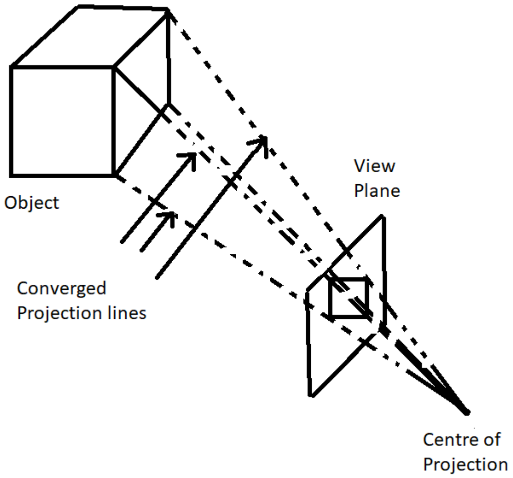
Inspiring Excellence

CSE463
Computer Vision: Fundamentals and Applications
Lecture 2
Image Formation and Filters

Geometry of Image Formation

The geometry of image formation studies the process by which 3D objects in the world are captured and represented on a 2D image plane. This field is foundational in understanding how cameras perceive depth, scale, and spatial relationships in a scene.

Perspective Projection

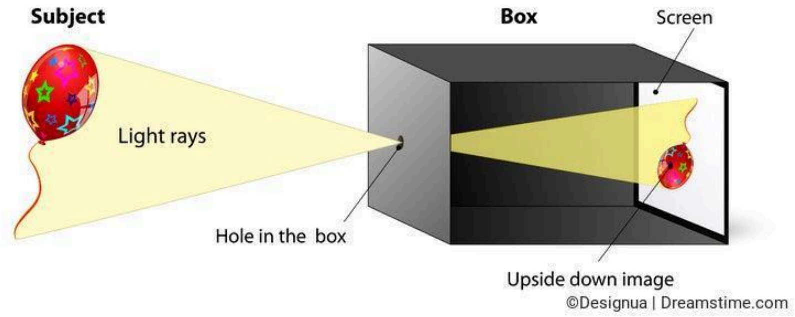


In perspective projection, objects appear smaller as they move further away from the camera, and lines that are parallel in the 3D world converge in the 2D image, typically towards a “vanishing point.” This principle explains why nearby objects appear large while distant objects appear small and are crucial for a realistic representation of depth in images.

Camera Image Formation

Camera image formation refers to the process of capturing a 3D scene from the world and projecting it onto a 2D image plane, which is a critical aspect of understanding how images are formed in computer vision and photogrammetry. This process involves several physical principles and geometrical concepts that ensure a 3D scene is represented correctly on a 2D plane.

Pin-Hole Camera Model



The simplest model of image formation is the pinhole camera model, which provides a conceptual framework for understanding how light from a scene is captured through a small aperture (the "pinhole") and projected onto an image plane (the camera sensor).

Pinhole Camera Model Components:

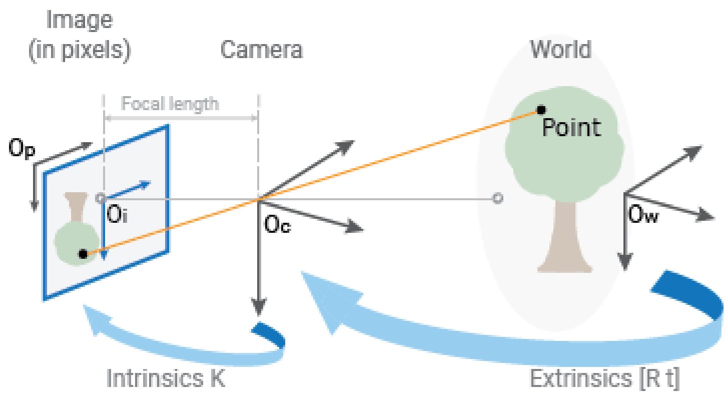
- Scene (3D world): A real-world scene consisting of objects in three-dimensional space.
- Camera: The device that captures the scene, consisting of a lens and an image plane.
- Pinhole: A small aperture through which light passes.
- Image plane: A 2D surface (typically a digital sensor or film) where the scene is projected.

How It Works:

1. Light rays from the 3D objects in the scene pass through the pinhole and hit the image plane.
2. Each light ray corresponds to a specific point in the scene and is projected onto a point on the image plane.
3. The resulting image on the image plane is inverted, meaning that objects higher in the scene appear lower on the image plane, and objects farther away appear closer.
4. The size of the image depends on the distance between the scene, the pinhole, and the image plane.

The pinhole camera model is a simple approximation, but it provides the basis for more sophisticated camera models that include lens effects like distortion and focus.

Camera Calibration Parameters



For accurate image formation and interpretation, a camera's internal and external properties must be understood. These properties are captured in the intrinsic and extrinsic parameters of the camera.

Intrinsic Parameters (Camera Intrinsics):

These are the internal properties of the camera that affect how it captures the scene.

- Focal Length: The distance between the camera's lens and the image plane. It determines the magnification and the field of view (FOV).
- Principal Point: The point on the image plane where the optical axis intersects (usually near the center of the image).
- Pixel Aspect Ratio: The ratio of the width to the height of a pixel in the camera sensor. This parameter is used to account for non-square pixels.
- Skew: A measure of non-orthogonality of the image axes (often assumed to be zero in most cameras).

These parameters are typically represented in a camera matrix KKK, which is used to transform 3D coordinates into 2D image coordinates.

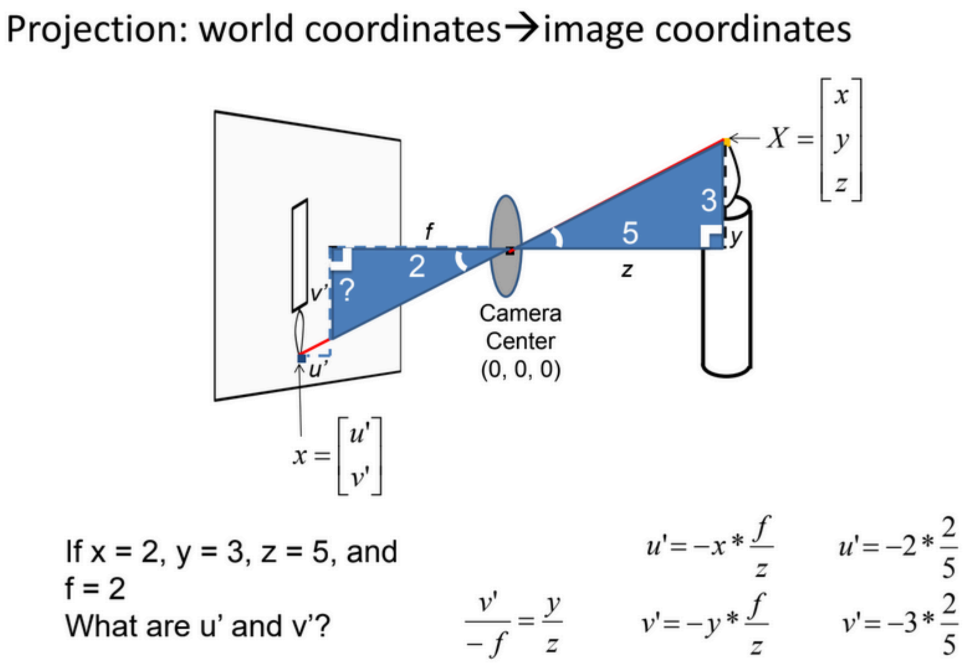
Extrinsic Parameters (Camera Extrinsic):

These parameters describe the position and orientation of the camera in the world.

- Rotation Matrix (R): A 3x3 matrix that describes the camera’s orientation in 3D space.
- Translation Vector (T): A 3x1 vector that describes the camera’s position in 3D space relative to the world coordinate system.

Extrinsic parameters define how the camera is positioned relative to the world and are critical for reconstructing 3D scenes from images.

Projection Models



Projection: world coordinates→image coordinates

f
2
?
v'
u'
5
3
y
z
Camera Center (0, 0, 0)
$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
$x = \begin{bmatrix} u' \\ v' \end{bmatrix}$

If x = 2, y = 3, z = 5, and f = 2
What are u' and v'?

$$u' = -x * \frac{f}{z} \qquad u' = -2 * \frac{2}{5}$$

$$\frac{v'}{-f} = \frac{y}{z} \qquad v' = -y * \frac{f}{z} \qquad v' = -3 * \frac{2}{5}$$

The projection from 3D space onto 2D space can be modeled using different projection techniques, such as **perspective projection** and **orthographic projection**.

Perspective Projection:

- Most common in real-world cameras and is responsible for the phenomenon where objects appear smaller as they get farther away from the camera (i.e., the vanishing point).
- In perspective projection, light rays converge towards a single point (the camera's focal point or pinhole).
- The transformation from 3D world coordinates (X, Y, Z) to 2D image coordinates (x,y) is a nonlinear operation and involves both intrinsic and extrinsic parameters.

The mathematical formulation for perspective projection is as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \cdot [R|T] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Where:

- [R∣T]is the extrinsic matrix (rotation and translation),
- Kis the intrinsic camera matrix,
- (X, Y, Z)are the 3D coordinates of a point in the world,
- (x,y)are the corresponding 2D image coordinates.

Orthographic Projection:

- Assumes parallel projection where objects appear the same size regardless of their distance from the camera.
- It is often used for technical drawings or engineering applications but not for real-world photography, as it doesn’t capture depth perception.

In orthographic projection, the transformation from 3D world coordinates to 2D coordinates is linear, and the depth dimension is ignored.

Image Formation:

- **Image formation models** describe the physics of how images are formed on the camera sensor.
- **Light and Aperture**: Light enters the camera through the aperture, which controls the amount of light hitting the sensor. The aperture and lens focus the incoming light, creating an image on the sensor.
- **Focal Length and Depth of Field**: The focal length determines the magnification of the image, while the depth of field affects the range of distances at which objects appear sharply in focus. Adjusting these parameters changes the scene's perspective and focus.

Image Filtering (2D Convolution)

Image filtering is a technique applied to images to enhance or preprocess them for analysis by modifying pixel values in systematic ways. Filters can remove noise, enhance edges, or sharpen an image, depending on the filter type and parameters. Filters are also known as kernels and can have a shape of 1x1, 3x3, 5x5, 7x7, and so on.

Types of Filters:

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Linear Filters:

- **Gaussian Filter**: A smoothing filter used to reduce noise by averaging pixel values in a local region, creating a blurring effect. It's widely used as a preprocessing step in computer vision tasks.
- **Box filtering:** this is an average-of-surrounding-pixel kind of image filtering that causes blurring. A 3x3 box filter is given as follows:

$$g[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- **Sobel Filter**: An edge-detection filter that calculates the gradient of image intensity, highlighting regions with rapid intensity change, which correspond to edges.

Non-Linear Filters:

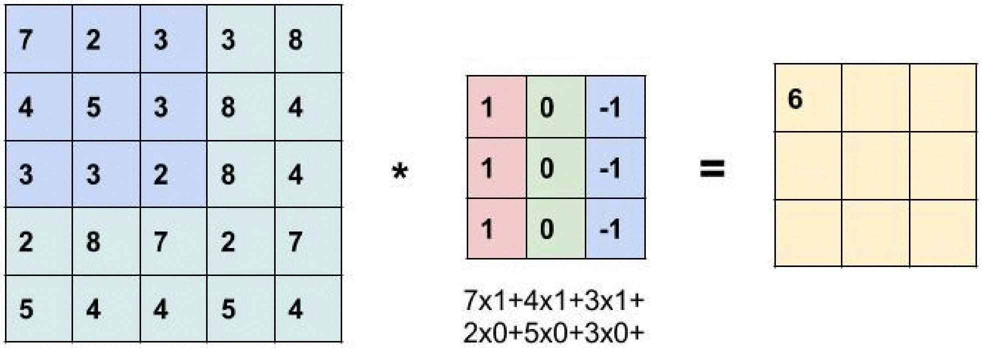
- **Median Filter**: A noise-reduction filter that replaces each pixel with the median value of neighboring pixels. It is effective at removing "salt-and-pepper" noise without blurring edges.
- **Bilateral Filter**: This filter smooths the image while preserving edges, by combining both spatial and intensity information, making it useful in preserving details in high-frequency areas.

Sliding of a Filter/Kernal Example

Eg 1 for 3x3 Kernal with explanation: https://www.songho.ca/dsp/convolution/convolution2d_example.html

Eg 2: https://www.youtube.com/watch?v=yb2tPt0QVPY

Eg 3:



$$\begin{bmatrix} 7 & 2 & 3 & 3 & 8 \\ 4 & 5 & 3 & 8 & 4 \\ 3 & 3 & 2 & 8 & 4 \\ 2 & 8 & 7 & 2 & 7 \\ 5 & 4 & 4 & 5 & 4 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 6 & & \\ & & \\ & & \end{bmatrix}$$

7x1+4x1+3x1+
2x0+5x0+3x0+
3x-1+3x-1+2x-1
= 6

Applications of Filters:

Filters are widely used in image processing tasks such as:

1. noise reduction (Gaussian, Median filters)
2. edge detection (Sobel filter)
3. image sharpening or smoothing

Filtering helps prepare images for higher-level tasks by enhancing specific features or reducing irrelevant data.

Exercises

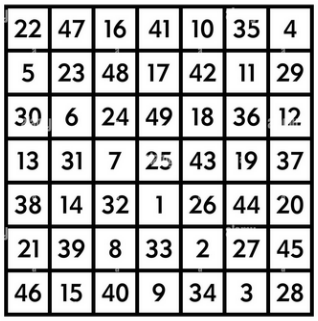
1. ✓ What is the pinhole camera model, and how does it explain the projection of a 3D world onto a 2D image plane?
2. ✓ Describe the difference between intrinsic and extrinsic camera parameters. Why are both necessary for accurate image projection?
3. ✓ What is perspective projection, and how does it affect the appearance of objects as they move farther from the camera?
4. ✓ Compare and contrast orthographic projection with perspective projection. In what scenarios might each be preferred?
5. ✓ In terms of image formation, explain how light enters through the aperture and is focused onto the image sensor. What role does the lens play in this process?
6. ✓ What is the purpose of using a Gaussian filter in image processing? How does it work to reduce noise in an image?
7. Explain the difference between linear and non-linear filters, and provide examples of each. How do these filters affect images?
8. ✓ Write down the matrix representation for a 5x5 box filter. And apply it over the image given below-

8

a.

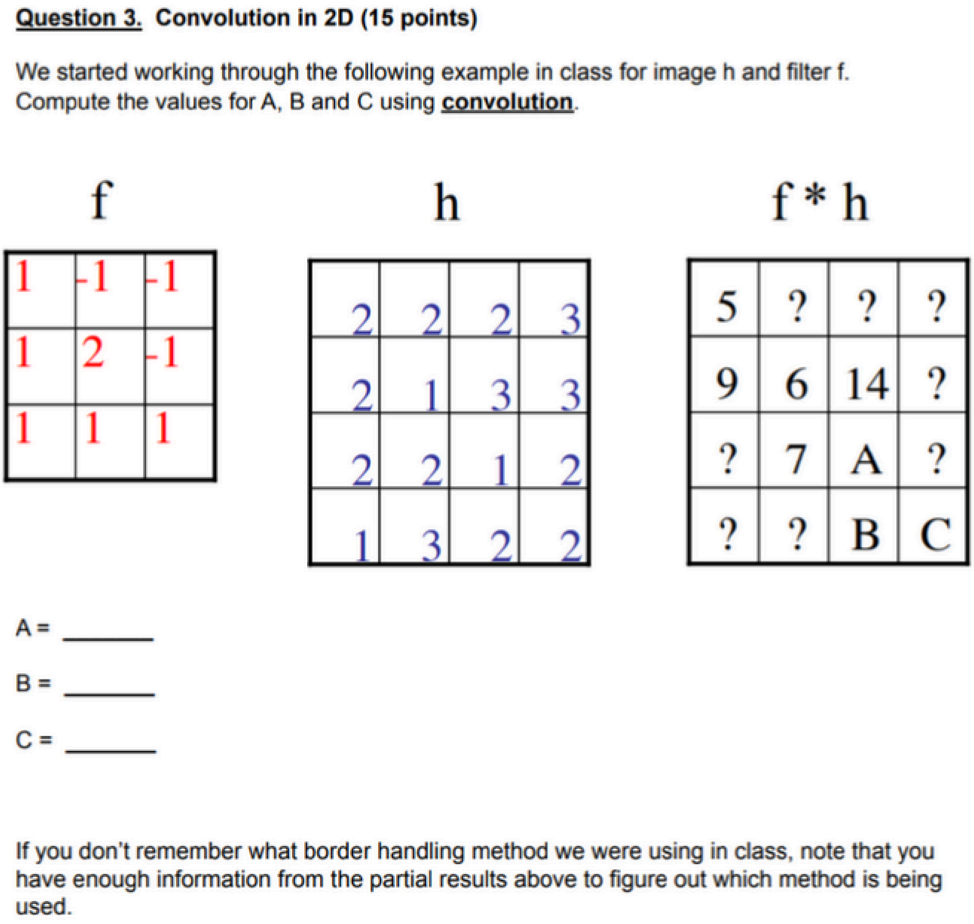
1	2	3
4	5	6
7	8	9

b.



22	47	16	41	10	35	4
5	23	48	17	42	11	29
30	6	24	49	18	36	12
13	31	7	25	43	19	37
38	14	32	1	26	44	20
21	39	8	33	2	27	45
46	15	40	9	34	3	28

9.



Question 3. Convolution in 2D (15 points)

We started working through the following example in class for image h and filter f. Compute the values for A, B and C using **convolution**.

f

1	-1	-1
1	2	-1
1	1	1

h

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

f * h

5	?	?	?
9	6	14	?
?	7	A	?
?	?	B	C

A = _______

B = _______

C = _______

If you don't remember what border handling method we were using in class, note that you have enough information from the partial results above to figure out which method is being used.

10. The image on the left shows a noisy image. What filter can be used to revert it to its original form?

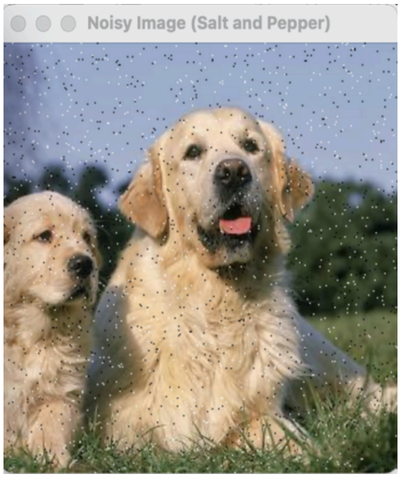


Image Filtering Mathematical Examples-

- **Output Size Calculation**

The output size H_out, W_out for a convolution is given by:

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} - K_h}{S} \right\rfloor + 1 \qquad W_{\text{out}} = \left\lfloor \frac{W_{\text{in}} - K_w}{S} \right\rfloor + 1$$

Input Size (H_in,W_in): The height or width of the original input image (before filtering).

Filter Size (K_h, K_w): The height and width of the filter (kernel) being applied to the image.

Stride (S): The number of pixels the filter moves (or "slides") horizontally or vertically in each step.

Assuming an **Image size** of 10×10 applying a **Filter size:** 3×3 with a **stride** of 2-

$$H_{\text{out}} = \left\lfloor \frac{10-3}{2} \right\rfloor + 1 = \lfloor 3.5 \rfloor + 1 \ = 4.5$$

$$W_{\text{out}} = \left\lfloor \frac{10-3}{2} \right\rfloor + 1 \quad = 4.5$$

Since we can't have fractional pixels, we need to **floor** the value: Output Size = 4 x 4.

Gaussian Filter-

- The Gaussian kernel is defined mathematically as: $G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$

➔ σ is the standard deviation controlling the extent of smoothing.
➔ For example, with σ=1, a 3x3 kernel might look like:

$$G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Imagine you have a image of size 5x5 and a filter of size 3x3 with a stride 1,

Sample Image Matrix,

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

Gaussian Kernel,

$$G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Output Image Size-

Simplified Formula for squared image = ((N - F + 2P)/S) + 1 where N=5 (input), F=3 (filter), P=0 (padding), S=1 (stride)

So, Output size = ((5-3+0)/1) + 1 = 3x3

Steps for Each Position:

We slide the kernel across the image, calculate the weighted sum for each 3×3 patch, and normalize the result by dividing by 16.

(a) For Position (0, 0):

- Extract the sub-image: $\begin{bmatrix} 1 & 2 & 3 \\ 6 & 7 & 8 \\ 11 & 12 & 13 \end{bmatrix}$

- Perform element-wise multiplication:

$$\frac{1}{16}\begin{bmatrix} 1\cdot 1 & 2\cdot 2 & 3\cdot 1 \\ 6\cdot 2 & 7\cdot 4 & 8\cdot 2 \\ 11\cdot 1 & 12\cdot 2 & 13\cdot 1 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 1 & 4 & 3 \\ 12 & 28 & 16 \\ 11 & 24 & 13 \end{bmatrix}$$

- Calculate the result (sum and normalize) and put it in the (0, 0) position:

$$(0, 0) = \frac{112}{16} = 7$$

(b) For position (0, 1): Slide the filter to cover the sub-image (stride = 1),

- Extract the sub-image: $\begin{bmatrix} 2 & 3 & 4 \\ 7 & 8 & 9 \\ 12 & 13 & 14 \end{bmatrix}$

- Repeat the weighted sum and normalization steps:

= $\frac{1}{16}$ * (2 * 1 + 3 * 2 + 4 * 1 + 7 * 2 + 8 * 4 + 9 * 2 + 12 * 1 + 13 * 2 + 14 * 1) = 8

Continue this for all positions.

Put it in the filtered image matrix-

7	8	**?**
?	**?**	**?**
?	**?**	**?**