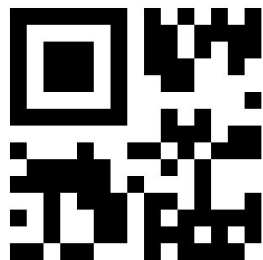


Own QR-code, motivation

- low capacity code
- detect at big distances
- “no training”

Applications: cleaners, cars, robots

Classic QR-code:



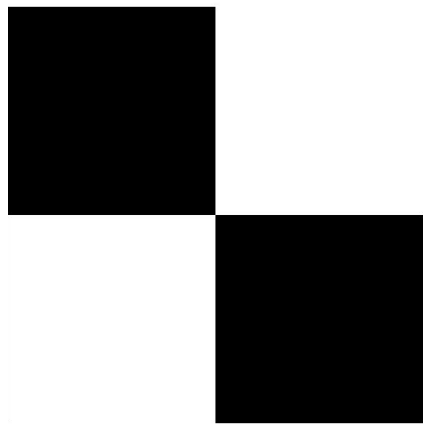
Too much info, too small squares

Proposed QR-code:



big squares

Own QR-code, marker to detect

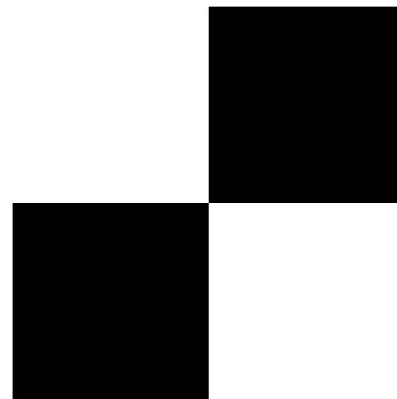


0

minimal kernel: $\begin{bmatrix} -1, 1 \\ 1, -1 \end{bmatrix}$

used kernel:

$\begin{bmatrix} -1, -1, 0, 1, 1 \\ -1, -1, 0, 1, 1 \\ 0, 0, 0, 0, 0 \\ 1, 1, 0, -1, -1 \\ 1, 1, 0, -1, -1 \end{bmatrix}$

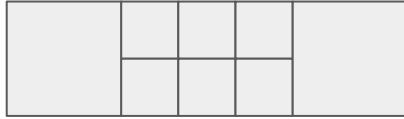


1

Convolution, high positive or high negative response, accelerate:
summed-area table:

https://en.wikipedia.org/wiki/Summed-area_table

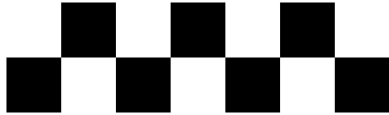
Own QR-code, coding superpixels:



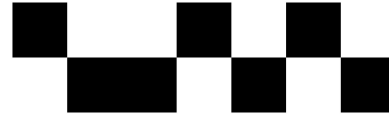
8 bits, codes range 0-255 (without checks)

Examples:

85:



213:



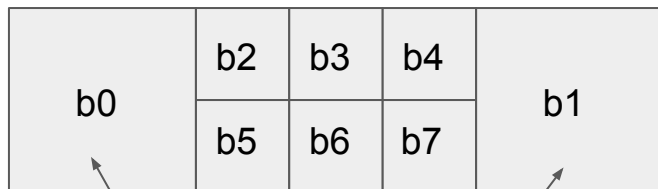
123:



30 (parity checks):



Own QR-code, parity checks



Parity checks bits

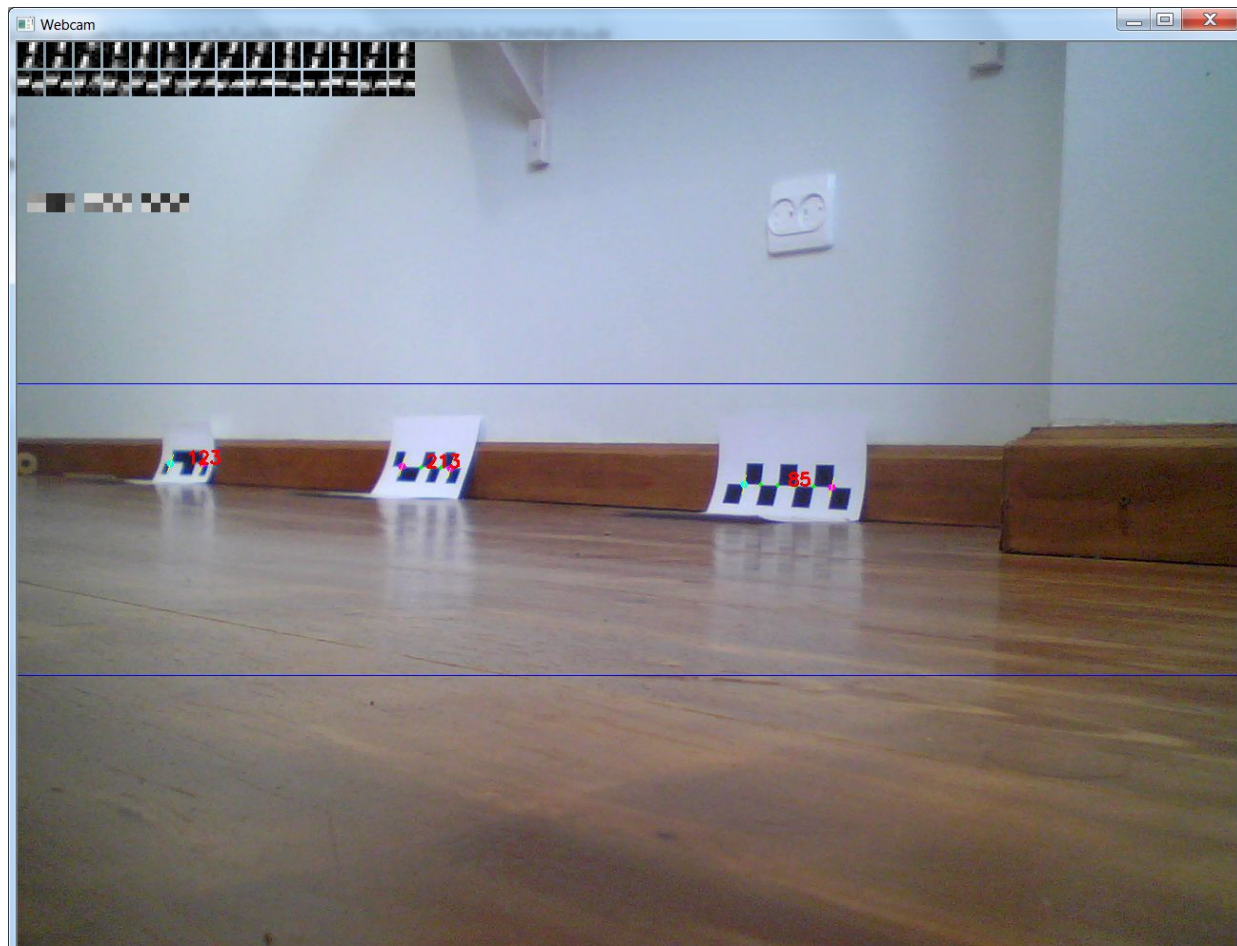
$$b0 = (b2 + b3 + b4) \% 2$$

$$b1 = (b5 + b6 + b7) \% 2$$

Classic QR codes (and CD-disks) use Reed–Solomon codes:

https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%A0%D0%B8%D0%B4%D0%B0_%E2%80%94%D0%A1%D0%BE%D0%BB%D0%BE%D0%BC%D0%BE%D0%BD%D0%B0

Own QR-code, prototype's detections



Own QR-code, prototype's false positives

