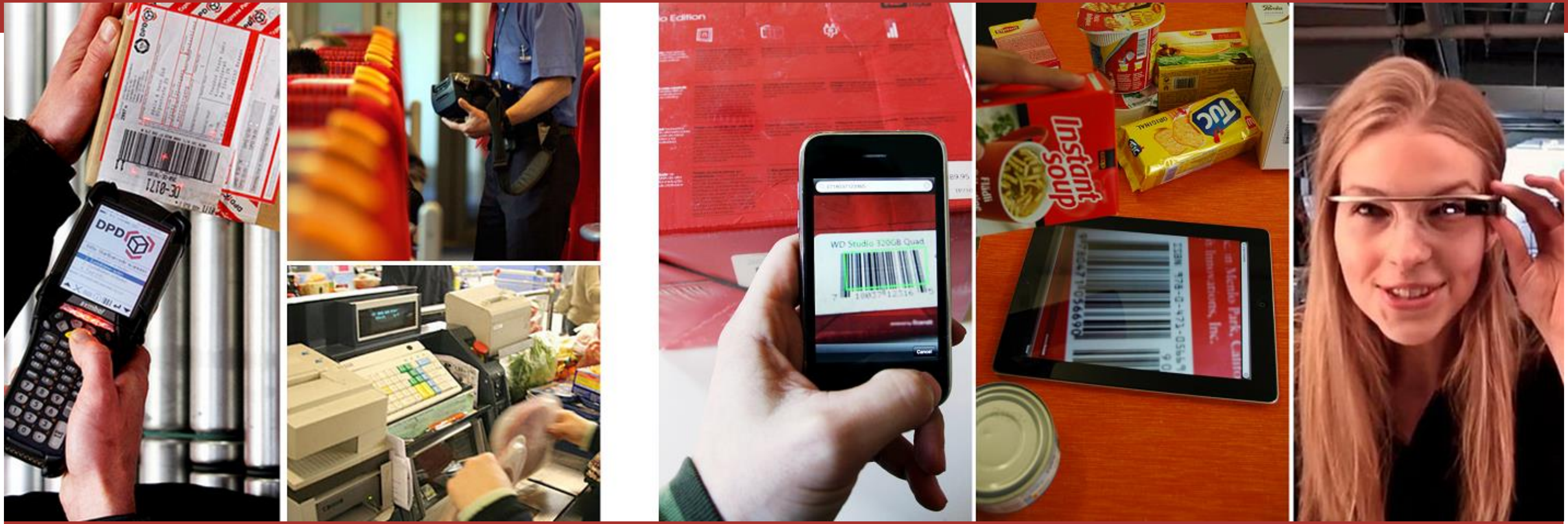
ETH zürich

Distributed Systems Group | Institute for Pervasive Computing



blur-resistant joint 1D and 2D barcode localization for smartphones

Gábor Sörös, Christian Flörkemeier

03.12.2013

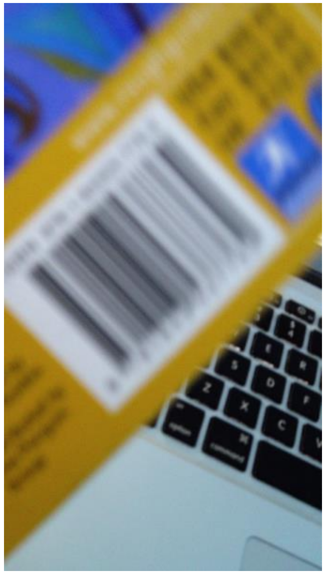
ubiquitous smartphone/tablet/glasses scanners would allow to access information on every physical object

smartphones/tablets/glasses

- are always with us
- have cameras, sensors, GPU, intuitive UI
- are easily programmable

difficulties with current scanning solutions

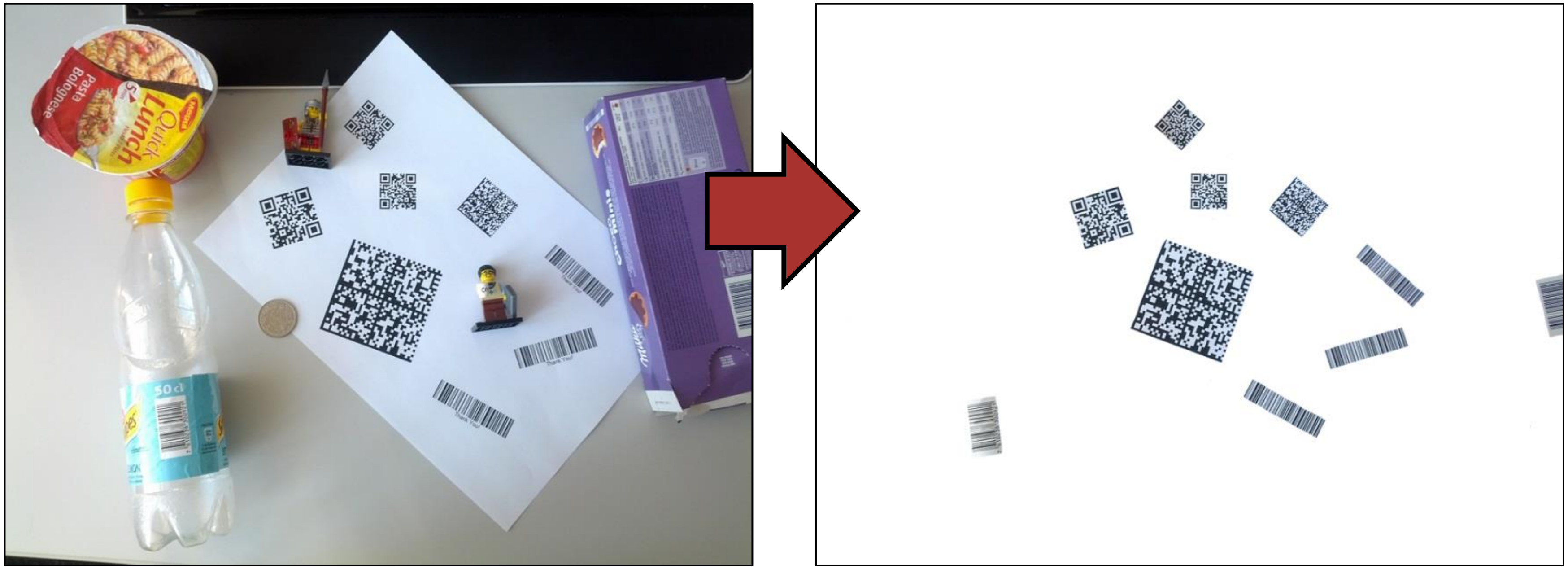
- no localization cue from laser
- various types of blur
  - out-of-focus blur
  - motion/shake blur
- limited image resolution



vision

- **turn every smartphone into a barcode scanner with laser-like performance**
- by adapting the latest results in image restoration to the specific properties of barcode images and
- by leveraging the advanced computing and sensing capabilites of smartphones

first step: barcode localization



challenges in barcode localization

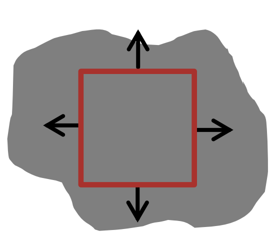
size orientation symbology blur

observations

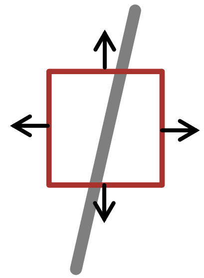
- 1D barcodes contain lots of edges
  - blur deletes many of them
- 2D barcodes contain lots of corners
  - blur smears corners but they still remain corners
- codes are almost always black and white
  - blur mixes black and white to gray

➡ *detect areas with edges and/or corners AND low saturation*

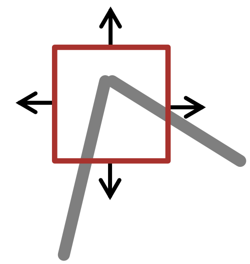
edge and corner measures (1)



flat region:
no change in any direction



edge/bar region:
change in one direction



corner region:
change in all directions

---

window-averaged change in intensity for the shift [u,v]:

$$E(u,v) = \sum_{x,y} w(u,v)[I(x+u, y+v) - I(x,y)]^2$$

window function

shifted intensity

intensity

edge and corner measures (2)

consider the second-order Taylor series expansion of E(u,v)
we have for small shifts [u,v] a bilinear approximation:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

where the 2x2 matrix M is calculated from the image derivatives:

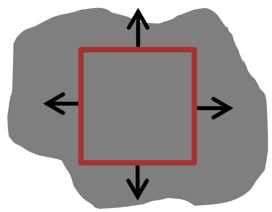
$$M = \begin{bmatrix} < I_x^2 > & < I_x I_y > \\ < I_x I_y > & < I_y^2 > \end{bmatrix} = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{xy} & C_{yy} \end{bmatrix} \qquad C_{i,j} = \sum_{(x,y)\in D} w(x,y) I_i(x,y) I_j(x,y)$$

---

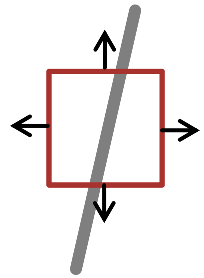
important properties of M:

- the eigenvectors form an orthonormal basis and show the directions of the fastest and the slowest change in E
- the eigenvalues λ_1 and λ_2 are real and non-negative and show the rate of change

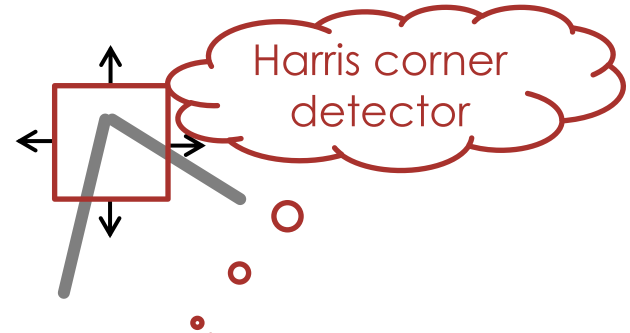
edge and corner measures (3)



flat region:
both eigenvalues are small



edge/bar region:
one eigenvalue is dominant



corner region:
both eigenvalues are big

… without any further details …

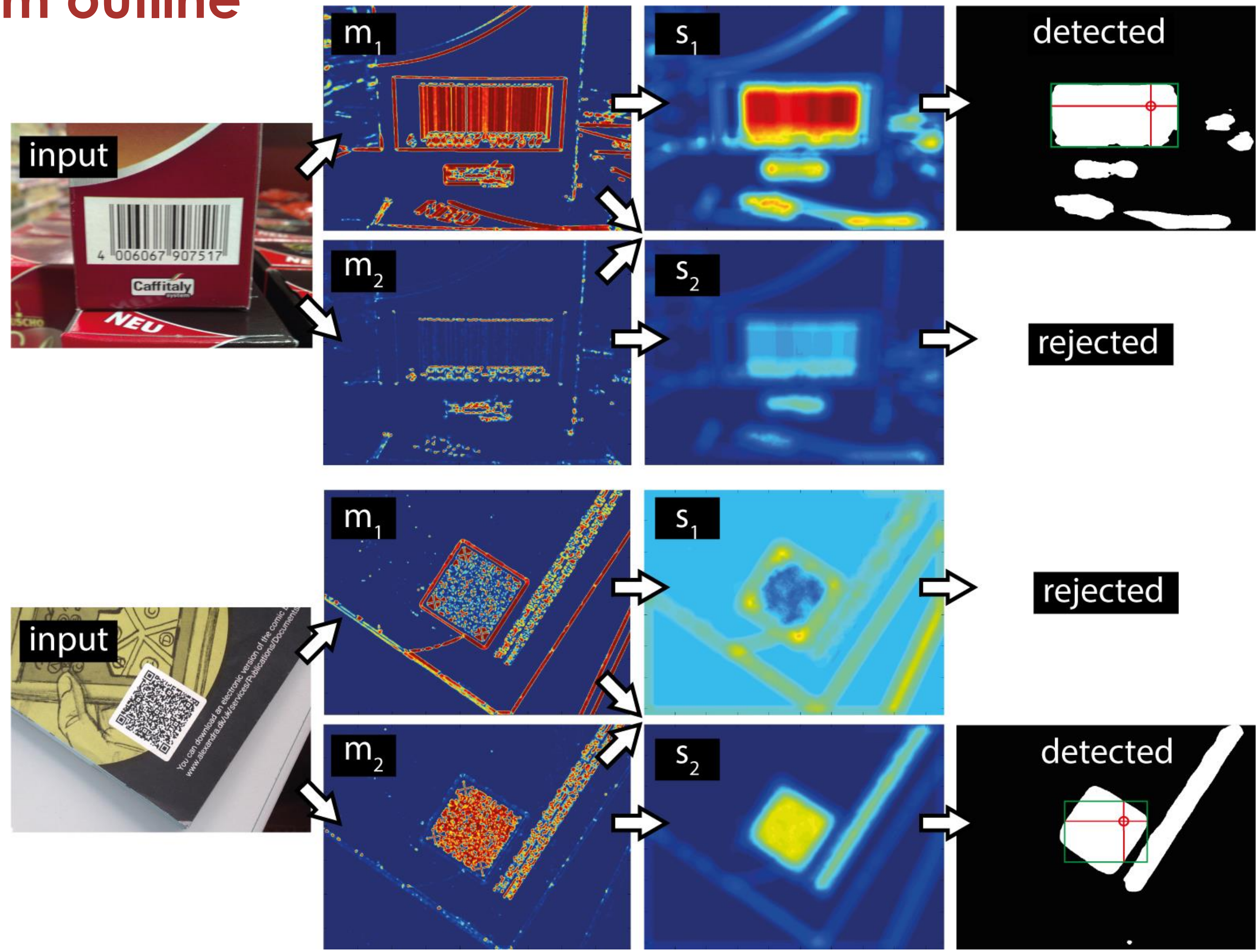
$$m_1 = \frac{(C_{xx} - C_{yy})^2 + 4C_{xy}^2}{(C_{xx} + C_{yy})^2 + \epsilon}, m_2 = \frac{4(C_{xx}C_{yy} - C_{xy}^2)}{(C_{xx} + C_{yy})^2 + \epsilon}$$

m_1 is big at edges and bars
m_2 is big at corners and pi/2-periodic patterns
both are normalized between [0,1]

algorithm outline

1D

2D



evaluation

we compare our algorithm to the state of the art:

- Gallo et al. [2011] – simple derivative filters (1D)
- Tekin et al. [2012] – orientation histograms (1D/2D)
- Katona et al. [2013] – mathematical morphology (1D+2D)

on public images:

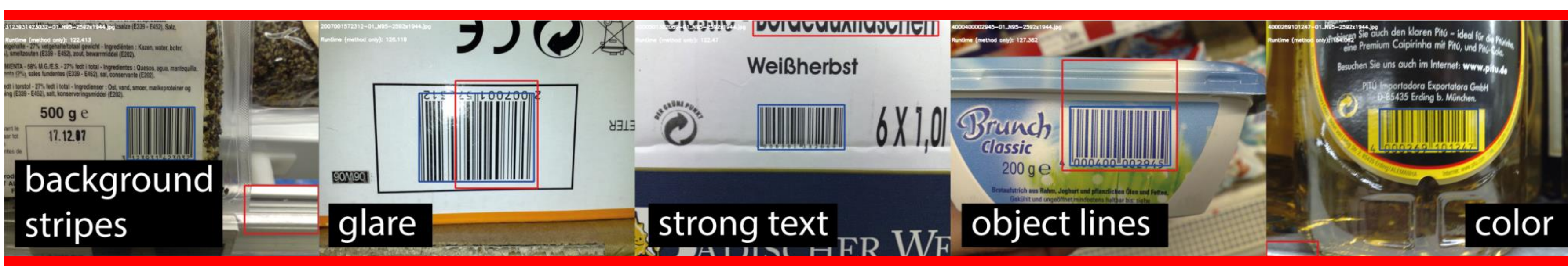
- Wachenfeld et al. Münster EAN dataset (~1000 images)
- Dubská et al. QR dataset (~400 images)
- New EAN images with iPhone 5 without AF (200 images)
- New QR images with iPhone 4S (120 images)

using the detection criterion:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \geq 0.5$$

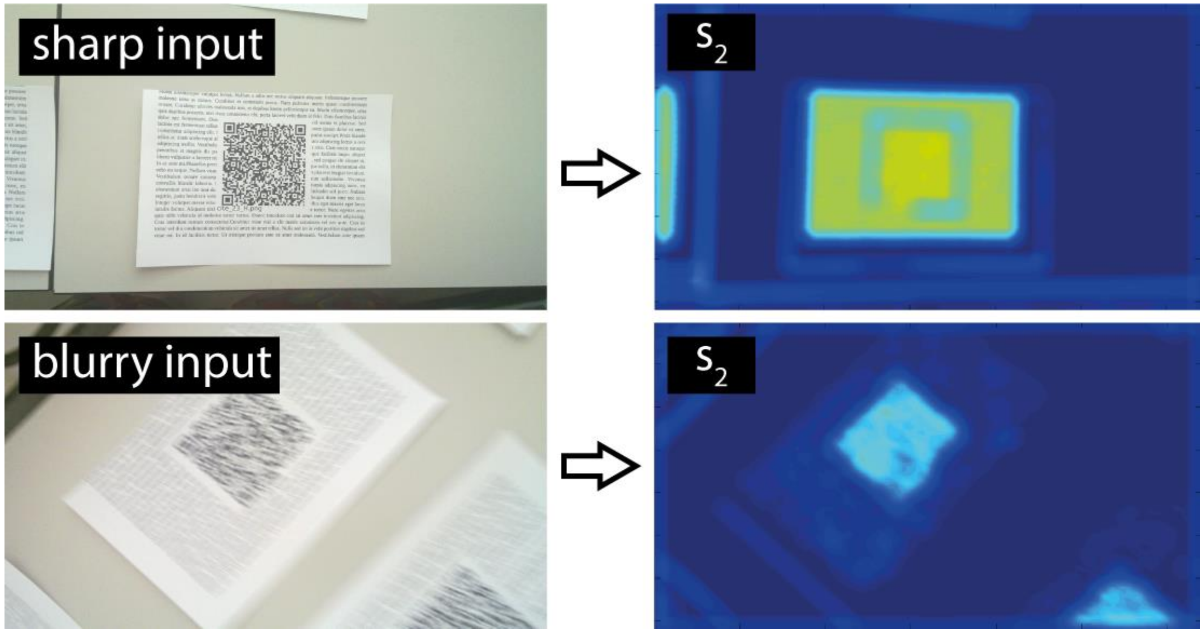
examples


sharp
text
tilted
rotated
colored
blurry
text
tilted
rotated
rotated-defocused
glare
text
round
curved


background stripes
glare
strong text
object lines
color

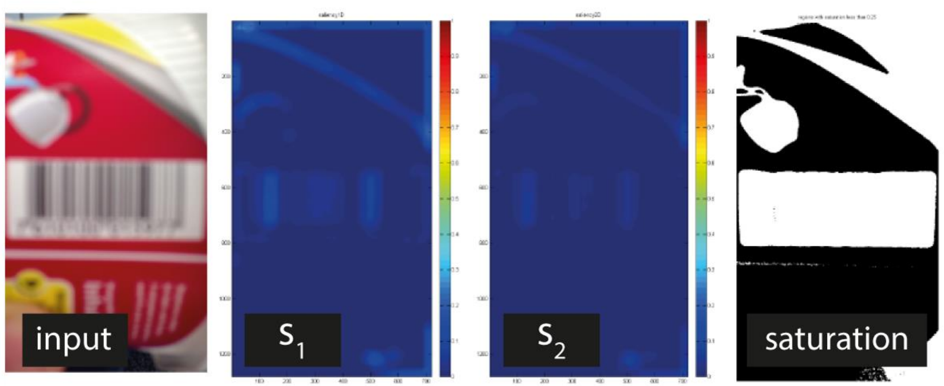
examples (difficulties)

2D



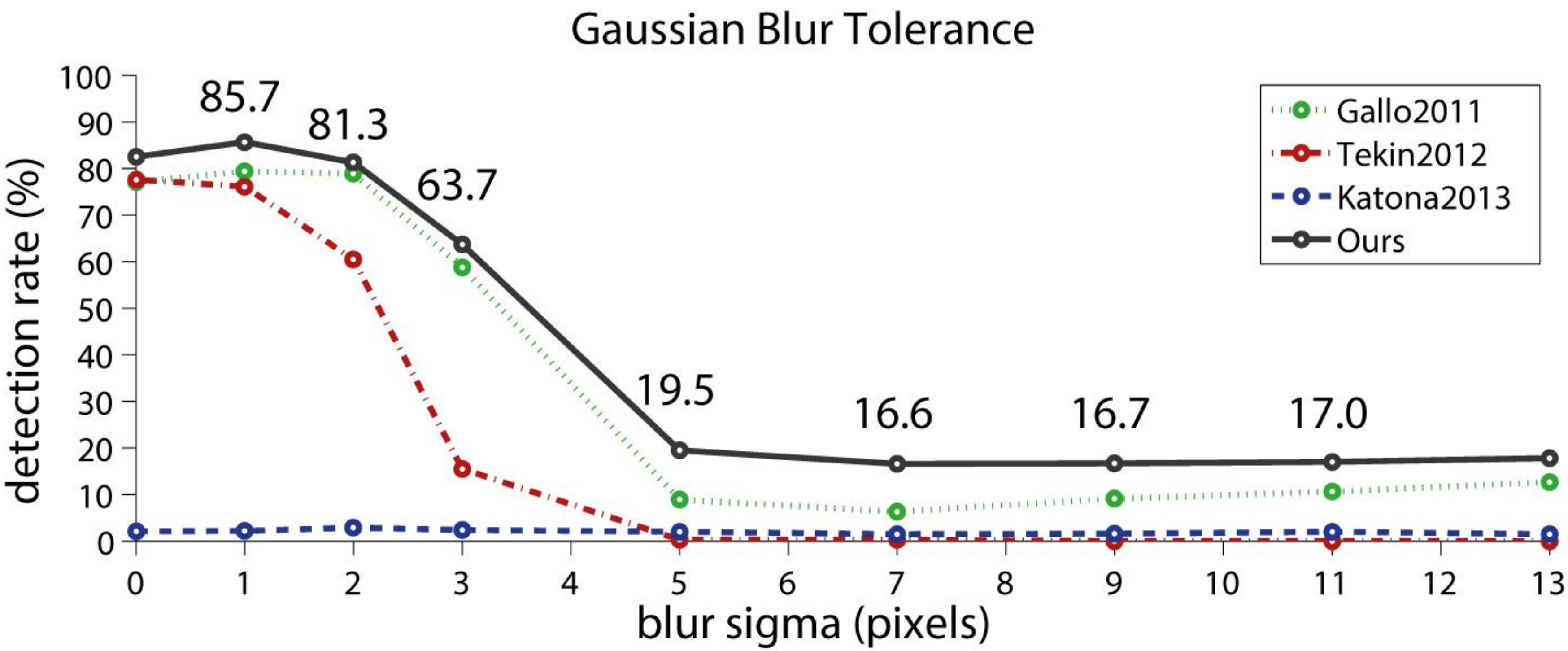
text lines look like a 1D code

1D

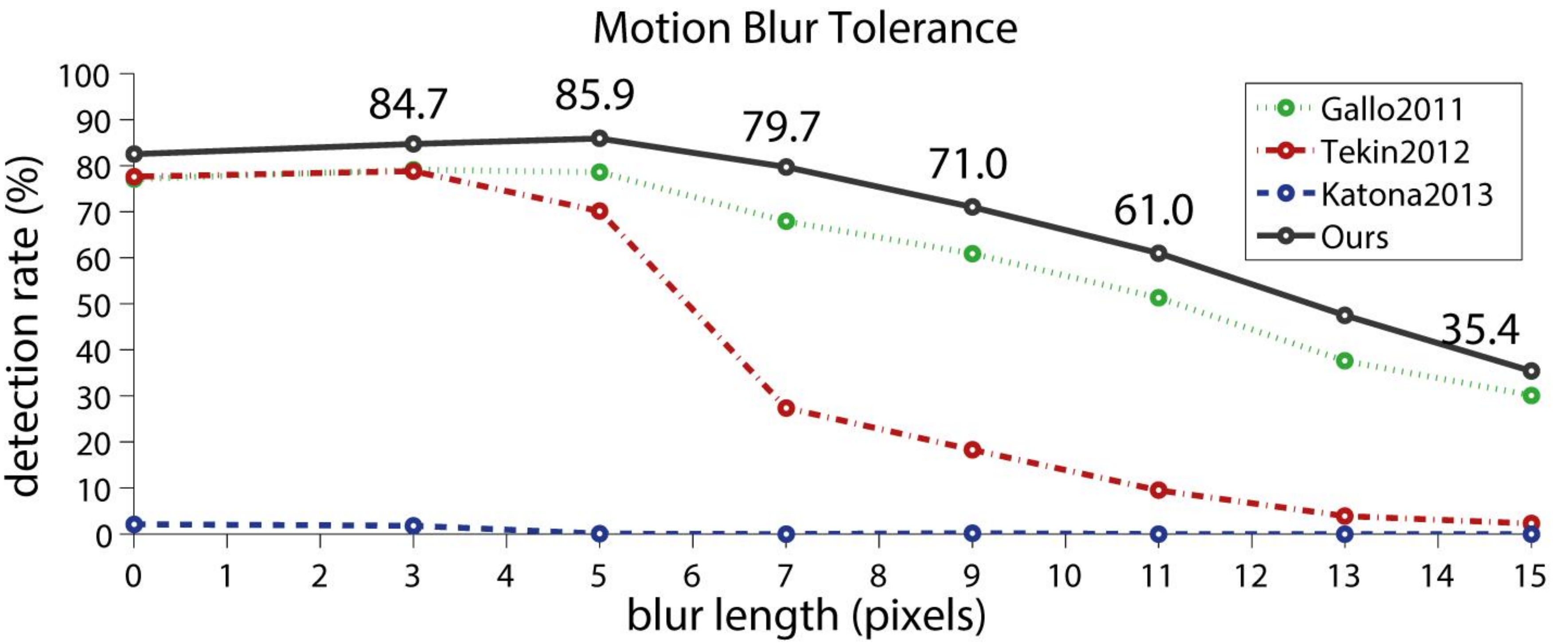


codes are almost always black and white → convert to HSV color system

experiments on ~1000 EAN images

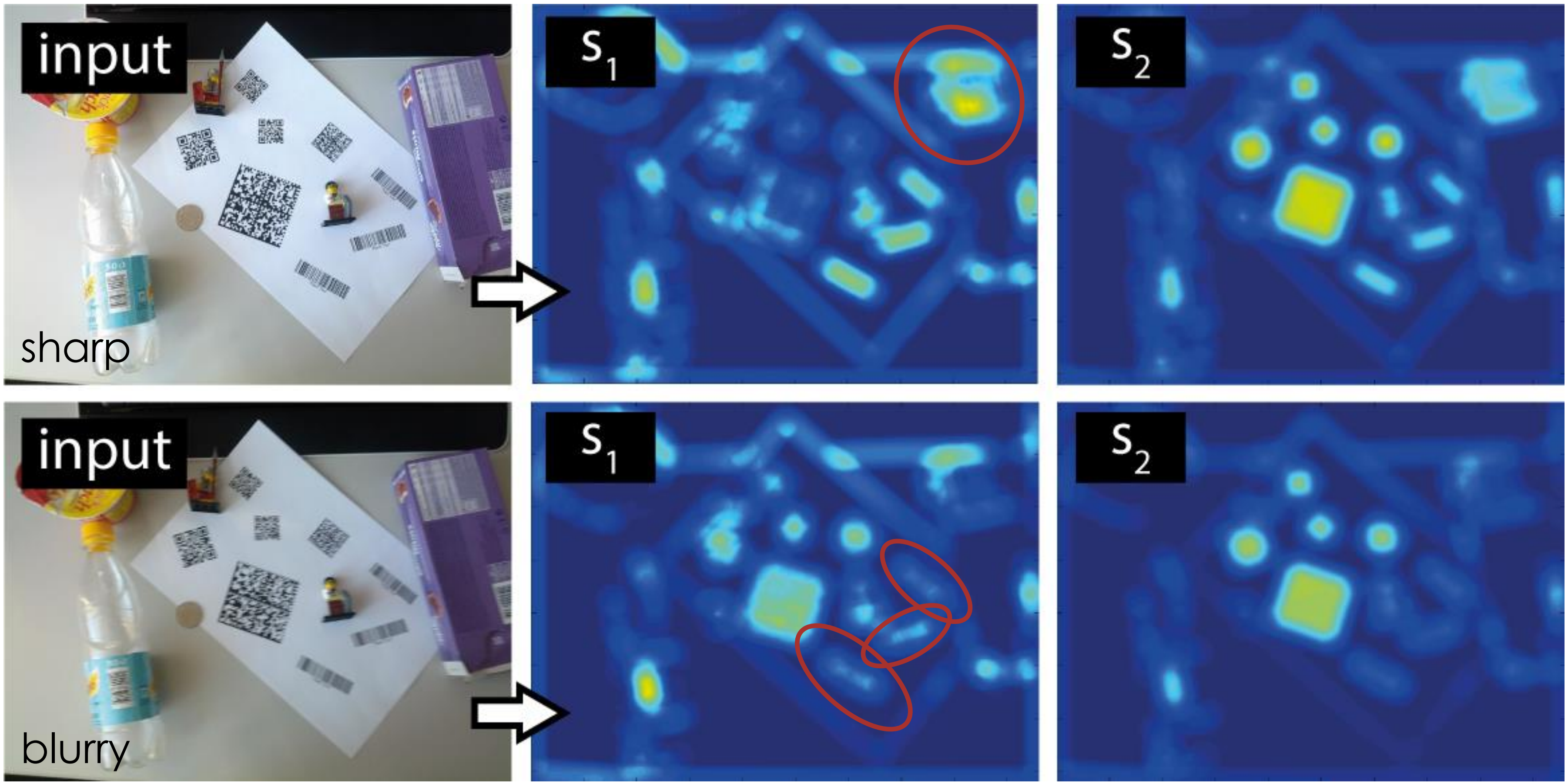


experiments on ~1000 EAN images



examples on our images

multiple codes



1D sensitive to blur

2D works well in both cases

runtime comparison

algorithm	Gallo2011	Tekin2012	Katona2013	Ours
runtime PC (960x720)	27 ms	49 ms †	63 ms	73 ms
runtime smartphone (640x480)	85 ms	26 ms	173 ms	380/118 ms ‡

† also including image format conversions
‡ CPU/GPU

summary

we proposed a new, combined barcode localization algorithm based on the structure matrix and the saturation of the pixels

the method can localize barcodes

- of various symbologies
- at different orientations
- at fairly wide scales
- more robust to blur than previous approaches

thank you