https://doi.org/10.2352/ISSN.2169-4451.2019.35.115

Barcodes on Non-Flat Surfaces

Matthew Gaubatz; HP, Inc.; Seattle, WA, USA.
Robert Ulichney; HP, Inc.; Stow, MA, USA.

Abstract

Product marking systems play a key role in many existing manufacturing systems. With the rise of 3D printing applications, there is host of new opportunities to provide object tagging capabilities directly on product surfaces. Ideally, a solution could take advantage of existing infrastructure, which in many instances is adapted to use 1D barcodes. This paper explores different mechanisms for rendering barcodes directly onto object surfaces, including the benefits and challenges associated with mapping a 1D marking onto a 3D surface v. forms of pre-compensation for perspective correction such that the marking can be interpreted as flat from certain viewpoints. Objects with circular geometry are of key interest as they highlight differences between these two approaches.

1. Introduction

To fully enable item tagging and tracking using existing scanning infrastructure, a scheme for producing 3D printed parts must be able to render scannable barcodes [1,2]. Recent work has characterized techniques for boosting barcode data capacity, particularly that of 2-D barcodes [3-6], and has also explored convenient, easy-to-deploy software solutions for scanning them [7,8]. This paper focuses on the more fundamental problem of rendering codes directly onto 3D surfaces. Spherical, semi-spherical and/or cylindrical objects represent important examples since there is no way to extend a barcoding scheme that is effective in flat geometries to be readable from all possible views of the object.

One assumption made is that a printing system is capable of rendering contrast differences that create a readable signal; the samples discussed involve bi-tonal processes. The approach taken herein is to compare two options for rendering barcodes on surfaces:

- Surface mapping: rendering the design to conform to an object's surface; and
- Pre-compensation: pre-rendering a design to appear as it appears on a flat surface from certain perspectives when rendered on non-flat objects.

A standard mechanism for printing a 3D object involves representing its surface. There are numerous file formats that support creation of 3D printed objects. Some of the most common include representations where the surface of the object is described by a mesh. Various formats can support texture mapping, whereby image resources may be mapped onto the surface of an object [9]. This paper focuses on using that functionality, texture mapping, as a mechanism to create barcodes on top of mesh-based object representations. How and/or why any pre-compensation could be applied prior to mapping a design onto a surface is discussed, and tradeoffs in both rendering and reading are compared.

This paper is organized as follows. Section 2 summarizes aspects of the geometry of 1D codes on circular shapes. Section 3 discusses examples of codes mapped to object surfaces v. pre-compensated codes. Section 4 examines 3D printed examples and section 5 concludes the paper.

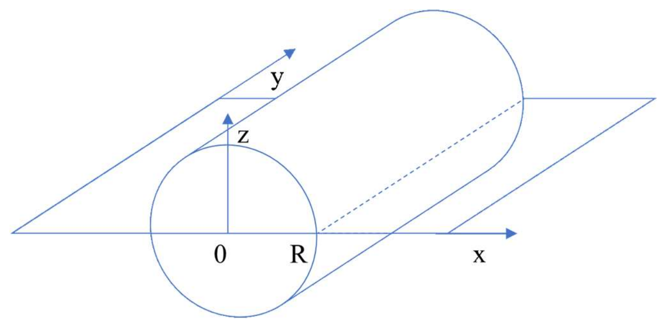

Figure 1. *Diagram of the intersection a radius R cylinder and a bisecting plane. The region of interest is the portion of the surface above the plane.*

2. Geometry of 1D Barcodes on Circular Curves

In order to compare texture-mapping with pre-compensation for rendering and reading 1D barcodes on curved surfaces, it is helpful to consider how circular geometry affects 1D barcodes, specifically, how the positions of the bars in the direction of the code are affected by mapping them onto a curved surface. When the barcode is flat, there is a single (1D) direction associated with the code, and the sizes and spacings of the bars along this trajectory define the data represented by the code. When the code is rendered onto a curved surface, when viewed from a distance above, i.e., the direction of the z-axis, the positions and sizes of the bars in the direction of the x-axis change. Since the perspective is a function of viewing distance, it is convenient to approximate these positions as viewed by an observer at infinity, such that lines between the respective bars and the observer are parallel.

Consider, for instance, a cylinder or sphere of (dimensionless) radius R, such as illustrated in Fig. 1. In a 3D space, the upper half of the intersection of this object and the xz plane can be modeled via

$$z = \sqrt{(R^2 - x^2)}. \quad (1)$$

The distance between any two points on the curve is given by the standard expression for its arc length,

$$\int_a^b \frac{R}{\sqrt{R^2-x^2}} dx = R \sin^{-1}\frac{b}{R} - R \sin^{-1}\frac{a}{R}. \quad (2)$$

If we assume that a barcode is rendered on the top half such that the entire code is length $L \leq \pi R$, and that it is rendered symmetrically around $x = 0$, the angle subtended by the code on the surface of the cylinder is $\pi \cdot L/(\pi R) = L/R$ radians, and the extent of the code on the x-axis is given by $[-R \sin^{-1} L/2R, R \sin^{-1} L/2R]$. Let $x_{flat\ surface}$ represent the horizontal position within this centered barcode on a flat surface. When the barcode is rendered onto the curved surface, the horizontal position of each element after this mapping has been applied, denoted $x_{curved\ surface}$, is

$$x_{curved\ surface} = R \sin \frac{x_{flat\ surface}}{R}. \quad (3)$$

This equation also suggests a relationship can be used to pre-compensate for the shape of the 3D surface such that barcode would *appear* flat in the x-direction when rendered onto the surface. Pre-compensation requires a bit more care in that it is important to note that even if the barcode does appear as it is flat when rendered on the curved surface, the size is contracted by a factor of $\pi/2$. An expression relating the horizontal positions of the pre-compensated signal to the original positions is as follows.

$$x_{pre-compensated} = R \sin^{-1} \frac{\frac{x_{uncompensated}}{R}}{\frac{\pi}{2}}. \quad (4)$$

Cascading these two equations indeed shows that x positions in the curved surface map to contracted x positions in the original (flat) barcode, i.e.,

$$x_{curved\ surface} = R \sin \frac{R \sin^{-1} \frac{\frac{x_{uncompensated}}{R}}{\frac{\pi}{2}}}{R} = \frac{x_{uncompensated}}{\frac{\pi}{2}}. \quad (5)$$

Figure 2. *Example barcode, in original form (top), with x-coordinates modified via a mapping equivalent to rendering it 180 degrees around on the surface of a cylinder (bottom).*

Figure 3. *Example barcode from Fig. 2, pre-compensated (top) such that when rendered on a surface such as defined in Fig. 1, the barcode appears dilated but readable as if it was rendered on a flat surface (bottom).*

Fig. 2 illustrates the effects of the operation described in (3). If the barcode is rendered 180 degrees around a surface with the geometry of the plot in Figure 1 and is viewed from above, the horizontal positions of elements within the design will be contracted as depicted. In this contracted form, the code is difficult to read with a standard reading device; and while it would be possible to create a pre-processing stage to recover from this distortion prior to decoding, the approach would be tied to this particular surface, and furthermore, the recovered image would contain an uneven number of samples of each bar due to the non-linearities present. An alternate solution to create a readable rendering on a curved surface is to pre-compensate the marking via (4), as shown in Fig. 3.

In higher dimensions, some of these effects are more pronounced, but along paths that are perpendicular to the bars in the design, similar trends are observed. Note also that the optimal pre-compensated shape is a function of the distance between the surface and the acquisition device. In practice, this consideration is less of a concern simply because using the compensation technique described in Fig. 3 is reasonably effective.

3. Surface Mapping v. Pre-compensation

Geometric analysis of the curved barcode suggests that a barcode will become unreadable when then relative dilation/contraction due to mapping it onto a surface is significant. While equation (3) implies at what point a bar will be interpreted as existing in the wrong region, the efficacy of different reading schemes with and without pre-compensation must be tested. An experiment was conducted using a cylinder to determine at what subtended angle a barcode might become unreadable, and to verify that in fact, the pre-compensation mapping defined by (4) improved reading performance. This test involved seven barcodes wrapped around the surface of a cylinder, as well as seven barcodes pre-compensated to appear flat while subtending the same angles as the corresponding uncompensated markings. The design was scanned with a mobile phone and a USB laser barcode scanner. The cylinder was 1.5 inches in diameter, and the barcode sizes were chosen to subtend seven different angles in between 90 and 180 degrees, each separated by 15 degrees. Each barcode was 0.75 inches tall.

An image of some of the codes used for this test is give in Fig. 4, and results are listed in Table 1. Without any pre-compensation applied, the laser scanner was only able to reliably read codes the subtending 90 degrees (a few scans of the 105-degree design succeeded, but this result was not robustly repeatable) and a mobile phone was able to read codes subtending up to 135 degrees. The difference in performance between these two devices remained just as pronounced with the pre-compensated imagery. The laser scanner read barcodes subtending up to 135 degrees, and the mobile phone scanned them all.

Table 1: Maximum scannable angle

Laser	90 (105) degrees
Laser + pre-compensated	135 degrees
Mobile phone	135 degrees
Mobile phone + pre-compensated	180 degrees

Figure 4 *Cylinders with barcodes subtending 90 through 180 degrees arc, spaced by increments of 15 degrees. The markings include both original (uncompensated, left) barcodes, as well as barcodes pre-compensated (right) to appear flat when viewed on this surface from some angles.*

One of the benefits of using the uncompensated imagery is that despite any additional complexity of creating the actual surface design, it is completely generic, i.e., it can be applied to any object in the same manner. A large class of objects exhibit surfaces with regularity properties such that this approach can be effective. Simple discontinuities can cause it to fail, but those artifacts would also impact the pre-compensation approach, though in a slightly different way. Also, despite the fact that the uncompensated imagery was not *as* readable, the mobile phone, in particular, performed well. The code subtending 135 degrees covers 75% of the visible half of the cylinder in actual 3D space, and over 92% of the cylinder as it appears from above. Extending a reader to handle a wider range of potential distortions could also lead to further gains; various techniques have been successfully applied to both 1D [10-11] and 2-D codes [11-14].

The pre-compensation approach, though it results in a simple-to-acquire signal, involved more surface-specific manipulations in order to render instructions to create the design, even in this simple example with a cylinder; the approach was equivalent to determining how to pre-distort the barcodes such that it could be included via a (more universal) surface mapping operation. With a more complex surface this cost could become prohibitively large, or in some cases, may fail if the object features enough discontinuities. Even if the pre-compensation could be achievably applied, it would still require geometric information about the object to recorded and/or inferred, which could, further constrain the workflow. The key benefit, however, is notable improved decoding performance.

4. 3D Print Testing

To evaluate surface barcode rendering capabilities, test objects were created to include spherically convex and conically concave surfaces. 3D color printing was used to create different examples, and the barcodes, as with the pre-compensated examples appearing on the cylinder, were rendered such that the codes varied in the x-direction just they would in a flat geometry. This choice is preferable for smaller spherical shapes as it ensures the presence of a code at a constant with across the object; as a result, there are multiple locations across the surface for a scan to take place. Different combinations of colors were used in part to evaluate robustness to different amounts of contrast. One example black-and-white semi-sphere is pictured in Fig. 5. This object scanned easily with multiple devices, and on several parts of the object. Web-based scanning tools as well as compiled apps were found to read this pattern. Other colors were found to succeed as well. Fig. 6. Illustrates a lower contrast example in yellow in white that was still decodable. In some cases, an existing luminance-based barcode reader was able to pick up barcodes represented in other color channels. This observation corroborates the idea that some readers have special enhancement built into the decoding process (such as an ability to recover a payload in the presence of significant non-linear distortion; see previous section).

Figure 5. *Example 1D barcodes rendered across a spherically convex surface. This example is created with a standard black on white design and is readily scannable by various devices and scanning programs. Other examples rendered in different colors present decoding challenges, but this example demonstrates that the problem of reading a code on an object with a curved surface is handled by the proposed approach.*

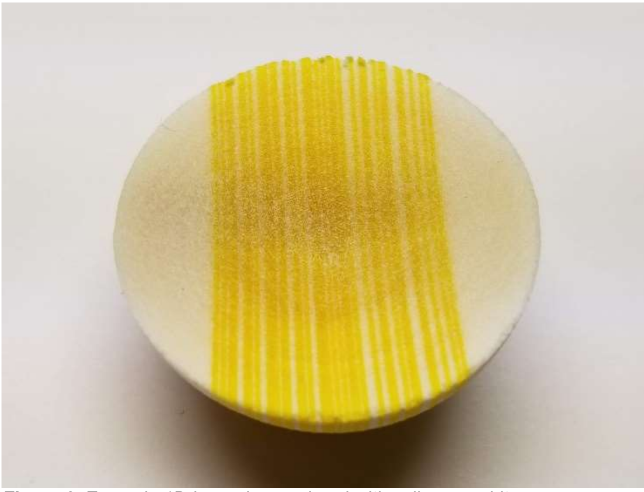

Figure 6. *Example 1D barcodes rendered with yellow on white, across a conically concave surface. The convex portion of this object (on the back side of the object in this view) was somewhat easier to interpret, in part because the singular point in the center of the cone, which can be seen surrounded by contours that cut across the bars in the code.*

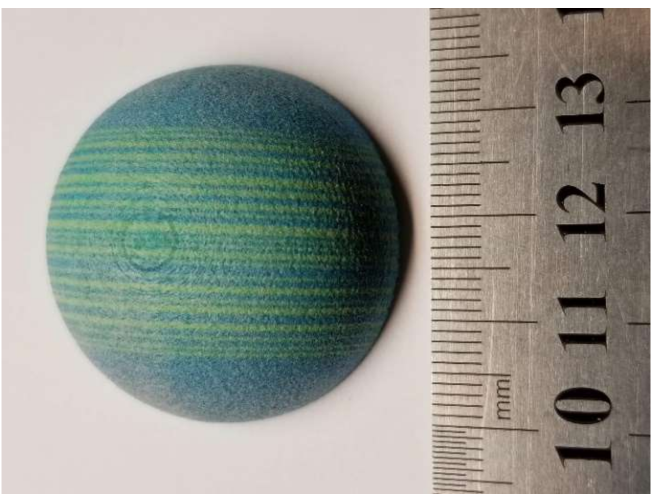

Figure 7. *An example semi-spherical object with a barcode rendered using cyan and green, with a ruler depicting the scale. This image represents an instance of how the reduced contrast of the colors helps reveal shadows induced by both global structure and local textural irregularities.*

Some of the test prints that proved more difficult to read included examples that used multiple colors (cyan and green, purple and black). Many of the challenges, however, had to do with distortions driven by factors other than surface geometry. As can be expected, lighting quality played a role in the outcome, both as a function of the geometry of the object at the macro level, as well as the local textural elements. Both effects are noticeable in Fig. 7 above. Despite these issues, because compact data embeddings create more opportunities to enhance printed objects, developing smaller markings is a continuing challenge. Indeed, the objects involved in this test were smaller in diameter than the cylinder used for testing pre-compensated codes. Nonetheless, because some of the color combinations yielded easily scannable codes it is possible that the geometry problems involved with reading the objects can be overcome with an appropriately strong signal.

5. Conclusion

In the studies performed herein, rendering barcodes that conformed to the shape of curved surfaces was found to be a surprisingly effective method for tagging the surfaces, particularly when used in conjunction with mobile phone scanners. At the same time, pre-compensating for surface features in rendered codes enabled both phones and laser scanners to decode markings on a wider range of surfaces. It was also found to be possible to 3D print codes onto objects with circular shapes and to successfully read them back, even when rendering them in different colors (and evaluating them at multiple positions across the surface). The convex surfaces were slightly easier to read than the concave surfaces, possibly due to the increased smoothness. One of the benefits of pre-compensation is that it allows existing scanning tools to operate with minimal changes, whereas surface mapping requires object-dependent, or at least requires curvature-dependent, signal restoration to be more broadly applicable. Future work will involve extending both types of capabilities to larger classes of markings.

References

[1] International Standard ISO/IEC 18004:2015, Third edition. “Information technology – Automatic identification and data capture techniques – QR Code bar code symbology,” 117 pp., 2015.

[2] International Standard ISO/IEC 16022:2006(E), Second edition 2006-09-15, “Information technology – Automatic identification and data capture techniques – Data Matrix bar code symbology specification,” 142 pp., 2006.

[3] O. Bulan and G. Sharma, “High-Capacity Color Barcodes: Per Channel Data Encoding via Orientation Modulation in Elliptical Dot Arrays,” IEEE Transactions on Image Processing, vol. 20, no. 5, pp. 1137-1350, 2011.

[4] S. Simske, M. Vans and G. Adams, “Error-Correcting Code (ECC) and Module Size Considerations in 2D Aztec Barcode Readability,” JIST, vol. 54, no. 6, November 2010.

[5] D. Parikh and G. Jancke, “Localization and Segmentation of a 2D High Capacity Color Barcode,” proceedings of the IEEE Workshop on Applications of Computer Vision, Copper Mountain, Colorado, 2008.

[6] M. Vans, M. Gaubatz and S. J. Simske. “Embedding a Standard within a Standard using Mobile Progressive Barcodes,” proceedings of Printing for Fabrication, Denver, Colorado, 2017.

[7] Cloud Vision API, https://cloud.google.com/vision.

[8] M. Gaubatz, M. Vans and S. J. Simske, “Extending the Reach of a Barcode-Based Imaging Ecosystem,” proceedings of Printing for Fabrication, Dresden, Germany, 2018.

[9] 8 Most Common 3D File Formats in 2019, https://all3Dp.com/3D-file-format-3D-files-3D-printer-3D-cad-vrml-stl-obj/.

[10] O. Gallo and R. Manduci, "Reading 1D Barcodes with Mobile Phones Using Deformable Templates," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no.9, pp. 1834-1843, 2011.

[11] A. Zamberletti, I. Gallo and S. Albertini, "Robust Angle Invariant 1D Barcode Detection," proceedings of the IAPR Asian Conference on Pattern Recognition, Okinawa, Japan, 2013.

[12] K. Li, F. Meng, Z. Huang and Q. Wang, “A Correction Algorithm of QR Code on Cylindrical Surface,” Journal of Physics: Conference Series, vol. 1237, pp. 022006, 2019.

[13] K.-T. Lay and M.-H. Zhou, "Perspective Projection for Decoding of QR Codes Posted on Cylinders," proceedings of the IEEE International Conference on Signal and Image Processing Applications, Kuching, Malaysia, 2017.

Author Biography

Matthew Gaubatz is a researcher at HP Labs in Seattle, Washington. His main interests are mathematics, security printing and extracting information from rendered surfaces. He has a Ph.D. in electrical engineering from Cornell University.