DOI:10.1111/cgf.14065
Eurographics Symposium on Geometry Processing 2020
Q. Huang and A. Jacobson
(Guest Editors)

Volume 39 (2020), Number 5

Fabricable Unobtrusive 3D-QR-Codes with Directional Light

Hao Peng[1], Peiqing Liu[1], Lin Lu†[1], Andrei Sharf[2], Lin Liu[1], Dani Lischinski[3] and Baoquan Chen[4]

[1]School of Computer Science and Technology, Shandong University, Qingdao, China
penghao@mail.sdu.edu.cn, lpq17@mail.sdu.edu.cn, llu@sdu.edu.cn, lorrain0407@gmail.com
[2] Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel
asharf@gmail.com
[3] School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel
danix@mail.huji.ac.il
[4] School of Electronics Engineering and Computer Science, Peking University, Beijing, China
baoquan@pku.edu.cn

Abstract

QR code is a 2D matrix barcode widely used for product tracking, identification, document management and general marketing. Recently, there have been various attempts to utilize QR codes in 3D manufacturing by carving QR codes on the surface of the printed 3D shape. Nevertheless, significant shape editing and modulation may be required to allow readability of the embedded 3D-QR-codes with good decoding accuracy. In this paper, we introduce a novel QR code 3D fabrication framework aimed at unobtrusive embedding of 3D-QR-codes in the shape hence introducing minimal shape modulation. Essentially, our method computes bi-directional carvings in the 3D shape surface to obtain the black-and-white QR pattern. By using a directional light source, the black-and-white QR pattern emerges as lighted and shadow casted blocks on the shape respectively. To account for minimal modulation and elusiveness, we optimize the QR code carving w.r.t. shape geometry, visual disparity and light source position. Our technique employs a simulation of lighting phenomena through carved modules on the shape to ensure adequate contrast of the printed 3D-QR-code.

CCS Concepts

• ***Computing methodologies*** → ***Volumetric models;*** *Parametric curve and surface models;*

1. Introduction

Since ancient times, humans explored lights and shadows with great interest and curiosity. Archeological monuments like pyramids and obelisks often bear alignment with astronomical light phenomena. “On summer solstice on the Stonehenge site, an observer standing within the stone circle and looking north-east, would see the sun rise aligned with the Sun-stone” [HW65]. Such combinations of light and structure continue to fascinate us even today. Our work drives inspiration from these phenomena and aims to control lights and shadow casting using 3D printed QR code structures.

As a pattern of high storage capacity and usability, QR codes were introduced used since the age of internet as a 2D matrix barcode printed on documents and products for tracking, identification, and management. Recently, QR codes have also been applied to 3D manufacturing [KYJ*18, PLL*19]. In these works, QRcodes are embedded onto 3D shapes by carving the black-and-white modules while utilizing lights and shadows to generate QR patterns which can be decoded by QR readers. Nevertheless, 3D-QR-codes must provide enough contrast to be recognized correctly and therefore require significant carving depth which in turn affect aesthetics and limit applicability to thin-shell structures.

Our work focuses on this problem and aims to compute unobtrusive 3D-QR-code structures that are minimal in terms of carving depth and shape modulation yet sufficient w.r.t. casted shadows and pattern contrast. Specifically, our 3D-QR-code optimization method targets to introduce minimal editing to the original 3D shape, thus preserving the visual appearance and geometry.

Due to 3D manufacturing constraints, additional 3D printability characteristics such as minimal feature size, structure strength, and stability must be considered. For example, in thin shell shapes and hollow objects, our carving module must account for shape stability and strength without introducing holes and cracks on the surface. Our 3D-QR-codes are minimal in terms of their appearance and typically shallow due to their optimized carving module. Therefore, they may be easily adapted to 3D printability requirements.

To compensate for the unobtrusiveness of our QR code structures, we use a directional light for QR code decoding. Compared

† Corresponding author.

to [KYJ*18, PLL*19], using directional light rather than ambient occlusions reduces visual effects and allows embedding 3D-QR-codes on 3D shapes with minimal modulations (Figure 1). Thus, our 3D QR-codes are usually of small size and depth and placed in unobtrusive parts of the object with minimal interference to visual appearance. Nevertheless, the direct light source allows a proper contrast between the white modules that are facing the light and the black modules that emerge when light projects from a predefined direction and casts shadows to generate the appropriate black-and-white pattern.

In summary, our paper makes the following contributions:

- An unobtrusive 3D-QR-code embedding method that accounts for minimal interference with the object's appearance and shape. Our technique considers both surface geometry, visibility, and light parameters in its optimization computation.
- A minimal depth 3D-QR-code carving module. Our QR code structures are applicable in challenging cases such as thin shells and hollow structures.
- Ambient light vs. direct light QR structure obtrusiveness. Our method embeds the 3D-QR-code such that it is unobtrusive in the ambient environment while highly discernable with high contrast in additional direct light for decoding purposes.

2. Related Work

Quick Response codes are popular machine-readable information means due to their fast readability and high storage capacity. Our work adapts this technique to digital fabrication. In the following, we discuss relevant works and recent developments in QR codes and 3D printing.

QR codes enhancements. Researchers have put much attention on enhancing QR-codes with various features. In this context, visually attractive QR codes have been explored [Cox, Vis, CCLM13, GALV14, LHLL15], aiming to incorporate visual features with ordinary QR codes without compromising their readability. Chu et al. [CCLM13] firstly combined halftone images and QR codes. They generated a binary embedding by subdividing each QR module into 3×3 submodules to bind the halftone image and optimized the binary pattern of each module to achieve an adequate trade-off between decoding robustness and image quality. Garateguy et al. [GALV14] further embedded QR codes into color images and optimized the concentration of pixels and its corresponding luminance to minimize a visual distortion metric.

QR codes have been utilized in watermarking techniques for years. They were either embedded in arbitrary images [HWW12] or vice versa [VR12]. For the latter case, data was embedded in the QR code, which acted as a container to hide information [HCF11, BMT13].

Digital 3D watermarking. Along with the prosperity of 3D capturing and creative design, the intellectual property protection of 3D models has attracted more and more attention. For this purpose, researchers have proposed 3D watermarking [OMA98, WLDB08, BL13, YPRI17]. Their general idea was to insert the watermark signal in a suitable feature space of the digital model.

Depending on the selection of the insertion space, 3D watermarking techniques can be broadly classified into two categories, spatial domain-based and transform domain-based. Spatial domain techniques suggested geometrical embedding of the watermark [OMA98], while transform-based techniques suggested first transforming the watermark into a frequency domain and embedding it there [OMT02].

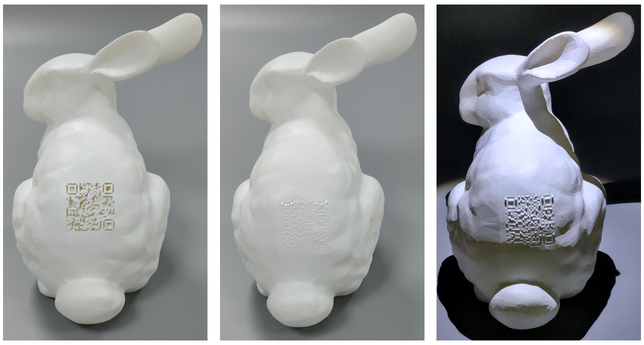


Figure 1: *3D-QR-code carvings on bunny's back. In (a) code is carved by [PLL*19] under ambient light. Our method generates unobtrusive codes under ambient light in (b), yet readable under directional light in (c).*

Similar to us, digital 3D watermarking techniques are concerned with unobtrusive embedding, where the watermark does not interfere with the visual appearance. Furthermore, the embedded watermark should be robust, i.e., making the watermark indestructible and irremovable.

3D printing, carving and light control. With the advent of commodity 3D printing, a common goal involves improving the quality and speed of shape manufacturing. The printing direction was investigated in the context of stress analysis [US13] and support optimization [ZLP*15]. Structural analysis was applied to detect stability and load problems [ZPZ13, LSZ*14]. Similarly, printing time was improved through slicing optimization in [WCT*15].

With the progress of 3D manufacturing tools, we observe novel methods focusing on editing 3D structures for controlling light projections through them and creating various optical phenomena. In this path, Mitra and Pauly [MP09] optimized 3D shapes to cast varying shadow images when lit from different directions. They introduced a geometric optimization that computes a 3D volume whose casted shadows approximate the input images. Alexa and Matusik [AM10] constructed relief surfaces whose diffuse reflection approximates given images under known directional illumination, without considering self-shadowing or other shading effects. They further proposed a method for generating images by self-occlusion of small tubes with varying length [AM11] .

In [BBAM12], surfaces were generated to both cast and receive shadows forming different prescribed images depending on the light direction. Schüller et al. [SPSH14] introduced appearance-mimicking surfaces that preserved the look and details of a 3D object on a surface when observed from designated viewpoints. Zhao et al. [ZLW*16] projected continuous grayscale images onto the surrounding walls as shadows via optimizing a set of tiny holes

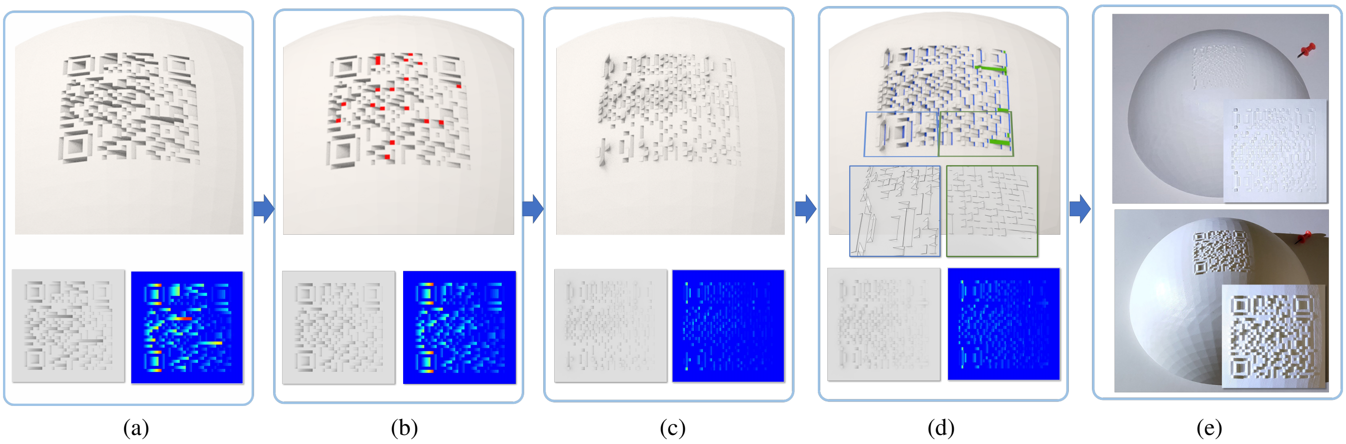


Figure 2: *Pipeline overview. Initially, a 2D QR-code image is projected onto the surface considering a predefined directional light (a). Next, black modules' continuity (red) is suppressed (b) and carving depth is optimized iteratively (c). In post processing (d), brightness compensation (green) and reflection elimination (blue) are computed to enhance decoding readability. The 3D printed QR-code (e) is readable only under a preset lighting direction. Bottom row (a-d) shows 3D QR-codes under ambient light (unobtrusive) and color-coded depth map.*

over the shell. Common to these works is their *play-of-light*, i.e. designing 3D structures and investigating patterns made by light as it casts over the surface. Our work follows in this path and generates the QR code using shadows casted by light. Nevertheless, QR-codes need high contrast for their readability and therefore geometry optimization w.r.t. light and shadow should yield highly accurate patterns.

In recent years, research efforts have been devoted towards embedding optical barcodes onto 3D printed objects. Wei et al. [WSHL18] embedded QR codes with a specific tagging material into inaccessible positions of metallic components using multiple material SLM technologies. Their QR code was recognized via X-ray imaging. Li et al. [LNNZ17] introduced a technique that allows tagging physically fabricated objects with given information. They used subsurface scattering to embed QR code-like patterns, such that the surface appearance is preserved. However, the models need high-end printing techniques and may have low contrast. Maia et al. [MLYZ19] extended this work and proposed an encoding algorithm that enables 3D printing layers to carry information such that it can be adapted to various types of 3D printing techniques. The above methods require specified decoders, which precludes the applications to standard QR codes. In contrast, our goal is to embed regular 3D QR-codes that can be decoded with standard readers.

Most related to our proposed method are [KYJ*18] and [PLL*19]. They showed a straightforward approach to 3D printed QR-codes generating readable black-and-white QR patterns from light casting. In contrast, our work takes a step further and optimizes QR-code structures for both unobtrusive shape embedding and readability. Thus, our technique yields optimally carved QR-code modules of minimal size and depth, that can be easily utilized in the 3D manufacturing pipeline.

3. Overview

To successfully decode a 3D-QR-code on a surface, the code should have sufficient resolution, high contrast and proper alignment. Given a 2D QR code and a 3D shape, we embed the QR code on the surface by computing its minimal carving with respect to surface modulation. Furthermore, our 3D-QR-codes should be unobtrusive under normal ambient light while easily readable and decoded under a predefined directional light setup. Thus, our 3D QR-code embedding optimization has the following objectives:

- **unobtrusiveness**: 3D-QR-code should be minimally visible (unobtrusive) under normal ambient light configuration.
- **printability**: 3D-QR-code carved structures should account for printability constraints such as minimal resolution and stability.
- **readability**: 3D-QR-code should be easily decoded under the correct directional light setup.

The objectives above conflict in a sense. While minimal carving of 3D-QR-codes yields unobtrusiveness, it also reduces readability robustness and printability resolution. Similarly, readable 3D QR-code structures may yield non-stable geometry (e.g., unsupported/hanging) which may be infeasible in terms of printability. To this end, we formulate the 3D-QR-code unobtrusive embedding as an optimization problem which balances between unobtrusiveness, readability and printability terms.

Given a 2D QR code image, a 3D shape and a predefined directional light, we project the 2D QR code image onto the 3D surface defining a carving region. We assume that the projection direction is in surface normal direction as QR codes are typically scanned from top-view. Furthermore, the carving region is assumed to be continuous, with no self occlusions.

Next, we optimize the QR code data and its 3D geometry w.r.t. the directional light. Our 3D-QR-code optimization calculation is based on a 3D shape and light interaction simulation. Since it is infeasible to accurately compute all lighting phenomena in the scene,

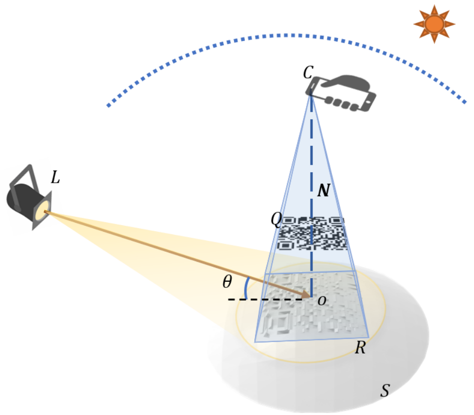


Figure 3: *Illustration of our 3D QR-code setup.*

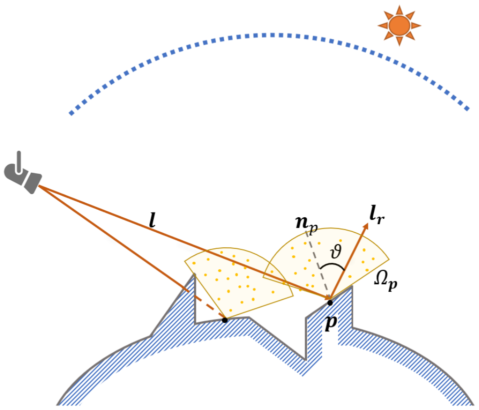


Figure 4: *1D illustration of directional occlusion computation.*

we use the Phong light model for simulation approximation and calibration of the 3D-QR-code.

A standard QR code [JTC15] is constructed of squared black-and-white modules, set in a regular square array and consists of function patterns and an encoding region. We do not try to change the function patterns for their indispensable role in the positioning and recognition of QR codes and thus focus on the encoding region consisting of data and error correction codewords. In our method, black modules are obtained by carving the surface geometry, yielding self occlusions as the surface casts shadows over the carved region w.r.t. the directional light. Similarly, white modules are obtained by casting direct light on the visible surface parts.

We propose a two-step method to compute the optimal 3D QR-code embedding i.e., a black module suppression and a carving optimization steps. Intuitively, large black modules require large occluders and consequently, deep carving. This conflicts with our minimal carving objective. Thus, we re-distribute black modules to optimize their size, suppressing widely connected black regions. This enables shallow carving depth and minimal modulation (Figure 2(b)). In the carving optimization step, we partition the overall carving region into a group of independent subregions. A subregion defines a continuous black module and a corresponding white module that serves as its occluder and casts shadows on the carved region. In order to simplify optimization, each subregion is optimized separately by changing the carving rate between black and white modules. The carving optimization iterates and ends when all subregions are minimized and converged (Figure 2(c)).

In a post-processing step, our method re-evaluates and further refines the QR code embedding. Specifically, we analyze high curvature regions that reduce the white module brightness and inner reflections that interfere with the black modules. We compensate low brightness in curved regions and eliminate inner reflections in the surface (see Figure 2(d)).

4. 3D QR Setup

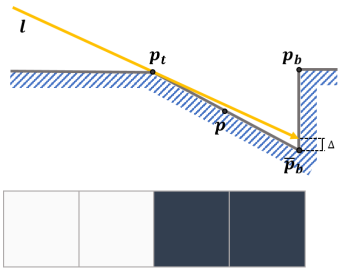
In this section, we introduce the construction of the QR-code black-and-white 3D modules on the surface.

4.1. 3D QR-code initial carving

Given a 2D QR image Q and a 3D surface S with a user-specified origin position $\mathbf{o}$, we project the QR image onto a region R on the surface centered at $\mathbf{o}$. The projection direction $\mathbf{N}$ is the averaged surface normals in R, assuming QR codes are typically scanned perpendicularly from top-view (see Figure 3).

QR carving proceeds by considering a point light source L, with direction $\mathbf{l}$ and angle θ, between the tangent plane of $\mathbf{o}$ and $\mathbf{l}$. In our setup, L is a spotlight far enough, thus regarded as point light with a range covering the whole QR region. We set L such that θ is the minimal angle that satisfies that all white modules are lighted.

The projected QR image partitions the surface vertices inside R into black and white modules. A module on the 3D surface refers to the rectangular area of $n \times n$ vertices that all lie inside a black or white projected QR code. To allow sufficient resolution, we remesh R into a dense triangulation. We then carve the black modules such that they become occluded by the white modules under the directional light.



As shown in the left figure, $\widehat{\mathbf{p}_t\mathbf{p}_b}$ denotes the region of black modules. The initial carving for black modules, i.e., $\widehat{\mathbf{p}_t\bar{\mathbf{p}}_b}$ is determined by the intersection between $\mathbf{p}_t$ and $\mathbf{l}$, which is Δ deeper along the projection direction. The parameter Δ is set as a small value $1e-6$. A vertex $\mathbf{p}$ is occluded by a white module and belongs to a black module if it is below the light rays in the direction $\mathbf{l}$. Carving depth $D(\mathbf{p})$ is defined as the unsigned distance from the original position.

4.2. Light model

We assume that the 3D surface is Lambertian and lighting consists of ambient and directional components. Thus, for each 3D vertex $\mathbf{p}$

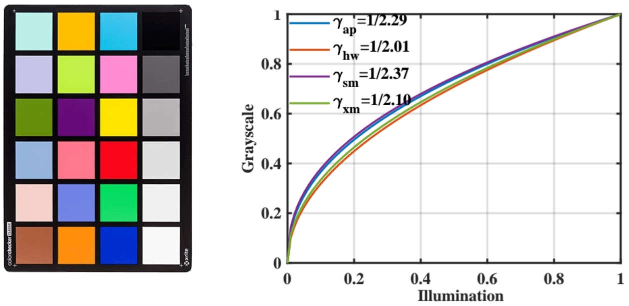


Figure 5: *Gamma correction curves for gray values and their according illumination value. We photograph the color checker(X-Rite) using mainstream mobile phones such as iPhone(ap), Huawei(hw), Samsung(sm) and Xiaomi(xm) and then test their gamma values using iQstest. The gamma correction curves for the gray values and their according illumination value are plotted in the right figure.*

on S with normal $\mathbf{n}$, we compute its illumination following ambient and directional models [Lan02, RGS09].

Ambient illumination can be computed by integrating the visibility function over the hemisphere Ω w.r.t. projected angle:

$$O_a(\mathbf{p}) = \frac{1}{\pi}\int_{\Omega} V_{p,\omega}(\mathbf{n}\cdot\omega)\,\mathrm{d}\,\omega = \frac{1}{\pi}\int_{\Omega_p}(\mathbf{n}\cdot\omega)\,\mathrm{d}\,\omega, \qquad (1)$$

where $V_{p,\omega}$ is the visibility function at $\mathbf{p}$, defined 0 if $\mathbf{p}$ is occluded and 1 otherwise (in direction ω). $\mathrm{d}\,\omega$ is the infinitesimal step of ω. For each vertex $\mathbf{p}$, we can restrict the integral domain on the hemisphere to its visible area, i.e., the integral is computed in $\Omega_p \subset \Omega$, where $\Omega_p = \{\omega | V_{p,\omega} = 1, \omega \in \Omega\}$.

To compute the directional illumination component, we denote $\mathbf{l}_r$ as the reflection vector at $\mathbf{p}$ from $\mathbf{L}$ where ϑ is the intersection angle between the reflected light and normal direction $\mathbf{n}_p$ at $\mathbf{p}$. We define the directional occlusion O_d at $\mathbf{p}$ as:

$$O_d(\mathbf{p}) = V_{p,\mathbf{l}_r}(\mathbf{n}_p \cdot \mathbf{l}_r), \qquad (2)$$

The indicator $V_{p,\mathbf{l}_r}$ denotes if the point $\mathbf{p}$ is visible (1) or occluded (0).

We denote the intensity of ambient and directional illumination as I_a and I_d, respectively. The measured illumination at $\mathbf{p}$ is computed as follows:

$$I(\mathbf{p}) = I_a\rho\pi O_a(\mathbf{p}) + I_d\rho O_d(\mathbf{p}) \qquad (3)$$

where ρ is the albedo at the point $\mathbf{p}$ (See Figure 4). We fabricate 3D QR-codes from white resin by an SLA 3D printer with material albedo value $\rho = 0.36$. Measured intensity values under ambient and directional light sources are $I_a = 32.6\,$lux and $I_d = 1227.6\,$lux in our cases.

Tone mapping. In digital photography, tone mapping [Hun04] is frequently used for converting high dynamic range into a more limited range. In our work, we apply tone mapping to map illumination of 3D QR-codes to simulated grayscale images.

A widely useful global tone mapping method in cameras is gamma compression, which has the filter:

$$G(\mathbf{p}) = I(\mathbf{p})^{\gamma}, \qquad (4)$$

where $0 < \gamma < 1$. This function maps the normalized luminance of the original pixel in the domain $[0,1]$ to an output grayscale range $[0,1]$. To compute a reasonable gamma, we capture the color checker with standard mobile phone cameras and test their resulting gamma values (see Figure 5). We observe that gamma values for all mobile phone cameras are similar, thus in all our experiments is set to $\gamma = 1/2.2$.

Finally, our light model and tone mapping allow us to compute the resulting QR image and simulate, validate and optimize our 3D-QR-code embedding. We define three images in the context of light and model simulation:

- Reference image $\mathbf{U}$: QR region projection on the original shape
- Ambient QR image $\mathbf{Q}_a$: 3D-QR-code under ambient light ($I_d = 0$)
- Lighted QR image $\mathbf{Q}_{ad}$: 3D-QR-code under both ambient and directional light

For validation purposes, we embed a 3D-QR-code on a sphere model and compare simulated images and corresponding photographed images (see Figure 6). The difference between the simulated images $\mathbf{U}$, $\mathbf{Q}_a$, $\mathbf{Q}_{ad}$ and corresponding photographs are 0.9%, 1.8% and 4.4%, demonstrating sufficient accuracy of our simulation.

5. 3D QR Optimization

5.1. Objective function

Our method essentially carves black modules to occlude black pixels while exposing white modules to directed light yielding white pixels. Thus, our optimization adjusts QR modules and vertices positions in the QR region independently, preserving the overall black-white module correctness. We formulate our objectives, i.e., unobtrusiveness, printability, and readability quantitatively. Given the QR image $\mathbf{Q}$, i.e. a binary image matrix $\{0,1\}$ to denote black and white modules and vertices positions $\mathbf{p}$ in the QR region, the objective function is defined as:

$$\begin{aligned} \min_{\mathbf{Q},\,\mathbf{p}} \quad & \alpha\|\mathbf{Q}_a - \mathbf{U}\|^2 + \beta\|\bar{D}(\mathbf{Q},\mathbf{p})\|^2, \\ s.t. \quad & \mathbf{Q}_{ad} \;\; \textit{is decodable}. \end{aligned} \qquad (5)$$

The first two goals are measured by the visual disparity between the *reference image* $\mathbf{U}$, *ambient QR image* $\mathbf{Q}_a$ and average carving depth. The average carving depth $\bar{D}$ of vertices in the QR region is defined as,

$$\bar{D} = \frac{1}{n}\sum_{i=1}^{n} D(\mathbf{p}_i), \quad \forall \mathbf{p}_i \in \Gamma, \qquad (6)$$

where Γ denotes the carved region with n carving vertices.

The readability goal is defined as three sub-constraints according to QR-code standards [JTC15]:

1. Symbol contrast constraint. Given a simulated image $\mathbf{Q}_{ad}$, the decoder first calculates means of top 10% lightest gray scale

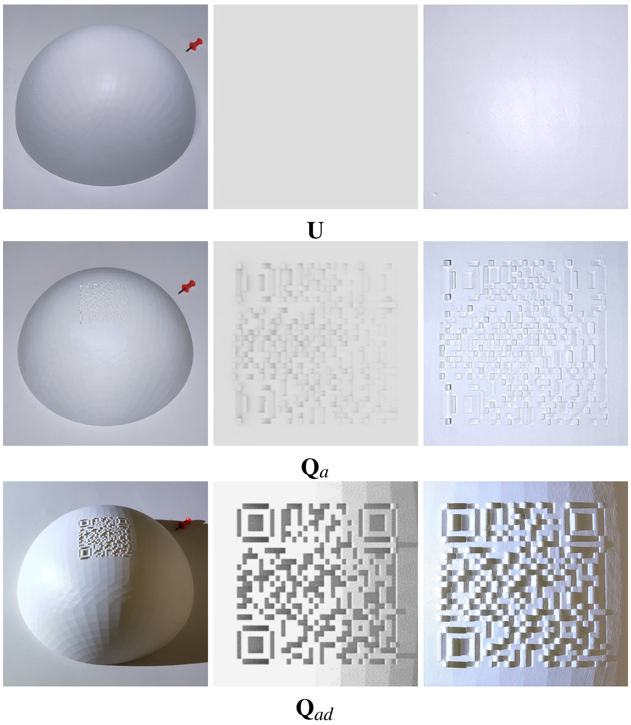


Figure 6: *3D-QR-code simulation. Left-to-right columns are the 3D model, simulated image and real photograph. Rows top-bottom depict un-carved model, carved model under ambient light and carved model under both ambient and directional light.*

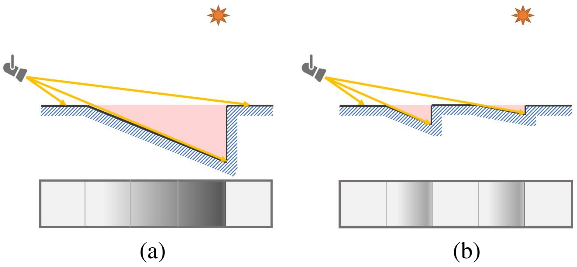


Figure 7: *Continuity suppression 1D illustration. By suppressing the black modules continuity (a), the carving depth is reduced accordingly and the QR code is less obtrusive under ambient light (b). Bottom row are corresponding ambient image pixels* $\mathbf{Q}_a$.

values a and 10% darkest gray scale values b. Symbol contrast is defined as $\varphi = a - b$, $\varphi \in [0,1]$, and $\varphi \geq 0.2$ for robust decoding.

2. Binarization constraint. The grayscale image $\mathbf{Q}_{ad}$ is binarized during decoding. The binarization threshold is $\psi = (a+b)/2$ (i.e, white regions in $\mathbf{Q}_{ad}$ are greater than ψ and black regions are less than ψ).
3. Error correction constraint. Reed-Solomon algorithm is used for error correction of $\mathbf{Q}_{ad}$ (after binarization). Thus, QR-codes are correct iff error correction function $RS(\mathbf{Q}_{ad})$ returns true.

Next, we describe our two stage optimization for optimizing the objective (Eqn. 5). It consists of a module optimization to suppress continuous black modules (Figure 7) and a carving optimization which modulates carved vertices to reduce their visual disparity Figure 8.

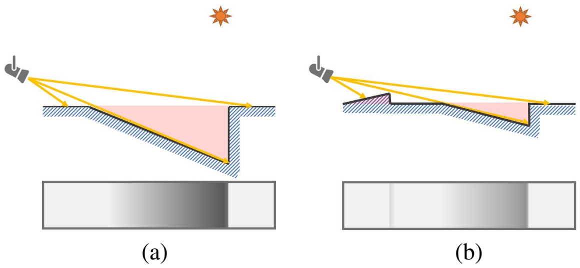


Figure 8: *Carving modulation 1D illustration. By modulating the carving region (a), the carving depth is reduced accordingly and the QR code is less obtrusive under ambient light (b). Bottom row are corresponding ambient image pixels* $\mathbf{Q}_a$.

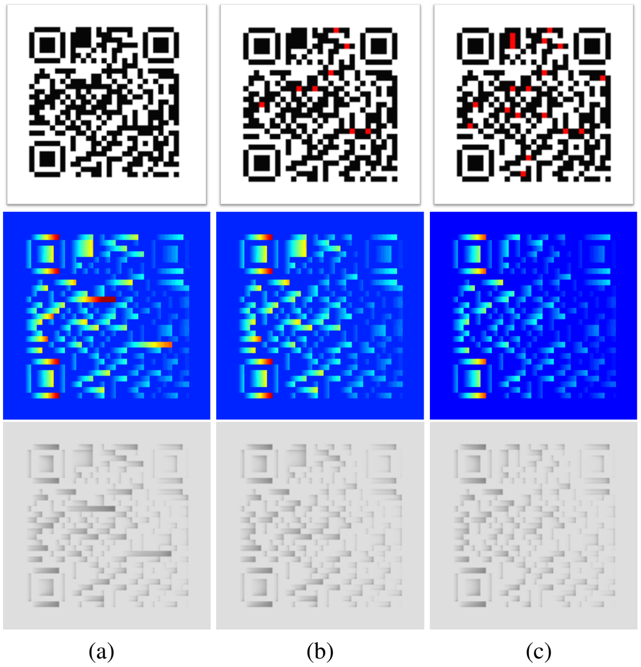


Figure 9: *Module optimization. Columns (a-c) are initial, mid, and final black module flipping, respectively. Rows, top to bottom, are the QR image (flipped modules in red), the corresponding depth map, and the ambient QR image* $\mathbf{Q}_a$. *Flipped QRs are still robust to decoding while consisting of fewer carving modulations.*

5.2. Module optimization

We suppress continuous regions of black modules through flipping in order to achieve minimal visual difference. Our goal here is to carve black modules as narrow bars along the light direction instead of large connected components (Figure 7).

Since flipping QR modules reduces readability of the QR-code, we apply a loss function $L(\mathbf{Q}_{ad})$ introduced in [PLL*19] to replace

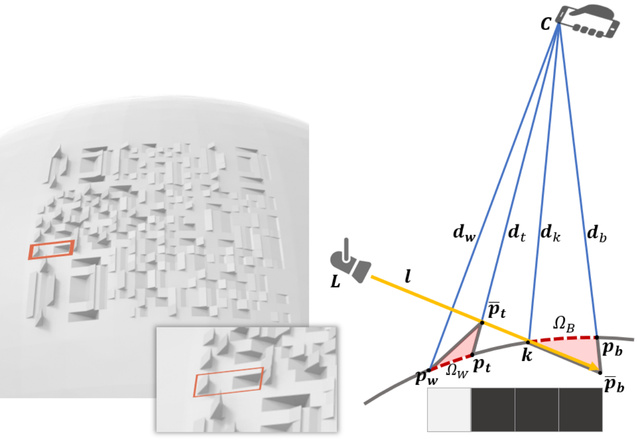


Figure 10: *A carving subregion (a) (red box) and its 1D sideview (b). Positive (outward) and negative (inward) carving are highlighted in pink.*

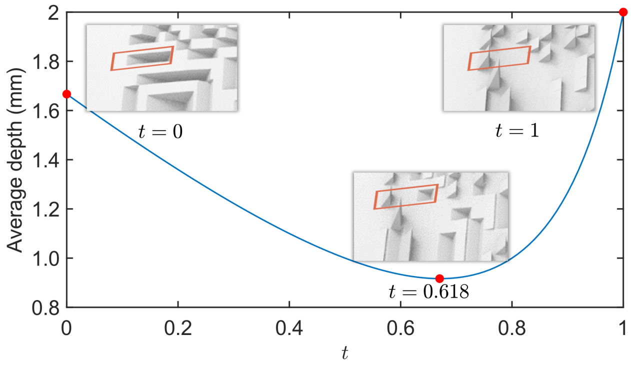


Figure 11: *The average carving depth for a carving subregion with $t \in [0,1]$ is a convex curve with minimum at $t = 0.681$.*

the error correction constraint. Our module optimization objective function is formulated as:

$$\min_{\mathbf{Q}} \quad \alpha \|\mathbf{Q}_a - \mathbf{U}\|^2 + \beta \|\bar{D}(\mathbf{Q})\|^2 + \|L(\mathbf{Q}_{ad})\|^2 . \qquad (7)$$

This objective can be expressed as a $0-1$ integer programming problem with nonlinear terms for visual disparity and QR readability. We solve the optimization iteratively, i.e., in each iteration, we search for all valid flippings, evaluate the change in objective function (Eqn. 7) and flip the module that causes the largest descend. We iterate until there are no feasible flipping of black modules (see Figure 9).

5.3. Carving optimization

We define a *carving subregion* as a region consisting of a white module together with adjacent black modules along the light direction. We use this definition to partition the QR region into *carving subregions* $\{\Gamma_i\}_{i=1}^m$. In Figure 10, $\widehat{\mathbf{p}_w\mathbf{p}_t}$ and $\widehat{\mathbf{p}_t\mathbf{p}_b}$ denote the regions for white module and black modules in a carving subregion, respectively. The outward and inward carving regions are denoted by Ω_W and Ω_B, respectively.

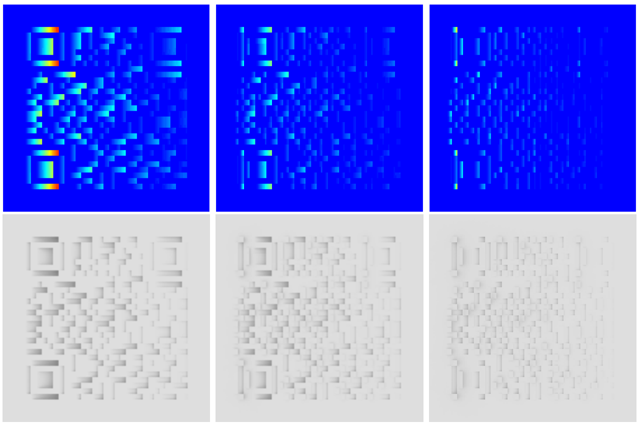


Figure 12: *An example of carving optimization. Columns (a-c) contain the carving optimization at initial, midway, and final iterations. Rows, top-bottom, are depth map and ambient QR image $\mathbf{Q}_a$.*

For each carving subregion, we introduce a reference point $\mathbf{k} \in \widehat{\mathbf{p}_t\mathbf{p}_b}$ to measure the relative inward and outward carving. Once $\mathbf{k}$ is settled, Ω_W and Ω_B can be determined accordingly. The position of outward carving $\bar{\mathbf{p}}_t$ is the intersection between $\mathbf{l}$ (crossing $\mathbf{k}$) and projection ray $\mathbf{d}_t$. The position of inward carving $\bar{\mathbf{p}}_b$ is the intersection between $\mathbf{l}$ and projection ray $\mathbf{d}_b$. For simplicity, we omit here the parameter Δ which is needed for computing the inward carving position $\bar{\mathbf{p}}_b$.

We represent $\mathbf{k}$ as the intersection between the projection vector $\mathbf{d}_k$ and the given surface, and calculate $\mathbf{d}_k$ from the linear combination of projection vectors $\mathbf{d}_t$ and $\mathbf{d}_b$. Therefore, we can use a scalar variable, called *carving rate* $t \in [0,1]$ for calculating $\mathbf{k}$ in each subregion, i.e.:

$$\mathbf{d}_k(t) = \frac{(1-t)\mathbf{d}_t + t\mathbf{d}_b}{|(1-t)\mathbf{d}_t + t\mathbf{d}_b|}, \; t \in [0,1]. \qquad (8)$$

Figure 11 demonstrates the relationship of carving rate t and average carving depth in a carving subregion.

Thus, for each point $\mathbf{p} \in \Gamma$, its carving depth can be calculated once t in the related carving subregion is set. We reformulate the objective of carving optimization as:

$$\min_{\mathbf{t}} \quad \alpha \sum_{i=1}^{m} \|\mathbf{Q}_a^{\Gamma_i}(t_i) - \mathbf{U}^{\Gamma_i}\|^2 + \beta \sum_{i=1}^{m} \|\bar{D}_{\Gamma_i}(t_i)\|^2 . \qquad (9)$$

We tune the rate of outward and inward carving for each subregion. Note that during the carving optimization the occlusions for black modules are unchanged. Thus readability constraints are not affected and are not considered in the objective function.

Due to the complexity of the relationship between t and the ambient image and average carving depth for each subregion, the derivatives of $\{t_i\}_{i=1}^m$ cannot be calculated analytically. Thus, we

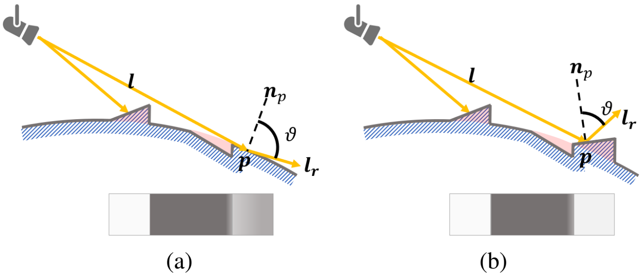


Figure 13: *Brightness compensation 1D illustration. The white module could be mistaken for the black (a) due to occlusion under the directional light. By modifying the inclination of the white module we compensate the lightness to make it the right color (b).*

employ the automatic differentiation method for computing the derivatives. We use the Ceres solver [AMO]'s conjugate gradients implementation to minimize the objective function. See Figure 12 for a carving optimization example.

5.4. Post processing

While our optimization accounts for QR readability, in some cases decoding failures may occur. In the following we introduce post-processing to detect and fix these problems.

Brightness compensation. High curvature surfaces (Figure 13(a)) may yield white module vertices $\mathbf{p}$ that are mistakenly read as black since they do not provide enough directional occlusion up to threshold ψ. Thus, we find ϑ for $\mathbf{p}$ whose grayscale is below the threshold to make sure $G(\mathbf{p}) \geq \psi$ as:

$$\underset{\vartheta}{\arg\min} \quad \|G(\mathbf{p}(\vartheta)) - \psi\|^2, \tag{10}$$

where ϑ is the intersection angle between the reflection light $\mathbf{l}_r$ and normal direction $\mathbf{n}_p$.

We further identify the position of $\mathbf{p}$ according to ϑ since the light direction $\mathbf{l}$ is fixed (Figure 13(b)). Suppose the ambient occlusion is constant for $\mathbf{p}$, we easily derive the gradient for $G(\mathbf{p}(\vartheta))$ using the chain rule. We employ the conjugate gradient method to solve ϑ. The result is shown in Figure 2(d).

Reflections elimination. Inner reflections inside a carved modules may reduce shadows and lower contrast (Figure 14(a)). To reduce this artifact, we further deform the carved module such that the light is always reflected outside the carving region (Figure 14(b)). In our experiment, we set the deformation angle $\tau = 90^\circ$ to make sure the light could be reflected outside.

5.5. Optimization computation

Optimization parameters. There are two terms in the objective function, visual disparity, and average carving depth. The two terms both affect the "unobtrusiveness" of the 3D QR codes, while focusing on different aspects.

The visual disparity term prefers outward carving rather than inwards, as outward carving is less noticeable. However, large outward carving would cause the model fragile and also affects the printability.

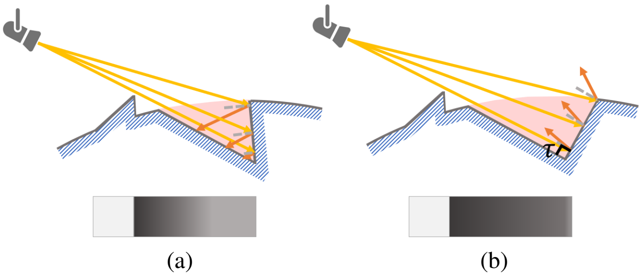
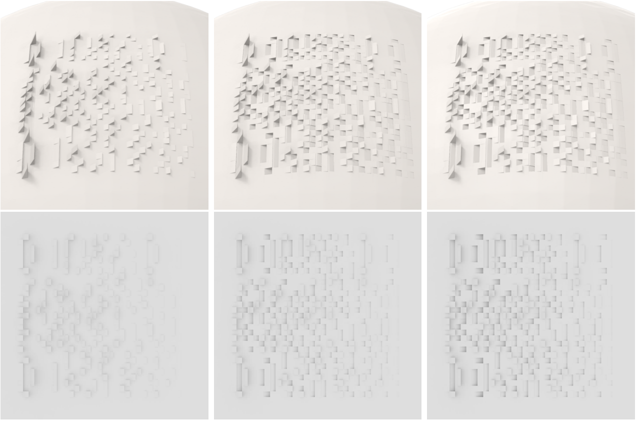


Figure 14: *Reflection elimination 1D illustration. Reflective light decreases shadow contrast under the directional light (a). By deforming and further carving we reduce this effect (b).*



$\alpha = 1,\ \beta = 0$ $\alpha = 0.5,\ \beta = 0.5$ $\alpha = 0,\ \beta = 1$

Figure 15: *Optimized 3D-QR-codes' geometry model (top row) and ambient QR image (bottom row) with different α and β. The average carving depths are 1.61mm, 1.10mm, and 1.04mm, from left to right. The visual disparities are 1.66%, 2.67%, and 3.44%, respectively.*

The average carving depth tends to fewer modulations of the shape, which is a rather conservative option.

As illustrated in Figure 15, the visual disparity term only ($\alpha = 1, \beta = 0$) drives to outward carving without inward carving; while, the carving depth term only ($\alpha = 0, \beta = 1$) drives to more inwards than outward carving. While, $\alpha = 0.5, \beta = 0.5$ provides a good compromise between both carving styles. Thus, we use these two parameter values in the experiments.

We note that both terms are normalized by the initial value, i.e., $\|\mathbf{Q}_a^{(0)} - \mathbf{U}\|^2$ and $\|\bar{D}^{(0)}\|^2$, respectively, to make the parameters α and β in the range $[0, 1]$.

Optimization convergence. To evaluate the convergence of our optimization and post-processing step, we plot the change in the value of average carving depth and ambient disparity in terms of iterations. As shown in Figure 16, both objectives have a pronounced and rapid monotone descent, but the average carving depth in-

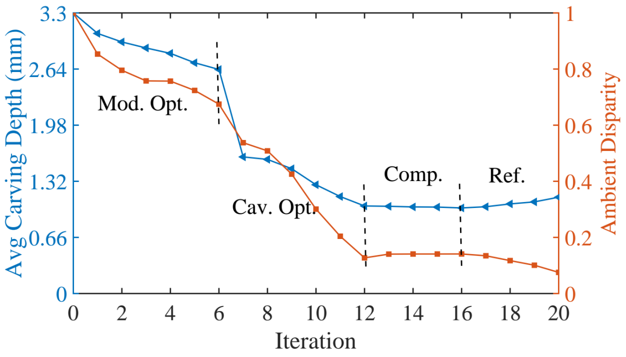


Figure 16: *Both average carving depth and visual disparity under ambient light decrease along optimization iterations while average carving depth increases in post-processing.*

Model	Ver.	Ang.(θ°)	ModT(s)	CavT(s)	PostT(s)
Cube	6	6	0.64	138.3	1.6e-3
Sphere	4	20	0.58	79.4	1.8e-3
Kitten	4	12	0.57	76.6	3.7e-3
Vase #1	4	12	0.54	75.8	2.9e-3
Vase #2	4	12	0.51	76.4	2.6e-3
Bunny	3	12	0.60	72.5	2.9e-3
Trophy	3	6	0.59	71.7	3.3e-3

Table 1: *Results statistics. Columns are QR version (Ver.), angle of the directional light (Ang.), time (in seconds) for module optimization (ModT), carving optimization (CavT), and post-processing (PostT).*

creases slightly in the post-processing stage, especially eliminating reflections. Since the post-processing makes a necessary condition to a hard constraint, a small increase of the objective function is reasonable. Besides, the corresponding simulated results for each step are demonstrated in Figure 2.

6. Results

Setup. We implemented our technique on an Intel® Core(TM) i5-6200U CPU @ 2.3GHZ and 8GB RAM. The geometry processing library *libigl* [JP*18] is used for mesh processing, remeshing and editing. Each module belonging to a 3D-QR-code is uniformly sampled by 5×5 vertex points in both simulation and carving process. We use the Ceres solver [AMO] for solving the objective function. The running time for both optimizations and post-processing are in Table 1.

We observe that the most time-consuming step is carving optimization since it performs numerical differentiation to evaluate gradients. Nevertheless, convergence of carving optimization is quite fast and in our experiments stayed below 5 minutes.

An SLA-based 3D printer (UnionTech Lite 600HD) with opaque photopolymer materials is used to fabricate the models. The nozzle diameter is 0.1mm of which provides sufficient printing precision. Compared to [KYJ*18] and [PLL*19], we use higher precision 3D printers to manufacture our 3D-QR-codes due to shallow carving and fine feature. The features of our 3D-QR-codes are very small, within the support-free tolerance allowed by the 3D printer, thus no additional support structures were required. Different printing orientations have little effect on decoding robustness, that is an advantage of our 3D-QR-codes over [KYJ*18] and [PLL*19]. We have no restrictions on the printing orientations for all the models shown in the paper.

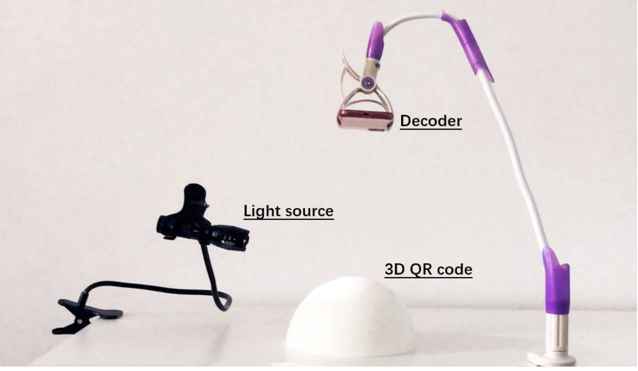


Figure 17: *Environment setup consists of a spotlight, a standard QR decoder and the 3D printed model with its optimized 3D-QR-code.*

Evaluation. For evaluation, we compute and print our 3D-QR-codes on different surfaces. We test decoding robustness of our 3D-QR-codes engraved on high curvature surfaces and thin shell models. All our printing results except cube and sphere are shown in Figure 25 and can be decoded robustly under a specified directional light.

We have performed various experiments to validate the decoding robustness of the fabricated 3D-QR-codes (Figure 17). We follow the criteria of *under 3 seconds* for successful QR decoding. We evaluated the robustness of our 3D-QR-codes in various aspects. We design four groups of experiments and quantify their decoding success rate. All experiments are repeated 50 times. We test the decoding rates (i) using different devices and readers, (ii) under different ambient environments, (iii) scanned via different scanning angles, and (iv) different directional light angles. The results are described and summarized in Figure 18.

We observe that our 3D-QR-codes can be decoded robustly by mainstream mobile phones and decoders (Figure 18(a)). The success rate of 3D-QR-codes decreases as the ambient light intensifies (Figure 18(a)). Although enhancing directional light could overcome this drawback, it would require powerful light sources. Another option is using our 3D-QR-codes under low-medium ambient light environments.

Figure 18 (c) demonstrates robustness to scanning angle, comparing between the original 2D QR code, 3D-QR-code generated by [PLL*19] and our method. Results show that our embedded QR codes are more robust to scanning angles than [PLL*19] although both 3D codes sacrifice certain robustness compared to 2D QR codes.

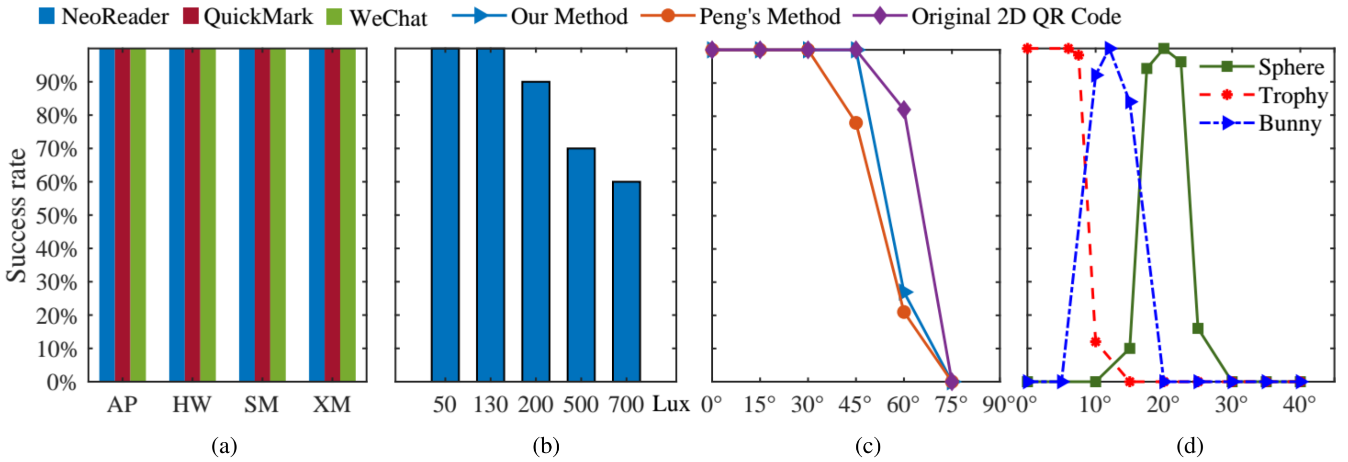


Figure 18: *Robustness of our 3D-QR-codes in terms of different decoders and mobile phones (a), different lighting environments (b), different scanning angles (c) and different directional light angles (d). AP, HW, SM, XM denote iPhone, Huawei, Samsung and Xiaomi mobile phones, respectively.*

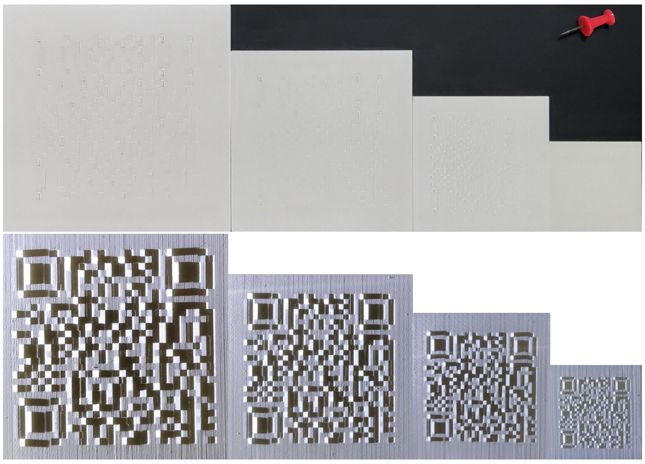


Figure 19: *3D-QR-code printed on different region sizes under ambient (top) and directional lights (bottom). Lengths (left-to-right) are 73.8mm, 61.5mm, 49.2mm, 36.9mm.*

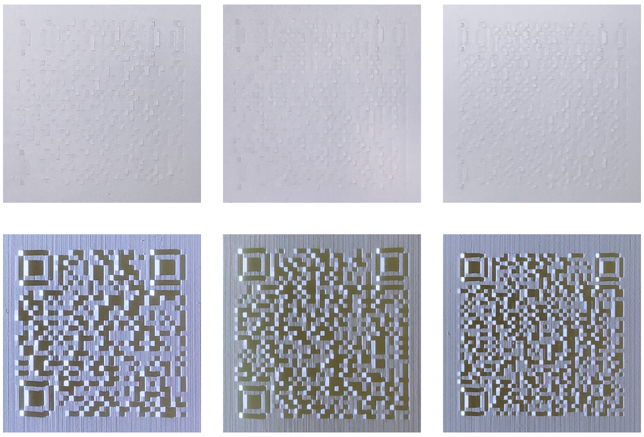


Figure 20: *Different QR-code versions under ambient (top) and directional lights (bottom) on the same model size (61.5mm). QR version is 4, 5, 6, from left to right, respectively.*

Figure 18 (d) demonstrates the robustness to directional light angles. We test on the sphere, trophy, and bunny models whose minimal light angles are $20^\circ, 6^\circ$, and 12°, respectively. We observe that our 3D-QR-code works well within a range of $\pm 2.5^\circ$.

We examined the printed 3D-QR-codes on a cube with different sizes, colors and versions. 3D-QR-codes printed with different sizes are shown in the first row of Figure 19, in which the module size is 1.8mm, 1.5mm, 1.2mm, 0.9mm from left to right, respectively. We observe that the success rate drops rapidly to 50% when the module size is less than 0.9mm. Modules of size $\leq$ 0.9mm consists significant noise due to printing accuracy which would be ignored in large sized modules. We suggest setting the module size of 3D-QR-codes no less than 1.2mm. In Figure 20, we show modules of sizes 1.85mm, 1.67mm, 1.43mm, from left to right, respectively. We have also tested 3D-QR-codes using different printing colors and showing they are robust to reading under directional light. Results are shown in Figure 21.

Comparison. To demonstrate the effectiveness of our optimization, we compared our 3D-QR-codes with [PLL*19]. We have measured the average shape carving of 3D-QR-codes embedded on the same objects and optimized using two methods. Our results are superior to those from [PLL*19] in terms of visual unobtrusiveness with shallower carving depth of approx. 50% lower (see Figure 22).

We also compare our method to relevant previous works w.r.t. visual quality. Under the same ambient environment, given the same 2D QR code and a 3D flange model, the carving depths of 3D-QR-codes generated by [KYJ*18], [PLL*19] and our method are

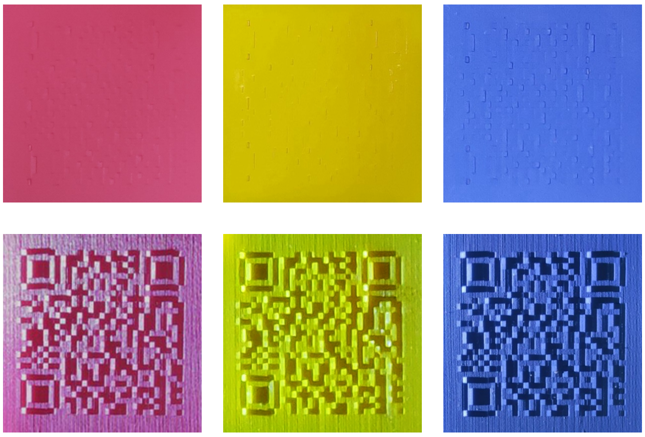


Figure 21: *A 3D-QR-code printed with different colors under ambient (top) and directional lights (bottom).*

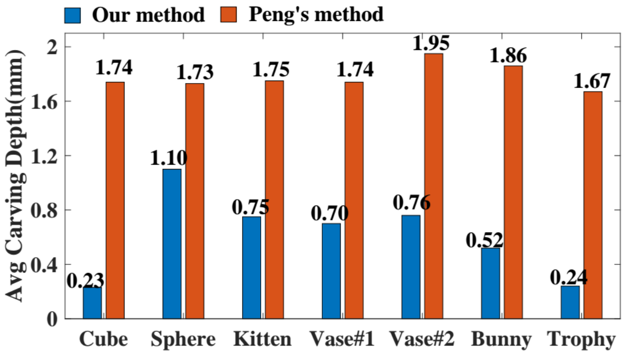


Figure 22: *Comparison of average shape carving by our method and [PLL* 19]'s method.*

2.18, 1.52, 0.31mm, respectively (Figure 23). Our method yields a minimal shape carving.

In Figure 24, we generate 3D-QR-codes at the same position on the Vase#1 model. The slice of 3D-QR-code for [PLL*19] in (a) shows that the average carving depth is large and introduces regions where the vase is punctured (red arrows). Our 3D-QR-code is significantly shallower.

Limitations. Our 3D-QR-codes embedding limitation is highly curved shapes. Although our method allows embedding QR-codes on non-flat and bumpy regions, phenomena such as self-occlusions, non-continuous projection and invisible QR regions may occur. To ensure readability, it is possible to overcome some of these problems using directional lights with large angles (in Figure 26, the minimal directional light angle is 32°). Additionally, highly curved regions may become fragile due to our carving editing on the surface and easily damaged resulting decoding failures (see Figure 26). In practice, we recommend embedding the QR code into relatively moderately cured regions whose minimal directional light angle is no more than 20°.

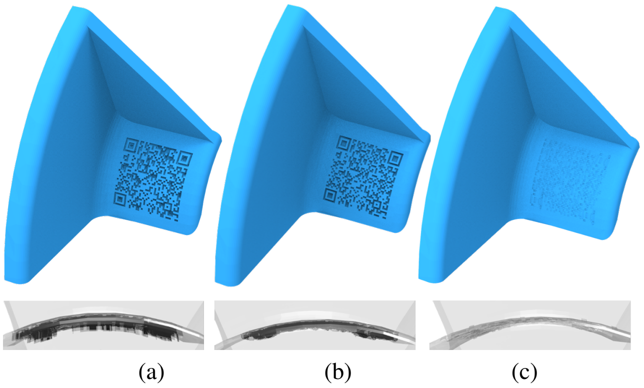
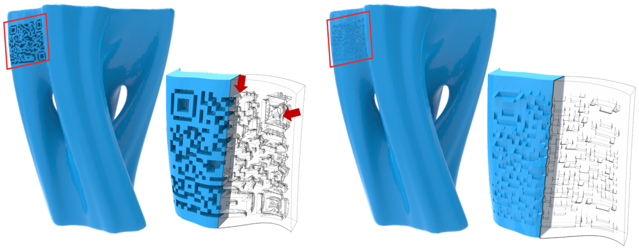


Figure 23: *Comparison of 3D-QR-code carving on a flange model using the method of [KYJ* 18] (a), [PLL* 19] (b) and ours(c). 1D views on bottom row show carving depths with average depth of* 2.18, 1.52, 0.31*mm respectively.*



(a) [PLL*19]'s method (b) Our method

Figure 24: *Comparison of 3D-QR-code carving on a thin vase model of* 2mm *thickness between [PLL* 19]'s method (a) and ours (b). Zoom-ins show that the average carving depth* (1.74*mm) is close to thickness of the vase where the vase is punctured (red arrows) (a), while our result is shallow* (0.70*mm).*

7. Conclusions and Future Work

In this paper, we introduced a novel 3D-QR-code embedding technique for printed objects. We computed 3D-QR-code structures by carefully carving black and white geometry modules on the surface, thus encoding the QR code. The 3D-QR-code is carved on the surface w.r.t. a predefined light direction. Hence, QR decoding is possible only by casting the directional light onto the 3D printed object. Our 3D-QR-code has little influence on the shape's surface and it is unobtrusive under ambient light.

To compute the correct geometry carving of the black and white QR modules, we rely on a light simulation instead of computing it directly from the scene parameters. Printed 3D objects have a digital representation which we use to efficiently simulate and approximate the lighting in the scene. Calculating light effects accurately involves complex calculations. Instead, our model is an approximation of lighting phenomena which we calibrate to further adjust it to the real scenes. For example, while considering inner specular reflections, we ignore inner scattering. Similarly, our reflection computation is limited, and thus translucent or transparent materials cannot be currently handled.

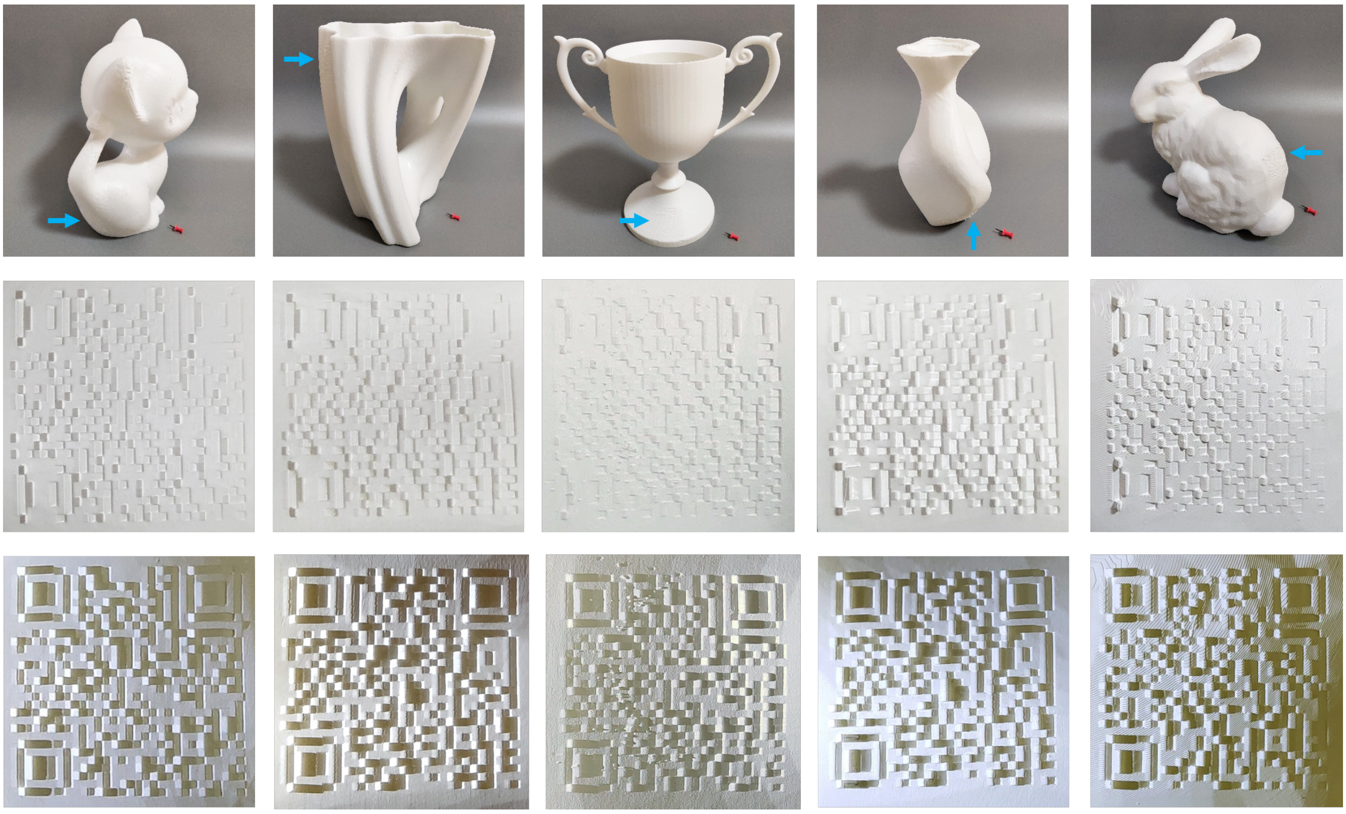


Figure 25: *3D-QR-codes printed on models with different geometries and light angles. Top-to-bottom are the 3D printed model and zoom-ins on the 3D-QR-code under the ambient and directional light environments. Models left-to-right: Kitten, Vase#1, Trophy, Vase#2, Bunny.*

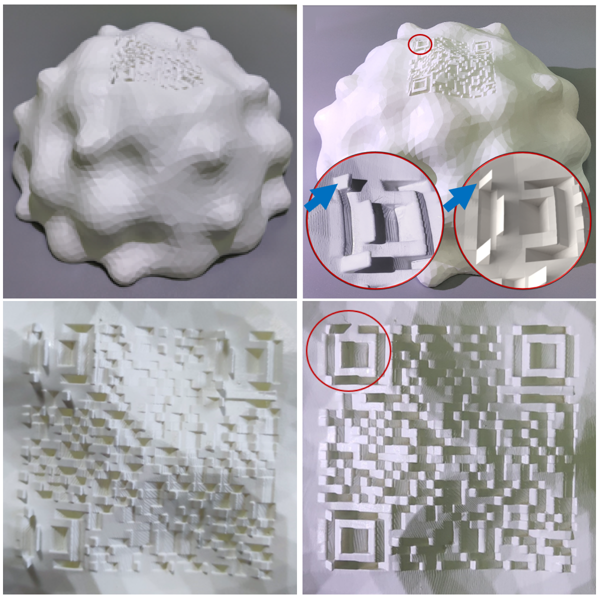


Figure 26: *3D-QR-code embedded on a highly bumpy surface. The directional light angle is 32°. The outward carving features circled in red in 3D-QR-code are too sharp and thus unstable.*

When generating 3D QR-codes, we do not optimize the position of QR region, and the scanning view is fixed to top-view. In the future, we would like to include in our optimization framework also position optimization and direction of scanning view to achieve better results.

Acknowledgements

We thank all the anonymous reviewers for their valuable suggestions. This work is supported by grants from NSFC, China (61972232) and State Key Laboratory of Virtual Reality Technology and Systems, China (VRLAB2019A01).

References

[AM10] Alexa M., Matusik W.: Reliefs as images. *ACM Trans. Graph. 29*, 4 (July 2010), 60:1–60:7. doi:10.1145/1778765.1778797. 2

[AM11] Alexa M., Matusik W.: Images from self-occlusion. In *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging* (New York, NY, USA, 2011), CAe '11, ACM, pp. 17–24. doi:10.1145/2030441.2030445. 2

[AMO] Agarwal S., Mierle K., Others: Ceres solver. http://ceres-solver.org. 8, 9

[BBAM12] Bermano A., Baran I., Alexa M., Matusk W.: Shadowpix: Multiple images from self shadowing. *Comput. Graph. Forum 31*, 2pt3 (May 2012), 593–602. doi:10.1111/j.1467-8659.2012.03038.x. 2

[BL13] Bors A. G., Luo M.: Optimized 3d watermarking for minimal surface distortion. *IEEE Transactions on Image Processing 22*, 5 (May 2013), 1822–1835. doi:10.1109/TIP.2012.2236345. 2

[BMT13] Barrera J. F., Mira A., Torroba R.: Optical encryption and qr codes: Secure and noise-free information retrieval. *Opt. Express 21*, 5 (Mar 2013), 5373–5378. doi:10.1364/OE.21.005373. 2

[CCLM13] Chu H.-K., Chang C.-S., Lee R.-R., Mitra N. J.: Halftone qr codes. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 217:1–217:8. doi:10.1145/2508363.2508408. 2

[Cox] Cox R.: Qart codes. http://research.swtch.com/qart. 2

[GALV14] Garateguy G. J., Arce G. R., Lau D. L., Villarreal O. P.: Qr images: Optimized image embedding in qr codes. *IEEE Transactions on Image Processing 23*, 7 (July 2014), 2842–2853. doi:10.1109/TIP.2014.2321501. 2

[HCF11] Huang H. C., Chang F. C., Fang W. C.: Reversible data hiding with histogram-based difference expansion for qr code applications. *IEEE Transactions on Consumer Electronics 57*, 2 (May 2011), 779–787. doi:10.1109/TCE.2011.5955222. 2

[Hun04] Hunt R.: The reproduction of colour sixth edition. *John Wiley & Sons* (2004). 5

[HW65] Hawkins G. S., White J. B.: *Stonehenge Decoded*. Doubleday Books, 1965. 1

[HWW12] Hsu F.-H., Wu M.-H., Wang S.-J.: Dual-watermarking by qr-code applications in image processing. In *Proceedings of the 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing* (Washington, DC, USA, 2012), UIC-ATC '12, IEEE Computer Society, pp. 638–643. doi:10.1109/UIC-ATC.2012.91. 2

[JP*18] Jacobson A., Panozzo D., et al.: libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 9

[JTC15] JTC1/SC I.: Iso/iec 18004:2015(e). information technology - automatic identification and data capture techniques - qr code bar code symbology specification. 117. 4, 5

[KYJ*18] Kikuchi R., Yoshikawa S., Jayaraman P. K., Zheng J., Maekawa T.: Embedding QR codes onto b-spline surfaces for 3d printing. *Computer-Aided Design 102* (2018), 215–223. doi:10.1016/j.cad.2018.04.025. 1, 2, 3, 9, 10, 11

[Lan02] Landis H.: Production-ready global illumination. *SIGGRAPH Course Notes* (2002). 5

[LHLL15] Lin S. S., Hu M. C., Lee C. H., Lee T. Y.: Efficient qr code beautification with high quality visual content. *IEEE Transactions on Multimedia 17*, 9 (Sept 2015), 1515–1524. doi:10.1109/TMM.2015.2437711. 2

[LNNZ17] Li D., Nair A. S., Nayar S. K., Zheng C.: Aircode: Unobtrusive physical tags for digital fabrication. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (2017), UIST '17, pp. 449–460. 3

[LSZ*14] Lu L., Sharf A., Zhao H., Wei Y., Fan Q., Chen X., Savoye Y., Tu C., Cohen-Or D., Chen B.: Build-to-last: strength to weight 3d printed objects. *ACM Transactions on Graphics 33*, 4 (jul 2014), 1–10. doi:10.1145/2601097.2601168. 2

[MLYZ19] Maia H. T., Li D., Yang Y., Zheng C.: LayerCode: optical barcodes for 3d printed shapes. *ACM Transactions on Graphics 38*, 4 (jul 2019), 1–14. doi:10.1145/3306346.3322960. 3

[MP09] Mitra N. J., Pauly M.: Shadow art. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 156:1–156:7. doi:10.1145/1618452.1618502. 2

[OMA98] Ohbuchi R., Masuda H., Aono M.: Watermarking three-dimensional polygonal models through geometric and topological modifications. *IEEE Journal on Selected Areas in Communications 16*, 4 (May 1998), 551–560. doi:10.1109/49.668977. 2

[OMT02] Ohbuchi R., Mukaiyama A., Takahashi S.: A frequency-domain approach to watermarking 3d shapes. *Comput. Graph. Forum 21*, 3 (2002), 373–382. 2

[PLL*19] Peng H., Lu L., Liu L., Sharf A., Chen B.: Fabricating QR codes on 3d objects using self-shadows. *Computer-Aided Design 114* (2019), 91–100. doi:10.1016/j.cad.2019.05.029. 1, 2, 3, 6, 9, 10, 11

[RGS09] Ritschel T., Grosch T., Seidel H.-P.: Approximating dynamic global illumination in image space. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2009), I3D '09, ACM, pp. 75–82. doi:10.1145/1507149.1507161. 5

[SPSH14] Schüller C., Panozzo D., Sorkine-Hornung O.: Appearance-mimicking surfaces. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 216:1–216:10. doi:10.1145/2661229.2661267. 2

[US13] Umetani N., Schmidt R.: Cross-sectional structural analysis for 3d printing optimization. In *SIGGRAPH Asia 2013, Hong Kong, China, November 19-22, 2013, Technical Briefs* (2013), Chen B., Sharf A., (Eds.), ACM, pp. 5:1–5:4. doi:10.1145/2542355.2542361. 2

[Vis] Visualead: Visual qr code generator. http://www.visualead.com/. 2

[VR12] Vongpradhip S., Rungraungsilp S.: Qr code using invisible watermarking in frequency domain. In *2011 Ninth International Conference on ICT and Knowledge Engineering* (Jan 2012), pp. 47–52. doi:10.1109/ICTKE.2012.6152412. 2

[WCT*15] Wang W., Chao H., Tong J., Yang Z., Tong X., Li H., Liu X., Liu L.: Saliency-preserving slicing optimization for effective 3d printing. *Comput. Graph. Forum 34*, 6 (2015), 148–160. doi:10.1111/cgf.12527. 2

[WLDB08] Wang K., Lavoue G., Denis F., Baskurt A.: A comprehensive survey on three-dimensional mesh watermarking. *IEEE Transactions on Multimedia 10*, 8 (Dec 2008), 1513–1527. doi:10.1109/TMM.2008.2007350. 2

[WSHL18] Wei C., Sun Z., Huang Y., Li L.: Embedding anti-counterfeiting features in metallic components via multiple material additive manufacturing. *Additive Manufacturing 24* (2018), 1 – 12. 3

[YPRI17] Yang Y., Pintus R., Rushmeier H., Ivrissimtzis I.: A 3d steganalytic algorithm and steganalysis-resistant watermarking. *IEEE Transactions on Visualization and Computer Graphics 23*, 2 (Feb 2017), 1002–1013. doi:10.1109/TVCG.2016.2525771. 2

[ZLP*15] Zhang X., Le X., Panotopoulou A., Whiting E., Wang C. C. L.: Perceptual models of preference in 3d printing direction. *ACM Trans. Graph. 34*, 6 (2015), 215:1–215:12. doi:10.1145/2816795.2818121. 2

[ZLW*16] Zhao H., Lu L., Wei Y., Lischinski D., Sharf A., Cohen-Or D., Chen B.: Printed perforated lampshades for continuous projective images. 2

[ZPZ13] Zhou Q., Panetta J., Zorin D.: Worst-case structural analysis. *ACM Trans. Graph. 32*, 4 (2013), 137:1–137:12. doi:10.1145/2461912.2461967. 2