Image Processing (In class)

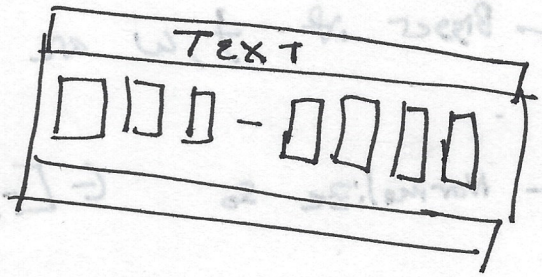
Lecture 14

OBJECT DETECTION / FEATURE DETECTION

We know (e.g.) How to detect STRAIGHT LINE Segments w/ Hough (could APPLY To finding Circles, Rectangles)

⊗ Many objects of Interest aren't So Simple

Eg License Plate



← hard with hough

Template Matching

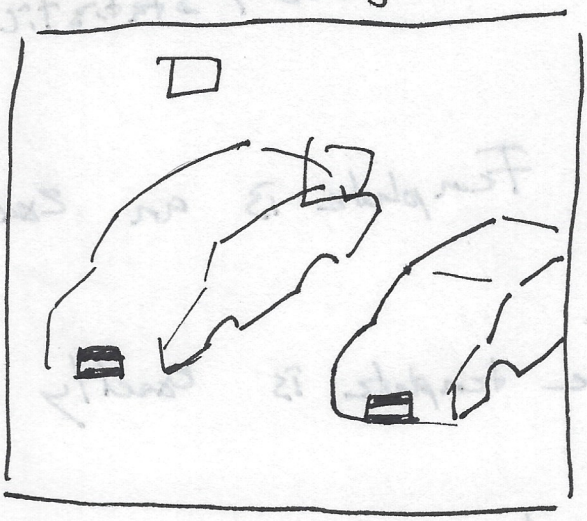
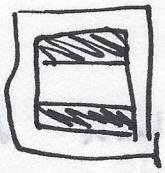


IMAGE $f(x,y)$



TEMPLATE

Idea:

Slide template around Image Looking for places that match

→ correlation (Difference)

(2) ⇒ Perhaps Look for Same Variances in Colors

Basic Approach: Correlation coefficient

$$\gamma(x,y) = \frac{\left(\sum_s \sum_t \left[(w(s,t) - \bar{w})(f(x+s, y+t) - \bar{f}_{xy}) \right] \right)}{\sqrt{\left(\sum_s \sum_t (w(s,t) - \bar{w})^2 \right) \left(\sum_s \sum_t (f(x+s, y+t) - \bar{f}_{xy})^2 \right)}}$$

$\bar{w}$ = Avg Value of template

$\bar{f}_{xy}$ = Avg value ~~surrounding~~ inside Image region surrounding (x,y)

$$\gamma(x,y) = \frac{Cov(f,w)}{\sigma_w \sigma_f}$$

← Bigger if f, w are similar

← Normalize so $\in [-1, 1]$

STANDARD Concept from Prob / Statistics

$\gamma(x,y) = 1$ when the Template is an Exact match

$\gamma(x,y) = -1$ when the template is Exactly opposite

$\gamma(x,y) = 0$ When No Correlation Between template and the Image Patch

- We Subtract off The means and normalize by the STD Deviations To Compensate for Illumination Changes.

```
im = imread ('many donut 5.jpg');
im = rgb2gray (im);

temp = imread ('many donut-temp.jpg');

doc normxcorr2 (template, A)

out = normxcorr2(temp, im);

imshow (abs(out), [])

or

No abs

imshow( out>0.9, [])  → 1 donut

imshow( out>0.6, [] ) → bigger blobs
                        → more donuts

findtemplate( im, temp, 0.7);
```

Rickett

(4) What if the template is Not the same size as the Image?

→ resize Image, I x 0.9

Then try again, → redo until it works.

findScaledtemplate 10:39 am

We can't always use template matching,

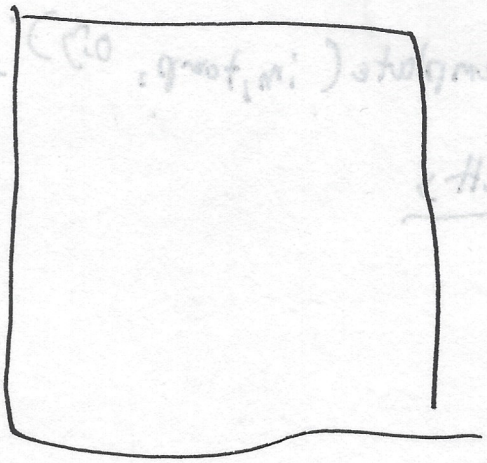
C.g. ENTIRE Object IN scene may Not be visible.

Instead, we Look for features of

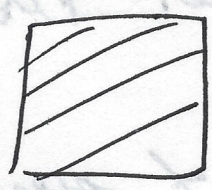
The object and try to match them.

What feature to choose?

Choose Distinctive Features of the Car



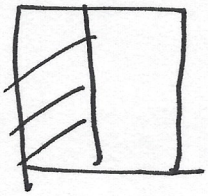
What ARE Good features?



- flat

Bad - no local distinctiveness

Low gradient

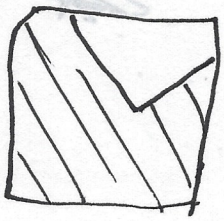


- Edge

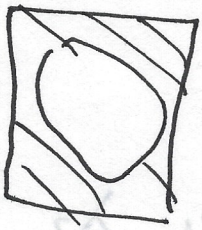
Bad, can slide up and down

gradient in one direction

gradient in all direction



- Corner



- blob

Good, can match the features well except at the right location

Algorithm to find Good features

How to Automatically find Good features?

Early Approach: Harris Corner detector.

Intuition: X and Y Gradients inside block should be large

(Should have large Gradients in 2 orthogonal Directions)

(c) following is called SHI-TOMASI DETECTOR

1) Compute g_x, g_y AT Each Pixel

2) for every NXN Block of pixels, Γ, Create the covariance matrix

$$H = \begin{bmatrix} \sum_{(x,y)\in\Gamma} g_x^2 & \sum_{(x,y)\in\Gamma} g_x g_y \\ \sum_{(x,y)\in\Gamma} g_x g_y & \sum_{(x,y)\in\Gamma} g_y^2 \end{bmatrix} \quad 2X2$$

3) Compute Eigenvalues of H, λ_1, λ_2

4) IF λ_1, λ_2 Both > threshold, Accept Γ as a Good FEATURE.

Once we have features on our object, we can match them in another image

Matching: Sum-of-Squared Differences

$$\sum\sum \left(\text{[image]} - \text{[image]} \right)^2$$

Low - a good match

- Normalized Cross-Correlation
- Nearest-Neighbour Distance Ratio

d_1 = distance to best match

d_2 = Distance to 2nd Best match

want d_1 is low, and $\frac{d_1}{d_2}$ low - for ~~Best~~ Best Match

---

Difficulties in feature Matching:

- Scale Invariance
- Rotation Invariance
- Perspective Invariance

(6) DEAL with Scale Invariance:

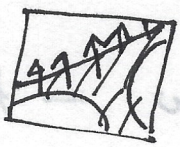
Estimate "Natural" Scale around Each good Feature

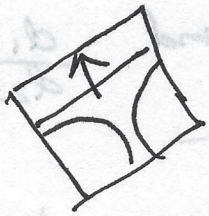
DEAL With Rotation Invariance:

Estimate Elliptical Region around feature instead of a square

Dealing with Rotation:

- Estimate Dominant Gradient (orientation) of Rotation





Scale + Rotation Invariance is obtained with the SIFT ~~DETECTION~~ Detector / Descriptor

Descriptor: Not Pixel intensities, but histograms of Gradient orientations inside subblocks

Descriptor: Not pixel intensities, but Histograms of Gradient orientations inside Subblocks

Sift features → Image object detection

match('scene.pgm', 'box

~~OpticalFlow~~ Optical flow