Signal Detection 2012 - 2015
The problem

- Determine discrete states
- Confidence estimate

visibility history:
# of successful consecutive detections
The problem:

- Determine discrete states
- Confidence estimate
- Blinking lights
2012: Simple classifier with colormap

- LUT: $256^3$ bits in RGB cube
- Problem: How to train?
2012: Simple classifier with colormap

- LUT: $256^3$ bits in RGB cube
- Problem: How to train?
2012: Simple classifier with colormap

- No blink detection
- Hard to train/tune
- Unstable tool support
  → Rudimentary solution, discarded
2013: Laser-cluster detection
2013: Laser-cluster detection

Complicated and noisy mechanics
2013: Laser-cluster detection

- Algorithmically simple
- Relatively little tuning required
- Error sources:
  - Mechanics noise
  - Clustering noise
  - Bright spots from outside light reflections

→ Not reliable enough
≥ 2014: „Similarity“ color model

The YUV color space

![Diagram of YUV color space with Y values 0, 86, 172, and 255]
≥ 2014: „Similarity“ color model
bool is_similar(u, v, u\text{ref}, v\text{ref}, \text{sat}\text{ref}, \text{thresh}_{\text{chroma}}, \text{thresh}_{\text{sat}}) {
  \text{sat} = \sqrt{u^2 + v^2}
  d_u = u_{\text{ref}} \cdot \text{sat} - u \cdot \text{sat}_{\text{ref}}
  d_v = v_{\text{ref}} \cdot \text{sat} - v \cdot \text{sat}_{\text{ref}}
  d^2 = d_u^2 + d_v^2
  \text{thresh} = \text{sat} \cdot \text{sat}_{\text{ref}}
  \text{return } (\text{sat} > \text{thresh}_{\text{sat}}) \land (d^2 \cdot \text{thresh}_{\text{chroma}} < \text{thresh}^2)
}

≥ 2014: „Similarity“ color model
The classifier

`neighborhood_min_match = 8`
The classifier

neighborhood_min_match = 8
neighborhood_min_match = 8
The classifier

neighborhood_min_match = 8
The classifier

neighborhood_min_match = 8

neighborhood matches = 7
The classifier

neighborhood_min_match = 8
The classifier

\[
\text{neighborhood\_min\_match} = 8
\]

\[
\text{neighborhood matches} = 9
\]
The classifier

neighborhood_min_match = 8
The classifier

`neighborhood_min_match = 8`

`neighborhood_matches = 11`
The classifier

neighborhood_min_match = 8
The classifier

neighborhood_min_match = 8
The classifier

neighborhood_min_match = 8

neighborhood_matches = 2
The classifier

neighborhood_min_match = 8
neighborhood_min_match = 8

neighborhood_matches = 6
The classifier

neighborhood_min_match = 8
The classifier

neighborhood_min_match = 8

# of points:
16
Signal assembly
Signal assembly
First Results

- „Pure“ color model: 10 LoC
- Initial implementation: ~ 300 LoC
  - Current implementation: > 2200 LoC
- 20% of the work solve 80% of the problem
- do 80% of the work to solve the other 20%
The difficulties
The difficulties
The difficulties
The difficulties
The difficulties
A solution: More data!

Complicated and noisy mechanics
MPS detection
The complete pipeline

- Laser ROI
- Raw image
- Classifier with color model
- Signal assembly
- History mapping
- Tuning image
- Select „best“ signals
- Shannon/Weaver Blink detection
- Brightness classifier
Results with laser-line MPS detection
Code complexity: Geometric heuristics
Code complexity: Geometric heuristics
Upsides / Downsides

- *almost* no false positives
- Live tuning
- High CPU load
- Complex code
- Laser requires mechanic precision/calibration
How to do it better?

- Choose a better classifier (Histogram…?)
  - But: Smarter classifier requires smarter training
  - 80/20 Rule will probably still apply
- Make better design decisions
  - Think about 80/20 Rule at design stage
  - E.g. don't hardcode search heuristics
Lessons learned

- 
- 
-