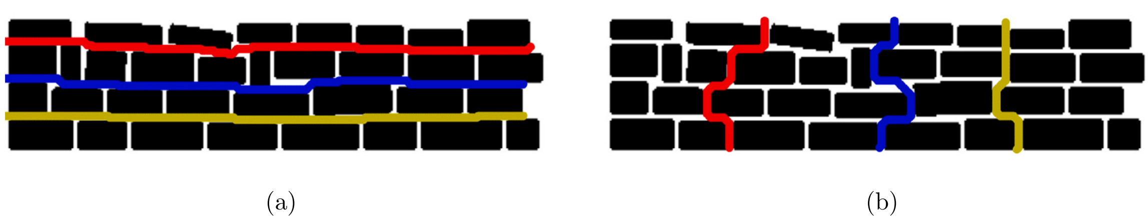


Fig. 17. (a) Three horizontal shortest paths of dry-joint stone masonry wall to calculate F_{AH}. (b) Seven vertical shortest paths of dry-joint stone masonry wall to calculate F_{AV}.

load cases (tilting towards the left and right)and use the smaller load multiplier in the evaluation.

$$F_{LR} = \frac{\min(\alpha_{left}, \alpha_{right})}{\mu}. \tag{28}$$

Fig. 18 shows the wall constructed using the five data sets. The first column of Fig. 18 shows the case number, the second column shows the clustering of the input stone data set. As the stacking algorithm is not deterministic, we first use a fixed seed to perform the construction. The resulting walls are shown in the third column. Then a total of 20 trials are performed for each set of stones. Subsequently, the wall with the highest filling ratio and kinematics multiplier(rounded to one decimal place) is selected from these trials and presented in the fourth column of the results. The last column is the reference wall built either by masons or by a reference algorithm. The stone layouts of the reference walls are optimized through stone repositioning and dilation as explained in Section 5.1.

It can be seen that the algorithm successfully build walls with available stones in all cases, while keeping masonry features such as straight course height and interlocking. On the other hand, case 3 and case 5 were more challenging because the input stone set is widely distributed in the size-eccentricity space. Therefore, the developed stacking algorithm has to use wedge stones to keep the stability of the larger ones, as shown in Fig. 19. Fig. 20 illustrates the stone filling ratio achieved and the area of wedging stones used through various stabilization methods. Each data point represents a wall constructed with the fifth set of stones. The results indicate that combining rotation and wedging techniques enhances the algorithm's performance in optimizing the stone filling ratio. Further, one can also observe that stones from the same cluster are placed in a near neighborhood in all cases in Fig. 18. This effect can be attributed to the NHR metric that imposes straight course height. Stones from the same cluster of similar size and eccentricity are close in terms of height. Thus, placing them together can yield a flat horizontal layer.

However, we can notice once again that filling the right side of the wall is a difficulty task. The algorithm can only place stones vertically to fill the empty spaces as in case 1, in case 3 and in case 5, or leave

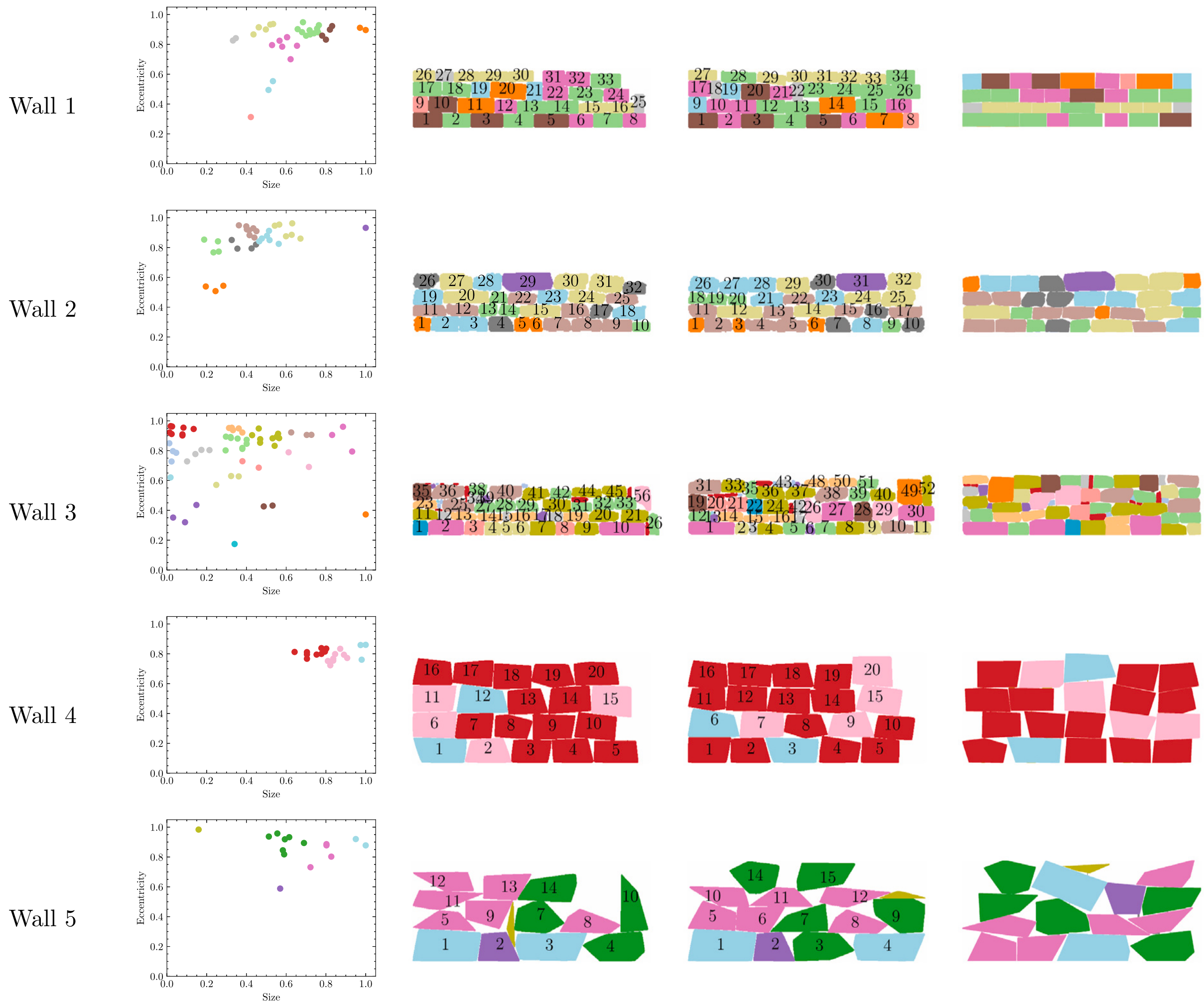


Fig. 18. Stone masonry walls built with five stone data set. The colors represent the cluster, the numbers show the sequence of actions. Column from left to right: Case number, clustering of the input stone data set, constructed stone wall with a fixed seed, best constructed wall out of 20 trials, and reference wall built by masons (Almeida et al., 2021) (case 1–3) and by model-free algorithm (Liu et al., 2018) (case 4 and 5). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(a) (b)

Fig. 19. (a) Wedge stones (in red) used in Fig. 18(a) wall 3, (b) wall 5. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

it empty as in case 2 and in case 4. Such challenge can be solved by giving the algorithm possibility to place stone from right to left as well. More specifically, this refers to reverting the sign of the third part on the center of the stone in Eq. (12a) (i.e. changing from minimizing to maximizing), using the right neighbor as reference stone in Eq. (7) and to compute the LFR with a local area determined by rays pointing to the positive direction of x. With the best action assuming both a construction from the left and from right, and chosen as the one with higher GDS such that

$$S_m *= \underset{S_{m*,left},S_{m*,right}}{\operatorname{argmax}} \ \mathrm{GDS} \tag{29}$$

Fig. 21 shows the wall when the stones of one layer are placed starting from the right and the left with the regular, partially regular, and irregular stone data sets. It can be seen that the stones are aligned with both the left and right boundaries. The weaker region of the wall in terms of geometrical misalignment and voids shifts from the right boundary to the middle part where the construction sequences from the left and right boundaries meet. For example, we observe voids on the right of stone 16 in Fig. 21a, on the left of stone 16 in Fig. 21b, and below stone 37 in Fig. 21c.

A comparison between the reconstructed and the reference walls is performed by using the metrics presented in Section 5.2. The obtained

Table 2
Comparison of metrics of and mason's walls and rebuilt walls.

	Reference walls (Fig. 18 the 5th column)				Constructed walls (Fig. 18 the 4th column)			
	F_{SF}	F_{AV}	F_{AH}	F_{LR}^{*}	F_{SF}	F_{AV}	F_{AH}	F_{LR}
Comparison with masons (Almeida et al., 2021)								
Wall 1	0.837	**39.780**	**0.000**	0.64	**0.974**	39.084	1.573	**0.83**
Wall 2	0.883	**40.810**	**0.830**	0.48	**0.883**	36.598	1.348	**0.83**
Wall 3	**0.957**	**22.410**	**3.440**	0.26	0.839	21.347	4.163	**0.52**
Comparison with model-free algorithm (Liu et al., 2018)								
Wall 4	0.800	6.765	**1.707**	0.60	**0.839**	**22.674**	2.207	**0.66**
Wall 5	**0.753**	15.783	6.767	0.05	0.725	**55.851**	**3.033**	**0.59**

*: The kinematic analysis on the reference walls are based on our annotations, which could be different from the actual performance of the wall.

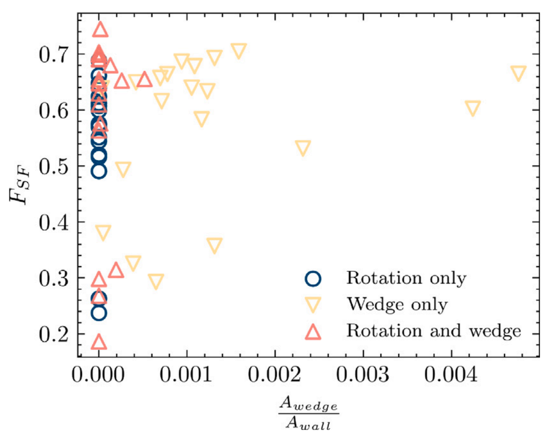


Fig. 20. Influence of the stabilization method on the stone filling ratio and area of wedge stones in the construction of Wall 5.

results are listed in Table 2, with the best scores shown in bold. Compared with the masons' performance, the proposed algorithm performs better in terms of filling the space (evaluated by F_{SF}) and in terms of lateral resistance (F_{LR}). However, it presents an reduced performance in terms of maximizing vertical interlocking (evaluated by F_{AV}) and arranging horizontal alignment (F_{AH}). Masons can build walls with very straight course height (small value of F_{AH}) and significant interlocking (large value of F_{AV}), possibly because they are more flexible in shaping stones and using wedge stones. Such flexibility gives them the possibility to create smooth supporting surfaces and match the stone's height with neighboring stones. When compared with the model-free algorithm (Liu et al., 2018), our algorithm outperforms it in almost all evaluations.

The versatility and flexibility of the algorithm can be attributed to two factors. On the one hand, the strategic sampling process that cluster stones and sample them gradually from the initial data set ensures the diversity of the candidate stone set, which is the key to maintaining high-quality geometric wall features throughout the whole construction process. On the other hand, we consider mason's practice by proposing eight indices to evaluate the quality of an action (choosing and placing a stone). These metrics consider the neighboring stones and the current landscape, providing a global view to assess a stone placement.

5.3. Influence of stone set on the constructed wall

The size, shape and distribution of stones in the given stone set influence the performance of the algorithm. To study this influence, we used the algorithm to build walls from two series of stone sets. The algorithm is allowed to stack stones from two sides, with the method explained in the previous section. The first stone series comes from Thangavelu et al. (2018), where stones are represented as polygon shapes with eight degrees of Gaussian noise applied to their vertices, resulting in a diverse range of sizes and shapes. For each set of stones, we built 10 walls and Fig. 22 presents an example from each set. The metrics of stone filling and lateral resistance of all 80 walls are shown in Fig. 23. The stone set with an 80% noise level is the most difficult, resulting in walls that were both unfilled and weak. As the noise level decreases, there is an uptrend in stone filling ratios. However, the trend is not linear as the algorithm performs surprisingly well in the construction with stone set with a 70% noise level. With noise level smaller than 60%, the algorithm consistently achieves stone filling ratios above 60%, outperforming the random strategy outlined in Thangavelu et al. (2018). The negative influence of noise level on the filling ratio of constructed walls can also be observed in Fig. 24(a). Our algorithm is able to fill the wall volume as much as the reference, despite the fact that the current task is more challenging compared to the one in the reference study (Thangavelu et al., 2018)—the stone set provided to our algorithm is limited to the ones from the resulting walls in that study without additional options.

Regarding lateral resistance shown in Fig. 24(b), there is no clear trend associated with changes in noise level. As we evaluate lateral resistance through push-over tests on the free-standing walls, the failure mode in most cases is the rocking of the corner stones on the top row, similar to the mode shown in Fig. 15(a), which depends on the stone's height-to-width ratio. Although variations in noise level can affect this ratio, the influence is neither positive nor negative because increasing the ratio in one orientation will conversely decrease it when the stone is rotated by 90 degrees. The fluctuation is also observed by analyzing the reference walls, as shown in Fig. 24(b). Nonetheless, higher noise levels do make it more challenging for the algorithm to find a stable position for the stones, as they are less similar to rectangular shapes. In this case, the best pose is not necessarily the one where the shape factor is minimized, explaining why the filling ratio decreases as the noise level increases. By considering more orientation options, such as orientations ranging between $[-\pi, \pi]$ with 10-degree intervals, the performance of the algorithm can significantly increase in terms of the filling ratio and the lateral resistance. Typical walls constructed with 10-degree interval orientations are shown in Fig. 25.

When comparing the interlocking metric F_{AV} of walls built using stone sets of different noise levels, we can observe a positive influence in both our walls and the reference walls, as illustrated in Fig. 24(c). This is because diverse stone shapes naturally lead to staggered joints when the stones are packed tightly. The vertical joints of our walls are better staggered due to our precise method for evaluating the distances between these joints (IWR), in contrast to the reference which uses an approximate method that equates vertical interlocking with maximizing the number of contacting stones (Thangavelu et al., 2018). As for the horizontality metric (Fig. 24(d)), their is no obvious trend in how noise level impacts this metric. Although stones in a low noise level set have similar heights and widths, the algorithm may rotate stones for interlocking, which causes misalignments in the course height.

The second series of stones is generated from a Fourier-based shape generator for granular materials, as described in Mollon and Zhao (2012). These stones exhibit a more rounded appearance in contrast

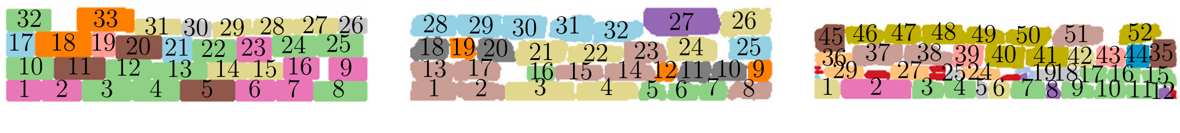


Fig. 21. Walls constructed from two sides. (a) Wall built with regular stones, (b) Wall built with partially regular stones, and (c) Wall built with irregular stones.

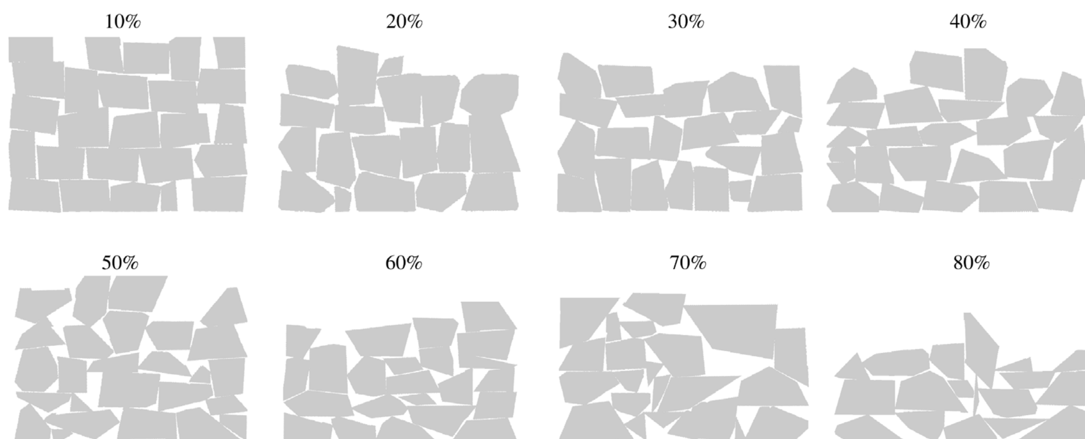


Fig. 22. Walls constructed with our algorithm using the stones from Thangavelu et al. (2018). The noise level of each stone set is indicated on top of the wall.

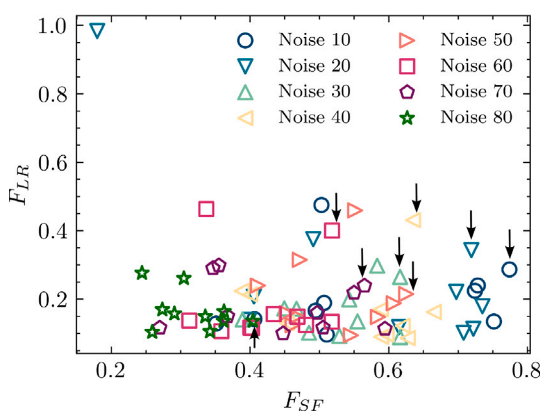


Fig. 23. Filling ratio F_{SF} and stability F_{LR} of walls constructed with data set of Thangavelu et al. (2018). The arrows point to walls shown in Fig. 22.

to the ones from Thangavelu et al. (2018). The contour of a particle is generated through the following series (Mollon & Zhao, 2012):

$$r_i(\theta_i) = r_0 + \sum_{n=1}^{N} [A_n \cos(n\theta) + B_n \sin(n\theta)] \tag{30a}$$

$$A_n = r_0 D_n \cdot \cos \delta_n \tag{30b}$$

$$B_n = r_0 D_n \cdot \sin \delta_n \tag{30c}$$

$$D_n = \begin{cases} 2^{\alpha \cdot \log_2(n/3) + \log_2(D_3)} & \text{for } 3 < n < 8 \\ 2^{\beta \cdot \log_2(n/8) + \log_2(D_8)} & \text{for } n > 8 \end{cases} \tag{30d}$$

Here (r_i, θ_i) denotes the polar coordinates of points on the contour, which in this study is discretized into 128 points. r_0 corresponds to the average radius of the particle. The second descriptor D_2 has an impact on the elongation of particles generated. The value for other descriptors are $D_1 = 0$, $D_3 = 0.05$ and $D_8 = 0.015$, with their specific influences on the particle's form detailed in Mollon and Zhao (2012). δ_n is a random phase angle between $[-\pi, \pi]$ that gives particles different shapes. The coefficients α and β are assigned a value of -2, same as in the reference.

We generate two types of stones by varying the value of D_2. $D_2 = 0$ results in circular stones while $D_2 = 0.2$ gives elliptic stones. Additionally, we explore two different stone size distributions: one with a constant radius of 30 pixels and another with a radius uniformly varying between 10 and 50 pixels. With these parameters, we form four sets of stones and construct 20 walls, each measuring 210 pixels in height and width. Typical walls constructed are shown in Fig. 26 and the filling ratio and interlocking metric of all walls are illustrated in Fig. 27.

It can be seen that the algorithm performs poor in filling the wall using stones of a fixed size. This is due to its strategy of prioritizing stone placement near the boundary, as in Eq. (12), which causes instability when stacking round-shaped particles. In contrast, when the algorithm is provided with stones of varying sizes, it can more easily achieve stable stacking by either placing a smaller stone atop a larger one or positioning a larger stone over two smaller ones, both of which yield more stable structures than stacking stones of uniform size. Moreover, the variation in stone size enhances interlocking, which is also observed in the previous analysis where F_{AV} increases with noise level.

The evaluation of the two stone series demonstrates that the performance of the algorithm is significantly influenced by the variety in stone sizes and shapes. A stone set with limited variability might be

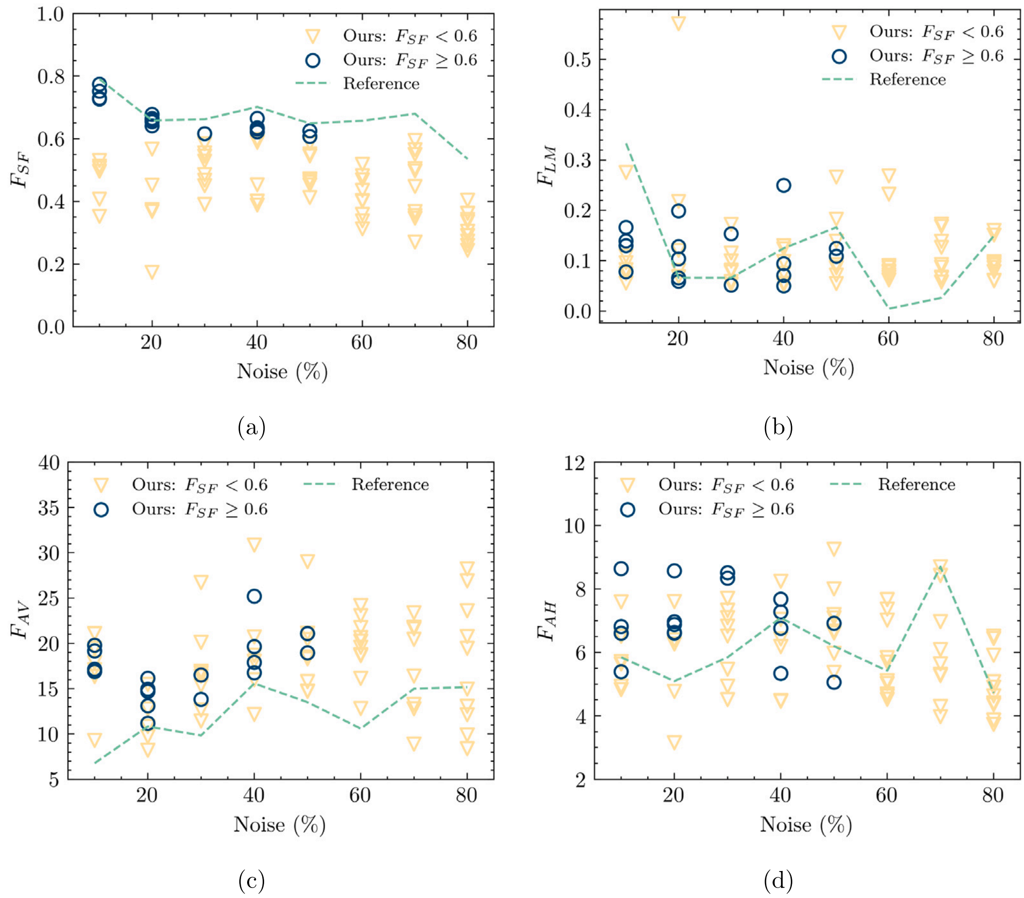


Fig. 24. Influence of noise level of stone set on the metrics of walls constructed with our algorithm. Walls with stone filling ratio larger than 0.6 are indicated with blue points and those smaller than 0.6 are indicated with yellow triangles. We also evaluate the walls provided in Thangavelu et al. (2018) for reference (green dashed line). (a) Stone filling ratio F_{SF}; (b) Lateral resistance F_{LR}; (c) Vertical interlocking F_{AV}; (d) Horizontality F_{AH}.

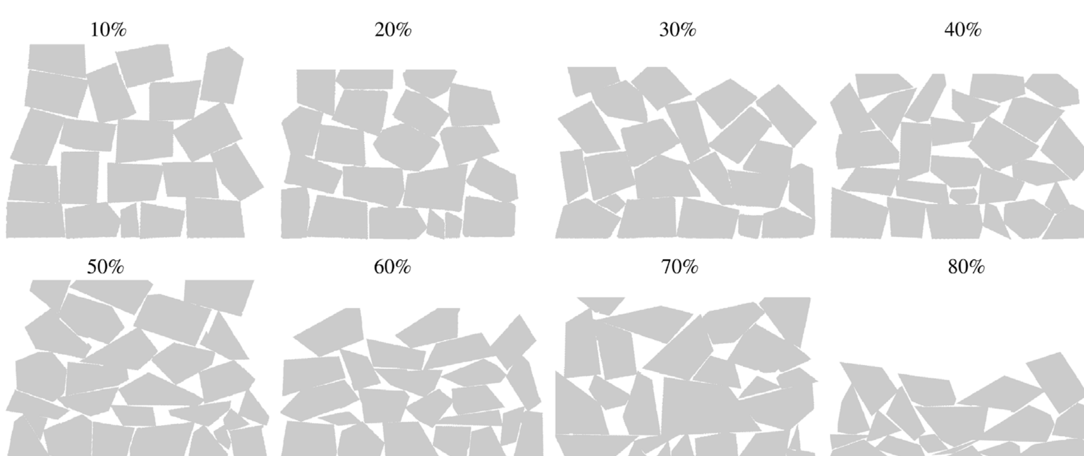


Fig. 25. Typical walls constructed with stone orientations ranging between $[-\pi, \pi]$ with 10-degree intervals.

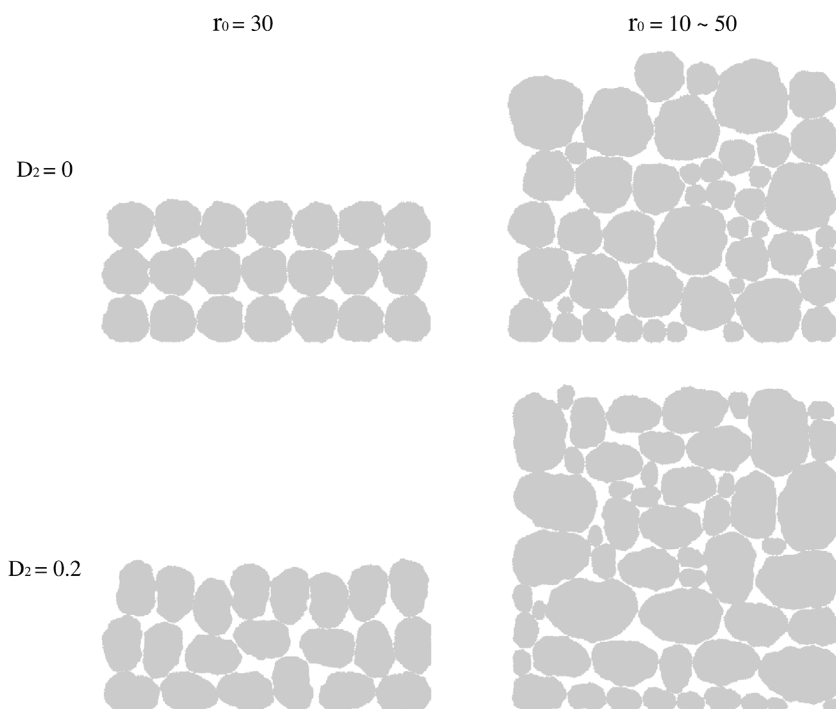


Fig. 26. Walls built with stone set generated by prescribed Fourier descriptors.

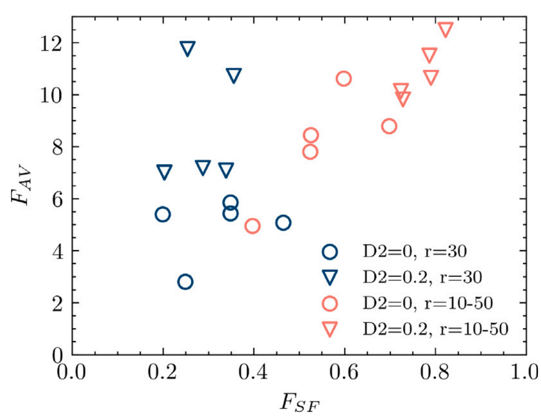


Fig. 27. Geometric metrics of walls constructed from stone set generated by prescribed Fourier descriptors.

easy to stack and the algorithm can achieve a high filling ratio, but it often results in walls with inferior interlocking, particularly when the stones' height and width are nearly equal. The algorithm benefit from the diversity in stone types, whether they are round or rectangular. However, it struggles in scenarios where the stones are uniform circular shapes or prismatic stones with acute angles, as these configurations challenge the assumption that the target wall has uniform width along height and the optimal pose of a stone is when the shape factor is minimized.

6. Computational performance

As the proposed algorithm aims at its application in automated construction, the computational cost is one of the main concerns. This section investigates computational performance by estimating the processing time of each component of the developed algorithm to identifying potential bottlenecks. The sensitivity of image resolution on the layout of the constructed wall and associated computational cost are also discussed. The examples presented next were run on Intel Core (TM) i7-10700 CPUs with 2.90 GHz of clock.

6.1. Computational cost for single procedures

One can clearly see that the computational demand to find the ultimate layout of a masonry wall with the algorithm presented in this paper depends on the size of the available stone set and the employed sampling method. In this section, we analyze the processing time to built two walls as shown in Fig. 14 and in Fig. 16c. The input and output image sizes of the two examples are equal to 280 × 80 pixels. The computational cost of other walls using other stone set and construction method can be found in Appendix B. To further improve the computational performance of the stacking algorithm, the stone trial task are performed in parallel with a shared memory scheme.

Fig. 28 shows the relation between the sequence of 32 stone placements and the associated computational cost for both considered walls. The first wall is constructed with basic variant sampling method, where in each iteration the candidate set is composed of one stone from each cluster; whereas the second is constructed with full batch sampling method, where all available stones are sampled as candidate stones in each step. Further, three procedures are considered for each step, namely the trial placement of candidate stones (see Section 3.6), the stabilization of placed candidate stones (see Section 3.7), and the selection of the best stone (see Section 3.8). The time spent on each procedure is colored in blue, green and orange respectively.

One can observe that using variant sampling method significantly reduces computational cost in comparison to the processing time to perform a full batch sampling, but without sacrificing the quality of

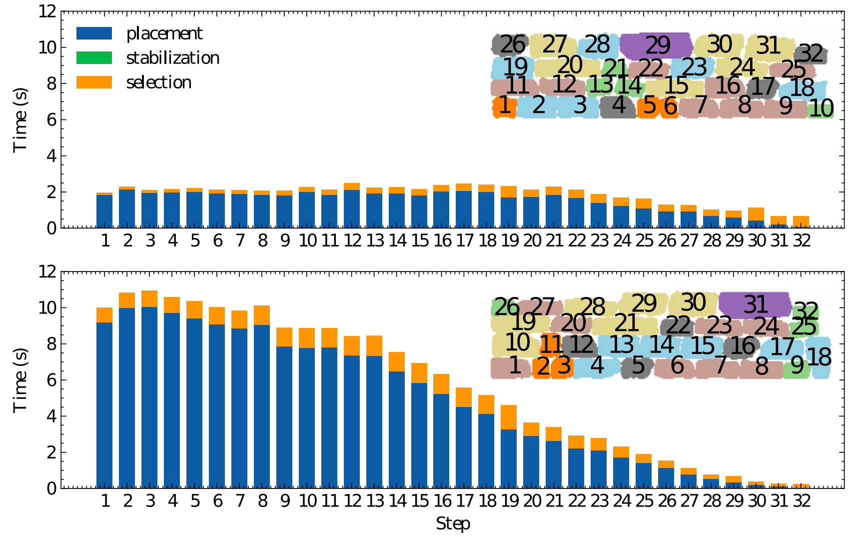


Fig. 28. Computation time for each construction step. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

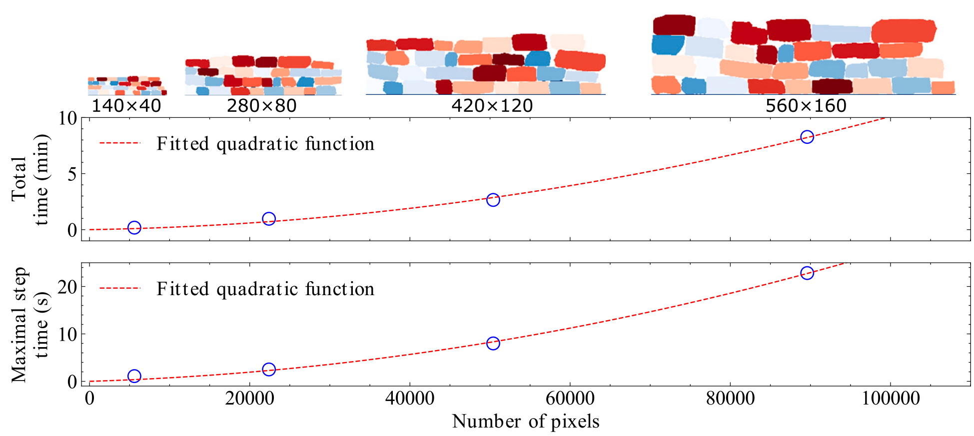


Fig. 29. Evolution of computation cost with image size. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the built wall. For both cases, the placement procedure takes most of the time, the selection ranks second, and the computational demand in the stabilization procedure is negligible. During construction, the computation cost of one step is almost stable for variant sampling method as the number of candidate stone set is unchanged. The time for the selection procedure (shown in orange in Fig. 28) increases slightly as the construction continues. This is because kinematic analysis cost most of the time in the process. As more stones are placed, there are more rigid bodies and contact points in the model, leading to a higher computation cost. But for full sampling method, the computational time decreases as the size of the candidate set decreases.

6.2. Influence of data set size on time complexity

The algorithm's time complexity is influenced by the number of stones (Q), the number of clusters (K), and the number of placed stones (less than Q). The key contributors to the overall complexity include:

1. Initial Rotation: Optimizing the orientation of each stone to minimize the shape factor, adding a complexity of $O(Q)$.
2. Clustering: Applying a hierarchical clustering algorithm with a time complexity of $O(Q^2 \log Q)$ (Ward, 1963).
3. Placement, Stabilization, and Selection: These involve a discrete search for feasible positions and stabilization for each stone candidate, with the selection based on individual stone evaluations. The strategic candidate sampling method used here results in a time complexity of $O(K)$, whereas the full batch sampling method has a complexity of $O(Q)$. It is important to note that these processes are repeated in every step. Therefore, for the entire wall construction, the time complexity using variant sampling is $O(QK)$, and using full batch sampling is $O(Q^2)$.

Given that placement is the most costly process (as demonstrated in Section 6.1), it dominates the algorithm's overall complexity, which is $O(QK)$ for the variant sampling method and $O(Q^2)$ for the full batch sampling method. Therefore, employing a strategic sampling method can significantly reduce the computational cost, especially for large data set with stones of similar shapes and sizes.

6.3. Sensitivity to image resolution

Since the stacking algorithm developed in this paper is based on images, changes in image resolution are likely to affect its performance. For this analysis, we consider the input stone data set in Section 5.1 with different image resolutions scaled by $0.5, 1.0, 1.5$ and 2.0. The desired wall sizes are also scaled accordingly. Fig. 29 shows the constructed wall in various resolutions. Each stone is assigned to a unique color (regardless of the image resolution) to identify the same stones in different resolutions. The difference in stone layout results from the non-deterministic nature of the algorithm. The computation time for constructing the whole wall and the maximal step time during construction are also illustrated in Fig. 29. It can be seen that there is a quadratic relationship between the resolution scale and the computation cost. It happens because both the number of possible positions in the trial placement and the time to evaluate one position increase as the image becomes larger. When the image is of size 140 × 40, it only takes 0.18 min to stack 32 stones. Even when considering the highest resolution, the maximum step time remains below 25 s, demonstrating a satisfactory level of speed for real-time automated construction. This is particularly significant considering that the robot manipulation to grip and transport one stone typically takes approximately 1 min (Johns et al., 2020).

7. Conclusions

This paper proposed an stacking algorithm based on imaging processing techniques and efficient kinematic analysis for the optimal placement of irregular stones in the construction of dry joint stone masonry walls. In this regard, images of stones were used as an input data set, and the developed algorithm provided the order of placement, pose, and position of each stone used in the construction of the wall, as well as the rigid block model used to assess its structural performance. The developed algorithm was able to build qualified walls in different scenarios with various stone data set and wall size. When compared with skilled masons, our algorithm was better in space filling and maximizing the lateral resistance of the wall, while maintaining comparable geometric features in terms of course straightness and vertical interlocking. We also outperformed the model-free method (Liu et al., 2018) in almost every respect.

By arranging stones of various shapes and sizes, we demonstrated the algorithm's effectiveness in handling rectangular stones. With rounded stones, the algorithm successfully attained dense and stable configurations when the stones were of diverse sizes. However, it was less effective for stacking stones with acute angles due to the study's methodology in optimizing stone orientation by minimizing the shape factor. Although considering more orientation angles improved the metrics, it also led to higher computational costs.

The analysis on the computation cost of the algorithm showed that it can be used in real-time robotic constructions, capable of planning a 32-stone wall in 10 s. Reducing the image resolution can efficiently reduce the computation cost. Further, we observed that the computational time was proportional to the size of candidate stone set. Using parallel computing, solving the best position of each candidate stone on separate processors, can accelerate the process and allow the use of large candidate set.

An immediate extension of this research will concentrate on the practical implementation of the proposed method for physical construction. In this regard, objects possessing a uniform section shape can serve as suitable candidates as 2D stones. The physical actions can be carried out using robotic arms (Johns et al., 2020) or by individuals equipped with augmented reality (AR) facilities (Settimi et al., 2023). Furthermore, there is potential for further advancement of the current algorithm, including the incorporation of 3D stones, planning mortar-joint stone masonry walls, and advanced wedging algorithms.

Data availability

The codes and data set are available on Github (https://github.com/eesd-epfl/StablePacking-2D/).

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Algorithms

Algorithm 2: Selecting the best action

Input : $\Pi_i = \{(M_1, S_1)\ (M_1, S_1)\ \dots (M_m, S_m)\}$, L_i, ω^T
Output: $(M_{m*},\ S_{m*})$

begin
  $\Pi_1 = \{(M,S) \mid \mathrm{LLC}_{(M,S)} > 0, (M,S) \in \Pi_i\}$;
  if $\Pi_1 = \emptyset$ **then**
    return Construction Terminates;
  else
    $\Pi_2 = \{(M,S) \mid \mathrm{LLC}_{(M,S)} \geq \overline{\mathrm{LLC}}(\Pi_1), (M,S) \in \Pi_1\}$;
    if $\Pi_2 = \emptyset$ **then**
      $(M_{m*}, S_{m*}) = \operatorname{argmax}\ \ \omega^T e^1 \mathrm{LLC}_{(M,S)}$;
    else
      $\Pi_3 = \{(M,S) \mid \mathrm{NHR}_{(M,S)} \geq \overline{\mathrm{NHR}}(\Pi_2), (M,S) \in \Pi_2\}\}$;
      if $\Pi_3 = \emptyset$ **then**
        $(M_{m*}, S_{m*}) = \operatorname{argmax}\ \ \omega^T (e^1 \mathrm{LLC}_{(M,S)} + e^2 \mathrm{NHR}_{(M,S)})$;
      else
        $\Pi_4 = \{(M,S) \mid \mathrm{IWR}_{(M,S)} \geq \overline{\mathrm{IWR}}(\Pi_3), (M,S) \in \Pi_3\ MS$ being inner stone$\} \cap \{(M,S) \mid \mathrm{IS}_{(M,S)} \geq \overline{\mathrm{IS}}(\Pi_3), (M,S) \in \Pi_3\}, MS$ being boundary stone$\}$;
        if $\Pi_4 = \emptyset$ **then**
          $(M_{m*}, S_{m*}) = \operatorname{argmax}\ \ \omega^T (e^1 \mathrm{LLC}_{(M,S)} + e^2 \mathrm{NHR}_{(M,S)} + e^3 (\mathrm{IWR}_{(M,S)} / \mathrm{IS}_{(M,S)}))$;
        else
          $\Pi_5 = \{(M,S) \mid \mathrm{GDS}_{(M,S)} \geq \overline{\mathrm{GDS}}(\Pi_4), (M,S) \in \Pi_4\}\}$;
          if $\Pi_5 = \emptyset$ **then**
            $(M_{m*}, S_{m*}) = \operatorname{argmax}\ \ \omega^T (e^1 \mathrm{LLC}_{(M,S)} + e^2 \mathrm{NHR}_{(M,S)} + e^3 \mathrm{IWR}_{(M,S)} + e^4 \mathrm{GDS}_{(M,S)})$;
          else
            $\Pi_6 = \{(M,S) \mid \mathrm{LFR}_{(M,S)} \geq \overline{\mathrm{LFR}}(\Pi_5), (M,S) \in \Pi_5\}\}$;
            if $\Pi_6 = \emptyset$ **then**
              $(M_{m*}, S_{m*}) = \operatorname{argmax}\ \ \omega^T (e^1 \mathrm{LLC}_{(M,S)} + e^2 \mathrm{NHR}_{(M,S)} + e^3 \mathrm{IWR}_{(M,S)} + e^4 \mathrm{GDS}_{(M,S)} + e^5 \mathrm{LFR}_{(M,S)})$;
            else
            end
          end
        end
      end
    end
  end
end

Appendix B. Example walls

B.1. Walls built with stone sets in Fig. 18

See Figs. B.30–B.32.

B.2. Walls with larger dimensions

See Figs. B.33–B.34.

Variant sampling (one stone per cluster), construction from left, 1.13 minutes

Full data set sampling, construction from left, 2.73 minutes

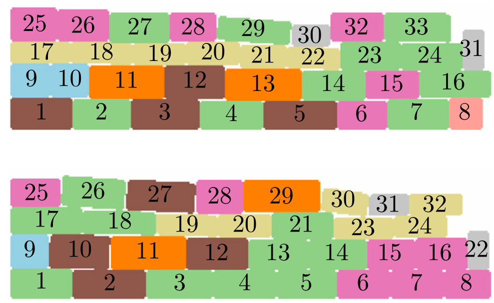


Fig. B.30. Constructed walls with regular stones using various construction method. Image size: 289 × 79 pixels.

Variant sampling (one stone per cluster), construction from left, 1.02 minutes

Full data set sampling, construction from left, 3.08 minutes

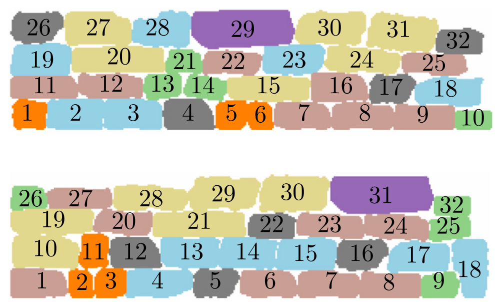


Fig. B.31. Constructed walls with partially regular stones using various construction method. Image size: 280 × 80 pixels.

Variant sampling (one stone per cluster), construction from left, 14.74 minutes

Full data set sampling, construction from left, 28.74 minutes

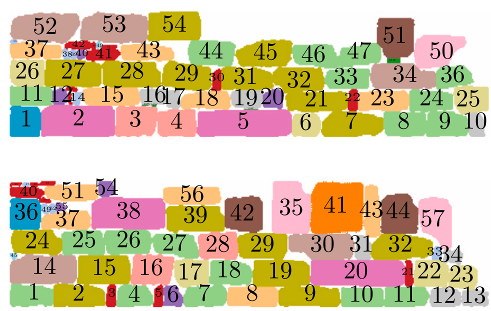


Fig. B.32. Constructed walls with irregular stones using various construction method. Image size: 433 × 115 pixels.

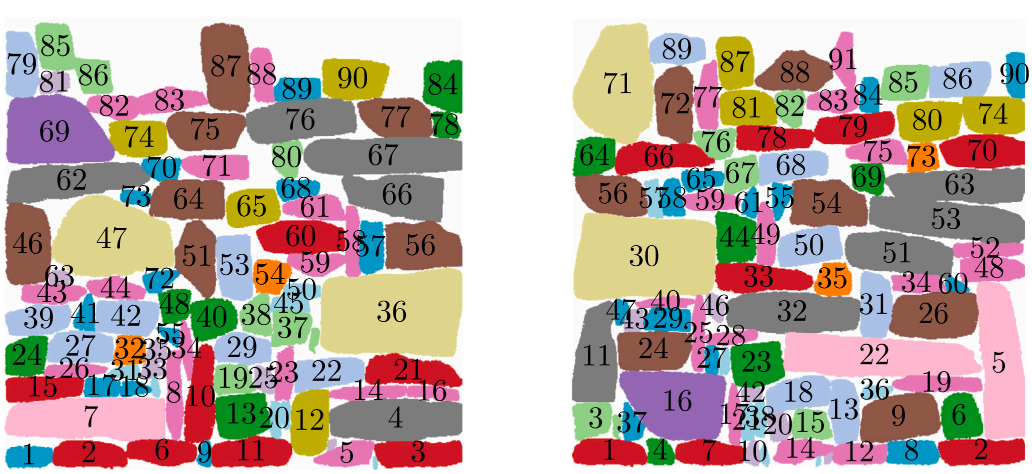


Fig. B.33. Walls constructed from a stone set of 91 irregular stones, of which 90 stones could be placed in the construction of the left wall and 91 stones could be placed in the construction of the right wall. Stones from the same cluster are in the same color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

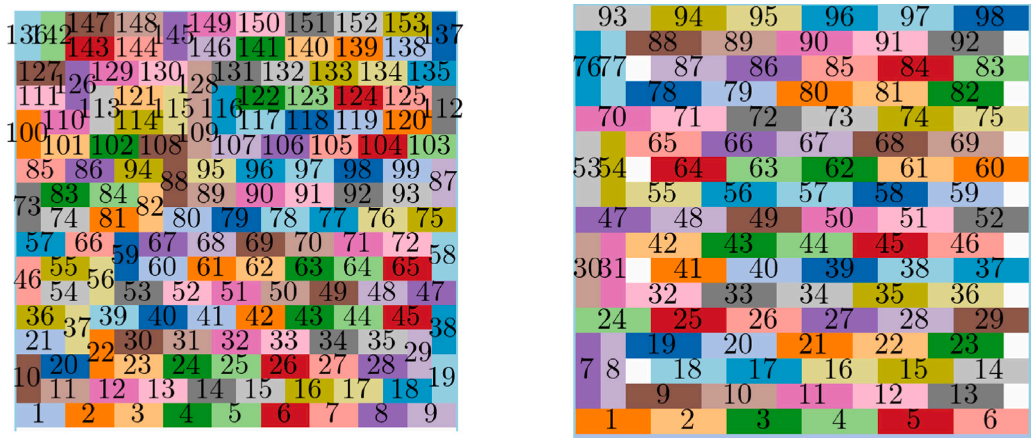


Fig. B.34. Walls constructed with bricks of the same size. The wall on the left is constructed with bricks of an aspect ratio of 2 and the wall on the right is constructed with bricks of an aspect ratio of 3. The bricks are colored for differentiation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

Almeida, C., Guedes, J. P., Arêde, A., & Costa, A. (2016). Geometric indices to quantify textures irregularity of stone masonry walls. *Construction and Building Materials*, *111*, 199–208. http://dx.doi.org/10.1016/j.conbuildmat.2016.02.038.

Almeida, C., Guedes, J. M., Arêde, A., & Costa, A. (2021). Compressive behaviour of old one-leaf stone masonry walls; The influence of patterns' regularity and constructive process. *Construction and Building Materials*, *311*, Article 125339. http://dx.doi.org/10.1016/j.conbuildmat.2021.125339.

Bothara, J., & Brzev, S. (2012). *A tutorial: improving the seismic performance of stone masonry buildings*. Earthquake Engineering Research Institut.

Brehm, E. (2019). Robots for masonry construction–status quo and thoughts for the German market. *Mauerwerk*, *23*(2), 87–94. http://dx.doi.org/10.1002/dama.201900004.

Dabaieh, M., Heinonen, J., El-Mahdy, D., & Hassan, D. M. (2020). A comparative study of life cycle carbon emissions and embodied energy between sun-dried bricks and fired clay bricks. *Journal of Cleaner Production*, *275*, Article 122998. http://dx.doi.org/10.1016/j.jclepro.2020.122998.

de Felice, G. (2011). Out-of-plane seismic capacity of masonry depending on wall section morphology. *International Journal of Architectural Heritage*, *5*(4–5), 466–482. http://dx.doi.org/10.1080/15583058.2010.530339.

EPFL (2023). *Facade of the wall built with plasselb quarry*. Hybrid timber-stone structures, URL https://epfl-enac.github.io/MANSLAB-IBOIS-EESD-timberstone/.

Furrer, F., Wermelinger, M., Yoshida, H., Gramazio, F., Kohler, M., Siegwart, R., & Hutter, M. (2017). Autonomous robotic stone stacking with online next best object target pose planning. In *2017 IEEE international conference on robotics and automation*. ICRA, IEEE, http://dx.doi.org/10.1109/icra.2017.7989272.

Grandcolas, S., & Pain-Barre, C. (2021). A hybrid metaheuristic for the two-dimensional strip packing problem. *Annals of Operations Research*, *309*(1), 79–102. http://dx.doi.org/10.1007/s10479-021-04226-6.

Johns, R. L., Wermelinger, M., Mascaro, R., Jud, D., Gramazio, F., Kohler, M., Chli, M., & Hutter, M. (2020). Autonomous dry stone. *Construction Robotics*, *4*(3–4), 127–140. http://dx.doi.org/10.1007/s41693-020-00037-6.

Kollsker, T., & Malaguti, E. (2021). Models and algorithms for optimising two-dimensional LEGO constructions. *European Journal of Operational Research*, *289*(1), 270–284. http://dx.doi.org/10.1016/j.ejor.2020.07.004, URL https://www.sciencedirect.com/science/article/pii/S0377221720306159.

Koroneos, C., & Dompros, A. (2007). Environmental assessment of brick production in Greece. *Building and Environment*, *42*(5), 2114–2123. http://dx.doi.org/10.1016/j.buildenv.2006.03.006.

Krabbenhoft, K., Lyamin, A., Huang, J., & da Silva, M. V. (2012). Granular contact dynamics using mathematical programming methods. *Computers and Geotechnics*, *43*, 165–176. http://dx.doi.org/10.1016/j.compgeo.2012.02.006.

Lambert, M., & Kennedy, P. (2012). Using artificial intelligence to build with unprocessed rock. *Key Engineering Materials*, *517*, 939–945. http://dx.doi.org/10.4028/www.scientific.net/kem.517.939.

Leao, A. A., Toledo, F. M., Oliveira, J. F., Carravilla, M. A., & Alvarez-Valdés, R. (2020). Irregular packing problems: A review of mathematical models. *European Journal of Operational Research*, *282*(3), 803–822. http://dx.doi.org/10.1016/j.ejor.2019.04.045, URL https://www.sciencedirect.com/science/article/pii/S0377221719303820.

Liu, Y., Choi, J., & Napp, N. (2021). Planning for robotic dry stacking with irregular stones. In *Field and service robotics* (pp. 321–335). Springer Singapore, http://dx.doi.org/10.1007/978-981-15-9460-1_23.

Liu, Y., Shamsi, S. M., Fang, L., Chen, C., & Napp, N. (2018). Deep Q-learning for dry stacking irregular objects. In *2018 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, http://dx.doi.org/10.1109/iros.2018.8593619.

McRaven, C. (1999). *Building stone walls*. Storey Publishing, LLC.

Menezes, A., Vicente, P., Bernardino, A., & Ventura, R. (2021). From rocks to walls: A model-free reinforcement learning approach to dry stacking with irregular rocks. In *2021 IEEE/CVF conference on computer vision and pattern recognition workshops*. IEEE, http://dx.doi.org/10.1109/cvprw53098.2021.00234.

Mollon, G., & Zhao, J. (2012). Fourier–Voronoi-based generation of realistic samples for discrete modelling of granular materials. *Granular Matter*, *14*(5), 621–638. http://dx.doi.org/10.1007/s10035-012-0356-x.

Morel, J., Mesbah, A., Oggero, M., & Walker, P. (2001). Building houses with local materials: Means to drastically reduce the environmental impact of construction. *Building and Environment*, *36*(10), 1119–1126. http://dx.doi.org/10.1016/s0360-1323(00)00054-8.

MOSEK (2024). Mosek optimizer api for python. release 10.1.26. https://docs.mosek.com/latest/pythonapi/index.html.

Mundell, C., McCombie, P., Heath, A., Harkness, J., & Walker, P. (2010). Behaviour of drystone retaining structures. *Proceedings of the Institution of Civil Engineers - Structures and Buildings*, *163*(1), 3–12. http://dx.doi.org/10.1680/stbu.2009.163.1.3.

Pantoja-Rosero, B., Achanta, R., Kozinski, M., Fua, P., Perez-Cruz, F., & Beyer, K. (2022). Generating LOD3 building models from structure-from-motion and semantic segmentation. *Automation in Construction*, *141*, Article 104430. http://dx.doi.org/10.1016/j.autcon.2022.104430.

Paxton, C., Xie, C., Hermans, T., & Fox, D. (2022). Predicting stable configurations for semantic placement of novel objects. In *Conference on robot learning* (pp. 806–815). PMLR, http://dx.doi.org/10.48550/arXiv.2108.12062.

Portioli, F. P. A. (2020). Rigid block modelling of historic masonry structures using mathematical programming: A unified formulation for non-linear time history, static pushover and limit equilibrium analysis. *Bulletin of Earthquake Engineering*, *18*(1), 211–239. http://dx.doi.org/10.1007/s10518-019-00722-0.

Portioli, F. P. A. (2021). *KinABlock_2D*. Zenodo, http://dx.doi.org/10.5281/zenodo.5018923.

Portioli, F. P. A., Godio, M., Calderini, C., & Lourenço, P. B. (2021). A variational rigid-block modeling approach to nonlinear elastic and kinematic analysis of failure mechanisms in historic masonry structures subjected to lateral loads. *Earthquake Engineering & Structural Dynamics*, *50*(12), 3332–3354. http://dx.doi.org/10.1002/eqe.3512.

Settimi, A., Wang, Q., Andò, E., Gamerro, J., Beyer, K., & Weinand, Y. (2023). Projector-based augmented stacking framework for irregularly shaped objects. *Construction Robotics*, http://dx.doi.org/10.1007/s41693-023-00099-2.

Sorensen, T. A. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biol. Skar.*, *5*, 1–34.

Thangavelu, V., Liu, Y., Saboia, M., & Napp, N. (2018). Dry stacking for automated construction with irregular objects. In *2018 IEEE international conference on robotics and automation*. IEEE, http://dx.doi.org/10.1109/icra.2018.8460562.

Umetani, S., & Murakami, S. (2022). Coordinate descent heuristics for the irregular strip packing problem of rasterized shapes. *European Journal of Operational Research*, *303*(3), 1009–1026. http://dx.doi.org/10.1016/j.ejor.2022.03.034.

Vivian, J. (2014). *Building stone walls*. Storey Publishing, LLC.

Wang, Q., dos Santos, K. R. M., & Beyer, K. (2023). Micro modeling of irregular stone masonry walls using mathematical programming. In *Structural analysis of historical constructions* (pp. 591–602). Springer Nature Switzerland, http://dx.doi.org/10.1007/978-3-031-39450-8_49.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244. http://dx.doi.org/10.1080/01621459.1963.10500845.

Wermelinger, M., Furrer, F., Yoshida, H., Gramazio, F., Kohler, M., Siegwart, R., & Hutter, M. (2018). Greedy stone tower creations with a robotic arm. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence*. International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2018/760.

Zhang, S., Mousavi, S. M. T., Richart, N., Molinari, J.-F., & Beyer, K. (2017). Micromechanical finite element modeling of diagonal compression test for historical stone masonry structure. *International Journal of Solids and Structures*, *112*, 122–132. http://dx.doi.org/10.1016/j.ijsolstr.2017.02.014.