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
• 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 capabilities 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 \([u,v]\):

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

$$M = \begin{bmatrix} <I_x^2> & <I_xI_y> \\ <I_xI_y> & <I_y^2> \end{bmatrix} = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{xy} & C_{yy} \end{bmatrix}$$

$$C_{i,j} = \sum_{(x,y) \in D} w(x,y)I_i(x,y)I_j(x,y)$$

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 $\lambda_1$ and $\lambda_2$ 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 ...

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

- \( m_1 \) 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**

*(difficulties)*

2D

![sharp input](image1)

![blurry input](image2)

2D codes are almost always black and white \(\rightarrow\) convert to HSV color system

1D

![input](image3)

![saturation](image4)

1D text lines look like a 1D code
experiments on ~1000 EAN images

Gaussian Blur Tolerance

Detection rate (%) vs blur sigma (pixels)

- Gallo2011
- Tekin2012
- Katona2013
- Ours

Sigma values:
- Sigma = 0
- Sigma = 3
- Sigma = 7
- Sigma = 13
experiments on ~1000 EAN images

Motion Blur Tolerance

<table>
<thead>
<tr>
<th>blur length (pixels)</th>
<th>detection rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>84.7</td>
</tr>
<tr>
<td>2</td>
<td>85.9</td>
</tr>
<tr>
<td>3</td>
<td>79.7</td>
</tr>
<tr>
<td>4</td>
<td>71.0</td>
</tr>
<tr>
<td>5</td>
<td>61.0</td>
</tr>
<tr>
<td>6</td>
<td>35.4</td>
</tr>
</tbody>
</table>

- Gallo2011
- Tekin2012
- Katona2013
- Ours

length = 0  length = 5  length = 9  length = 15
examples on our images
multiple codes

1D sensitive to blur

2D works well in both cases
## runtime comparison

<table>
<thead>
<tr>
<th>algorithm</th>
<th>Gallo2011</th>
<th>Tekin2012</th>
<th>Katona2013</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>runtime PC (960x720)</td>
<td>27 ms</td>
<td>49 ms †</td>
<td>63 ms</td>
<td>73 ms</td>
</tr>
<tr>
<td>runtime smartphone (640x480)</td>
<td>85 ms</td>
<td>26 ms</td>
<td>173 ms</td>
<td>380/118 ms ‡</td>
</tr>
</tbody>
</table>

† also including image format conversions
‡ CPU/GPU
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