Annular Barcodes

In this publication[1] we present a generic design for a novel annular barcode. On round media, circular or annular barcodes are more natural and preferable instead of traditional rectangular barcodes. Especially on round media with a distinctly convex or concave conic surface, an annular barcode is even more preferable than a full circle, because printing the barcode in the center of the media might be much more complicated or imprecise. Our generic annular barcode design supports different barcode and module sizes and therefore different data sizes. It includes different marker, synchronization, and data areas. For improved data robustness a data protocol and error correcting codes similar to them in QR codes are suggested. The feasibility of the design is shown by an implementation of an efficient and effective barcode decoder in C++ and a series of tests with distorted and noisy pictures of our annular barcodes.

Robin Brügger, Paul Glendenning, Christoph Stamm | christoph.stamm@fhnw.ch

A barcode is an optical machine-readable representation of data relating to the object to which it is attached [Wiki]. Originally barcodes systematically represented data by varying the widths and spacings of parallel lines. These may be referred to as linear or one-dimensional (1D). Later they evolved into rectangles, dots, hexagons and other geometric patterns in two dimensions (2D). Although 2D systems use a variety of symbols, they are generally referred to as barcodes as well. Barcodes are usually scanned by special optical scanners called barcode readers, but smartphones with built-in cameras and interpretive software can do the same work.

There are applications and situations where circular or annular barcodes are more natural and preferable instead of traditional rectangular barcodes, because of the round form of the barcode medium. Especially on round media with a distinctly convex or concave conic surface, an annular barcode is even more preferable than a full circle, because printing the barcode in the center of the media might be much more complicated or imprecise.

Circular Barcodes

Circular 1D barcodes fulfill the required annularity. They were extremely popular barcodes used by CD/DVD retailers for tagging their items [TR]. Figure 1 shows such a circular barcode on the left hand side, made up of a series of concentric circles and typically based on a standard barcode symbology like I2of5 (Interleaved 2 of 5). The barcode is thus readable by the same devices used to read traditional barcodes. Its payload depends mainly on its radius and is usually only a few decimal digits (2 digits in the example). Its data density (bits/area) is quite low, because of the large redundancy in all directions.

A ShotCode, depicted in Figure 1 on the right hand side, is one of the few existing circular 2D barcodes [BC09]. It was developed to share links and can be read with low resolution mobile phone cameras or webcams. In some application ShotCode is not suitable either because of its limited payload (5 to 6 bytes) or because of its used circle center. Today, ShotCode has been widely replaced by QR codes (Figure 2), because of its very limited payload.

Since we are interested in annular barcodes with high data density and have not found any such codes, we have designed a novel barcode format for codes with a data density comparable to QR codes. As a proof of concept we also implemented an encoder and decoder for our annular barcodes.

Requirements

In addition to the basic requirement of an annular form, there are further requirements for our barcode design:

- *Minimum payload:* The payload of the barcode should be at least 25 alphanumeric characters. Assuming extend 8 bit ASCII code as character code results in a minimum payload of 25 bytes or 200 bits.

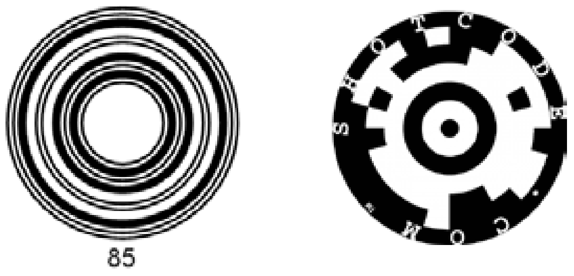

Figure 1: left) Circular Barcode (1D) [TR], right) Shotcode [Mud06]

1 This publication is based on Robin Brügger's bachelor thesis „Ring-shaped Barcodes“, 2015, in collaboration with NANO4U AG.

Figure 2: QR-Code

- *Support for different sizes:* The barcode design has to support different sizes, from a few millimeters to a dozen of centimeters. In a barcode with a maximum outer diameter of 8 mm, for example, the smallest module size should be not less than 0.2 mm or 5 times the lateral resolution of the scanner.
- *High robustness:* Perspective distorted and noisy images of barcodes should be either correctly decoded or rejected. The usage of error correction codes is allowed as long as the minimum payload is guaranteed.

Annular Barcode Design

In this Section our novel annular barcode design for a code with a data density comparable to QR codes is explained in detail. For reasons of improved robustness in large barcodes two additional small adaptions are useful. The three design types differ only in the number of repetitions of the so called start marker and radial zebra patterns (Figure 3). Figure 4 shows a typical barcode of type 1 generated by our encoder (left), and the same barcode with its highlighted special purpose zones for recognition, perspective correction and timing (right):

- *Solid black ring*: for locating the barcode in an image. The black ring surrounds a white circle;
- *Square markers*: for a rough perspective correction. They are arranged in a perfect square;
- *Start marker*: defines the beginning in the otherwise continuous circle. It is also used to determine the outer radius of the barcode;
- *Angular zebra pattern*: determines the angles between the modules. It is also used to achieve an accurate perspective correction. Therefore, the angular module count is always dividable by 8. Thus, it is possible to select four modules of the angular zebra pattern which lay in a perfect square, facilitating the perspective correction calculation;
- *Radial zebra pattern*: determines the radial timing of the rings (how far apart the rings are) and the reading order (counter-clockwise or clockwise);
- *Data zone*: The data zone can be interpreted as a curved table, addressing modules by their polar coordinates. The addressing of the modules is organized in concentric circles, beginning immediately after the radial zebra pattern and from outside towards the inner and smaller modules near the solid black ring. The data zone is usually used for payload interpretation (header information), payload, and error correction codes.

The size and shape of the barcode are determined by four basic dependent parameters:

- Type 1-3: The type defines the number of repetitions of the start marker and radial zebra pattern;
- Minimal module size: The minimal module size s_{min} is the minimal width and height of a single module. It is also the thickness of the individual rings;
- Minimal radius: r_{min} is the distance from the center of the barcode to the beginning of the inner square markers. The square markers are the smallest module-sized structure in the barcode. r_{min} has to be clearly larger than two-times s_{min}.
- Maximum radius: r_{max} is the distance from the center of the barcode to its outermost point. The maximum radius is dependent of the minimal module size and the minimal radius: $r_{\text{max}} = r_{\text{min}} + k\, s_{\text{min}}$, where $k - 2$ is the number of rings in the data zone.

The concrete values of the last three parameters depend mainly on the resolution of the printing and scanning devices, the available area, and the payload requirements.

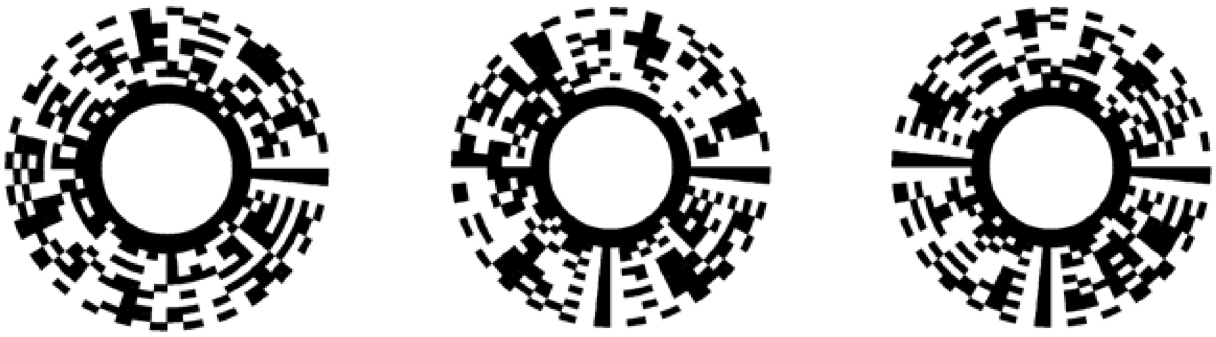
Figure 3: The three types of generic annular barcodes. Repeating start marker and radial zebra pattern increases robustness.

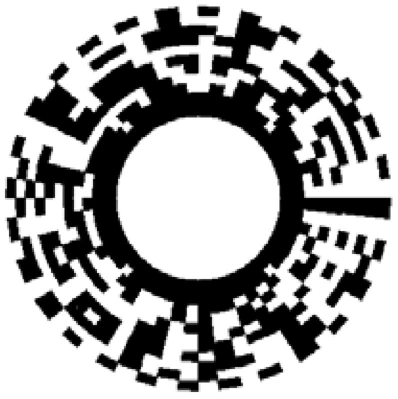
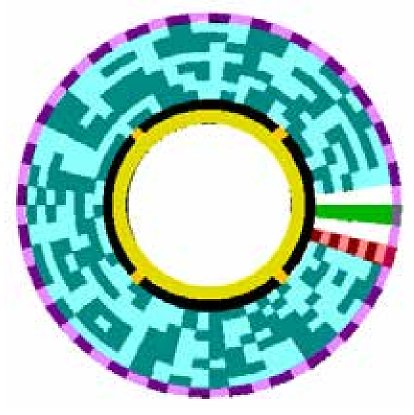

Figure 4: left) Annular barcode of type 1 with a data zone of 56 bytes; right) the same barcode but with highlighted special purpose zones (see also back inside cover for a color image)

Decoder

The barcode decoder is responsible for detecting, recognizing, and decoding a barcode in an image. Its input is an image of an annular barcode and its output is the data stored in the barcode. In case of a colored input image, it is first converted to a grayscale image and then binarized. For binarization the well-known method of Otsu usually results in a good binary image. In order to cope with noisy images, a Gaussian blur with a 5x5 kernel is applied prior to binarization.

The decoder cannot fully follow a simple process-chain of four stages („recognition", „perspective correction", „detecting timing information", „module evaluation"), because the angular zebra pattern (a part of the timing information) is also needed to achieve an accurate perspective correction. Therefore, the decoder steps „perspective correction" and „detecting timing information" are not strictly sequential. Furthermore, a two-step approach is used for the „perspective correction", resulting in the seven process stages depicted in Figure 5.

Each stage of the pipeline works on a best effort basis. If for example the recognition stage fails because it does not find the basic characteristics of our barcode design, the input image is discarded and the rest of the decoding-chain is not activated. However, if the recognition stage succeeds, then it returns the center of the barcode which is needed as input in the second stage. This is done by blob[2] analysis. General blob analysis allows filtering the blobs by area, circularity, inertia, convexity, color, and further criteria [Mal15]. Since the center of our barcode lies inside an empty white circle, we can search for white blobs and analyze them. To be a candidate for the barcode center, the white blob needs to have a circular shape or because of perspective distortion at least an elliptical shape. Circularity values between 0.8 and 1.0 are useful.

- The barcode's size lies between certain boundaries. If it is bigger than the field of view of the camera, it cannot be decoded because some information is not visible and therefore missing. On the other hand, it can be expected that the barcode has a minimum size which is a multiple of the smallest reasonable module size. Module sizes below 3 pixels are too small for robust decoding and hence ignored.

2 blob: binary large object

Detection of Square and Start Markers

In the second stage of our decoding-chain we try to detect the four square markers for rough perspective correction and the one to three start marker(s). In an unmodified barcode the start markers are black and the square markers are white. In a barcode without perspective distortion the four square markers lay in a perfect square.

Based on the observations that a start marker is the longest radially continuous stretch of black pixels, the algorithm uses a radial scanline revolving around the estimated center of the barcode (Figure 6). The scanline starts at 70% of the minor axis of the central ellipse (depicted by a green circle) and ends with the first pixel after the first black pixel. Along the radial scanline the length of the first continuous stretch of black pixels is evaluated. Figure 7 shows a possible plot of the first continuous black distance at different angles. The four smallest values signify the location of the four square markers, the maximum is where the start marker is located.

Because of the small discreet angular steps the scanline uses to revolve around the center, it is possible that two of the four lowest points belong to the same square marker. Errors are avoided by enforcing a minimum angular distance from one perspective marker to the next.

Sometimes the longest continuous black distance is not one of the start markers but an optical edge between black modules located at alternating positions on both sides of the edge. In order to avoid such errors, the found start markers are verified by checking that they are surrounded by continuous white pixels on both sides.

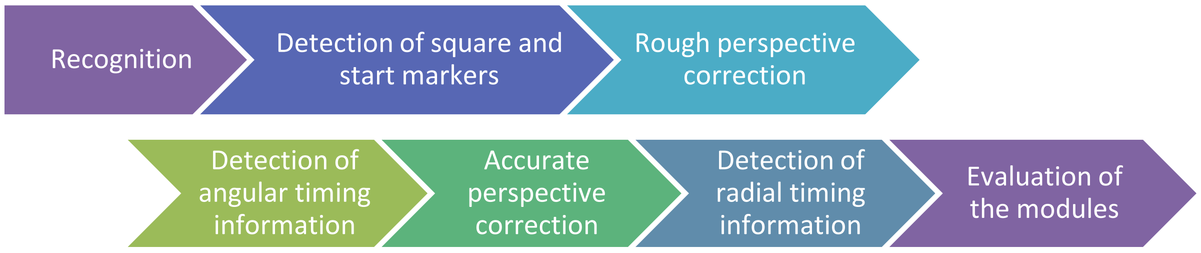
Recognition
Detection of square and start markers
Rough perspective correction
Detection of angular timing information
Accurate perspective correction
Detection of radial timing information
Evaluation of the modules

Figure 5: The seven stages of our decoding chain

The four square markers form a square in our barcode design. Due to the perspective distortion induced by the barcode scanner, the detected square markers in the input image may not form a square, but any quadrangle. Using a pseudo perspective transformation model for the distortion it is possible to compute a four-point linear transformation matrix T_1 between the positions of the detected square markers and their expected positions in a square. This transformation matrix T_1 is then applied on the entire input image. Further detection steps are performed on the transformed image.

The limited accuracy of this perspective correction is mainly due to the small size of the quadrangle formed by the square markers. Any inaccuracy in the location of the markers is amplified in outer areas of the barcode. Therefore, a more accurate perspective correction is applied during the next stage in the decoding-chain.

Detection of Angular Timing Information

The roughly undistorted image is the input of the fourth stage in our decoding-chain. In this stage the angular timing information has to be detected to determine the angle between two consecutive data modules. It also sets the foundation for a more accurate perspective correction which is necessary in the last stage. The angular timing information is stored in the zebra pattern around the outer edge of the barcode.

The angular zebra pattern is detected by a scanline revolving around the estimated center of the barcode (Figure 8). The scanline starts on a circle just outside the barcode and scans towards the center of the barcode trying to find the black modules. The distance to the occurrence of the first black pixel is evaluated. If this distance is within certain bounds around a value B, we have found a black module. An initial value of B is determined by the distance between the circle around the barcode and the outside of the start marker.

Although the input image has been roughly perspective corrected prior to this processing stage, the correction might not be perfect. B must therefore be updated at black modules in the angular zebra pattern to reflect the local situation. This is done by choosing B as the mean value of the already processed distances between the circle and the black module in the angular zebra pattern. Even with this adaption, detecting the angular zebra pattern occasionally fails. To prevent the decoder from progressing with wrong settings, the detected black modules are subject to verification.

Since the number of modules in the outer zebra pattern is dividable by eight, the number of black modules is dividable by four. Given an array with the position of every black module in the angular zebra pattern, selecting four of them lying in a square is easy. Such four positions are used to compute a second four-point linear transformation matrix T_2. This matrix T_2 performs a mapping between the roughly perspective corrected image and the accurately perspective corrected image. Since the aim is to avoid precision loss by doing two transformations on an image (first a rough correction T_1, than the accurate one T_2), the two perspective correction matrices T_1 and T_2 can be multiplied to allow transformation from the in-

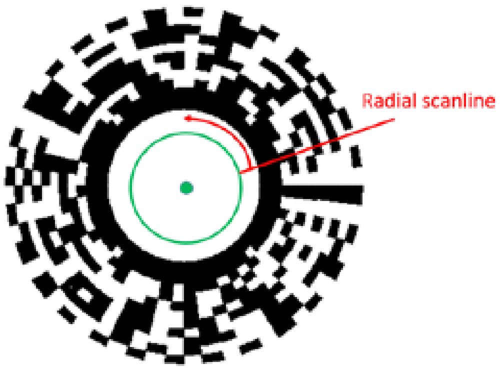

Figure 6: Radial scanline revolving around the estimated center of the barcode

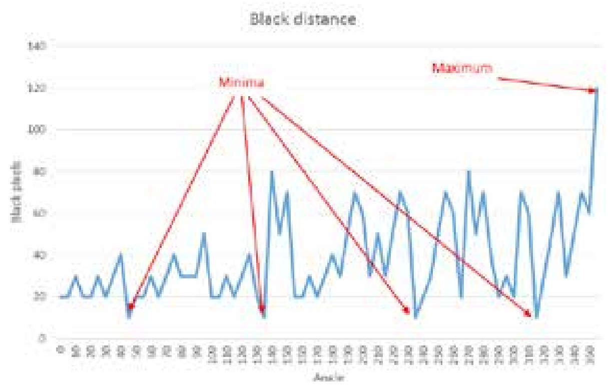

Figure 7: Plot of the continuous amount of black pixels at various angles

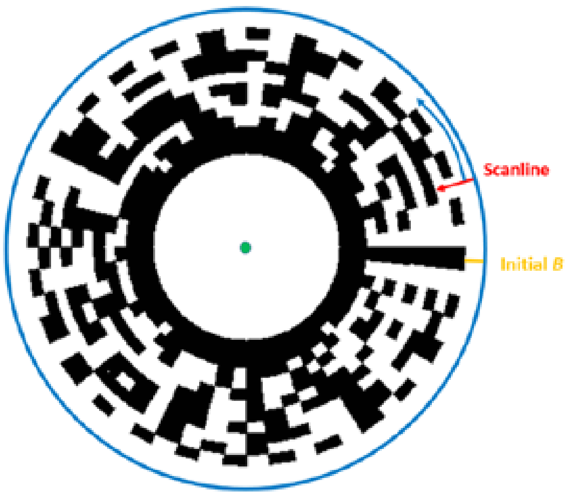

Figure 8: Scanline starting outside the barcode and revolving around its center

put image to the accurately perspective corrected image in one transformation step. This more accurately undistorted image is the input of the following last two stages.

Detection of Radial Timing Information

The radial zebra pattern defines how many rings a barcode has and how far apart they are spaced. It is always situated next to a start marker. Because the barcode has possibly been mirrored, it could lie either to the right or to the left of the start marker. Therefore, two radial lines, one on each side of a start marker, are projected from the barcode center through the potential locations of the vertical zebra pattern. To evaluate whether one of these lines goes through a radial zebra pattern, the alternating continuous stretches of black and white pixels are measured. Because all rings are equal in thickness, a perfect zebra pattern would have n continuous black pixels followed by n continuous white pixels followed by n continuous black pixels again. If no continuous stretch of equally colored pixels significantly deviates in length from the average, a radial zebra pattern has been found.

The radial zebra patterns also determine the angular reading direction of the barcode payload. If it is to the right of the start marker, the reading direction is clockwise.

Evaluation of the Modules

In the last stage of the decoder-chain the colors of all modules inside the data zone have to be evaluated and stored in the right order in a bit vector.

The information provided by the angular and the radial zebra patterns in combination with the barcode center allows us to virtually draw a grid over the barcode in the undistorted input image. Figure 9 depicts such a grid. At every intersection of the grid a module has to be evaluated as either black or white. To evaluate each module, a total of five pixels are sampled in a cross-shaped pattern. Each pixel votes for the module to be either black or white according to its color. Because the sampling is performed on a binarized image and an uneven number of pixels are sampled, a module's color is never ambiguous, so it can always be classified as either black or white.

Storage

The maximum amount of data our annular barcode design supports depends on the number of barcode modules, the datatype (input character set), and error correction level. All these parameters are equal to those in a QR codes. Therefore, the concepts and techniques used in QR codes are also applicable to our annular barcodes.

Tests

During the development phase, most tests were done using a laptop webcam as scanner to read a barcode shown on a mobile phone display. The scanning process is fast and reliable enough to cause no inconvenience to the user, even under non-ideal conditions such as a considerable amount of perspective distortion.

Because the results are dependent on the actions of the person holding the barcode, it is impossible to reliably quantify the success rate and overall quality with this test method. To measure the quality and capabilities of the decoder in a reproducible way, a batch testing tool has been developed. It generates a barcode, adds some defects to it and then checks if it can be correctly decoded. To obtain reproducible results, each test is run several times.

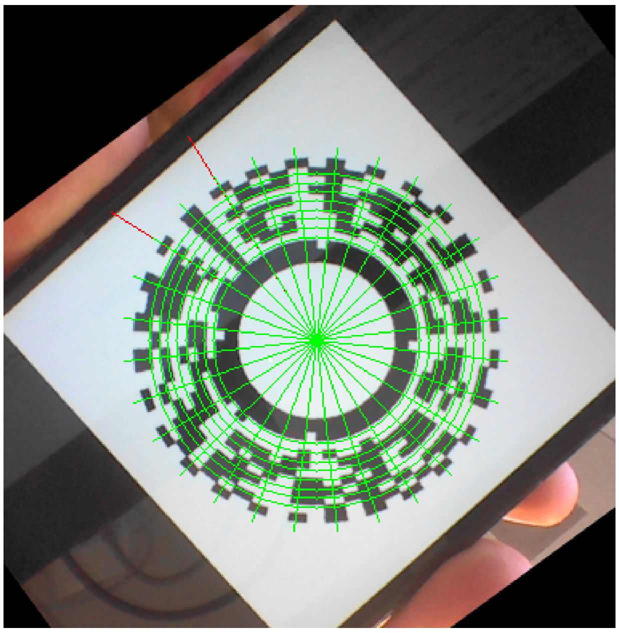

Figure 9: Grid matching the barcode modules. To keep the grid visualization readable a line was drawn only through black angular modules.

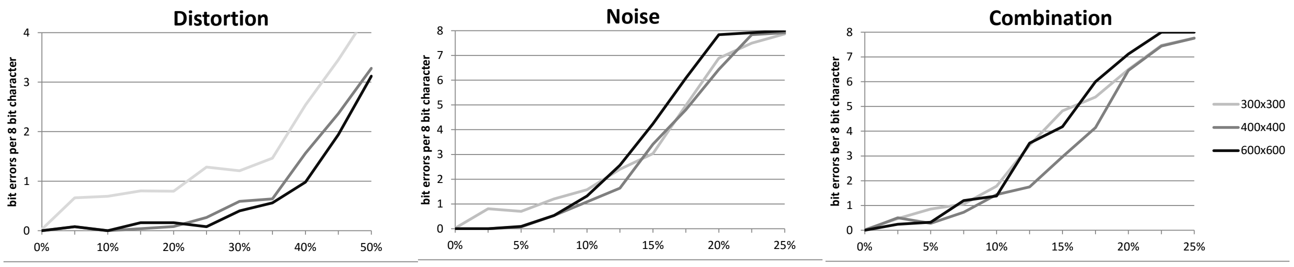

Figure 10: Test results: left) Distortion up to 35% only results in an average bit error larger than 1 in the smallest barcode; middle) Noise up to 10% results in an average bit error of less than 2; right) Distortion up to 10% and noise up to 10% together result in an average bit error of less than 2.

Out test tool is capable of adding two types of defects to a barcode:

- *Perspective distortion:* The perspective distortion can be controlled in percent. 0% means no distortion, with 100% it is possible that all four corners of the barcode image are mapped to a single line in the distorted image.
- *Salt and pepper noise:* This defect adds random black and white pixels to the image and is controlled by a percentage value. At 100%, the image is fully covered by noise.

All tests were conducted with three small barcodes of type 1 (see Table 1). The influence of the two different defect types has been tested both in individual test series and in a series of tests with both defects combined. Figure 10 shows the average bit errors depending of the amount of distortion and noise. These diagrams help to choose an appropriate error correction level. For example, if we use an error correction code which is able to correct two bit errors per eight bit character, then our annular barcodes can be distorted up to 35-40% (depending on barcode size) and contain noise up to 10%.

Conclusion and Further Work

We have presented a novel generic design for annular barcodes and we have shown that our decoder is able to decode very small barcodes (5 mm diameter at 300 dpi) of type 1 under real conditions with a small average bit error rate that can be corrected with an appropriate error correction code. On our test machine (i7-3770 quad core with HT @ 3.4 GHz, 12 GiByte RAM) the mean decoding times for the three barcode sizes are 13, 11, and 18 ms, respectively.

There is a chance that the data zone in a barcode forms a pattern similar to the start marker. If such a pattern is present, then it might be possible that the decoder selects the wrong one and hence the payload cannot be read. The chance of a pattern equal to the start marker occurring in a barcode is dependent on the number of rings and the number of modules per ring. Due to the zebra pattern around the outside of the barcode, a start marker lookalike can only occur at every second angular module. For a barcode that has eight rings, the chance of a lookalike is about 1:50000. However, if the number of rings is reduced this quickly becomes a big problem. A possible solution is to check the generated barcode for start marker lookalikes. If such a lookalike is detected, a random bitmask is selected and an XOR-Operation is performed on the data zone and the bitmask. To be able to decode the barcode again the used bitmask has to be encoded into the barcode data zone as additional header information.

So far we focused on very small angular barcodes and therefore on type 1 with just one start marker. Further work should be done for larger barcodes of type 2 and 3.

Size	Module Size	Min. Radius	Max. Radius
300x300	6	60	120
400x400	8	80	160
600x600	12	120	240

Table 1: Annular barcode parameters (pixels) used in our tests

References

[BC09] Barcoding Connected: Shotcode Barcode: The circular 2D barcode for mobile tagging: http://blog.barcoding.com/2009/02/shotcode-barcode-the-circular-2d-barcode-for-mobile-tagging/

[Brü15] Brügger, Robin. Ring-shaped Barcodes. Bachelor Thesis, Computer Science, University of Applied Sciences Northwestern Switzerland, FHNW, 2015.

[Mal15] Mallick, Satya. Blob Detection Using OpenCV (Python, C++). http://www.learnopencv.com/blob-detection-using-opencv-python-c/

[Mud06] Mudie, Stuart: Shotcode. Licensed under CC BY-SA 2.0 via Wikimedia Commons: https://commons.wikimedia.org/wiki/File:Shotcode.png

[TR] TechnoRiver. Barcode software and components: http://www.technoriversoft.com/CircularBarcode.html

[Wiki] Barcode: https://en.wikipedia.org/wiki/Barcode

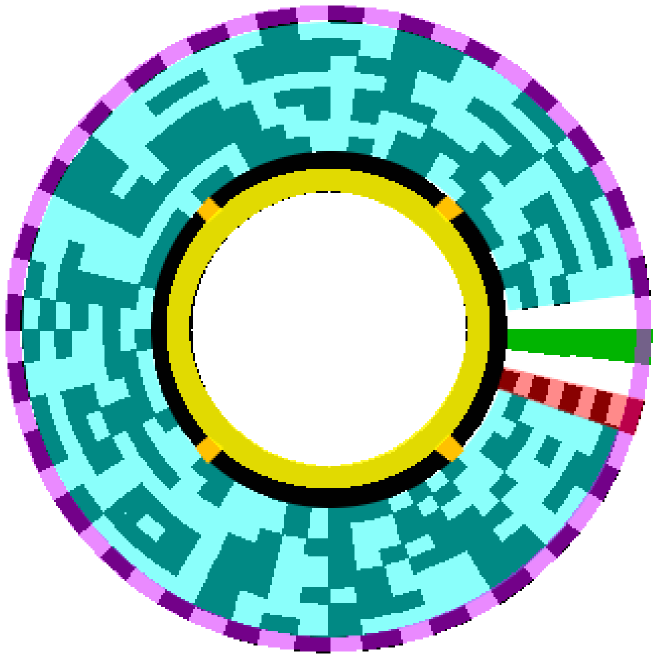

Figure 4: Annular barcode of type 1 with a data zone of 56 bytes with highlighted special purpose zones: a yellow ring for locating the barcode, orange square markers for rough perspective correction, a blue data zone containing all data modules, a green start marker defines the beginning of the data zone, a red angular zebra pattern determines the angles between the modules, and a purple radial zebra pattern determines the radial timing of the rings.