



Special Section on CAD & Graphics 2019

3D printed perforated QR codes

Jingru Yang, Hao Peng, Lin Liu, Lin Lu*

School of Computer Science and Technology, Shandong University, Qingdao, China



ARTICLE INFO

Article history:

Received 10 March 2019

Revised 1 April 2019

Accepted 7 April 2019

Available online 4 May 2019

Keywords:

QR codes

Perforated

Digital fabrication

3D printing

ABSTRACT

QR codes are machine-readable optical 2D matrix barcodes which can be easily displayed physically on printed media or digitally on a screen. In recent years, artists and researchers have paid much attention to beautify and enhance QR codes with visual features beyond their plain standard format. We propose a novel method for carving a QR code on a surface in a perforated manner and fabricating it with single material via ordinary 3D printing techniques. However, it is nontrivial to directly fabricate such artistic QR codes because there are a large amount of disconnected components and small details that are difficult to be fully realized. To this end, the embedded information must be consistent and the 3D printed QR code should be readable with standard decoders, like mobile phones. With the original code, we introduce a QR evaluator to estimate its readability and then optimize the code in terms of the printability and robustness. We demonstrate the effectiveness of our techniques with a wide range of examples printed with homogeneous materials via consumer-level 3D printers.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

QR codes (abbreviation for Quick Response Codes) [1] are a type of machine-readable 2D matrix barcodes designed to store data efficiently. They were first applied in 1994 for the automotive industry to allow high-speed component scanning and track vehicles during manufacturing. Applications of QR codes over past decades have been broadened to product tracking, item identification, document management and general marketing (e.g. ticketing, product labeling, mobile payment systems and etc.).

QR codes become popular outside the car industry for its fast readability and increasing storage capacity. While the visual format is far from aesthetic due to pixelating by black and white squares only. Therefore, many researchers have devoted efforts to beautify the QR codes. Approaches to enhance the artistic quality of QR codes are generally grouped into two types: embedding images into the QR codes such as halftone QR codes [2] and stylizing modules in terms of shapes or colors like [3].

Artists design the QR codes in a perforated way and implement them by cutters, taking the material and environment as two tones for decoding, as shown in Fig. 1. Our work is inspired by this aesthetic design form, while also benefited from the emergence of 3D printing [6,7]. We aim at carving the QR code on a surface in a perforated manner automatically and fabricate it using single material via ordinary 3D printing techniques.

This is a challenging task due to a large amount of disconnected components, especially in high version QR codes. New connectors need to be added to make the code fabricable and enhance the stability of the 3D QR model. It is nontrivial to maintain the readability along with such manipulations on the code.

Therefore, we propose an evaluation function to quantitatively measure the readability of the manipulated QR code. The rationale is the redundancy and error correction mechanism for the QR code, i.e., a QR code has the ability to withstand certain extent of damage without information loss. For example, an artistic QR usually sacrifices its readability for enhanced visual appearance.

With this readability function, we can fulfill our goal with an optimization framework. We formulate the manipulation on the given QR code as a constrained optimization problem. Given a QR code and a target surface, we minimize the change on the code and the loss of decoding robustness and carve it on the surface in a perforated manner, with the constraints that the generated 3D QR is 3D printable, sustainable and readable by standard decoders in normal environments.

Our main contributions are:

- We propose a novel approach to automatically generate the fabricable perforated QR codes.
- We introduce an evaluation function to quantitatively measure the readability of the QR code.
- We develop an optimization framework for manipulating the QR code into a 3D printable and readable model, with minimized visual change and loss in decoding robustness.

* Corresponding author.

E-mail address: llu@sdu.edu.cn (L. Lu).

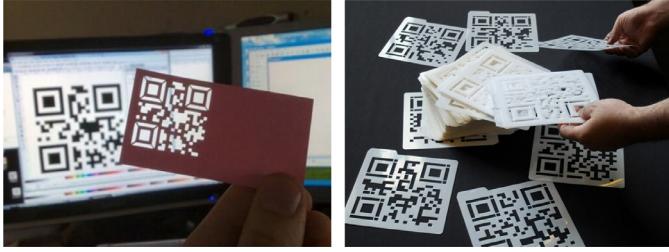


Fig. 1. Perforated QR codes. Left: business card by Wyatt [4]; Right: Digital Nomads by Levin and Foster III [5].

2. Related work

In our work, we enhance the QR code for carving it on the surface and the model with embedded QR code is 3D printed and decoded in normal environments. In the following, we discuss works from both QR code and digital manufacturing domains in our context.

QR-code enhancement. In recent years, researchers have paid much attention to the modification and enhancement of QR codes with visual features beyond their ordinary appearance [8,9]. Cox [10] proposes an algorithm to encode binary image content as redundant numeric strings appended to the original data. Chu et al. [2] firstly combine halftone images and QR codes. They generate a binary embedding by subdividing each QR module into 3×3 submodules to embed the halftone image and then optimize the binary pattern of each module to achieve both decoding robustness and image quality. Lin et al. [3] facilitate the process of embellishing a QR code by embedding images into a QR code using an error-aware warping technique and stylizing the black and white squares using a binary exemplar. Garateguy et al. [11] embed QR codes into color images and optimize the concentration of pixels and its corresponding luminance to minimize a visual distortion metric. Lin et al. [12] further propose a nearly real-time rendering mechanism to improve the visual quality of QR codes while avoiding affecting their decodability.

QR-code images captured by mobile phones are usually distorted, low quality and may consist of nonuniform illumination, noise, and blur. Researchers have made significant efforts to enhance the robustness of the decoding process [13–15]. Efforts are also made to enhance the decoding capabilities and robustness of non-planar QR codes. Li et al. [16] manage to extract the finder patterns and code boundary on a cylindrical surface under the constraints that the two vertical boundaries are parallel to each other and parallel to the generatrix of the cylinder. Lay et al. [17] further lift the constraints and rectify the distortion for QR images posted on cylinders, such that the QR code can be recognized. Our carved QR codes can be scanned and decoded with mobile phone cameras in the same manner as 2D QR-codes are processed. Thus, our method has the same readability as well as camera noise and distortions as the 2D case.

Fabrication-aware carving. Recently, the growing popularity of personalized fabrication has motivated researchers to develop algorithms to stylize 3D shapes via creative or artistic design. Zhou et al. [18] generate single connected ornaments along curves via considering 1D pattern synthesis while constraining the topology. Dumas et al. [19] synthesize fabricable patterns along surfaces by combining structural optimization with appearance optimization. Martinez et al. [20] target the synthesis of flat shapes that are manufactured with laser cutting. Subsequently, Chen et al. [21] synthesized filigree-like structures along surfaces. They relaxed the packing problem by allowing appearance-preserving overlaps between elements. Their method produces appealing patterns, but cannot guarantee the base shape is

preserved. Chen et al. [22] further propose a method for generating packed patterns on a base surface with a set of flat tiles. Zehnder et al. [23] explored the interactive design of curve networks onto surfaces, and simulated the curves as elastic rods, such that a tight packing is achieved via minimized deformations. The produced airy curve networks can be fabricated on high-end printers. These work share the same scenario that taking predefined ornamental pattern as input and performing the structure synthesis.

There are some creative researches on designing or stylizing 3D shapes to generate target images. The shadow art work [24] computes a 3D sculpture that casts several 2D shadows close to the given binary images via geometric optimizations. The perforated lampshades [25] generate continuous projected images on the wall by carving optimized distributed tiny tubes on the shell-like lampshades. More related work can be found in [7]. As far as we know, we are the first that carving the QR code on the surface, considering both fabrication constraints and QR readability.

3D QR codes. There are some attempts that apply QR codes in 3D objects in the context of fabrication. Li et al. [26] introduce AirCode to store data like QR-codes beneath the surface of a 3D printed object using carefully designed air pockets that can be read using computational imaging techniques. Similarly, QR codes are embedded inside 3D objects for identification purposes [27]. By taking advantage of additive layer-by-layer manufacturing, codes are segmented and embedded in numerous object layers without interfering with the surface. Since their pattern lies beneath the surface it suffers from low contrast and is not easily decoded without proper lighting or when printed with challenging matte materials. In contrast, our goal is to embed QR codes with robust readability and minimal geometry distortion. Wei et al. [28] embed QR codes for anti-counterfeiting utilizing SLM-based 3D printing. The QR is fabricated with a specific “tagging” material such that it can be recognized via X-ray imaging. Recently Kikuchi et al. [29] suggested an embedding of QR codes onto CAD models represented by B-spline surfaces. In contrast, our method is not limited to only CAD models or B-spline representations but may be applied in a general manner to various models.

3. Overview

Given an artistic QR code and a target surface, we aim at carving the code on the surface in a perforated manner, such that the QR-embedded model can be 3D printed and decoded by standard decoders in normal environments. The resulting 3D QR code should satisfy the connectivity, structural soundness, and decoding robustness. We minimize the visual change of the QR code with all these constraints.

Taking an input artistic QR code image I (Fig. 2(a)), we extract the embedded QR matrix Q , and recover its original symbol Q_0 via the Reed Solomon algorithm. Then we regard Q_0 as the baseline of the QR code and propose a loss function to quantitatively measure the decoding robustness of the manipulated QR code.

We perform a two-stage optimization on the QR code, a connectivity optimization stage to guarantee the printability, and a structure optimization stage to make sure the printed model is sustainable.

We set the black modules in the QR code as the negative part to be carved out, and the white modules as the positive part that are fabricated in solid. Thus, the white modules must be connected in one piece, otherwise the 3D QR model cannot be fully fabricated. The connectivity optimization first analyzes all white components, and then adds the minimal amount of additional connectors between isolated white components such that all solids connect, considering the readability loss. We employ the minimum spanning tree algorithm to iteratively add the connector until all white modules form one single component (Fig. 2(b)).

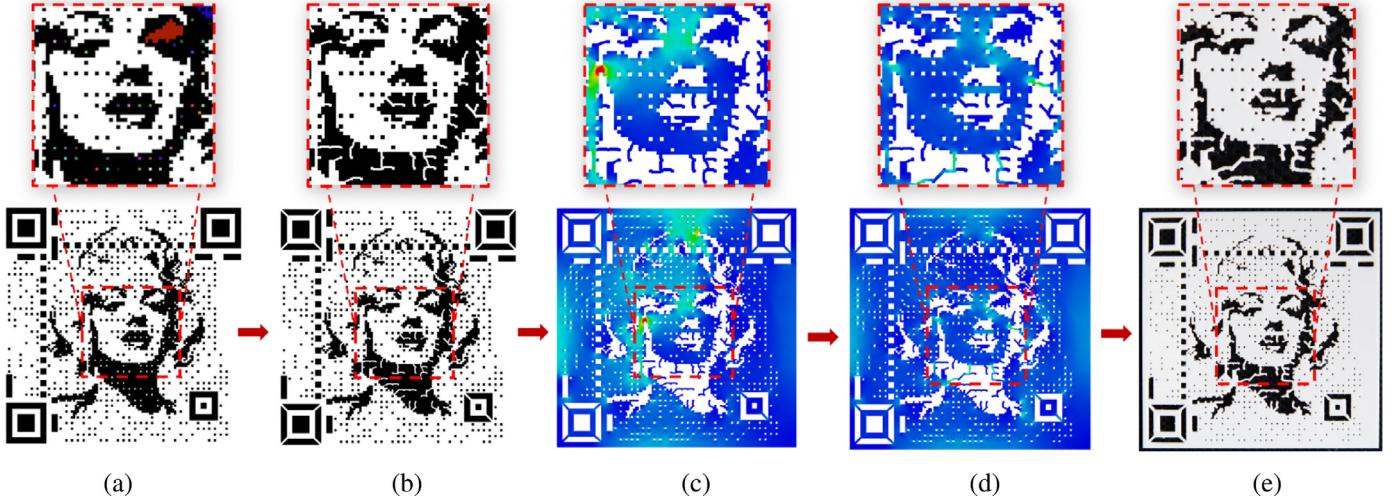


Fig. 2. Given a QR code and a 3D surface, our method first geometrically analyze the QR code, identifying all isolated white components (a). Then the QR code is optimized to guarantee the connectivity of all white components (b). Based on the stress analysis via FEM (c), the structural soundness is iteratively optimized by adding additional connectors while maintaining the readability of the QR code (d). The resulted 3D QR model can be 3D printed and readable by a standard decoder (e).

To guarantee that the perforated model is stable and sustainable, we further optimize the 3D QR code on the surface based on structural analysis. We basically follow the standard workflow of structural optimization, except that the QR readability function is fully involved. We first obtain the stress map and displacement map of the 3D QR model under given forces via finite element methods (FEM) (Fig. 2(c)). We propose a weight computation criteria based on both the distance and readability loss to evaluate the cost of the connector. Then the stability of the whole QR code is iteratively enhanced by adding connectors between the weak region and its neighboring part with the minimal cost (Fig. 2(d)). For each iteration, the readability is evaluated and guaranteed.

Finally, we obtain the fabricable 3D perforated QR code with both structural soundness and decoding robustness (Fig. 2(e)).

4. QR codes evaluation

In this section, we introduce an evaluation function to quantitatively measure the readability of the QR code.

4.1. Preliminaries on QR codes

A standard QR code is an $n \times n$ square matrix, which contains amounts of black and white squares called modules. Each module m denotes a bit of data, and $m = \{0, 1\}$ when the module color is black and white, respectively. The QR code symbol is classified into 40 versions according to the number of modules, i.e., a QR code of version $V = 1, \dots, 40$ includes $(17 + 4 \times V)^2$ modules. The QR code symbol is designed to have the error correction capability and redundancy to recover data when damaged. It has four levels for error correction: *Low*, *Medium*, *Quartile* and *High* that support 7%, 15%, 25% and 30% recovery capacity, respectively.

The structure of a QR code symbol (Fig. 3) takes the form of two parts, the *function patterns* and *encoding regions*.

The function patterns play a pivotal role in the positioning and recognition of QR codes. They include: *position detection patterns* used to recognize QR code and identify QR version; *timing patterns*, enabling module coordinates in the symbol to be determined; and *alignment patterns* used to correct QR code distortions.

The encoding regions contain *data/error correction codewords*, the *version region* (if $V > 6$), the *format information* and *remainder bits* as necessary.

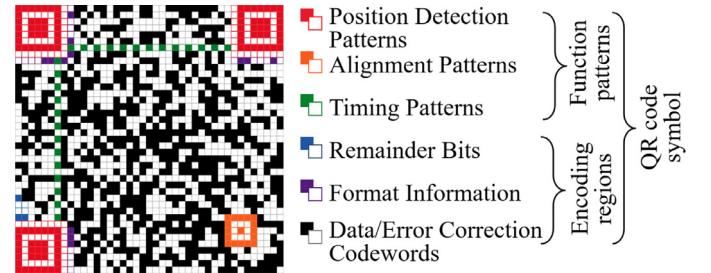


Fig. 3. Structure of a QR code symbol. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

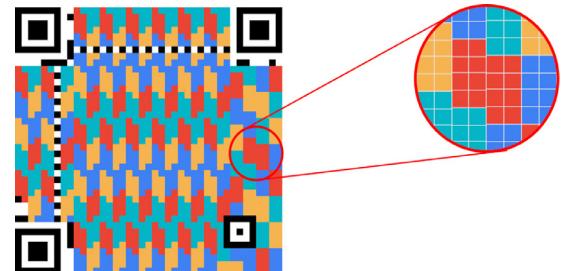


Fig. 4. A version 6, high error correction level QR code has 4 blocks highlighted by different colors.

With regard to the region of data/error correction codewords, it is split into several *blocks* depending on the symbol version and error correction level (see Fig. 4). The data is evenly distributed among each block B , and then the block exploits the Reed Solomon algorithm [1] to generate a series of error correction codewords placed after the data codeword sequence. Each codeword C in data/error correction codewords is formed by eight neighboring modules together. A defect converting one module from dark to light or vice versa will result in the affected symbol character misdecoding as an apparently valid but different codeword. Such error causes a substitution error in the data and requires two error correction codewords to correct it. For withstanding damage without loss of data, the *maximum number of erroneous codewords* r must be less than half of the error correction codewords number per block.

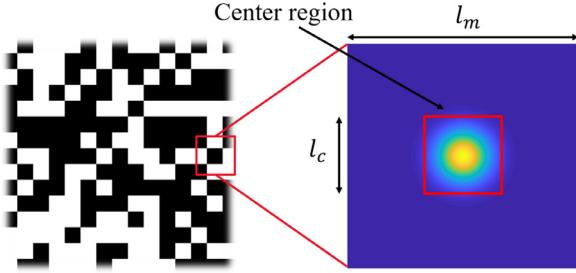


Fig. 5. Left: a module in QR code highlighted in red. Right: diagram of this module showing possible sampling points hit in it and its center region using the Gaussian function. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

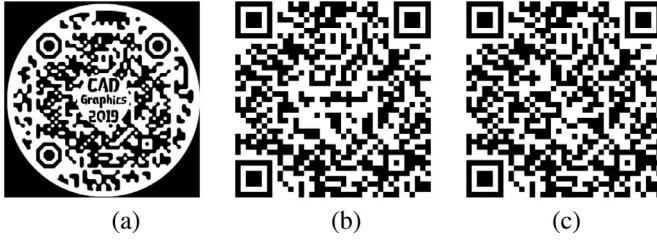


Fig. 6. (a) Input artistic QR code image; (b) the QR code Q extracted from the input; (c) the original QR code Q_0 recovered from Q .

In general, a QR code symbol has four format regions to provide sufficient redundancy. Each format region F is a set of neighboring modules (see Fig. 3, format regions are highlighted in purple). Similar to the codeword, format region F will be damaged by any erroneous of a module. The remainder bits padded into the encoding region do not carry information, just to fill data codeword capacity of the symbol when necessary.

4.2. QR codes recovery

Taking an artistic QR code image I as input, we first binarize the image and then take an open source library Zxing [30] to detect the *position detection patterns* and *alignment patterns* of QR code. The position detection pattern can be recognized easily for it construes a dark-light-dark-light-dark sequence the relative widths of each module of which are in the ratios 1:1:3:1:1.

Once all position detection patterns identified, we measure the module width l_m via drawing parallel lines and then extract the module matrix in the QR code region. To safely extract module matrix, Lin et al. [3] suggest drawing a center region for each module that contains the correct information. The width of the center region l_c shall be at least 1/3 of module width (Fig. 5). A Gaussian distribution of sampling points proposed in [11] is adopted to address this requirement and then we extract the QR matrix by sampling on a grid (Fig. 6(b)).

Let Q be the QR module matrix extracted from the input QR image. We recover the original QR code Q_0 from Q , which is regarded as the baseline of Q via Reed Solomon algorithm [1] (Fig. 6(c)).

4.3. Loss of QR readability

We define $L(Q)$ as the loss of QR readability caused by the difference between Q and Q_0 . Specifically, only errors occur in the data/error correction codewords or the format regions could affect QR code readability as reported on the QR standard documents. Let ΔQ denote QR code errors. The loss of QR readability $L(Q)$ can be described as a total loss of data/error correction capability L_C and

format regions redundancy L_F :

$$L(Q) = L_C + L_F. \quad (1)$$

We denote each codeword in the form of $C = \{m_i\}_{i=1}^8$ and define ΔC as an indicator function to indicate whether C is erroneous or not. $\Delta C = 1$ indicates the total square deviation of this codeword $\sum_{i=1}^8 \Delta m_i^2 > 0$, otherwise $\Delta C = 0$. In order to measure the loss caused by the erroneous codewords, we formulate the loss function of data/error correction codewords L_C as:

$$L_C = \begin{cases} \frac{1}{n_B} \sum_{i=1}^{n_B} \left(\frac{1}{r} \sum_{j=1}^{n_C} \Delta C_{ij} \right)^2, & \forall B_i \subset Q, \sum_{j=1}^{n_C} \Delta C_{ij} \leq r \\ \infty, & \exists B_i \subset Q, \sum_{j=1}^{n_C} \Delta C_{ij} > r \end{cases} \quad (2)$$

where n_B is the number of blocks, and each block has n_C codewords. Here we propose a min-max normalization for each block to scale the loss per block in $[0, 1]$ and L_C is to be infinity if the error number of a block exceeds r .

Let ΔF be an indicator of whether this format region is damaged ($= 1$) or not ($= 0$). Let $\Delta F = 1$ when the total square deviation of $F \sum_{m_i \in F} \Delta m_i^2 > 0$, while $\Delta F = 0$ when $\sum_{m_i \in F} \Delta m_i^2 = 0$.

To guarantee QR code readability, the damage on format regions needs to be as little as possible and at least one format region cannot be changed. Thus, we formulate the loss of format regions redundancy L_F as:

$$L_F = \begin{cases} \frac{1}{4} \sum_{i=1}^4 \Delta F_i, & \sum_{i=1}^4 \Delta F_i < 4 \\ \infty, & \sum_{i=1}^4 \Delta F_i = 4 \end{cases} \quad (3)$$

Therefore, the loss of QR readability $L(Q)$ can be computed by (Eqs 2) and (3).

5. QR codes optimization

We propose a two-stage optimization to automatically generate the connected and stable perforated QR codes.

5.1. Connectivity optimization

To ensure perforated QR codes printability, a connectivity optimization algorithm shall be put forward to connect whole white components into a single component. Otherwise, the isolated white regions suspending in the void lead to a printing failure.

We divide the connectivity optimization into two steps based on the QR code structure. For function patterns, we follow the diagonal design done by the artist as shown in Fig. 1 to make the white components connected (Fig. 7). For encoding regions, we build an iterative optimization algorithm to connect the isolated white components.

A straightforward way to obtain a single connected image is to link among isolated white components via minimal Euclidean distance using the spanning tree algorithm. It makes the minimum change on the visual appearance of the QR code. However, this may cause decoding failures or decrease the decoding robustness. To ensure the readability, the effect on the QR codes error correction capabilities must be considered. Therefore, we consider both Euclidean distance and the cost of QR codes error correction capabilities in the connectivity optimization on the encoding regions. Given a QR image I , we label all white components $\{S_i\}_{i=1}^N$. Let p and q be the points on the contours of S_i and S_j respectively. Then we formulate the weight w_{ij} between S_i and S_j as:

$$w_{ij} = \min_{p, q} (\lambda L(Q) + \|p - q\|^2), \quad (4)$$

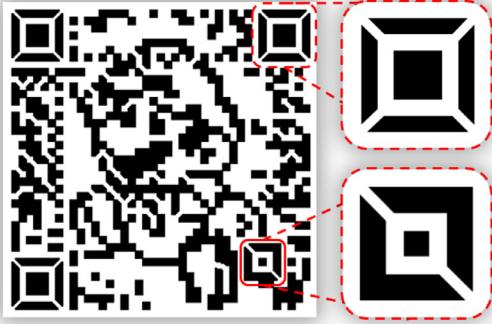


Fig. 7. The connectivity treatment for the position detection patterns and alignment patterns.

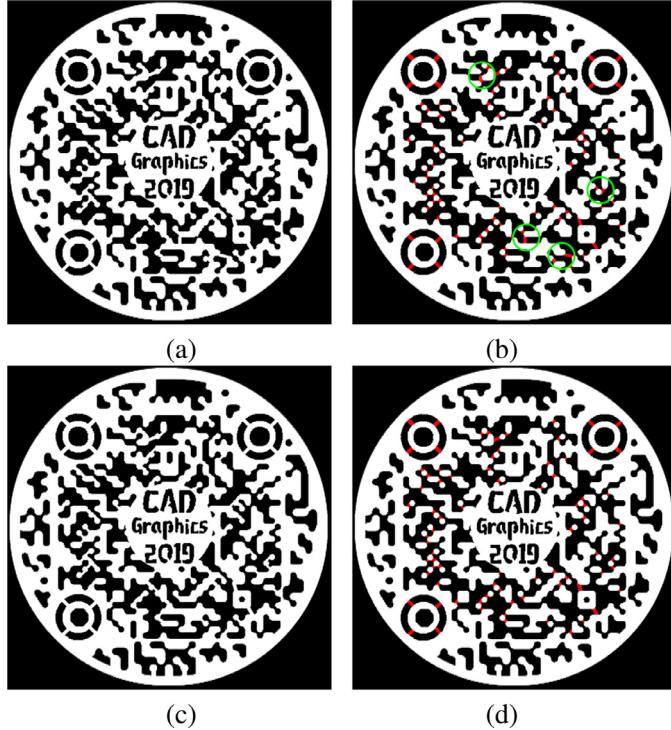


Fig. 8. (a) The connectivity enforcement by minimal Euclidean distance without considering the QR readability ($\lambda = 0$ in Eq. (4)) cannot be decoded. The new connectors are highlighted in red (b). (c) The connectivity optimization result considering the QR readability ($\lambda = 1$ in Eq. (4)) with new connectors highlighted in red (d). The green circle marks the difference between (b) and (d). The input QR code is in Fig. 6. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $L(Q)$ is defined in Section 4.3, indicating the loss on the readability when adding the connector between p and q , λ is the weight parameter for the readability.

We note that there is no need to compute weights of all white component pairs. For each point $p \in S_i$, we just cast its 2D visible region V_p and calculate the weight of p and $q \in V_p \cap S_j$ to reduce the computational cost. Then we randomly pick out a white component and connect its neighbor with the minimal weight iteratively until all white components are connected.

Fig. 8 shows an example on connectivity optimization, together with the comparison result using the shortest Euclidean distance path. The change of the readability loss value is plotted in Fig. 9. The loss value increases along with the iterations. However, it goes

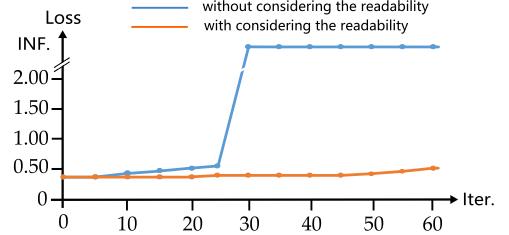


Fig. 9. The readability loss along with the connectivity optimization iterations.

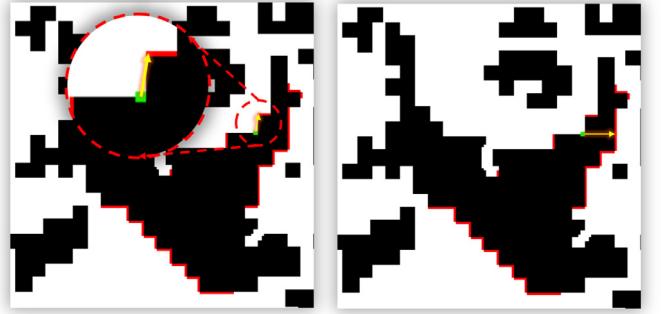


Fig. 10. Selection of the connectors in structural optimization. The starting point is marked green, all possible connecting points are marked red and the yellow line is the connector with the shortest path. Left: only the Euclidean distance is adopted; Right: both Euclidean and geodesic distance are adopted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to infinity value if only considering the distance in generating the connectors, causing the QR code nondecodable.

5.2. Structural optimization

We first carve the QR code on the surface and generate the mesh. Specifically, we project the optimized QR code image from Section 5.1 on the surface with target QR position defined by users. We assume that the carving region is continuous, with no self occlusions and that the projection direction is in the normal direction to the surface as QR codes are typically scanned from top-view. The projected QR image on the surface defines a partition of the surface vertices inside the QR region into black and white modules. The black modules are hollowed out as the negative part and white modules are the remaining solid. Note that we remesh the QR region, i.e., white modules, on the surface into a dense set of triangles.

Then we perform the static analysis via FEM on the shell model, in which the shell thickness is taken as a parameter. We fix the QR boundaries as the boundary condition and apply uniform pressure on each face along its normal direction. The stress map and displacement map of the QR code are obtained for the optimization.

Similar to connectivity optimization, both distance and the cost of QR readability shall be considered. In particular, the new connector should comprise the shortest connecting path and cause the minimum readability loss. Refer to Fig. 10, endpoints of the new connector should be on the boundary of the white component and cross the black component. It is clear that the Euclidean distance is not enough to evaluate the cost of the connector (Fig. 10(a)). Intuitively, a good connector is made by two points with the shortest path in Euclidean distance, but the largest path in geodesic distance (Fig. 10(b)).

Therefore, we propose a criteria for the connectors based on both the distance and readability loss to evaluate the cost of the connector as follows. The connector by points p and q has the cost

Table 1

Statistics of the results.

| Model | QR image | Resolution | QR version | Printed QR size (cm) | Connectivity optimization time (s) and #Iter. | Structural optimization time (s) and #Iter. |
|--------|-------------|------------|------------|----------------------|---|---|
| Cuboid | Flower | 700 × 700 | 8 | 14 × 14 × 0.2 | 842 and 140 | 387 and 7 |
| Cuboid | Monroe | 640 × 640 | 6 | 13 × 13 × 0.2 | 456 and 79 | 156 and 3 |
| Cuboid | Einstein | 600 × 600 | 5 | 12 × 12 × 0.2 | 2465 and 493 | 588 and 12 |
| Cuboid | Mario | 260 × 260 | 6 | 5 × 5 × 0.2 | 179 and 35 | 232 and 6 |
| Cuboid | CADGraphics | 370 × 370 | 5 | 9 × 9 × 0.2 | 336 and 61 | 168 and 4 |
| Bunny | | | | 18.9 × 18.2 × 14.6 | | |

 c_{pq} :

$$c_{pq} = \lambda L(Q) + \frac{\|p - q\|^2}{\|p, q\|_g}, \quad (5)$$

where $\|p, q\|_g$ indicates the geodesic distance between p and q .

In the structural optimization, we pick the point with the largest deformation after the FEM analysis and add the connector from this point to the other with the smallest cost defined in (Eq 5). Then we update the model and perform FEM until the whole model is free of weak regions, i.e., all stress is below the yielding stress value χ . In our experiments, $\chi = 10$ MPa for the PLA material. The detailed algorithm is given in [Algorithm 1](#).

Algorithm 1 Structural optimization on the perforated QR.**Input:** The perforated model M , yielding stress value χ ;**Output:** The optimized model;

```

1: Compute the stress map  $\sigma$  and strain map  $\epsilon$  of  $M$  via FEM;
2: while  $\sigma_{max} > \chi$  do
3:   Find maximum strain point  $p$ ,  $c_{min} \leftarrow \infty$ ;
4:   Compute visible points  $V_p$  of  $p$  on the white contour;
5:   for  $q \in V_p$  do
6:     Compute  $c_{pq}$  following Eq. (5);
7:     if  $c_{pq} < c_{min}$  then
8:        $c_{min} \leftarrow c_{pq}$ ;
9:     end if
10:   end for
11:   if  $c_{min} = \infty$  then
12:     return the current model;
13:   end if
14:   Connect line  $pq$  where  $c_{pq} = c_{min}$ , update  $M$ ,  $\sigma$  and  $\epsilon$ ;
15: end while
16: return the stable model  $M$ .

```

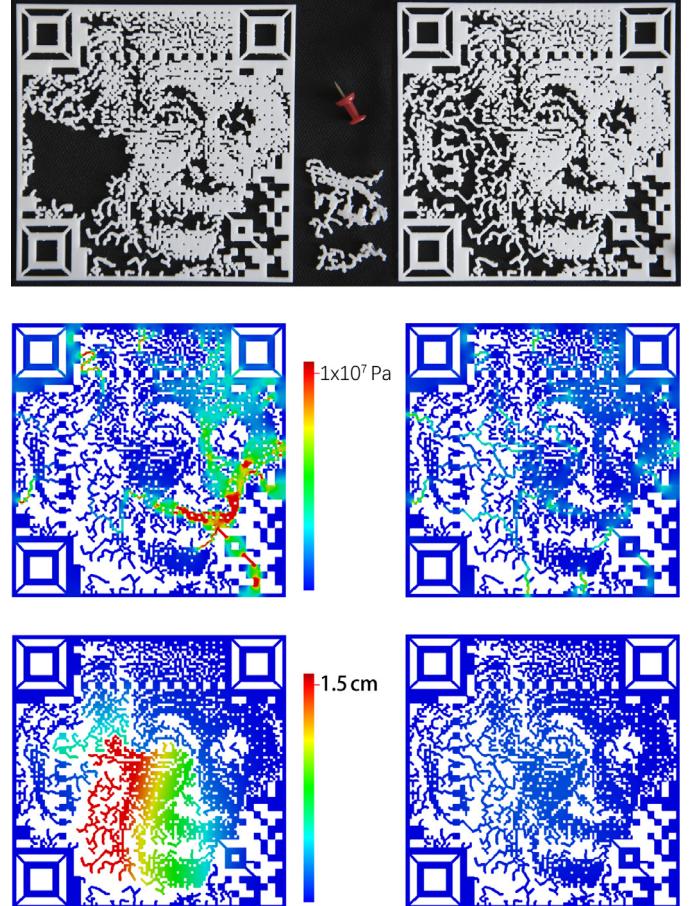
[Fig. 11](#) shows an example of structural optimization, in which the physical size of the QR code is 12×12 cm, the thickness is 0.5 cm and the pressure on each face is 5 Pa.

6. Results and discussions

We implemented our approach in C++ on an Intel Core™ i7-7700K CPU@4.0 GHz and 16 GB RAM. We use OpenCV library [31] for image processing, CGAL [32] for mesh manipulation, and Abaqus Unified FEA for stress analysis in the implementation.

We show the input QR codes, the optimized codes and 3D printed results of the carving QR codes in [Fig. 15](#). The models are fabricated by an FDM-based printer (HORI Z500) with white PLA material, whose nozzle diameter is 0.4 mm. Statistics of the results including QR resolution and version, printing size, running time and the number of iterations are listed in [Table 1](#).

As shown in [Table 1](#), both connectivity optimization and structural optimization perform well for different versions of QR codes. Especially, our algorithm works for the artistically enhanced QR codes generated by previous work. For connectivity optimization, the number of iterations is determined by the number of isolated



[Fig. 11](#). First row: the 3D printed QR code without structural optimization (left) and with structural optimization (right). The second row shows the related stress map and the third row shows the related displacement map.

white regions. It takes the Einstein code 493 iterations for generating the single connected white region due to a large number of fractures for the halftone effect. For structural optimization, the number of iterations corresponds to the structure of the QR code. Since the entire QR code needs to be evaluated to update the loss in each iteration, the time for each iteration is similar. The loss evaluation is also the most time-consuming part in the optimization, which relates to the given QR resolution.

To validate the readability of the fabricated carving QR codes, we designed and completed four experiments to evaluate the robustness of our QR codes. We regard a decoding as a successful one if the QR code is decoded within 2 s.

First, we use different mainstream mobile devices standard QR readers to test the decoding success rate of QR codes under the black background. If the decoding is successful, we mark it with ✓, otherwise we mark it with ✗ in [Table 2](#). It can be seen from the

Table 2
Readability test on different decoders.

| | Flower | Monroe | Einstein | Mario | CADGraphics (cuboid) | CADGraphics (bunny) |
|--------------------------|--------|--------|----------|-------|----------------------|---------------------|
| iPhone and QuickMark | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| iPhone and NeoReader | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Samsung and WeChat | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Mi8 and Scanner | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Huawei and Huawei Vision | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |



Fig. 12. Perforated QR models in different background colors can be decoded.



Fig. 13. Perforated QR models printed in different colors can be decoded.

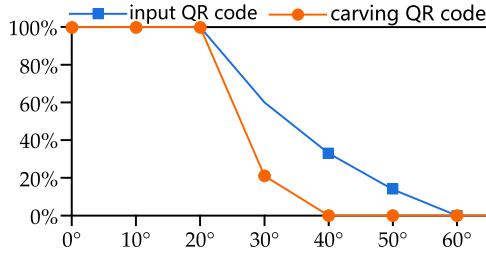


Fig. 14. Robustness on the scanning angles for the Einstein QR code.

data in Table 2 that carving QR codes can be decoded successfully by mainstream decoders and mobile phones.

Second, we validate the readability of the fabricated QR codes over different background colors. We scan each QR code 20 times in Fig. 15 under six specific background colors (red, green, blue, cyan, magenta, yellow), 120 times for each color. The corresponding success rate are 95.8%, 96.7%, 98.3%, 95.8%, 97.5% and 94.2% for the six background colors, respectively. As the QR codes are all printed in white materials, dark background colors indicate high contrast and thus raise the successful rate. Fig. 12 shows the Mario model in three background colors can all be successfully decoded.

Third, we tested our method on different printing materials. Our method is friendly to various materials as long as the printed and background color produce enough contrast. Fig. 13 shows the Mario model in different colors of the PLA material, which can all be easily decoded.

Finally, we tested the robustness of the printed QR codes to different scanning angles. For the Einstein QR code, we scan both the input QR image and the perforated QR code on the cuboid by Mi8 Scanner 20 times from a range of scanning angles and list the average statistics in Fig. 14. It shows that the original input QR codes

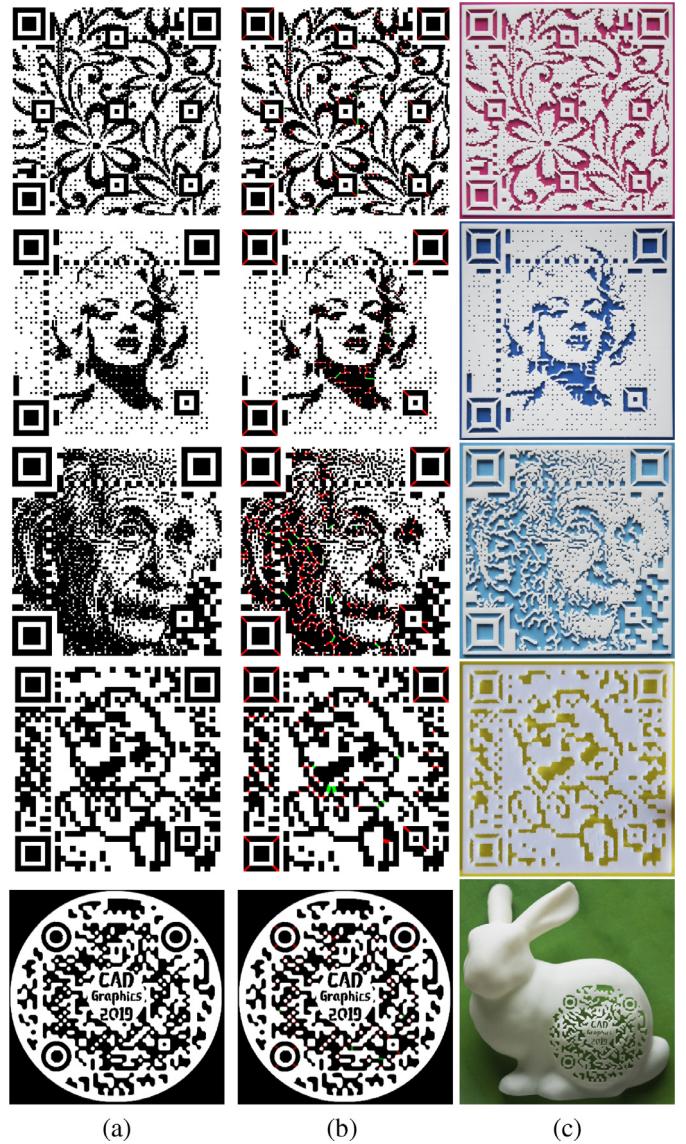


Fig. 15. (a) Input artistic QR code. (b) The optimized QR code, in which the connectors introduced in connectivity optimization are highlighted in red, connectors introduced in structural optimization are highlighted in green. (c) 3D printed QR code. The Flower, Monroe, and Einstein in the first three rows are generated by Chu et al. [2]. The Mario code is generated by Cox [10]. The original pictures in the first four QR codes are from <https://project-tideas.com/famous-clipart-patterns.html> and <https://www.shutterstock.com/image-vector/>. The CADGraphics code is generated by <https://www.unitag.io/qrcode>. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

possess robustness within 30° tilt from the orthogonal direction to QR codes. There is a clear reducing trend of decoding success rate with the increase of the scanning angle. Especially the QR codes cannot be decoded when the tilt angle is greater than 60°. Comparing to the original QR code, our 3D printed carving QR code

allows smaller tilt, which has good robustness within 20° of tilt. It is difficult to decode successfully when the tilt angle is greater than 30°. The reason is that the sidewall introduces self-occlusion to QR code. We remark that the CADGraphics code on the Bunny model needs to be scanned from the top view.

7. Conclusion and future work

In this paper, we introduce a method for manipulating the QR code into a 3D printable and readable model, with minimized visual change and loss of decoding robustness. We propose a significant QR code evaluator to evaluate the loss of QR readability throughout the whole optimization, which is used to ensure readability. Then we discuss two major optimization components to obtain the fabricable carving QR code with structural soundness automatically. One is the connectivity optimization to make sure that the QR code is printable and decoding robustness. The other is the structural optimization to generate decodable and stable 3D carving QR code. Finally, we validated the effectiveness of our method on a series of QR codes and evaluate the robustness of the resulting QR codes in various aspects.

For the future work, we would like to investigate on the perforated QR code on disconnected regions. Exploring the functionalities of the 3D printed QR code is also interesting.

Declaration of competing interest

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

Acknowledgments

We thank all the anonymous reviewers for their valuable suggestions. This work is supported by NSFC (61572291) and Young Scholars Program of Shandong University (YSPSDU).

References

- [1] Wave D. Information technology automatic identification and data capture techniques QR code bar code symbology specification. International Organization for Standardization; 2015. ISO/IEC 18004.
- [2] Chu H-K, Chang C-S, Lee R-R, Mitra NJ. Halftone QR codes. ACM Trans Graph 2013;32(6):217:1–217:8. doi:[10.1145/2508363.2508408](https://doi.org/10.1145/2508363.2508408).
- [3] Lin Y-S, Luo S-J, Chen B-Y. Artistic QR code embellishment. Comput Graph Forum 2013;32(7):137–46. doi:[10.1111/cgf.12221](https://doi.org/10.1111/cgf.12221).
- [4] Wyatt T. QR code business card. 2008. URL <https://www.flickr.com/photos/fucksocks/2425729926>.
- [5] Levin G., Foster III A. QR codes for Digital Nomads. 2011. URL <http://www.flong.com/projects/qr-codes-for-digital-nomads/>.
- [6] Bo Z., Lu L., Sharf A., Xia Y., Deussen O., Chen B. Printable 3D trees. 2017. ;<https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13269>. 10.1111/cgf.13269
- [7] Bernd B, Paolo C, Luigi M, Nico P. State of the art on stylized fabrication. Comput Graph Forum 2018;37(6):325–42. doi:[10.1111/cgf.13327](https://doi.org/10.1111/cgf.13327).
- [8] Visualead. Visualead: visual QR code generator. 2019. URL <http://www.visualead.com/>.
- [9] FLICKR. QR code art. <http://www.flickr.com/groups/qr-art/>; 2009.
- [10] Cox R. Qart codes. <http://research.swtch.com/qart>; 2012.
- [11] Garateguy GJ, Arce GR, Lau DL, Villarreal OP. QR images: optimized image embedding in QR codes. IEEE Trans Image Process 2014;23(7):2842–53. doi:[10.1109/TIP.2014.2321501](https://doi.org/10.1109/TIP.2014.2321501).
- [12] Lin SS, Hu MC, Lee CH, Lee TY. Efficient QR code beautification with high quality visual content. IEEE Trans Multimed 2015;17(9):1515–24. doi:[10.1109/TMM.2015.2437711](https://doi.org/10.1109/TMM.2015.2437711).
- [13] Chu C-H, Yang D-N, Chen M-S. Image stabilization for 2D barcode in handheld devices. In: Proceedings of the fifteenth ACM international conference on multimedia, MM '07. New York, NY, USA: ACM; 2007. p. 697–706. ISBN 978-1-59593-702-5. doi:[10.1145/1291233.1291394](https://doi.org/10.1145/1291233.1291394).
- [14] Yang H, Kot AC, Jiang X. Binarization of low-quality barcode images captured by mobile phones using local window of adaptive location and size. IEEE Trans Image Process 2012;21(1):418–25. doi:[10.1109/TIP.2011.2155074](https://doi.org/10.1109/TIP.2011.2155074).
- [15] van Gennip Y, Athavale P, Gilles J, Choksi R. A regularization approach to blind deblurring and denoising of QR barcodes. IEEE Trans Image Process 2015;24(9):2864–73. doi:[10.1109/TIP.2015.2432675](https://doi.org/10.1109/TIP.2015.2432675).
- [16] Li X, Shi Z, Guo D, He S. Reconstruct algorithm of 2D barcode for reading the QR code on cylindrical surface. In: Proceedings of the 2013 international conference on anti-counterfeiting, security and identification (ASID); 2013. p. 1–5. doi:[10.1109/ICASID.2013.6825309](https://doi.org/10.1109/ICASID.2013.6825309).
- [17] Lay KT, Wang LJ, Han PL, Lin YS. Rectification of images of QR codes posted on cylinders by conic segmentation. In: Proceedings of the 2015 IEEE international conference on signal and image processing applications (ICSIPA); 2015. p. 389–93. doi:[10.1109/ICSIPA.2015.7412222](https://doi.org/10.1109/ICSIPA.2015.7412222).
- [18] Zhou S, Jiang C, Lefebvre S. Topology-constrained synthesis of vector patterns. ACM Trans Graph 2014;33(6):215:1–215:11. doi:[10.1145/2661229.2661238](https://doi.org/10.1145/2661229.2661238).
- [19] Dumas J, Lu A, Lefebvre S, Wu J, Dick C. By-example synthesis of structurally sound patterns. ACM Trans Graph 2015;34(4):137:1–137:12. doi:[10.1145/2766984](https://doi.org/10.1145/2766984).
- [20] Martinez J, Dumas J, Lefebvre S, Wei L-Y. Structure and appearance optimization for controllable shape design. ACM Trans Graph 2015;34(6):229:1–229:11. doi:[10.1145/2816795.2818101](https://doi.org/10.1145/2816795.2818101).
- [21] Chen C, Huang W, Zhou B, Liu C, Mow WH. Picode: a new picture-embedding 2D barcode. Trans Image Proc 2016;25(8):3444–58. doi:[10.1109/TIP.2016.2573592](https://doi.org/10.1109/TIP.2016.2573592).
- [22] Chen W, Ma Y, Lefebvre S, Xin S, Martinez J, Wang W. Fabricable tile decors. ACM Trans Graph 2017;36(6):175:1–175:15. doi:[10.1145/3130800.3130817](https://doi.org/10.1145/3130800.3130817).
- [23] Zehnder J, Coros S, Thomaszewski B. Designing structurally-sound ornamental curve networks. ACM Trans Graph 2016;35(4):99:1–99:10. doi:[10.1145/2897824.2925888](https://doi.org/10.1145/2897824.2925888).
- [24] Mitra NJ, Pauly M. Shadow art. ACM Trans Graph 2009;28(5):156:1–156:7. doi:[10.1145/1618452.1618502](https://doi.org/10.1145/1618452.1618502).
- [25] Zhao H, Lu L, Wei Y, Lischinski D, Sharf A, Cohen-Or D, et al. Printed perforated lampshades for continuous projective images. ACM Trans Graph 2016;35(5):154:1–154:11. doi:[10.1145/2907049](https://doi.org/10.1145/2907049).
- [26] Li D, Nayar SK, Zheng C. Aircode: unobtrusive physical tags for digital fabrication. In: Proceedings of the thirtieth annual ACM symposium on user interface software and technology, UIST '17; 2017. p. 449–60.
- [27] Chen F, Luo Y, Tsoutsos NG, Maniatakos M, Shahin K, Gupta N. Embedding tracking codes in additive manufactured parts for product authentication. Adv Eng Mater 2018;0(0):1800495. doi:[10.1002/adem.201800495](https://doi.org/10.1002/adem.201800495).
- [28] Wei C, Sun Z, Huang Y, Li L. Embedding anti-counterfeiting features in metallic components via multiple material additive manufacturing. Addit Manuf 2018;24:1–12. doi:[10.1016/j.addma.2018.09.003](https://doi.org/10.1016/j.addma.2018.09.003).
- [29] Kikuchi R, Yoshikawa S, Jayaraman PK, Zheng J, Maekawa T. Embedding QR codes onto B-spline surfaces for 3D printing. Comput-Aided Des 2018;102:215–23. Special Issue on SPM 2018. doi: [10.1016/j.cad.2018.04.025](https://doi.org/10.1016/j.cad.2018.04.025)
- [30] Owen S, et al.. Zxing-multi-format 1D/2D barcode image processing library with clients for Android, Java. Github; 2014.
- [31] Open source computer vision library. <https://github.com/opencv>; 2015.
- [32] The CGAL Project. CGAL user and reference manual. 412. CGAL Editorial Board; 2018.