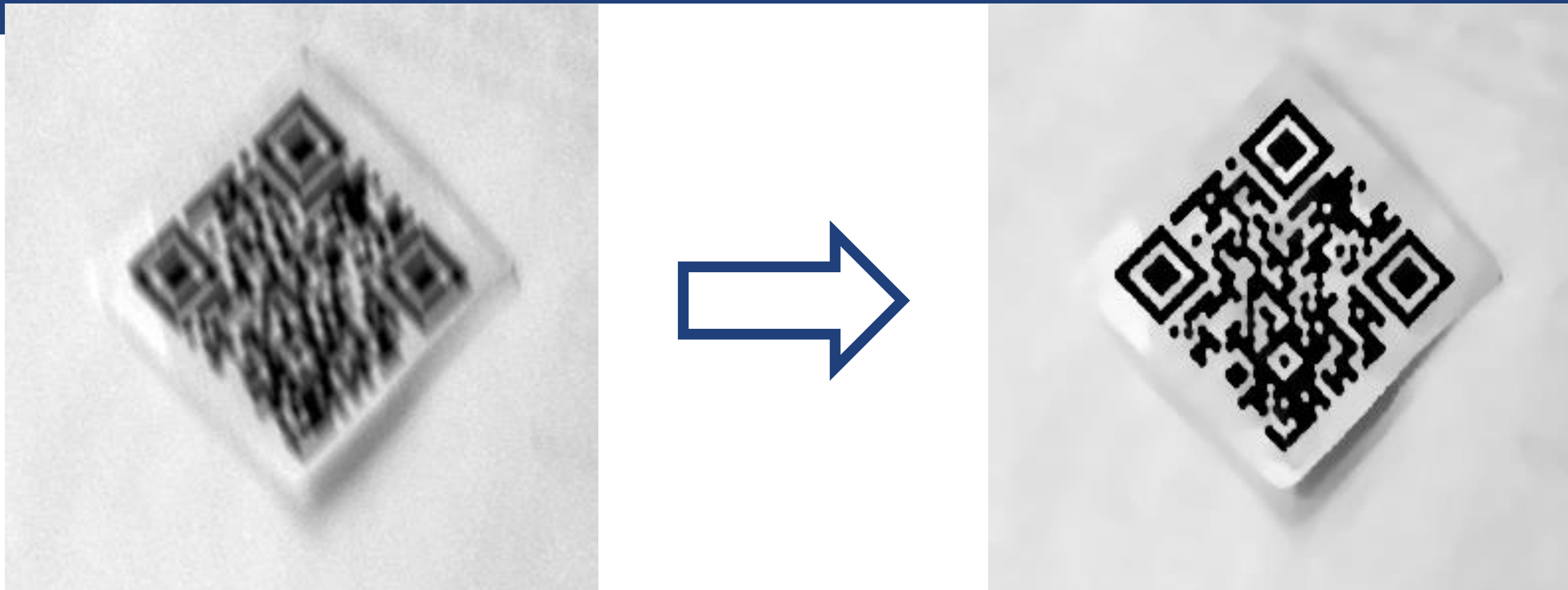
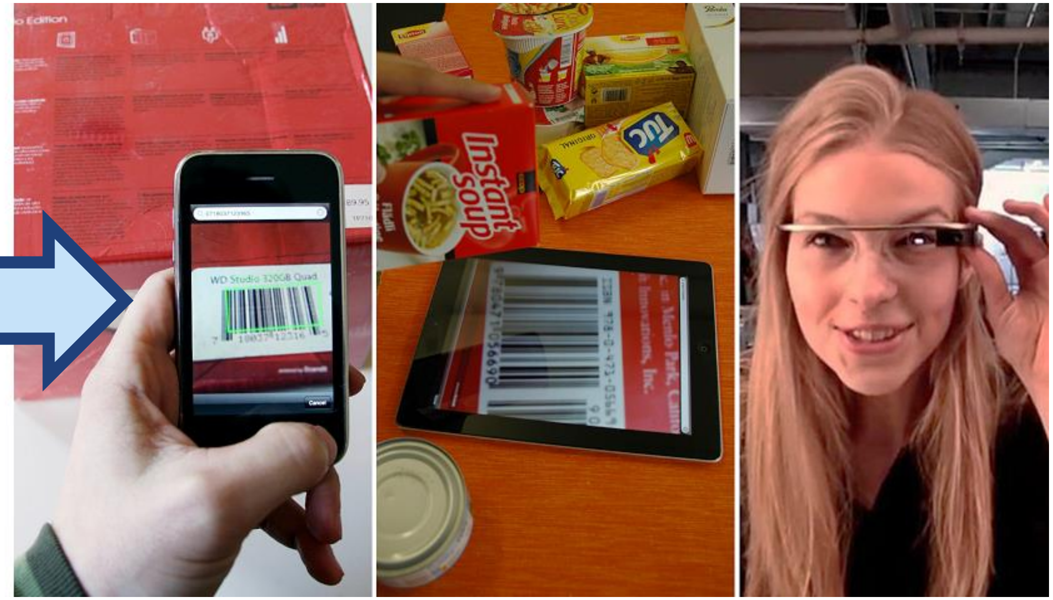


fast blur removal for wearable QR code scanners

Gábor Sörös, Stephan Semmler, Luc Humair, Otmar Hilliges

ISWC 2015, Osaka, Japan

traditional barcode scanning



next generation barcode scanning

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

smartphones/tablets/watches/glasses

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

scanning QR codes with wearable devices

Quick Response (QR) codes

- are found in numerous applications
  ticketing, shopping, logistics, etc.
- encode more information than barcodes
- have stronger error correction than barcodes

- wearable scanner SDKs are available for free

scanning QR codes with wearable devices

motion blur makes the codes unreadable

our goal: recover the information from motion-blurred QR codes

our input

our output

basics of blurry image formation

uniform blur model

sharp scene

$\boldsymbol{I}$

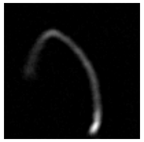
blurry scene

$\boldsymbol{I} * \boldsymbol{k}$

observed image

$\boldsymbol{B} = \boldsymbol{I} * \boldsymbol{k} + \boldsymbol{n}$

convolution with a blur kernel $\boldsymbol{k}$



adding camera noise $\boldsymbol{n}$

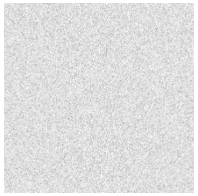


figure inspired by Robert Fergus

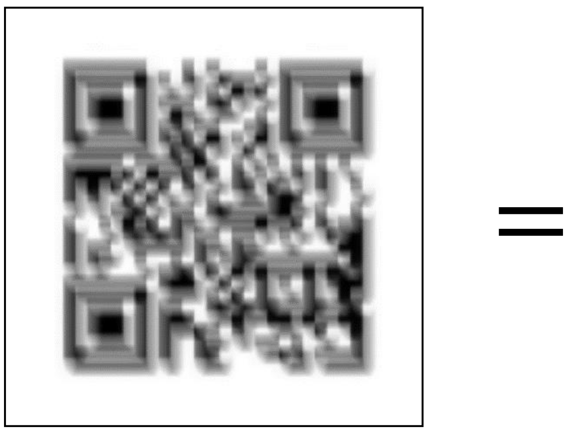
blur removal problem

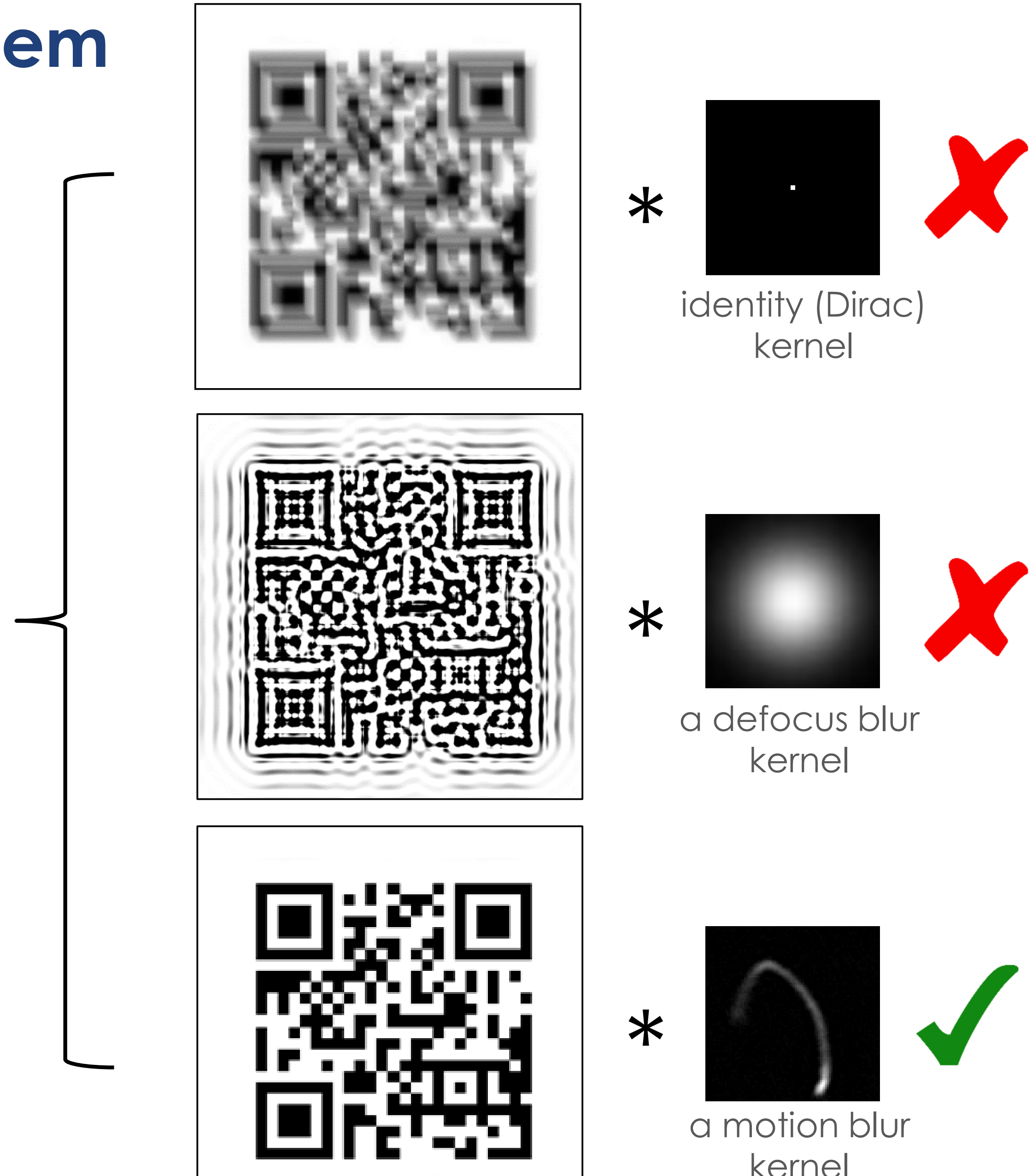
deconvolution:

$$B = ? * k + n$$

blind deconvolution:

$$B = ? * ? + n$$

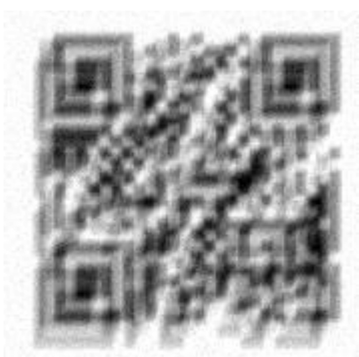




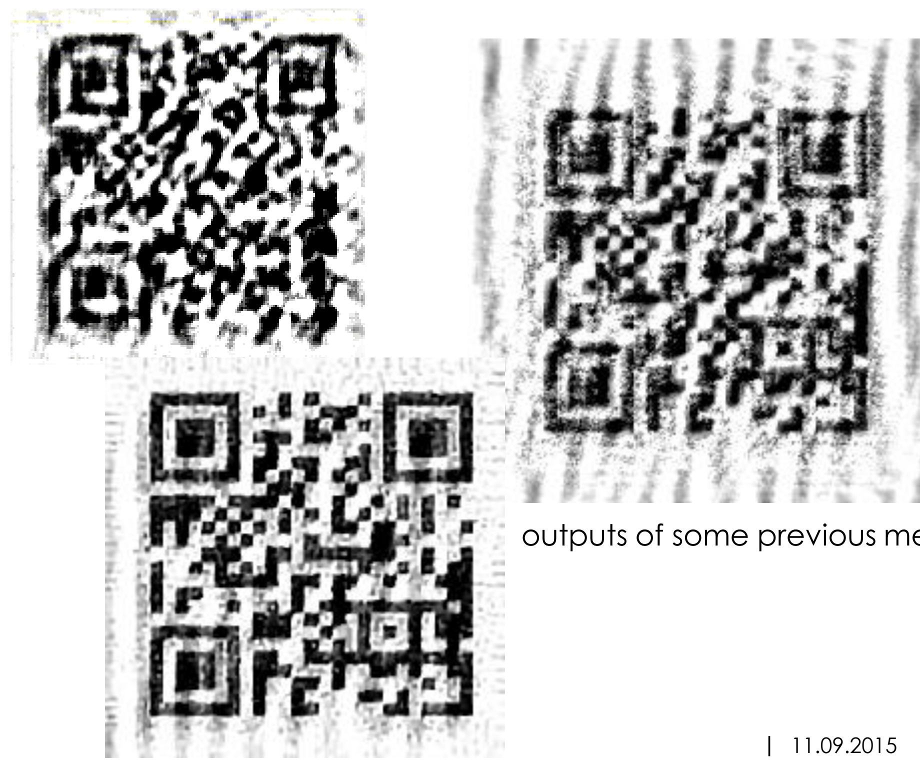
blind deconvolution for QR scanning?

existing blind deconvolution algorithms

- are slow even on PC
- are tuned to natural images
- usually fail on QR codes (structure very different!)



input



outputs of some previous methods

observations for deblurring QR codes

- blur can be estimated from the **many QR edges**
  - but we need to suppress the small structures [Xu2010]

- QR codes do not need to **look** good for **decoding**
  - in contrast to photographs, where restoration quality counts
  - our main concern is speed

- QR codes include **error correction** / **checksum**
  - the algorithm can stop when the checksum is correct
  - false decoding is practically impossible
  - only partially restored codes might be decoded too

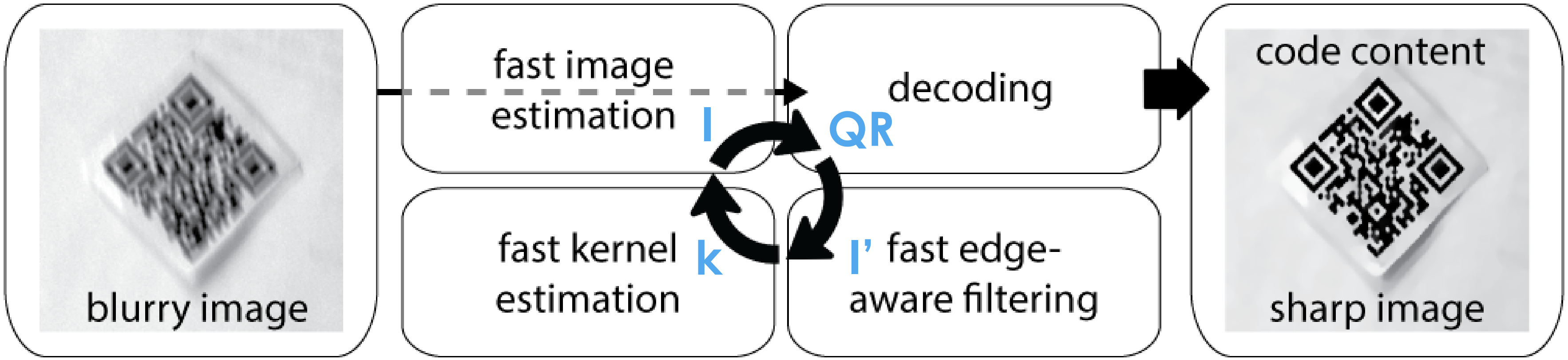
restoration-recognition loop

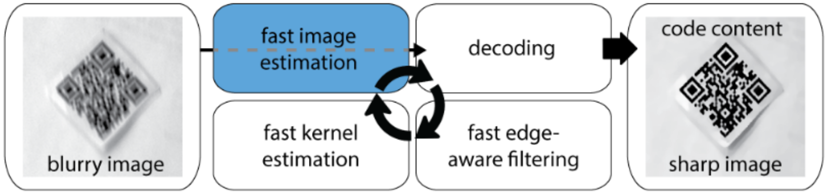
blind deconvolution via energy minimization

$$\underset{I,k}{\operatorname{argmin}} \|B - k * I\| + \lambda_I p_I(I) + \lambda_k p_k(k)$$

we follow a common recipe for blind deconvolution [Cho2009]

- alternate between solving for **I** and solving for **k**
- suppress noise and boost edges
- enforce QR properties
- try to decode at every iteration





fast image estimation

given **B** and **k**, estimate **I**

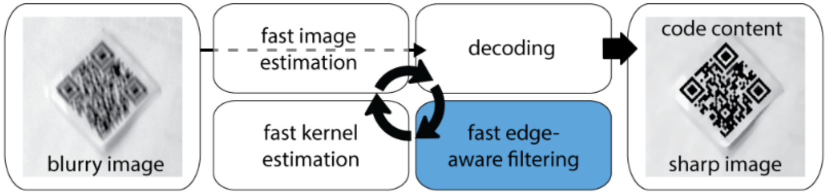
$$\underset{I}{\operatorname{argmin}} \; \|B - k * I\|_2^2 + \lambda_I \| \nabla I \|^{\alpha}$$

prior on gradients

- algorithm of Krishnan and Fergus [Krishnan2009]
- with $\alpha = 1$ the solution is particularly simple [Wang2008]

- solution via FFTs and pixel-wise thresholding equations further details omitted
- fast and good quality (compared to others)

fast edge-aware filtering



- the restored **I** is often imperfect (contains ringing and noise) and cannot be used directly to estimate **k**

- use image filters to **suppress noise** and **boost edges** [Cho2009]

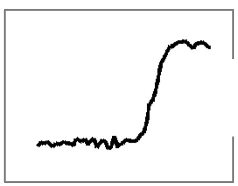
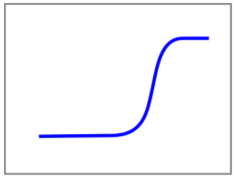


image from [Cho2009]

→



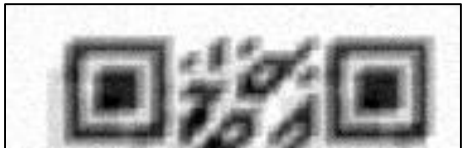
Bilateral filter

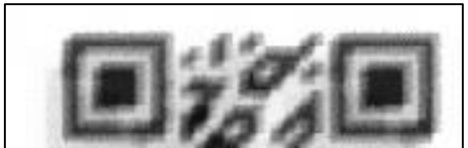
Suppress noise and small details

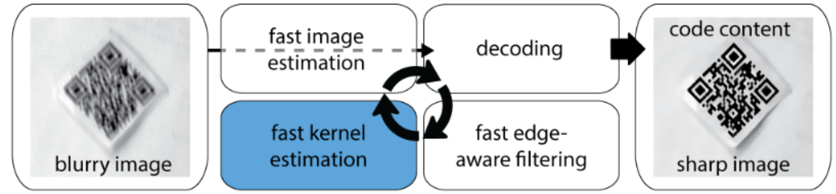
Shock filter

Restore strong edges

- in our work: Joint Weighted Median Filter [Zhang2014]
significantly faster while similar quality on black & white images





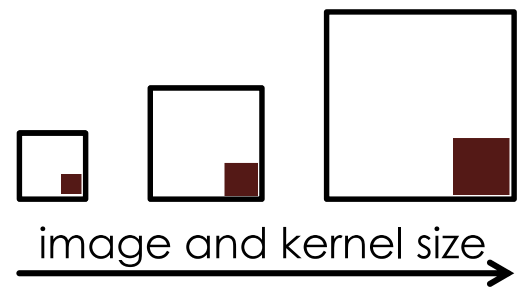


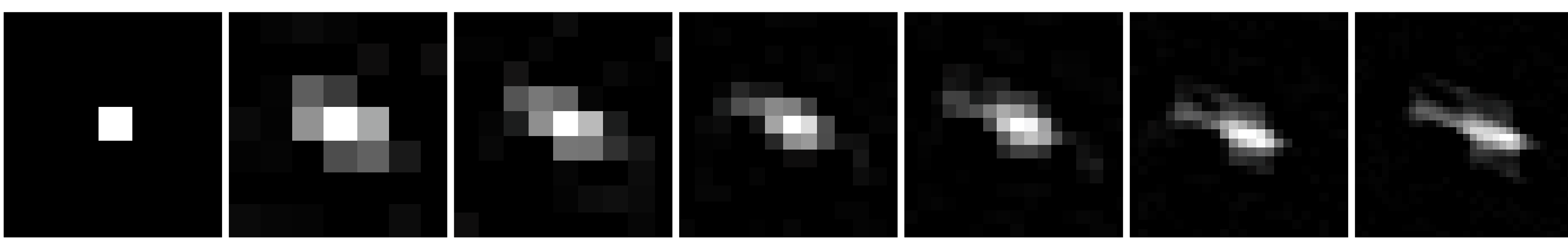
fast kernel estimation

given **B** and **I**, estimate **k**

$$\underset{k}{\operatorname{argmin}} \; \|\nabla B - k * \nabla I\|_2^2 + \lambda_k \underbrace{\| k \|_2^2}_{\text{prior on kernel}}$$

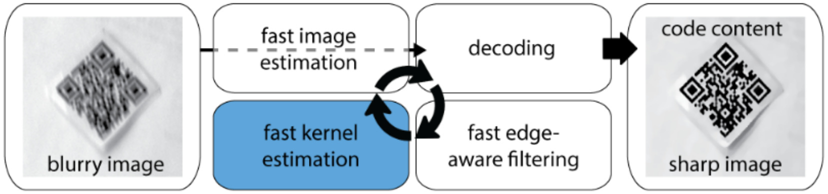
- in image gradient domain
  - not using pixel values simplifies the equations [Cho2009]
- solve via conjugate gradients and FFTs
- shift to geometrical center
  - discard small disconnected parts
- repeat over multiple image scales
  - aids the convergence to the correct kernel

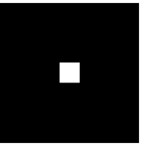




fast kernel estimation

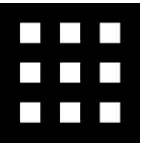
initialization





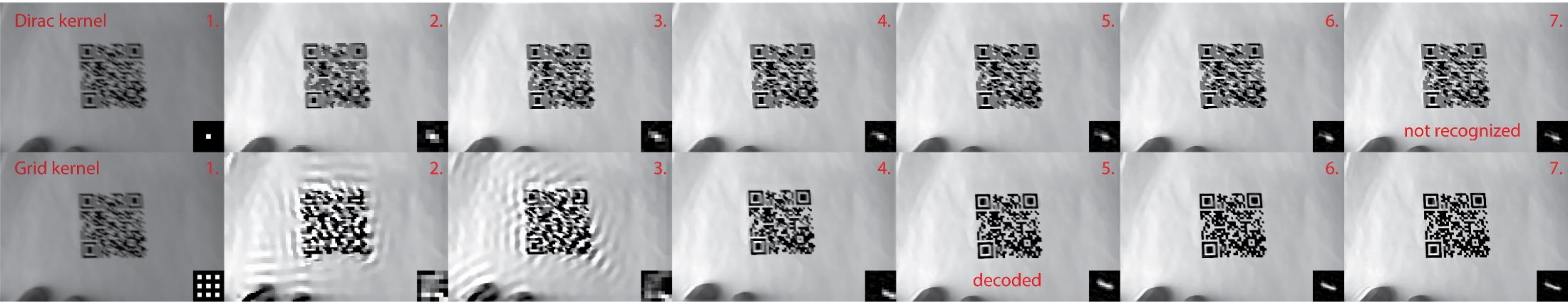
peak (Dirac) kernel

- usual choice
- faster

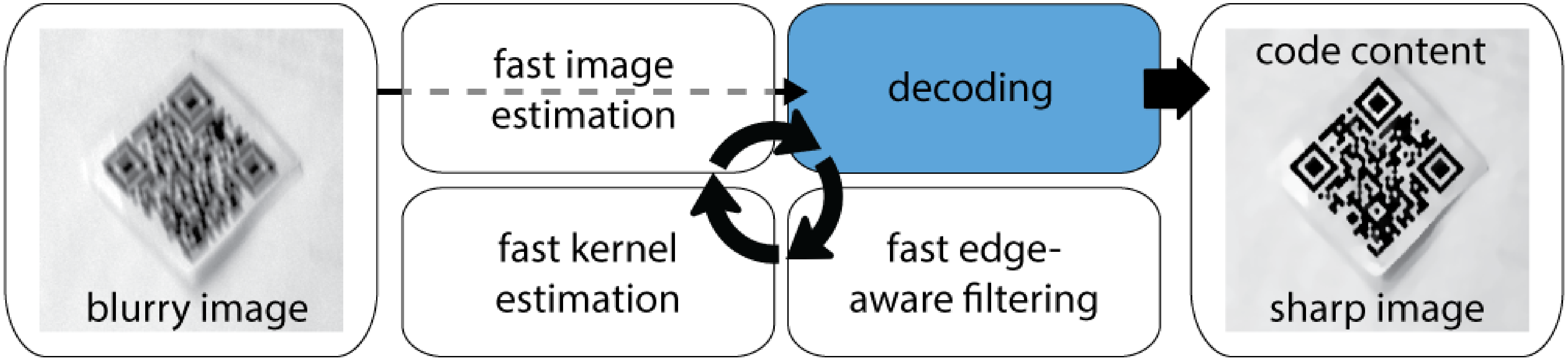


grid kernel

- helps with large blur, but converges slower in general
- (use motion sensors to decide which one is better)



restoration-recognition loop



- we iterate on each scale for refinement
- we iterate over multiple scales for better convergence

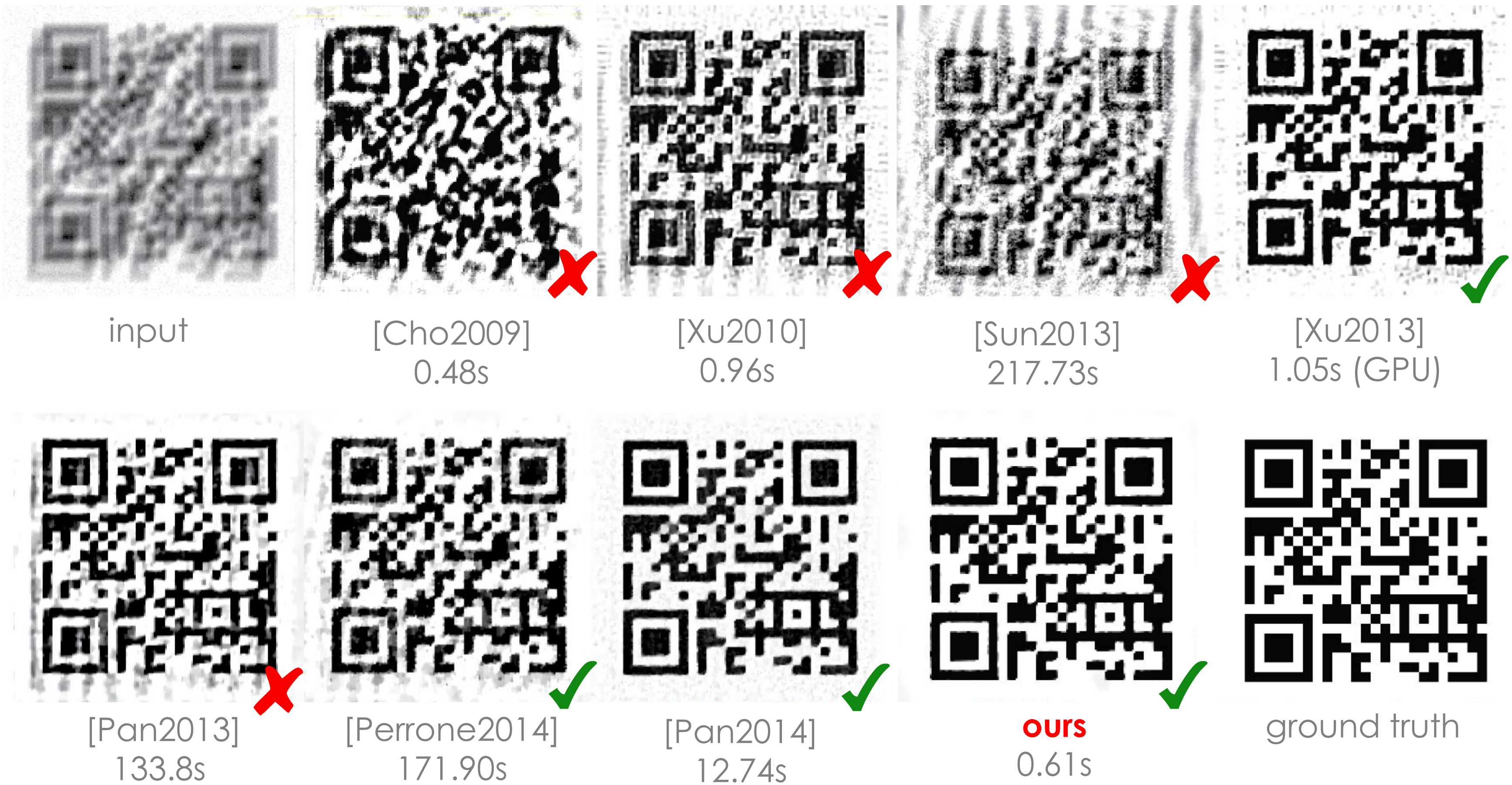
- we use a conventional open-source QR decoder

implementation

- OpenCV – cross-platform image processing in C++
- FFTW – fast Fourier transform
- ZBar – open-source decoder
- Android recorder application
  720x480 preview frames
  300x300 search window (~uniform blur)
- camera response function (CRF) must be linear

- experiments on
  - Lenovo T440p laptop
  - Motorola Nexus 6 smartphone
  - Google Glass smartglasses

experiments (synthetic blur)

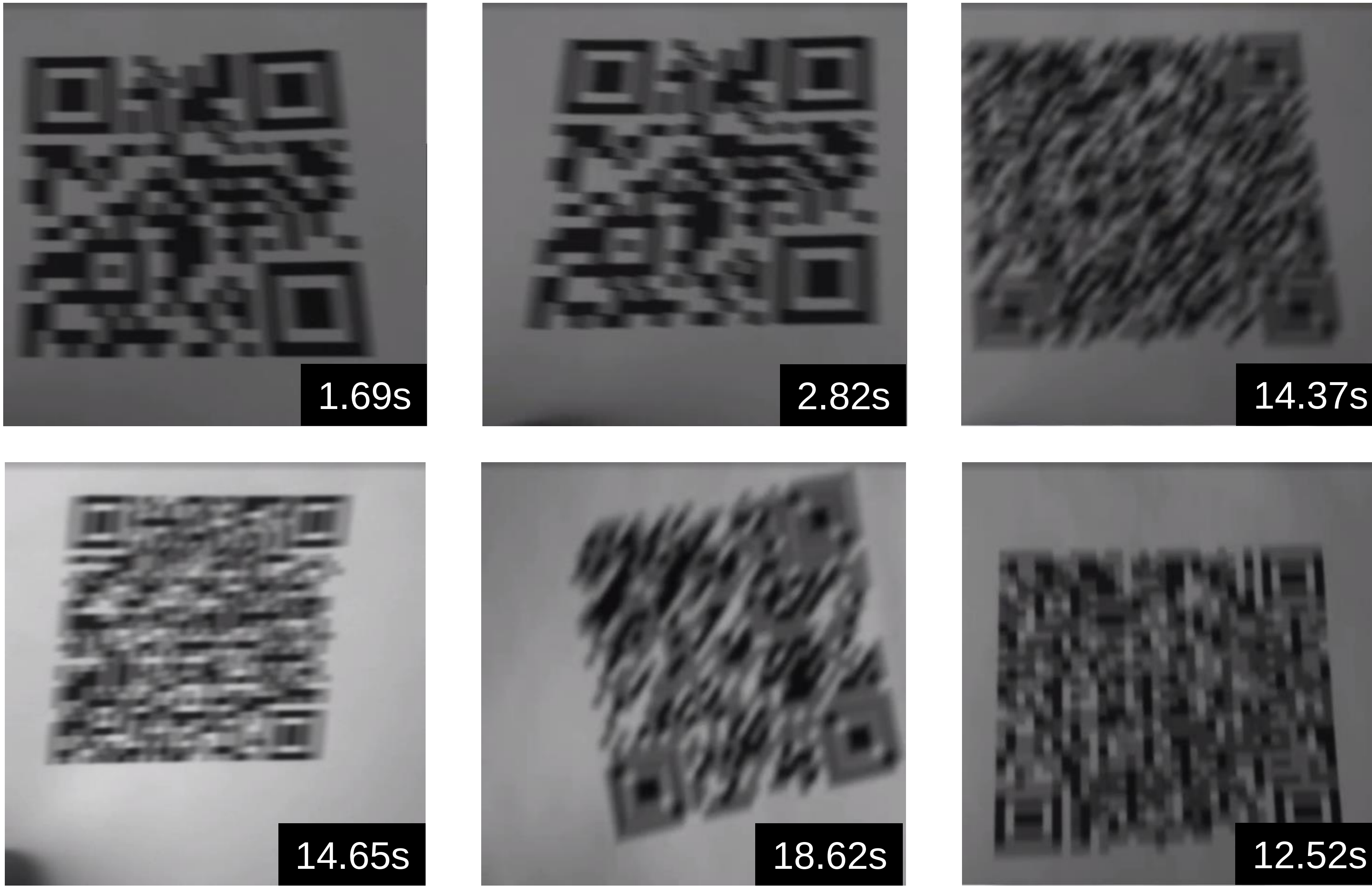


quality is on par with the state of the art, and a magnitude faster

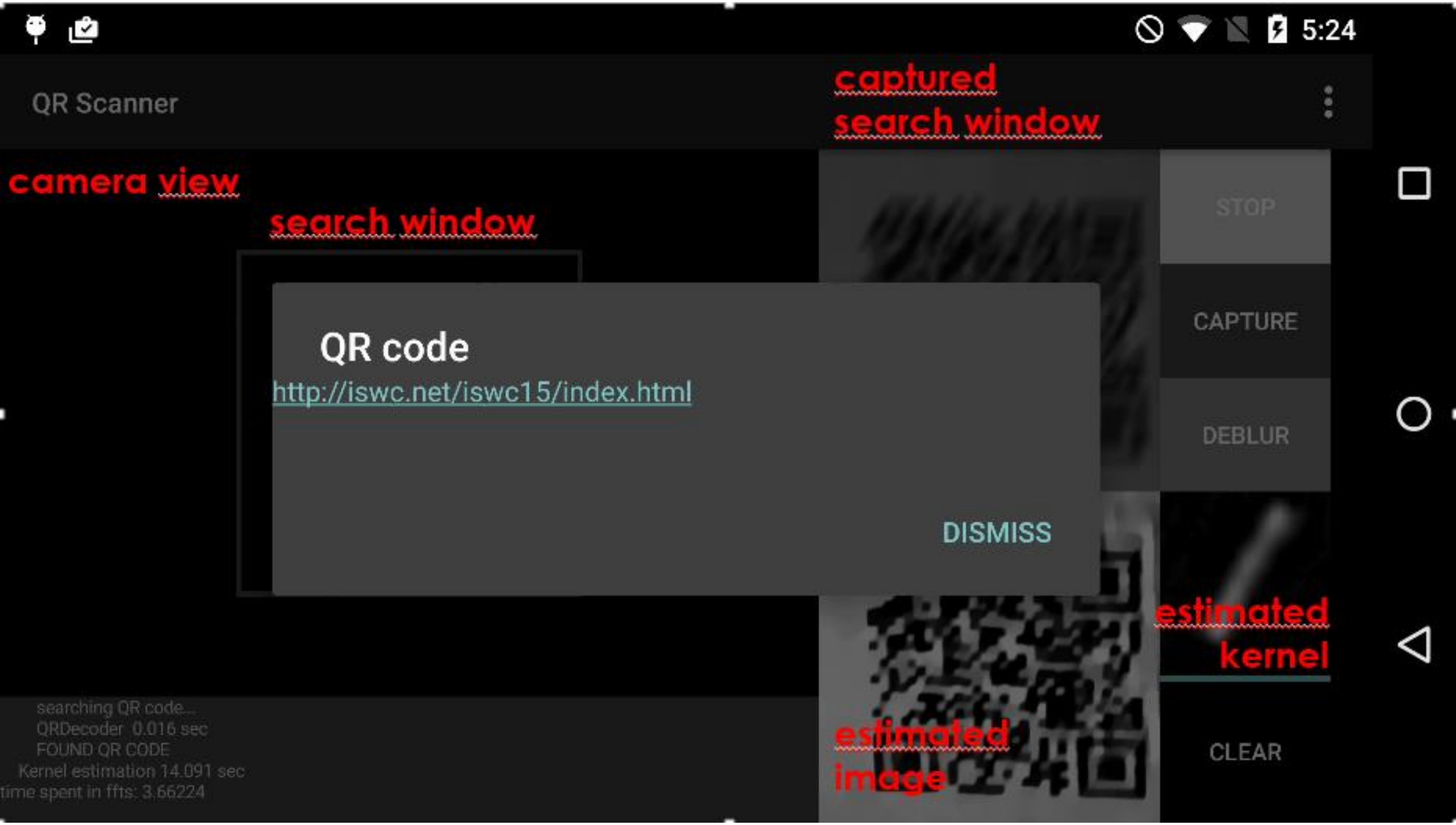
experiments (real blur)

experiments (real blur)

experiments (real blur)



a challenging example



Nexus 6 screen capture

limitations

- uniform blur
  - QR error correction helps with slightly non-uniform blur
- camera response function
  - online calibration possible?
- speed: still not real time
  - calculate FFT on mobile GPU
  - run in parallel with decoding other frames

future work

- use inertial sensors to estimate camera motion
  → requires precise camera-IMU synchronization
  → need to know the camera - code distance

- use multiple images from the camera stream
  → requires blurry image alignment

- other types of blur (defocus blur, upscaling blur)
  → requires different kernel priors

summary

We presented a robust blur removal algorithm for QR code images captured by wearable scanners

- bringing image deblurring to wearables
- exploiting QR code properties
- introducing new initialization scheme for large blur
- PC and Android implementations

We showed promising restoration results and proposed future directions for research.

thank you

references

- [Joshi2008] N. Joshi, R. Szeliski, D. Kriegman – PSF estimation using sharp edge prediction, CVPR, 2008
- [Cho2009] S. Cho, S. Lee – Fast motion deblurring, SIGGRAPH Asia, 2009
- [Krishnan2009] D. Krishnan, R. Fergus – Fast image deconvolution using hyper-Laplacian priors, NIPS, 2009
- [Tai2013] Y.-W. Tai, X. Chen, S. Kim, S. J. Kim, F. Li, J. Yang, J. Yu, Y. Matsushita, M. Brown – Nonlinear camera response functions and image deblurring: Theoretical analysis and practice, PAMI, 2013
- [Sun2013] L. Sun, S. Cho, J. Wang, J. Hays – Edge-based blur kernel estimation using patch priors, ICCP, 2013
- [Pan2013] J. Pan, R. Liu, Z. Su, X. Gu – Kernel estimation from salien structure for robust motion deblurring, Signal Processing: Image Communication, 28, 9, 2013
- [Pan2014] J. Pan, Z. Hu, Z. Su, M.-H. Yang – Deblurring text images via L0-regularized intensity and gradient prior, CVPR, 2014
- [Perrone2014] D. Perrone, P. Favaro – Total variation blind deconvolution: the devil is in the details, CVPR, 2014
- [Xu2010] L. Xu, J. Jia – Two-phase kernel estimation for robust motion deblurring, ECCV, 2010
- [Xu2013] L. Xu, S. Zheng, J. Jia – Unnatural L0 sparse representation for natural image deblurring, CVPR, 2013
- [Zhang2014] Q. Zhang, L. Xu, J. Jia – 100+ times faster weighted median filter (WMF), CVPR, 2014

image sources and links

Reuters - Days numbered for barcodes as shoppers demand more data

http://www.reuters.com/article/2015/08/28/us-retail-consumers-barcodes-insight-idUSKCN0QX0FD20150828

http://i.ytimg.com/vi/30Pjl31cyDY/maxresdefault.jpg

http://static1.1.sqspcdn.com/static/f/458611/17438941/1333367246603/qr-code-shopping-scan-item-and-buy.jpg

http://www.ubimax.de/media/k2/items/cache/7f2cd38b7681e6e2ef83b5a7a5385264_L.jpg

OpenCV – www.opencv.org

FFTW – www.fftw.org

ZBar – www.github.com/zbar