

Real time video annotations for augmented reality

Ed Rosten, Dr. Gerhard Reitmayr,
Dr. Tom Drummond



UNIVERSITY OF
CAMBRIDGE

Label Placement



- Screen stabilized labels
- Object labels (nearby with follower lines)

Label placement in video

Place labels in uninteresting parts of the image

- Screen stabilised
 - Sports broadcasts / subtitles
 - Screen edges / corners deemed uninteresting
- Modelled world (Bell *et. al.*)
 - Uninteresting areas explicitly modelled and tracked
- Non temporal (Thanedar and Höllerer)
 - Motion / uniformity of color determines interest
 - Optimized over many frames

High speed and causal necessary

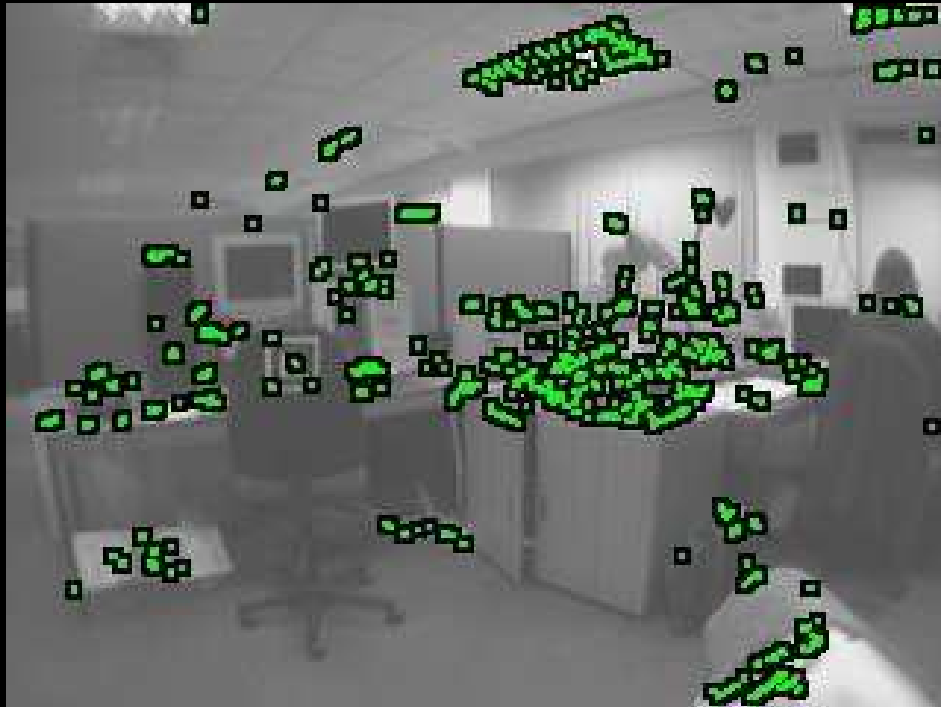
- Real time required for live video
- 30ms per frame available
- Limited CPU on portable devices
- AR application requires CPU
- **High speed operation necessary**

Image of interest



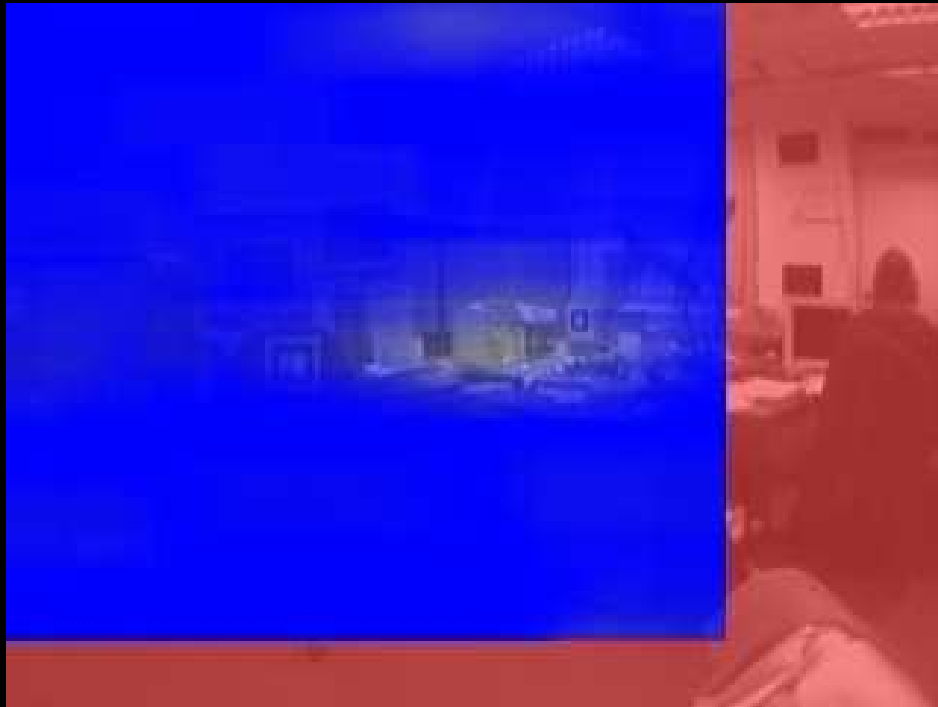
- Image has:
 - Cluttered areas (interesting)
 - Uniform areas (uninteresting)

Image of interest



- Image has:
 - Cluttered areas (interesting)
 - Uniform areas (uninteresting)
- Corners detected on clutter

Image of interest



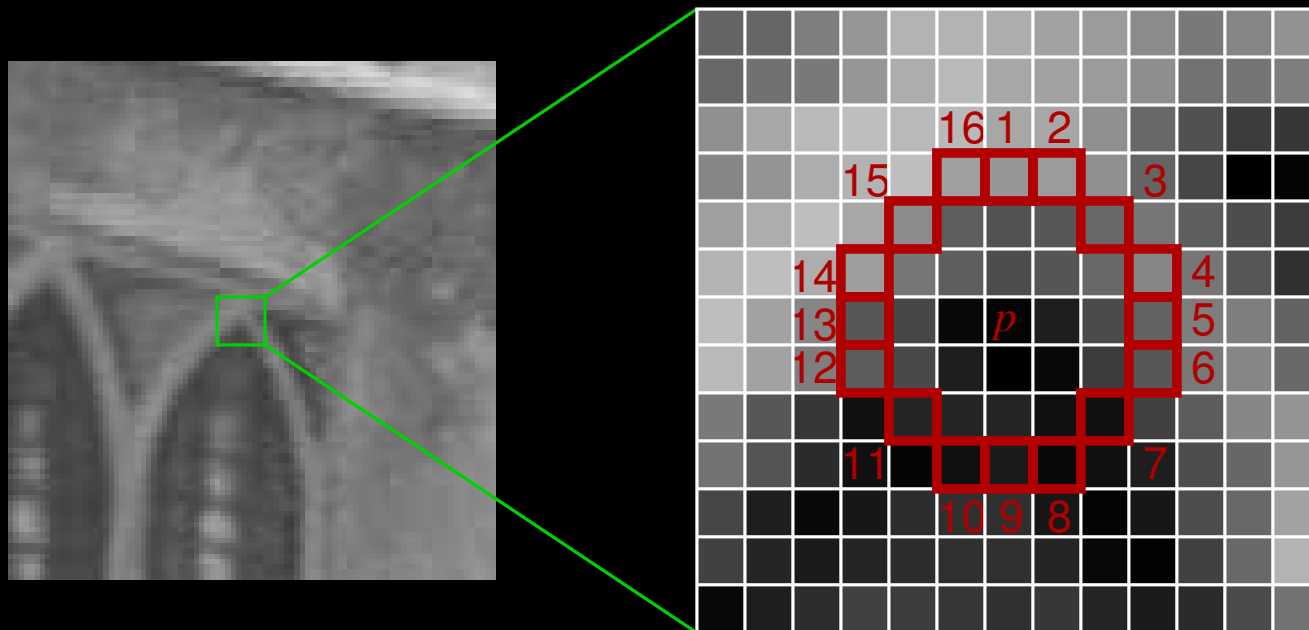
- Blue is uninteresting
- Red is disallowed

- Image has:
 - Cluttered areas (interesting)
 - Uniform areas (uninteresting)
- Corners detected on clutter

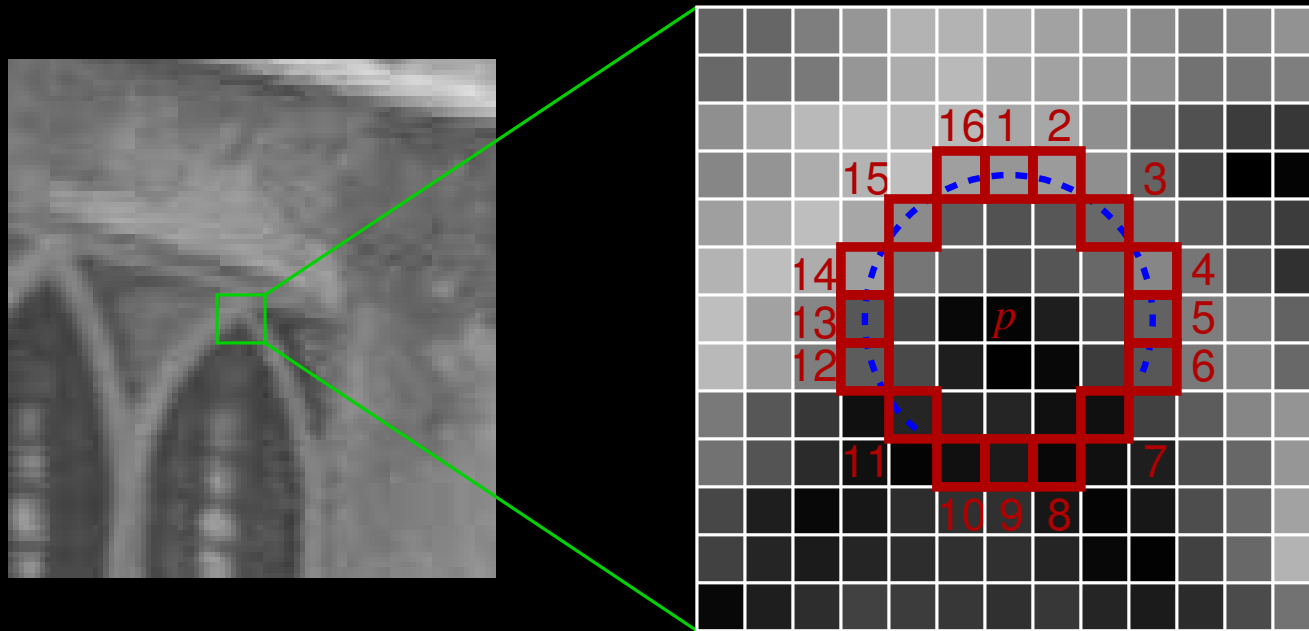
The FAST feature detector



The FAST feature detector

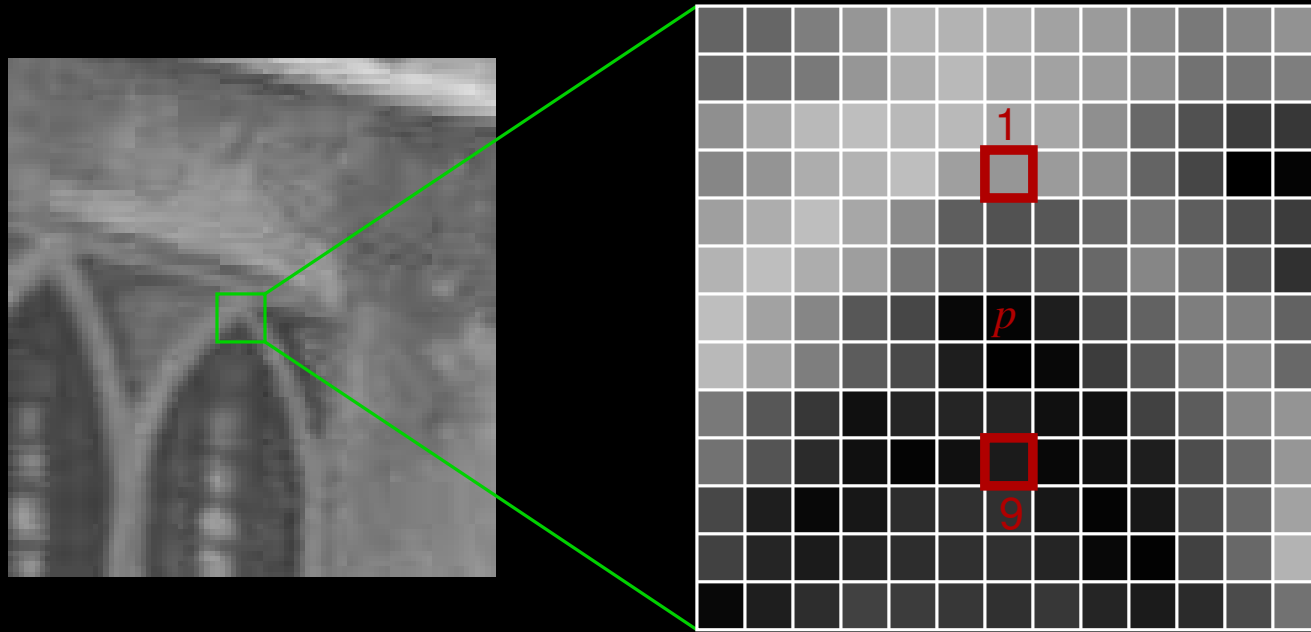


The FAST feature detector



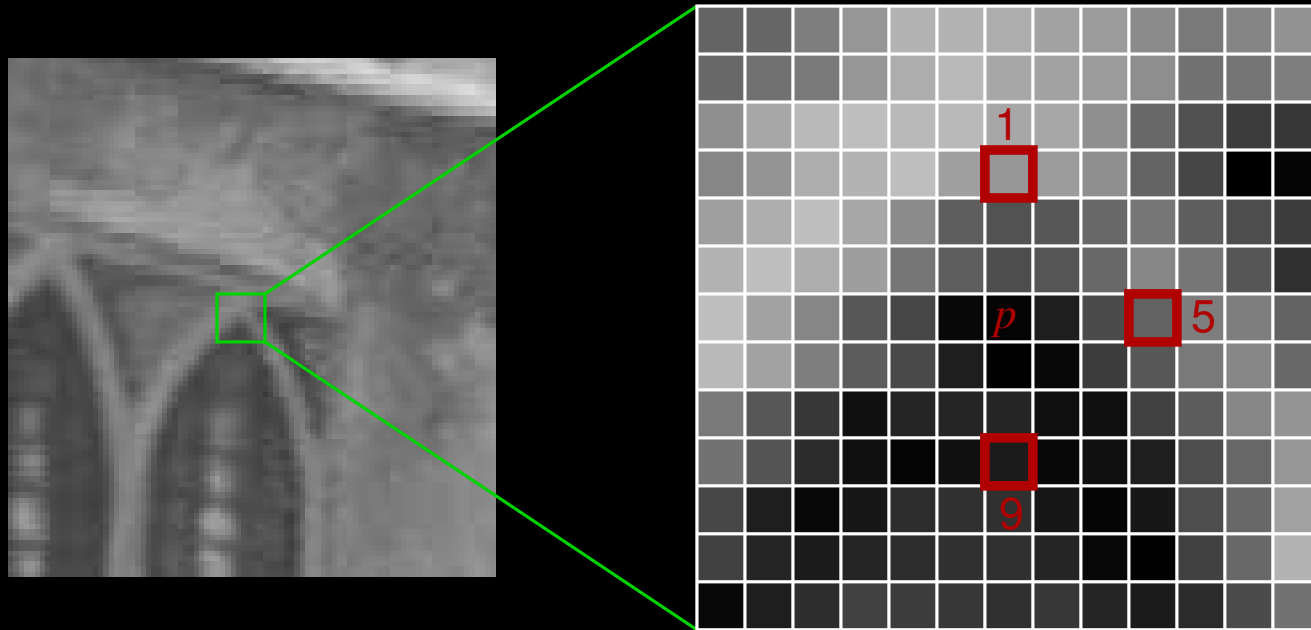
- ≥ 12 contiguous pixels brighter than $p + \text{threshold}$

The FAST feature detector



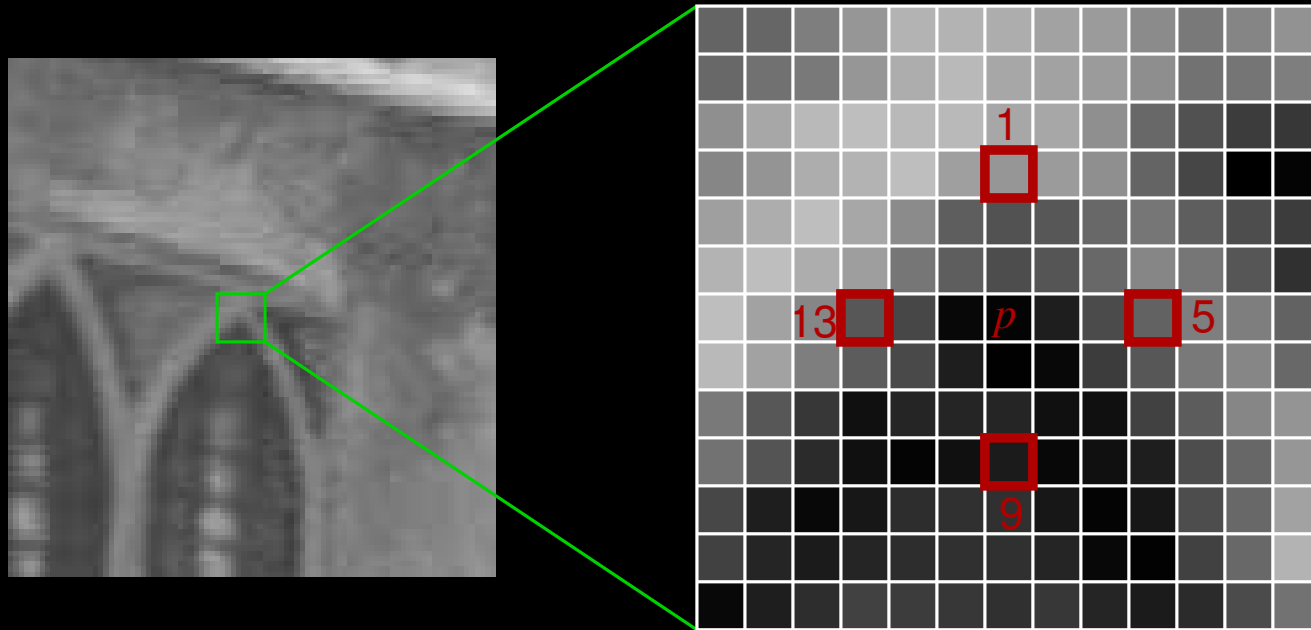
- ≥ 12 contiguous pixels brighter than $p + \text{threshold}$
- Rapid rejection by testing 1, 9

The FAST feature detector



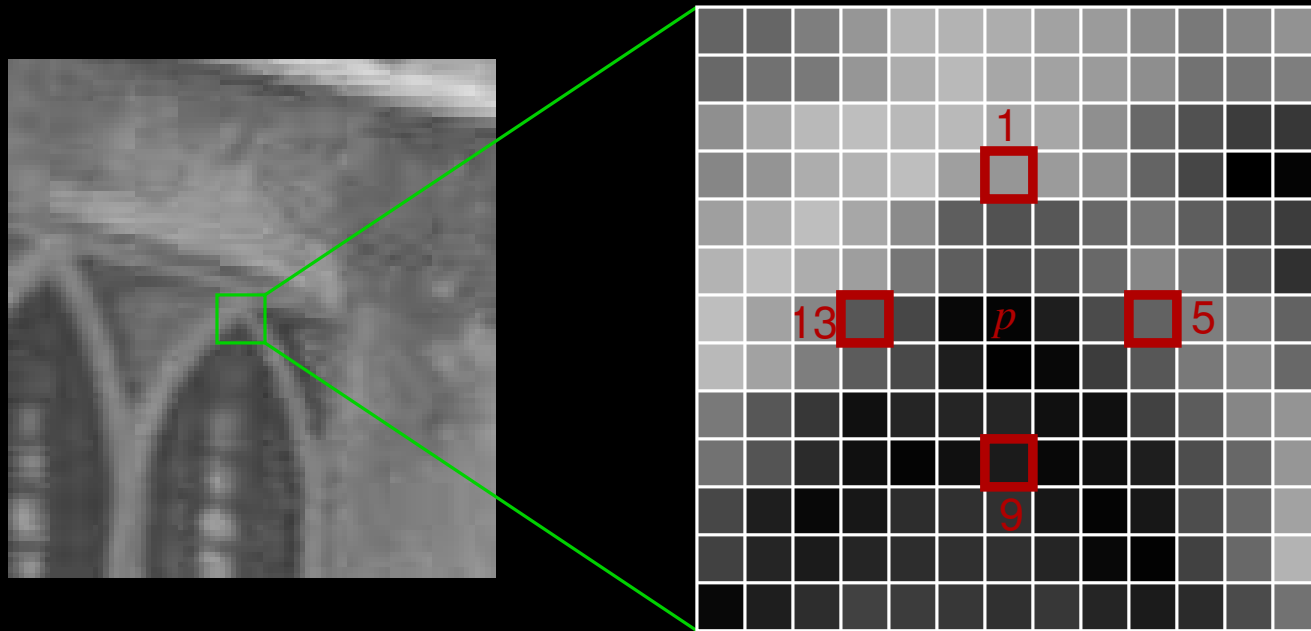
- ≥ 12 contiguous pixels brighter than $p + \text{threshold}$
- Rapid rejection by testing 1, 9, 5

The FAST feature detector



- ≥ 12 contiguous pixels brighter than $p + \text{threshold}$
- Rapid rejection by testing 1, 9, 5 then 13

The FAST feature detector



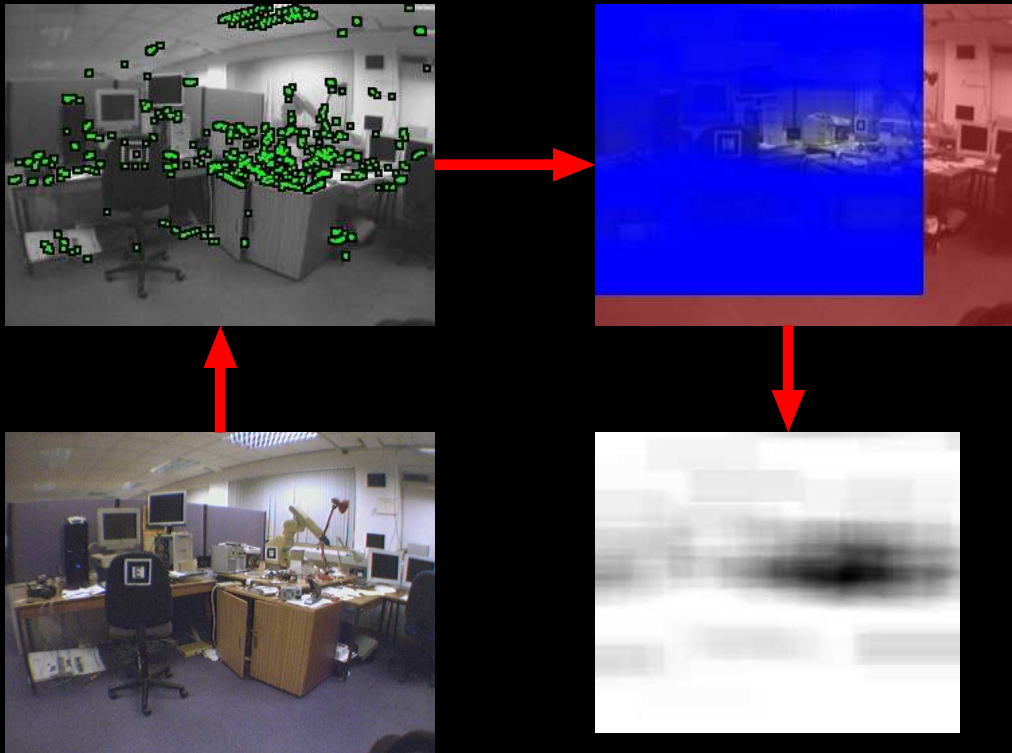
- ≥ 12 contiguous pixels brighter than $p + threshold$
- Rapid rejection by testing 1, 9, 5 then 13
- 1.59ms (Opteron 2.6GHz) - 8% of available CPU time
- Source code available
- <http://savannah.nongnu.org/projects/libcvsd>

Suitability image



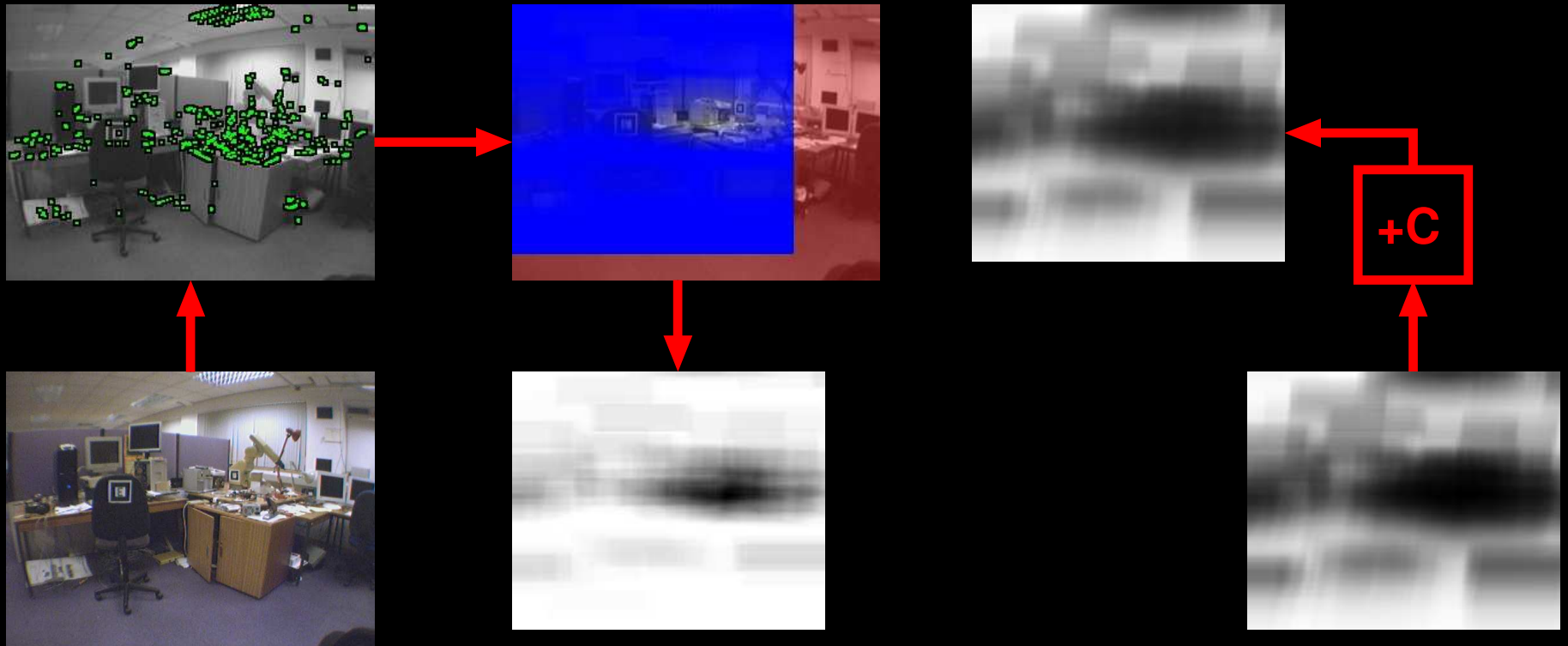
- $\frac{\text{max corners occluded} - \text{number of corners occluded}}{\text{max corners occluded}}$
- Efficiently computed
- Gives suitability for label as a function of image position

Filtering



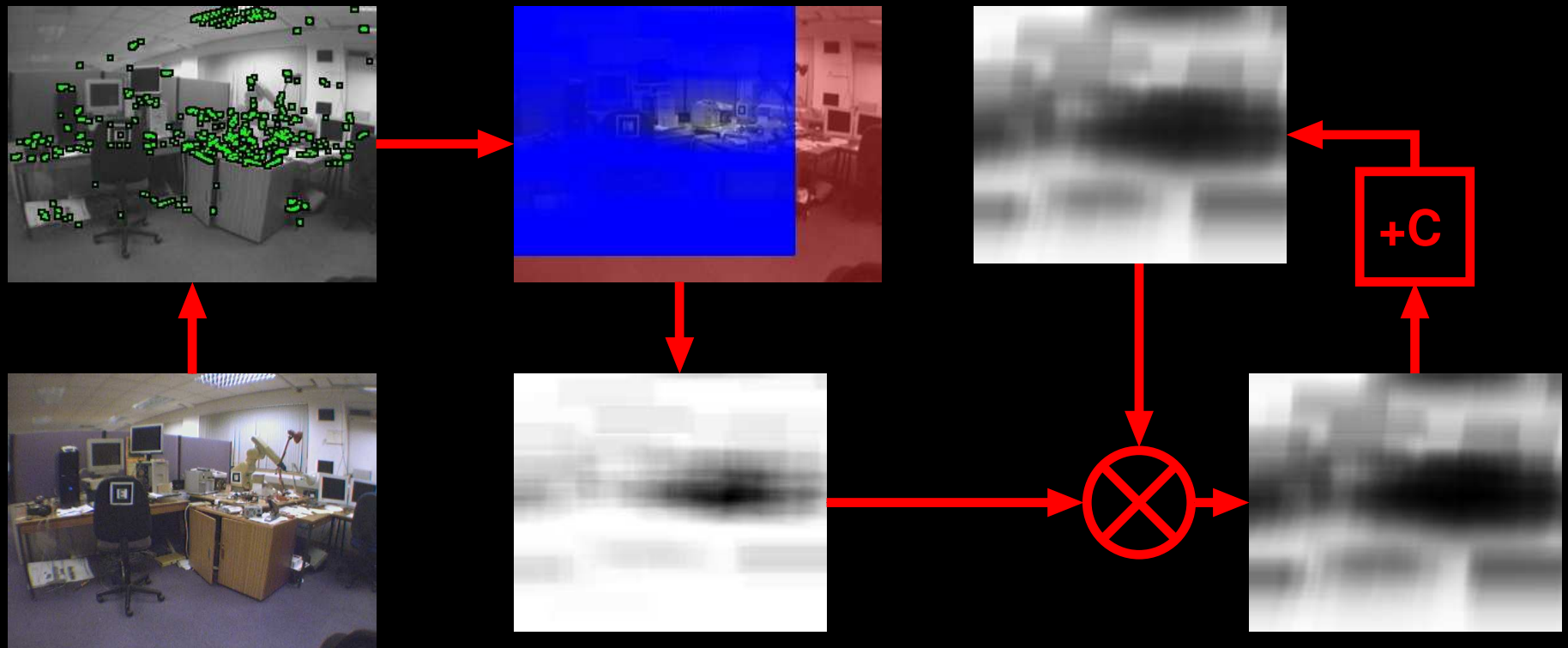
- Detect corners and compute suitability

Filtering



- Detect corners and compute suitability
- Add constant to previous filter

Filtering



- Detect corners and compute suitability
- Add constant to previous filter
- Multiply new filter by new suitability

Filtering



- Detect corners and compute suitability
- Add constant to previous filter
- Multiply new filter by new suitability

Adding constraints

- Filter gives potential position of labels
- Position of maximum is the best position

Adding constraints

- Filter gives potential position of labels
- Position of maximum is the best position

- Label shouldn't jump too often
- May want
 - labels near objects
 - screen stabilized labels

Adding constraints

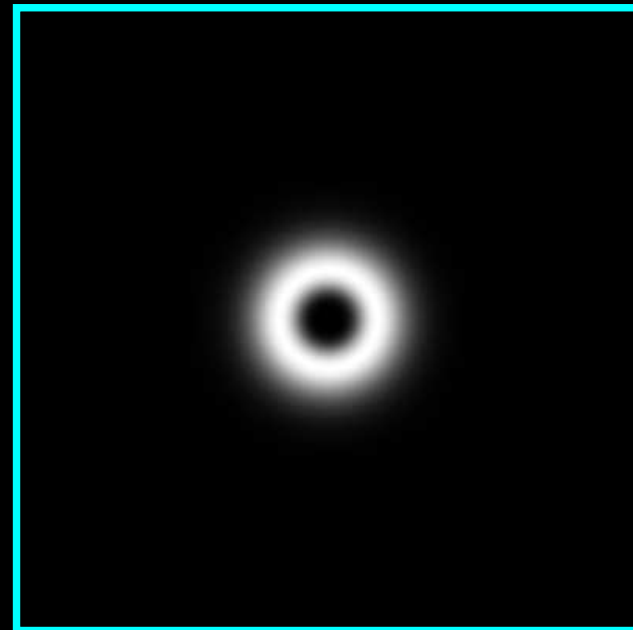
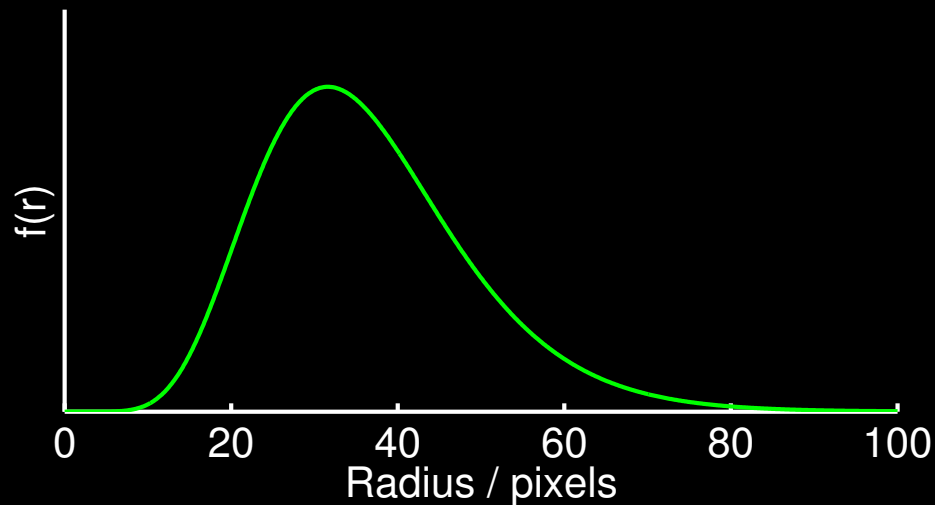
- Filter gives potential position of labels
- Position of maximum is the best position

- Label shouldn't jump too often
- May want
 - labels near objects
 - screen stabilized labels

- Arbitrary constraints can be added by multiplication

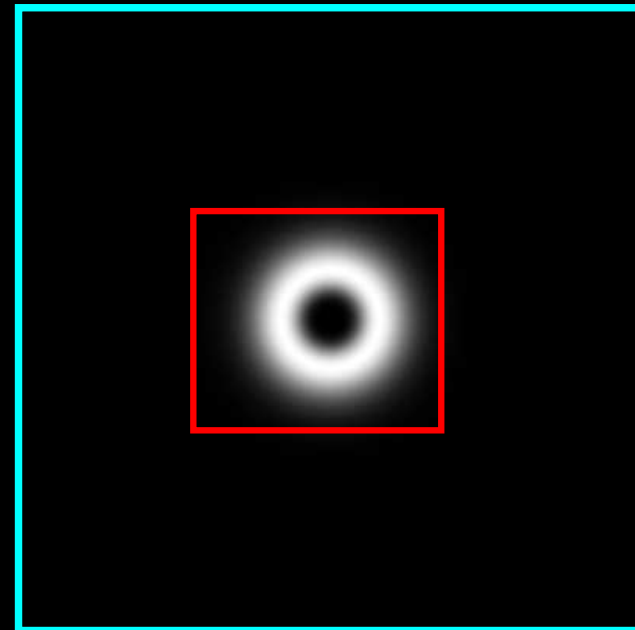
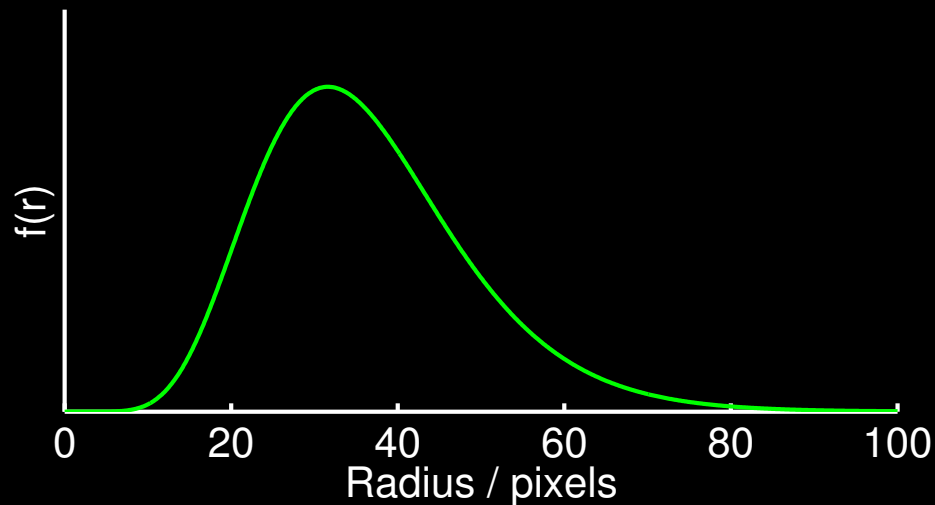
Circular constraint

- Detect object location (AR toolkit)
- Center circular constraint on object position
- $f(r) = r^a e^{-r^b}$, $r = \sqrt{x^2 + y^2}$
- Close, but not too close



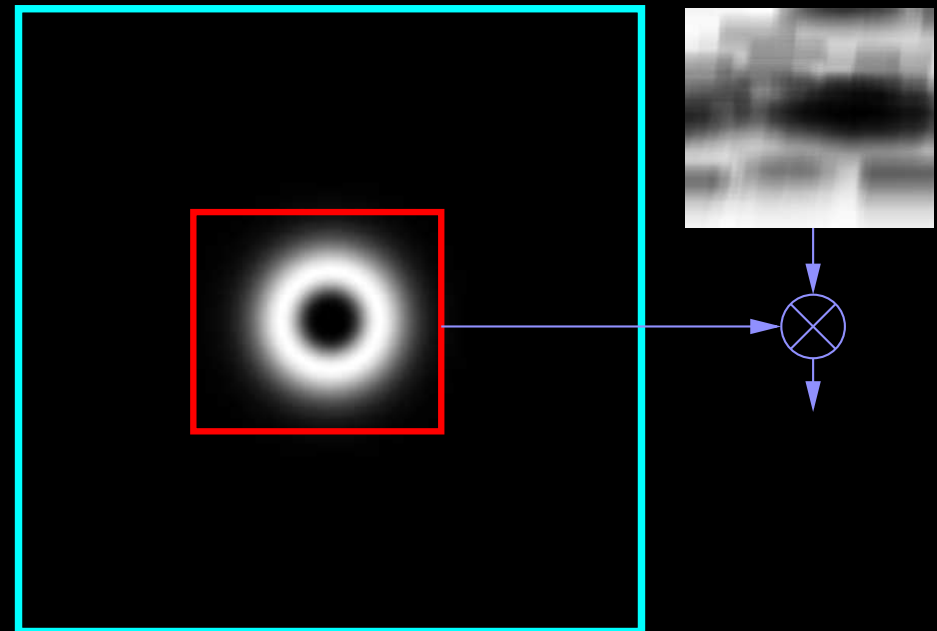
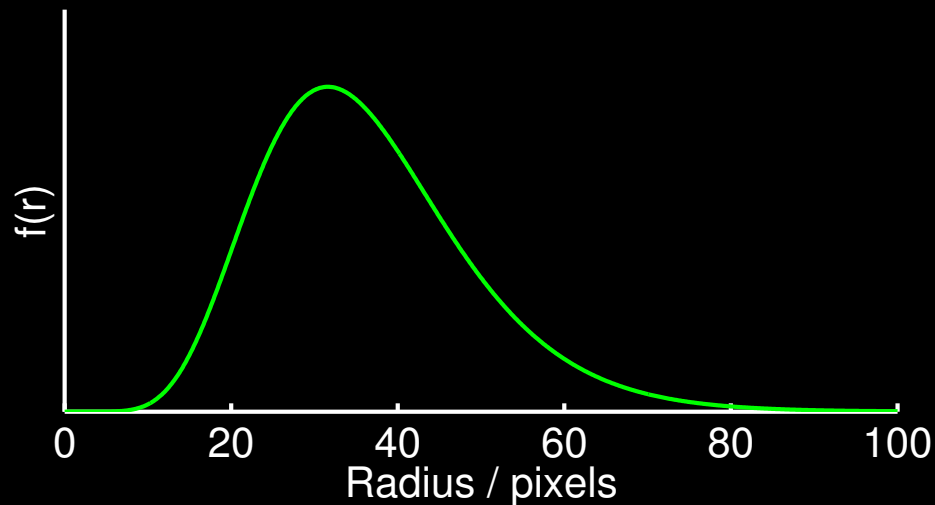
Circular constraint

- Detect object location (AR toolkit)
- Center circular constraint on object position
- $f(r) = r^a e^{-r^b}$, $r = \sqrt{x^2 + y^2}$
- Close, but not too close



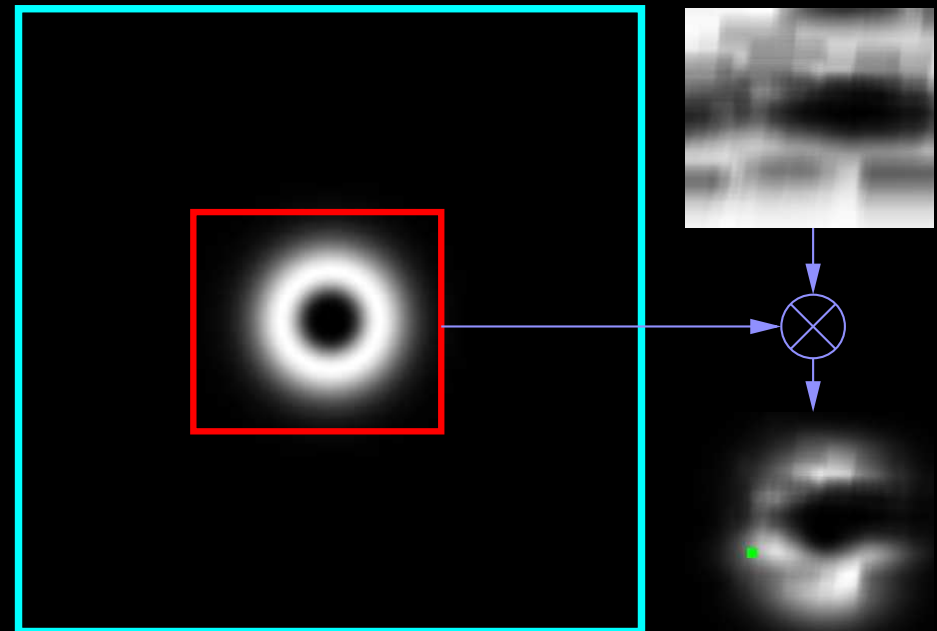
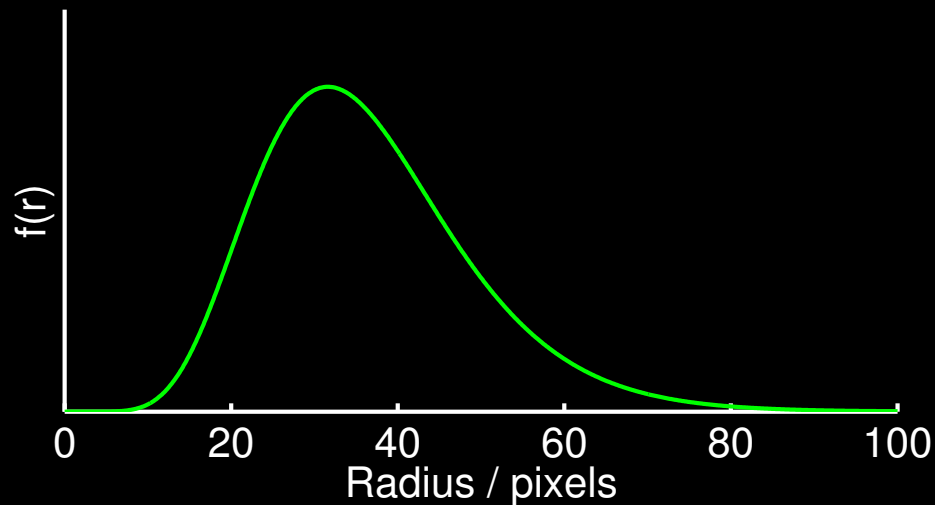
Circular constraint

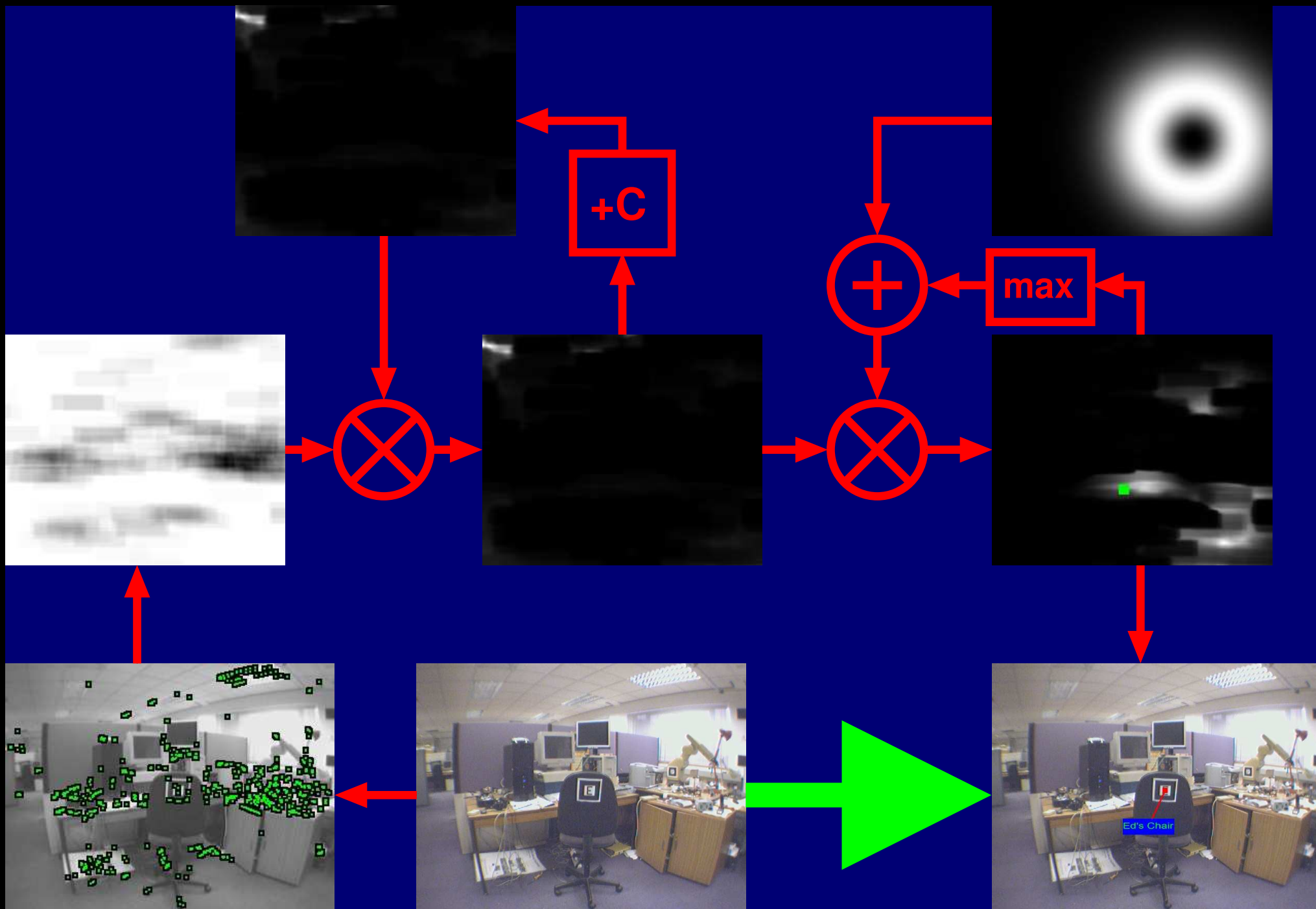
- Detect object location (AR toolkit)
- Center circular constraint on object position
- $f(r) = r^a e^{-r^b}$, $r = \sqrt{x^2 + y^2}$
- Close, but not too close



Circular constraint

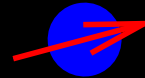
- Detect object location (AR toolkit)
- Center circular constraint on object position
- $f(r) = r^a e^{-r^b}$, $r = \sqrt{x^2 + y^2}$
- Close, but not too close





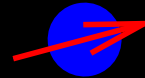
Other constraints

- Directional constraint

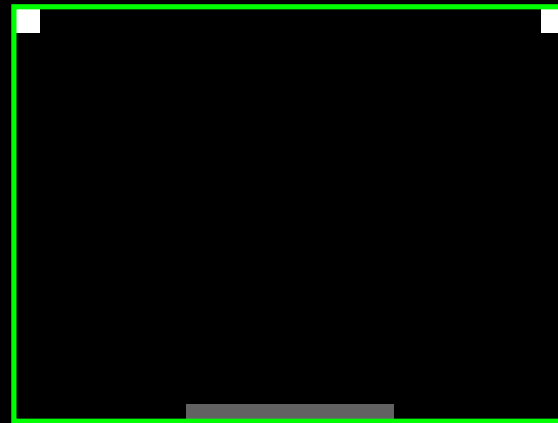


Other constraints

- Directional constraint



- Screen stabilized



Timing breakdown

		Time (ms)
Per-frame cost	Feature detection	1.76
	Integral image	0.65
Per-label cost	Suitability measurement	1.03
	Filter measurements	0.63
	Insert constraint	0.51
Total cost	For one label	4.58
	For n labels	$2.41 + 2.17n$
Cost per frame	over 10 frames	0.46

Video

**Label
placement**

4.6 ms/frame

Summary

- Labels placed to avoid occluding interesting items
- Interest defined by density of corners
- Filtering allows smooth motion and infrequent jumps of label
- Purely image based—no modelling of world required
- Filter allows arbitrary constraints
- Efficient implementation
 - using very efficient corner detector
 - computation every n frames only

Any Questions?