Reading QR Code Symbols with an Ultra-miniature Computational Diffractive Imager

David G. Stork and Patrick R. Gill
Rambus Labs
1050 Enterprise Way, Suite 700
Sunnyvale, CA 94089 USA
{*dstork,pgill*}*@rambus.com*

Abstract: We demonstrate in hardware that an ultra-miniature, lensless computational diffractive imager can yield images of QR code symbols sufficient for accurate code reading, even amidst modest variation in target orientation.

OCIS codes: 050.1960, 110.1758, 110.3010, 130.3120, 350.2770, 070.5010

1. Introduction

Inspired by the demonstration of integrated CMOS diffraction-based imaging in [1], we have developed and prototyped a new class of ultra-miniature lensless computational sensors and imagers. [2–6] Our devices rely on special phase anti-symmetric optical gratings that produce signals on a CMOS photodetector array that are invariant to variations in grating manufacture and to variations in source optical wavelength. The sensor signals retain the full Fourier image information (up to the two-dimensional Nyquist rate). Final images are computed from the raw sensor signals—by Fourier methods, matrix-inversion methods based on Tikhonov or total variation regularization, Bayesian estimation, or other methods—chosen based on prior information about the source, available computational resources, required fidelity of the final digital image or accuracy of the image sensing task.

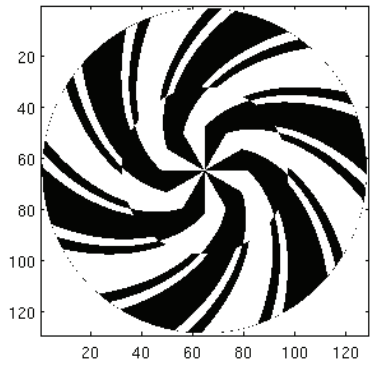

Fig. 1. Schematic of a phase anti-symmetric spiral grating covering 127 × 127 pixels, each of dimension 1.67 × 1.67 μm. Black represents an optical phase delay of $\lambda/2$ of typical incident optical wavelength, 550 *nm*, compared to that of white. This grating is one of 29 grating experiments on a single silicate layer atop an Aptina MT9J003 10*Mp* sensor connected through a USB 3.0 port to a PC, with signals read from each experiment's sensor area separately. The full grating thickness is 500 μm, and because the angle of view is 100°, light strikes the photosensor in an area extending beyond that immediately beneath the phase grating which in the present QR code reader case is 400 pixels across. [2, 3]

While the range of scales and resolutions available in this family of lensless sensors is large (from a few to a few hundred pixels on an edge) and can be chosen based on the application and desired final resolution, the sensor reported here has a circular spiral grating aperture 212 μm in diameter or 127 pixels (Fig. 1). The sensor field of view is 100°. By geometry, the total sensor area that can receive light corresponds to 400 × 400 pixels, thus extending beyond the phase grating itself. The sensory input to processing is thus 160,000 12-bit monochromatic pixels.

Whereas some image analysis tasks, such as motion estimation, may be solved by applying appropriate algorithms directly to the raw sensor input, QR code reading is very unlikely to work in the direct sensor domain because most of the code information is in the spatial phase domain. As such, we first compute an image of the QR symbol and then apply existing QR code reading software, with its fiducial-based localization and error correction. This processing architecture allows the same sensor to be used for general image acquisition.

We represent the input scene (such as an image of a QR code) as a vector $\mathbf{x}$ and the optical transfer matrix as $\mathbf{A}$, and thus the signals on the sensor are $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, where $\mathbf{n}$ is pixel-wise noise. The final 128 × 128-pixel digital image is an estimate of a general scene, computed for instance by Tikhonov regularization. A QR

code symbol is a binary image and any estimate should vary solely along the black-white boundaries—a small number of the total pixels. We used Tikhonov regularization to minimize the expected square error throughout the image, then found the alignment and spacing of the QR code grid using a radon transform followed by the Fourier transform of an oriented absolute different filter. The mean of the image pixels residing within each QR pixel was thresholded to provide our estimate of the sign of the QR code pixel. The computational cost of computing each image is roughly 5.26G flops, which is easily implemented even at video rates in parallel hardware such as an FPGA or GPU. (Our barcode images and reader results, below, were computed on a PC offline.) The computational cost of simple thresholding is negligible, and we do not consider the QR code reader, which operates in real time. Figure 2 illustrates the processing steps from target image to thresholded digital image.

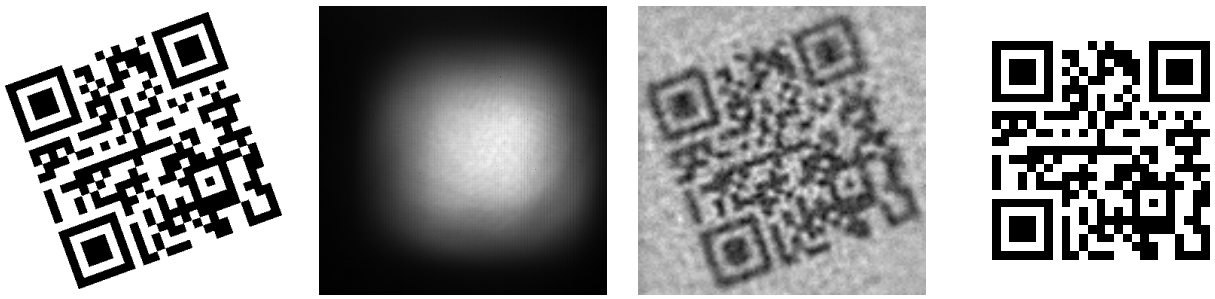

Fig. 2. Left-to-right: An instance of a *Version 2* (25×25) target QR code symbol (31 bytes); the raw signals in the 400 × 400 pixels array in our sensor; the digital image computed from the sensor signals using Tikhonov regularization; the final image, rotated and thresholded by line to yield roughly 50% white pixels. This final image is presented to *ZXing* QR code reading software, which decodes the image to extract its 31-byte code.

2. QR code reading accuracy

All target QR code payloads were 31-byte representations of snippets of text by Shakespeare, and the corresponding QR code target images were rendered on a rear-projection screen of approximately 500 lumens in an otherwise dark room, subtending a visual solid angle of $100^\circ \times 100^\circ$ from the position of the sensor, 15 *cm* away. The final computed digital image was 128 × 128 pixels. The generation of all QR code target images, the acquisition of sensor signals, the digital image computation (using Tikhonov norm regularization with regularization parameter γ set once, by hand), and simple automatic image thresholding were done on a PC. The resulting digital images were decoded using the open-source *ZXing* QR decoder, which has no adjustable parameters. [7]

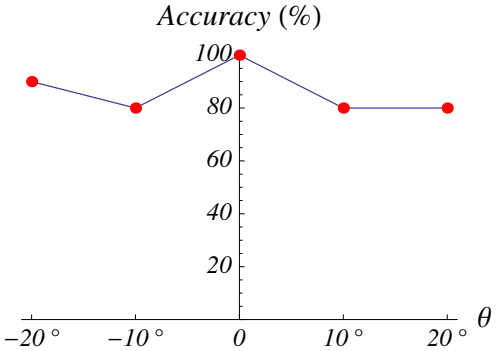

Fig. 3. Average QR code-reading accuracy as a function of the orientation of the source. There has been no optimization of the system based on parameters governing grating, image reconstruction, thresholding, luminance correction, edge detection and rectification, and code component square estimation (black or white), all of which are likely to improve reading accuracy and robustness amidst variations in orientation, scale and pixel noise.

We report here results from ten example codes, each presented in conditions $-20^\circ \leftrightarrow +20^\circ$ in 10° increments. The accuracy reported is based on the percentage of QR codes read perfectly (Fig. 3). These preliminary results for classification accuracy as a function of orientation of the code mark are encouraging. Moreover, in all but one case of a classification error, the software returned a "not decodable" signal—a signal that in a fielded commercial system could enable a user to interactively adjust the position or orientation of the sensor until the code is read.

There are a number of straightforward steps that should improve accuracy and insensitivity to scale, orientation and source noise. [8] The angle-dependency of the screen luminance and the physical geometry of the sensor lead to the overall light intensity (and hence digital signals) in the central sensor photoreceptors being higher than in the peripheral photodetectors. These facts lead to the contrast in the computed digital image in the central regions being higher than that in the peripheral regions, thereby degrading the accuracy of image binar-

ization. This inaccuracy can be removed by conditioning the raw sensor signals by multiplication by a simple radial function, boosting the signals at the periphery more than those at the center. Standard straight oriented edge-segment detection, with priors over the segment lengths and right-angle intersections, may improve the extraction of code squares, thereby improving reader accuracy and robustness. [9] The assignment of the binary value (white or black) to each extracted mark squares may be improved by pixel-based majority voting, or watershed transform, or other pixel-based estimation methods.

3. Conclusions

We have designed and built the world's smallest passive QR code reader sensor (e.g., requiring no scanning laser illumination) and have shown that the full system can yield digital images of quality sufficient for reading by QR code software. A number of straightforward modifications can be used to improve the accuracy and robustness to transformations and variations in the code symbols. Our work is the first to show that the principles of computational diffractive sensing and imaging can be successfully applied to the problem of mark reading and thus, by extension, related problems in pattern recognition. [10]

More distant, future work, will center on designing special-purpose phase gratings, tailored to the expected range of scales (and hence source spatial frequencies) of code symbols, and on improving algorithms for digital image creation and image processing for code reading. Such a specialized QR symbol reader would likely be ineffective for general image acquisition.

This work leads the way to many other application-specific diffraction-based image capture devices and applications.

References

1. P. R. Gill, C. Lee, D.-G. Lee, A. Wang, and A. Molnar, "A microscale camera using direct Fourier-domain scene capture," Optics Letters **36**, 2949–2951 (2011).
2. D. G. Stork, "Joint optics/signal processing design for computational diffractive sensing and imaging," in "Computational Sensing and Imaging (COSI)," (Kohala Coast, HI, 2014).
3. P. R. Gill and D. G. Stork, "Lensless ultra miniature imagers using odd-symmetry phase gratings," in "Proceedings of Computational Sensing and Imaging (COSI)," (Alexandria, VA, 2013).
4. P. R. Gill, "Odd-symmetry phase gratings produce optical nulls uniquely insensitive to wavelength and depth," Optics Letters **38**, 2074–2076 (2013).
5. P. R. Gill and D. G. Stork, "Digital camera with odd-symmetry spiral phase gratings supports full-resolution computational refocusing," in "Advanced Photonics (Optical Society of America Sensors Congress)," (2013).
6. D. G. Stork and P. R. Gill, "Lensless ultra-miniature computational sensors and imagers," in "SensorComm 2013," (Barcelona, Spain, 2013).
7. (2014). http://zxing.org/w/decode.jspx.
8. E. Ouaviani, A. Pavan, M. Bottazzi, E. Brunelli, F. Caselli, and M. Guerrero, "A common image processing framework for 2D barcode reading," Image processing and its applications **465**, 652–655 (1999).
9. R. Szeliski, *Computer Vision: Algorithms and applications* (Springer, New York, NY, 2011).
10. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification* (Wiley, New York, NY, 2001), 2nd ed.