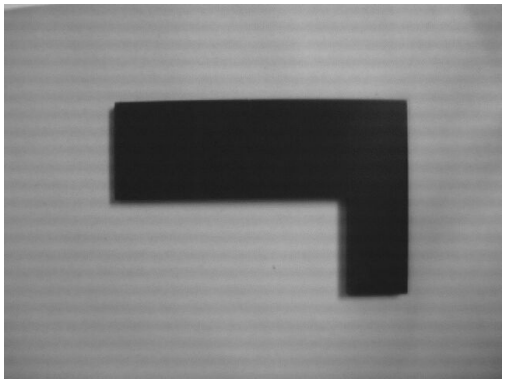
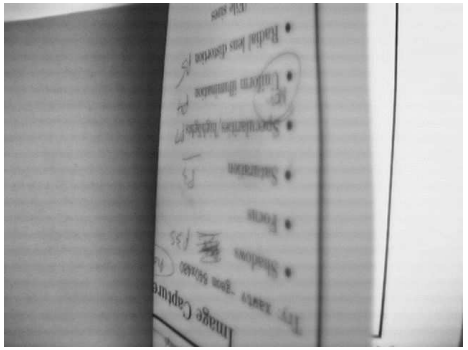
Image Capture and Problems



A reasonable capture

Image Capture: Focus problems



Focus set to one distance. Nearby distances in focus (depth of focus). Further not so well focused. Compare 'identical' lines.

Image Capture: Shadow problems

False color to emphasize the shadow location.
Often hard to separate from part.

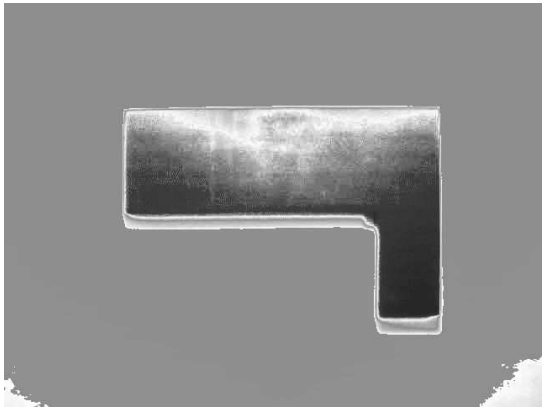
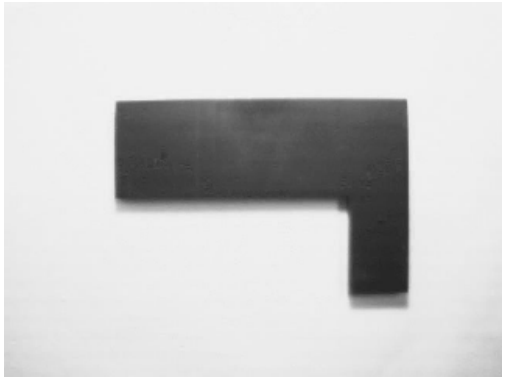
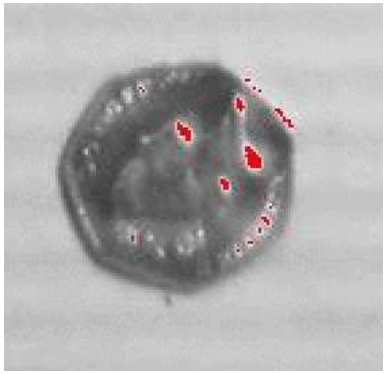


Image Capture: Saturation problems



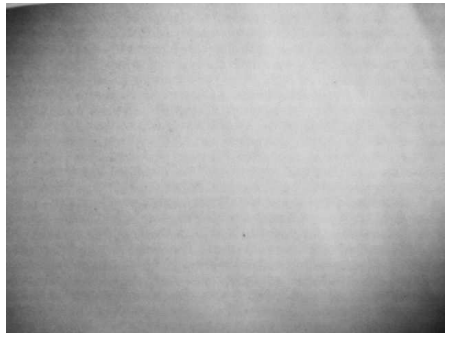
Pixels clip at 255.

Image Capture: Specularities/highlights



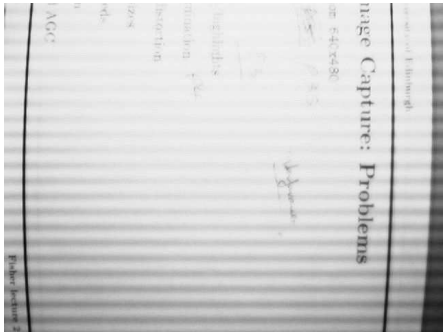
Saturated pixels set to red.

Image Capture: Non-uniform illumination



Contrast on background enhanced: may cause analysis problems.

Image Capture: Radial lens distortion



Note 'straight' lines at image edge. May make accurate measurements hard.

Image Capture: Overcoming Problems

- **Shadows, specularities, non-uniform illumination**: increase ambient lighting by using light diffusing panels or lots of point lights
- **Depth of Focus**: use smaller aperture and brighter light
- **Motion Blur**: use shorter capture time and brighter light
- **Saturation**: use smaller aperture, reduce gain and adjust gamma

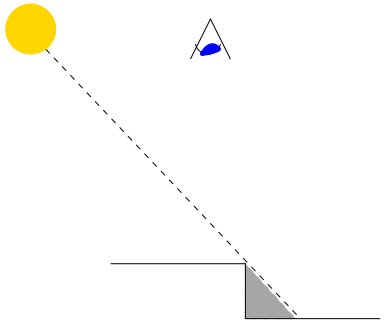
- **Lens Distortion**: more expensive lenses, view from further away
- **Aliasing**: use incandescent lights

Illumination control techniques

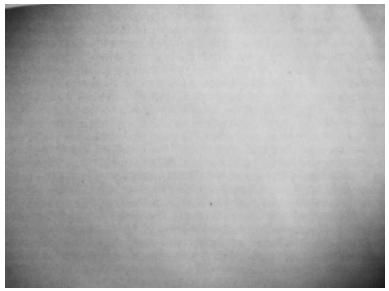
Main cause of problem: point light sources

Brightness = B / (surface distance from source)^2

Sharp shadows:

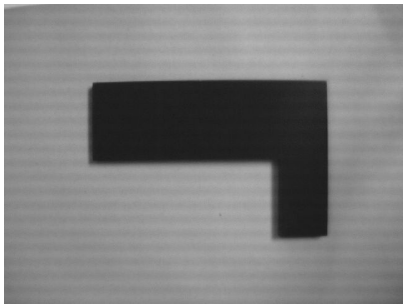


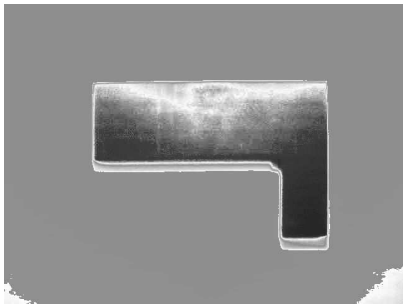
Strong illumination variations



Shadow Example

Figure and shadow at bottom left emphasized

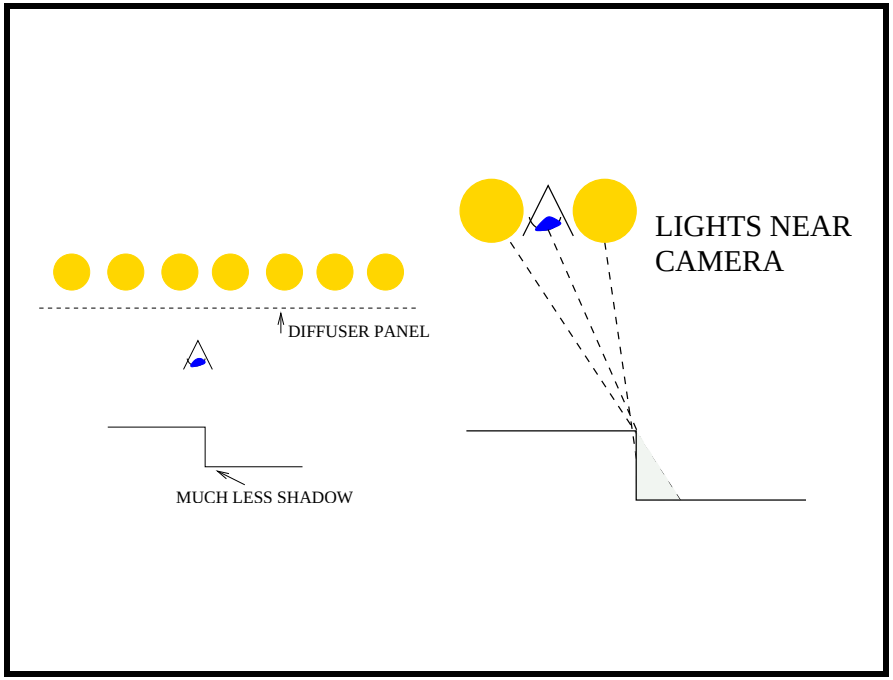




Lighting control

To reduce complications arising from illumination:

- Increase ambient (all direction) light with light diffuser panels
- Illumination by camera to move shadows to non-visible places
- Backlighting panel



Isolating flat parts

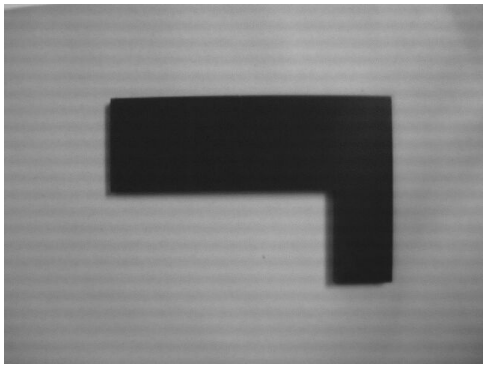
Isolate parts, then characterise later

Assume

- Dark part
- Light background
- Reasonably uniform illumination $->$ distinguishable parts

Midlecture Problem

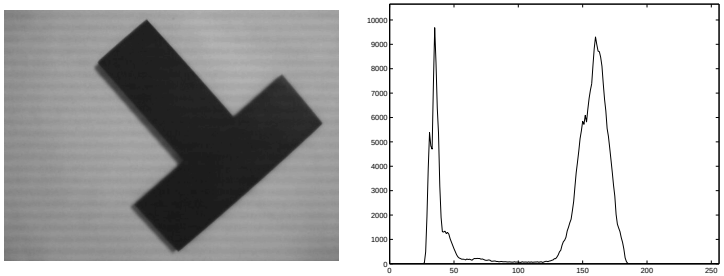
Given this image, how might we label pixels as object and background?



Thresholding Introduction

Key technique: thresholding
Assume pixel values are separable

Part and typical distribution

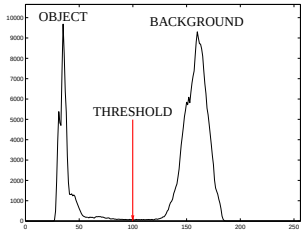


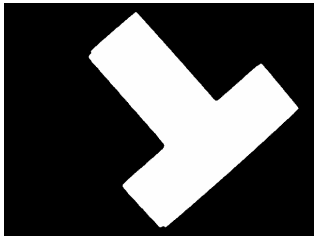
Spread: not quite uniform illumination + part color variations + sensor noise

Thresholding

Thresholding: central technique

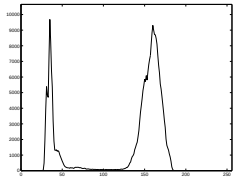
```
for row = 1 : height
 for col = 1 : width
  if value(row,col) < ThreshHigh % inside high bnd
  % & value(row,col) > ThreshLow % optional low bnd
      output(row,col) = 1;
  else
      output(row,col) = 0;
  end
```



Threshold Selection

Exploit bimodal distribution



But:

- Distributions broad and some overlap − > misclassified pixels
- Shadows dark so might be classified with object
- Distribution has more than 2 peaks

So: smooth histogram to improve shape for selection

Convolution

General purpose image (and signal) processing function

Computed by a weighted sum of image data and a fixed mask

Linear operator: conv(a*B,C) = a*conv(B,C)

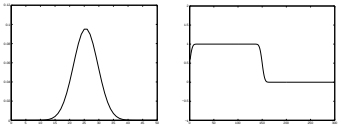
Used in different processes: noise removal, smoothing, feature detection, differentiation, ...

Convolution in 1D

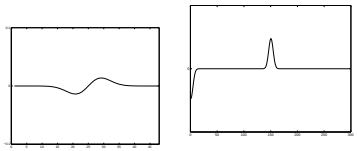
$$Output(x) = \sum_{i=-N}^{N} weight(i) * input(x-i)$$

Input:

Gaussian Mask and Output:

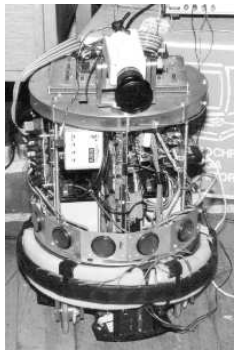


Derivative of Gaussian Mask and Output:



2D Convolution - Smoothing

$$Output(x,y) = \sum_{i=-N}^{N} \sum_{j=-N}^{N} weight(i,j) * input(x-i, y-j)$$

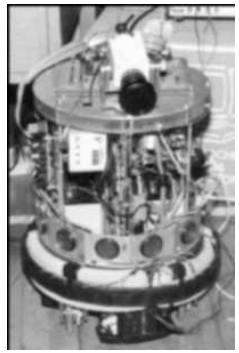


$*$

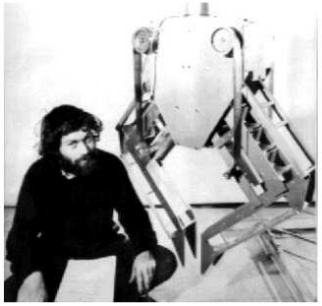
$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

$=$



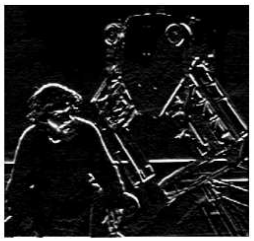
Convolution for Edge Detection



$*$

1	2	1
0	0	0
-1	-2	-1

$=$



Edge detection

$*$

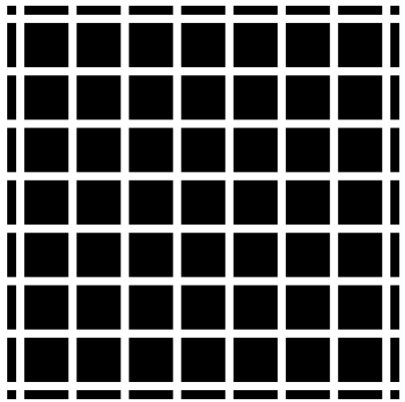
1	0	-1
2	0	-2
1	0	-1

$=$

Convolution 'Explains' Illusions

Edge enhancement in human vision

Centre-surround receptors - convolved with retinal image

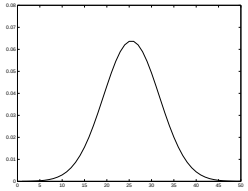


Hermann grid illusion – full explanation more complex

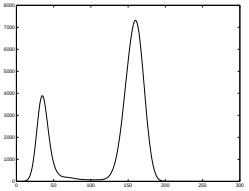
Histogram Smoothing for threshold selection

Histogram Smoothing (in `findthresh.m`)
Convolve with a Gaussian smoothing window

```
filterlen = 50;                          % filter length
thefilter = gausswin(filterlen,sizeparam); % size=4
thefilter = thefilter/sum(thefilter); % unit norm
tmp2=conv(thefilter,thehist); % makes longer output
% select corresponding portion
offset = floor((filterlen+1)/2);
tmp1=tmp2(offset:len+offset-1);
```

FILTER SHAPE



SMOOTHED HISTOGRAM

What We Have Learned

1. Image Capture Problems and Fixes
2. Differentiating object from background
3. Convolution
4. Histogram smoothing & threshold selection