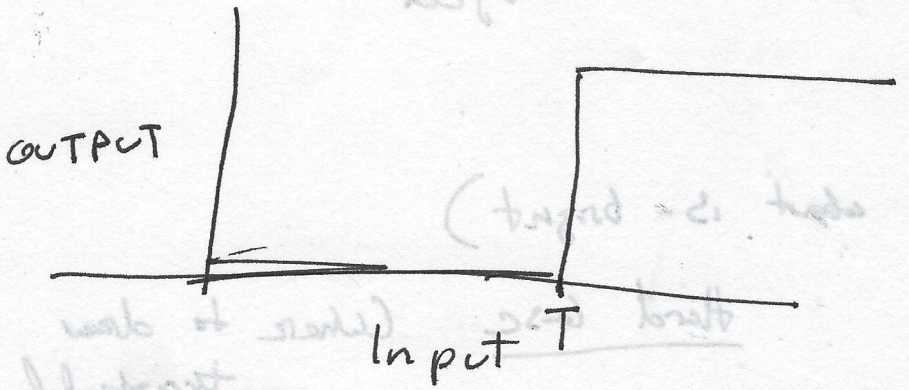
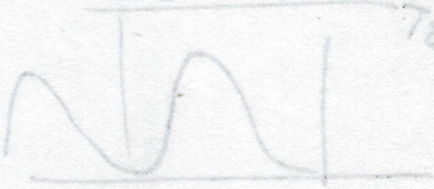
Image Processing (In class)

Lecture 12

Thresholding



Histogram ?



MatLab:

im

imshow(im(30,:))

raise to 50, we have some unwanted stuff

We can get a good threshold

best we must automatic thresholding

No humans needed

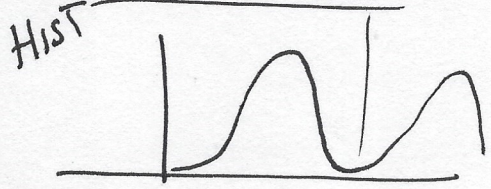
(2)

$$g(x,y) = \begin{cases} 1 & \text{IF } f(x,y) > T \\ 0 & \text{IF } f(x,y) \leq T \end{cases}$$

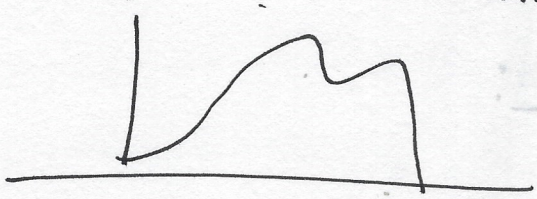
Generally this is used for object segmentation

* (stuff we care about is bright)

Easy Case



Hard Case (where to draw threshold)



We Talk of Global or Local thresholding

→ Mat lab

(*) Thresholding is hard

```
[X, Y] = meshgrid(1:201, 1:201);
im = (X-101).^2 + (Y-101).^2 < 50^2;
imshow(im, [])  - easy
imn = imnoise(im, 'gaussian', 0, 0.02);  add noise
imshow(imn, [])
imshow(imn > 0.5)  OK
```

Increase σ to 0.2

→ we have a bad result

Median filter ?

```
g = ones(201,1) * linspace(-.5, .8, 201)';

img = im + g

imshow(img > 1/2, [])
(img > .8, [])
```

> we lose parts of the circle

Note:

Thresholding is affected by:

- Contrast Between object and Background
- Image Noise
- Relative size of object
- Variability in Illumination
- Variability of object reflectance

How to choose a good T ?

("best" T) How to tell when it's good?

Relative?

In Easy Cases pick one visually

Most commonly used Method:

OTSU's Algorithm

MATLAB: graythresh

IDEA: Separate the intensities into 2 clusters that are as TIGHT as possible.

Based on the Image Histogram

P_i = { Prob. that $f(x,y) = i$ }

$i = 0, 1, \ldots L-1$

Suppose we set a threshold K.

$C_1 = \{ (x,y) \mid f(x,y) \le K \}$

$C_2 = \{ (x,y) \mid f(x,y) > K \}$

$$P_1 = \sum_{i=0}^{k} p_i \qquad P_2 = \sum_{i=k+1}^{L-1} p_i$$

$$= 1 - P_1$$

Mean of Pixels in each class:

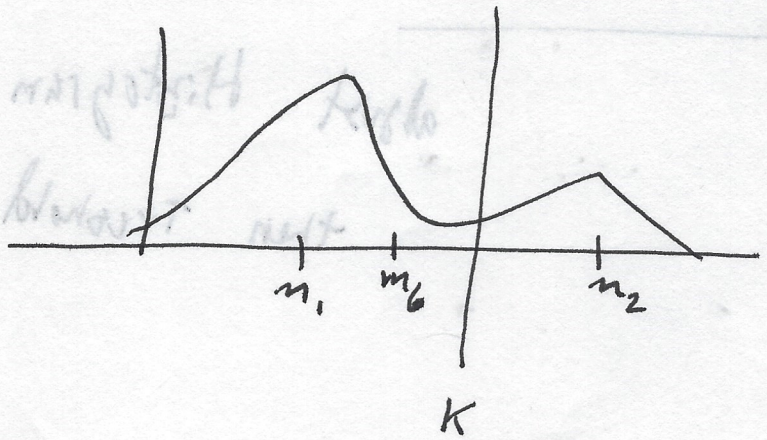
$$m_1 = \frac{\sum_{i=0}^{k} i\,p_i}{P_1} \qquad m_2 = \frac{\sum_{i=k+1}^{L-1} i\,p_i}{P_2}$$

Overall mean and variance (Global)

$$m_G = \sum_{i=0}^{L-1} i\,p_i$$

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i$$

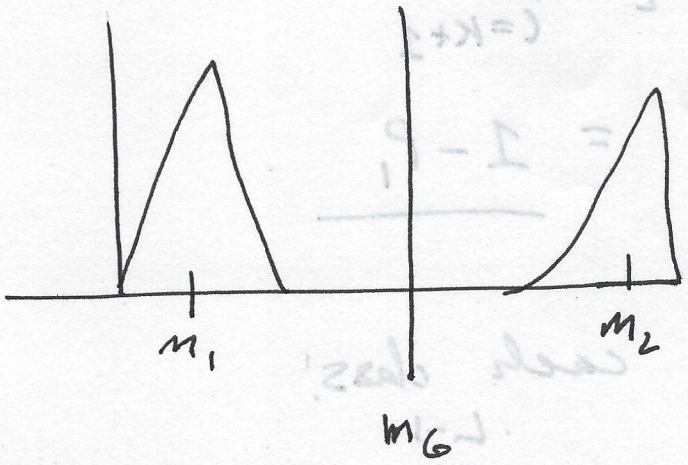
Histogram



OTSU's IDEA: Maximize the between-class variance

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$

Find T to maximize this number



The Measure

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

Is a Good Measure of Separability

(Large: Clusters Far Apart
Small: Clusters Close)

↑ effectiveness metric?

doc graythresh

Matlab
im = sunset

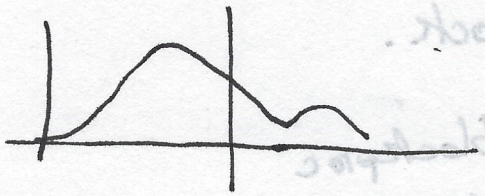
adjust Histogram then Treshold

From MatLab Example:

OTSU Gives a Reasonable Automatic estimate (NoT Perfect)

Issues:

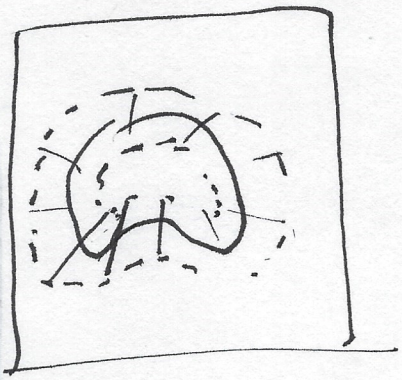
- No strong Peaks in histogram
- object is very small



IMProving Global Thresholding

- Low-Pass Filter, Then apply OTSU (mitigate Noise)

- ONLy Consider Pixels near Edges
  - compute EDGE Image
  - collect Pixels of $f(x,y)$ on or (say $\pm 1, 2$) away from Edge
  - Determine OTSU threshold using only These Pixels



(8) # Local \Variable Thresholding

There may be No Good Global Threshold (eg. Illumination Gradient)

Simple Idea: SPLIT IMAGE INTO blocks, Apply OTSU Independently I EACH Block.

MATLAB: blockproc

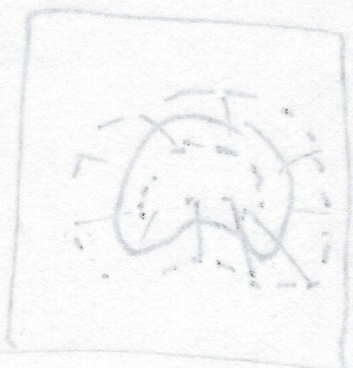
otsublock.m

```
out = blockproc(im, [100 100], @otsublock);
```

↳ output thresholds for each block

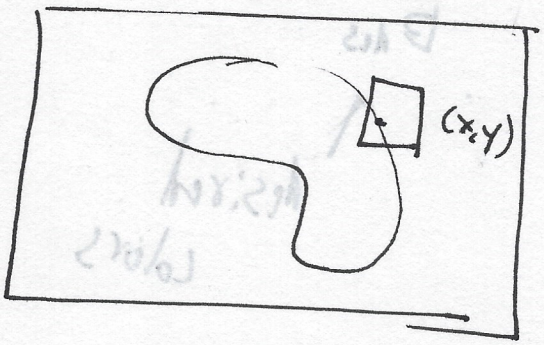
brighter areas have higher thresholds

↳ Turn down block size for more action!



Instead of using OTSU on every block

Alternatively, could Apply a Local Threshold based on local statistics of block



AT every Pixel (x, y), Create a Neighborhood (block S_{xy}

Compute the mean μ_{xy} and STD Deviation σ_{xy}

$$g(x,y) = \begin{cases} 1 & f(x,y) > [\mu_{xy} + 2\sigma_{xy}] \\ 0 & \text{Else} \end{cases}$$

The Condition (or "Predicate") could be

$$f(x,y) > \mu_{xy}$$

$$|f(x,y) - \mu_{xy}| > 2\sigma_{xy}$$

$$f(x,y) > \mu_{xy} + \sigma_{xy} \text{ and } f(x,y) > T$$

(10)

Color Thresholding

Create a Distance function

$$d(x,y) = \| f(x,y) - C_{des} \|$$

R	R_{des}
G	G_{des}
B	B_{des}

↑ desired colors

distance mapping

dark is close

Light is far away

— sphere near it

Threshold outside

may be use HSI

Talk to TA about Project