- CV in 106A is short but meant to be practical
- We don't go into much depth but cover some general concepts:
  - Convolutions
    - Filters
    — Homography
  - Pinhole cameras
  - Perspective Projection
  - Two-View Geometry (on homework)

1 Convolutions

A **convolution** is the treatment of one matrix M (the original image) by another usually smaller one K (the kernel). The result of a convolution is a filtered image matrix $K * M$. The convolution is performed by sliding the kernel over the original image and taking the matrix dot product of the kernel and the part of original matrix that it covers. Let's look at an example:
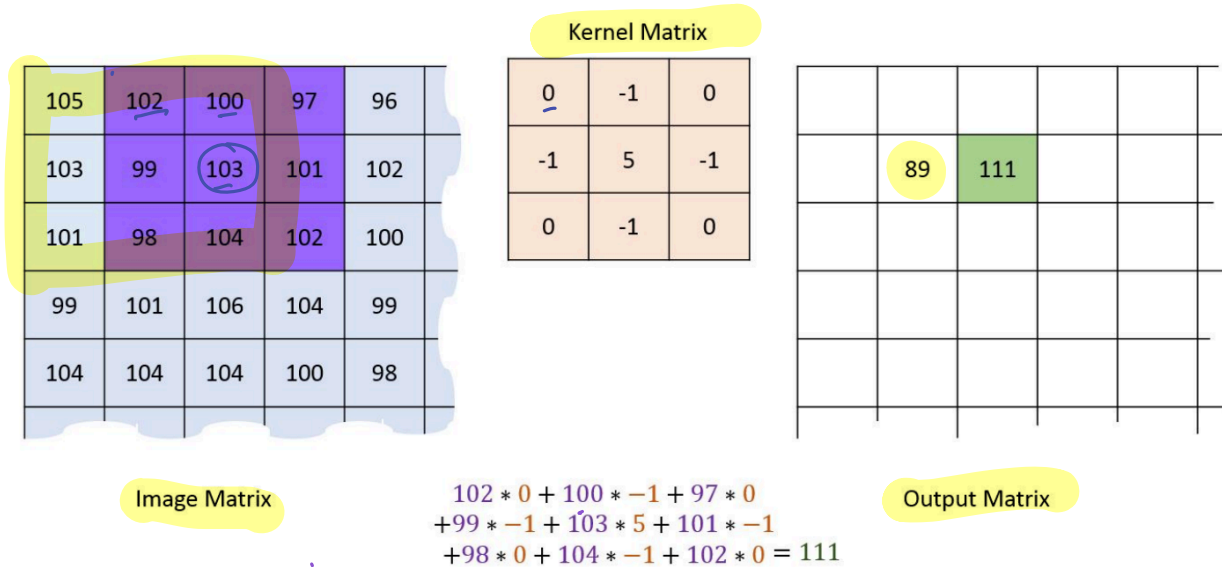
Kernel Matrix

105	102	100	97	96
103	99	103	101	102
101	98	104	102	100
99	101	106	104	99
104	104	104	100	98

0	-1	0
-1	5	-1
0	-1	0

	89	111		

Image Matrix

$$102 * 0 + 100 * -1 + 97 * 0$$
$$+99 * -1 + 103 * 5 + 101 * -1$$
$$+98 * 0 + 104 * -1 + 102 * 0 = 111$$

Output Matrix

Figure 1: Example of a convolution.

Problem 1. *If the original matrix has dimensions $m \times n$ and the kernel has dimensions $p \times q$, what will be the size of the matrix resulting from the convolution?* → no padding

n cols

m×n

q wide

p×q

$$(m - p + 1) \times (n - q + 1)$$

$$0\times1 + 0\times2 + 0\times3$$
$$\cdots + 1\times5 + 0\times6\cdots$$
$$= 5$$

Problem 2. *Convolution kernels are useful for applying effects to images and extracting features for machine learning. Match the resulting image to the kernel applied as well as the name of its effect.*

Result			
Kernel	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ — Softens edge — Average vals around pixel	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ — Enhance bright pixels — Diminish surrounding values
Name of effect	Box Blur	Identity	Sharpen

might have to normalize after convolution

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 255 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Kernels:

$$\left\{ \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right\}$$

Names of effects: $\{$Identity, Box Blur, Sharpen$\}$

Problem 3. *What is the form of a Gaussian blur kernel?*

value of center prioritized

- Largest value in center
- As we go to edges, follow Gaussian curve
- Choose σ as desired

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

$\rightarrow$ Normalized Gaussian function

1.1 Sobel Kernels

The Sobel kernels are kernels (let's call them K_x and K_y) that approximate the horizontal and vertical derivatives of an image respectively upon convolution. That is, given an image M, the horizontal derivatives are found by $G_x = K_x * M$, and the vertical by $G_y = K_y * M$.

Problem 4. *Which of these following kernels is K_x? Which one is K_y?*

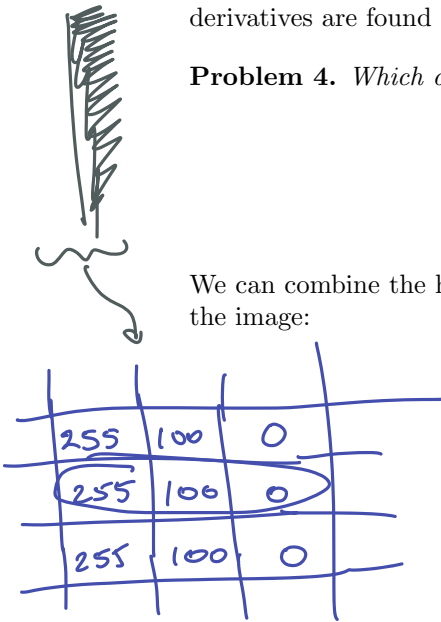$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

K_x $\qquad$ K_y

We can combine the horizontal and vertical derivatives into G to approximate the overall gradient of the image:

$$\boxed{G = \sqrt{G_x^2 + G_y^2}}$$

$\rightarrow$ Both together

$$\begin{bmatrix} 255 & 100 & 0 \\ 255 & 100 & 0 \\ 255 & 100 & 0 \end{bmatrix}$$

$$-(255 + 255\times2 + 255)^2$$
$$+ 0$$
$$+ 0 = -(\text{very large } \#)$$

$\rightarrow$ Derivative change in intensity values

Rapid shift in color

Problem 5. *How is G useful in edge detection?*

— Really large #s → rapid shift in color

Edge

2   Pinhole Camera Model

When we take a 2D picture, we are essentially transforming points in the real 3D world to points on a 2D image. Let's see if we can work out what this transformation is, which we call the *intrinsic camera matrix*. To do so, we model our camera as a standard pinhole model camera, in which light rays from the real world are projected onto an image plane and there is no lens involved.
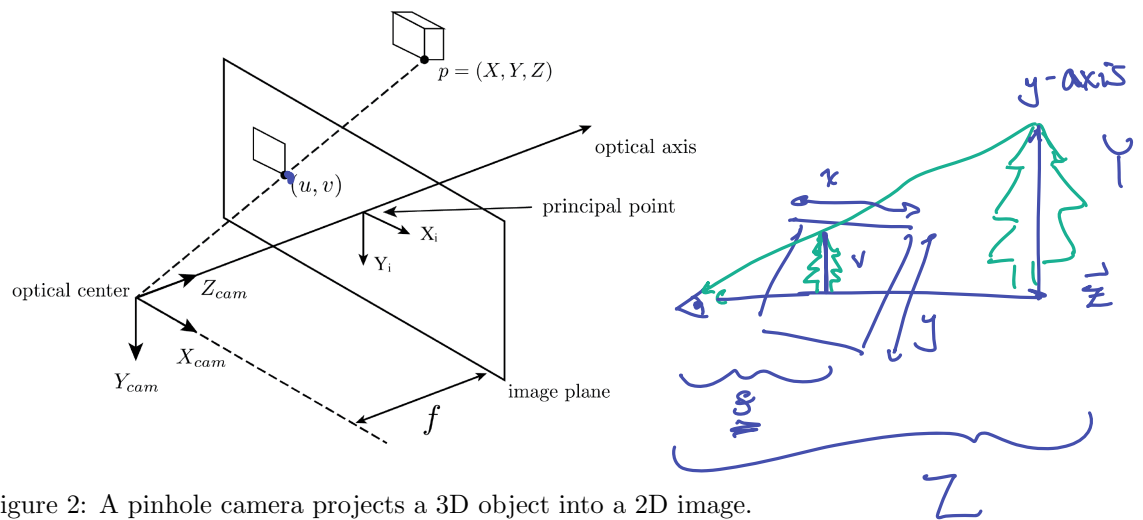
Figure 2: A pinhole camera projects a 3D object into a 2D image.

Let's look at Fig. 2 to derive our matrix. We have two frames of reference here. One is the camera frame with axes $X_{cam}, Y_{cam}, Z_{cam}$ from which 3D locations can be expressed. For example, point p has position (X, Y, Z).

We have another 2D frame of reference in the image plane with axes X_i, Y_i. The point p projected onto the image frame has coordinates (u, v) in this image frame.

Problem 6. *Write expressions for u and v as functions of the 3D position (X, Y, Z) and the focal length f.*

$$\frac{u}{f} = \frac{X}{Z}$$

$$\frac{v}{f} = \frac{Y}{Z}$$

$$u = f \cdot \frac{X}{Z}$$

$$v = f \cdot \frac{Y}{Z}$$

3

2.1 Scaling

In the real world, there may be scaling effects present that may differ for the horizontal and vertical directions. This is as if f is scaled by a factor m_x when doing projections in the x direction, and scaled by m_y when doing projections in the y direction. Thus, we replace the f in the expressions for u and v by $f_x := m_x \cdot f$ and $f_y := m_y \cdot f$ respectively.

Problem 7. *Update the expressions for u and v using f_x and f_y instead of f.*

$$u = f_x \cdot \frac{X}{Z} \qquad\qquad v = f_y \cdot \frac{Y}{Z}$$

2.2 Translation of origin

In computer vision, the origin of a 2D image may not actually lie on the principal point. Thus, let's allow the image frame with axes X_i, Y_i be free to move around, such that the principal point is now at an arbitrary coordinate (x_o, y_o).

Problem 8. *Update the expressions for u and v to take into account this arbitrary origin shift.*

$$u = f_x \cdot \frac{X}{Z} + x_o$$

$$v = f_y \cdot \frac{Y}{Z} + y_o$$

(0,0) $\to +x$

$+y$

$x\ (u,v)$
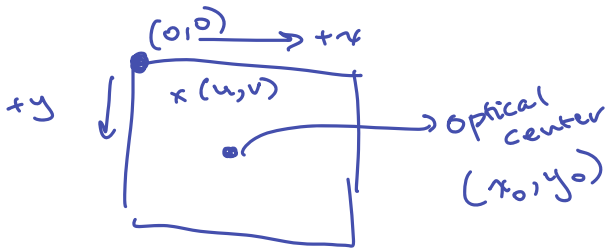
$\to$ optical center (x_o, y_o)

2.3 Homogeneous coordinates

We define a clever way to express 2D coordinates in the image plane with three dimensions. We do this by appending a w to the end of the vector and dividing u and v by this w. That is, $\begin{bmatrix} u \\ v \end{bmatrix}$ turns

into $\begin{bmatrix} u' \\ v' \\ w \end{bmatrix}$ where $\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{w} \begin{bmatrix} u' \\ v' \end{bmatrix}$.

Not $SE(2)$ don't f_x, f_y to form need $SO(2)$

multiply Z on both sides

Problem 9. *In homogeneous coordinates where w is set to be Z, find the intrinsic camera matrix that maps a real 3D point to a point in the image plane.*

$$Z \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & x_o \\ 0 & f_y & y_o \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsic camera matrix}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$u' = u \cdot Z = f_x \cdot X + x_o \cdot Z$$
$$v' = v \cdot Z = f_y \cdot Y + y_o \cdot Z$$

– only info · focal lengths · origin

↓

– Given by camera
– Published by Sawyer

4

tinyurl.com/106a - disc - 2