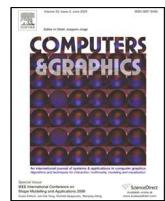




Contents lists available at ScienceDirect

Computers &amp; Graphics

journal homepage: [www.elsevier.com/locate/cag](http://www.elsevier.com/locate/cag)

## 3D printed perforated QR codes

---

### ARTICLE INFO

---

*Article history:*

Received January 20, 2019

**Keywords:** QR Codes, Perforated, Digital Fabrication, 3D Printing

---

### ABSTRACT

We propose a novel method for carving a QR code on a surface in a perforated manner and fabricating it using single material with ordinary 3D printing techniques. 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. However, it is nontrivial to directly fabricate such artistic QR codes as there usually is a large amount of disconnected components and small details that can hardly be fully realized. The hard constraints include that the embedded information must be consistent and the 3D printed QR code is readable with standard decoders, like mobile phones. To achieve this goal, 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 using consumer-level 3D printers and common homogeneous materials.

© 2019 Elsevier B.V. All rights reserved.

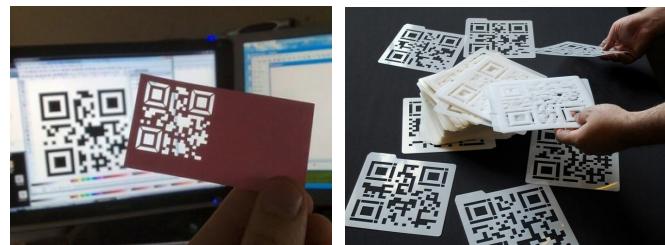
---

### 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: (a) embedding images into the QR codes such as halftone QR codes [2] and (b) stylizing modules in terms of shapes or colors like [3].

Artists designed the QR codes in a perforated way and implemented them by cutters, taking the material and environment as



**Fig. 1. Perforated QR codes. Left: Business card by Rodion Kovenkin; Right: Digital Nomads by Golan Levin and Asa Foster III.**

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. We aim to carve the QR code on a surface in a perforated manner automatically and fabricate it using single material with ordinary 3D printing techniques.

This is surely a challenging task. Specifically, there might be a large amount of disconnected components especially for high version QR codes. New connectors need to be added to make the code fabricable. Actually, connectors are also required to enhance the stability of the 3D QR model. It is nontrivial to

1 maintain the readability along with such manipulations on the  
2 code.

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

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

18 Our main contributions are:

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

## 27 2. Related work

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

33 *QR-code enhancement.* In recent years, researchers have  
34 paid much attention to the modification and enhancement of  
35 QR-codes with visual features beyond their ordinary appearance  
36 [4, 5]. Cox [6] propose an algorithm to encode binary image  
37 content as redundant numeric strings appended to the original  
38 data. Chu et al. [2] firstly combine halftone images and QR  
39 codes. They generate a binary embedding by subdividing each  
40 QR module into  $3 \times 3$  submodules to embed the halftone image  
41 and then optimize the binary pattern of each module to achieve  
42 both decoding robustness and image quality. Lin et al. [3] facilitate  
43 the process of embellishing a QR code by embedding images  
44 into a QR code using an error-aware warping technique  
45 and stylizing the black and white squares using a binary exemplar.  
46 Garateguy et al. [7] embed QR codes into color images  
47 and optimize the concentration of pixels and its corresponding  
48 luminance to minimize a visual distortion metric. Lin et al. [8]  
49 further propose a nearly real-time rendering mechanism to im-  
50 prove the visual quality of QR codes while avoiding affecting  
51 their decodability.

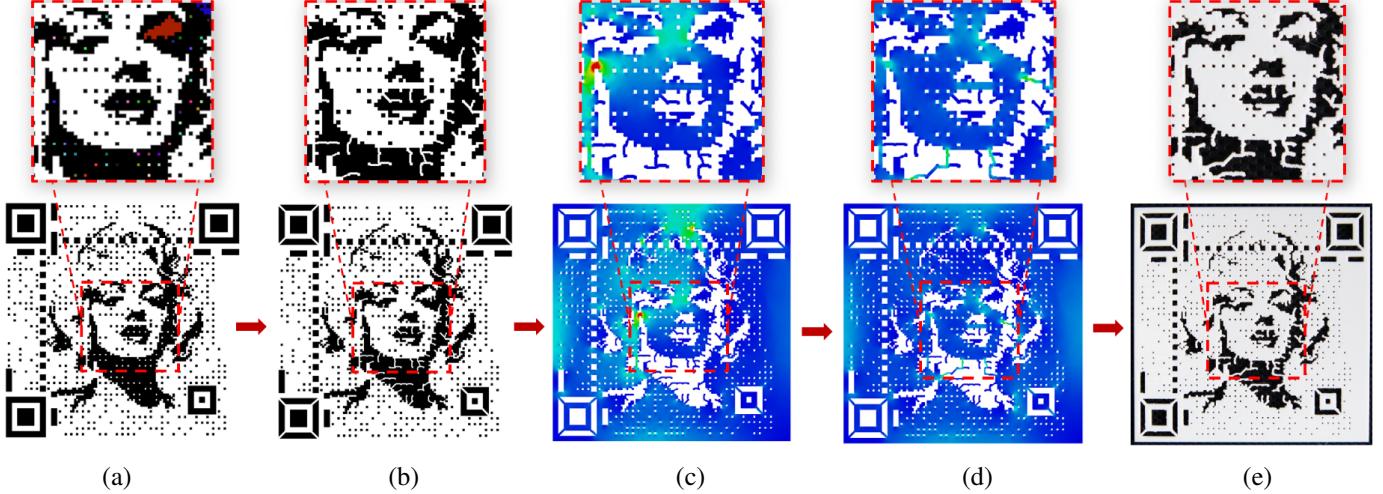
52 QR-code images captured by mobile phones are usually dis-  
53 torted, low quality and may consist nonuniform illumination,

54 noise and blur. Researchers have made significant efforts to en-  
55 hance the robustness of the decoding process [9, 10, 11]. Efforts  
56 are also made to enhance the decoding capabilities and robust-  
57 ness of non-planar QR codes. Li et al. [12] manage to extract  
58 the finder patterns and code boundary on a cylindrical surface  
59 under the constraints that the two vertical boundaries are paral-  
60 lel to each other and parallel to the generatrix of the cylinder.  
61 Lay et al. [13] further lift the constraints and rectify the distor-  
62 tion for QR images posted on cylinders, such that the QR code  
63 can be recognized. Our carved QR codes can be scanned and  
64 decoded with mobile phone cameras in the same manner as 2D  
65 QR-codes are processed. Thus, our method has the same read-  
66 ability as well as camera noise and distortions as the 2D case.

67 *Fabrication-aware carving.* Recently, the growing popular-  
68 ity of personalized fabrication has motivated researchers to de-  
69 velop algorithms to stylize 3D shapes via creative or artistic  
70 design. Zhou et al. [14] generated single connected ornaments  
71 along curves via considering 1D pattern synthesis while con-  
72 straining the topology. Dumas et al. [15] synthesized fabrica-  
73 ble patterns along surfaces by combining topology optimization  
74 with appearance optimization. Martinez et al. [16] targeted the  
75 synthesis of flat shapes that are manufactured with laser cut-  
76 ting. Subsequently, Chen et al. [17] synthesized filigree-like  
77 structures along surfaces. They relaxed the packing problem  
78 by allowing appearance-preserving overlaps between elements.  
79 Their method produces appealing patterns, but cannot guaran-  
80 tee the base shape is preserved. Chen et al. [18] further pro-  
81 posed a method for generating packed patterns on a base surface  
82 with a set of flat tiles. Zehnder et al. [19] explored the interac-  
83 tive design of curve networks onto surfaces, and simulated the  
84 curves as elastic rods, such that a tight packing is achieved via  
85 minimized deformations. The produced airy curve networks  
86 can be fabricated on high-end printers. These work share the  
87 same scenario that taking predefined ornamental pattern as an  
88 input and performing the structure synthesis.

89 There are some creative researches on designing or stylizing  
90 3D shapes to generate target images. The shadow art work [20]  
91 computes a 3D sculpture that casts several 2D shadows close to  
92 the given binary images via geometric optimizations. The per-  
93forated lampshades [21] generate continuous projected images  
94 on the wall by carving optimized distributed tiny tubes on the  
95 shell-like lampshades. More related work can be found in [22].  
96 As far as we know, we are the first that carving the QR code  
97 on the surface, considering both fabrication constraints and QR  
98 readability.

99 *3D QR codes.* There are some attempts that apply QR codes  
100 in 3D objects in the context of fabrication. Li et al. [23] in-  
101 troduce AirCode to store data like QR-codes beneath the sur-  
102 face of a 3D printed object using carefully designed air pockets  
103 that can be read using computational imaging techniques. Sim-  
104ilarly, QR codes are embedded inside 3D objects for identifi-  
105 cation purposes [24]. By taking advantage of additive layer-  
106 by-layer manufacturing, codes are segmented and embedded  
107 in numerous object layers without interfering with the surface.  
108 Since their pattern lies beneath the surface it suffers from low  
109 contrast and is not easily decoded without a proper lighting or  
110 when printed with challenging matte materials. In contrast our



**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).

goal is to embed QR codes with robust readability and minimal geometry distortion. Wei et al. [25] 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. [26] 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.

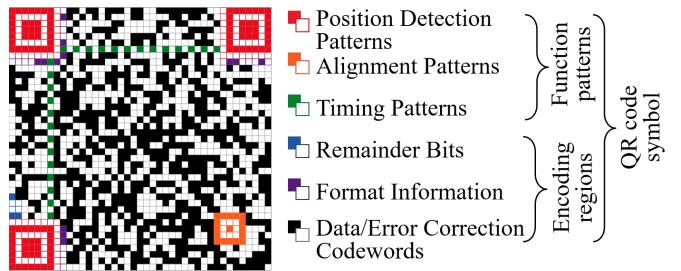
### 3. Overview

Given an artistic QR code and a target surface, we aim to carve 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. This implies the constraints that the 3D QR should be of 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. 2a), 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, connectivity optimization to guarantee the printability, and structural optimization to make sure the printed model is sustainable.

We set the black modules in QR code as the negative part to be carved out, and the white modules as 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



**Fig. 3. Structure of a QR code symbol.**

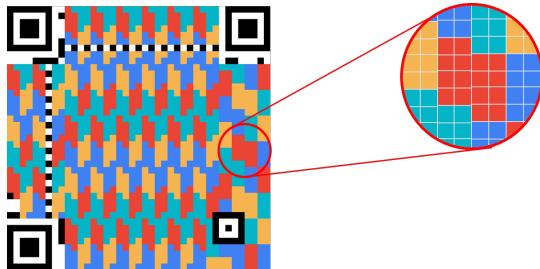
the minimum spanning tree algorithm to iteratively adding the connector until all white modules form one single component (Fig. 2b).

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. 2c). Then the whole stability is iteratively enhanced by adding connectors between the weak region and its closest neighboring part (Fig. 2d). For each iteration, the readability is evaluated and guaranteed.

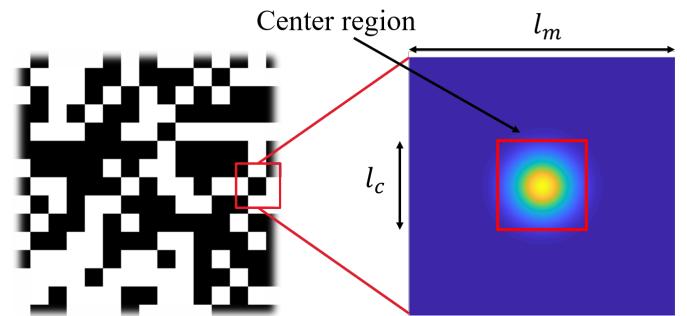
Finally, we obtain the fabricable 3D perforated QR code with structurally soundness and decoding robustness (Fig. 2e).

### 4. QR codes evaluation

In this section, we introduce a QR code evaluator to evaluate the loss of QR readability. The objective of this evaluation is to guide QR code optimization in the next section to avoid a decoding failure.



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



**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.

#### 4.1. Preliminaries on QR codes

A standard QR code is a  $n \times n$  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 it contains: 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.

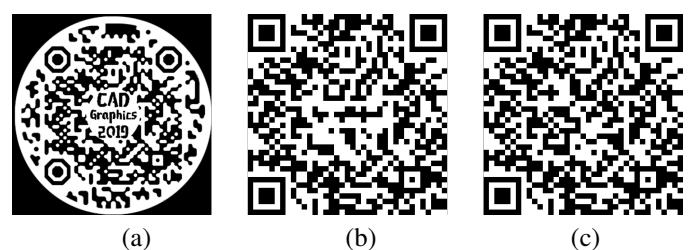
The structure of a QR code symbol (Fig. 3) takes the form of two parts, the *function patterns* and *encoding regions*.

When scanned by QR reading devices, The function patterns play a pivotal role in the positioning and recognition of QR codes. They include: (a)*position detection patterns* used to recognize QR code and identify QR version; (b) *timing patterns*, enabling module coordinates in the symbol to be determined; and (c) *alignment patterns* which are used to correct QR code distortion.

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

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 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 an error causing a substitution error in the data 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.

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



**Fig. 6.** Given an artistic QR code (a), we can extract the QR code (b) from input image and then recover its original QR code (c) via error correction algorithm.

padded into the encoding region do not carry information, just to fill data codeword capacity of 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 [27] 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 module matrix in the QR code region. To safely extract module matrix, Lin et al. [3] suggest to draw a center region for each module where contains the correct information and width of center region  $l_c$  shall be at least 1/3 of module width, see Fig. 5. A Gaussian distribution of sampling points proposed in [7] is adopted to address this requirement and then we extract the QR matrix by sampling on a grid, see Fig. 6(b).

Let  $Q$  be the QR module matrix extracted in the input image.

We recover the original QR code  $Q_0$  from  $Q$ , which regressed as the baseline of  $Q$  via Reed Solomon algorithm [1] as shown in 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  $\Delta Q$  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  $\Delta C$  as an indicator function to indicate whether  $C$  is erroneous or not.  $\Delta C = 1$  indicates the total square deviation of this codeword  $\sum_{i=1}^8 \Delta m_i^2 \geq 0$ , 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:

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

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

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

To guarantee QR code readability, it's better to damage format regions as little as possible and at least one format region should not 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 Equ. 2 and Equ. 3.

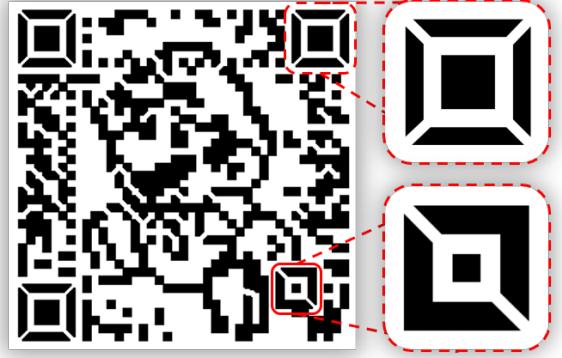
## 5. QR codes optimization

In this section, we propose a two-stage optimization to generate the connected and stable perforated QR codes automatically.

### 5.1. Connectivity optimization

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

On the basis of QR code structure, the connection optimization is divided into two steps. For function patterns, the artists provide a diagonal design to connect and reinforce position detection and alignment patterns in previous works (see Fig. 7), which proved these patterns are connected and stable enough. Thus, we just follow this design to connect and reinforce the



**Fig. 7.** In Rodion Kovenkin's work, he provides a diagonal design to connect and reinforce the position detection and alignment patterns. It is proved that the function patterns of QR code in the form of this diagonal design are connected and stable enough.

function patterns. On the other hand, we build an iterative optimization to connect the isolated white components in the encoding region.

Naturally, we can obtain a connected image by linking among isolated white components via minimal Euclidean distance like the spanning tree algorithm. Although this shortest path first method makes a minimum break to the visual appearance of artistic QR code, it might fail to decode without considering QR codes error correction capabilities, as shown in Fig. 8(b).

To avoid this situation, both Euclidean distance and the cost of QR codes error correction capabilities shall be take account into connection optimization. Given a QR image  $I$ , we first label all white components  $\{S_i\}_{i=1}^N$  in this image. Let pixel  $p$  and  $q$  be the points on the contours of  $S_i$  and  $S_j$  respectively. Then the weight  $w_{ij}$  between  $S_i$  and  $S_j$  can be formulated as:

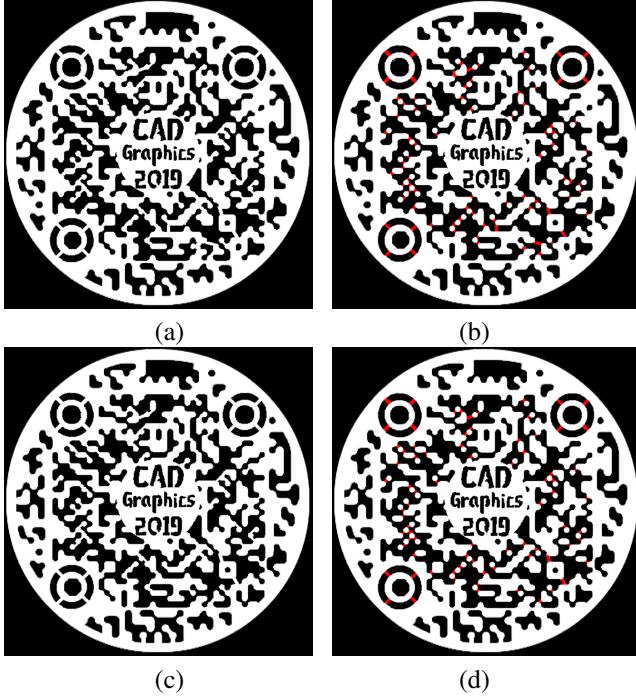
$$w_{ij} = \min_{p,q} (\lambda L(Q) + \|p - q\|^2), \quad (4)$$

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

We note that computing weights of all withe component pairs is undesired. 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 computational complexity. Then we randomly pick out a white component and connect its neighbor with the minimal weight iteratively, until all white components are connected. We show the detail iteration result in Fig. 8(c) and the comparison between shortest Euclidean distance path and our method in Fig. 8.

### 5.2. Structural optimization

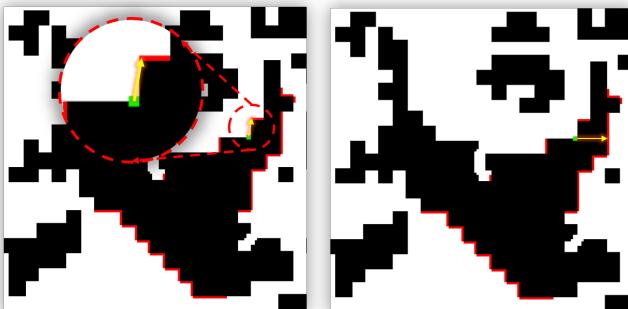
After connection optimization, we project the image on the surface and generate the perforated QR model via computing the intersections between image white regions and the surface mesh.



**Fig. 8.** (a) The connectivity enforcement by minimal Euclidean distance, i.e.,  $\lambda = 0$ , cannot be decoded. The new connectors are highlighted in red. (b), (c) The connectivity optimization result considering the QR readability ( $\lambda = 1$ ) with new connectors highlighted in red (d). The input QR code is in Fig. 6.

At first, we perform structural analysis via finite element methods to obtain the stress and displacement map of the perforated QR model under given source. To improve physical soundness, we strength the largest displacement to decrease maximum stress of the model.

Similar to connection optimization, both distance and the cost of QR readability shall be considered when choosing the neighbor part of the largest displacement region. However, it has no effect on helping us to find the optimal neighbor point if we only adopt Euclidean distance and the loss of QR readability (see Fig. 9a). For all points are connected after connection optimization, the geodesic distance is introduced to measure the shortest path along white region contour between two points. Intuitively, a well defined neighbor pairs has a shortest path



**Fig. 9.** The optimal neighbor point can not be found if we only adopt Euclidean distance and the loss of QR readability.

in Euclidean distance, but a largest path in geodesic distance. Hence, we describe the weight  $w_{pq}$  between neighbor pair  $\{p, q\}$  as

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

Then we iteratively search the largest displacement region and add connectors between the weak part and its closest neighboring part, until the whole piece is stable. The detail algorithm is shown in Alg. 1.

---

#### Algorithm 1 Structural Optimization

---

**Input:** The perforated model  $M$ , stress threshold  $\delta$ ;

**Output:** The optimized model;

```

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

```

---

We demonstrate the effectiveness of our structural optimization in Fig. 10. Both stress map and displacement map are shown.

## 6. Results and discussion

We implemented our proposed approach in C++ on an Intel Core™ i7-7700K CPU @ 4.0GHz and 16GB RAM. We use OpenCV library [28] for image processing, and CGAL [29] for mesh manipulation in the implementation. The Abaqus Unified FEA product suite offers powerful for stress analysis and complete solutions to structural optimization.

We show the input QR codes, the optimized codes and 3D printed results of the carving QR codes in Fig. 14. The models are printed by FDM-based HORI Z500 with white PLA material, whose nozzle diameter is 0.4mm. Statistics of the results including QR resolution and version, printing size, running time and number of iterations are listed in Tab. 6.

As show in Tab. 6, the algorithm both connectivity optimization and structural optimization performs well in different version of QR codes. The algorithm works well for the artistically beautiful QR code generated by the work of two predecessors. The correlation in optimization of CADGraphics is satisfactory and interesting although such QR code has a special shape . We

**Table 1. Statistics of the results.**

Model	QR Image	Resolution	QR Version	Printed QR size (cm)	Connectivity Optimization time(s) & #Iter.	Structural Optimization time(s) & #Iter.
Cuboid	Flower	700x700	8	14x14x0.2	842 & 140	387 & 7
Cuboid	Monroe	640x640	6	13x13x0.2	456 & 79	156 & 3
Cuboid	Einstein	600x600	5	12x12x0.2	2465 & 493	588 & 12
Cuboid	Mario	260x260	6	5x5x0.2	179 & 35	232 & 6
Cuboid	CADGraphics	370x370	5	9x9x0.2	336 & 61	168 & 4
Bunny				18.9x18.2x14.6		

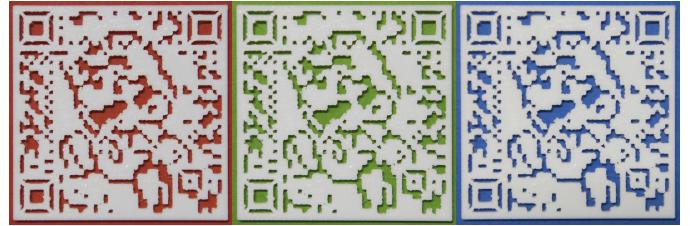
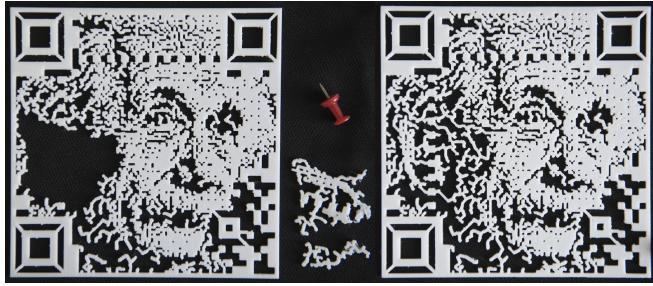


Fig. 11. The perforated QR model in different background colors can all be decoded.

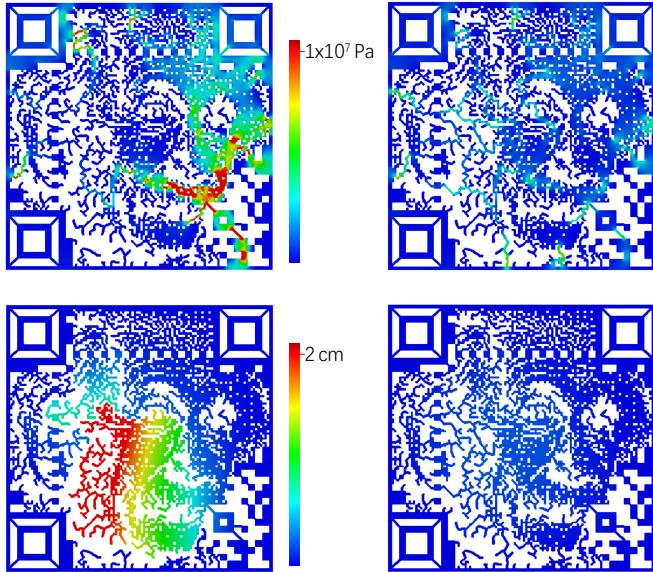


Fig. 12. The perforated QR model printed in different colors can all be decoded.

The next experiment is detecting the success rate of decoding under different background colors. We regard a successful decoding if the QR code is decoded within 2 seconds. We scan each QR code 20 times in Fig. 14 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%. Fig. 11 shows the Mario QR code in three background colors can all be decoded successfully.

The results of the rate presents the success rate is high in dark background color, particularly in blue, but gets lower in light color such as cyan and yellow. This foregone result because the more similar the background color is to the material color, the easier it is for the device to recognize QR code. The response time of the CADGraphics (Bunny) is slower on each background color, sometimes more than three seconds. The reason is the QR code on the surface need to be scanned vertically. In addition, we used different materials to print the Mario model and test the decoding under the black background. Fig. 12 illustrates that the perforated QR model printed in different colors can all be decoded successfully.

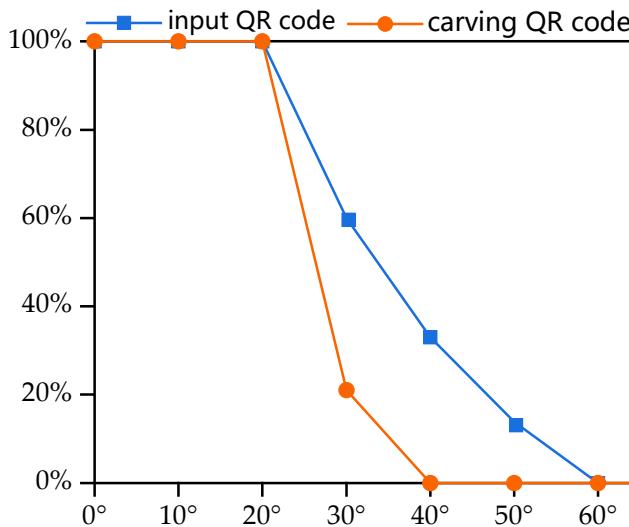
Moreover, our experiments consider aspect of the detection in different scanning angles. We remark that only when the camera is perpendicular to CADGraphics (Bunny) QR code, it

observe that the most time consuming step is the connection component to the 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. First of all, we use different mainstream mobile devices standard QR readers to tested the decoding success rate of QR codes under the black background. If the decoding is successful, we mark it with  $\checkmark$ , otherwise we mark it with  $\times$  in Tab. 2. It can be seen from the data in Tab. 2 that carving QR codes can be decoded successfully by mainstream decoders and mobile phones. Surprisingly, NeoReader can't even decode caving QR code, even the input CAGGraphics QR code.

**Table 2. Readability test on different decoders.**

	Flower	Monroe	Einstein	Mario	CADGraphics (Cuboid)	CADGraphics (Bunny)
iPhone & QuickMark	✓	✓	✓	✓	✓	✓
iPhone & NeoReader	✓	✓	✓	✓	✗	✗
Samsung& weChat	✓	✓	✓	✓	✓	✓
Mi8 & Scanner	✓	✓	✓	✓	✓	✓
Huawei & Huawei Vision	✓	✓	✓	✓	✓	✓

**Fig. 13. Robustness on the scanning angles for the Mario QR code.**

can be decoded successfully. We scan each input QR codes and carving QR codes (Cuboid) by Mi8 Scanner 20 times from a range of scanning angles and listed the average statistics in Fig. 13. The line chart above shows the origin input QR codes possess robustness within 30 degrees tilt from the orthogonal direction to QR codes. There is a clear reducing trend of decoding success rate with the increase of the tilt angle. Especially the QR codes can not be decoded when the tilt angle is greater than 60 degrees. Comparing the origin QR code, our 3D printed carving QR code allow the smaller tilt, which have good robustness within 20 degrees of tilt. It is difficult to decode successfully, when the tilt angle is greater than 30 degrees. The reason is that the height of sidewall introduces self-occlusion to QR code.

## 7. Conclusion and future work

In this paper, we introduce a method for generating readable and structural soundness QR code for carving on any surface by digital fabrication. Above all, 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 structurally soundness automatically. One is the connectivity optimization to make sure QR code is printable and decoding robustness. The other is structural optimization to generate decodable and stable 3D

carving QR code. Finally, we proved our approach the effectiveness of our 3D QR codes on a series of QR code and evaluate the robustness in various aspects.

Even though we accelerate our iteration by pre-obtaining candidate join points and parallel computing, it still takes a few minutes generate a connected readable QR code. Besides, the prerequisite for extracting the QR code at input image and recovering its prototype is input QR code has readability. In other words, we can not continue the processing to optimize if the input QR code is unable to decode. In the future, we will attempt to hide our QR code in the image to enhance its visual appearance.

## References

- [1] Wave, D. Information technology automatic identification and data capture techniques qr code bar code symbology specification. International Organization for Standardization, ISO/IEC 2015;18004.
- [2] Chu, HK, Chang, CS, Lee, RR, Mitra, NJ. Halftone qr codes. ACM Trans Graph 2013;32(6):217:1–217:8. URL: <http://doi.acm.org/10.1145/2508363.2508408>. doi:10.1145/2508363.2508408.
- [3] Lin, YS, Luo, SJ, Chen, BY. Artistic qr code embellishment. Computer Graphics Forum 2013;32(7):137–146. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12221>. doi:10.1111/cgf.12221. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12221>.
- [4] Visualead: Visual qr code generator. <http://www.visualead.com/>; ????
- [5] FLICKR, . Qr code art. <http://www.flickr.com/groups/qr-art/>; 2009.
- [6] Cox, R. Qart codes. <http://research.swtch.com/qart>; 2012.
- [7] Garateguy, GJ, Arce, GR, Lau, DL, Villarreal, OP. Qr images: Optimized image embedding in qr codes. IEEE Transactions on Image Processing 2014;23(7):2842–2853. doi:10.1109/TIP.2014.2321501.
- [8] Lin, SS, Hu, MC, Lee, CH, Lee, TY. Efficient qr code beautification with high quality visual content. IEEE Transactions on Multimedia 2015;17(9):1515–1524. doi:10.1109/TMM.2015.2437711.
- [9] Chu, CH, Yang, DN, Chen, MS. Image stabilization for 2d barcode in handheld devices. In: Proceedings of the 15th ACM International Conference on Multimedia. MM '07; New York, NY, USA: ACM. ISBN 978-1-59593-702-5; 2007, p. 697–706. URL: <http://doi.acm.org/10.1145/1291233.1291394>. doi:10.1145/1291233.1291394.
- [10] 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 Transactions on Image Processing 2012;21(1):418–425. doi:10.1109/TIP.2011.2155074.
- [11] van Gennip, Y, Athavale, P, Gilles, J, Choksi, R. A regularization approach to blind deblurring and denoising of qr barcodes. IEEE Transactions on Image Processing 2015;24(9):2864–2873. doi:10.1109/TIP.2015.2432675.
- [12] Li, X, Shi, Z, Guo, D, He, S. Reconstruct algorithm of 2d barcode for reading the qr code on cylindrical surface. In: 2013 International Conference on Anti-Counterfeiting, Security and Identification (ASID). 2013, p. 1–5. doi:10.1109/ICASID.2013.6825309.

- [13] Lay, KT, Wang, LJ, Han, PL, Lin, YS. Rectification of images of qr codes posted on cylinders by conic segmentation. In: 2015 IEEE International Conference on Signal and Image Processing Applications (IC-SIPA). 2015, p. 389–393. doi:10.1109/ICSIPIA.2015.7412222.
- [14] Zhou, S, Jiang, C, Lefebvre, S. Topology-constrained synthesis of vector patterns. ACM Trans Graph 2014;33(6):215:1–215:11. URL: <http://doi.acm.org/10.1145/2661229.2661238>. doi:10.1145/2661229.2661238.
- [15] 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. URL: <http://doi.acm.org/10.1145/2766984>. doi:10.1145/2766984.
- [16] Martínez, J, Dumas, J, Lefebvre, S, Wei, LY. Structure and appearance optimization for controllable shape design. ACM Trans Graph 2015;34(6):229:1–229:11. URL: <http://doi.acm.org/10.1145/2816795.2818101>. doi:10.1145/2816795.2818101.
- [17] Chen, C, Huang, W, Zhou, B, Liu, C, Mow, WH. Picode: A new picture-embedding 2d barcode. Trans Img Proc 2016;25(8):3444–3458. URL: <https://doi.org/10.1109/TIP.2016.2573592>. doi:10.1109/TIP.2016.2573592.
- [18] Chen, W, Ma, Y, Lefebvre, S, Xin, S, Martínez, J, wang, w. Fabricable tile decors. ACM Trans Graph 2017;36(6):175:1–175:15. URL: <http://doi.acm.org/10.1145/3130800.3130817>. doi:10.1145/3130800.3130817.
- [19] Zehnder, J, Coros, S, Thomaszewski, B. Designing structurally-sound ornamental curve networks. ACM Trans Graph 2016;35(4):99:1–99:10. URL: <http://doi.acm.org/10.1145/2897824.2925888>. doi:10.1145/2897824.2925888.
- [20] Mitra, NJ, Pauly, M. Shadow art. ACM Trans Graph 2009;28(5):156:1–156:7. URL: <http://doi.acm.org/10.1145/1618452.1618502>. doi:10.1145/1618452.1618502.
- [21] 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. URL: <http://doi.acm.org/10.1145/2907049>. doi:10.1145/2907049.
- [22] Bernd, B, Paolo, C, Luigi, M, Nico, P. State of the art on stylized fabrication. Computer Graphics Forum 2018;0(0). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13327>. doi:10.1111/cgf.13327. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13327>.
- [23] Li, D, Nair, AS, Nayar, SK, Zheng, C. Aircode: Unobtrusive physical tags for digital fabrication. In: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology. UIST ’17; 2017, p. 449–460.
- [24] Chen, F, Luo, Y, Tsoutsos, NG, Maniatakos, M, Shahin, K, Gupta, N. Embedding tracking codes in additive manufactured parts for product authentication. Advanced Engineering Materials 2018;0(0):1800495. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adem.201800495>. doi:10.1002/adem.201800495. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/adem.201800495>.
- [25] Wei, C, Sun, Z, Huang, Y, Li, L. Embedding anti-counterfeiting features in metallic components via multiple material additive manufacturing. Additive Manufacturing 2018;24:1 – 12. URL: <http://www.sciencedirect.com/science/article/pii/S2214860418305189>. doi:<https://doi.org/10.1016/j.addma.2018.09.003>.
- [26] Kikuchi, R, Yoshikawa, S, Jayaraman, PK, Zheng, J, Maekawa, T. Embedding qr codes onto b-spline surfaces for 3d printing. Computer-Aided Design 2018;102:215 – 223. URL: <http://www.sciencedirect.com/science/article/pii/S0010448518302537>. doi:<https://doi.org/10.1016/j.cad.2018.04.025>; special Issue on SPM 2018.
- [27] Owen, S, et al. zxing-multi-format 1d/2d barcode image processing library with clients for android, java. Github 2014;.
- [28] Open source computer vision library. <https://github.com/opencv>; 2015.
- [29] The CGAL Project, . CGAL User and Reference Manual. 4.12 ed.; CGAL Editorial Board; 2018. URL: <https://doc.cgal.org/4.12/Manual/packages.html>.



**Fig. 14.** (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 [2]. The Mario code is generated by [6]. The CADGraphics code is generated by an on-line generator "<https://www.unitag.io/qrcode>".