The influence of printing technologies on QR-code recognition

B.Eng. Tahir Terzi, Prof. Dr. Karl Schaschek

Druck- und Medientechnologie
Hochschule der Medien
Nobelstr. 10
70569 Stuttgart
schaschek@hdm-stutgart.de

Abstract: More or less all industries need a way to label their products to enable identification for internal or external purposes. Automatic reading of these labels facilitates the process and increases its security. The introduction of 1D-codes (bar codes), which could be read by special devices (scanner), provided means to access multi digit numbers. Induced by the boost of performance of computers and cameras 2D-codes (Data-Matrix-, Quick-Response-(QR)-codes, etc.) were developed and introduced. Nowadays CPU performance of smart phones suffices to use these devices as 2D-code readers. Beside the initial use of labeling goods while adding more information, now these 2D-codes are used to gain access to internet sites. This link feature opens new possibilities of customer dealer interaction. The QR-code was developed by DENSO WAVE INCORPORATED and released in 1994. DENSO WAVE retains the patent rights to the QR Code, but declared that it would not exercise them. The details of the structure of QR-codes is outlaid in ISO 18004 (QR code 2005 bar code symbology specification) also explaining the de- and encoding and error correction algorithms used. The QR-code is the most frequently used in public life. It can be found on products like packed goods, brochures, magazines, billboards, large area prints and newspapers. For this reason the current study deals with QR-codes on printed matter analyzing the influence on recognition due to different printing technologies (electro photography, flexography, ink jet, offset and rotogravure). All Samples were printed on paper or foil. The later one being a white on with low opacity. In practice QR-codes will be scanned by special scanners or by a smart phone using the build in camera system. There are a lot of apps available that perform this task. A special feature of the QR-code technology makes it more or less impossible to use these apps directly to quantitatively analyze the prints: An error detection and correction algorithm is used. Thus errors to a certain limit are suppressed, even those imposed by printing. Therefore a way had to be found to perform this task independent of the regular used devices. A flat bed scanner with a physical resolution of 2400 dpi was used to acquire the raw data. This procedure eliminates possible errors of cameras taken images like improper positioning and cushion distortion. The pictures were analyzed by an ImageJ plugin developed for this purpose. It was incorporated according to the standard scheme for developing plugins for ImageJ using the Eclipse Java IDE. The basic idea of this procedure is to use the *"Analyse particles"* to identify the tag and its modules. Special care was taken to reduce artifacts being introduced by intermediate process steps. After proper alignment and identification of the outer skirt, the tag is cropped. The tag is identified by the three finder pattern. The alignment pattern were not used, but could be implemented easily. After identification of the finder pattern firstly the QR-code is rotated to match the inherent axes and secondly the linear number of modules is calculated. Based upon this the image is scaled to a nxn matrix, where n is the number of modules per edge. A histogram of the gray scales of this images shows the distribution of black and white modules. The distance of peak positions in comparison to the variance are a means to characterize the chance of recognition. To reduce the influence of arbitrary deviations multiple QR-codes of the same kind (printing technology, resolution, paper) were scanned and analyzed. The final check calculates the number of misinterpreted modules by comparing the beforehand known pattern with the analyzed one.

Finally a recognition test was undertaken using two smart phone models varying the distance of image capture and the size of the tag. As expected results not only vary with the model used, but also depend on the app being used. Even though these test were done briefly, it was obvious that some apps are able to correct slightly deformed tags or cushion and or trapezoidal distortion. This may be an indication, that QR-code Level 2 corrections, i.e. using the alignment pattern, are applied. Furthermore the influence of the quiet zone was studied. The finding is, that some apps savely detect QR-codes even without any quiet zone and even with a surrounding of high contrast.

Regarding resolution the classical printing technologies pose no obstacles. Thus tags can be outlaid comparatively compact, but the tiny size restricts recognition by smart phones. The linear size of an individual module should be mapped to several pixel of the camera. Depth of field restricts getting closer to the object. So called digital Printing still has some more difficulties to achieve high resolution and evenness of print. Especially ink jet samples showed an artifact called zigzag contours. Therefore the recognition level is reduced as pixel are mistakenly assigned to modules. Furthermore using light colors yield in modules that are not evenly filled, but covered with dots. These samples could not be analyzed by the developed software.

The analysis of printed QR-codes was based on a newly developed tool. A correlation of the recognition level to the print method was used to show the inherent differences. Nonetheless a future field study seems to be meaningful to firstly give an overview of capabilities smart phones possess (wide angle lenses yield cushion distortion, auto focus capabilities may be slow or inefficient, automatic white balance can result in a weak contrast in the target area) and secondly to the placement of QR-codes on non-flat surfaces. The later yields

in a distorted projection (trapezoidal distortion) of the image. Only a few of the briefly inspected apps could handle this to a limited extend. Future development of the QR-tool should include corrections for printed matter with curved surface. This would also help to correct the shape distortions mentioned above. Finally an app could be developed that helps judging the feasibility of new print designs and their application in production runs. In this regard another road map could be chosen and available packages might be used, for example the ZXing (”Zebra Crossing“) project. Thus the development would then focus on image enhancement and methods to correlate print quality and QR-code recognition.

1 1D- and 2D-codes

During the past decades an increasing need of machine readable signs was observed. Historically 1D-codes or linear bar codes started in the 1970’s. The most well known representative being the code 25 or EAN 8 / 13. Over the years different bar codes have been designed taking into account a rising need for larger capacity of information. As optical data were gathered with rotating scanners a limit to information density was given. Finally a step into another dimension was necessary.

Abbildung 1: Ëxamples of ”Code 25“ and ”EAN 8“”

Whereas 1D-codes are usually read by scanning technology in professional application like stores or storages today even smart phones are capable of reading this codes by means of interpreting camera pictures.

The introduction of 2D-codes was accelerated by the provision of even more storage capacity, higher CPU-Power and camera quality used by smart phones. Today it seems naturally that the technical requirements can be fulfilled even by a 50 Euro smart phone. Today a variety of 2D-codes is available like ”Aztec Code“ (ISO/IEC 24778), ”Data Matrix“ (ISO/IEC 16022), ”Maxi code“ and ”QR-Code“ (ISO/IEC 18004, [18015]) to mention the most commonly used.

Abbildung 2: Examples of ”Aztec Code“, ”Data Matrix“, ”Maxi Code“ and ”QR-Code“

This paper will focus on QR-Codes as they are wide spread used in commercial and industrial applications. Another advantage is the commonly available definition of it’s design and background ideas. They will be explained in the next section.

2 Properties of Quick Response Code

As mentioned earlier the Quick Response Code (QR-Code) was developed in 1994 by DENSO CORPORATION (now DENSO WAVE). Being proprietary at the beginning, and protected by several patents, today all details of QR-codes are public accessible through international standards by ISO/IEC 18004:2015 [18015]. This norm explains in detail how to generate a QR-code by its constituents. Furthermore the encoding and decoding is described. Therefor several open source projects exist which provide tools to encode or decode information to or from QR-codes. One of the strong features of QR-code is the capability to recover from partially misinterpreted or destroyed image surfaces. This is directly linked to information density, meaning that a higher degree of recoverable information is related to a lower information density.

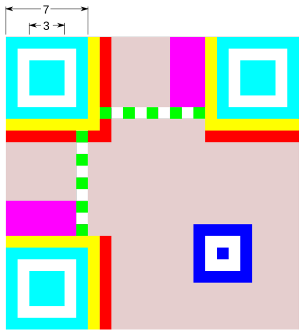

Abbildung 3: Explanation of the QR-code setup, shown for Version 2 patch of 25x25 module size

As pointed out in figure 3 the QR-code consists of colored and uncolored square areas, which are called modules. When bearing information a module carries one bit and is interpreted as being ”0“ or ”1“. This example consists of 25 times 25 modules. Common to all versions are the three large finder patterns (Marked in cyan, they consist of an inner square of 3 times 3 modules and a surrounding square frame of one module thickness). They bear no information beside the fact that they help to adjust (rotate, scale) and identify the patches. Starting from version 2 additional alignment patterns, indicated in blue, are implemented. The first mentioned patterns are surrounded by a void area, called separator, depicted in yellow. The finder patters are connected by timing patterns, drawn in column 6 and row 6 respectively (upper left corner being the origin), which help to determine module size. They are shown in green. Furthermore version (magenta) and format (red) information is placed in the patch. Finally for this particular patch 359 out of 625 data modules are left for use of storage. Depending on the error correction (L, M, Q, H) mode chosen the data content may shrink even further, gaining a higher recover rate for erroneous detected modules (7%, 15%, 20%, 25%). To achieve a proper recognition the patch itself should be surrounded by a quiet zone four modules wide, consisting of void modules. As this work did not focus on encoding or decoding issues further details are not discussed. As pointed out in Chapter 5 only the finder patterns were used to perform the analysis.

3 Printing Technologies

As 2D-codes become more wide spread and the range of available printing technologies is increasing one has to take the influence of these technologies into consideration when discussing readability. Most important contrast, smoothness of colored areas, sharpness of contours and lateral resolution are important for high recognition rates of QR-codes. The norm ISO/IEC 18004:2015 [18015] asks for white background to improve the contrast. If QR-codes are used in promotional environment, sometimes a light coverage of the background is asked for. One has to keep in mind, that for all of the printing technologies mentioned below a so called rastering or rendering process is necessary to get all the printable shades. Furthermore different raster technologies are used for various printing processes. Particularly in Offset printing a wide range of raster technologies is used. The fact holds for solid areas to. Especially solid tones impose a challenge for the detection Algorithm of QR-codes, as depending on the desired tonal vale no connected area for the particular solid area may exist (see also section (5)). In this study various printing technologies were used to get an overview. It has to be mentioned that apart from these shown examples results may differ as machinery may impose some additional influence. For the ease of analysing the prints were black and white only. Some patches used lower tonal vales. The trials were carried out using local available equipment.

A Xerox Phaser 5500DN electrographic printer with a physical resolution of 1200x1200 dpi was used.

Flexographic samples were printed using a Fischer & Krecke Flexpress 6S and plates imaged by an Esko CDI Spark 4835 plate setter and exposed by an DuPont Cyrel FAST 1000 ECLF. In flexography small single areas may impose print defects. Slurring may be visible thus increasing the printed area. Another difficulty is that edges are lacking ink thus reducing the contrast in this areas.

An Epson Stylus Pro 9600 Ink jet with a physical resolution of 2880 dpi was used.

The Offset samples were printed on an Heidelberg CD 74 using AGFA Azura plates. The later ones were imaged by a Heidelberg SupraSetter with a nominal resolution of 2540 dpi. A Heidelberg ImageControl system was used to comply to ISO 12647-2 [12413] regarding the print quality. The paper was of category $1/2$, which means white matt or glossy paper. In this case the coloration of black has to be set to $L^*a^*b^* = 16/0/0$.

The Rotomec MW 60 rotogravure press used cylinders imaged by a HelioKlischograph K500. In rotogravure real tonal value are possible due to the variation of cell size. On the other hand solid areas may impose a problem as an solid area consists of closely arranged deep cells.

4 Preparation of Samples

In order to get comparable results on all graphic output devices the QR-code had to generated in a device independent way. For this purpose the encapsulated PostScript format (eps) is an appropriate means. The test files were created (Adobe InDesign) in such a way that module of size one to four pixels at the resolution of the output device were generated. Also a physical module size from 0.2 to 0.5 mm was used. Also a series with a quiet zone of one to four modules was added. Finally four pattern with different coverage ranging from 20% to 80% are placed at the bottom of the sheet. Finally a total of 16 patterns was compiled. To prepare for an easy data acquisition the size of the test area had to match DIN A4. Depending on the printing machine substrates had to be cut to the final size. Thus the Perfection V330 Photo scanner could be used. The resolution being 2400 dpi.

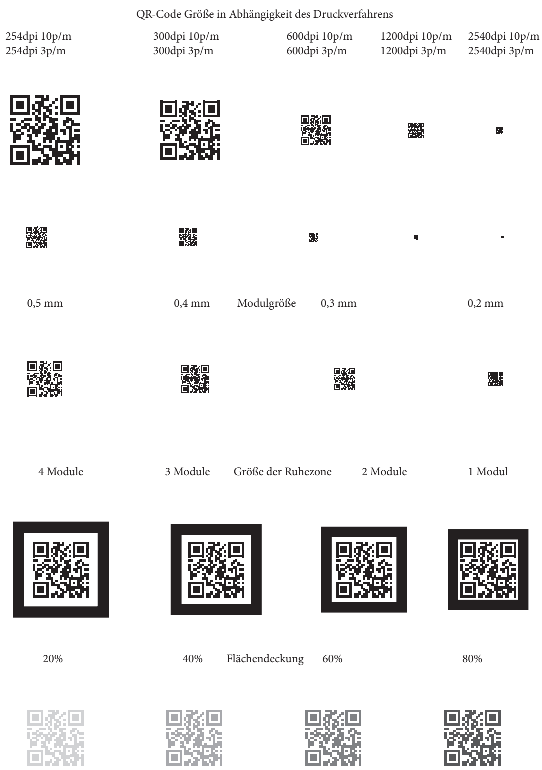

Abbildung 4: Test form for the QR-code print tests

Due to some inconsistencies in data handling and / or rendering setting the patches in row one and two didn't have proper settings. This was detected at a late time, which restricts the results presented.

For each of the printing processes individual substrates had to be chosen to ensure runnability. Whereas the Offset press and the Electrophotographic printer worked on sheet material the other were running on web

Process	Substrate	Specific weight	Format
Offset	BD matt	$135g/m^2$	$500 \times 700mm$
Rotogravure	UPM ultra matt	$57g/m^2$	$600mm$
Flexography	LD-PE (white)	$55\mu m$	$410mm$
Ink Jet	proof paper BD	$220g/m^2$	$432mm$
Electrophotography	Naturpapier Offset	$80g/m^2$	$210 \times 297mm$

Tabelle 1: List of substrates used for each printing process

material. An overview is given in table 1.

For ease of interpretation the contents of the QR-code studied was simply ”abcdefghijklmnopqr“.

5 Analysis

The Analysis of the scanned images was performed by using ImageJ. The images were scanned in grey scale mode with a resolution of 2400 dpi. ImageJ was developed by the NIH and can be used for research and educational purposes. Beside many already incorporated functions and filters it is comparatively easy to use a macro language to write scripts. One powerful process is the tool called ”Analyze particles“. It works on an binary bitmap and identifies connected areas of similar 'color'. A build in filter can be used to omit the areas that are to small or big. Finally this tool generates a list of all found areas fitting the filter. Several parameter may be stored. In this case the area and centroid parameters were used. An area within another one (same centroid) can be identified as an potential candidate for the finder pattern of the QR-code. If taking into account the relative areas of the pattern these can be isolated easily and assigned properly. To ease the procedure the studied patches were isolated manually. A more complex routine could even detect multiple QR-code in a single image. This is beyond the scope of this work.

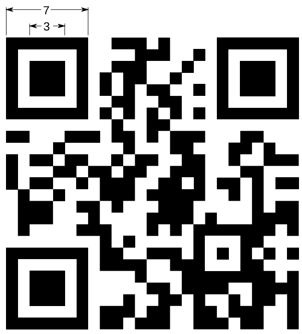

Abbildung 5: Example QR-Code showing the ”Finder pattern“ and the length of vertices

5.1 Rotation of Scanned Data

Even though the samples were printed, cut and scanned with great care a minimum tilt of the received images was noticed. To reduce this influence on the analysis the images had to be rotated by the software. To make sure, that the influence of a rotation is minimal the two methods implemented in ImageJ were tested. An image can be rotated by interpolating the weight and position using bilinear or bicubic interpolation.

An exemplary result is shown in images (6) and (7). It is obvious that the bicubic interpolation has a larger influence on dispersion compared with the bilinear one. Throughout the following study all necessary rotations were performed using bilinear interpolation.

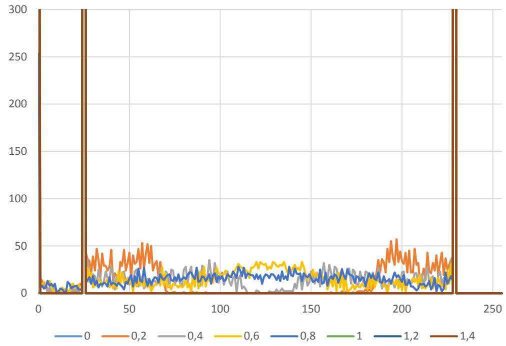

Abbildung 6: Influence of the Rotation with biliniear interpolation on occurrence density. The initial image was rotated by 0.2° to 1.4° in steps of 0.2°.

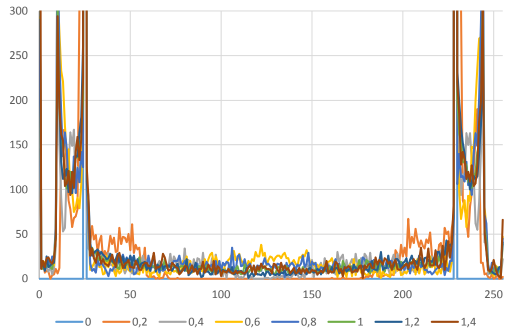

Abbildung 7: Influence of the Rotation with bicubic interpolation on occurrence density. The initial image was rotated by 0.2° to 1.4° in steps of 0.2°.

5.2 Identifying QR-code

As described in section 2 there are several features that can be used to easily identify a QR-code and to do the necessary rotation and sizing to assign the bit values properly. The following lines describe the procedure used.

First the image is converted to a binary image. Here contiguous areas are determined. Within this step the area and centroid are calculated by means of "Analyze particle".

In the second step areas with a similar centroid are searched. This step is successful if three pairs are found, that have a certain ratio of their areas:

$$R = \frac{7^2 - 5^2}{3^2} = \frac{A_1}{A_2} \tag{1}$$

The ratio defined is the same for all QR-codes.

If the second step was successful the original image is rotated by an angle determined from the position of the three finder patterns. Bilinear interpolation is used.

The fourth step is similar to the second in so far as again the finder patterns are searched. Now the result is used to crop the image to the outskirt of the pattern.

The final step is composed of the sizing of the image to a 25x25 grey scale image. Finally a one bit depth bitmap is generated by thresholding.

Abbildung 8: QR-code with module size 0.5 mm printed in Flexo, original image

Abbildung 9: Module size 0.5 mm printed in Flexo; rotated

Abbildung 10: Modul size 0.5 mm printed in Flexo

Abbildung 11: Module size 0.5 mm printed in Flex; rotated, croped and resized

In figures 8 to 11 the sequence of images generated is displayed. Obviously the size of the bitmaps shrinks in the process from the original 1325x1333 pixels to finally 25x25 pixels. To see how these operation work on the data of this sequence histograms for these images were generated. Figure 12 gives an example for the above mentioned patch. Rotating the image only marginally changes the frequency distribution because the observed angle were in the range of $\pm 2°$. The difference is below the 2% range and not noticeable in the diagram. Cropping the image reduces the white area surrounding the patch, which can be seen in a reduced frequency in the white region. This operation has very limited influence on the center of the frequency distributions.

Regarding contrast it is easy to distinguish between ”black“ and ”white“. Assuming a gaussian distribution for both peaks their mean value and standard deviation can be calculated using a nonlinear fitting routine. To implement a robust solution background has to be modeled. FOur the purpose of this work background was omitted. For this particular example $\mu_{black} = 43.7$, $\sigma_{black} = 7.7$ and $\mu_{white} = 230.5$, $\sigma_{white} = 5.2$ were achieved. The distance between the distributions being $\Delta\mu = \mu_{white} - \mu_{black} = 196.8$. Comparing this value to twice the standard deviations of both distribution yields: $2 \cdot (\sigma_{black} + \sigma_{white}) = 25.8$. Thus both distributions are clearly distinguished. After resizing the frequency distribution have visibly changed. The mean values have shifted towards the center and standard deviations have become larger. Now it is found that $\mu_{black} = 75.3$, $\sigma_{black} = 16.3$ and $\mu_{white} = 199.5$, $\sigma_{white} = 17.7$. Nonetheless it is found that $\Delta\mu = \mu_{white} - \mu_{black} = 124.2$ which has to be compared to $2 \cdot (\sigma_{black} + \sigma_{white}) = 68$ leading to the assumption that this distribution can still be easily distinguished.

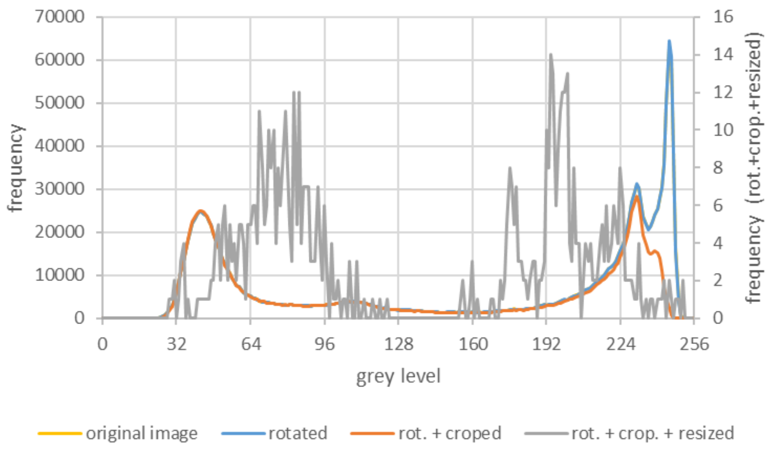

Abbildung 12: Comparison of frequency distribution of the original image, the rotated, the rotated and cropped, and the rotated, cropped and resize one. The latter one is characterized by the right axis

The result of fitting the two frequency distribution with nonlinear fit function $g(x)$ is given in figure (13).

$$g(x) = \frac{A}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2}$$

Another way to look at the distribution is to check the integral of the lower and upper half. In this case 318 modules are ”black“ and 307 ”white“ totaling up to 625, which is 25x25. Obviously this approach is only valid if both centers are equally distributed from the middle of the total range.

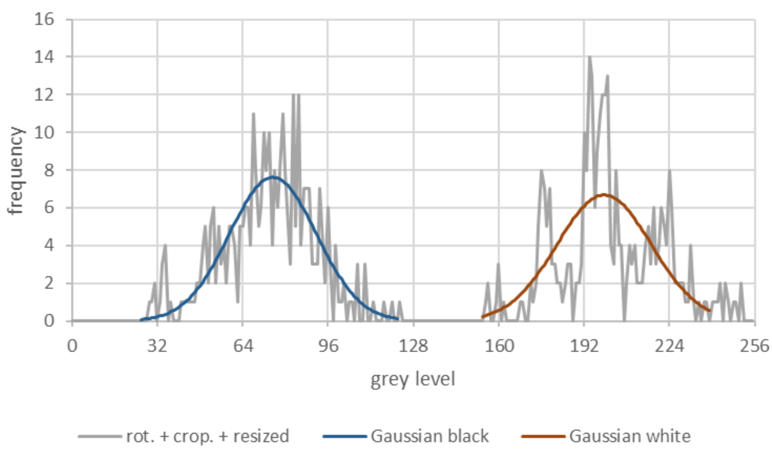

Abbildung 13: Gaussian fit of the two frequency distributions after rotating, cropping and resizing

According to the definition of QR-codes [18015] the intention is to arrange an even distribution between both modules. On the other hand a deviation from this evenness does not deliver a clear hint on an increased failure rate to detect the modules correctly. For this purpose the final 1-Bit image was compared to the original data file. By an XOR of both images deviations are easily displayed. For the above mentioned patch the derived images are shown in figures (14) to (17). The number of deviant modules is 16 in this case and can easily be calculated with one of standard functions.

Abbildung 14: Electrophotographic patch of 0.3 mm module size after rotating, cropping and resizing

Abbildung 15: Figure (14) after thresholding and converting to 1-bit image

Abbildung 16: Generated Patch

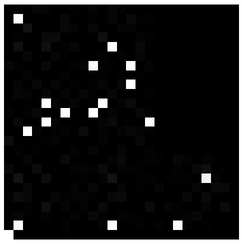

Abbildung 17: Calculated image after XOR-ing figure (15) and (16)

The analysis of the print trials is based on the last described procedure. All printed samples were analyzed this way. Here only the analysis referring to the module size will be presented. Electrophotographic samples were printed with two different resolutions of 600 dpi and 1200 dpi as shown in figure (18). All modules could be recognized by the algorithm. This process was the one showing highest error rates.

The ink jet samples were printed with a resolution of 360 dpi, 720 dpi, 1440 dpi and 2880 dpi as shown in figure (19). In all cases the $0.2\ mm$ patches could not be interpreted. Bleeding of the ink may be attributed to this observation, whereas also misplaced droplets are causing errors.

The flexographic samples were print in a single variety (see figure 20) as the patches were added to another print trial. All sizes could be recognized. Only the $0.2\ mm$ patch showed a low error rate.

The rotogravure sample wer printed with three different raster methods the "Xtreme Raster 230 R6 Xt+", a raster of 68 lines per cm and a graver angle of 130° and a raster of 80 lines per cm and a graver angle of 130°. Here small error rate were observed analyzing the $0.2\ mm$ and $0.3\ mm$ patches.

It seems that the offset results are missing, but with this process all samples were analyzed with a zero error rate.

Furthermore samples with reduced tonal values were intended to analyze. for these samples the algorithm failed to work. The reason found so far is that for all processes lower tonal values are achieved by using raster methods. This leads to non contiguous modules. The used ImageJ tool "Analyze particles" however then breaks down the raster in individual small areas.

To compare these results with the ones achieved with QR-code interpreting apps one has to bare in mind, that then a different approach is used. In addition smart phone cameras do not take images like a scanner does. Images will be influenced by uneven illumination, distortion of the image by the wide view lens used (barrel distortion), the often non-perpendicular view, the field of curvature and possible defocusing. Nonetheless these and other distortions increase the error rate. In this sense studying scanned images helps to focus on errors caused by printing processes.

The printed samples were also analyzed by a "Sony XperiaZ1 Compact" and an "Archos Neon"running the "Barcode Scanner" app, that is available for free at "Google play store". To reduce the effects of image distortion only a smaller area in the center may be used to identify the QR-code. The trials showed that QR-codes could properly read by this tool even when the ImageJ macro fails. This is especially the case if for lower tonal values the modules have been rasterized.

6 Conclusion

This study tried to use ImageJ and it's tools to analyze QR-Codes printed with different processes. From the result it is obvious, that the classical processes are leading in regard of low error rates.

Offset samples showed no error rate, where as the flexo ones show some at small module size. Rotogravure

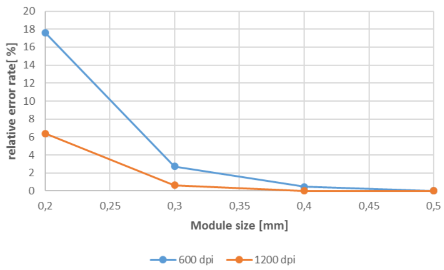

Abbildung 18: Relative error rates for electrophotographic printed QR-codes with resolution of 600 dpi and 1200 dpi

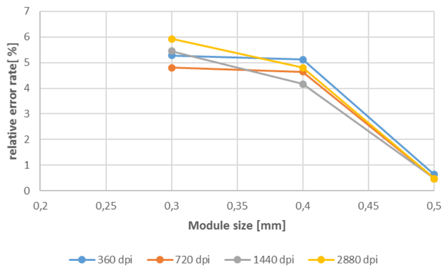

Abbildung 19: Relative error rates for ink jet printed QR-codes with resolution of 360 dpi, 720 dpi, 1440 dpi and 2880 dpi

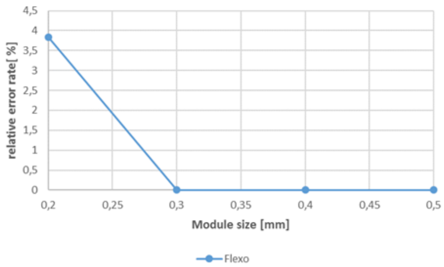

Abbildung 20: Relative error rates for flexographic printed QR-codes

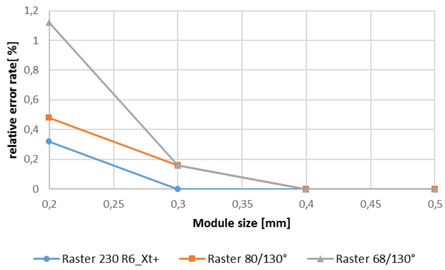

Abbildung 21: Relative error rates for rotogravure printed QR-codes

samples were similar to the flexographic one once the resolution was high enough.

Comparing ink jet and electrophotographic printing the latter process showed a steep dependency on resolution. Only at at high resolution an lager module size it was superior to ink jet. On the other hand ink jet error rates seemed to be more or less independent of the resolution set.

To analyze the Software packages used in QR decoding apps was beyond the scope of this work. But it is impressing how a variety of errors can eliminated. This is important in so far as a most public available QR encoders use only error correction mode L, which gives to to 7% error correction.

To gather even more information the tool has to be developed even further to automate it completely. Furthermore the analysis of rasterized modules could be solved with some of the tools at hand, but it is unknown whether results will be comparable at the end.

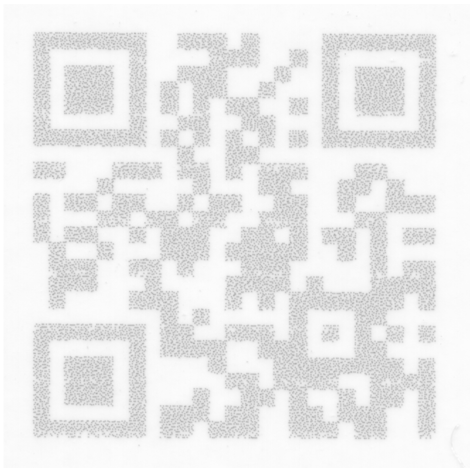

Abbildung 22: Ink jet QR-code with a tonal value of 20% at a resolution of 360 dpi

For this reason it would be better to use data of QR encoders they use when they analyze the image of their camera. Again that's beyond the scope of further work.

7 Acknowledgment

We like to thank the DFG and the Hochschule der Medien, Stuttgart for providing the printing machine (CD 74) and other infrastructure.

We like to thank several parties that supported our work directly. Prof. Dr. Martin Dreher at DFTA-TZ, Stuttgart and his group helped us to prepare the flexographic prints. Prof. Weichmann and his group at the University of applied science (HdM), Stuttgart provided us rotogravure samples. Offset prints we printed at the lithographic lab at the University of applied science (HdM), Stuttgart by the help of Mr. Koekert.

We used ImageJ as a tool to analyze the gathered images. To automate the procedure the macro faciliy was used. ImageJ was developed by the National Insitiue of Health (NIH), USA.

Literatur

[12413] ISO 12467. *Graphic technology – Process control for the production of half-tone colour separations, proof and production prints – Part 2: Offset lithographic processes*. ISO, Switzerland, 2013.

[18015] ISO/IEC 18004. *Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification*. ISO, Switzerland, 2015.