Thresholding Based Segmentation

Robert B. Fisher
School of Informatics
University of Edinburgh

Isolating flat parts

Isolate parts, then characterise later

Assume
- Dark part
- Light background
- Reasonably \[ \text{illumination} \rightarrow \text{distinguishable parts} \]

Motivating Example

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

Thresholding Introduction

Key technique: thresholding
Assume pixel values are

Part and typical distribution

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

Thresholding: 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
    end
end

Threshold Selection 1

Exploit 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 sum of image data and a fixed mask

Linear operator: \( \text{conv}(a \ast B, C) = a \ast \text{conv}(B, C) \)

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

---

**Histogram**

**for Threshold Selection**

Histogram Smoothing (in `findthresh.m`)

Convolve with a Gaussian smoothing window

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

---

**Convolution in 1D**

\[ \text{Output}(x) = \sum_{i=-N}^{N} \text{weight}(i) \ast \text{input}(x - i) \]

Input:

Gaussian Mask and Output:

---

**Convolved Histogram Example**

FILTER SHAPE  SMOOTHED HISTOGRAM
Threshold Selection 2
Assume 2 big peaks, brighter ? is higher:
1. Find biggest peak (background)
2. Find next biggest peak in darker direction
3. Find lowest point in trough between peaks

Peak Pick Code
Omit special cases for ends of array and closing ‘end’s.

```
peak = find(tmp1 == max(tmp1));  % find largest peak
% find highest peak to left
xmaxl = -1;
for i = 2 : peak-1
    if tmp1(i-1) < tmp1(i) & tmp1(i) >= tmp1(i+1) ...
        & tmp1(i)>xmaxl
        xmaxl = tmp1(i);
        pkl = i;
    % find deepest valley between peaks
```

Lecture Overview
1. Thresholding to differentiate object from a constant and simple background (not just white backgrounds: see also bluescreening or chroma keying)
2. 1D
3. Histogram smoothing & threshold selection