Machine Vision made easy with Ruby

Brown St / Paternoster Row, Sheffield, England, United Kingdom Address is approximate
Photos »
Street View
WORKSTATION
Paternoster Row
Brown St
© 2010 Google
Report a problem

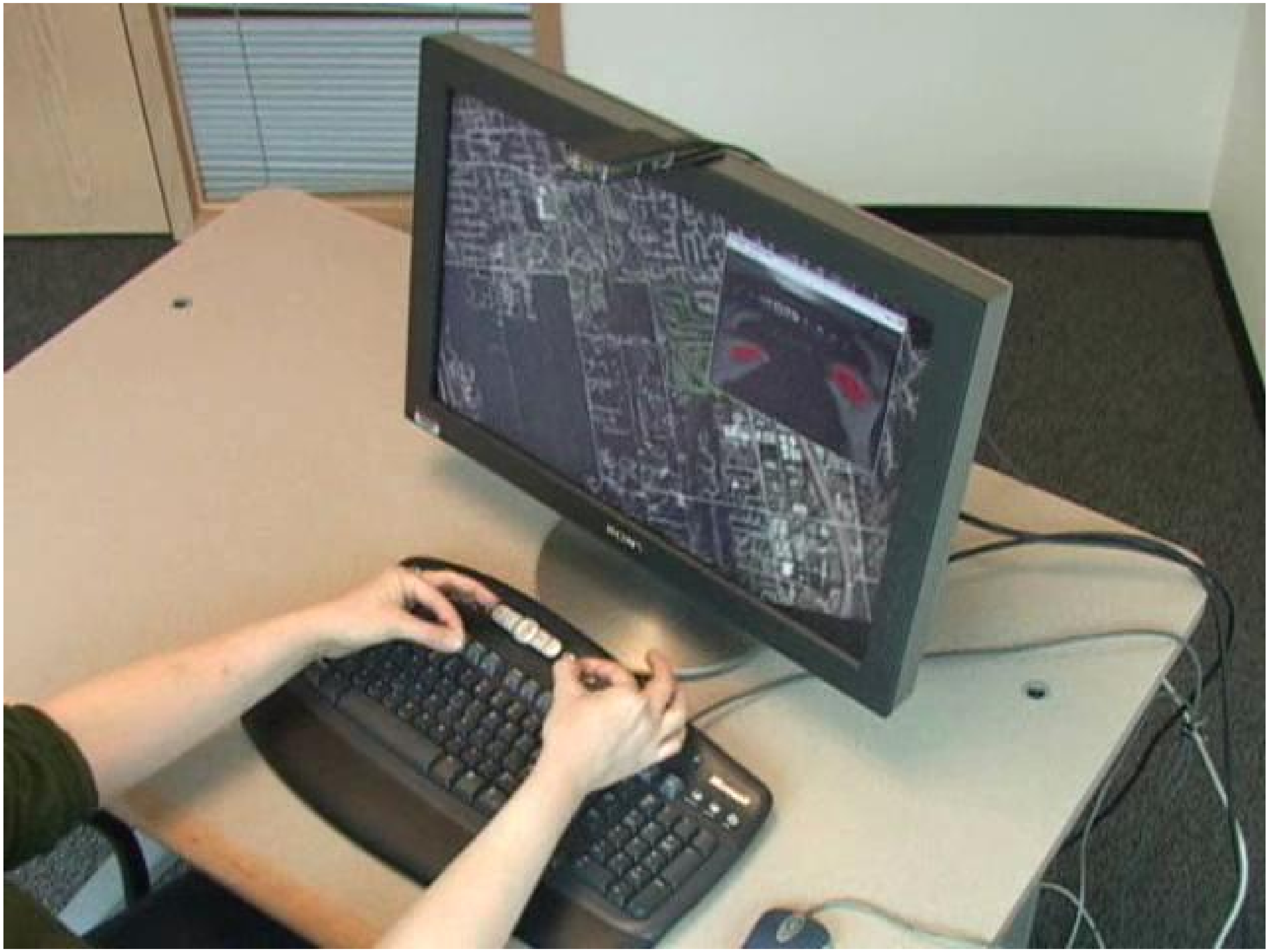
Machine Vision made easy with Ruby

Jan Wedekind

Mon Jun 14 19:00:00 BST 2010

Motivation I/II

Andrew Wilson: Robust Computer Vision-Based Detection of Pinching for One and Two-Handed Gesture Input



http://research.microsoft.com/~awilson

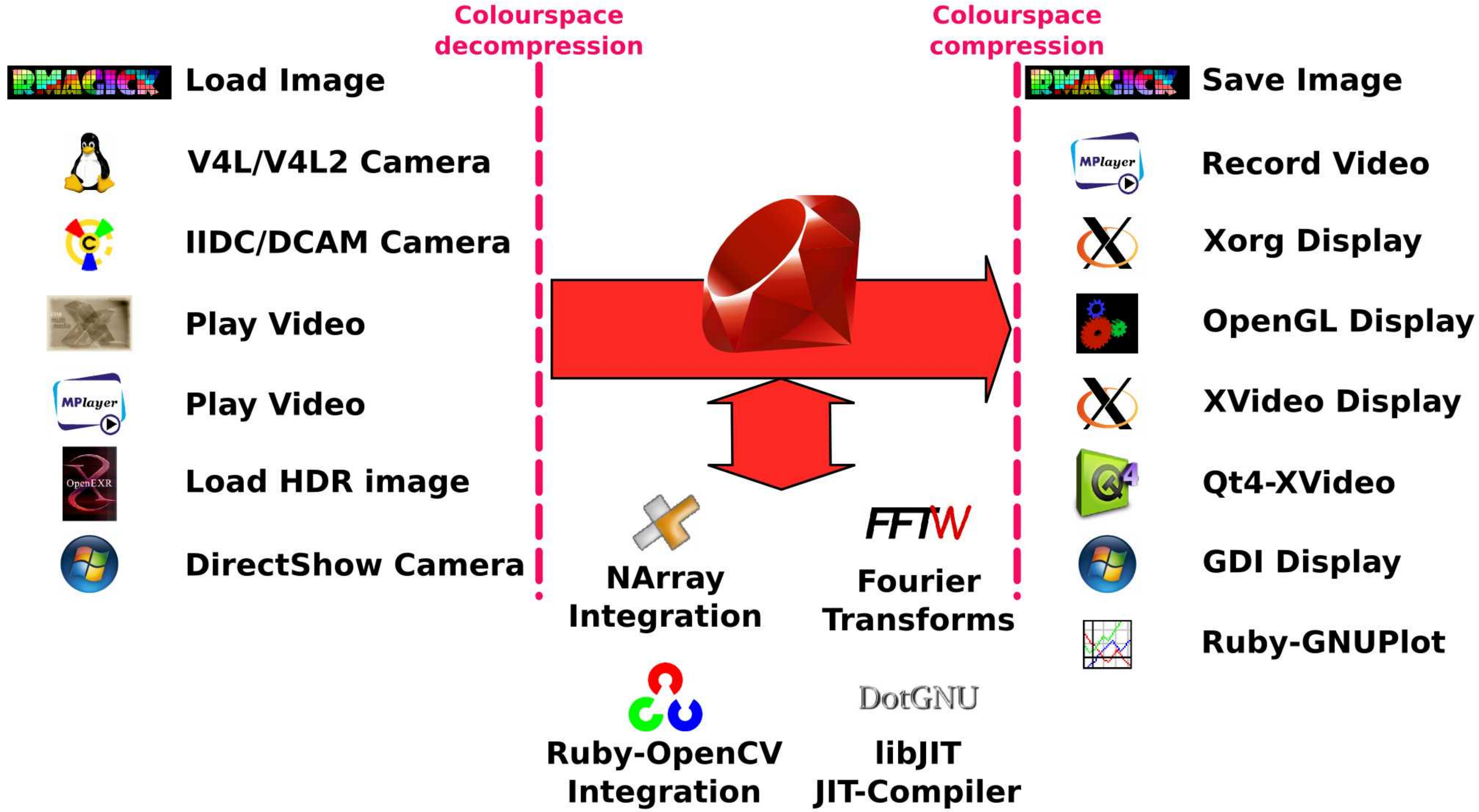
Motivation II/II

- subtract background from input image
- threshold resulting difference image
- connected component labeling
- discard components touching border of image
- select component of significant size
- extract centroid
- . . .

http://bit.ly/b9sCgw

Hornetseye

Input/Output



Colourspace decompression

Colourspace compression

- Load Image
- V4L/V4L2 Camera
- IIDC/DCAM Camera
- Play Video
- Play Video
- Load HDR image
- DirectShow Camera

- Save Image
- Record Video
- Xorg Display
- OpenGL Display
- XVideo Display
- Qt4-XVideo
- GDI Display
- Ruby-GNUPlot

- NArray Integration
- Fourier Transforms
- Ruby-OpenCV Integration
- libJIT JIT-Compiler

Hornetseye

Malloc Objects

```
m = Malloc.new 10
m.write '0123456789'
# "0123456789"
m.read 5
# "01234"
( m + 2 ).read 5
# "23456"
```

Hornetseye

Array Operations I/II

operation	loop body	loop variable
write element	r[b] =a	-
read element	r =a[b]	-
write sub-array	r[b+i]=a[i]	i
read sub-array	r[i] =a[i+b]	i
fill	r[i] =a	i
index array	r[i] =i	i
unary function	r[i] =f(a[i])	i
binary function	r[i] =f(a,b[i])	i
binary function	r[i] =f(a[i],b)	i
binary function	r[i] =f(a[i],b[i])	i
accumulate	r =f(r,a[i])	i
⋮	⋮	⋮

Hornetseye

Array Operations II/II

operation	loop body	loop variable
⋮	⋮	⋮
warp/mask	r[i] =a[b[i]]	i
unmask	r[b[i]]=a[i]	i
downsampling	r[i] =a[b*i]	i
upsampling	r[b*i] =a[i]	i
integral	r[i] =r[i-1]+a[i]	i
map	r[i] =b[a[i]]	i
histogram	r[a[i]]=r[a[i]]+1	i
weighted hist.	r[a[i]]=r[a[i]]+b[i]	i
correlation	r[i] =r[i]+a[i+j]*b[j]	i,j

Installation

Kubuntu 8.04

- Install required packages:
  ```
  sudo aptitude install ruby1.8 ruby1.8-dev \
  irb1.8 imagemagick librmagick-ruby1.8 g++ ccache libboost-dev libxine-dev \
  libxine1-all-plugins libdc1394-13-dev xorg-dev libfftw3-dev libopenexr-dev \
  bison flex texinfo
  ```

- Install libJIT:
  ```
  wget http://vision.eng.shu.ac.uk/jan/libjit-0.1.3pre.tar.bz2
  tar xjf libjit-0.1.3pre.tar.bz2
  cd libjit-0.1.3pre
  ./configure && make && sudo make install
  ```

- Download hornetseye-x.x.tar.bz2 from Rubyforge

- Install Hornetseye:
  ```
  tar xjf hornetseye-*.tar.bz2
  cd hornetseye-*
  ./configure.ruby18 && make && sudo make install
  ```

Installation

Microsoft Windows

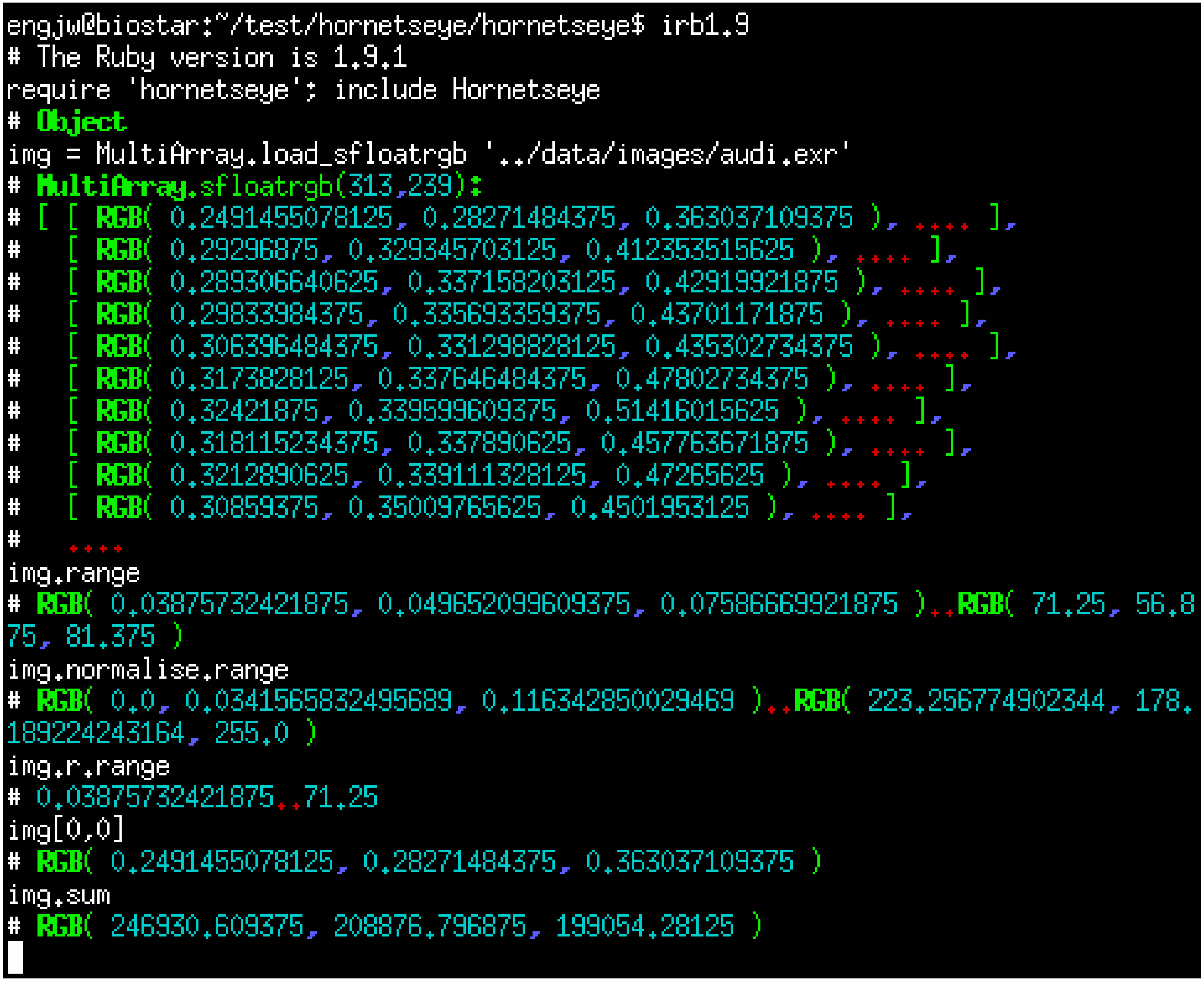
- Run Ruby one-click installer
- Reboot
- Unpack Ghostscript fonts to **c:\gs** (fonts should end up in **c:\gs\fonts**)
- Run ImageMagick installer
- Download RMagick Rubygem and install using the command
  gem install rmagick-2.6.0-x86-mswin32.gem
- Download NSIS installer for Hornetseye from Rubyforge and run it
- Optionally install NArray, MPlayer, Qt4 (requires MinGW), and Qt4QtRuby

Installation

Future: Installation with Rubygems?

```
gem install malloc
gem install multiarray
gem install hornetseye-xine
gem install hornetseye-video4linux
gem install hornetseye-video4linux2
gem install hornetseye-x11
...
```

Interactive Ruby



```
engjw@biostar:~/test/hornetseye/hornetseye$ irb1.9
# The Ruby version is 1.9.1
require 'hornetseye'; include Hornetseye
# Object
img = MultiArray.load_sfloatrgb '../data/images/audi.exr'
# MultiArray.sfloatrgb(313,239):
# [ [ RGB( 0.2491455078125, 0.28271484375, 0.363037109375 ), .... ],
#   [ RGB( 0.29296875, 0.329345703125, 0.412353515625 ), .... ],
#   [ RGB( 0.289306640625, 0.337158203125, 0.42919921875 ), .... ],
#   [ RGB( 0.29833984375, 0.335693359375, 0.43701171875 ), .... ],
#   [ RGB( 0.306396484375, 0.331298828125, 0.435302734375 ), .... ],
#   [ RGB( 0.3173828125, 0.337646484375, 0.47802734375 ), .... ],
#   [ RGB( 0.32421875, 0.339599609375, 0.51416015625 ), .... ],
#   [ RGB( 0.318115234375, 0.337890625, 0.457763671875 ), .... ],
#   [ RGB( 0.3212890625, 0.339111328125, 0.47265625 ), .... ],
#   [ RGB( 0.30859375, 0.35009765625, 0.4501953125 ), .... ],
#   ....
img.range
# RGB( 0.03875732421875, 0.049652099609375, 0.07586669921875 )..RGB( 71.25, 56.8
75, 81.375 )
img.normalise.range
# RGB( 0.0, 0.0341565832495689, 0.116342850029469 )..RGB( 223.25677490234, 178.
189224243164, 255.0 )
img.r.range
# 0.03875732421875..71.25
img[0,0]
# RGB( 0.2491455078125, 0.28271484375, 0.363037109375 )
img.sum
# RGB( 246930.609375, 208876.796875, 199054.28125 )
```

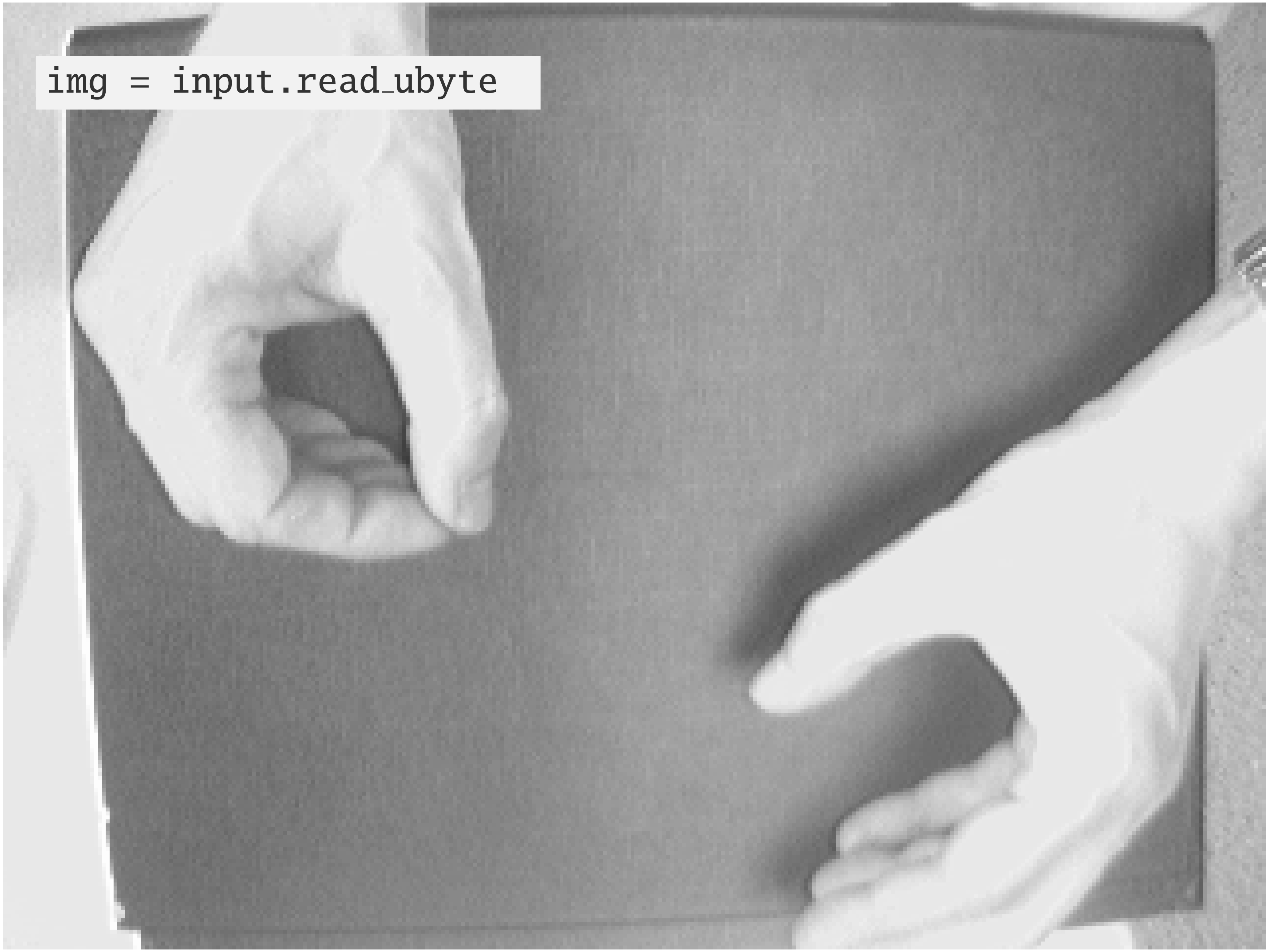
Gesture Recognition Example

Background Image

```
input = DC1394Input.new '', 0, 0,
   DC1394Input::FORMAT_VGA_NONCOMPRESSED,
   DC1394Input::MODE_320x240_YUV422
bg = input.read_sint
```

Gesture Recognition Example

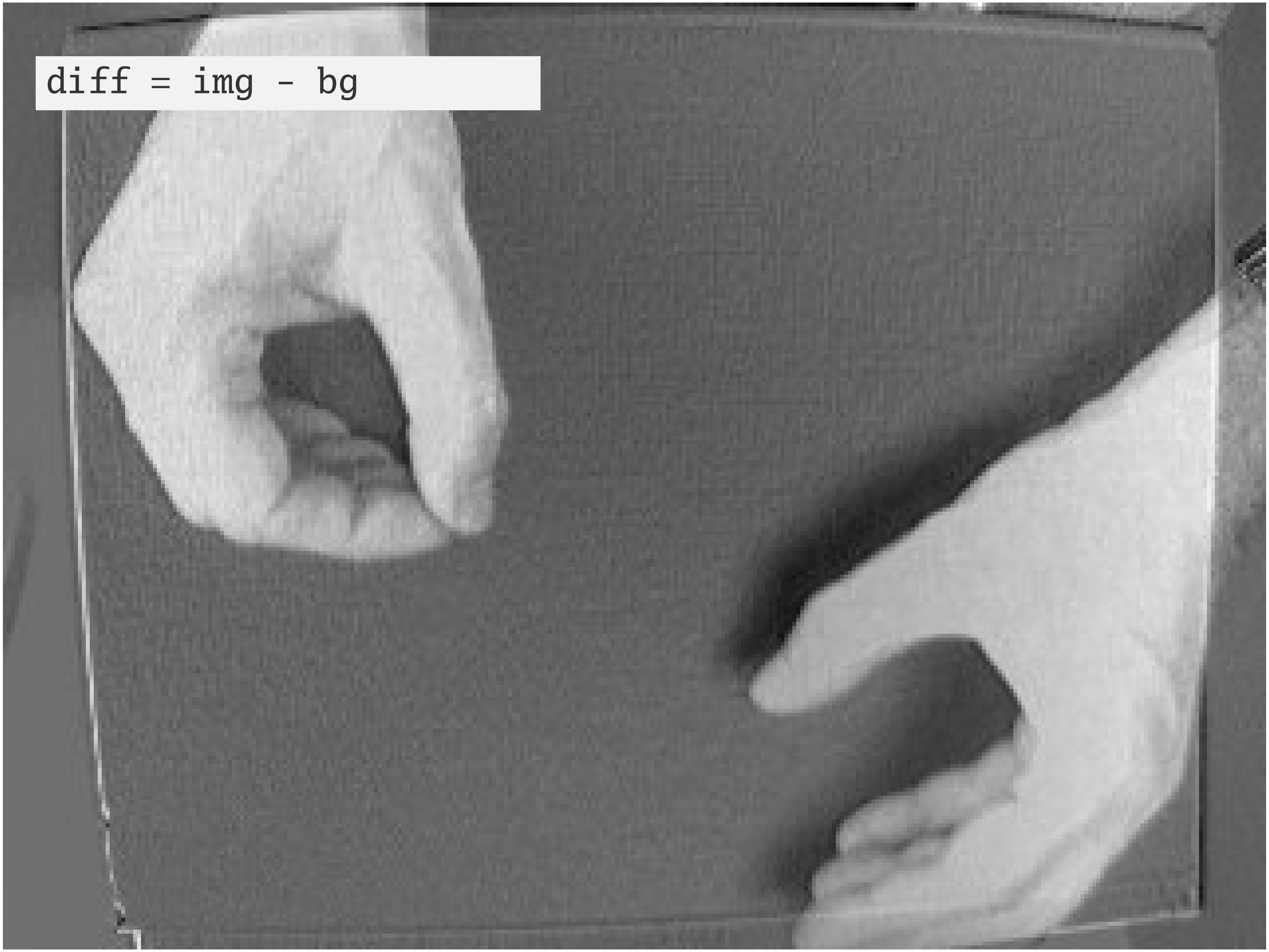
Input Image



```
img = input.read_ubyte
```

Gesture Recognition Example

Difference

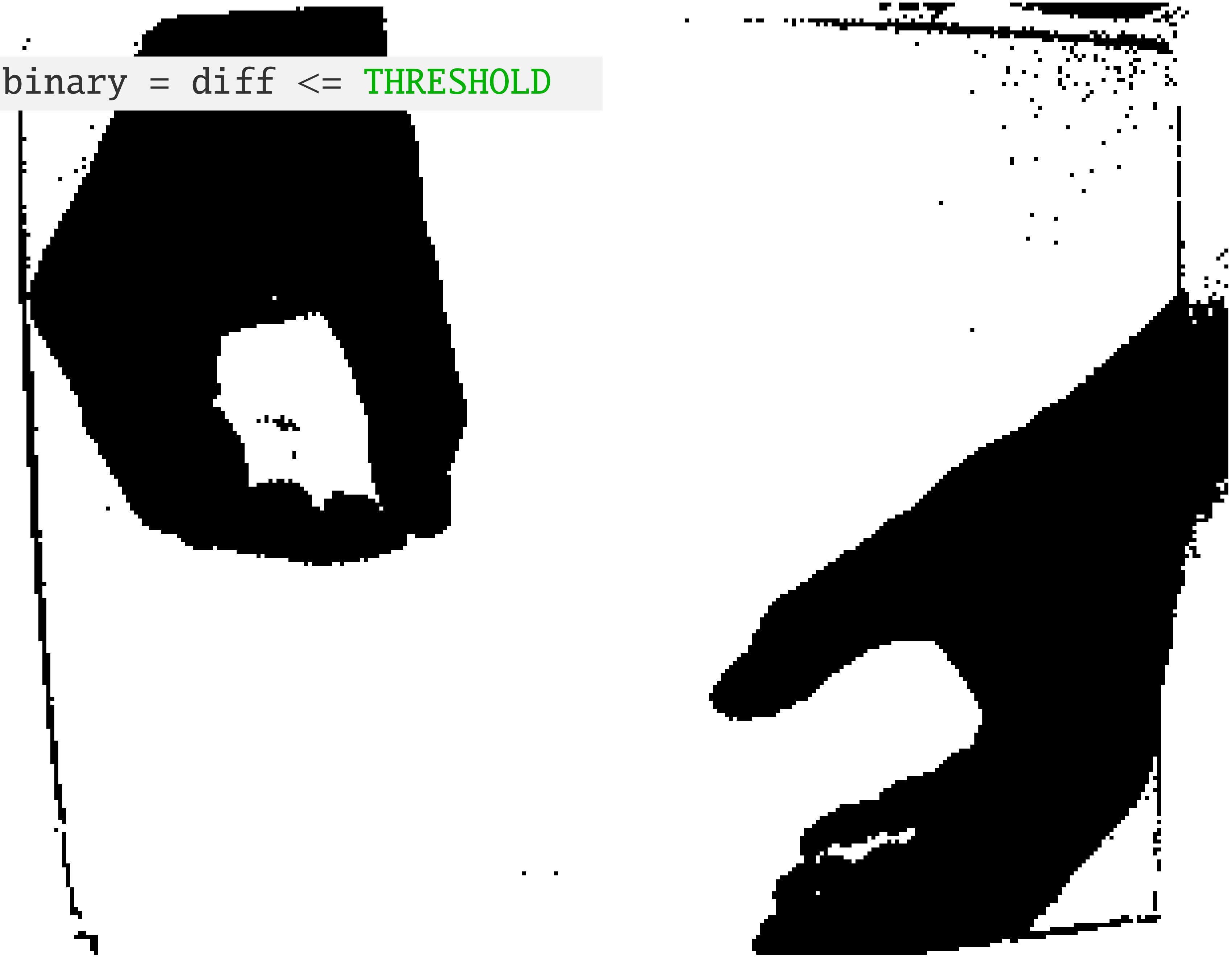


```
diff = img - bg
```

http://vision.eng.shu.ac.uk/jan/shrug7.pdf

Gesture Recognition Example

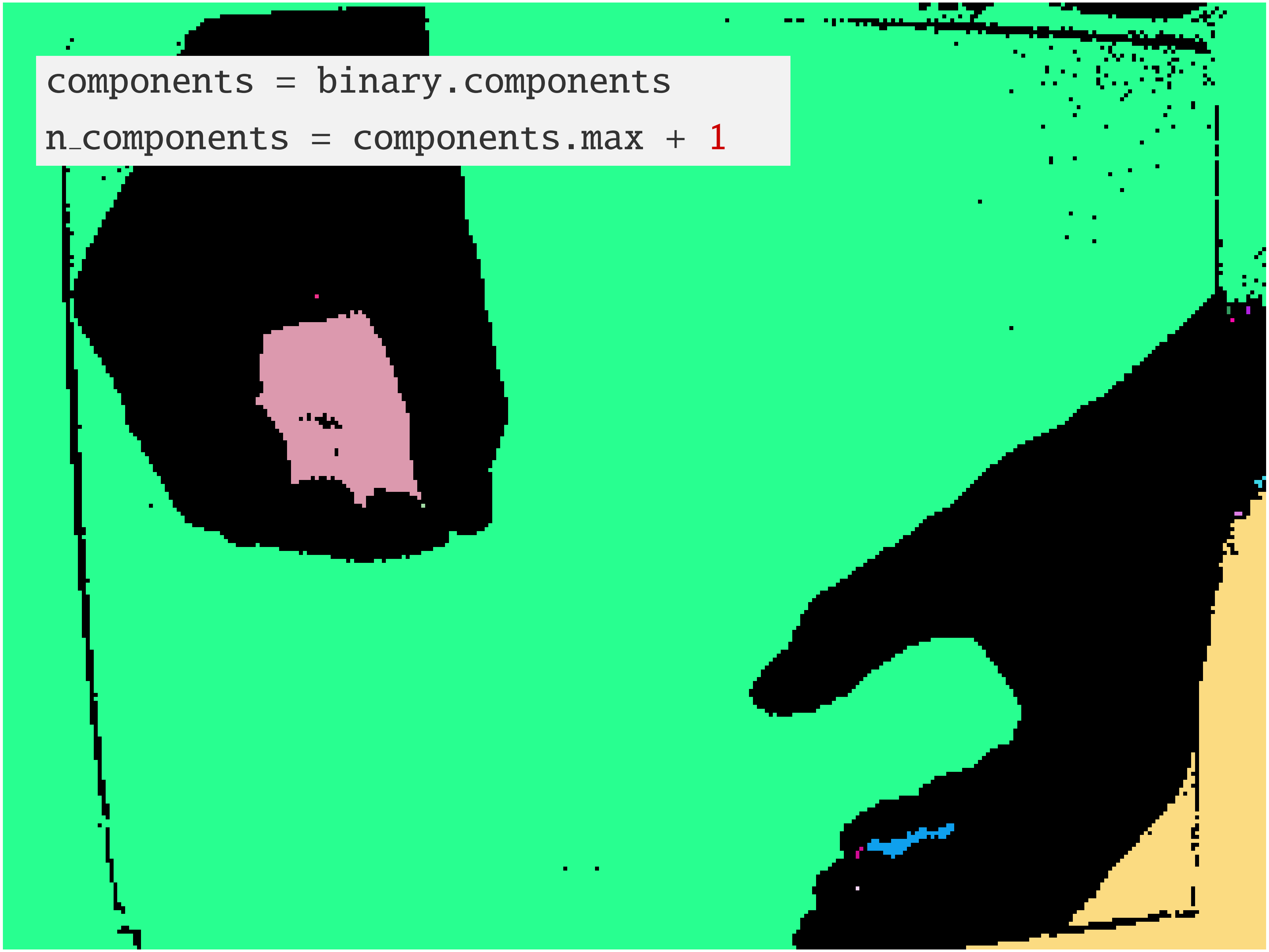
Thresholding



```
binary = diff <= THRESHOLD
```

Gesture Recognition Example

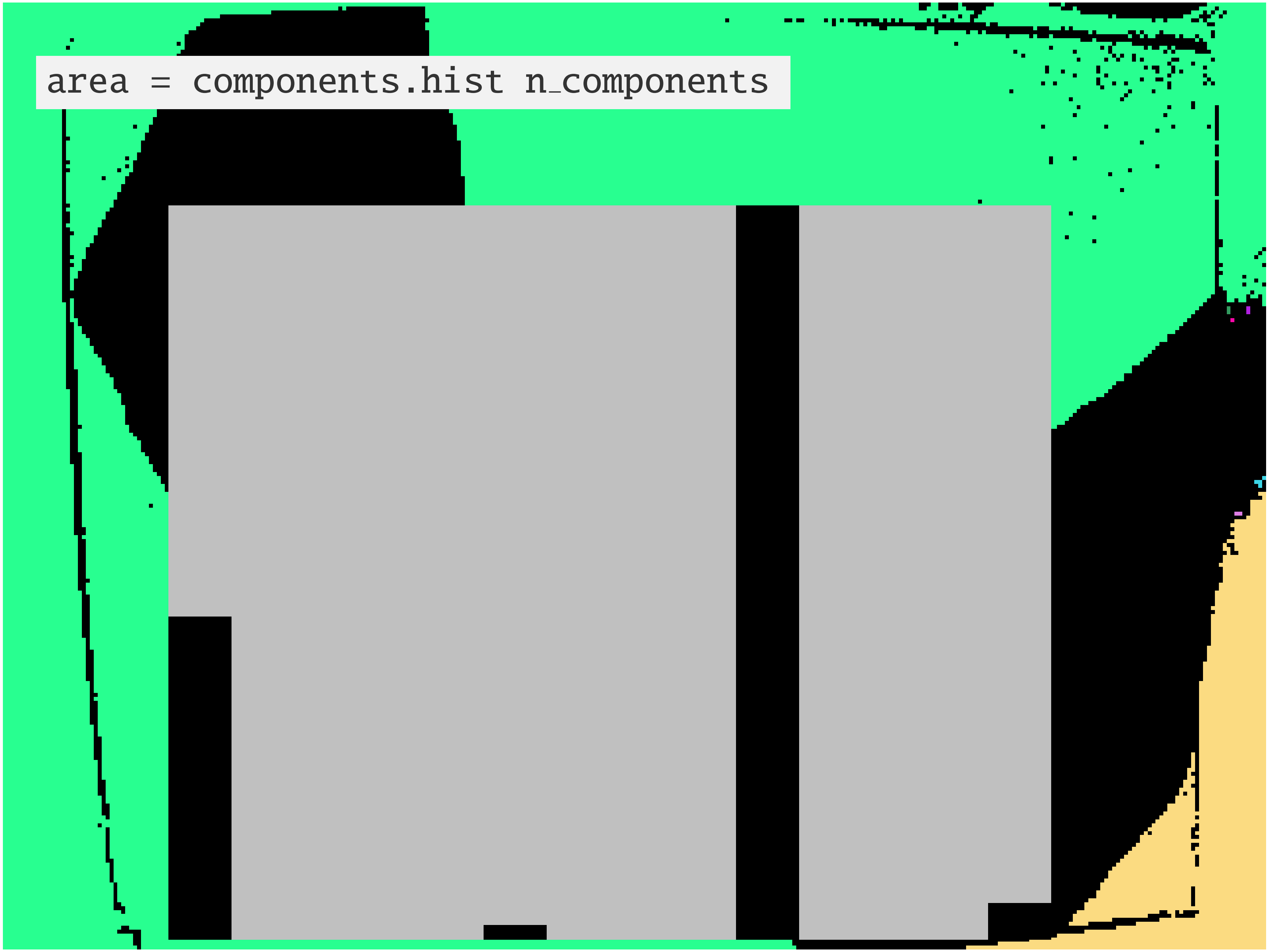
Components



```
components = binary.components
n_components = components.max + 1
```

Gesture Recognition Example

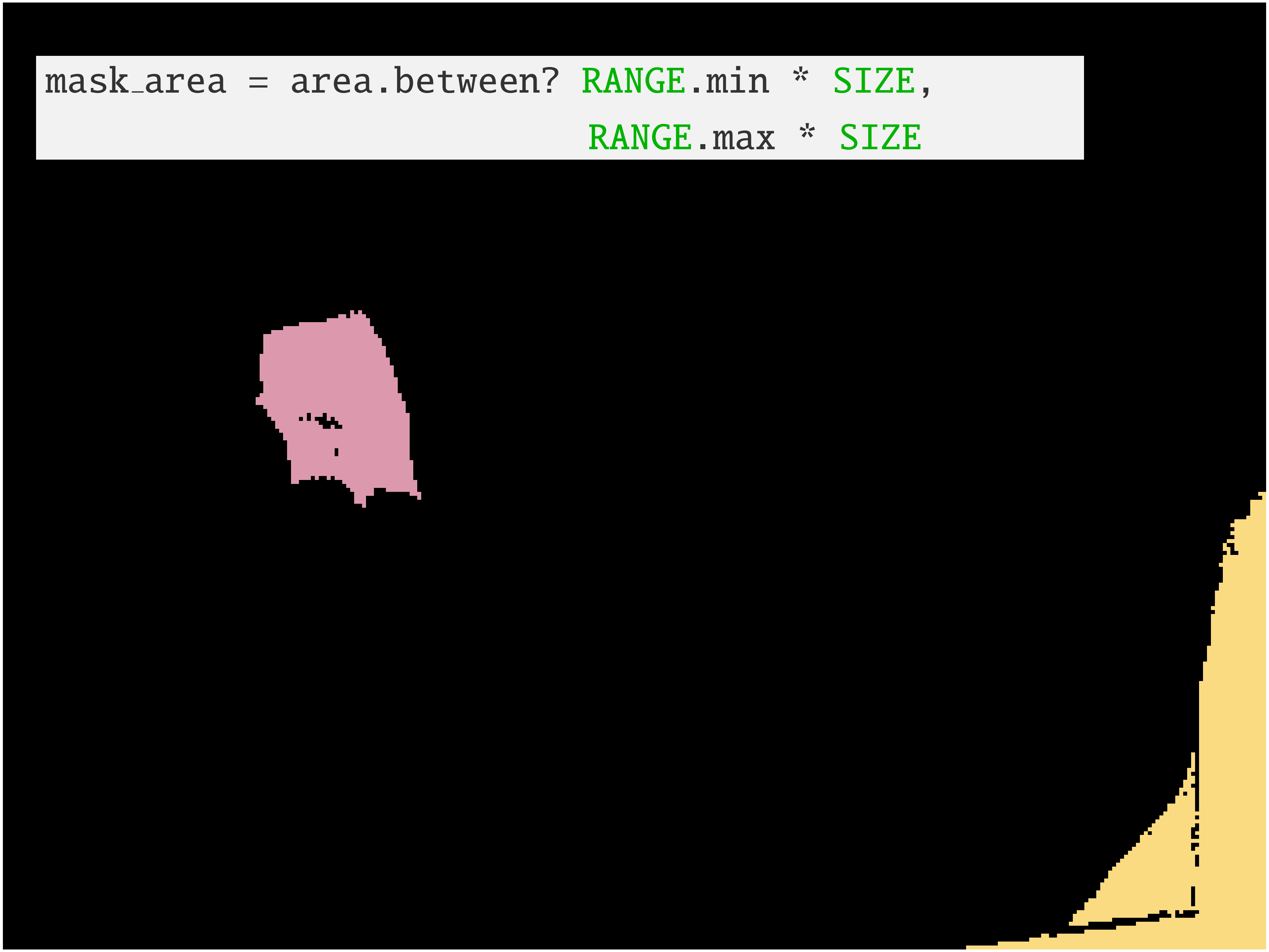
Compute Area of Components



```
area = components.hist n_components
```

Gesture Recognition Example

Impose Size Constraint



```
mask_area = area.between? RANGE.min * SIZE,
                          RANGE.max * SIZE
```

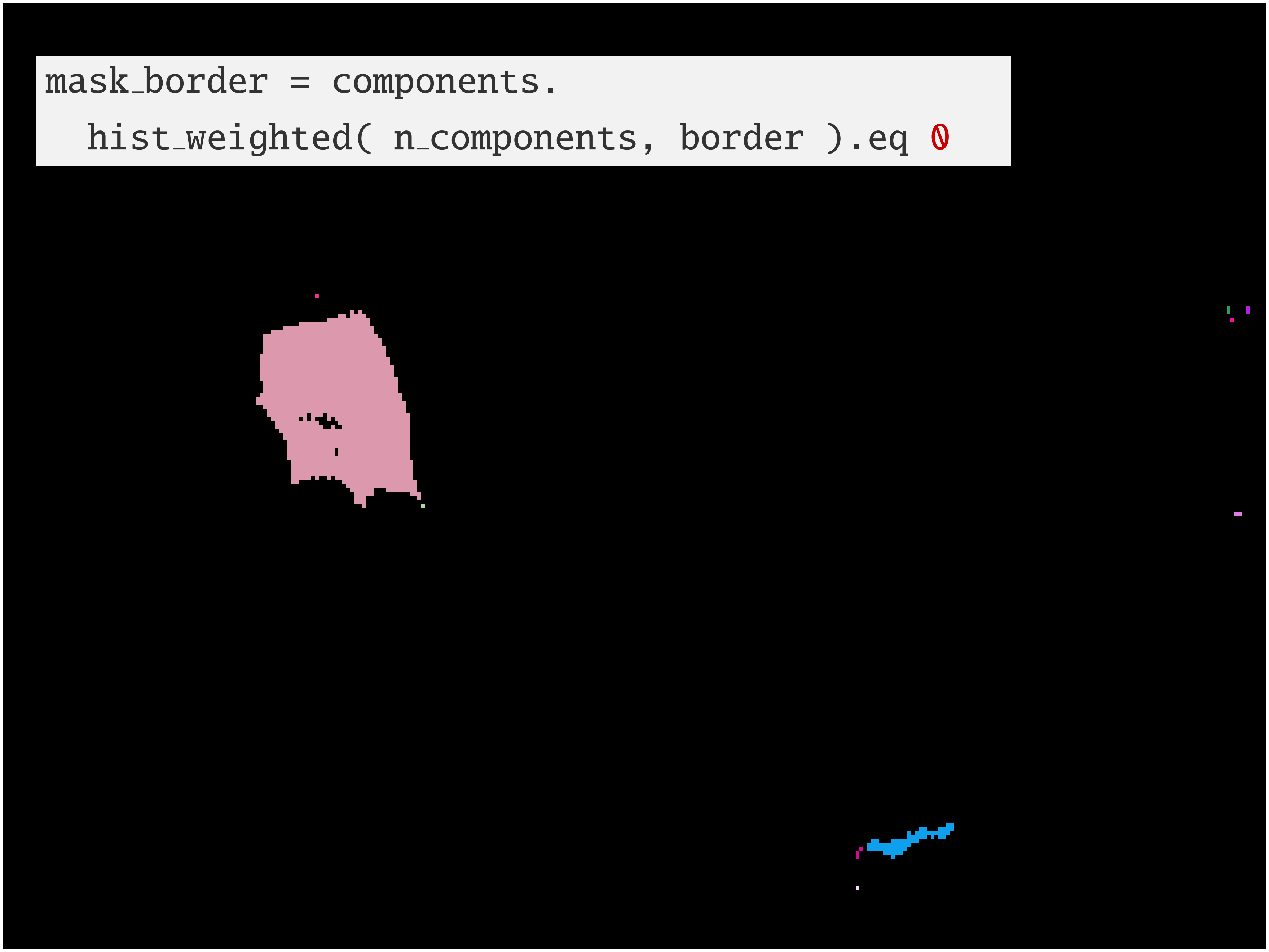
Gesture Recognition Example

Border Pixel

```
border = MultiArray.int( *SHAPE ).fill! 1
border[ 1 ... SHAPE[ 0 ] - 1, 1 ... SHAPE[ 1 ] - 1 ] = 0
```

Gesture Recognition Example

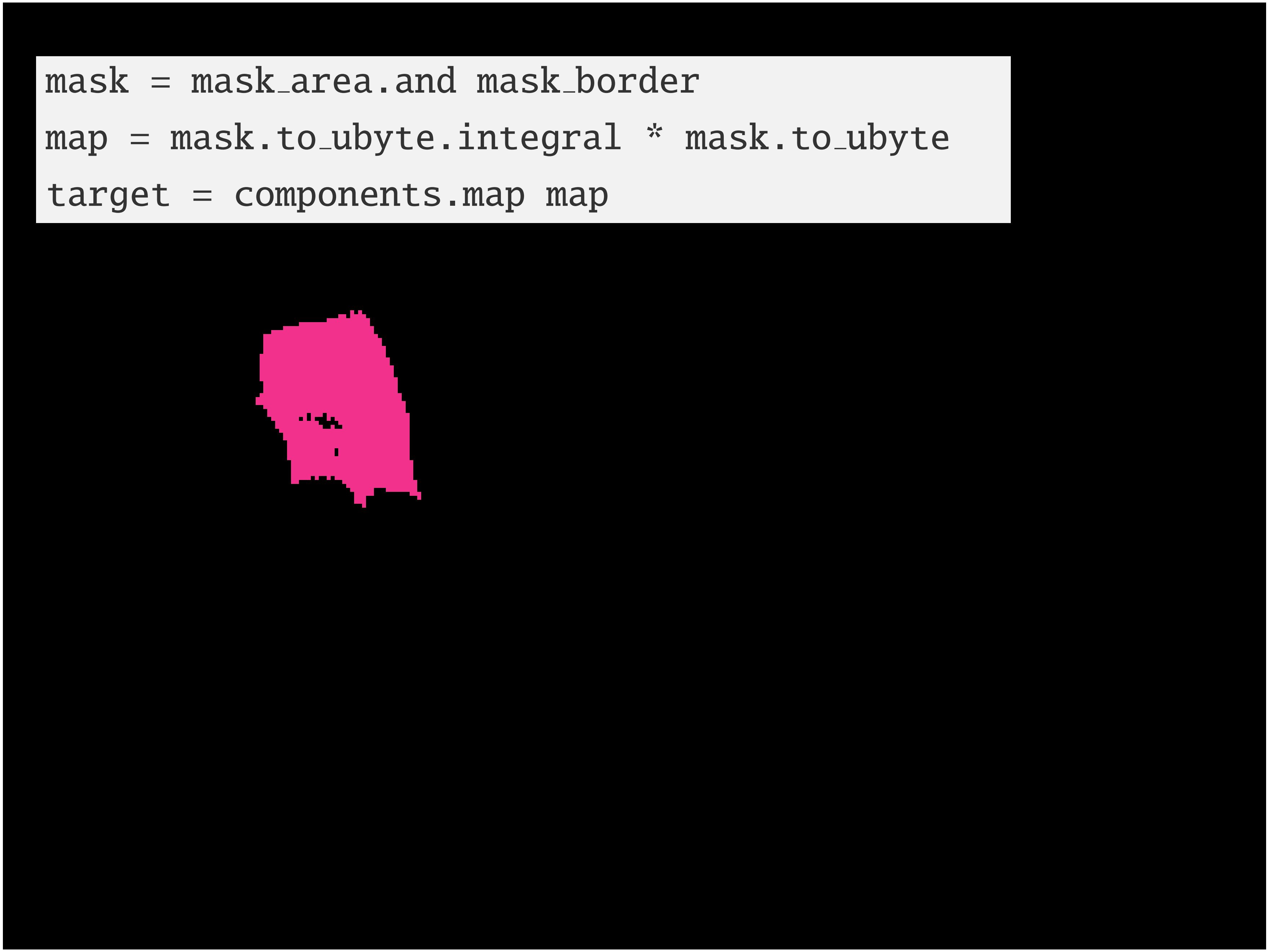
Reject Components touching Border



```
mask_border = components.
  hist_weighted( n_components, border ).eq 0
```

Gesture Recognition Example

Remaining Component(s)



```
mask = mask_area.and mask_border
map = mask.to_ubyte.integral * mask.to_ubyte
target = components.map map
```

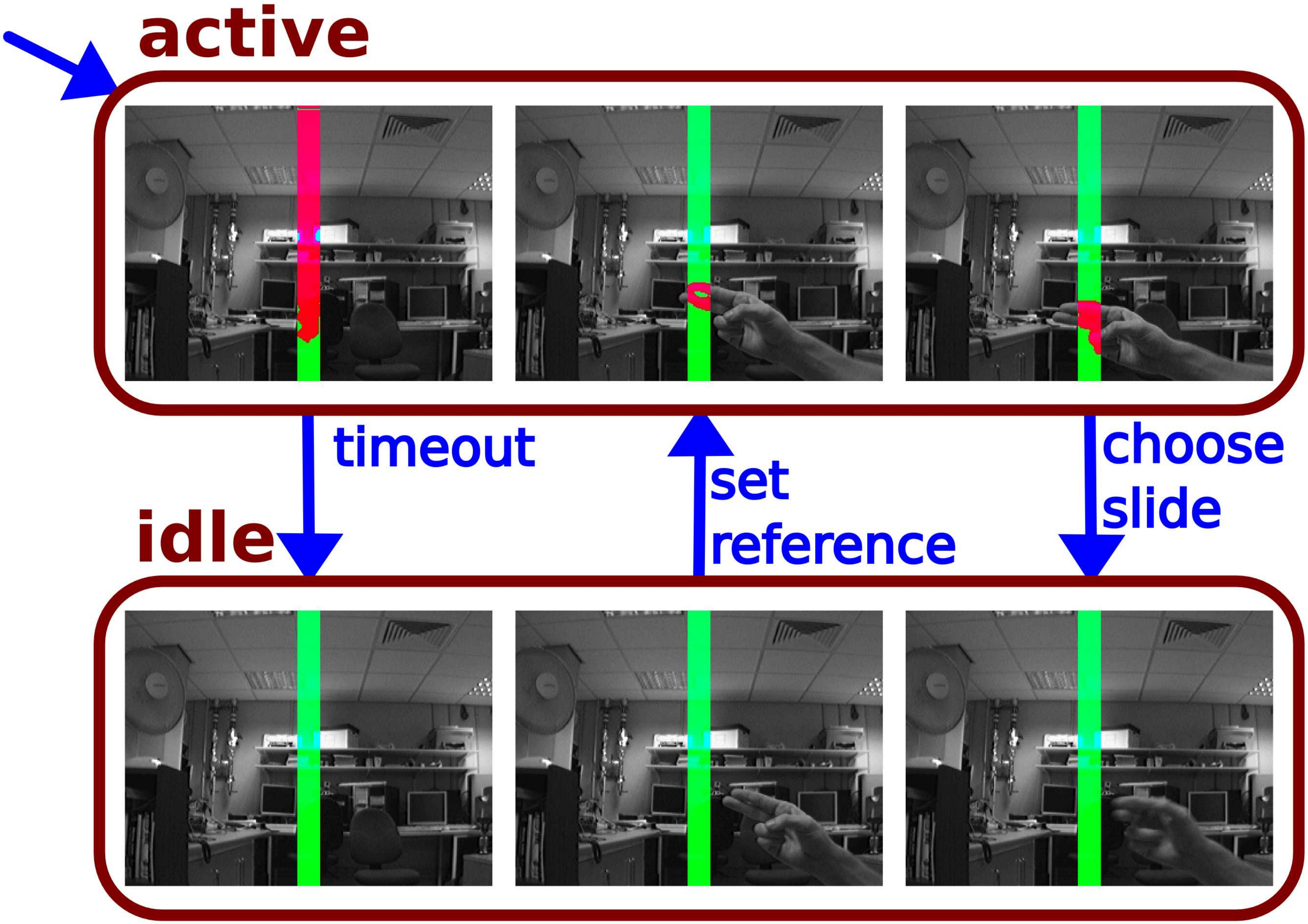
Gesture Recognition Example

Centre of Gravity

```
index = MultiArray.int( *SHAPE ).indgen!
x, y = index % SHAPE[ 0 ], index / SHAPE[ 0 ]
sum_target = target.sum.to_f
x_target = x.mask( target.to_bool ).sum / sum_target
y_target = y.mask( target.to_bool ).sum / sum_target
```

Other Examples

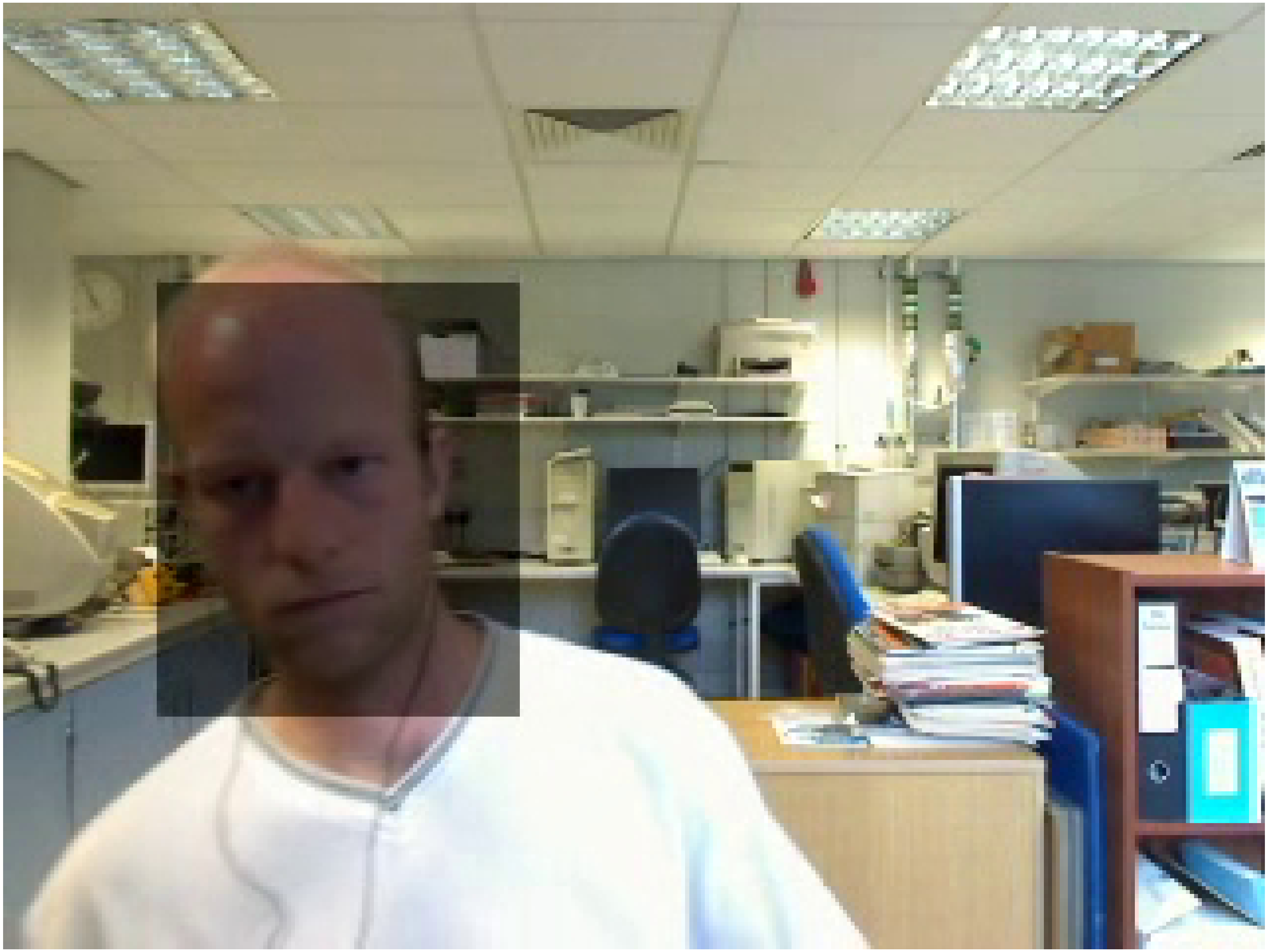
Presentation Software



http://www.youtube.com/watch?v=wNFr7RNWeCs

Other Examples

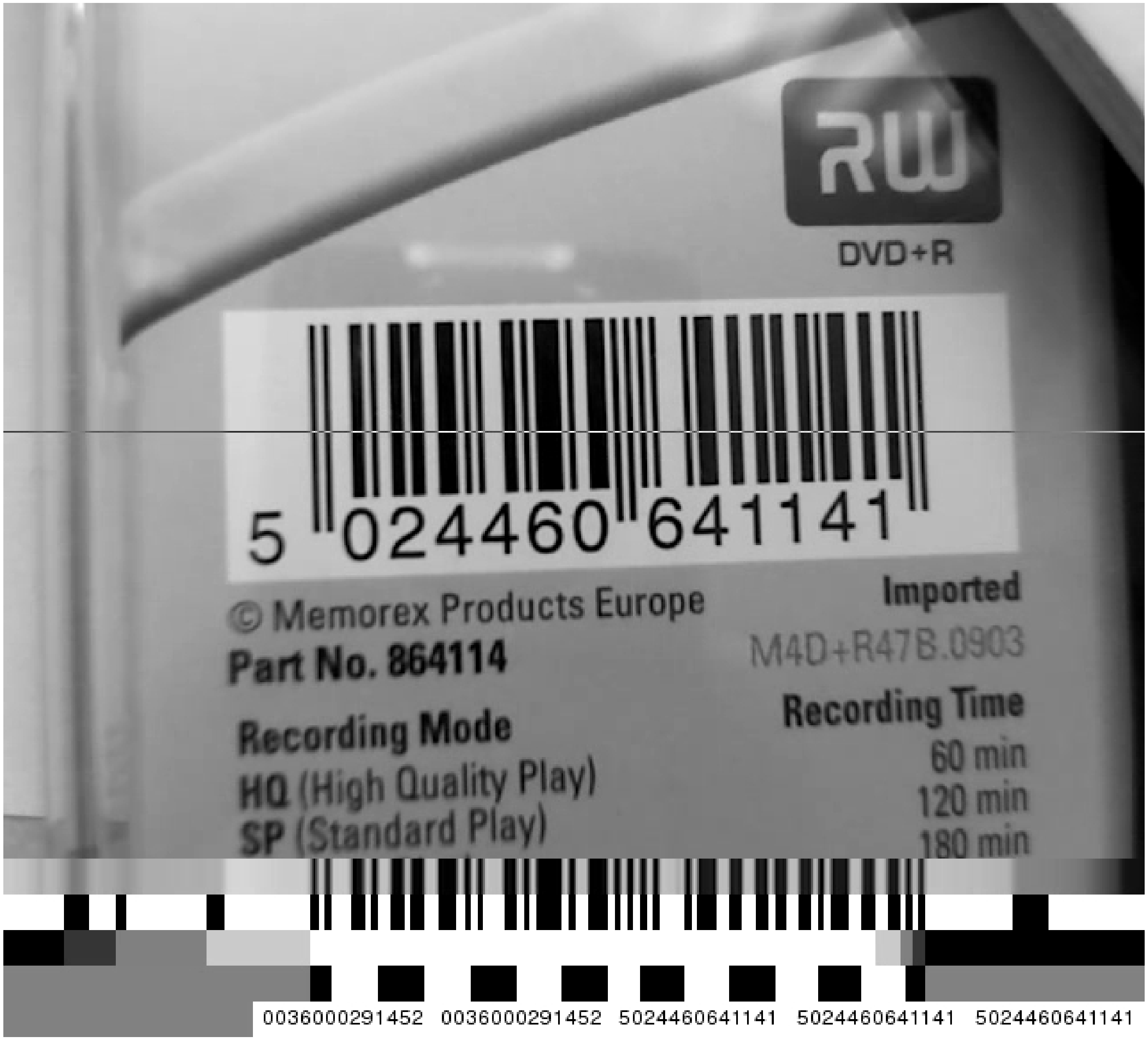
Camshift Tracking



http://www.wedesoft.demon.co.uk/hornetseye-api/files/camshift-txt.html

Other Examples

Barcode Reader

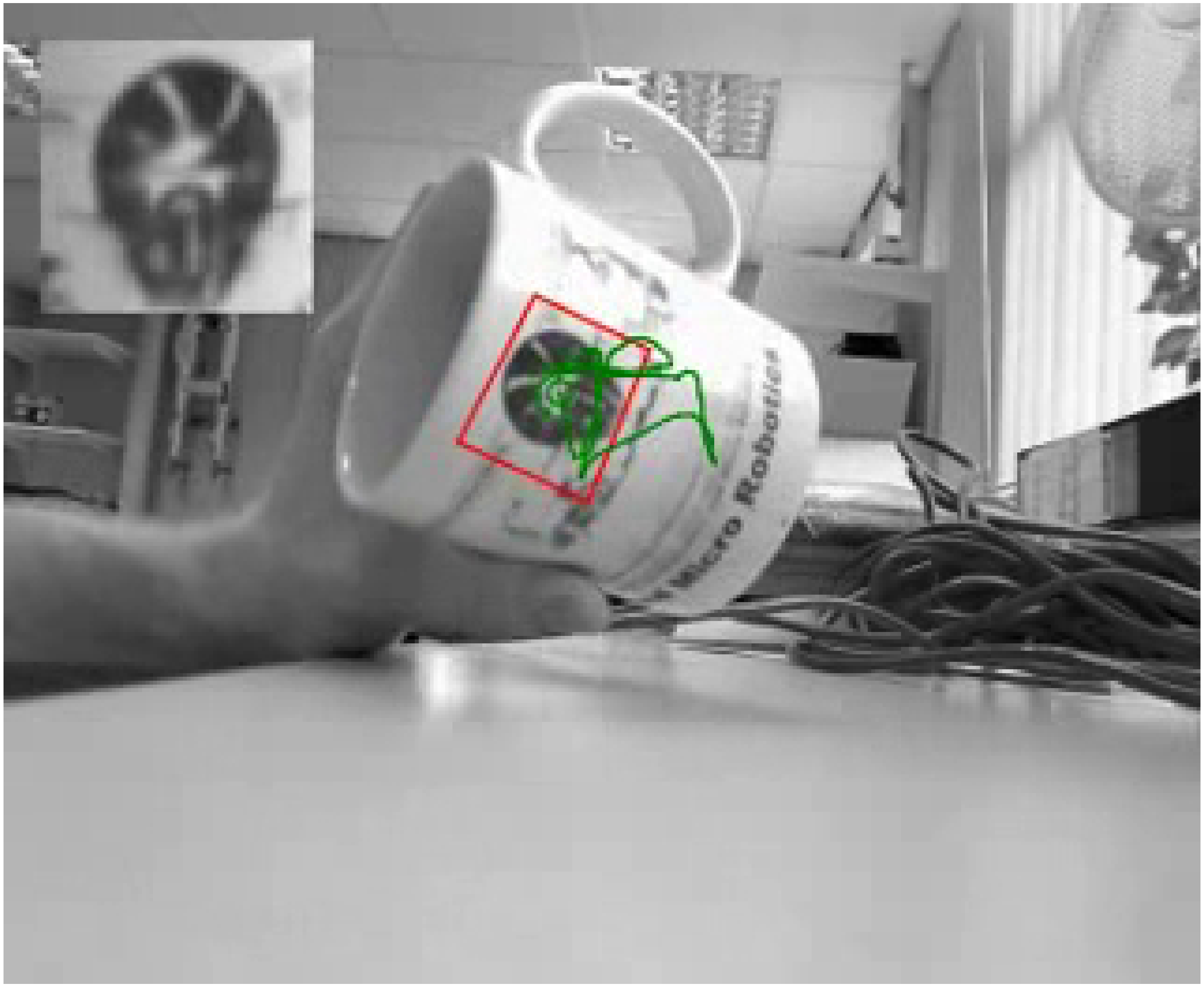


```
0036000291452  0036000291452  5024460641141  5024460641141  5024460641141
```

http://www.wedesoft.demon.co.uk/hornetseye-api/files/barcode-txt.html

Other Examples

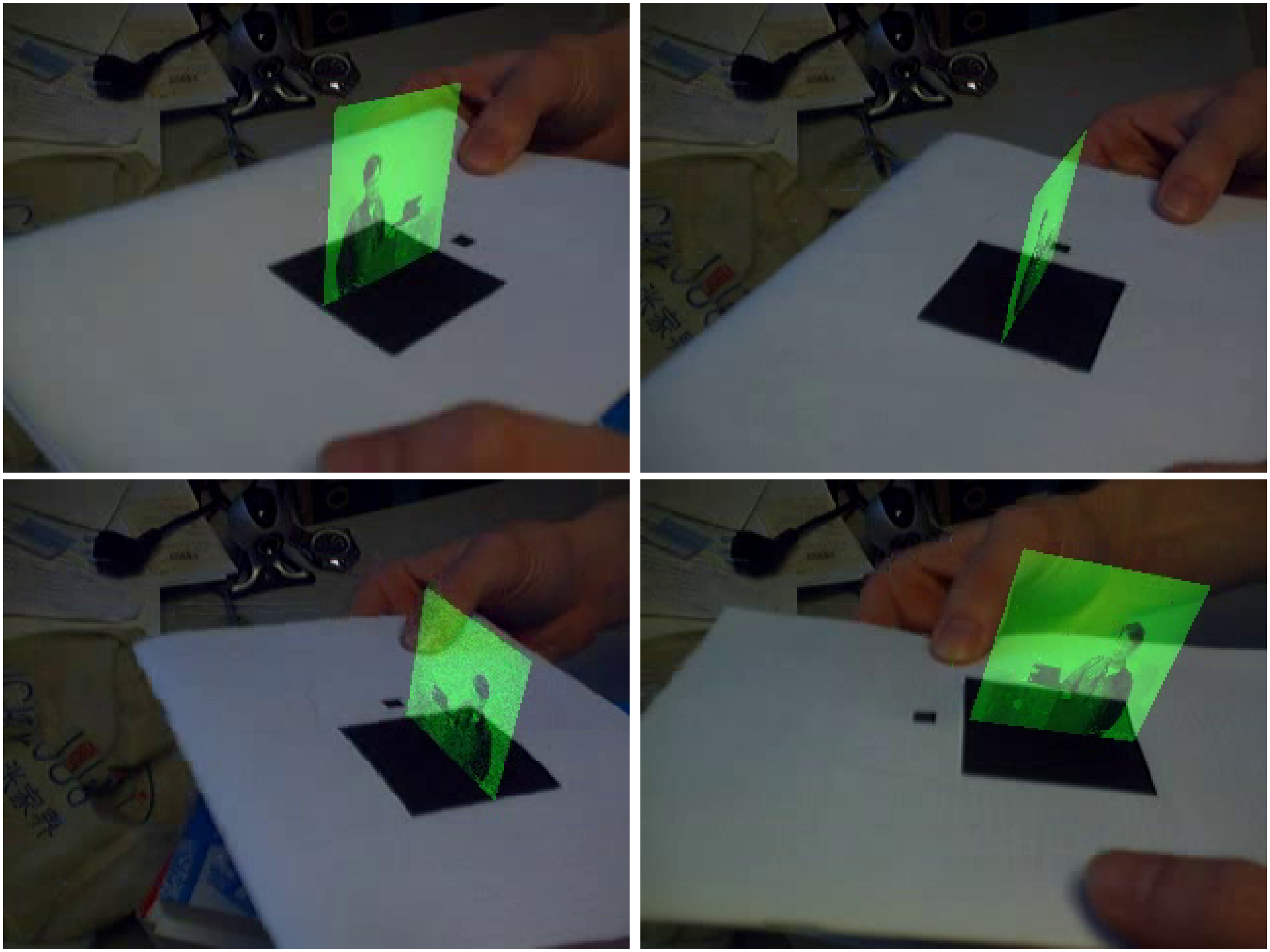
Lucas Kanade Tracker



http://www.wedesoft.demon.co.uk/hornetseye-api/files/lktracker-txt.html

Other Examples

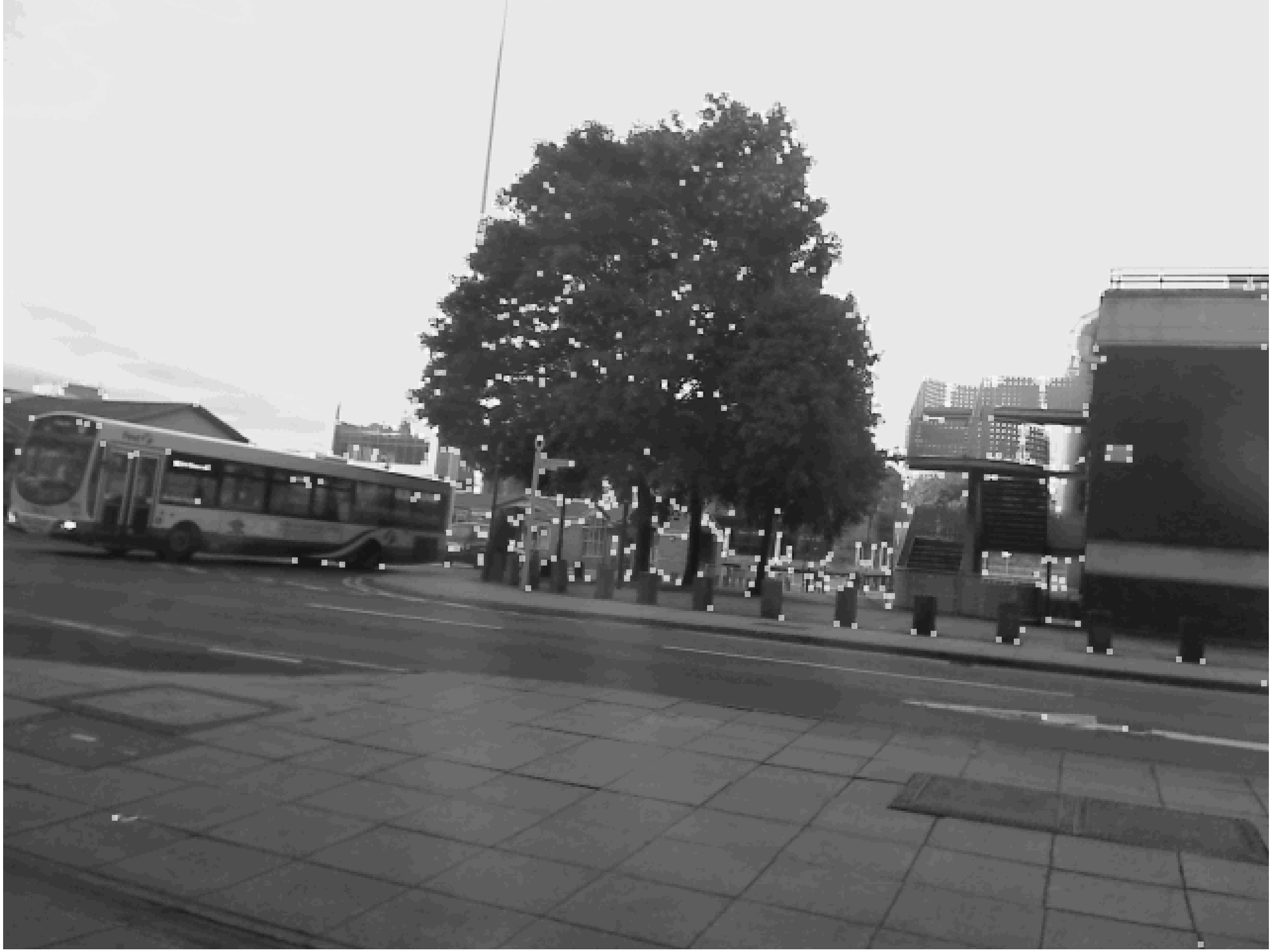
Planar Marker Tracking



http://rubyconf2009.confreaks.com/
19-nov-2009-13-15-computer-vision-using-ruby-and-libjit-jan-wedekind.html

Future Work

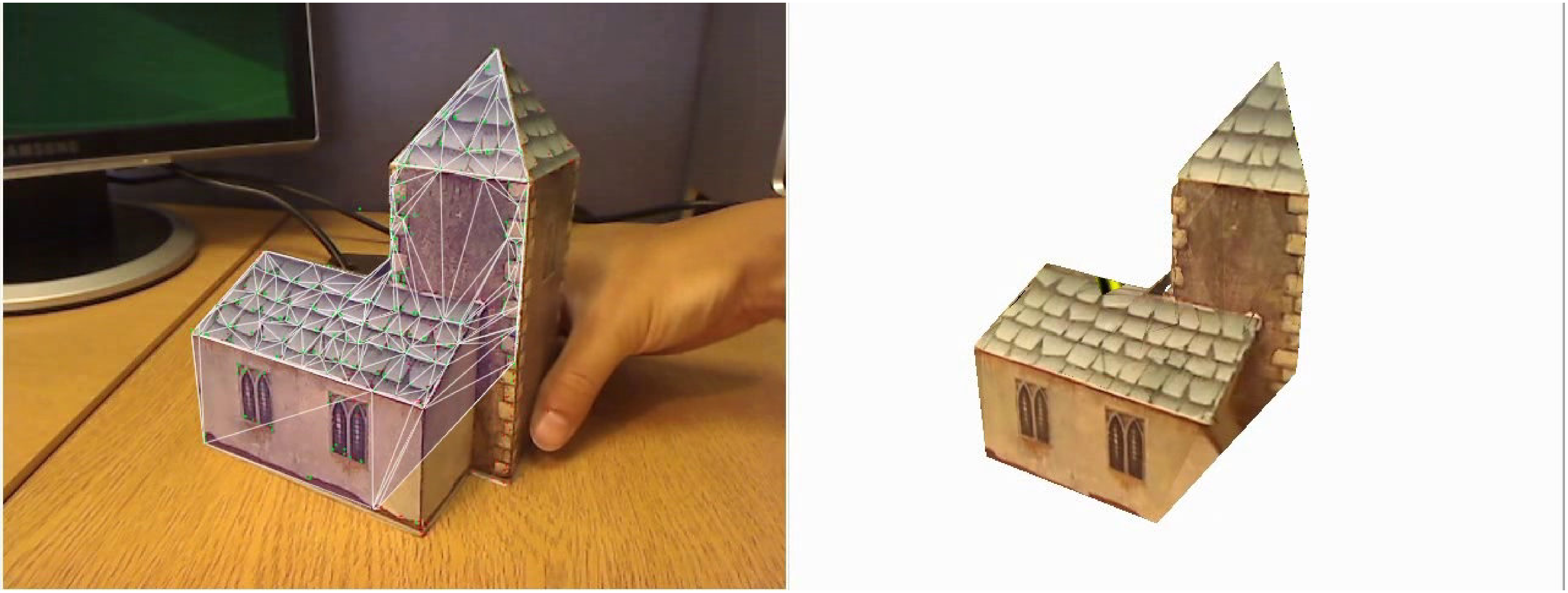
Feature Locations and Descriptors



http://www.wedesoft.demon.co.uk/hornetseye-api/files/features-txt.html

Future Work

Inspiration: Probabilistic Feature-based On-line Rapid Model Acquisition

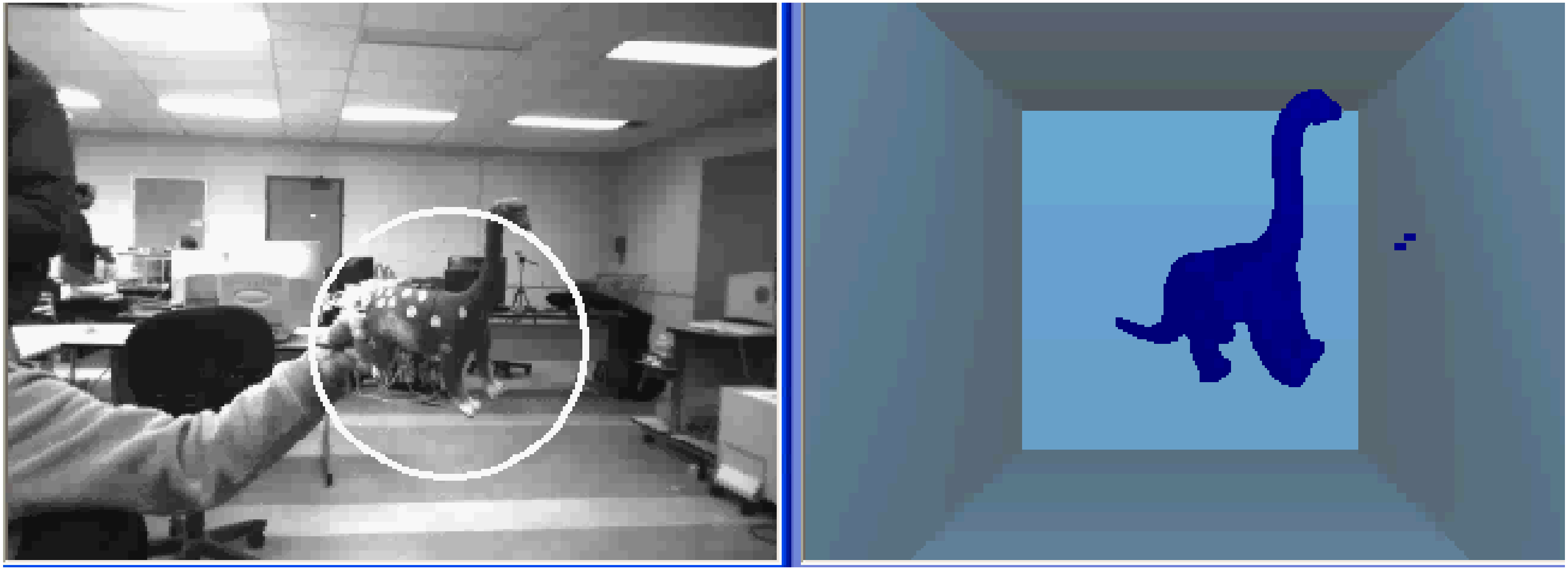


http://mi.eng.cam.ac.uk/~qp202/my_papers/BMVC09

http://mi.eng.cam.ac.uk/~twd20/

Future Work

Inspiration: Bounded Hough Transform

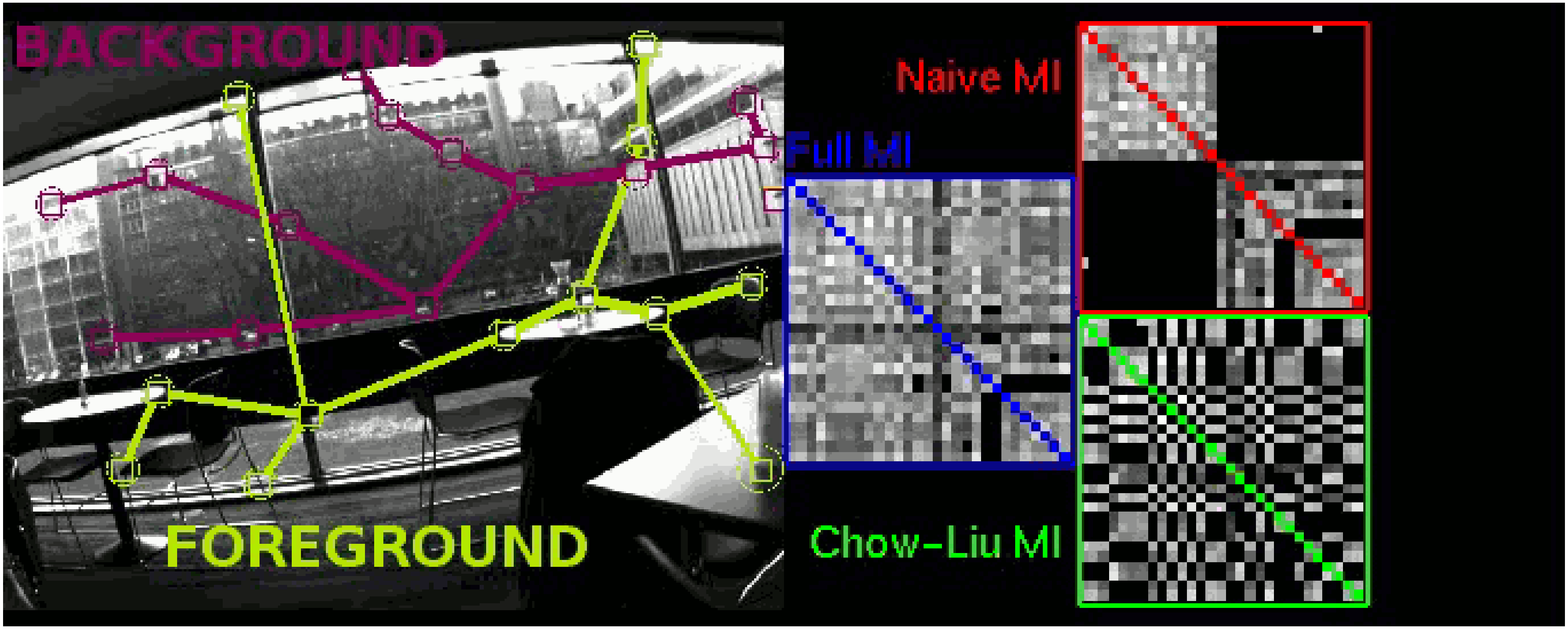


http://www.ptgrey.com/newsletters/dec2004.html

http://www.ptgrey.com/newsletters/images/GreShaJas04.pdf

Future Work

Inspiration: SceneLib



http://www.doc.ic.ac.uk/~ajd/

Future Work

Inspiration: Avatar

http://seqmag.com/2010/01/making-of-avatar/

http://seqmag.com/2010/01/exclusive-45minute-making-of-avatar/

Thanks

Credits

Aiden Lockwood, Aleksey Demakov, Annemie Wedekind, Arul Nirai Selvan, Ashley Moran, Balasundram Amavasai, Beverly Inkson, Chinwe Lucy Ozoegwu, Damien Douxchamps, Daniel Martín Marín, Géraud De La Mensbruge, Gerhard Wedekind, Hussein Abdul-Rahman, Jacques Penders, Jag Gill, Jing Jing Wang, Jon Travis, Jong Peng, Juan Roldan, Julien Demarest, Julien Faucher, Julien Lacheray, Ken Dutton, Kim Chuan Lim, Kirill Kononenko, Klaus Treichel, Manuel Boissenin, Martin Howarth, Matthias Stumpf, Michael Doronin, Ralph Gay, Richard Dale, Sonia Fernández Rodríguez, Tan Kang Song, Ushakiran Soutapalli, Volkan Karaca, Warren Jasper, Zineb Saghi, ...

http://www.wedesoft.demon.co.uk/hornetseye-api/

http://rubyforge.org/projects/hornetseye/

http://sourceforge.net/projects/hornetseye/

http://launchpad.net/hornetseye/

http://raa.ruby-lang.org/project/hornetseye/

http://www.ohloh.net/p/hornetseye/