



Brick Patterning on Free-Form Surfaces

Shaghayegh Rajabzadeh¹  · Mario Sassone¹

Published online: 1 July 2016
© Kim Williams Books, Turin 2016

Abstract With the aim of proposing vaults and domes in contemporary architecture, this paper explores an innovative method to help designers model the brick patterning on a curved surface automatically and interactively, especially during the early design phases, when most important decisions are made. The focus is on the development of a modeling approach to patterning, capable of defining the brick courses, handling mortar thickness, and controlling all the related geometrical issues of free-form surfaces. A computational environment has been developed and implemented to simulate patterning in 3-D space, allowing the user to develop the arrangement of bricks on a desired structure. With this digital tool it is possible to model different kinds of patterns such as stretcher bond or herringbone pattern on any kind of free-form surface, increasing the accuracy and speed of construction and enabling the designer and builder to estimate the brick requirements before fabrication.

Keywords Masonry vaulting · Brick dome · Computational morphogenesis · Brick pattern · Bricklaying · Interactive tool · Automatic brick pattern modeling

Introduction

In 2013 an architecture workshop entitled Computational Morphogenesis Workshop took place at the School of Architecture of the Politecnico di Torino, focusing on the construction of a digitally designed free-form brick vault (Rajabzadeh and Sassone

✉ Shaghayegh Rajabzadeh
shaghayegh.rajabzadeh@gmail.com

Mario Sassone
Mario.Sassone@polito.it

¹ Politecnico di Torino, Turin, Italy

2014). The resulting free-form dome, measuring about 5 m², was constructed as a one-layer brick surface, with bricks laid in a typical interlocking herringbone pattern to avoid continuous brick joints and provide a stronger structural bond (Fig. 1).

Due to the complexity of the geometry, many bricks had to be cut in non-regular shapes, interrupting the work flow. It became clear that the placement and bonding of each brick is a step in a sequence of technical decisions that have an influence on strength, stability, constructability and economy of the structure.

From this practical construction experience was born the idea of a research project on the design and optimization of the patterns of bricks in domes and vaults involving the development of digital tools customized for such a purpose. In 2014 the research results were tested on a new construction, during the second edition of the Computational Morphogenesis workshop, in which the brick pattern was fully designed with the aid of digital tools.

Masonry Construction

Masonry consists of elementary units, such as bricks or stones, and binding layers, to form architectural elements in a variety of colors, textures and dispositions. In this paper the term “brick pattern” refers the position of bricks in a systematic manner according to some criteria of repetition or motif in order to provide the suitable interlocking units to cover a surface without any gaps over overlaps (Grunbaum and Shephard 1990: 1).

Many kinds of brick patterns have developed in the history of masonry construction. Sumerians laid plano-convex bricks in herringbone pattern for constructing walls (Delougaz 1933: 37). This arrangement provides double interlocking in two directions, preventing vertical compression and possible horizontal shear (Fig. 2a). Stacking pieces of limestone in a sloping system formed a corbelled structure named “False vault” (Fig. 2c). The vertical compression of the



Fig. 1 The free-form dome constructed with non-regular bricks during the 2013 Computational Morphogenesis Workshop held at the Politecnico di Torino

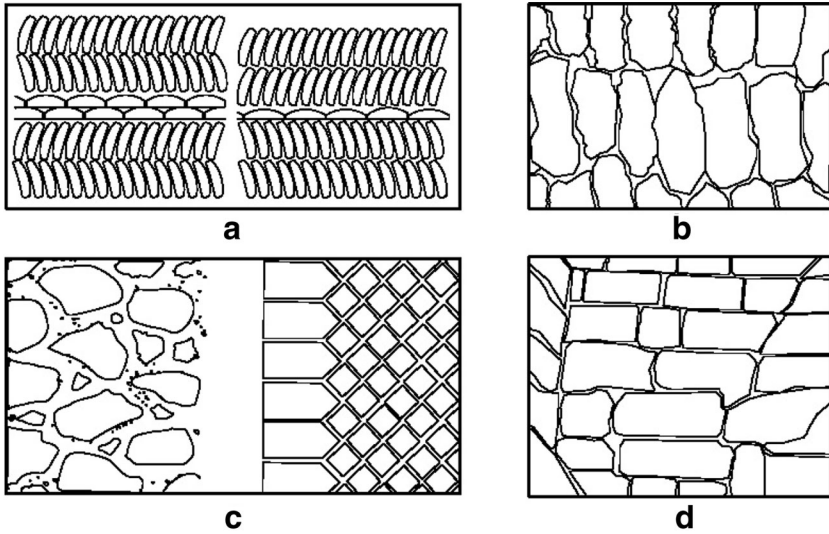


Fig. 2 a Herringbone pattern at al-Ubaid, 3800 BC; b Plano-convex bricks at Mesopotamia, 800 BC (after Delougaz 1933: 7, 21); c *Opus reticulatum* and *opus incertum* brickwork (after Fletcher 1976: 116); d false arch of the Lion Gate at Mycenae, thirteenth century BC (after Como 2013: 130)

units prevents the overturning of the lower units (Como 2013: 130). Romans used masonry units in diverse arrangement and shapes for different applications (Fig. 2b): among others, *opus quadratum* used rectangular blocks of stone with or without mortar joints; *opus incertum* used irregularly shaped and randomly placed uncut stones; *opus reticulatum* involved a grid of joints laid in diagonal lines (Fletcher 1976: 117).

Byzantine brickwork included meander, chevron, horizontal and oblique brick patterns, but the herringbone was the most common pattern (Fletcher 1976: 194). The dome of Santa Maria del Fiore in Florence by Brunelleschi is one of the most famous examples of the use of herringbone pattern in a very large building (Escrig 2006: 110). The Catalan vault basically consists of three or four layers of bricks, where the bricks of the first layer are bonded to each other by fast-setting plaster, providing a bed for subsequent layers (Ochsendorf 2010: 127). In the Nubian vault, having the maximum bonding between the units, bricks are laid full bed, face to face and slope into the cross section of the vault (Davis and Block 2012). In the late nineteenth and early twentieth centuries the Guastavino company constructed many Catalan vaults in United States.

Contemporary Studies

A large number of studies focus on historical issues and on the behavior of existing buildings but few address the design and construction of masonry vaults in contemporary architecture. The Gramazio Kohler research group at the ETH Zürich

(<http://gramaziokohler.arch.ethz.ch/web/e/forschung/index.html>) focuses on the integration between masonry and industrial robotics (Bonwetsch et al. 2007). They have accomplished an innovative brick laying method for decorative panels and walls, based on the use of small portable robots. The robotic construction is supported by the BrickDesign digital tool, a Rhinoceros[®] plug-in developed by the ROB company (<http://www.rob-technologies.com/en/company/news/brickdesign>), which enables the user to apply patterns and images onto a straight brick facade by mapping an image with rotated bricks. At Georgia Institute of Technology, a team of researchers is trying to improve this technique by developing an interactive and real-time digital tool to generate arbitrary brick patterns interactively and map them onto the wall (Afsari et al. 2014). Researchers at Harvard University have worked on a kind of framework to help designers generate complex shapes through different brick tessellations. This group also tried to combine this workflow with robotic fabrication and surface tessellation technology (Imbern 2014).

The issues related to the digitally-assisted tool for constructing of masonry vaults are the focus of a researchers at the Universidad Politécnica de Madrid, who applied the augmented reality technique to the construction of timber vaults in order to achieve a construction method without formworks (Garcia et al. 2012). Moussavian and Gentry are working on digital tool for automated generation of brick vaults. At the 9th International Masonry Conference they presented a tool for modeling a pattern based on bricklaying rules applied to three type of Persian vaults (Moussavian and Gentry 2014).

In order to re-propose the masonry vaulting system for contemporary architecture, Giuseppe Fallacara and Claudio D'Amato from the Politecnico di Bari have combined the stereotomy method, three-dimensional graphical modeling techniques, and computer-guided cutters to design and construct the complex stereotomic shapes by using elaborate notions and easy actions (Fallacara 2009).

Designing and constructing free-form masonry vaults is one of the main research core of Block Research Group (BRG) at ETH University (<http://www.block.arch.ethz.ch/>). BRG develops structural design, form finding tools and computational frameworks such as RhinoVAULT, a free plug-in for Rhinoceros[®] being implemented based on the concept of Thrust Network Analysis. Through the process of form-finding methods, they have constructed many prototypes of free-form brick vaults (Rippmann et al. 2012). Lachauer, on the other hand, has developed a technique to generate the tessellation of funicular vaults (Lachauer et al. 2010) used to generate and cut masonry units during fabrication.

Aim of the Present Research

Bricklaying on vaults is reliant on the skill of craftsmen. The orientation and the interlocking order of bricks may be decided by the designer in theory but are applied by the mason in practice. It is a hard task to define before construction the position of each brick on curved elements such as arches, vaults, domes and free-form shapes. Further, some amount of non-regular bricks are almost always required. In the absence of a precise model of the pattern, masons spend time cutting the bricks during the construction of geometrically complex elements.

These construction issues largely depend on the intrinsic difficulty of handling all the geometric aspects of the problem during the design stage, when usually only the overall shape of the dome is provided with only generic indications regarding pattern. In complex projects not governed by simple geometric shapes, the early design phase is the most crucial moment (Méndez Echenagucia et al. 2015), the one in which even small decisions can be effective on the final realization. A more advanced design effort, supported by digital tools, can result in stronger control of the project itself and may allow the exploration and exploitation of a wide set of alternatives, but in such a perspective, the design tools must have strong interaction capabilities through their interfaces in order not to hinder the flow of the design process.

This present research is aimed at automating the modeling of brick patterns and enabling the designers to identify the material requirements, such as quantity of blocks or the shapes of unusual blocks, before construction begins, in order to reduce both construction time and material waste. Considering these aspects, it was decided to focus on the creation of a modeling environment capable of defining the brick courses, determining the mortar thickness, controlling the geometrical aspects such as rotation and orientation, and shaping the special pieces. These features assist designers in designing complex patterns, and assist craftsmen during construction. In order to model brick patterns on free-form surfaces, a computational environment has been developed and implemented. In the development of this digital tool, we have taken into account the informational aspect of masonry construction and its adaptability to the skills of masons. The tasks of bricklaying have been simulated in 3-D space and the user is able to model the brick arrangement on a desired structure. The tool proposed in this paper is result of an integration of commercial CAD software and a common scripting language. This new tool provides an interactive method for designers and masons to design and model the brick patterns on complex shapes.

The Geometry of Brick Patterning

The construction of brick vaults is usually accomplished with the aid of a formwork that plays a double role: temporary support and shape reference. In some special cases, such as the Catalan vault, where the temporary support is not required, the formwork can be reduced to a set of curved templates. The formwork surface, or the surface implicitly defined by templates, is then the reference geometry of the vault and will be assumed as the basis of the patterning process. The reference surface has boundaries that define and constrain the bricklaying process: the boundaries can be structurally free or can form the abutments of the vault, which is supported by other structures such as columns or walls, or the ground. Generally, the abutments are the starting points for the construction, but the design process is free to start at any given point. In addition, the face of brick laid on the formwork should be aligned relative to the tangent plane of the formwork on the central point of the brick's surface. However, sometimes in brick adjustment leveling the surface is preferred to the perfect alignment of the bricks to the tangent plane. Lastly, the brick is oriented to the desired direction on the tangent plane mostly rotating around the normal

vector crossing the central point. Tangency and the orientation of the brick are actually considered at the same time, in order to find the best position, which is obtained by smoothing the curvature, reducing the mortar gap and producing the union pattern. The brick pattern basically follows a specific kind of brick arrangement (running bond, herringbone, etc.) that consists of a line of adjacent bricks called a brick course.

In surfaces with non-zero Gaussian curvature, generally positive curvature or small parts with negative Gaussian curvature, brick courses might diverge or converge. If the size of the curvature is very large in comparison to the size of the brick, these errors can be solved by adjusting the bricks. The divergence or convergence of brick courses creates a need for specially shaped bricks cut from regular ones, although ideally the bricklayer should attempt to minimize the need for such bricks. Because the forces inside the domes act parallel to the reference surface, the joints must be orthogonal to this surface.

The structural behavior is the main feature of masonry. Bricks and stones are required to work mainly in compression because they are made of brittle material and they do not carry significant tensile stresses. Mortar mainly provides only a bed to distribute the compressive forces between the bricks, thus its internal cohesion and its adherence to the bricks are frequently much smaller than the cohesion of the bricks themselves; mortar layers are in fact the weak points of structure. This is particularly clear when no mortar is present. It is therefore important that the stresses within the mortar beds are only compressive and, if possible, that no shear stress occurs. This can be achieved by orienting the mortar beds orthogonally to the direction of compressive forces. Besides compression and shear stresses, it is almost impossible to avoid the presence of residual tensile stresses. Even in members subjected only to uniform compression along a single direction, small tensile stress can occur in the orthogonal direction. It is therefore important to avoid continuous mortar joint that could allow tensile cracks to spread through the structure. This is achieved by interlocking the bricks horizontally as it is usually done in brick walls. In more complex architectural elements such as vaults and domes, tensile stresses due to bending or the effect of confinement can occur anywhere. The interlocking then can be extended to both directions, as in the herringbone pattern, to avoid any continuous joints.

Interlocking between units can guarantee the stability of the building even during the bricklaying. The method used to arrange the units can free the masonry structure from the requirement of formwork in the construction process, or can at least reduce the complex formwork to the simple centering or guides. In this condition the gluing effect of mortar and the brick orientation play very important roles in keeping bricks from falling during bricklaying and increasing the speed of the construction process. For instance, in the running bond, bricks are mostly joined each other on two sides, while in the herringbone it is possible to insert a brick between three others.

Parametrization of a Brick

A regular brick is a rectangular cuboid with definite dimensions of length, width, and height. In a Cartesian three-dimensional coordinate system, a cuboid consists of

six points with definite coordinates, each of which is specified by three values of (x,y,z) . Displacement, rotation or any other geometrical modification of the cuboid is possible by applying it on any point belonging to the shape. Such a point is called the “reference point”.

Here, in order to simplify the application of geometrical modifications to the brick, the reference point is considered to be the center of the cuboid where the space diagonals intersect. Thus each brick is assumed a cuboid whose reference point is at the origin $(0,0,0)$ and whose edges are parallel to the coordinate axes. In Fig. 3, dimensions a , b and c indicate respectively half of the length, width, and height. This allows us to define a reference plane parallel to the faces formed by the edges of the length and width and passing through the origin.

The bricklaying on a surface means locating a brick at a specific point on the surface, aligning it to the tangent plane of the formwork and simultaneously orienting it to the desired direction. Let us now define the following considerations of displacement, alignment and orientation:

Displacement

In order to move a cuboid which is parallel to the XY plane of a Cartesian system, it is sufficient to displace a reference point from the previous position to the new position.

Alignment

As the brick reference plane defines a normal vector, the surface tangent plane defines the surface normal vector at a given point (Fig. 4).

Two bricks are aligned when their two normal vectors are coincident. This can be achieved by rotating the brick in a direction perpendicular to both the normal

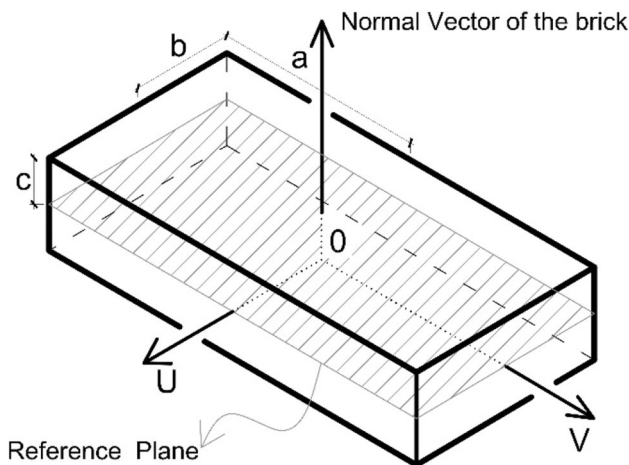
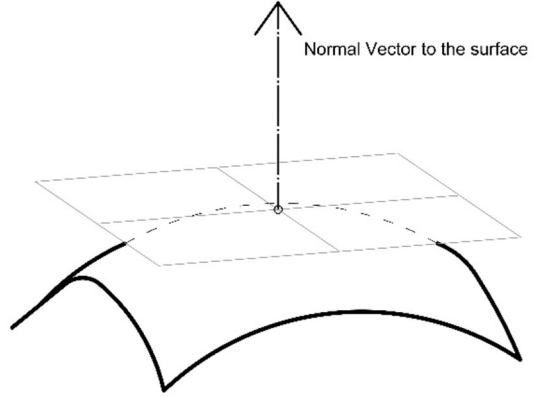


Fig. 3 Parametrization of a brick

Fig. 4 The normal vector on a specific point of a surface



vectors, by the angle they form. The vector representing such a direction is their cross product, while the angle amplitude is obtained from their dot product:

$$\{R\} = \{N_b\} \times \{N_s\} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ N_{b,x} & N_{b,y} & N_{b,z} \\ N_{s,x} & N_{s,y} & N_{s,z} \end{vmatrix} = \begin{Bmatrix} N_{b,y}N_{s,z} - N_{b,z}N_{s,y} \\ -N_{b,x}N_{s,z} + N_{b,z}N_{s,x} \\ N_{b,x}N_{s,y} - N_{b,y}N_{s,x} \end{Bmatrix}$$

$$\cos \alpha = \{N_b\} \cdot \{N_s\} = N_{b,x}N_{s,x} + N_{b,y}N_{s,y} + N_{b,z}N_{s,z}$$

where $\{N_b\}$ and $\{N_s\}$ are the normal vectors, $\{R\}$ is the rotation axis and α is the rotation angle.

As Fig. 5 shows, by rotating the brick in such a way that its perpendicular axis becomes parallel to the normal vector of the selected point, the brick become tangent to the surface.

Orientation (Yaw Rotation)

The yaw allows designers to design the brick pattern. In fact, the whole process of designing brick patterns is based on this application. It is achieved by rotating the brick on the tangent plane at the reference point. In aviation the yaw axis is that drawn from the top to the bottom of the aircraft that is perpendicular to the other axes. Yaw rotation in bricklaying is a rotation about the yaw axis (Afsari et al. 2014) and indicates the rotation of bricks about the axis which crosses the reference point and is perpendicular to the face aligned to the reference surface.

By aligning a brick to the tangent plane of the surface, the normal vector of the surface at the reference point and yaw axis of the brick are superimposed on each other. After that, the brick can be freely rotated round its yaw axis (Fig. 6).

A pattern arrangement is based the relative position of adjacent bricks, considering brick size and the thickness of mortar joints. The relative position is then used incrementally to add new bricks. The position of the new brick (the output

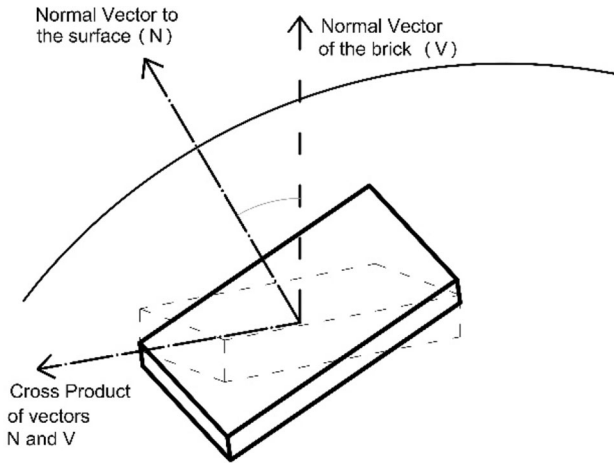


Fig. 5 Aligning the brick to the tangent plane

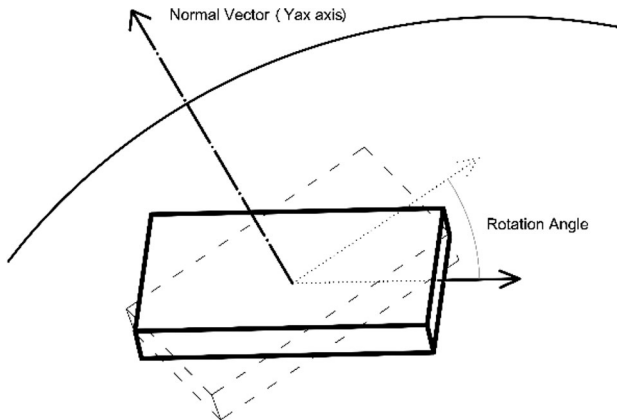


Fig. 6 Yaw rotation

brick) with respect to the previous (the input brick) is defined both by relative position of reference point and the relative yaw rotation.

Limiting the brick shapes to cuboid and starting from a flat layout, there are many ways to place one brick close to another, but only a few are considered truly relevant from a practical point of view. Further, for a cuboid brick, a set of relative positions are shown in Fig. 7 involving respectively no rotation or 90° rotation. In order to align the output brick to the input brick, it is necessary to know the angle between the brick direction vectors.

Assuming the center of the input brick as a reference point, the position of the output brick can be determined through some simple calculations involving brick size, mortar joint thickness and rotation. Considering the origin of the coordinate system as

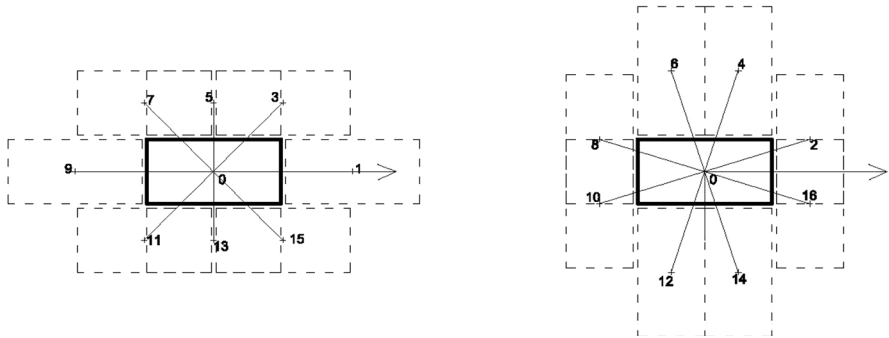


Fig. 7 A set of possibilities for brick location

$(0, 0)$, ' a ' as the half of the length of brick, ' b ' as the half of the width, and ' d ' as the mortar gap, the coordinates of each reference point will be as shown in Table 1:

values in Table 1 refer to these a , b and d dimensions:

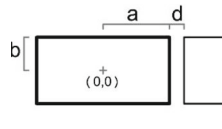


Table 1 The coordinates of bricks regarding the brick dimensions and mortar gap

Point	X	Y
1	$2a + d$	0
2	$a + b + d$	$a - b$
3	$a + d/2$	$2b + d$
4	$a - b$	$a + b + d$
5	0	$2b + d$
6	$-a + b$	$a + b + d$
7	$-a - d/2$	$2b + d$
8	$-a - b - d$	$a - b$
9	$-2a - d$	0
10	$-a - b - d$	$-a + b$
11	$-a - d/2$	$-2b - d$
12	$-a + b$	$-a - b - d$
13	0	$-2b - d$
14	$a - b$	$-a - b - d$
15	$a + d/2$	$-2b - d$
16	$a + b + d$	$-a + b$

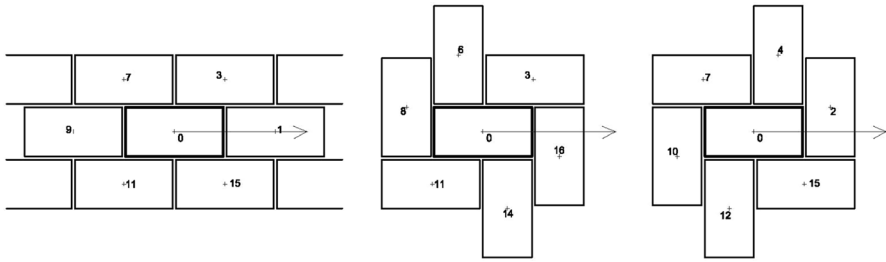


Fig. 8 Stretcher and herringbone patterns

Many regular patterns can be obtained in this way: stretcher, shiner, header, sailor, soldier, herringbone, and others. In the stretcher bond pattern, each brick course is a half brick stretcher from the adjacent course. In herringbone pattern, each brick is rotated 90° about the yaw axis relating to the side brick (Fig. 8).

On a curved surface, bricks lay on different tangent planes so that the brick directions are skew lines that are neither intersecting nor parallel. If the input brick direction is translated from its origin to the origin of the output brick, the plane defined by these two lines is not coincident with the reference plane of the output brick. A rotation in this plane would destroy the correct alignment of output brick to the surface. From a geometrical point of view, in order to find the angle between two bricks, it is enough to know the angle between the direction line of one of the bricks and the projection of the other brick direction on the reference plane of the prior one. Since the output brick should be rotated in its reference plane, the projection of the direction line of the input brick should be defined on the reference plane of the output brick.

The projecting plane must be perpendicular to the reference plane of the output brick and crossing the direction line of input brick (after copying) and then is defined by three points: the reference point of output brick, as the origin point of plane, a second point on the copied direction and a third point on the normal vector of the input brick after being copied to the reference point of output brick. The intersection of the reference plane of the output brick and the projecting plane will be a line crossing the center point. The angle between the intersection line and the direction line of the output brick indicates the rotation angle of the output brick to be aligned with the input brick. The output brick should be rotated about its yaw axis but as the normal vector direction is concerned by the properties of surface curvature; the rotation axis is the cross product of direction line and intersection line (Fig. 9).

When patterning causes two bricks to overlap, the output brick must be cut to shape with reference to the input brick. The cutting plane has to be parallel to the normal vector of the output brick, and at the same time parallel to the direction of the intersecting side of the input brick (Fig. 10). The cutting plane is then offset to take into account the thickness of the mortar joint.

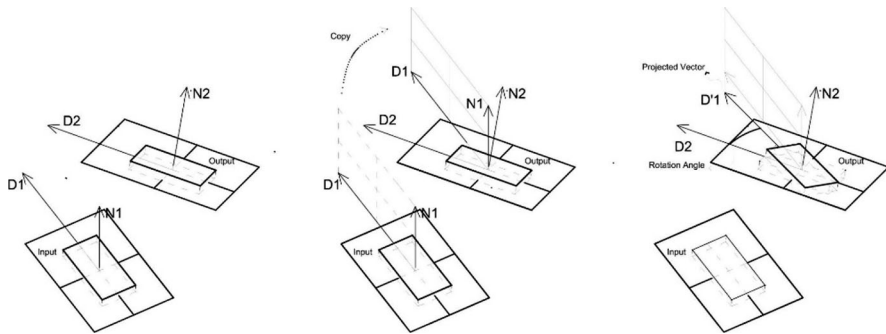


Fig. 9 Arranging the brick with the direction of the previous one

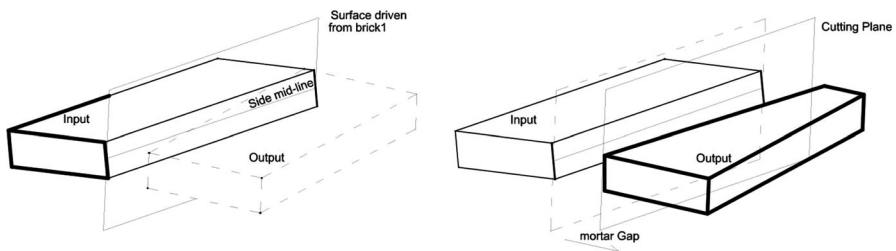


Fig. 10 Splitting a brick overlaid by another brick

Brick Pattern: an Interactive Plug-In

All the geometrical concepts of a bricklaying process are converted by the authors to develop a new digital tool named “Brick Pattern” which enables designers to design and model the brick pattern on a free-form surface. The development of this digital tool has required three kind of digital environments: first a programming language powerful enough to implement all the logic involved in the algorithms; second, a 3D modeling environment, capable of performing all the geometric manipulations in an optimal way; third, a user-oriented interface that allows the appropriate level of interaction and makes the application useful at different design stages. In the development of “Brick pattern”, two programming environments were used: the Rhinoceros[®], commercial graphic application, and Python (http://developer.rhino3d.com/guides/rhinopython/what_is_rhinopython/), an open source programming language from the IronPython version implemented inside Rhinoceros[®].

Geometric concepts needed to be translated into programming language. Since Rhinoceros[®] is a graphical design application, many concepts and objects have a specific representation inside of it. Basic custom objects and functions are obtained by directly calling native Rhinoceros[®] objects and commands. Intermediate-level subroutines, on the other hand, are obtained by combining

Table 2 Geometric objects and functions allowable in the digital graphic environment

Geometrical objects	Rhinoceros® objects	Input data
Cuboid (brick)	Box(pts array)	8 points
Cuboid reference point	Point(x,y,z)	3 reals
Cuboid reference direction	Line(pt1, pt2)	2 points (6 reals)
Cartesian axes system	CPlane(ptO, ptX, ptY)	3 points (9 reals)
Plane normal direction	VectorCrossProduct(u1, v1, w1, u2, v2, w2)	6 reals
Intersection of two planes	PlanePlaneIntersection()	2 planes
Angle between directions	Angle(ln1, ln2)	2 lines
	VectorAngle(u1, v1, w1, u2, v2, w2)	6 reals
Generic spatial surface	Surface(cp1, cp2,...)	n control points
Surface normal direction (at a given point)	SurfaceNormal(srf, u,v)	Surface, UV parameters
...
Geometrical operations	Rhinoceros® functions	Arguments
Displacement	Move(obj, pt1, pt2)	Object, 2 points
Rotation	Rotate(obj, pt, a, ln)	Object, point, angle, line
Offsetting	OffsetSurface(srf, d)	Object, real
Cutting (slicing)	SplitBrep(obj, srf)	Object, surface
...
User's actions	Rhinoceros® functions	Return value
Selecting a surface	GetObjects (srf)	Surface
Selecting a point on a surface	GetPointOnSurface(srf)	Point
...

the basic low-level functions. The reference free-form surface that describes the vault shape is represented by a NURBS Surface object, the cuboid that describes a brick is defined as a Box object, and so forth for all the other geometric objects involved in the process. These are shown in Table 2, which also shows how basic operations and user interfaces are allowable in the graphic environment.

The access to graphic objects and commands at the IronPython programming stage is provided by the “rhinoscriptsyntax” package that is imported as a library with the custom name ‘rs’. One example of a low-level custom object is used to handle the set of possible positions of a brick with respect to the adjacent one, described above. Such a compound object is formed by a star of Line objects linked to the Box brick. Analogously, the following code shows a low-level custom function that defines the x–y plane, tangent to a *Surface* (“surface”) at a *Point* (“point”):

```

import rhinoscriptsyntax as rs
def DefiningPlane(surface, point):

    pointParameters = rs.SurfaceClosestPoint(surface, point)
    surfaceParamArray = rs.SurfaceEvaluate(surface, pointParameters, 1)
    pointOnSurface = surfaceParamArray(0)
    surfaceUdirection = surfaceParamArray(1)
    surfaceVdirection = surfaceParamArray(2)

    PointOnU = rs.PointAdd(pointOnSurface, surfaceUdirection)
    PointOnV = rs.PointAdd(pointOnSurface, surfaceVdirection)
    return rs.PlaneFromPoints(pointOnSurface, PointOnU, PointOnV)

```

An example of an intermediate-level object is given by the “Gizmo” functionality. The Gizmo helps the user to define the bricks orientation interactively. The following pseudocode describes its workflow.

```

Gizmo (surface, previousBrick)
    Get: newBrickPoint(surface, previousBrick)

    Add: Circle
    Add: direction line → from oldBrick to newBrick
    Add: adornments
    Align: adornments → to newBrick direction

    Get: direction on circle

    Return: selected direction

```

“Brick Pattern”, a free plug-in for Rhinoceros® downloadable from www.food4rhino.com, currently provides ten new commands, available on the graphical-icon menu (Fig. 11).

After setting the dimensions of brick using the first command, a group of commands makes it possible to handle individual bricks by creating, moving, rotating around the yaw axis and slicing. Two further commands introduce the interactive patterning: with “Smart Line” brick courses can be created by manually setting the orientation of each brick, while with “Smart Stain” a portions of the surface can be manually filled by arranging the position and orientation. The final three commands allow automatic brick patterning: “Brick Line” generates a course of bricks, “Stretcher Bond” and “Herringbone Bond” generate courses of bricks in such two patterns.

As an example, the “New Brick” command first asks the user to select a surface by picking the first brick point on it, then the gizmo is shown (Fig. 12, left), centered at this point, to select the brick yaw. The brick is automatically aligned to the tangent plane and oriented in the selected yaw direction (Fig. 12, right).

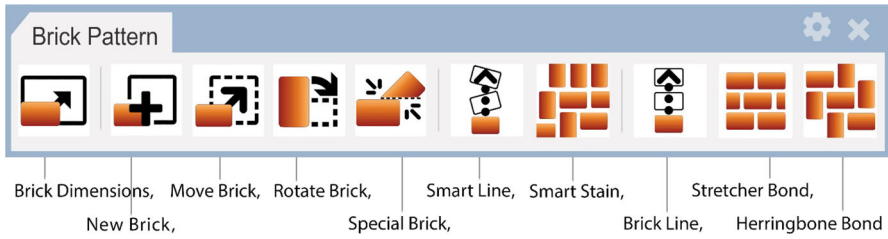


Fig. 11 The icons menu of the “Brick Pattern” plug-in for Rhinoceros

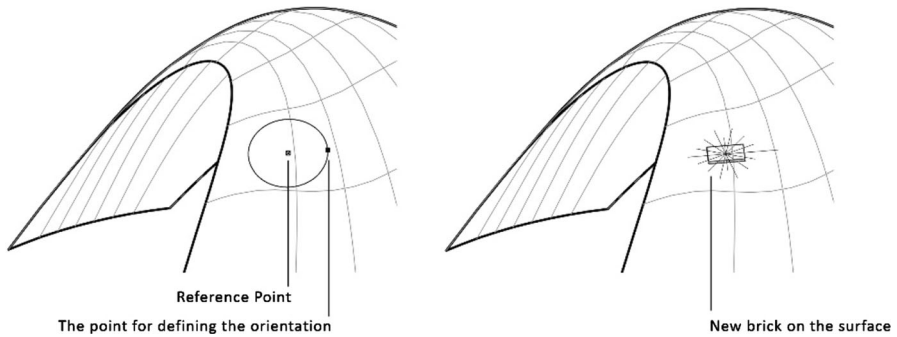


Fig. 12 Creating a new brick on a surface: the insertion point and the direction (left) and new brick (right)

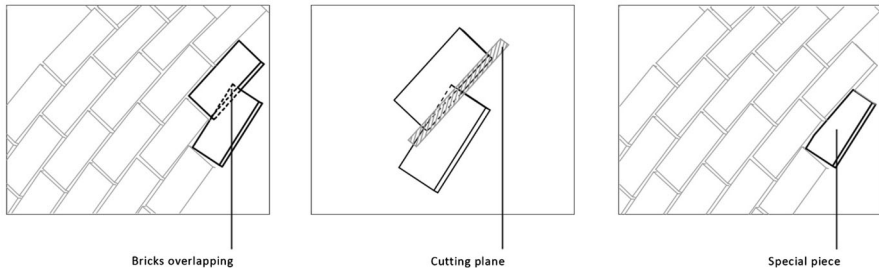


Fig. 13 The procedure of the “Special Bricks command”

The “Special Bricks” command helps the user to create customized bricks by cutting one brick when it overlaps another. The user is asked to first select the overlapped brick, then to select the brick to be cut, then the side of the overlapped brick that must be cut, and finally to pick the part to keep: the leftover portion is eliminated (Fig. 13).

A single course of bricks can be always created linking two points of the surface, except when the sequence is self-intersecting. In fact, the course approximately follows a geodesic curve of the surface. On surfaces with positive Gaussian curvature, geodesic curves that are initially parallel will converge and cross, while on surfaces

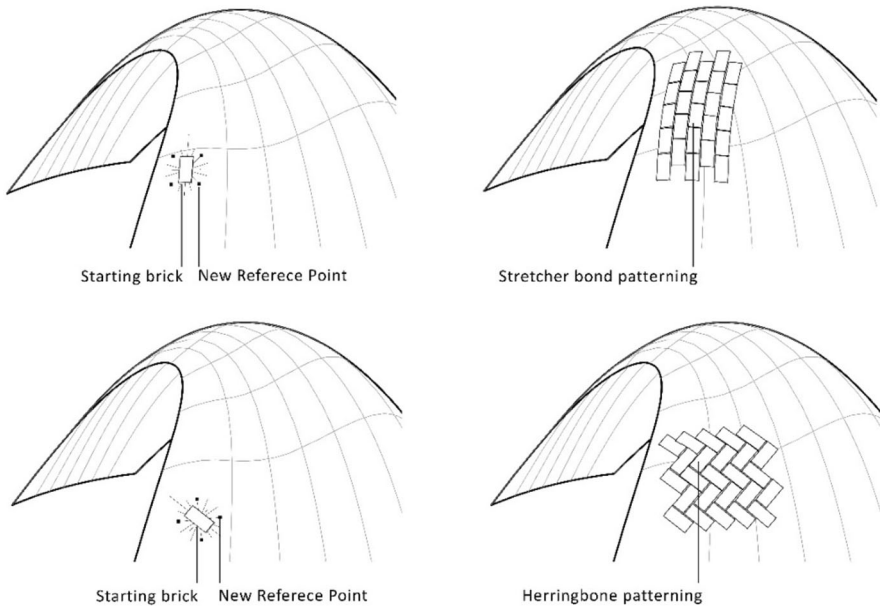


Fig. 14 Automated patterning: **a** stretcher bond (*above*), **b** Herringbone bond (*below*)

with negative Gaussian curvature they diverge. Situations where bricks overlap and joints widen are not allowed, and customized bricks are required to complete the pattern. Such a problem can be partially reduced, at the design stage, by modifying and optimizing the courses layout, but cannot be avoided altogether. This fact limits the automation of the patterning process just to portions of the structure, and the user must decide how to resolve the discontinuity. Patterning has to be regarded as a creative design operation rather than a mere optimization process.

The “Brick Line” command generates a course or a generic sequence of n bricks, each aligned to the previous, in accordance with the chosen alignment rule.

The last two commands, “Stretcher Bond” (Fig. 14a) and “Herringbone Bond” (Fig. 14b) cover a portion of the surface with n bricks, arranged in stretcher or herringbone bond. The user inputs the number of courses and the number of bricks in each course for stretcher bond, or the number of bricks in herringbone bond and specifies the direction of bricklaying. Depending on the kind of surface, the joints between brick can shrink or enlarge, within a range of minimum and maximum values.

Application: Modeling the Brick pattern on Catalan Vaults

The program of the 2014 Computational Morphogenesis Workshop called for the construction a free-form Catalan vault, as a thin tile vault or as “Volta in Folio” in Italy (Ochsendorf 2010: 20). Compared with other vaulting methods, it is thinner and weighs less, so there are the fewer thrust forces. In order to save time and material, it was decided to construct a Catalan vault with two layers. The overall

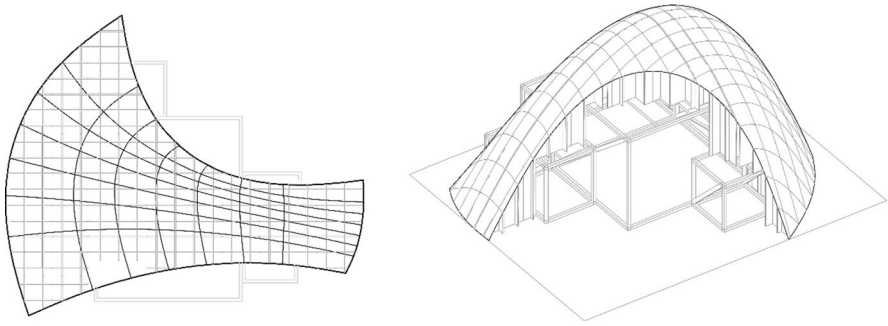


Fig. 15 The design of the vault and its formwork

shape of the vault was modeled in the Rhinoceros[®] NURBS environment and parametrically designed by means of the Grasshopper[®] tool (Fig. 15).

The reference surface was initially defined to achieve the architectural requirements and to design the formwork, consisting of intersected layers of cardboards. Some wooden boxes were the platform of the formwork. The bricks measured $24 \times 11.5 \times 2.5$ cm.

The structural behavior was analyzed and checked in order to avoid or limit the presence of bending and tensile stresses. After a preliminary FEM analysis, further adjustments and optimization steps were performed by using the free RhinoVault software developed by the Block Research Group, which helped to find a structurally optimal shape through the thrust-network method.

When the vault shape was defined, the bricklaying patterns were studied by means of the Brick Pattern code. Two different concepts for laying the first layer were tested. The first was based on laying a brick course perpendicular to the abutment of the vault, to be continued by keeping the other courses parallel to it. Modeling this concept was relatively time-consuming because plenty of bricks had to be oriented or displaced in order not to overlap the adjacent bricks but still remain approximately parallel to the neighboring course (Fig. 16).

The second concept was based on the brick courses being parallel to the abutment. Modeling this concept was faster and simpler. Bricks were laid from two baselines and each new course was parallel to its adjacent course as well as to the baselines. The special brick shapes were required on top of the shell. This concept was selected to form the first layer of the vault, because from the constructive point of view it provided the possibility of starting the bricklaying from two baselines (Fig. 17).

The second layer was to be the exposed layer, so its pattern required more aesthetic attention. It was also necessary that the brick courses of the second layer would be rotated 30° – 45° with respect to the layer below it. Modeling this layer was not pure bricklaying but the pattern was deliberately designed. The edges of the surface were covered first. The arch-shaped edges were started to cover the ground so the special shaped bricks were placed in the crown part. Further, the bricks followed three different courses so that the interference areas required special

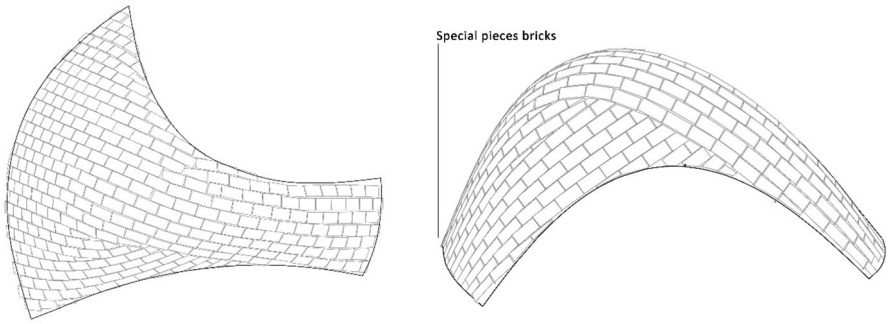


Fig. 16 The first concept of brick patterning designed for the vault of the workshop

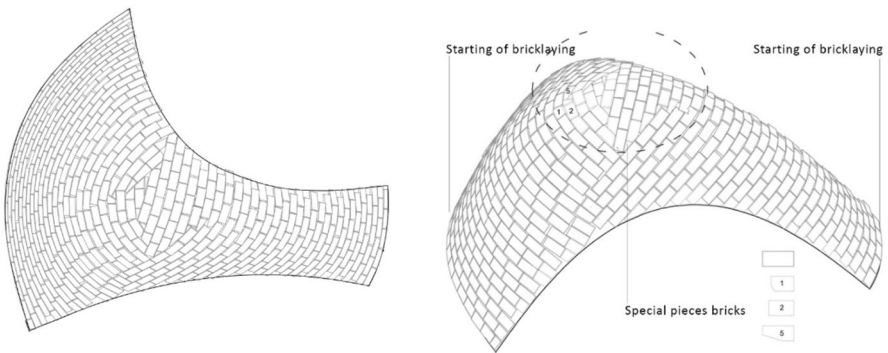


Fig. 17 The second concept of brick patterning designed for the vault of the workshop

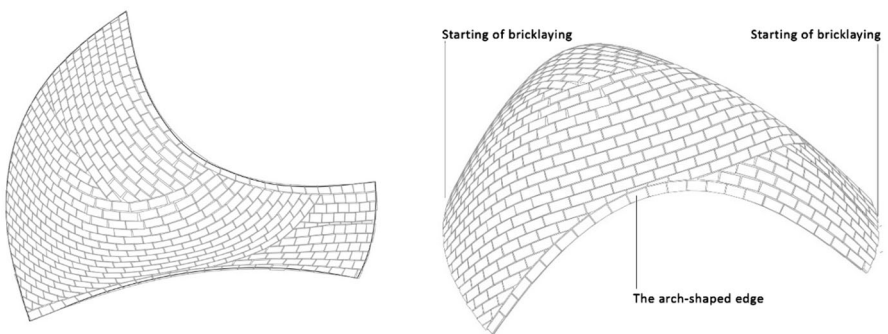


Fig. 18 The concept of brick patterning for the second layer of the vault of the workshop

shaped bricks. This design facilitated the bricklaying process and had a decorative role (Fig. 18).

The digital models included all the required special bricks. The special shapes were projected on a plane as a module for cutting the bricks. The bricks were cut in parallel with the constructing process. The cutting models approximately matched their digital samples. For the second layer, similar to digital modeling, at first, the edges of two arches were covered by bricks then according to the model generated, other bricks were laid. The modeling made it possible for several bricklayers to work adjacent each other, increasing the speed of construction. The entire process of construction was completed in 3 days (Figs. 19, 20).

Conclusion and Future Research

From the experience of the workshop, some advanced requirements for this tool have been found that would make it both more interactive and more automatic. The code must be developed in such a way so that it is possible to distinguish the edges of the surface. The overlap errors would be approximately solved if the gap between the courses is considered with regard to the curvature of the surface. The process of modeling would be accelerated if the overlapping bricks are replaced automatically by special shapes. A step forward for this tool would be developing the code in order to recognize the overlapping bricks, trim the shapes and achieve the brick cutting patterns of each surface, enabling the designer and constructor to estimate the brick requirements before fabrication. Joined with parametric modeling tools, this plug-in can be used as a real-time and feedback design-revision method. In addition, in order to link the tool to real-time software, such as Grasshopper[®], it was decided to convert the code to the brick shells optimization tool which helps designers to provide an optimized brick shell regarding the material requirements and construction time. Developing the code so



Fig. 19 The process of constructing the first layer of the Catalan vault according to the pre-modeled brick pattern



Fig. 20 The process of constructing the second layer of the Catalan vault according to the pre-modeled brick pattern

that it enables the user to predict a precise evaluation of the brick structure's behavior, the reaction of mortar to the loads and probable displacement could also be the future step for this research.

References

- Afsari, K., M. Swarts, and T. Gentry. 2014. Integrated generative technique for interactive design of brickworks. *Journal of Information Technology in Construction (ITcon)* **19**: 225–247.
- Bonwetsch, Tobias, Fabio Gramazio, and Matthias Kohler. 2007. Digitally fabricating non-standardised brick walls. In: *ManuBuild*, Proceedings of the 1st International Conference, Rotterdam, Mark D. Sharp, ed., pp. 191–196. London: CIRIA. http://poelman-partners.nl/download/papers_articles_essays/Manubuild+first+conference+proceedings.pdf, accessed 28 May 2016.
- Como, Mario. 2013. *Statics of Historic Masonry Constructions*. Springer Series in Solid and Structural Mechanics 1. Berlin-Heidelberg: Springer.
- Delougaz, Pinhas. 1933. *Plano-convex Bricks and the Methods of their Employment*. Studies in Ancient Oriental Civilizations 7. Chicago: The Oriental Institute of the University of Chicago.
- Davis, Lara and Philippe Block. 2012. Earthen masonry vaulting: Technologies and transfer. In: *Building Ethiopia, sustainability and innovation in architecture and design*, vol. I, Zegeye Chernet and Helawi Sewnet, eds., pp. 218–231. <http://www.hebel.arch.ethz.ch/wp-content/uploads/2012/10/Vaulting-Davis.pdf>, accessed 28 May 2016.
- Escrig, F. 2006. *The Great Structures in Architecture*. Ashurst, Southampton: WIT Press.
- Fallacara, Giuseppe. 2009. Toward a Stereotomic Design: Experimental Constructions and Didactic Experiences. In: *Proceedings of the Third International Congress on Construction History, Cottbus*. http://www.atelierfallacara.it/_download/Articoli/42.pdf, accessed 28 May 2016.

- Fletcher, Bannister. 1976. *History of Architecture on the Comparative Method*, 17th ed. New York: Charles Scribner's Sons.
- García, Julián, Joaquín Grau, Carlos Martín, Roberto Molinos, and Juan M. Pérez. 2012. Laying out Tile Vaults with Local Positioning Systems. Proceedings of the 3rd Biennial Meeting of The Construction History Society Of America, MIT, 2 November, 2012. http://web.mit.edu/cron/Backup/project/guastavino/www/Garcia-Laying_out_tile_vaults_with_local_positioning_systems.pdf, accessed 28 May 2016.
- Grunbaum, Branko and G. C. Shephard. 1990. *Tilings and Patterns*. New York: W.H. Freeman.
- Imbern, Mattias. 2014. (Re)Thinking the brick: digital tectonic, masonry systems. In: Rethinking Comprehensive Design: Speculative Counterculture, Proceedings of the 19th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2014)/Kyoto 14-16 May 2014, pp. 211–220. http://papers.cumincad.org/data/works/att/caadria2014_237.content.pdf, accessed 28 May 2016.
- Lachauer, Lorenz, Matthias Rippmann and Philippe Block. 2010. Form finding to fabrication: A digital design process for masonry vaults. In: *Proceedings of the International Association for Shell and Spatial Structures Symposium*, 8-12 November 2010, Shanghai. http://www.block.arch.ethz.ch/brg/files/IASS2010_lachauer-rippmann-block.pdf, accessed 28 May 2016.
- Méndez Echenagucia, Tomas, A. Capozzoli, Y. Cascone, and M. Sassone. 2015. The early design stage of a building envelope: Multi-objective search through heating, cooling and lighting energy performance analysis. *Applied Energy* **154**: 577–591.
- Moussavian, E. and R. Gentry, R. 2014. Digital tools for automated generation of vaulted brick assemblies for construction and structural analysis. Presented at the 9th International Masonry Conference, Guimaraes, Portugal, 7–9 July 2014.
- Ochsendorf, John. 2010. *Guastavino Vaulting: The Art of Structural Tile*. New York: Princeton Architectural Press.
- Rajabzadeh, Shaghayegh and Mario Sassone. 2014. The BRICKSHELL Meditation Center: A collaborative Masonry Project. In: *Shells, Membranes and Spatial Structures: Footprints. Proceedings of the IASS-SLTE 2014 Symposium*, 15–19 September 2014, Brasilia, Brazil.
- Rippmann, Matthias, Lorenz Lachauer and Philippe Block. 2012. Interactive Vault Design. *International Journal of Space Structures* **27**(4): 219–30.

Shaghayegh Rajabzadeh is graduated from Department of Architecture and Design (DAD) in Politecnico di Torino, Italy in 2015. Her research is integration between innovative technology and traditional material to adopt the masonry vaulting to contemporary architecture. The objective of her research is to program a new tool for modeling the brick pattern on free-form surfaces. Before joining to the Politecnico di Torino, she was working as an architect in Rahshahr International Group in Tehran, and also as a lecturer in Damavand University, Iran. Currently she is the member of Tehran construction engineering organization and works as a self-employed architect and an independent researcher.

Mario Sassone Assistant Professor at Politecnico di Torino, is member of the 209 Committee of American Concrete Institute and of the International Association of Spatial and Shell Structures (IASS). His research field mainly concerns the computational analysis of time dependent effects behaviour in concrete and steel concrete composite structures, and the structural and architectural optimization of RC shells by means of artificial intelligence techniques. He is currently working on bridging the gap between architectural issues and computational approaches to structural and physical problems. Several members of his research group have been awarded with the IASS HANGAI prize.