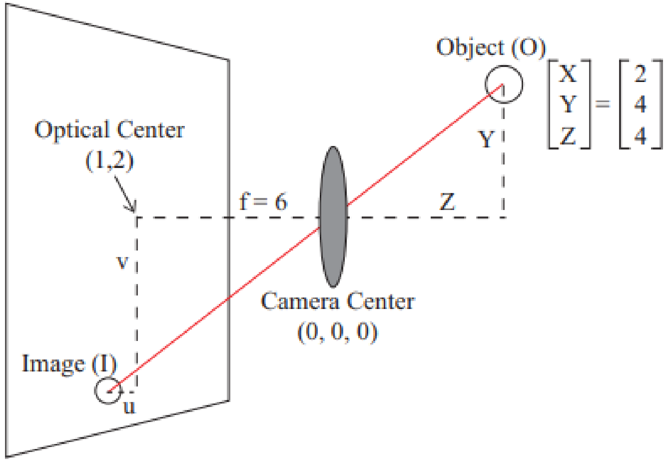
Exam 1 Review

CS 4476 Fall 2025

Digit Images

Consider the following image formation scenario for a camera with no skew, no rotation, and a square aspect ratio.



(e) (6 points) Specify the intrinsic matrix, K, rotation matrix, R, and translation vector t for the imaging scenario above:

$K =$

$R =$

$t =$

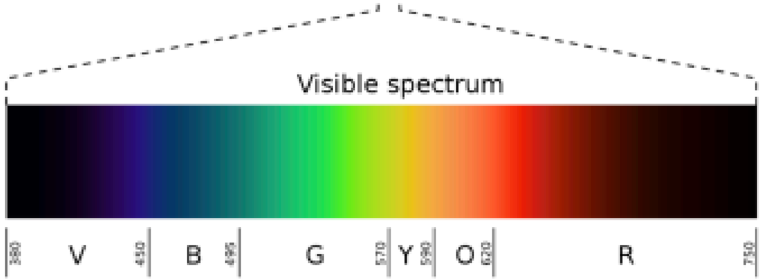
Solution:

$$K = \begin{bmatrix} 6 & 0 & 1 \\ 0 & 6 & 2 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Digit Images

- (204, 101, 28)
- (150, 220, 55)
- (157, 145, 61)
- (0, 345, 230)

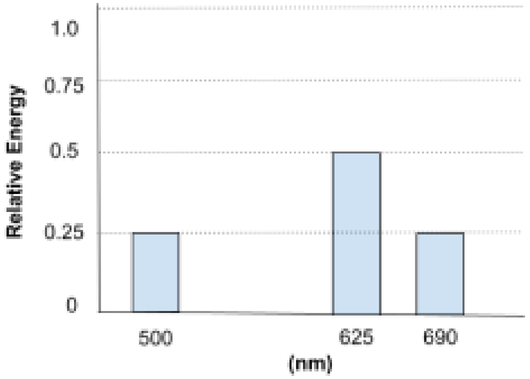
Question 1 1 pts



A color matching experiment was run to determine RGB values for the 3 wavelengths of 500nm, 625nm, and 690nm. The results of this experiment are given below:

- Cyan: 500 nm [0, 254, 244]
- Orange: 625 nm [252, 164, 0]
- Dark Red: 690 nm [124, 0, 0]

You have found a mysterious color's spectogram. It constitutes the colors cyan, orange, and dark red with the following relative energies.



Determine the RGB values of the mysterious color under the color matrix found through color matching above.

Filters

Filter A	Filter B	Filter C
$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

difference
(vertical Sobel)

sharpen

Gaussian
blur

$$10\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

impulse
(no change)

box/average
blur

Filters

Filter A	Filter B	Filter C
$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Original Image

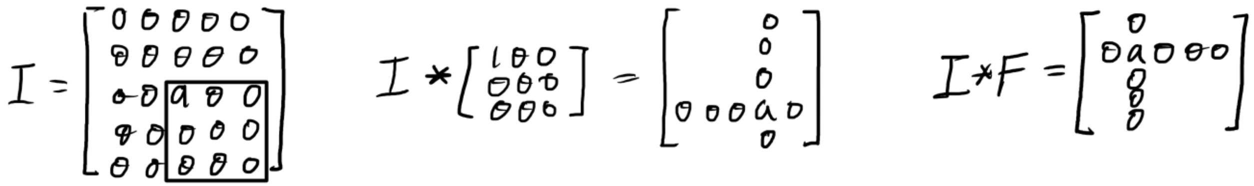
Image A

Image B

Which filter generated image A? C

Which filter generated image B? B

Filters



Write a 3x3 filter that shifts an image up 1 pixel and left 1 pixel when applying cross-correlation.

$$F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

When computing image derivatives, why do we apply a Gaussian filter before a differencing filter?

reduce high frequency noise in the image and avoid spurious edges

We want to perform edge detection using a simple gradient magnitude threshold of 7. At some pixel *p*, the x and y derivative filter outputs are 5. Is *p* part of an edge?

yes

$$\sqrt{5^2+5^2} = \sqrt{50} > \sqrt{49} = 7$$

Features

1. You are given an interest point detector that can choose regions of interest at different scales and rotations. Now you need to select an algorithm to describe these regions. You have both RGB pixel values and per-pixel gradient angles, re-oriented by the direction of max gradient. You consider the following:

   - **A:** RGB pixel values concatenated into a vector
   - **B:** A normalized histogram of RGB pixel values
   - **C:** Re-oriented gradient angles concatenated into a vector
   - **D:** A normalized histogram of re-oriented gradient angles

   To help you make your choice, select all features that satisfy the following properties.

   i. Invariant to scaling

   ☐ A ☐ B ☐ C ☐ D

   ii. Invariant to rotations

   ☐ A ☐ B ☐ C ☐ D

   iii. Retains local geometric information

   ☐ A ☐ B ☐ C ☐ D

   iv. Invariant to photometric variations

   ☐ A ☐ B ☐ C ☐ D

Features

2. Recall the Harris corner detector starts by computing M by summing over a windowed region, w, to form the following 2×2 matrix:

$$M = \begin{bmatrix} \sum_w \frac{dI^2}{dx} & \sum_w \frac{dI}{dx}\frac{dI}{dy} \\ \sum_w \frac{dI}{dx}\frac{dI}{dy} & \sum_w \frac{dI^2}{dy} \end{bmatrix}$$

Given M above and a particular threshold, t, indicate which of the following are necessary for an axis-aligned corner to be present? (Mark all necessary)

- ☐ $\sum_w \left(\frac{dI}{dx}\right)^2 \approx 0$
- ☐ $\sum_w \left(\frac{dI}{dy}\right)^2 \approx 0$
- ☑ $\sum_w \frac{dI}{dx}\frac{dI}{dy} \approx 0$
- ☑ $\sum_w \left(\frac{dI}{dx}\right)^2 > t$
- ☑ $\sum_w \left(\frac{dI}{dy}\right)^2 > t$
- ☐ $\sum_w \frac{dI}{dx}\frac{dI}{dy} \gg \sum_w \left(\frac{dI}{dy}\right)^2$
- ☐ $\sum_w \left(\frac{dI}{dy}\right)^2 \gg \sum_w \left(\frac{dI}{dx}\right)^2$
- ☐ $\sum_w \left(\frac{dI}{dx}\right)^2 \gg \sum_w \left(\frac{dI}{dy}\right)^2$
- ☑ $\sum_w \left(\frac{dI}{dx}\right)^2 \approx \sum_w \left(\frac{dI}{dy}\right)^2$
- ☐ $\sum_w \frac{dI}{dx}\frac{dI}{dy} > t$

Features

3. Recall that the general version of the Harris corner detector computes a *cornerness score* f as a function of the eigenvalues of M. Which property/properties does this step add to the Harris detector? *(select all that apply)*

   - [ ] Noise reduction
   - [x] Rotation invariance
   - [ ] Scale invariance
   - [ ] Selects strongly activated corners

4. Which factor/parameter in the Harris corner detector determines the scale of the corners detected?

   - [ ] The f-score threshold
   - [ ] The choice of derivative filter
   - [x] The window size

Features

5. If we have detected many neighboring pixels with high f-scores, what algorithm can we use to reduce the redundancy and select only a single corner pixel in that region?

Non-max suppression

Features

6. Given x, y gradients over a 3×3 windowed region, we will assess if a corner exists.

$$\frac{dI}{dx} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \frac{dI}{dy} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Recall the Harris corner detector starts by computing M by summing over a windowed region, w, to form the following 2×2 matrix:

$$M = \begin{bmatrix} \sum_w \frac{dI^2}{dx} & \sum_w \frac{dI}{dx}\frac{dI}{dy} \\ \sum_w \frac{dI}{dx}\frac{dI}{dy} & \sum_w \frac{dI^2}{dy} \end{bmatrix}$$

i. Compute the matrix M for the windowed region.

Solution:

$$M = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$

Transformations

Recall the following equations that describe a transformation of (x, y) into (x', y').

2D rotation around (0,0)	2D shear
x' $= x\cos\theta - y\sin\theta$	x' $= x + \alpha y$
y' $= x\sin\theta + y\cos\theta$	y' $= \beta x + y$

(a) (4 points) Please fill in the transformation type corresponding to the transformation matrix (a, b and θ are non-zero real numbers).

(a): ______________________

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(b): ______________________

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(c): ______________________

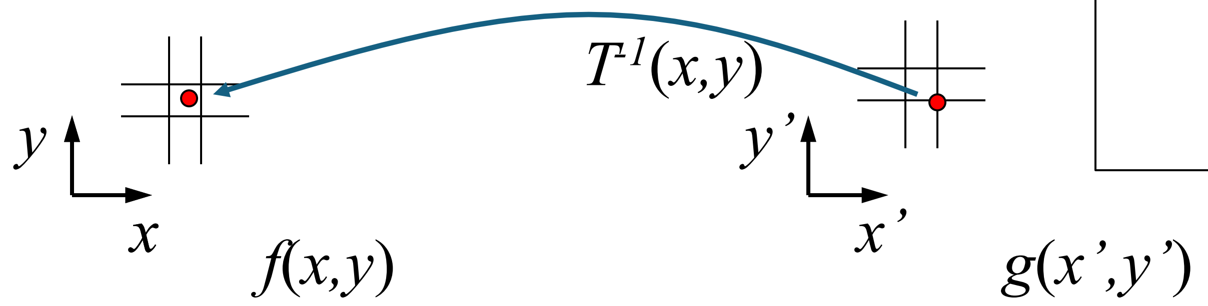
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(d): ______________________

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Solution:
(a): Shear (b): Rotate
(c): Scale (d): Translate

Inverse warping



Commonly suffers from less aliasing than forward warping

Get each pixel $g(x',y')$ from its corresponding location

$(x,y) = T^{-1}(x',y')$ in the first image

Q: what if pixel comes from "between" two pixels?

A: *Interpolate* color value from neighbors

– nearest neighbor, bilinear…

`>> help interp2`

Slide from Alyosha Efros, CMU

Transformations

Here is the RANSAC Algorithm:
Repeat $\boldsymbol{N}$ times.

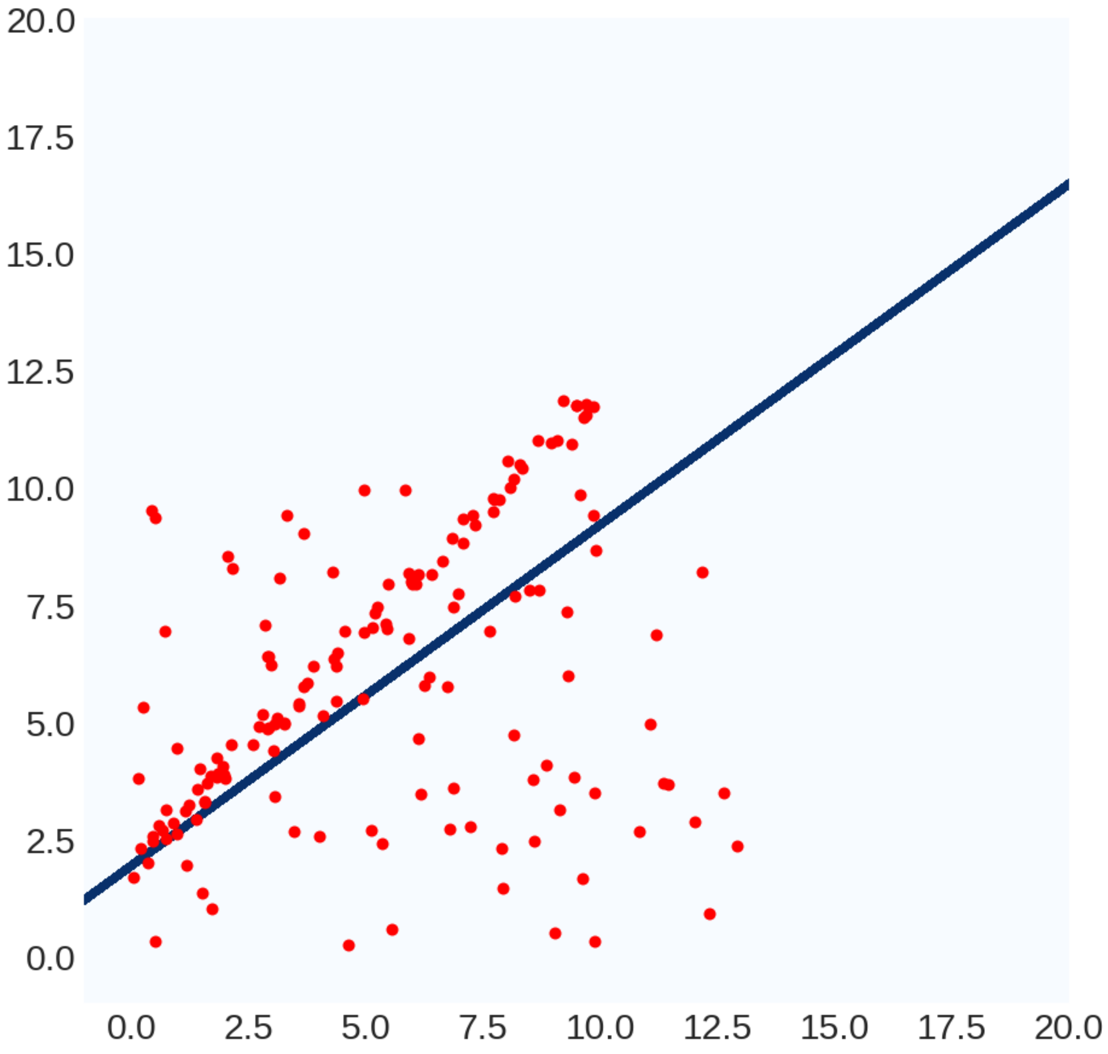
- Draw $\boldsymbol{s}$ points uniformly at random
- Fit line to these $\boldsymbol{s}$ points
- Find inliers to this line among the remaining points.
  - Inlier = points whose distance from the line is less than $\boldsymbol{t}$
- If there are $\boldsymbol{d}$ or more inliers, accept the line and refit using all inliers

Answer the next two questions based on the above algorithm.

(g) (2 points) What is the overarching optimization objective in the above algorithm?

- ☐ Reduce mean-square error of the line against all points.
- ☐ Find the threshold $\boldsymbol{t}$, given the set of all points.
- ■ Find the line with the highest possible number of inliers.
- ☐ Find the optimal number of points to be sampled ($\boldsymbol{s}$).

Ruining Least Squares



Far from best fit!

Slide credit: David Fouhey

RANSAC for estimating homography

RANSAC loop:

1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute *inliers* where $SSD(p_i', \boldsymbol{H} p_i) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

Slide credit: Steve Seitz