Luxapose: Indoor Positioning with Mobile Phones and Visible Light

Ye-Sheng Kuo, Pat Pannuto,
Ko-Jen Hsiao, and Prabal Dutta

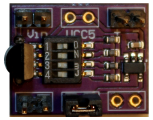
University of Michigan

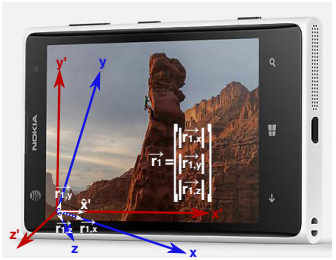
What is Luxapose?

10101010

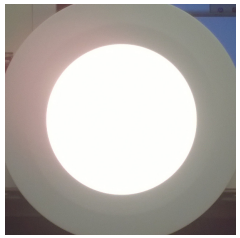
10001000

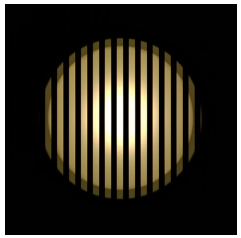
10000000

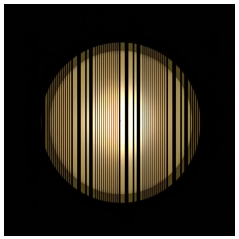




Smart Phone







Captured using a rolling shutter

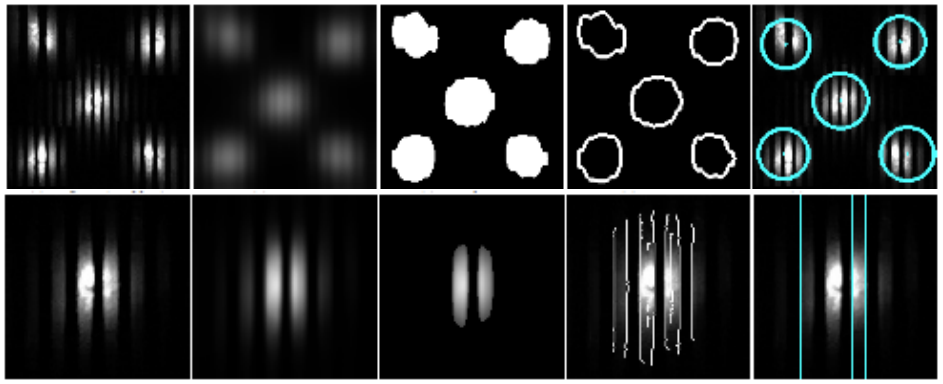
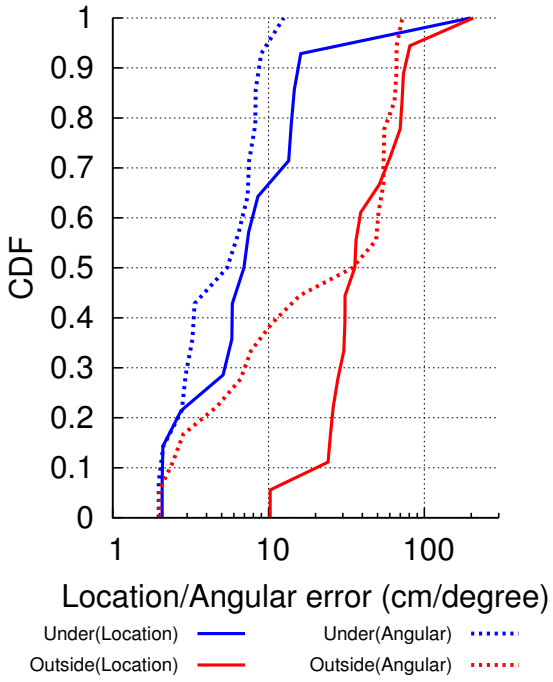


Image processing extracts beacon locations and frequencies



$$d_{01}^2 = (K_0 a_0 - K_1 a_1)^2 + (K_0 b_0 - K_1 b_1)^2 + Z_f^2 (K_0 - K_1)^2$$

Compute

$$d_{01}^2 = (x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2$$

$$\sum_{m=1}^{N} \left\{ (T_x - x_m)^2 + (T_y - y_m)^2 + (T_z - z_m)^2 - K_m^2 \left(a_m^2 + b_m^2 + Z_f^2 \right) \right\}^2$$

Minimize

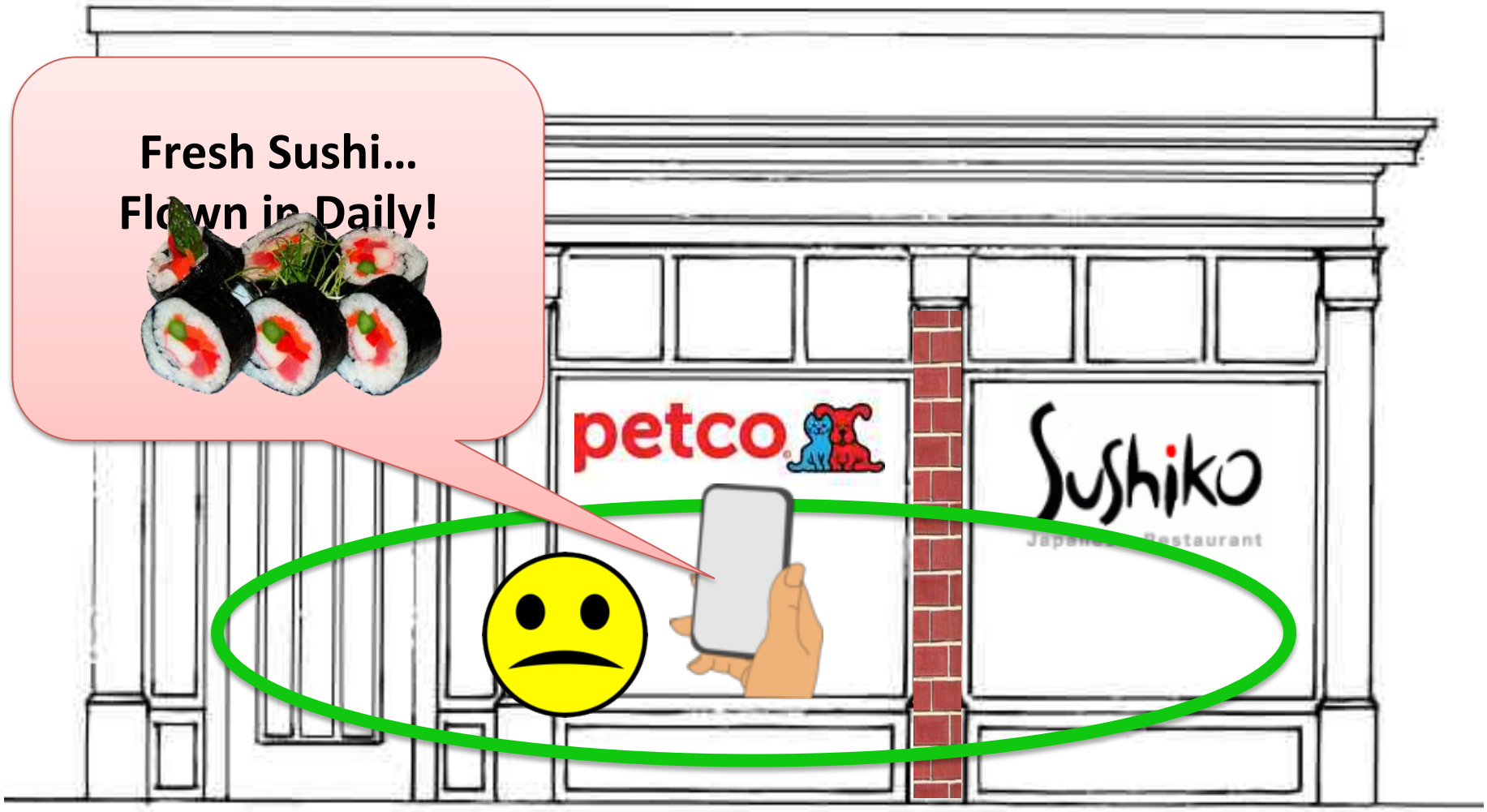
Existing RF-based techniques

- Why are RF-based techniques not good enough?

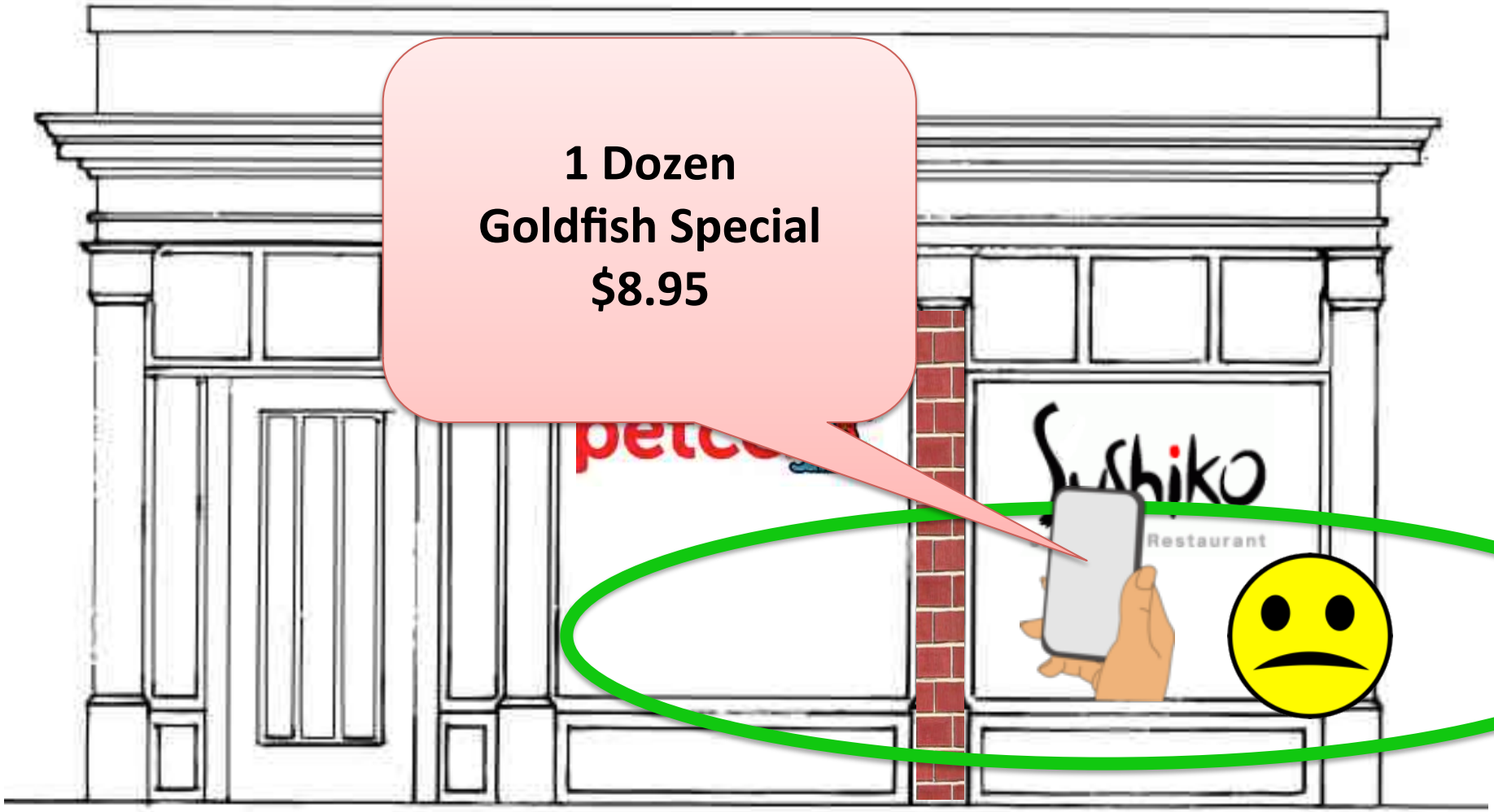
Range-based				
Technique	**Accuracy**	**Infrastructure**	**RX complexity**	**Orientation**
RSS	Low	Low	Low	No
ToA	High	High	Low	No
TDoA	High	High	Low	No
AoA	High	High	High	No
Range-free				
	Low	High	Low	No

- Moreover…

Barrier problem

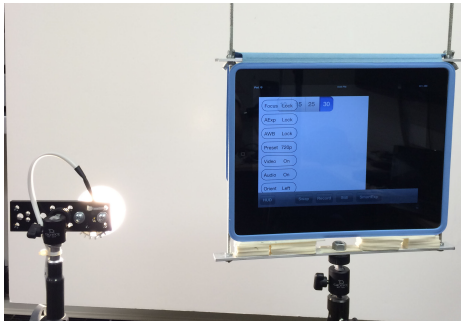


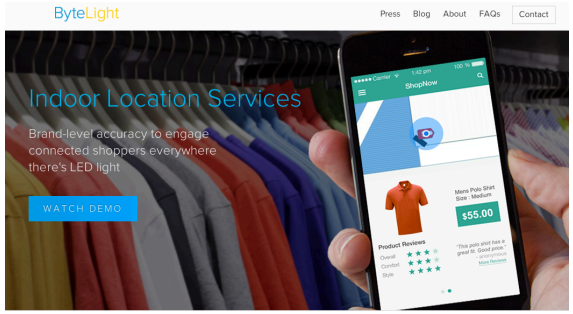
Barrier problem



Using VLC for positioning

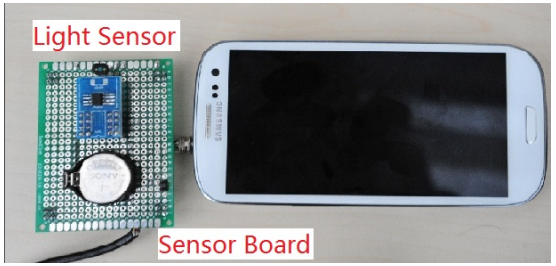
- Semantic localization
  - Room-level accuracy





  - Visual Light Landmark for mobile device (IPSN '14)
  - Bytelight

- Received Signal Strength (RSS)
  - Additional hardware

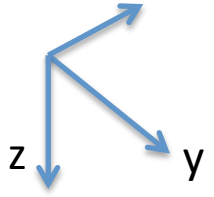


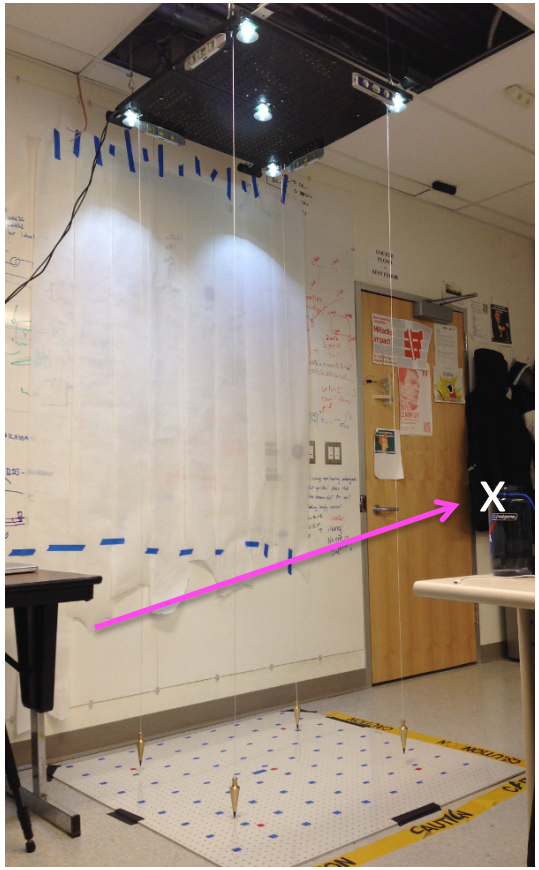
  - Epsilon (NSDI '14)

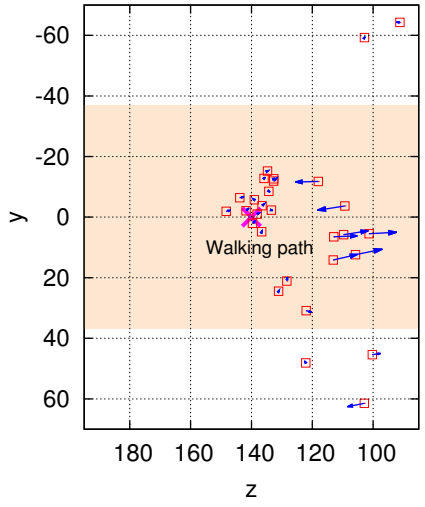
Emerging retail environment

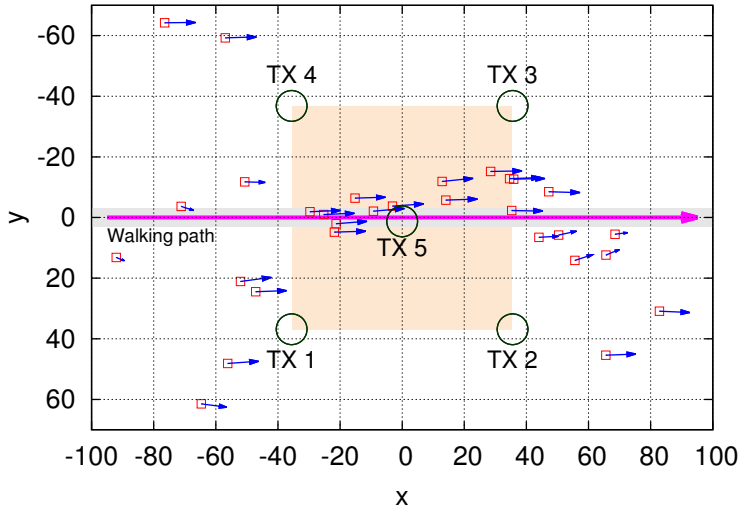
- Often have line-of-sight to lighting
  - Groceries
  - Drugstores
  - Megastores
  - Hardware stores
  - Enterprise settings
- Lots of overhead lighting in retail
- Retailers deploying LED lighting
- Customers using phones in stores
  - Surf, Scan, Share
- Customers installing retailer apps
  - Maps, Barcodes, Deals, Shopping

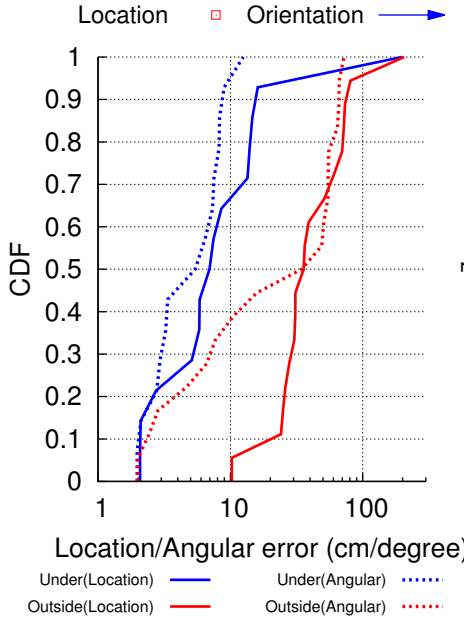
Localization accuracy using our testbed

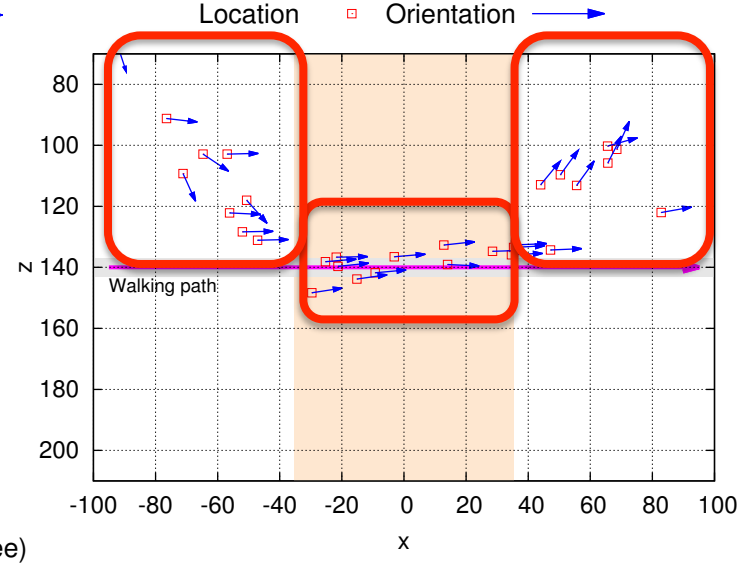








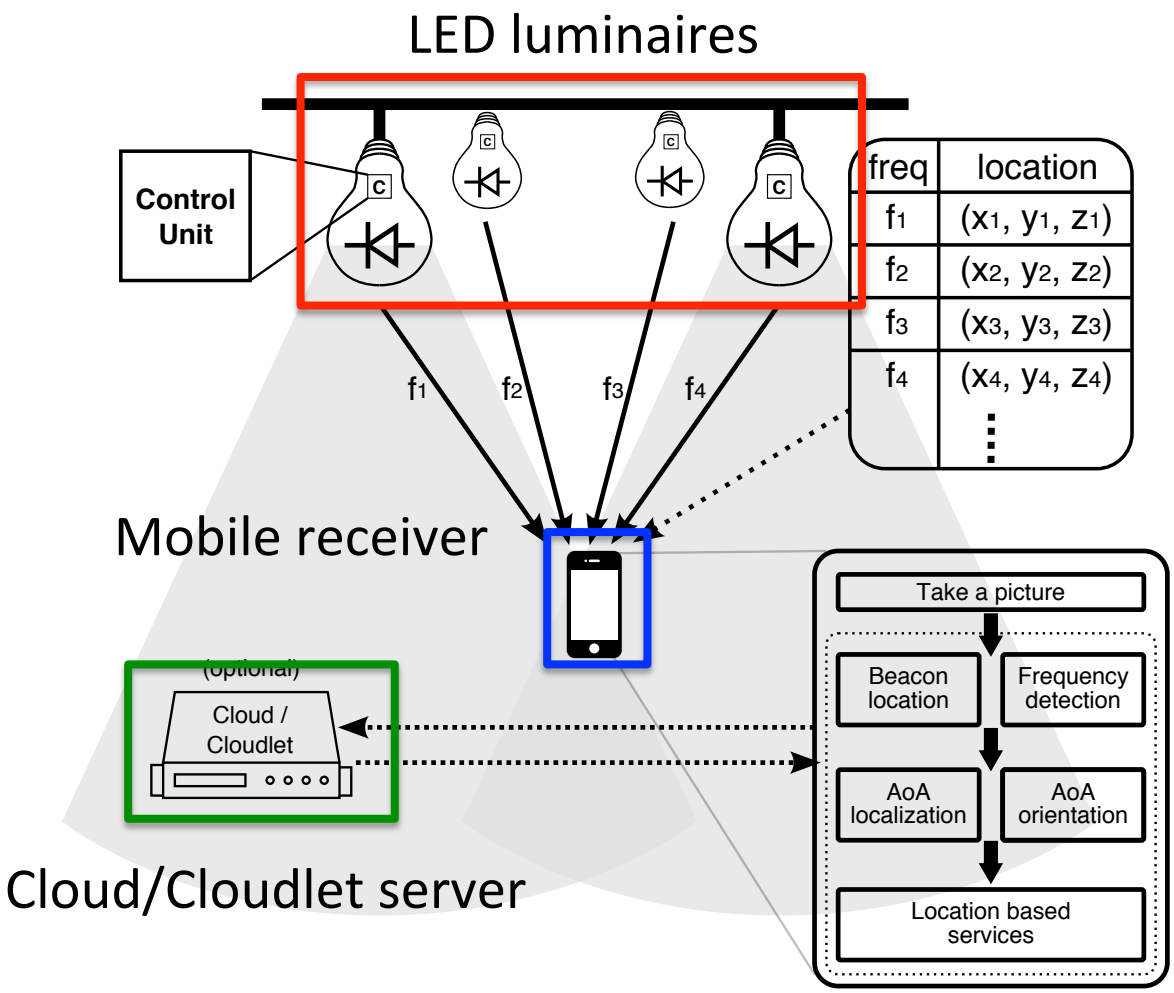




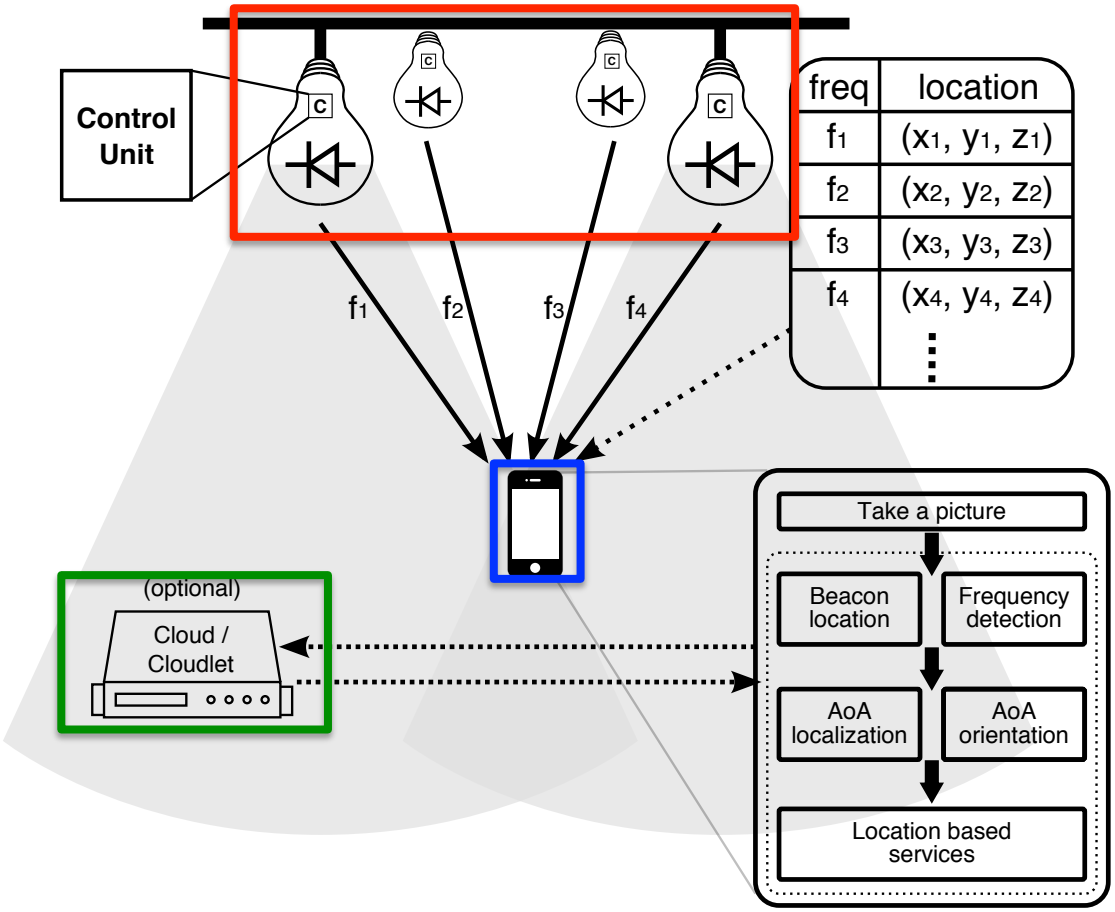
Outline

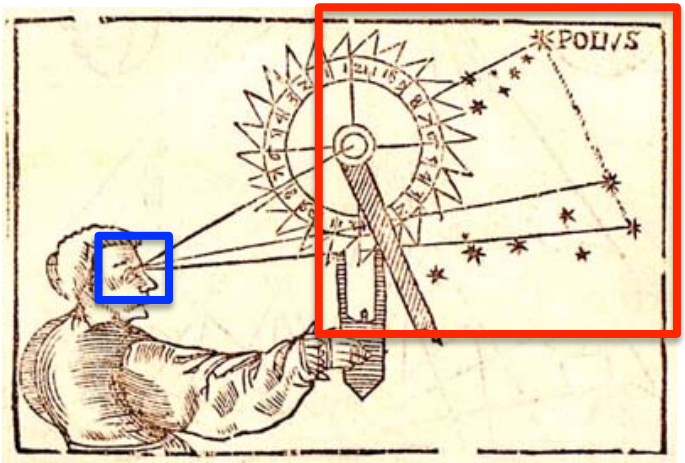
- Introduction
- System Architecture
- Localization Principle
- Implementation / Evaluation
- Conclusion

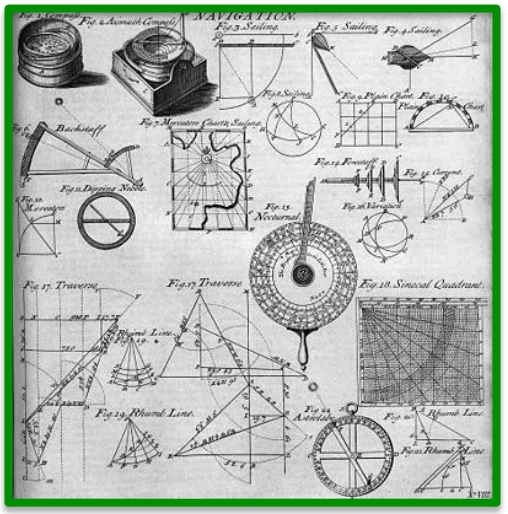
Luxapose system architecture



Architectural analogy







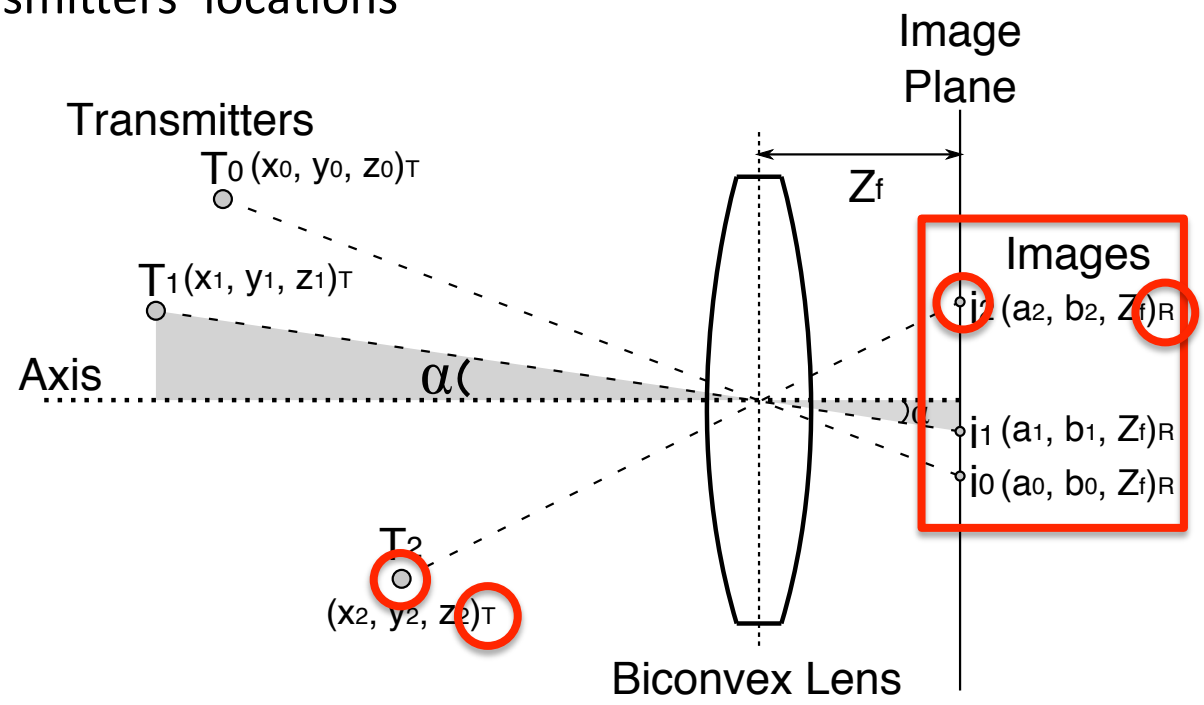
http://www.mhs.ox.ac.uk/students/98to99/Inst/Instpgs/3.tinyNOCTURNAL.html
http://www.unlikelyboatbuilder.com/2010/07/celestial-navigation-fun.html

Outline

- Introduction
- System Architecture
- Localization Principle
- Implementation / Evaluation
- Conclusion

Localization principle

- Known:
  - Transmitters' locations



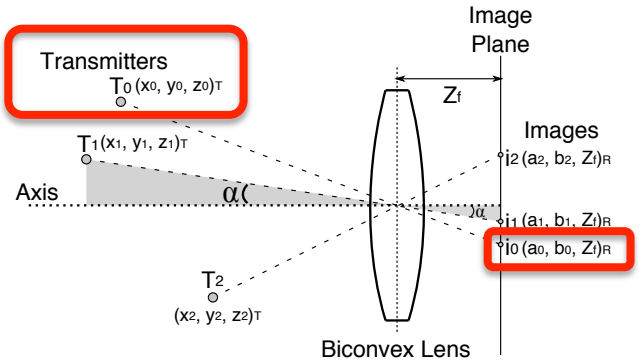
- Goal:
  - Find receiver's location in Transmitters' frame of reference

Find scaling factors for all transmitters

K_0 : Scaling factor converts pixel to distance

- Transmitters' location in receivers' frame of reference

$$T_0 = (x_0, y_0, z_0)_T \leftrightarrow (K_0 a_0, K_0 b_0, K_0 Z_f)_R$$



- Find all scaling factors by pairwise distance

$$d_{01}^2 = (K_0 a_0 - K_1 a_1)^2 + (K_0 b_0 - K_1 b_1)^2 + Z_f^2 (K_0 - K_1)^2$$

$$d_{01}^2 = (x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2$$

Find receiver's location and orientation

- Pairwise distance between receiver and transmitters

$$(T_x - x_m)^2 + (T_y - y_m)^2 + (T_z - z_m)^2 = K_m^2(a_m^2 + b_m^2 + Z_f^2)$$

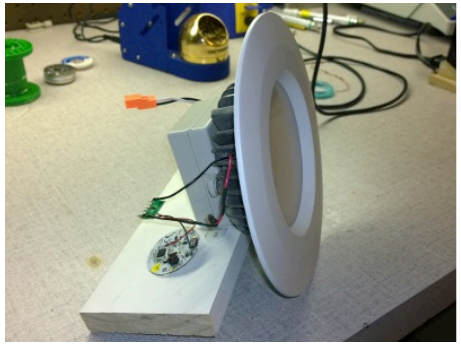
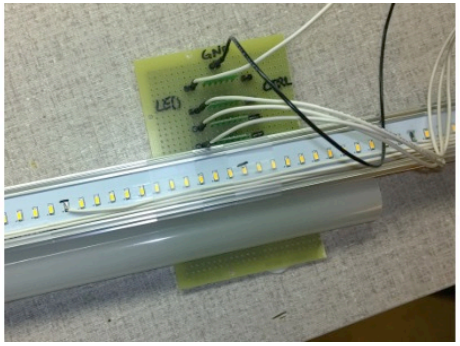
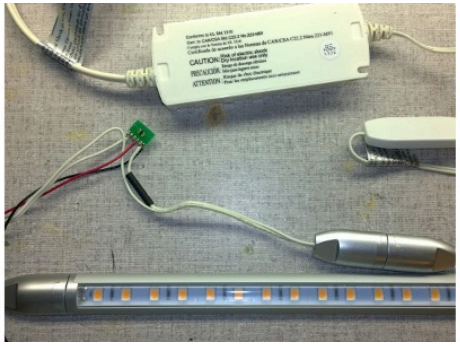
- Transmitters' frame ←→Receiver's frame

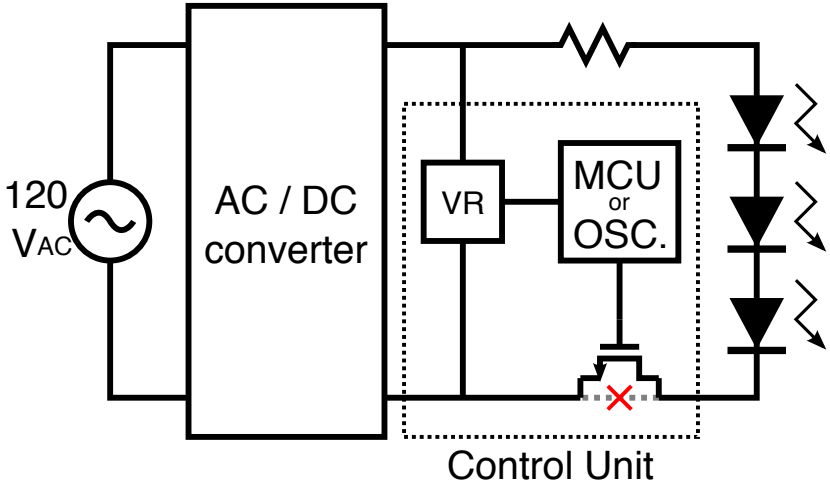
$$\begin{bmatrix} x_0 & x_1 & \cdots & x_{N-1} \\ y_0 & y_1 & \cdots & y_{N-1} \\ z_0 & z_1 & \cdots & z_{N-1} \end{bmatrix} = R \times \begin{bmatrix} K_0 a_0 & K_1 a_1 & \cdots & K_{N-1} a_{N-1} \\ K_0 b_0 & K_1 b_1 & \cdots & K_{N-1} b_{N-1} \\ K_0 Z_f & K_1 Z_f & \cdots & K_{N-1} Z_f \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

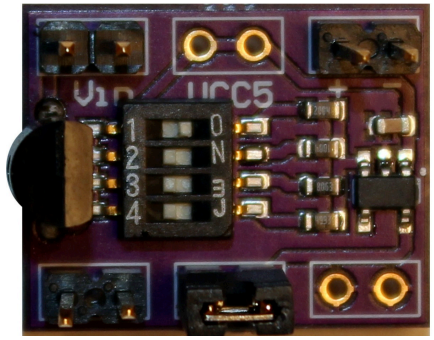
Outline

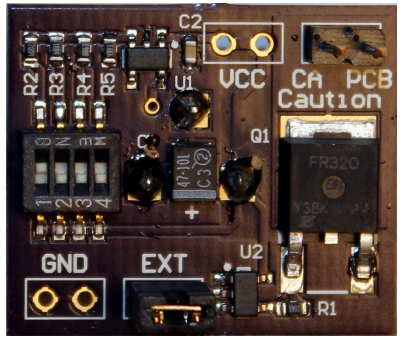
- Introduction
- System Architecture
- Localization Principle
- Implementation / Evaluation
- Conclusion

LED luminaires

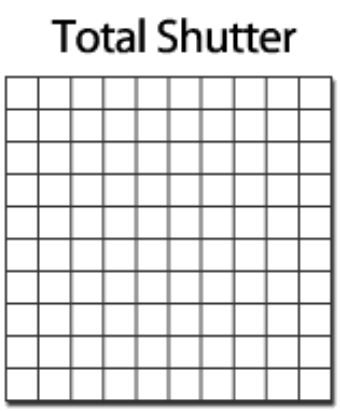



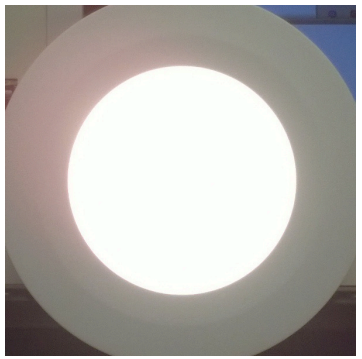




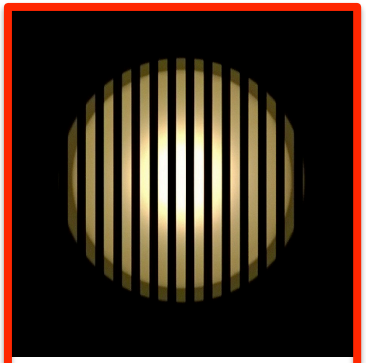
$3 (1K quantity)
~5 mA

Mobile receiver – CMOS rolling shutter

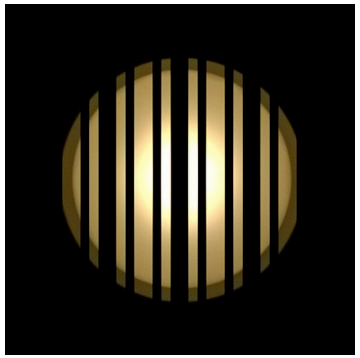




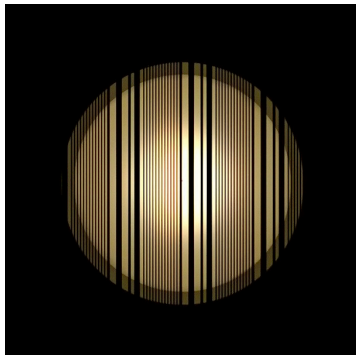
Human eyes



Pure tone



Manchester

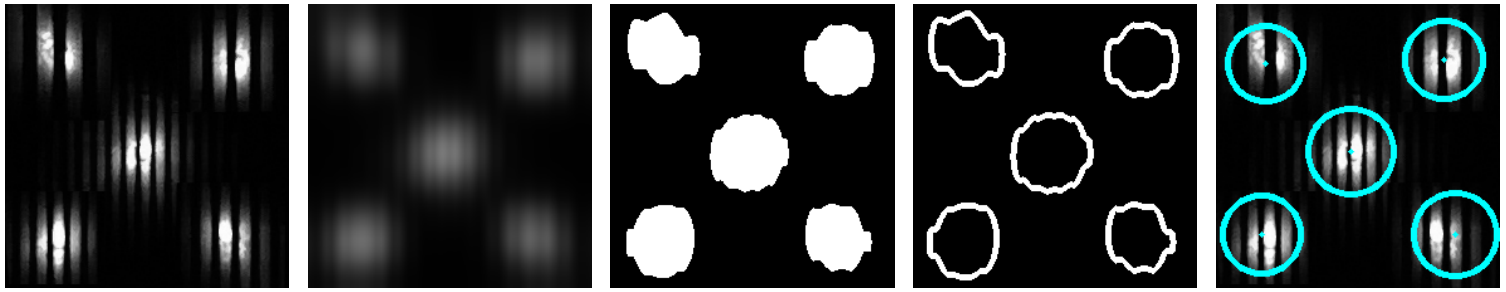


Manchester w/ idle pattern

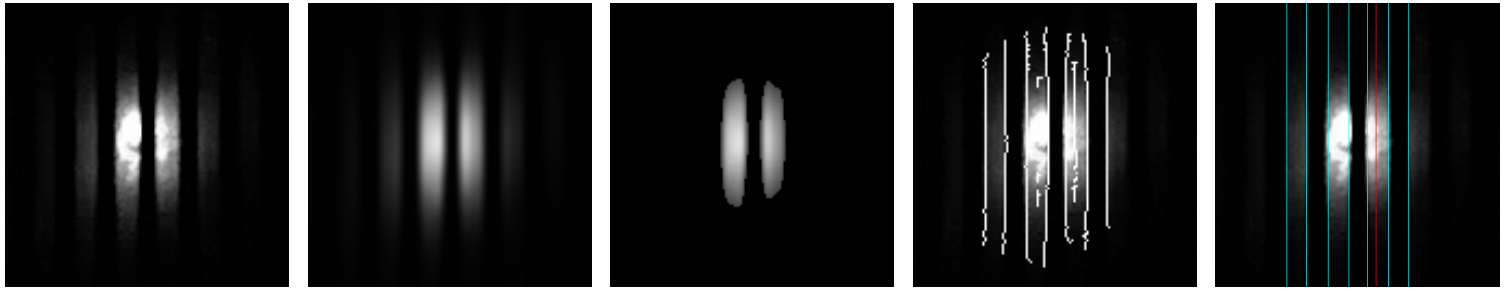
http://www.diyphotography.net/everything-you-wanted-to-know-about-rolling-shutter/

Cloudlet – Image processing

- Identify centroid and frequency



Original image | Blurred | Thresholding | Contours | Center

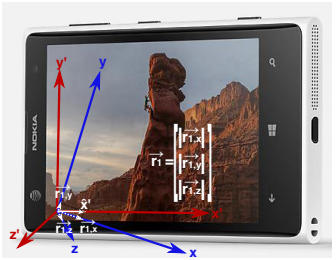


Original image | Vertical blurred | Thresholding | Edge finding | Frequency

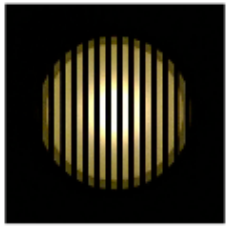
Putting it all together

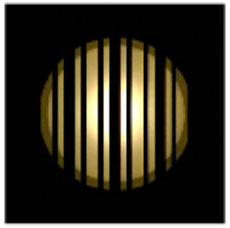
01100101000

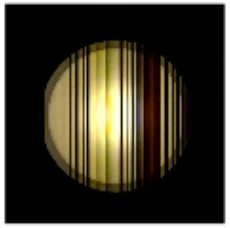
LED Luminaire



Smart Phone

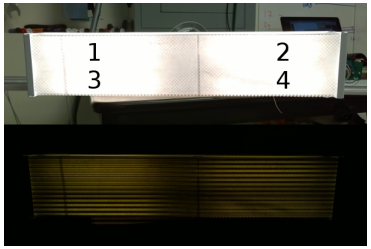


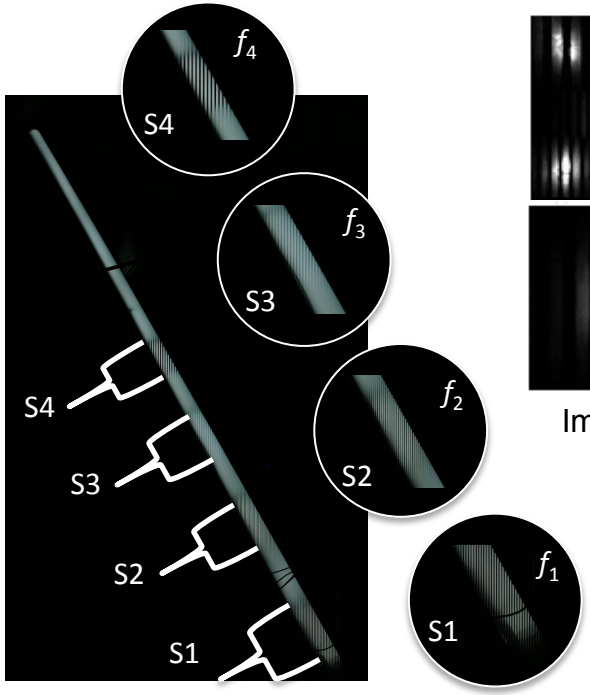




Illuminate Idle TX <66> TX packet

Captured using a rolling shutter





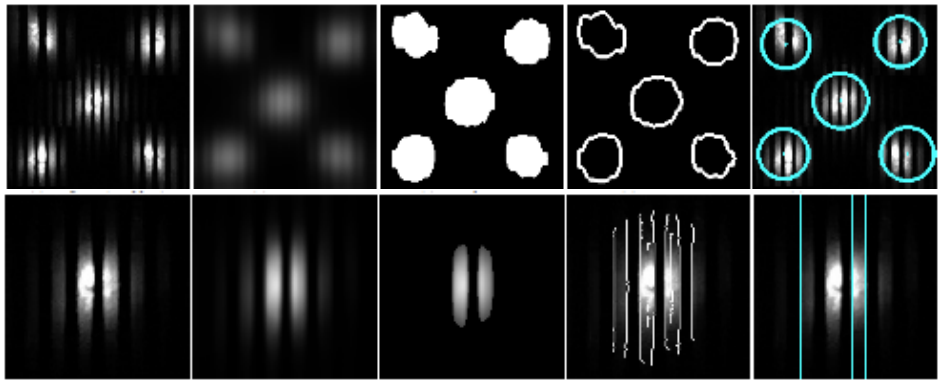


Image processing extracts beacon locations and frequencies

$$d_{01}^2 = (K_0 a_0 - K_1 a_1)^2 + (K_0 b_0 - K_1 b_1)^2 + Z_f^2 (K_0 - K_1)^2$$

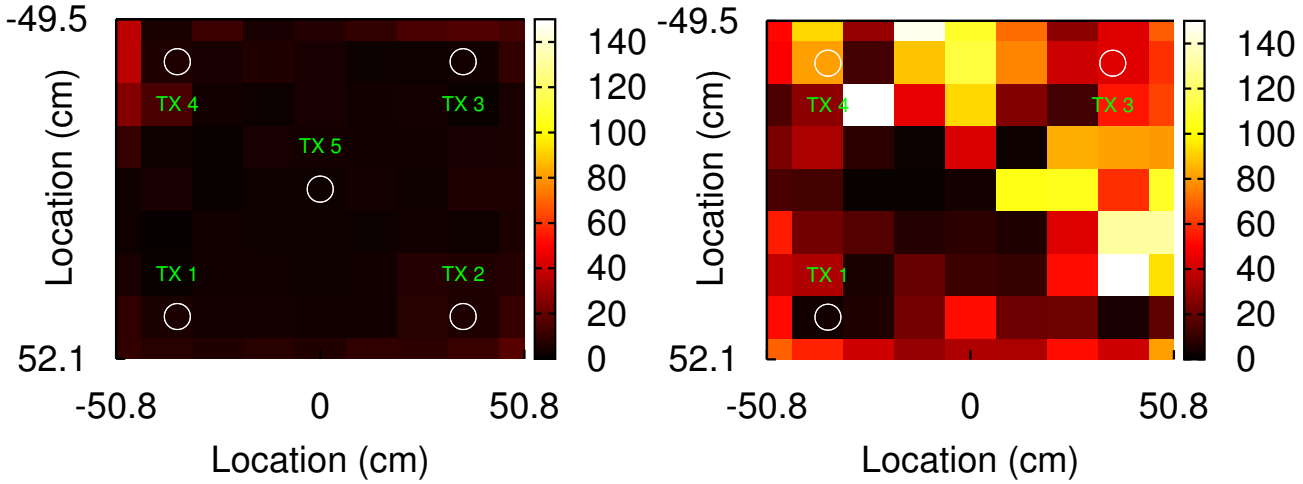
Compute

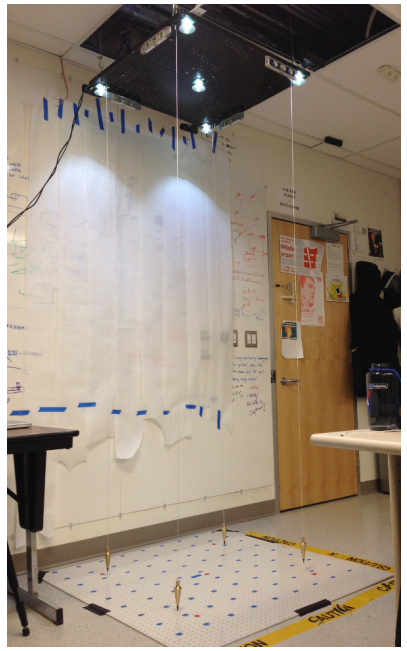
$$d_{01}^2 = (x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2$$

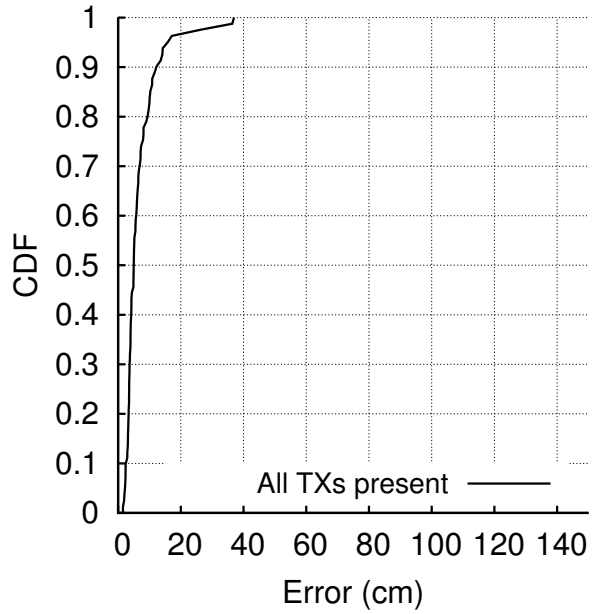
$$\sum_{m=1}^{N} \left\{ (T_x - x_m)^2 + (T_y - y_m)^2 + (T_z - z_m)^2 - K_m^2 \left(a_m^2 + b_m^2 + Z_f^2 \right) \right\}^2$$

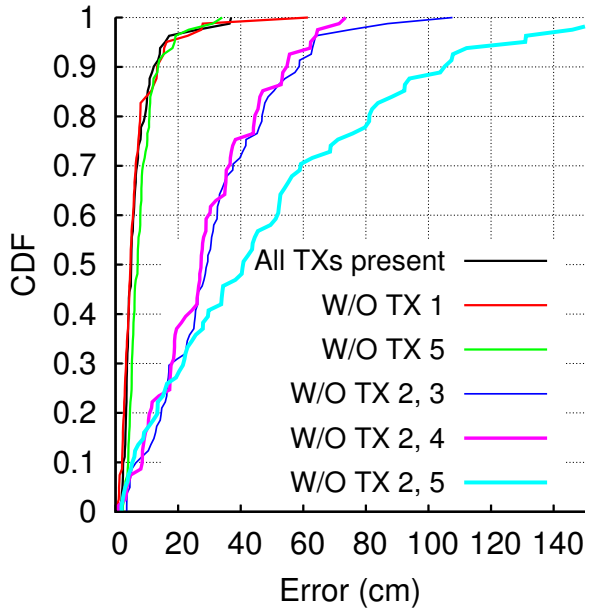
Minimize

Decimeter location (3D) accuracy

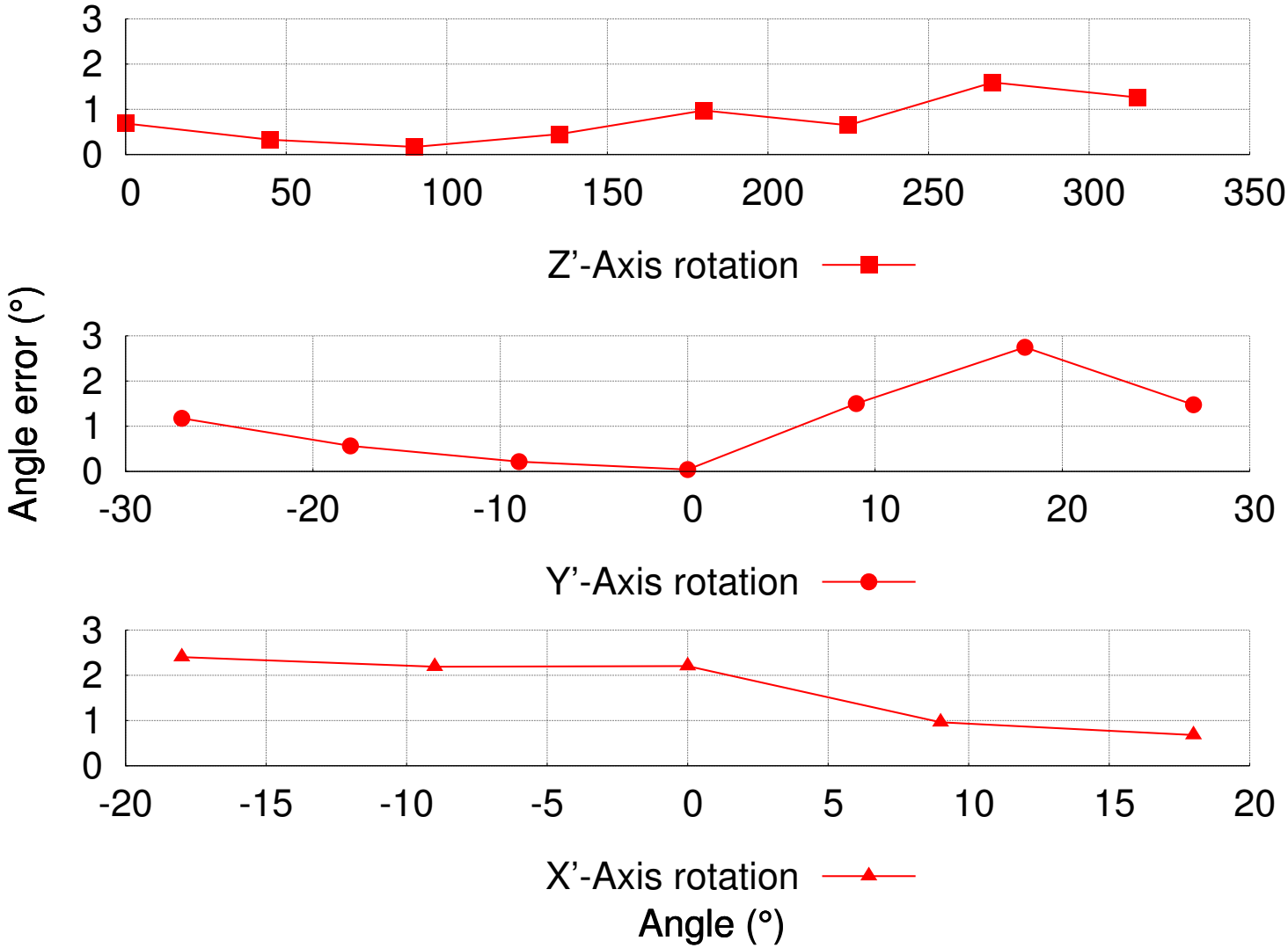




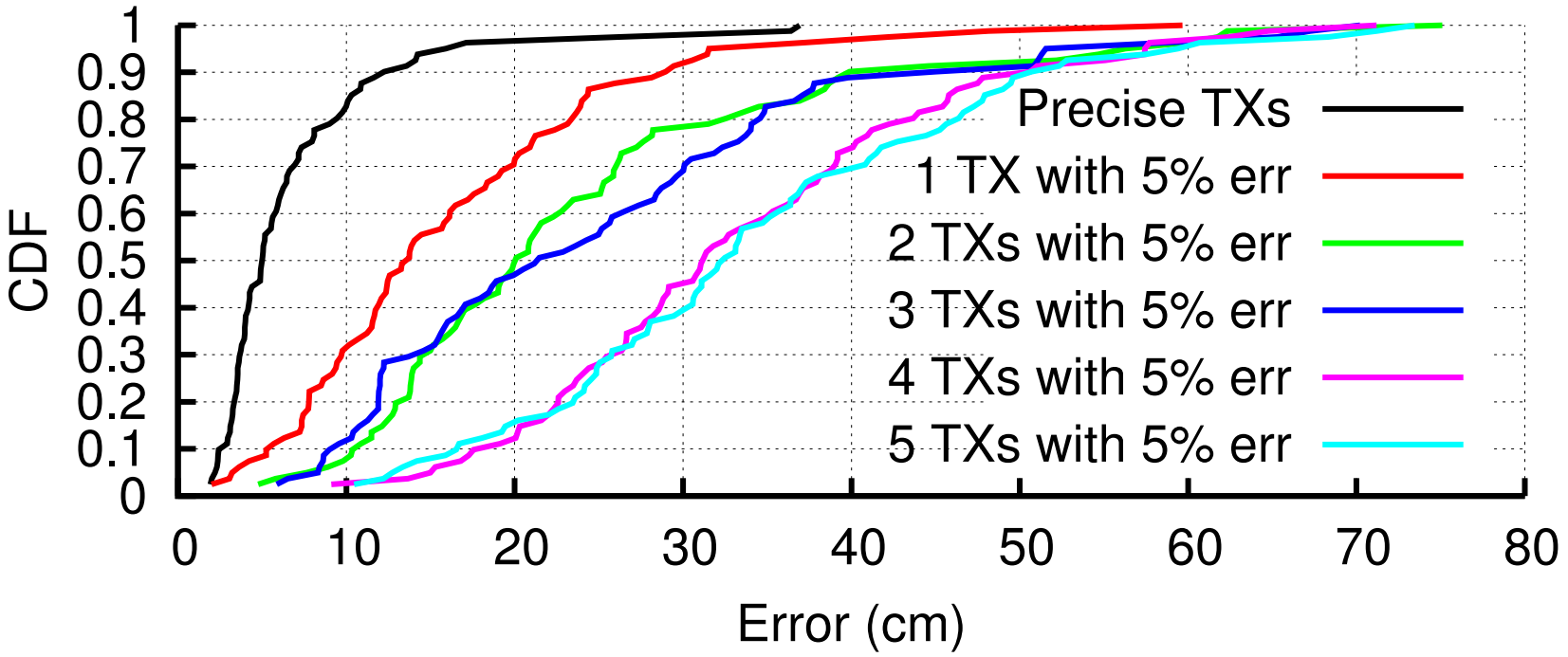




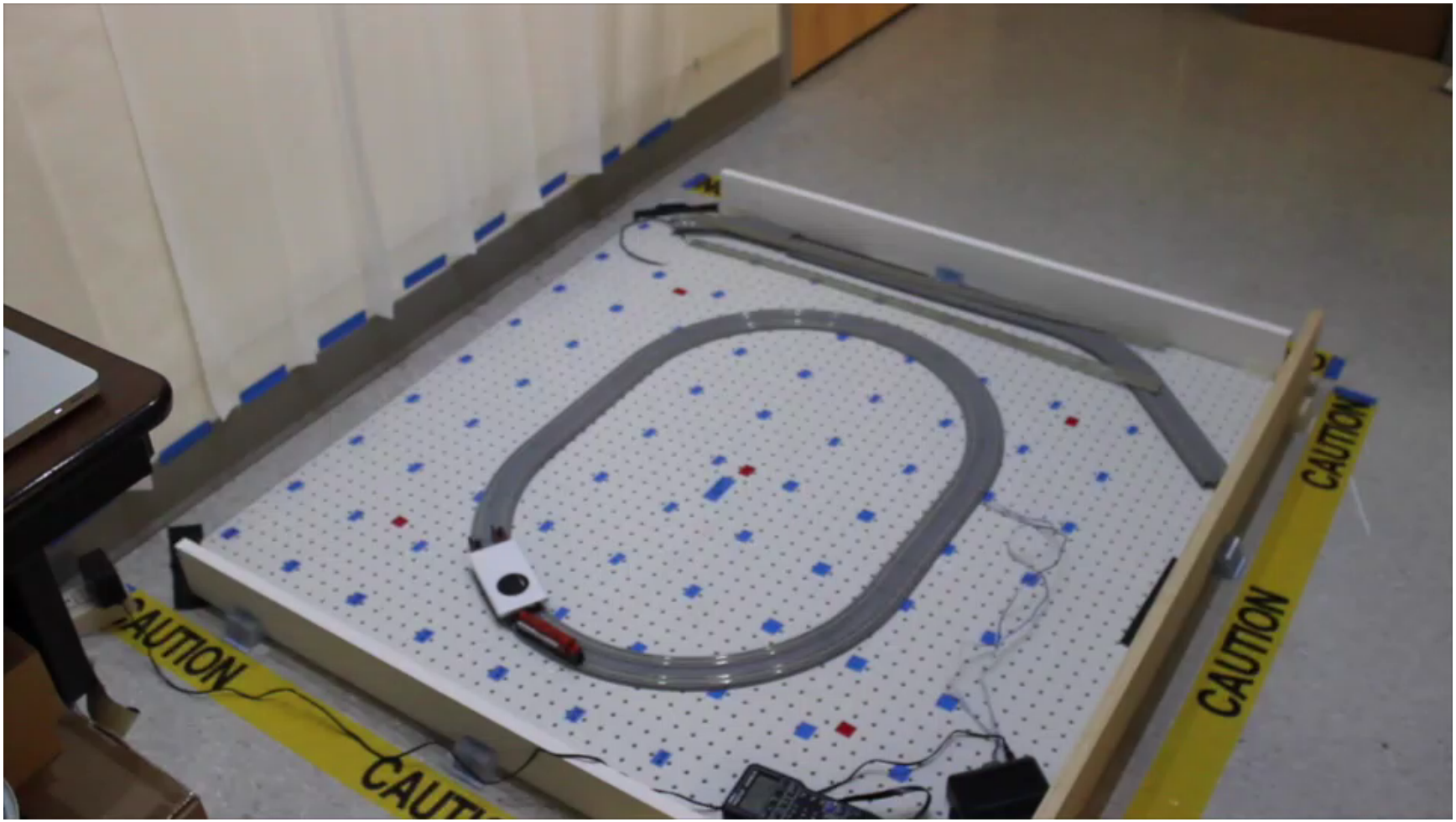
High orientation accuracy



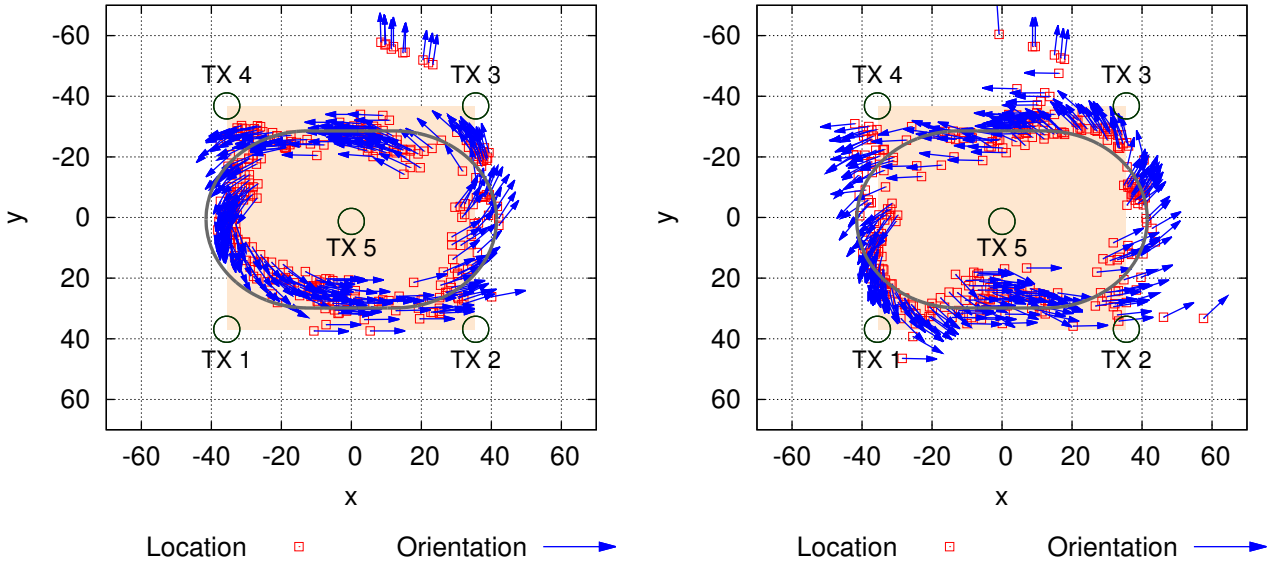
Accuracy is affected by installation error



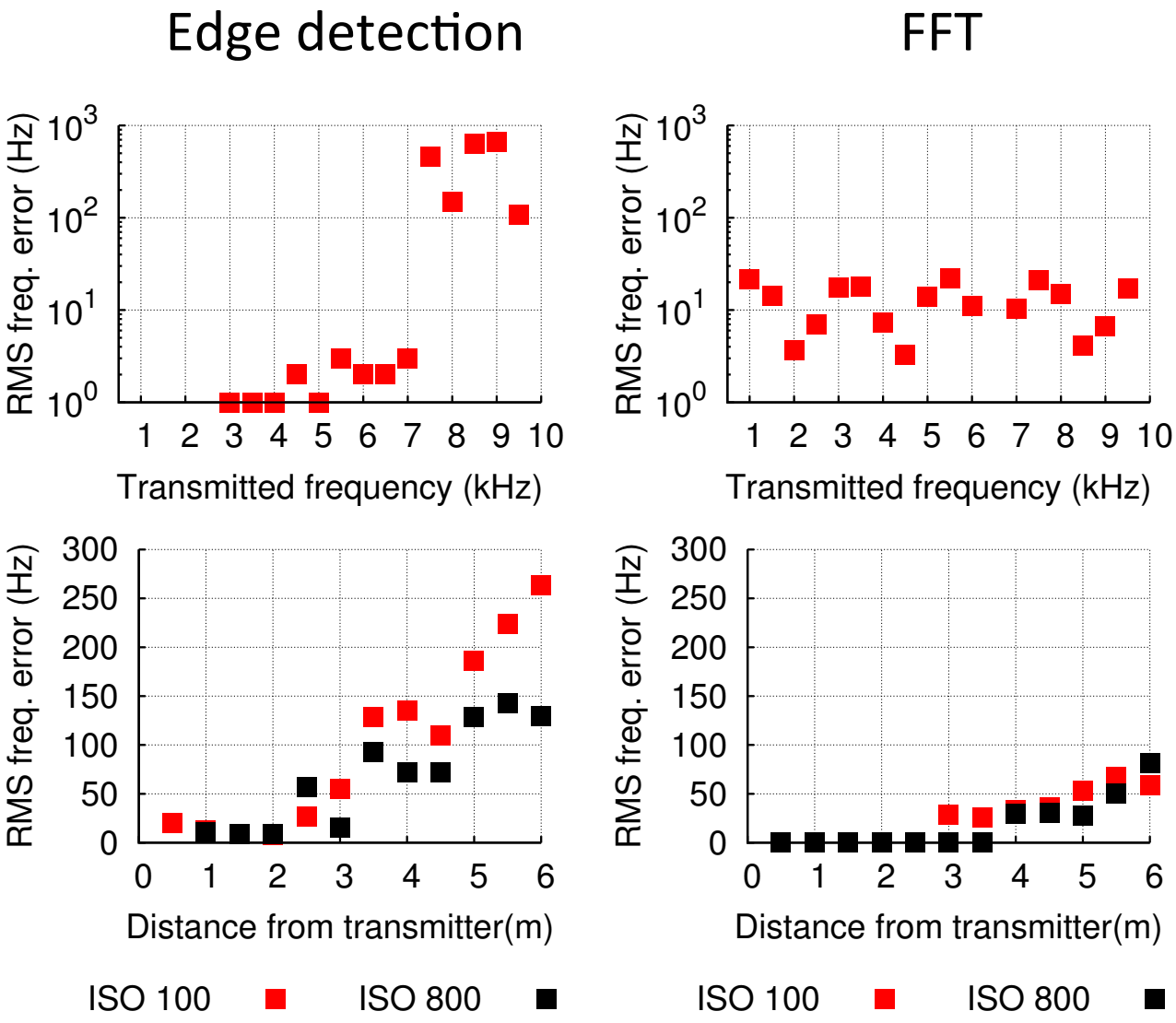
Locate a model train



Accuracy is affected by motion



Frequency recovery → #channels

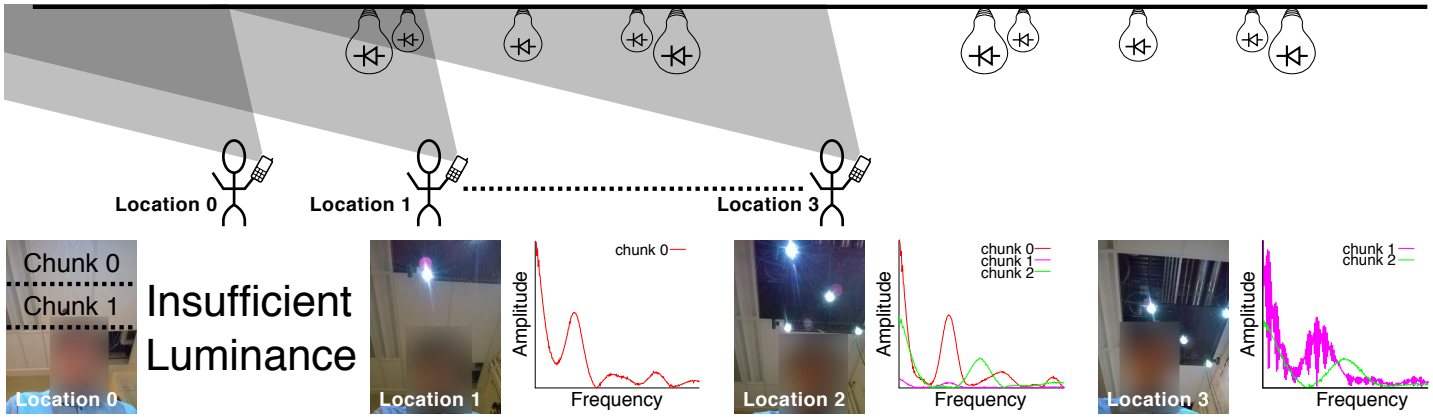


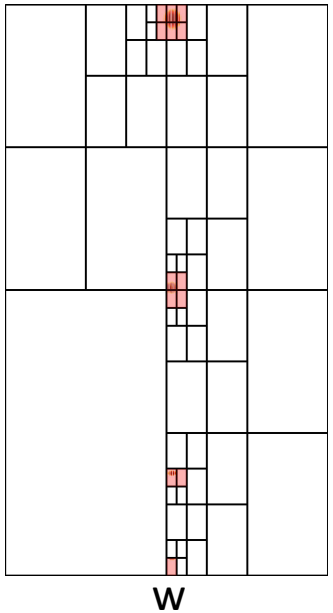
Limitations

- Minimum #pixels required
  - Distance / TX size / #pixels on imager
- >= 3 transmitters in field of view
- Requires known anchor locations
- Requires 9 s to complete a localization process
  - 4.46 s (taking picture)
  - 3.41 s (uploading image)
  - 0.3 s (extracting/labeling transmitters)
  - 0.8 s (performing calculation)

Optimization – perform local processing?

- Current system takes 3.41 s to upload an image (33 M pixels)





$$\frac{13}{1024} \times 33M$$

$$O(N \times \log(N)) \rightarrow K \times \left[{}^{W}\!/_{K} \times \log({}^{W}\!/_{K}) \right] = W \times \log({}^{W}\!/_{K})$$

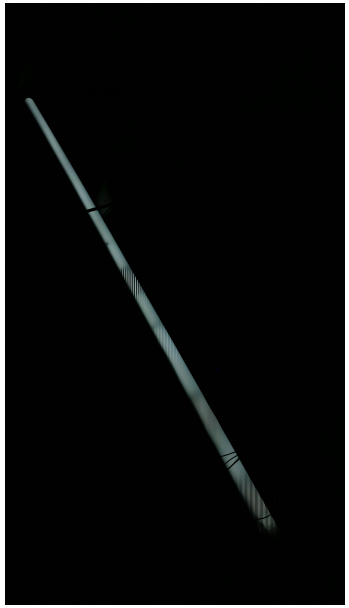
W: width of image
K: #chunks

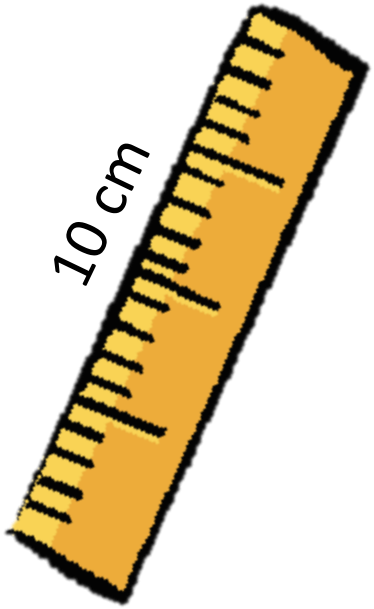
Future work

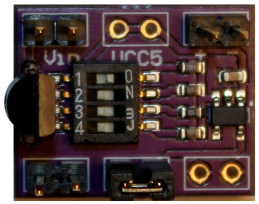
- Optimization
  - Algorithm
  - Image processing
  - Local processing
- Porting Luxapose to iOS/Android
- Integrating with iBeacon

Conclusion

- Luxapose:
  - Angle of arrival indoor positioning w/ single image
  - Decimeter accuracy
  - Orientation information
  - Slightly modified LED luminaires
  - Unmodified mobile phone

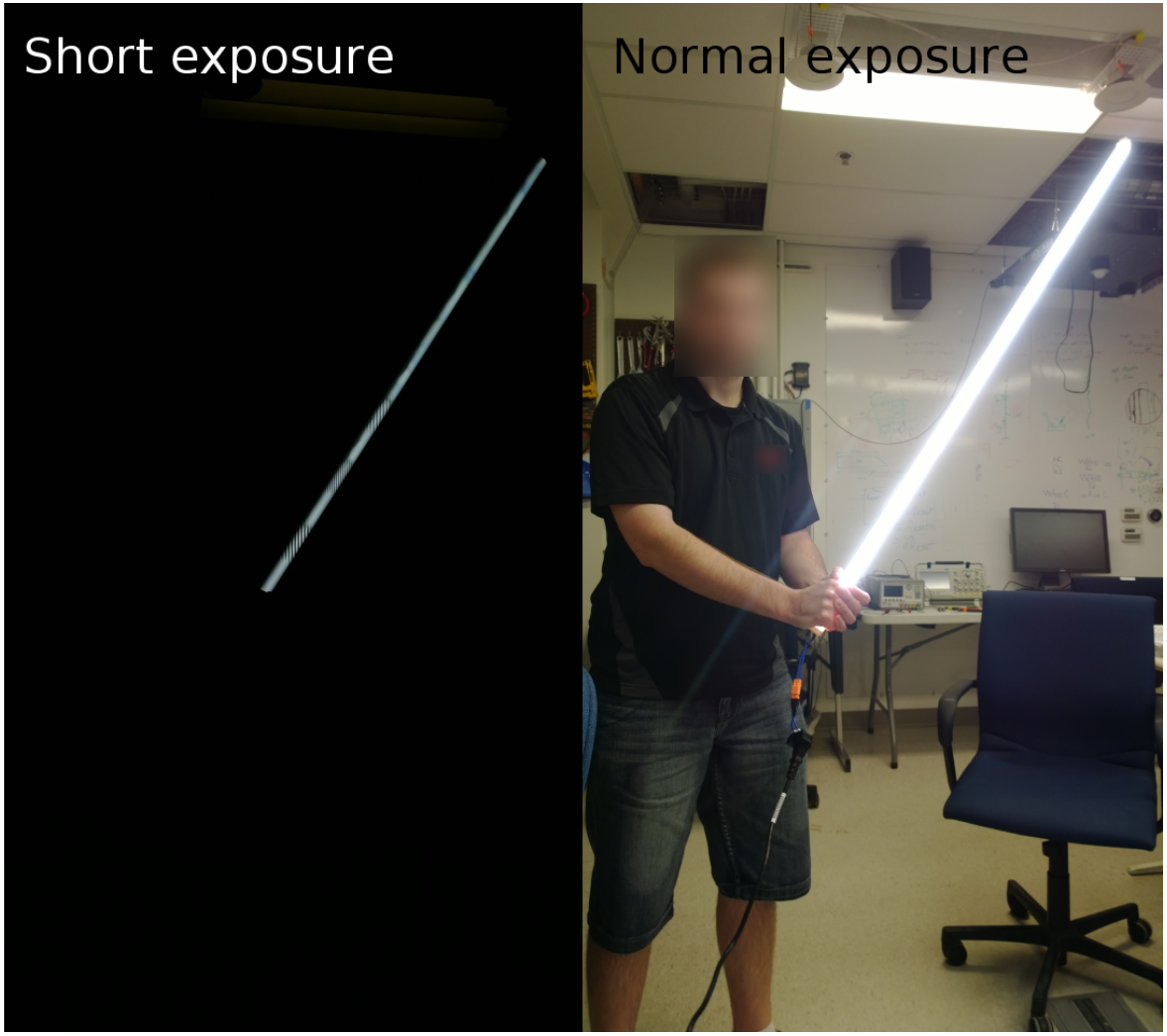






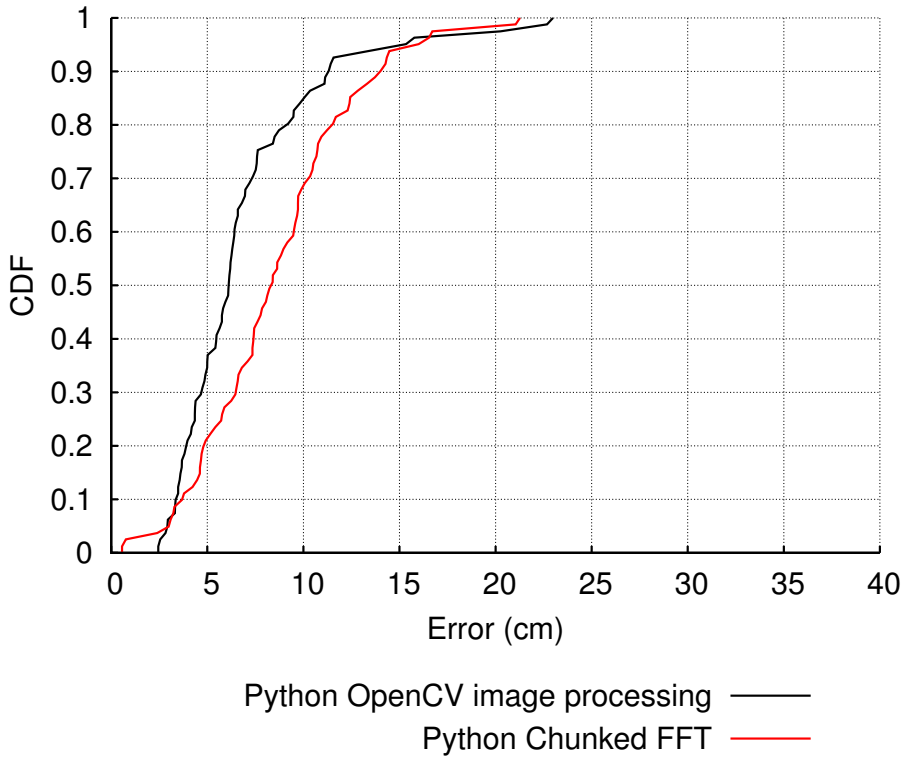
Backup

Privacy is not compromised



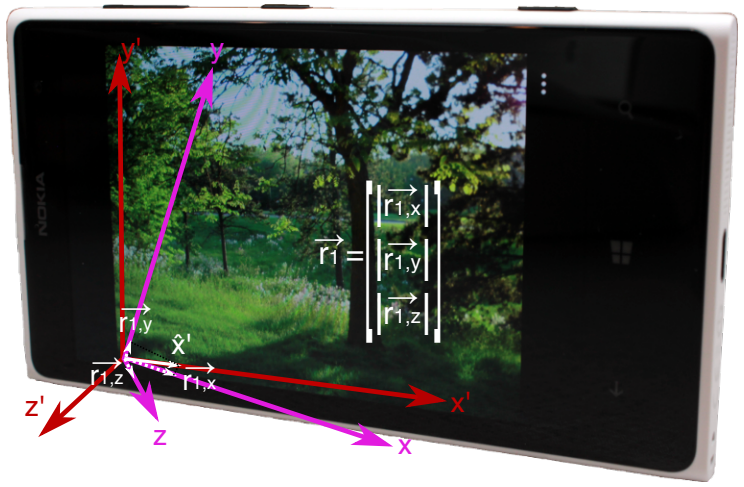
Chunked-FFT reduces accuracy

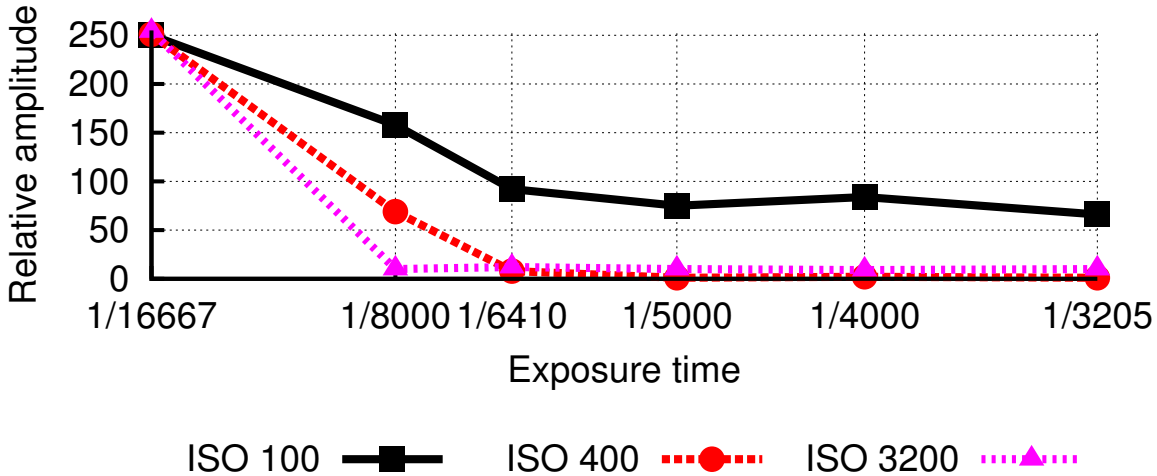
- 81 images (33 M)
- Python Open CV image processing
  - 79 s → 67 s (15% time reduction)



Mobile receiver + mobile app

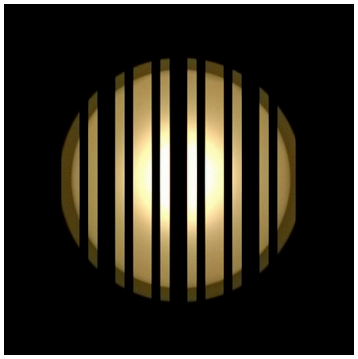
- Requires phone OS API support
  - Exposure control
  - Film speed (ISO)
  - Windows phone ✓
  - Android ✗
  - iOS 8 ✓

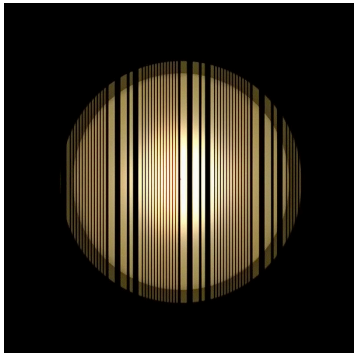


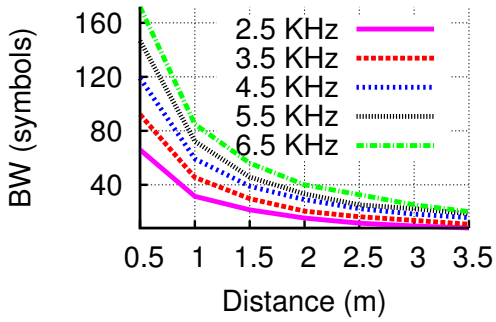


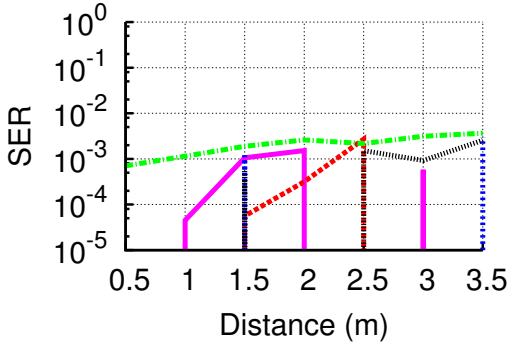
Increase #channels?

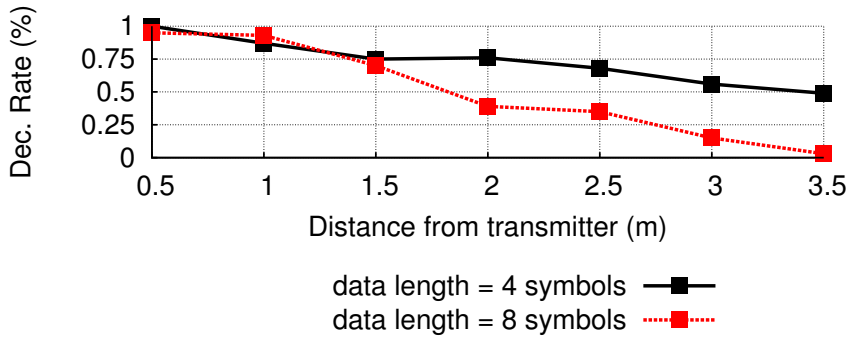
- Encoding data in a single frame











Google glass

- Manually break exposure control loop
- Imager resolution