https://doi.org/10.2352/ISSN.2470-1173.2020.3.MOBMU-207

JAB Code - A Versatile Polychrome 2D Barcode

Waldemar Berchtold, Huajian Liu, Martin Steinebach, Dominik Klein, Tobias Senger, Nicolas Thenee
Fraunhofer Institute for Secure Information Technology (SIT); Darmstadt, Germany
German Federal Office for Information Security (BSI); Bonn, Germany

Abstract

This paper presents the design and implementation of an encoder and decoder of a colored barcode with high data density and storage capacity and freedom in shape. The approximately three times higher data density compared to conventional 2D matrix codes such as DataMatrix, QR or Aztec code is achieved by the use of eight colors and enables new applications, especially in the endconsumer market as well as in IT security. The challenges associated with the use of the color channel in printing with conventional office printers and recording by smartdevices under typical scenarios are addressed. The flexibility in the barcode shape is achieved by combining a primary and several secondary symbols according to a given scheme and give the necessary freedom for various applications. The presented code stores colors redundantly in a color palette as a reference in order to provide high robustness. JAB code, Just Another Barcode, has been specified, implemented, tested and is avaiable in github and www.jabcode.org under the license LGPL 2.1. JAB code is currently in the standardization process at the International Organization for Standardization ISO.

Introduction

Barcodes are optical representation of data. Often, barcodes are printed on objects and carry information about the object itself or information in its context. They are widely used to exchange data automaticly without errors in a simple and fast way in various processes in everyday life. A typical example is the Universal Product Code [1] developed by the GS1 organization. Originally, such barcodes carry information by laying out consecutive lines with different widths. The storage capacity of such barcodes is very limited and amounts to storing a number, i.e. a product code. Barcode technology significantly evolved throughout the 1990's. Better printing and scanning devices allowed to lay out binary information in grid structures. Binary information are encoded by black or white squares called modules. By making use of two dimensions the storage capacity of 2D barcodes [2, 3, 4] can be improved significantly compared to older 1D barcodes. They offer enough capacity to store complex product codes, URLs, on business cards to encode contact data, or in general short textual information.

Applications in which a higher data density than 2D barcodes provide is required, can be found especially in the end consumer market and in IT security. In the following, the example of ensuring the integrity of data without having to query data online will be examined in more detail.

Let us assume the scenario where a doctor's prescription is required. If a patient wants such drugs, he must present a valid prescription from his doctor at the pharmacy. The prescription is made on a special form with a doctor's stamp and its signature. However, this does not offer a high level of protection against forgery. If a patient wants to receive medication without a prescription in the future, he can, for example, reproduce the stamp and signature by scanning them from a valid prescription, preparing them and printing them on the following prescriptions. The paper can be obtained online in various ways and the prescription requirement can be circumvented.

In IT security, the established digital signature procedure is used to verify integrity and source authenticity. For this purpose, the issuing body, e.g a doctor, receives a digital certificate authorising it to issue prescriptions. With this digital certificate, the issuing body signs the content of the prescription certificate, such as the name, address, date, drug, validity and reason, and can store the signed content together with the digital certificate in a barcode that is printed on the presciption. The pharmacy can scan the barcode on the prescription and compare the signed content in the barcode with the printed one. If both match and the signature is valid, the pharmacy has successfully verified the authenticity of the prescription. This approach can also be applied to reports, certificates or documents [13]. The amount of data required for the described scenario is approximately 1363 bytes. This results in a QR code with a size of 133x133 modules and thus a total of 17,689 modules. JAB Code requires 85x85 modules and thus a total of 7,225 modules, with an approximate size of 7 by 7 cm where the size of the QR Code is approximatly 2.5 times bigger with comparable error correction performance.

In the past, some polychrome barcodes were developed with the aim of increasing data density [5, 6, 7, 8, 11, 9, 10]. Reasons why these have not become common place are various. By adding color channels one can achieve a higher data density, but at the same time challenges arise. Influencing factors are printers, environmental influences such as lighting and abrasion, and the characteristics of the reading units, which handle the colors in their own way and in some cases robustness suffers. Robustness of a barcode however is of course one of the most important aspects. In this paper we show how JAB code addresses all these influencing factors.

If one considers the requirements on barcodes for different applications, then apart from a higher data density also the flexibility in the shape is important. It is particularly shown by the fact that in practise both a rectangular DataMatrix code and QR code are required and to be brought to a standard by the ISO.

To ensure practical usability of a polychrome barcode, the current state of hardware has to be considered as well. In industry and logistics, hardware with monochrome reading units is mostly used, whereas polychrome reading units are available in the end consumer market through smartphones. The latter is also where most application scenarios for a polychrome barcode exist. In addition, the availability of a national or international standard plays a crucial role in its implementation.

Requirements and Proposed Barcode

In this chapter we first define the requirements for a barcode and then present the design of JAB code.

Requirements

Shape flexibility Shape flexibility enables custom designs for specific use-cases. For example, one could build a U-shape around a hologram or other physical security features to maximize using any available space on documents with fixed design specification.

Finder Pattern, Alignment Pattern and Clock Pattern The three finder, alignment and clock patterns are functional patterns used for locating, aligning and correcting the code. To find a code quickly and reliably, finder patterns on fixed positions within the code are required. The finder pattern should be very distinct in style in order to have a low false positive rate. At the same time, it should be as simple as possible so that it can be detected quickly using a computationlly simple algorithm. In addition, a finder pattern does not carry any data and thus should therefore be both as small as possible as well only as large as absolutely necessary. Roughly, the larger the pattern, the easier it is to identify the barcode. If the finder patterns are large in size however, the ratio between net capacity and space required for the finder pattern becomes quite uneconomical. For example for a 21x21 QR code, roughly one third of the whole code is used for the finder pattern. The clock pattern and alignment pattern are used to correct distortions in a captured code. Since the clock pattern requires a lot of modules and grows with the code size, QR codes for example use alignment patterns to correct distortions in larger codes. In comparison, DataMatrix and Aztec codes use the clock pattern repeatedly. This means that QR codes require fewer modules for code correction, which can be used instead for data encoding. Robustness however does not suffer and therefore the clock pattern can be replaced with alignment patterns.

Error Correction Since errors can occur when reading a barcode, e.g. caused by shadow, blurring, dirt or ambient lighting, an error correction scheme is always used for 2D barcodes. If color channels are used, the influencing factors mentioned play an increasingly important role. In general, one can distinguish between two types of errors: Burst errors occur when large sequences of data are erroneous, for example if some part of the barcode is missing. Random errors occur when single modules are flipped or destroyed. Such random errors usually occur more often if one considers a polychrome barcode compared to a monochrome one, since not just two, but four, eight or more colors need to be distinguished. Hence, the used error correction scheme should be able to handle random errors particularly well.

Color flexibility It is possible to use colors for monochrome barcodes, i.e. one can just use lighter and darker modules instead of black and white. This offers advantages if colored paper is used or there are specific lighting conditions during the decoding process. A polychrome barcode should also have this characteristic of selecting colors in order to be flexible in specific applications. After a barcode has been generated, it is either printed or shown on a display and read with a recording device. Typically, one chooses colors that have the maximum possible distance to each other in e.g. the RGB space in order to achieve the best robustness. This color selection is usually different for a printed code compared to a code shown on the display. Therefore, it is important that the colors can be selected according to the application scenario, so that they have the maximum possible distance from each other after printing, for example. Figure 8 shows a JAB code printed by a laser printer, where the color red and magenta are almost indistinguishable by the eye, where the digitally generated colors for magenta in RGB space are 255 0 255 and for red 255 0 0. Other office printers don't make this misprint, but the colors are also printed there differently compared to the theoretical color space. The reason why this happens ist the conversion of the colorspace by the printers driver. Since JAB code should also be printable by office printers color flexibility is a core requirement.

JAB Code Design

JAB code takes the experience of existing barcodes and combines the most established approaches.

Shape of the Modules: JAB code uses square modules. The decoding algorithm however is flexible enough to also detect a JAB code with circular modules.

Shape Flexibility In order to allow a certain flexibility in the shape of the code, JAB code implements two different symbol types. One type is the *p*rimary symbol, which is always the starting point. The second type is a *s*econdary symbol. A code can consist of one primary and several secondary symbols. The secondary symbols are docked to the primary or to other secondary symbols according to a predefined scheme in order to create different shapes. Both, the primary and the secondary symbols, can have a minimum size of 21x21 modules and a maximum of 145x145 modules, whereby the step size to the next side length is 4 modules, i.e. 25x25, 29x29, etc. In addition, both can also be rectangular and accept all combinations between 21x25 and 141x145. To dock a secondary to the primary or another secondary, both symbols have to have the same number of modules on the docked side.

Finder Pattern and Alignment Pattern The primary symbol uses four finder patterns, each consisting of two squares of 3x3 modules, overlapping in one module. The finder pattern are close to the corners. This has the advantage that distortions can be corrected easily and a quiet zone can be omitted. Each of the four finder patterns has its own color order and orientation. This allows not only to correct rotations, but also mirroring. Figure 1 shows an example of a primary symbol and figure 2 shows an example of the secondary symbol, with the positions, colors and orientations of the finder patterns. The color palette is stored redundantly around the finder patterns as well as the metadata with the corresponding error correction bits. Alignment patterns are used in the same way for both primary and secondary symbols to correct geometric distortions. These are placed according to a defined scheme only for larger codes. The alignment patterns consist of two squares of the size 2x2 modules, which overlap in one module. A secondary symbol uses inverse alignment patterns rather than finder patterns.

The design of the finder patterns makes it possible to meet all mentioned requirements. The detector binarizes the obtained image into one of the three color channels and smoothes it. The detector then searches for an alternating sequence. If one is found, the center point is determined and at this point it is checked horizontally and in both diagonals whether the same pattern also occurs in two of the three cases. If this test is successful, it is carried out equally in the two remaining color channels. Only if an alternating pattern in three of the four directions can be found in all three color channels it is a finder pattern. Figure 10 shows the procedure for detecting a finder pattern.

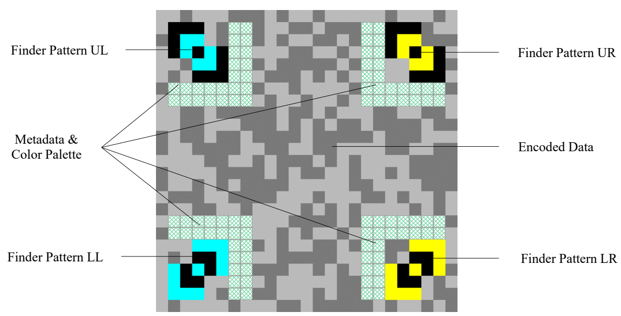

Figure 1. *Square primary symbol with its finder pattern, metadata and color pallet placement.*

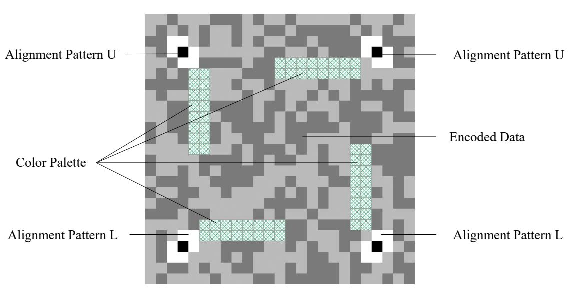

Figure 2. *Secondary symbol with its finder pattern, metadata and color pallet placement.*

Error Correction JAB code uses Low Density Parity Check LDPC [12] code as error correction method. LDPC codes have the advantage of being able to correct random errors well, while other correction methods such as Reed Solomon codes (RS codes) correct burst errors more effectively. In order to be able to correct burst errors with the LDPC Code as well with a high probability, which occur e.g. due to soiling, the data and error bits are randomized and thus evenly spread over the code. A disadvantage of the LDPC code is the higher complexity on the encoder and decoder side compared to RS codes. For current generation hardware, this does not pose a problem.

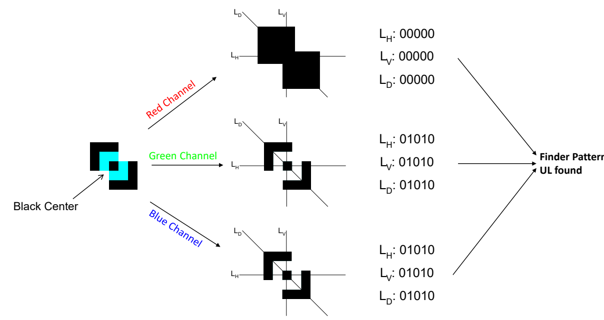

Figure 3. *Example of decoding procedure for the upper left finder pattern.*

Color flexibility In order to provide the flexibility in color selection required for some applications, JAB code uses a color palette that is stored redundantly in the code. During decoding, the code first reads the color information from the color palette and assigns all modules to the colors they are closest to. The color selection must be designed in such a way that the selected colors in the color space maintain the largest possible distance. Furthermore, the number of usable colors can be chosen as 4 and 8, but in the metadata space it is reserved that the code can also be used with 16, 32, 64, 128 and 256. Our experiment however show, that with current printer and reader hardware, using more than 8 colors is difficult in practice. More colors should only be used if the printing and capturing processes are optimized for a particular color set. The option to use more colors, and thus to further increase the data density however make JAB code future-proof, since we anticipate improvements in printer and camera technology within the next years. In the world of barcodes, life cycles of 30 years are not untypical (see 1D and 2D barcodes) and this is the reason to include such color options.

Code creation process During code generation, the encoder proceeds in a number of steps that are described in this part. The input message is compressed lossless to the shortest bit sequence using the encoding table. The message is then encoded using the ECC method. Based on the message and its the error corrected representation and the necessary modules for finder and alignment pattern results in a JAB code symbol size. The message is pseudo-randomly interleaved to take full advantage of the strengths of the LDPC code to ensure random errors in every distortion event. Subsequently, the message is represented by either 4 or 8 colors. For the example of 8 colors, three consecutive bits of the interleaved message are represented by one of the colors in the corners of the RGB color qube, e.g. the bits 0 0 0 are represented by black, 1 1 1 by white, 0 0 1 by blue and so on. The colors are written columnwise into the symbol, starting from the top left to the bottom. It is then checked that neither the finder nor alignment pattern appear erroneously in the symbol caused by the message. Such a collision is prevented by make use of mask patterns. The resulting symbol is a JAB code as shown in figure 4.

Code examples In Figure 4 there are some examples, where data ist encoded with 8 colors. A reader is invited to use www.Jabcode.org to experiment with JAB code.

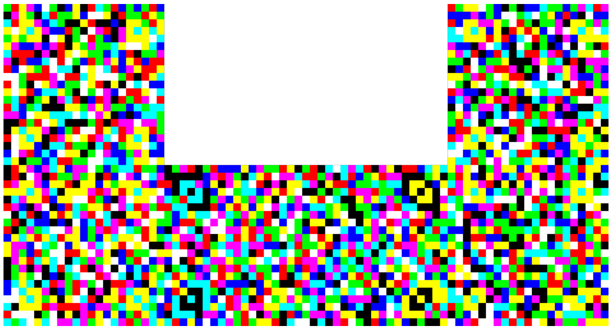

Figure 4. U-shape JAB Code. The example is encoded with the message: 'This is a JAB Code test created for you in 2018!' Note: the generated code can carry much more data, but this message has been chosen for the sake of simplicity.

JAB Code Detector and Decoder To find and decode a JAB code the following steps will be processed by the detector and decoder:

1. Preprocess the captured image and classify colours.
2. Locate the primary symbol within the image by finding the finder patterns.
3. Locate the alignment patterns, if exist.
4. Establish the sampling grid using the found finder patterns and the alignment patterns and sample the symbol.
5. Decode the metadata of primary symbol.and determine the code parameters, including side-versions, error correction parameters and mask pattern reference.
6. Extract and construct the colour palettes.
7. Decode the data modules by determining their colour index in the colour palette.
8. Release the data masking using the mask pattern corresponding to the decoded mask pattern reference.
9. Deinterleave the data stream.
10. Detect and correct errors in the data stream.
11. Decode the data stream into the original message in accordance with the encoding modes in use.
12. If exist, locate the secondary symbols docked to the primary symbol and decode the data stream in the secondary symbols.
13. If exist, locate and decode further docked secondary symbols recursively, according to the symbol decoding order.
14. Concatenate the decoded data out of all symbols.

Figure 5. Example of JAB code printed by an unaligned Inkjet printer.

In figure 5 the first steps of the detector are shown. The left subimage shows the captured image. The subimage next to it is created after the color correction in step 1. From this image the detector classifies the colors and represent the colors in each channel R, G and B. The finder pattern search is performed in each of the three color channel, where the approach is as shown in figure 10. If the finder patterns were found successfully, the sampling grid is created afterwards. Figure 6 show the JAB code and viualize the center of the finder patterns with red crosses and white dots in the center of each module.

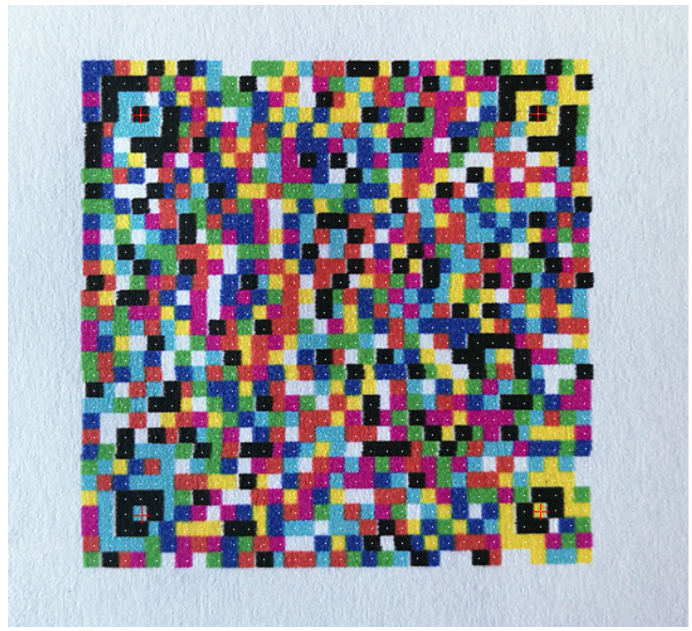

Figure 6. Detected JAB code. Red crosses indicate the found center position of the four finder pattern and the white dots indicate the center of each module.

Experimental Results

It is essential that JAB code meets all requirements of the intended application scenario. In particular, robustness is of high importance. This includes reliable decoding. Unfortunately, to our best knowledge, no standard test is available to evaluate robustness and effectiveness for colored barcodes. Many variables can affect the decoding performance. Of particular interest for our scenario are the following questions:

- How does printing on white paper vs printing on colored paper with an inkjet printer or laser printer affect the performance?
- How does capturing with a mobile phone vs. using a flatbed scanner affect the performance?
- How robust are JAB codes to wear and tear?
- How robust is JAB code in relation to different light temperatures?

In order to answer these questions, we designed the following test: JAB codes of various sizes are printed out on standard white paper and on colored and textured paper using both a Canon inkjet printer, and a Triumph-Adler laser printer. These are then captured using smartphones (Samsung Galaxy S4 and S8, Google Nexus 5, iPhone 7, Huawei Mate 9) and a dedicated scanner (Triumph-Adler), and decoded. Decoding for smartphones is conduced directly on the phone, whereas the scanned images of the

Paper	Inkjet	Laser	Phone	Scanner
White Paper	96%	95%	94%	100%
Colored Paper	90%	86%	84%	94%

JAB code performance on Inkjet- vs Laser-Printer and Mobile Phones vs Flatbet Scanner and white paper vs colored paper

Demage	Inkjet Printer	Laser Printer
folded/scrumbled	100%	100%
seals and stamps	100%	98%
shadow	100%	97%
coffee stains	95%	78%

JAB CoderRobustness after various damages which represent the abrasion

flatbed scanner are first transferred to a computer. We then measured the recognition rate. An image is counted as successfully recognized, if it is successfully decoded. For a smartphones, we fixed the distance from the camera to the JAB code to about 30 cm and captured the code in parallel to the paper and in a 45° angle. The light temperature was changed between 3000 and 6500 Kelvin in about 1000 K step size. Furthermore, the printed codes are folded and scrumbled, seals, stamps and coffe stains are added and the codes are captured with shadows on it.

Result analysis The impact of the printing process can be seen in table 1. JAB code recognition performance is more stable when printed by an inkjet compared to being printed by a laser printer. Scanning JAB codes with the used flatbed scanner gave better results than using a mobile phone. This is partly due to the fact that the images were sharper when they were taken with a scanner. Aside from the quality of the mobile phone's camera, the reason for the lower performance on the phones is also due to motion blur the phone when pressing the capture button. Therefore, recording from a video stream and extracting optimal image information out of it is gives much better results.

Printing on colored and textured paper shows potential for improvement. The analysis shows that in most cases, if the code could not be successfully decoded, the finder pattern could not be found or not completely found and therefore decoding failed. In the other cases the error correction failed due to too many errors. Figure 8 shows a fragment of a JAB Code printed with the laser printer, where red and magenta are hardly distinguishable and thus cause a too high error rate. In the clean digital code red and magenta are distinguishable (see figure 4). It is also a result of the printing profile used for printing. This demonstrates the need for colour flexibility and thus the need to adapt the choice of colour to the printer to raise the robustness. A perfect choise of the colors for a printer need to be done only once.

The impact of a printer on the result is also shown in a microscope picture in figure 9. Here it can be seen that the colors disperse into other module areas and thus interferences occur. This causes further errors during reading. If one is interested in printing JAB code as small as possible, this type of printing inaccuracy must be taken into account beside the unalignment of a printer as indicated by figure 10. This results in a limit for the minimum physical size of a module, so that the code is still readable. The size of the affected barcode area has a key influence when it comes to robustness against wear and tear. In particular in case of coffee spots, the size of damage was difficult to control in order to give reproducible results. The results of table 2 show that coffee stains in a laser printout have a greater influence the detection success. Smartphones behaved differently w.r.t. the impact of the light temperature, as shown in table 3. The iPhone 7 for example detected less than the Samsung or the Huawei smartphone and lowered the overall recognition performance. This is

Light Temperature	Mobile Phones
3000 Kelvin	94%
4000 Kelvin	92%
5000 Kelvin	96%
6500 Kelvin	96%
daylight	98%

JAB Code performance on different ligth temperature

Figure 7. *Comparison Huawei Mate 9 (top) and Apple Iphone 7 (bottom) of the different Light colour levels from cold to warm shown from left to rigth*

Figure 8. *A snippet of a captured JAB Code by the Samsung S8 and printed by the laser printer Triumph-Adler*

Figure 9. *A snippet of a captured JAB Code by a microsope*

Figure 10. *Example of JAB code printed by an unaligned Inkjet printer.*

due to the very different image processing steps performed by the manufacturers immediately after recording a picture. Figure 7 shows the photos taken under the various lighting temperatures that have been tested. One can see the captured images with the Huawei Mate 9 and the iPhone 7 in comparison. The images of the iPhone reflect the light temperature more precise. From this one can see that the decoder has slight difficulties with warm light temperatures and has to be optimized accordingly. Here, we are currently investigating on how to make the robustness of the decoder independent of the post-processing steps of the deifferent manufacturer.

Conclusion and Future Work

We presented JAB Code, a versatile polychrome 2D barcode, specifically targeted for the use case when higher data density is necessary. Experimental results show that JAB code provides robust codes with high recognition results, especially if printed on a clear background. Future work includes improving the recognition performance if printed on a colored and textured (different colors and patterns), e.g. on security paper and how to overcome the vendor-specific camera post-processing operations on mobile phones. Last but not least we plan to improve detection on mobiel phones by capturing from a video stream to eliminate motion blur the phone while pressing the capture button. JAB code has not been evaluated in comparison to other color codes, as there is either only a commercial and patented solution or no free implementation or specification available. The specification is openly available as BSI Technical Guideline BSI TR -03137 Part 2, and the source code of a reference implementation is freely available on Github under LGPL v2.1.

References

[1] GS1, Global Standards 1, https://www.gs1.org/standards/barcodes/ean-upc

[2] ISO/IEC 16022:2006 Information technology – Automatic identification and data capture techniques – Data Matrix bar code symbology specification.

[3] ISO/IEC 24778:2008 Information technology – Automatic identification and data capture techniques – Aztec Code bar code symbology specification.

[4] ISO/IEC 18004:2015 Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification.

[5] A. Grillo and A. Lentini and M. Querini and G. F. Italiano, "High Capacity Colored Two Dimensional codes,"Proceedings of the International Multiconference on Computer Science and Information Technology,2010.

[6] US7210631B2, Erez Sali, Doniel M. Lax, "Color bar code system," 2007.

[7] D. Parikh, G. Jancke, "Localization and Segmentation of A 2D High Capacity Color Barcode," IEEE Workshop on Applications of Computer Vision,2008.

[8] O. Bulan, V. Monga, G. Sharma, "High capacity color barcodes using dot orientation and color separability,' 'Proc.SPIE, 2009.

[9] https://www.aimglobal.org/store/ListProducts.aspx?catid=&ftr=Ultracode

[10] https://speechcode.eu/#!/product/specification

[11] Gavin Jancke, https://www.microsoft.com/en-us/research/project/high-capacity-color-barcodes-hccb/, 2003.

[12] Todd K. Moon, "Error Correction Coding: Mathematical Methods and Algorithms," Wiley-Interscience., New York, NY, USA, 2005.

[13] Christian Winter, Waldemar Berchtold, Jan Niklas Hollenbeck, Securing physical documents with digital signatures, IEEE SigPort, 2019 http://sigport.org/4832