Computer Vision

Faisal Bukhari, PhD

Punjab University College of Information Technology (PUCIT)

Textbook

Multiple View Geometry in Computer Vision, Hartley, R., and Zisserman

Richard Szeliski, **Computer Vision: Algorithms and Applications**, 2nd edition, 2022

Reference books

Readings for these lecture notes:

- Hartley, R., and Zisserman, A. **Multiple View Geometry in Computer Vision**, Cambridge University Press, 2004, Chapters 1-3.

- Forsyth, D., and Ponce, J. **Computer Vision: A Modern Approach**, Prentice-Hall, 2003, Chapter 2.

- **Linear Algebra and its application by David C Lay**

These notes contain material from the above resources.

References

These notes are based from the following resources:

- Dr. Matthew N. Dailey's course: AT70.20: Machine Vision for Robotics and HCI

- Dr. Sohaib Ahmad Khan CS436 / CS5310 Computer Vision Fundamentals at LUMS

- http://rimstar.org/science_electronics_projects/pinhole_camera.htm

Grading breakup

I. Midterm = 35 points
II. Final term = 40 points
III. Quizzes = 6 points (A total of 6 quizzes)
IV. Group project = 15 points
    a. Pitch your project idea = 2 points
    b. Research paper presentation relevant to your project = 3 points
    c. Project prototype and its presentation = 5 points
    d. Research paper in IEEE conference template = 5 points
V. OpenCV based on Python presentation = 2.5 points
VI. Matlab presentation = 2.5 points

Some top tier conferences of computer vision

I. Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition **(CVPR).**
II. Proceedings of the European Conference on Computer Vision **(ECCV).**
III. Proceedings of the Asian Conference on Computer Vision **(ACCV).**
IV. Proceedings of the International Conference on Robotics and Automation **(ICRA).**
V. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems **(IROS).**

Some well known Journals

I. International Journal of Computer Vision **(IJCV).**
II. IEEE Transactions on Pattern Analysis and Machine Intelligence **(PAMI).**
III. Image and Vision Computing.
IV. Pattern Recognition.
V. Computer Vision and Image Understanding.
VI. IEEE Transactions on Robotics.
VII. Journal of Mathematical Imaging and Vision

Recall: Vector Space

A **vector space** is a **nonempty set** V of objects, called *vectors*, on which are defined **two operations**, called ***addition*** and ***multiplication by scalars*** (real numbers), subject to the **ten axioms (or rules)** listed in the next slide.

Recall: Vector Space

The axioms must hold for all vectors **u**, **v**, and **w** in V and for all **scalars** c and d. The sum of **u** and **v**, denoted by **u** + **v**, is in V .

1. The sum of **u** and **v**, denoted by **u** + **v**, is in V .
2. **u** + **v** = **v** + **u**.
3. (**u** + **v**) + **w** = **u** + (**v** + **w**).
4. There is a **zero** vector **0** in V such that **u** + **0** = **u**.
5. For each **u** in V , there is a vector - **u** in V such that **u** + (-**u**) = **0**.
6. The scalar multiple of **u** by c, denoted by c**u**, is in V .
7. c(**u** + **v**) = c**u** + c**v**.
8. (c + d) **u** = c**u** + d**u**.
9. c(d**u**) = (cd)**u**
10. 1**u** = **u**.

Subspaces of $\mathbb{R}^n$

Definition: A **subspace** of $\mathbb{R}^n$ is any set *H* in $\mathbb{R}^n$ that has **three** properties:

a) The **zero vector** is in *H*.

b) For each ***u*** and ***v*** in *H*, the sum **u + v** is in *H*.

c) For each **u** in *H* and each scalar *c*, the **vector** ***c*u** is in *H*.

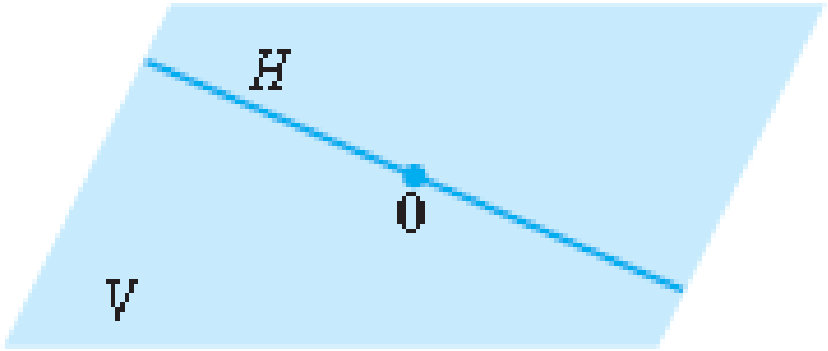
In words, a **subspace** is ***closed*** under **addition** and **scalar multiplication**

Subspace vs. vector space

- Every **subspace** is a **vector space**.

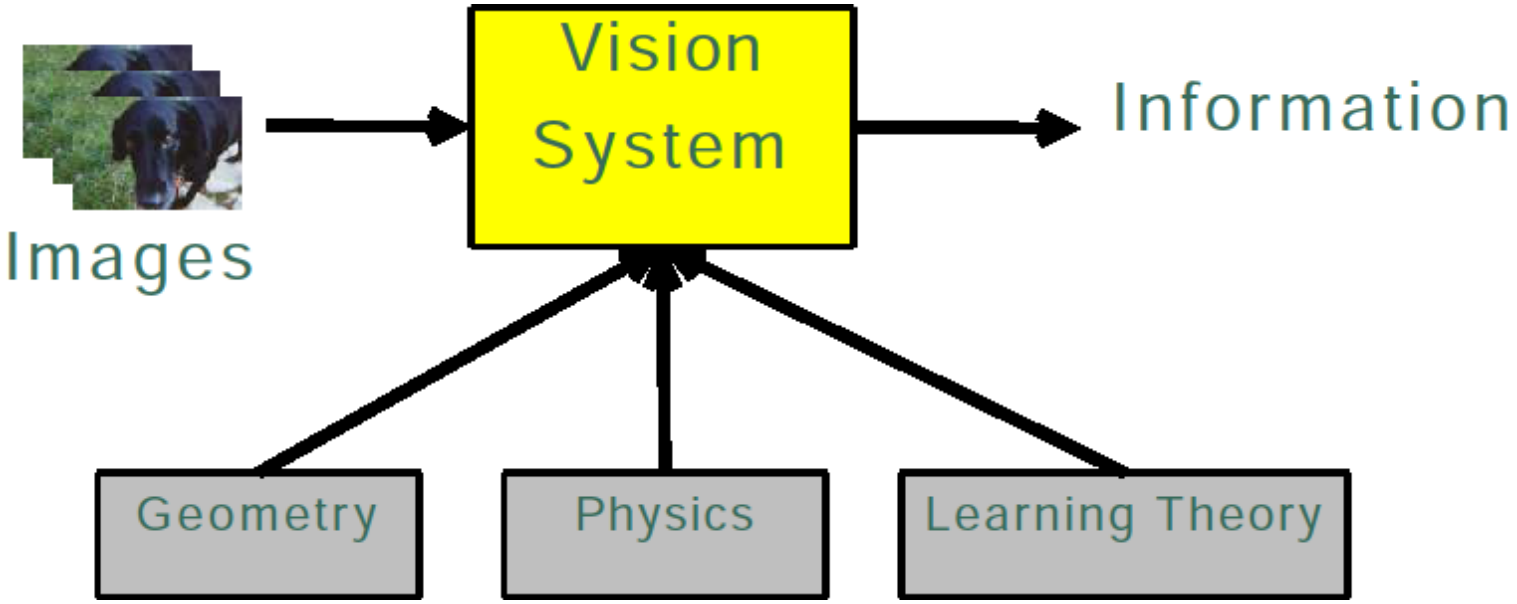
- **Conversely**, every **vector space** is a **subspace** (of itself and possibly of other larger spaces).

- The **term *subspace*** is used when **at least two vector spaces** are in **mind**, with **one inside the other**, and the phrase ***subspace of* V** identifies **V** as the **larger space**. See figure in the next slide.



A subspace of V

Introduction: Vision systems



The kind of information we want is application specific:

3D models

Object categories

Object poses

Camera poses

Parts of the system [1]

The “vision system” includes:

- **Image acquisition hardware**
  - **Analog camera** plus **digital frame grabber**

    or
  - **Digital camera** with a **fast serial interface** (Firewire, USB, etc.)
- **Image processing** support software.
- **Computer vision** algorithms.

Parts of the system [2]

- Different **applications** have varying requirements for the type of information that needs to be captured and processed.

- To achieve this, we will **acquire images**, which requires specific **hardware**.

- Depending on the setup, we may use:

- **An analog camera** with a **digital frame grabber**.

- **Digital cameras** equipped with serial interfaces such as **FireWire, USB 3.0, or other high-speed connections**.

Parts of the system [3]

- Modern **IP cameras** use **Ethernet**, with some models supporting **Gigabit Ethernet** (1 GigE or 1 Gbps) or **10 Gigabit Ethernet** (10 GigE or 10 Gbps) for high-speed image transmission.

- We will focus more on our **image processing support software** and **vision algorithms** to build these systems.

Types of Ethernet

Ethernet is a **wired networking technology** that allows devices to communicate over a **local area network (LAN)** using cables.

Type	Speed	Cable Type
Fast Ethernet	100 Mbps	Cat5
Gigabit Ethernet (1 GigE)	1 Gbps	Cat5e, Cat6
10 Gigabit Ethernet (10 GigE)	10 Gbps	Cat6a, Cat7
40/100 Gigabit Ethernet	40-100 Gbps	Fiber Optic, Cat8

Our focus in this semester

- This semester we focus on algorithms for **3D reconstruction.**

- To understand modern **3D reconstruction techniques,** we **need to understand how cameras transduce** the **world into images**.

- This requires a deep understanding of **projective geometry and camera models**, which serve as the mathematical foundation of computer vision.

- Once we grasp these concepts, we can begin to **invert the camera's transformation process** and develop methods to **reconstruct 3D scenes** from **2D images.**

2D Projective Geometry

- We begin with **2D projective geometry** because it's simple, then we'll **generalize to 3D**.

- In particular, we consider what happens when we perform **projective transformations** of the **plane**.

- **Projective transformations** model the distortions introduced by **projective cameras** (more on cameras later).

- In projective cameras, **funny things happen**. Although **straight lines stay straight, parallel lines are no longer parallel.** Projective geometry gives us mathematics for these kinds of transformations.

Introduction to 2D Projective Geometry

- We start with **2D projective geometry** because it is simpler, then extend to **3D**.

- Our focus is on **projective transformations** of the **plane**.

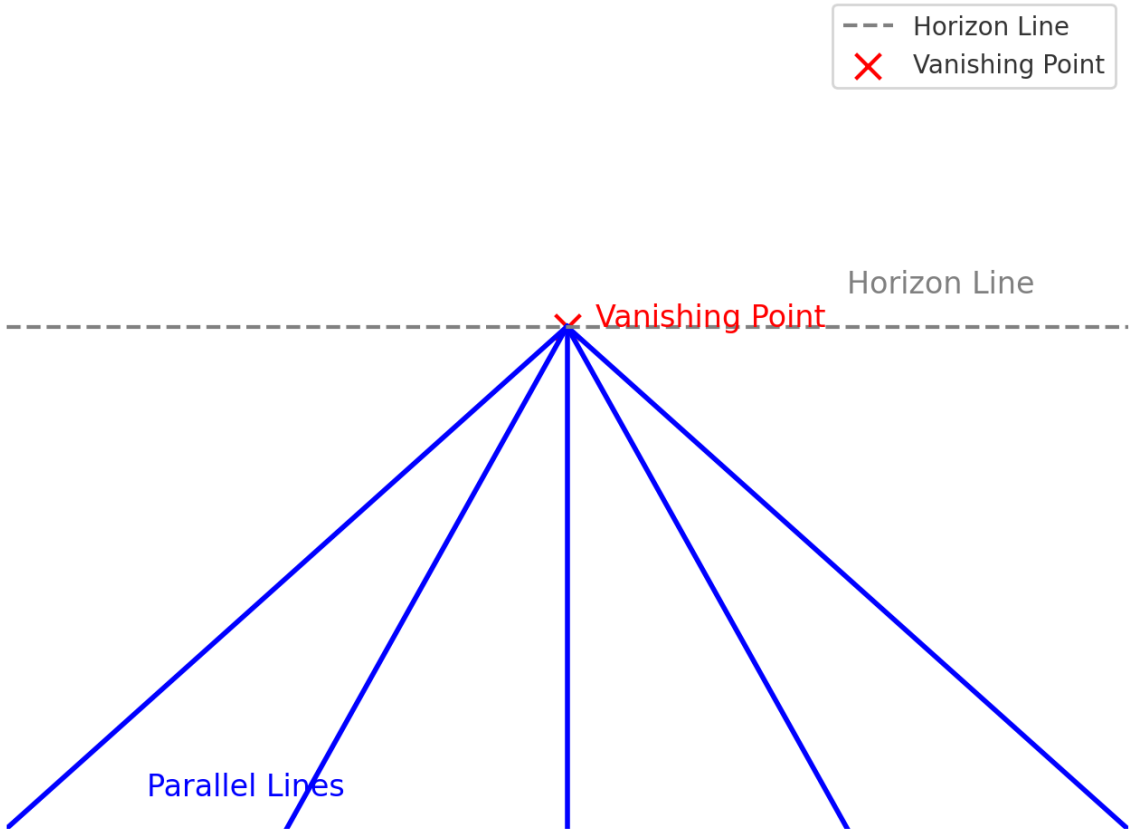
Projective Transformations

- These transformations model the **distortions** introduced by **projective cameras**.

- They preserve **straight lines**, but **parallel lines may no longer remain parallel**

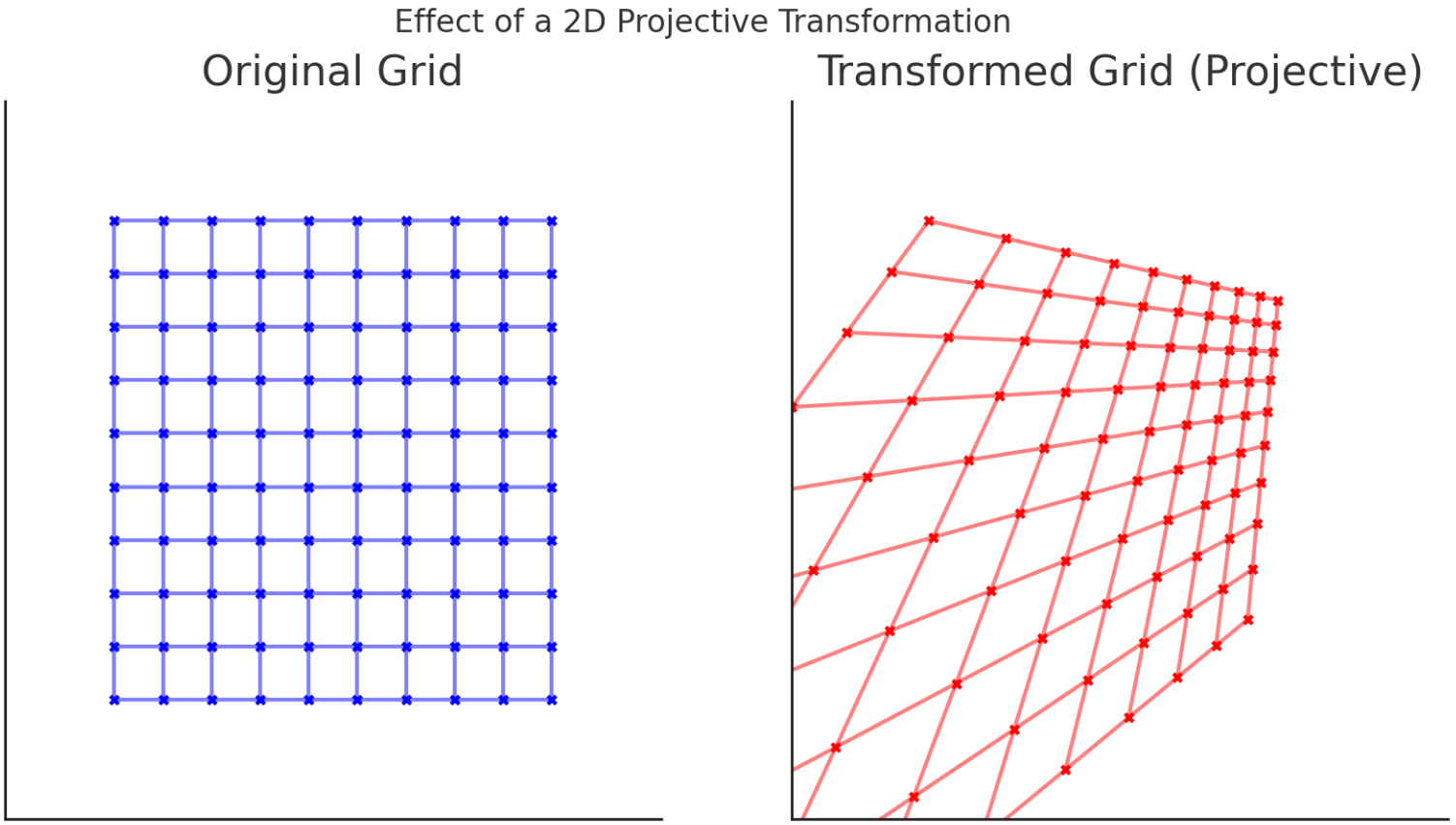
Effects of Projective Cameras

- **Perspective distortion** occurs, leading to changes in how objects appear
- **Straight lines remain straight**, but **parallel lines converge** at vanishing points.
- **Projective geometry** provides the mathematical framework to handle these transformations.



Parallel lines (in blue) appear to converge at a **vanishing point (red)** due to perspective distortion

The **horizon line** (dashed gray) represents where all vanishing points lie for lines parallel to the ground.



Left (Original Grid, Blue): Represents a standard Euclidean grid.

Right (Transformed Grid, Red): The same grid after applying a **projective transformation (homography).**

Key Takeaways:

- **Straight lines** remain straight, but the **overall shape distorts**.
- Perspective warping occurs, resembling how a camera would capture an image with depth.
- This is a fundamental principle in computer vision and image rectification.

2D Projective Geometry

- The thing we have to take care of 2D projective geometry is **projective transformations**.

- In projective transformations, we take **some arbitrary dimensional environment,** and we capture images of it on (in this case, as we talk about) **2D plane**.

- **Projective Transformations**. A transformation that maps **lines to lines** (but does not necessarily **preserve parallelism**) is a projective transformation.

2D projective geometry

The 2D projective plane: points in $\mathbb{R}^2$

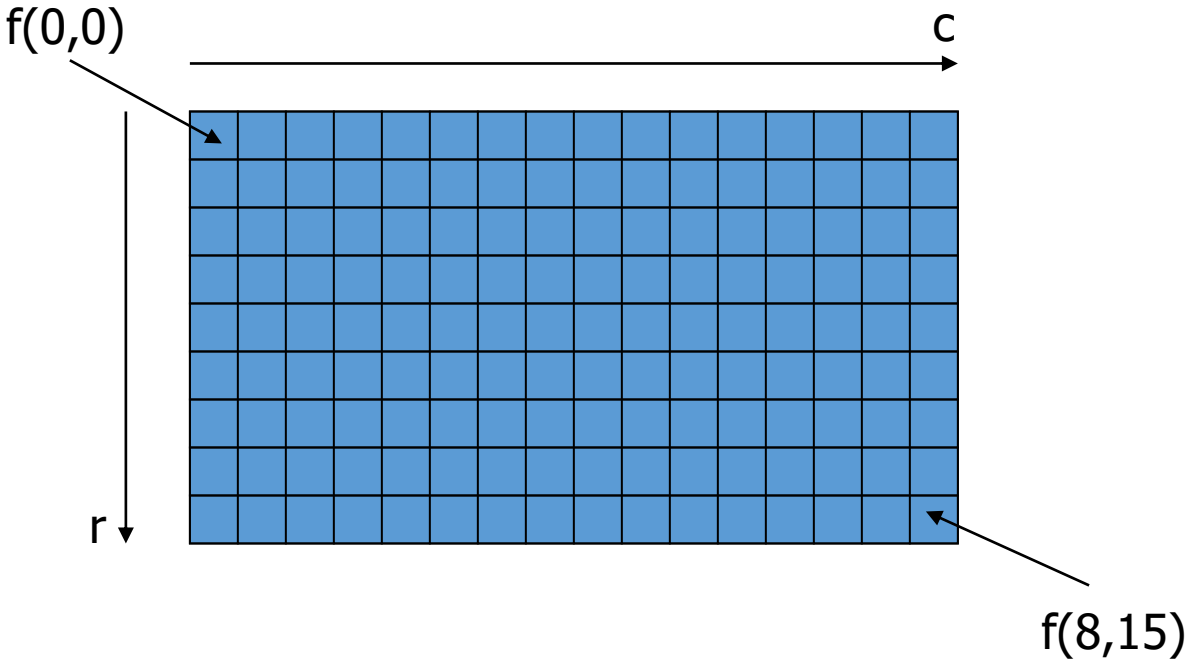
- A point in the plane can be represented as a pair (x, y) in $\mathbb{R}^2$.

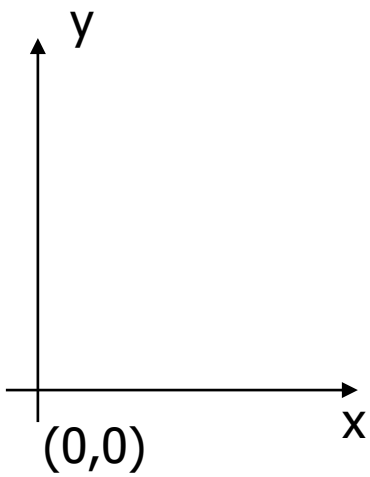
- We consider $\mathbb{R}^2$ as a vector space, and we write the point (x, y) as a vector. This step makes it possible to write transformations of points as matrices.

- Generally, we write transformations on the left and points on the right, so we need to write points as column vectors, i.e., $(x, y)^T$.

- We will typically write column vectors using bold upright symbols, e.g., $\mathbf{x} = (x, y)^T$ or $\vec{x} = (x, y)^T$.

Representing a Digital Image



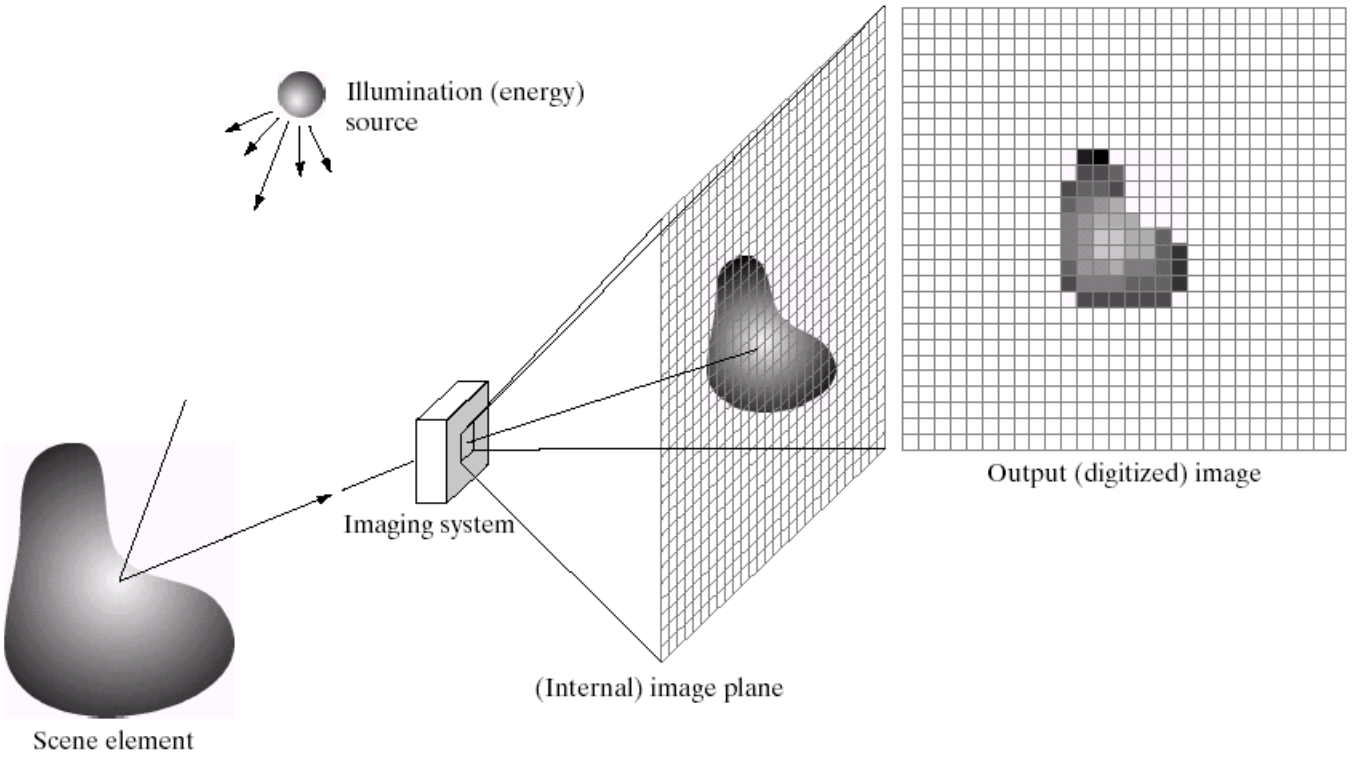


$r \leftrightarrow x$

$c \leftrightarrow y$

90° cc rotation

Image Acquisition

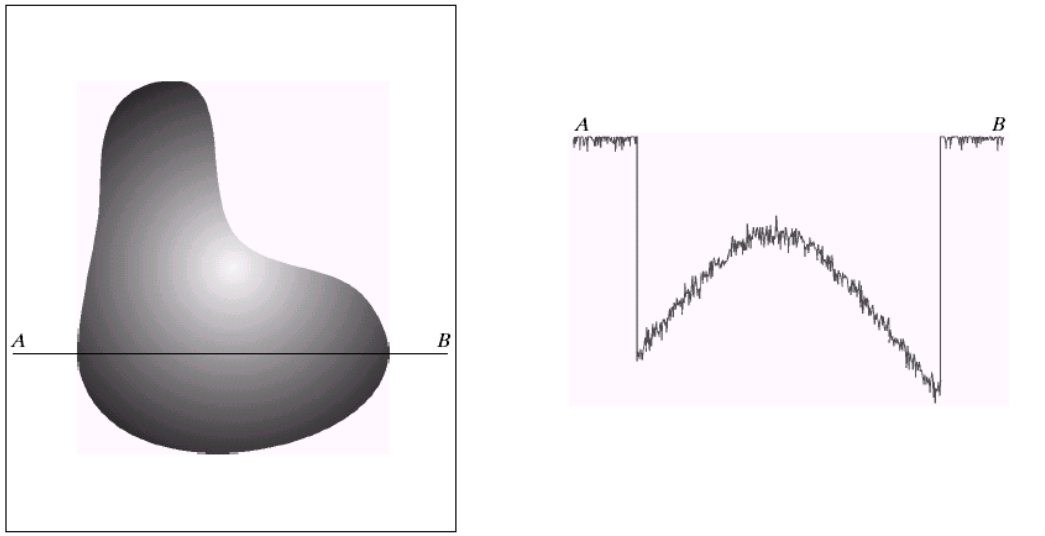


a
b c d e

FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Figure Credit: Gonzales & Woods: Digital Image Processing

Generating a Digital Image



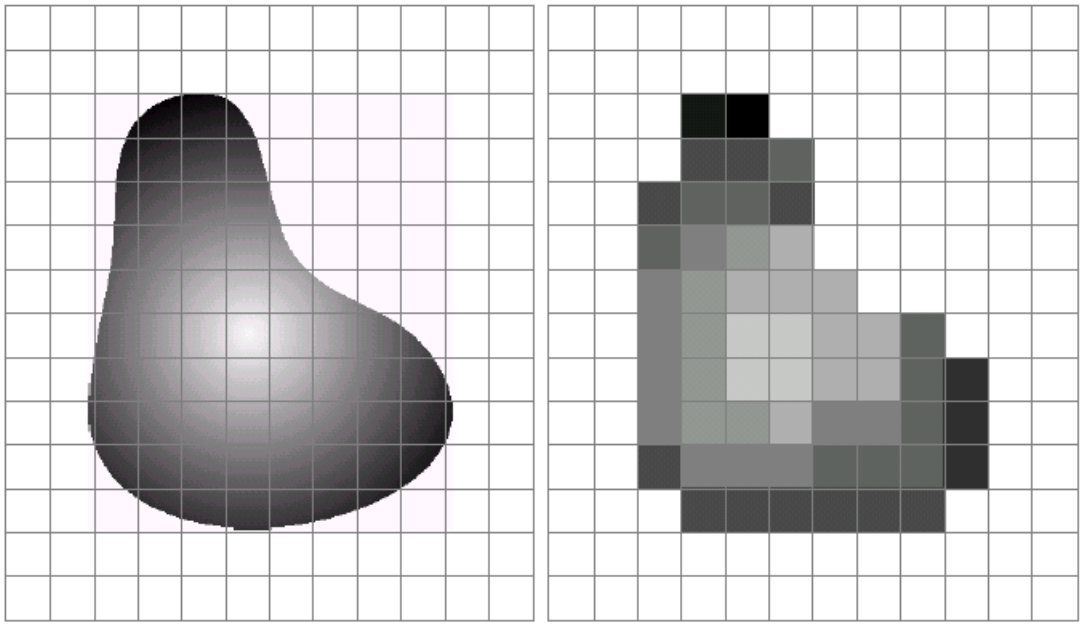
a b
c d

FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

- Digitization of coordinate values is called **sampling**
- Digitization of amplitude values is called **quantization**

Figure Credit: Gonzales & Woods: Digital Image Processing

Generating a Digital Image



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

- How many samples to use?
- How many quantization levels to use?

Figure Credit: Gonzales & Woods: Digital Image Processing

Image Size and Resolution

256 x 192

128 x 96

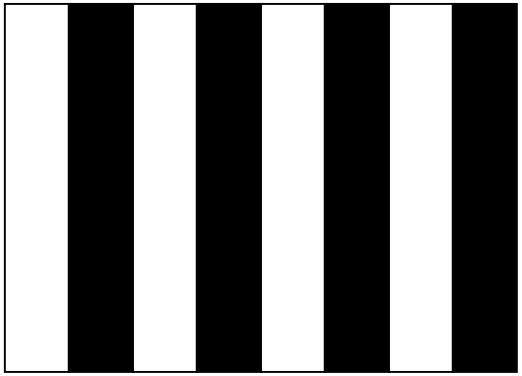
64 x 48

32 x 24

These images were produced simply by picking every *n*-th sample horizontally and vertically and replicating the value *n* x *n* times

Resolution

- **Spatial Resolution** is the smallest discernable detail in an image
- What is the minimum width of lines W such that they are discernable in the image?
- **Gray-level resolution** is the smallest discernable change in gray-level



Different Number of Gray Levels

256

32

16

8

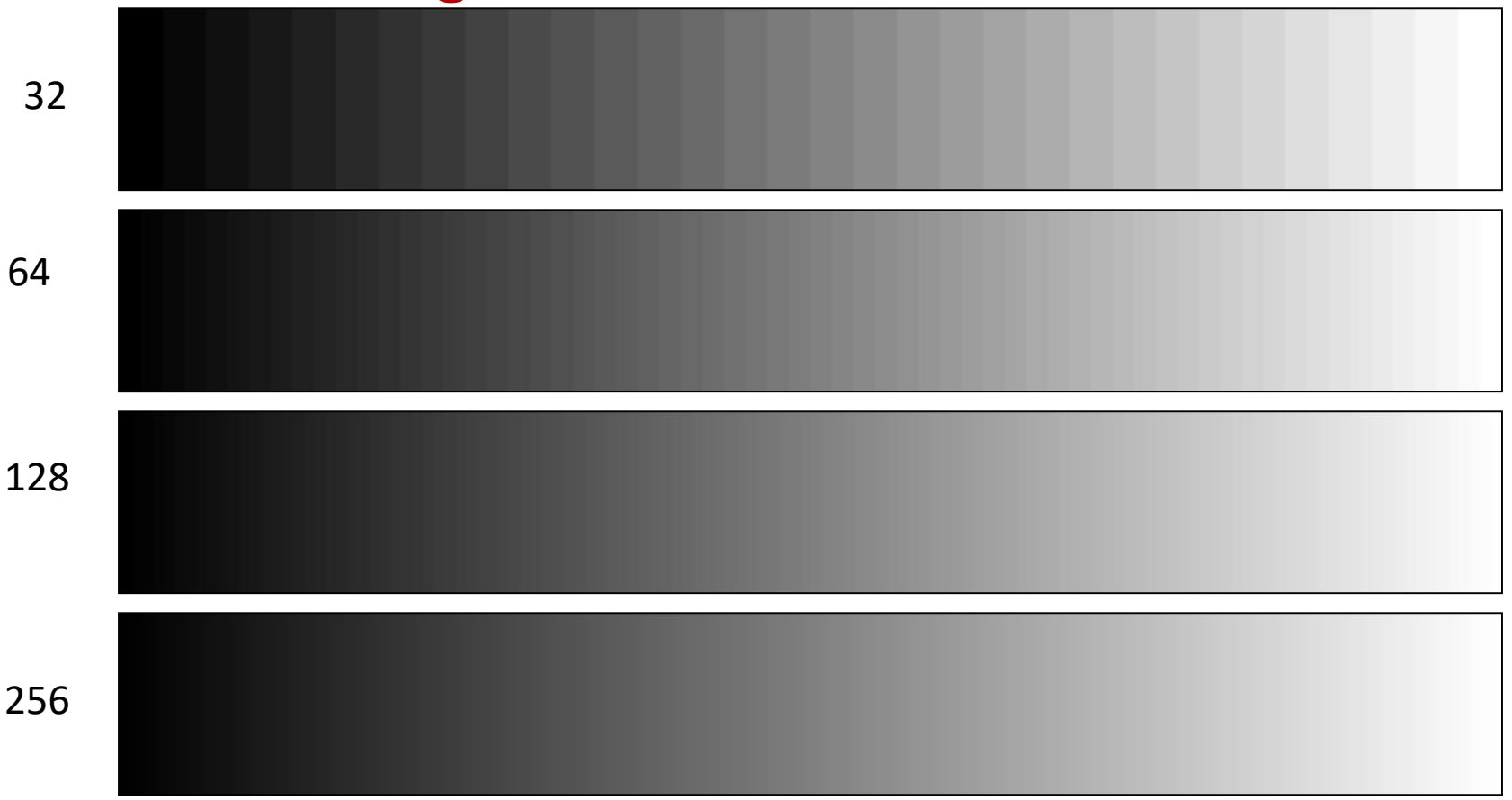
4

2

Contouring

How many gray-levels are required?

Contouring…



Digital Images are usually quantized to 256 gray-levels

Examples of Common Image Resolutions

Resolution	Common Usage
640 × 480	Old VGA displays
1280 × 720	HD (720p) videos
1920 × 1080	Full HD (1080p) videos
3840 × 2160	4K Ultra HD
7680 × 4320	8K resolution

Introduction to Digital Image Processing

Digital images are formed by converting analog signals into discrete values.

This conversion involves two key processes: **Sampling** and **Quantization.**

Understanding these concepts is essential for image processing and computer vision applications.

What is Sampling?

- **Sampling** is the process of **converting a continuous image** into a **discrete image** by selecting **values at fixed intervals**.

- Determines the **spatial resolution** of an image (i.e., the number of pixels per unit area).

- **Higher sampling** results in more details and sharper images.

- **Measured in pixels per inch (PPI)** or total pixel count (e.g., 1920 ×1080 pixels).

What is Quantization?

- **Quantization** is the process of mapping **pixel intensity values** to a finite number of discrete levels.
- Determines the number of **colors or shades** that can be represented in an image.
- **Higher quantization** levels result in **smoother color transitions**.
- **Measured in bits per pixel (bpp),** e.g., 8-bit (256 levels), 16-bit, 24-bit (True Color).

Effects of Low Sampling and Quantization

- **Low Sampling:** Results in **pixelated** and **blocky** images due to insufficient resolution.

- **Low Quantization:** Causes color banding and loss of smooth transitions in grayscale or color images.

- **High-quality images** require both **high sampling** and **high quantization**.

Applications in Computer Vision

- **Medical Imaging:** High-resolution and high-bit-depth images are crucial for accurate diagnosis.
- **Facial Recognition:** Image quality impacts feature extraction and recognition accuracy.
- **Autonomous Vehicles:** Object detection models require clear and high-quality image inputs.
- **Satellite Imaging:** High sampling and quantization improve remote sensing and analysis.

Key Differences Between Sampling and Quantization

- **Sampling** controls the **image resolution** (spatial details), while **quantization** controls **color depth**.
- **Low sampling leads to pixelation**, whereas low quantization causes color banding.
- **Higher sampling and quantization** improve image quality but require more **storage and processing power**.

Introduction to Image Types

- Digital images are classified based on how they store pixel information.
- The main types are **Binary, Grayscale**, and **Color images**.
- Each type has different applications in image processing and computer vision.

What Are Binary Images?

- **Binary images** use only **two colors**: **black** and **white** (0 and 1).
- Each pixel is either **ON (white)** or **OFF (black)**.
- Used in **document scanning**, **edge detection**, and **barcodes**.
- Requires very **low storage space**.

What Are Grayscale Images?

- **Grayscale images** use shades of gray ranging from **black (0)** to **white (255).**
- **Each pixel** has a **single intensity value**.
- Commonly used in **medical imaging**, **pattern recognition**, and **image processing**.
- Captures **more details** than **binary images**.

What Are Color Images?

- **Color images** use **multiple channels** (e.g., RGB: Red, Green, Blue).
- **Each pixel** has multiple values to define its color.
- **Typically 24-bit (8-bit per channel)** or higher (e.g., HDR images).
- Used in photography, graphics design, and displays.

Applications of Each Image Type

Binary Images: OCR (Optical Character Recognition), QR codes, edge detection.

Grayscale Images: Medical imaging, security cameras, image enhancement.

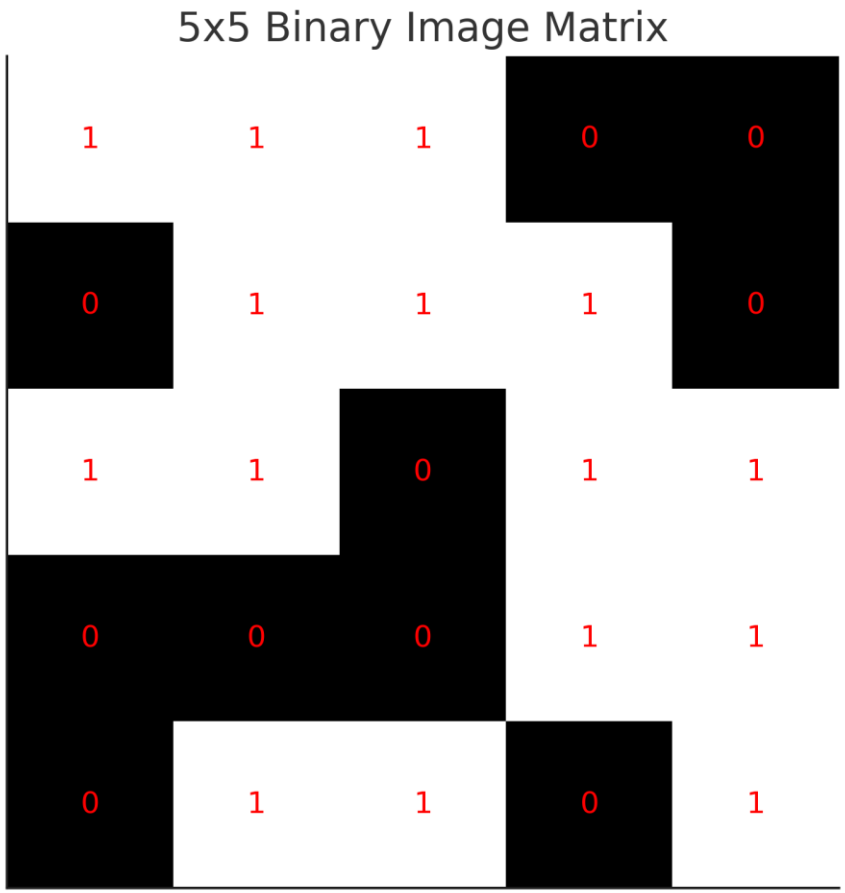
Color Images: Photography, video processing, augmented reality.

Choice of image type depends on the application and required detail.

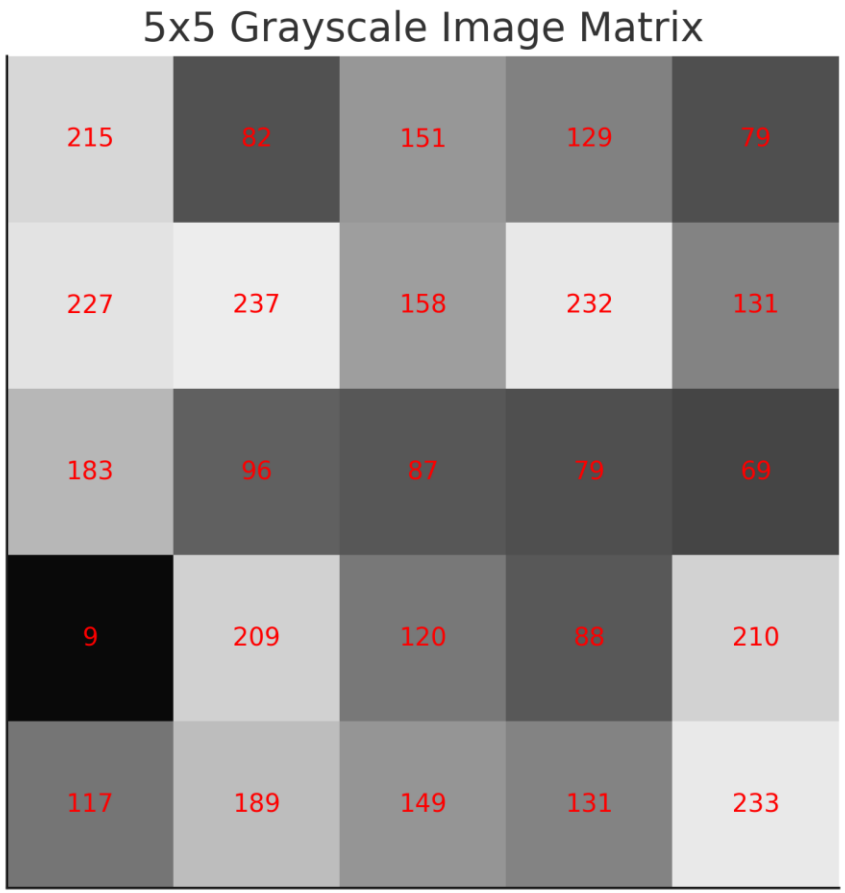
Summary and Conclusion

- **Binary images** are the simplest, storing only black and white.
- **Grayscale images** provide better detail using intensity levels.
- **Color images** use multiple channels to represent full color.
- Understanding these differences is key in image processing and computer vision.

Binary Image



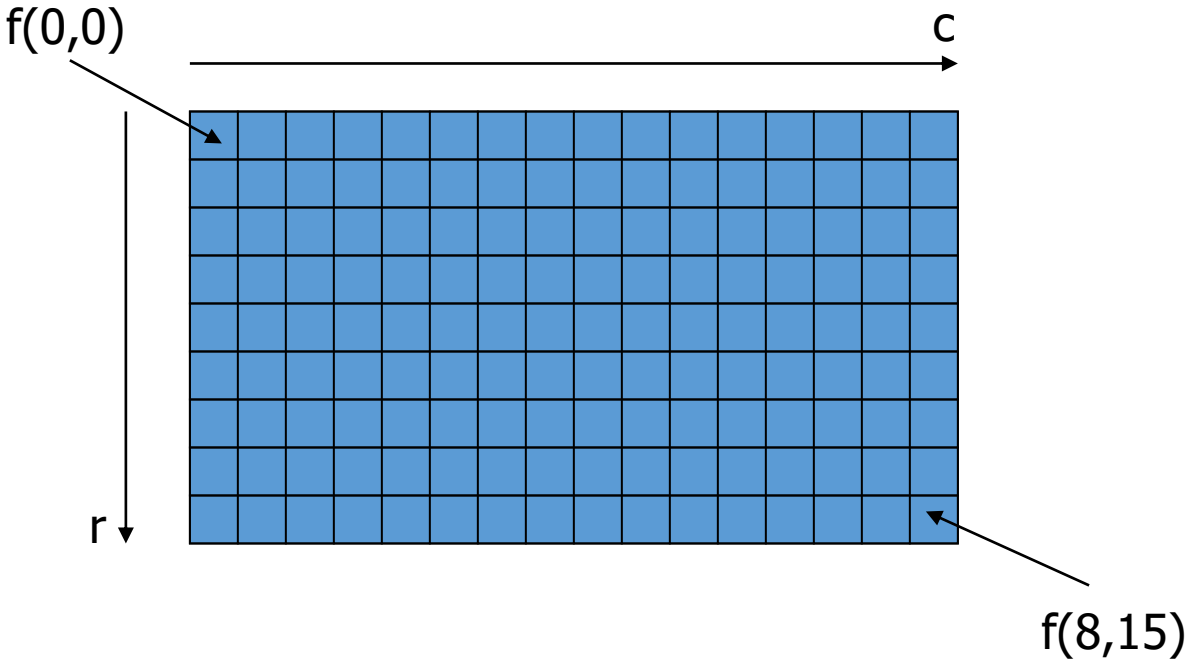
Grayscale Image

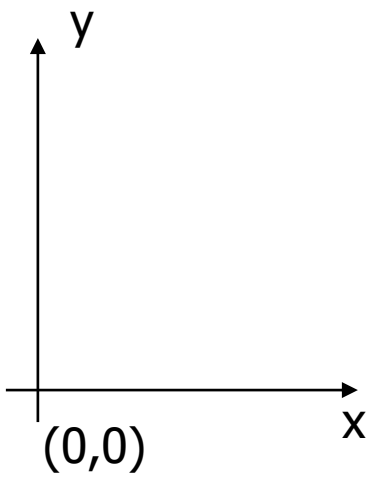


Color Image


5x5 Color Image Matrix (RGB)
(187,127,7)
(251,100,9)
(191,220,83)
(101,25,190)
(82,216,6)
(141,117,129)
(122,156,252)
(214,184,13)
(91,80,166)
(108,90,42)
(8,119,135)
(62,90,179)
(247,50,235)
(59,137,23)
(122,2,68)
(223,82,166)
(19,173,177)
(52,101,70)
(224,43,205)
(142,124,178)
(238,148,235)
(164,3,217)
(150,110,199)
(136,184,234)
(29,113,75)

Representing a Digital Image





$r \leftrightarrow x$

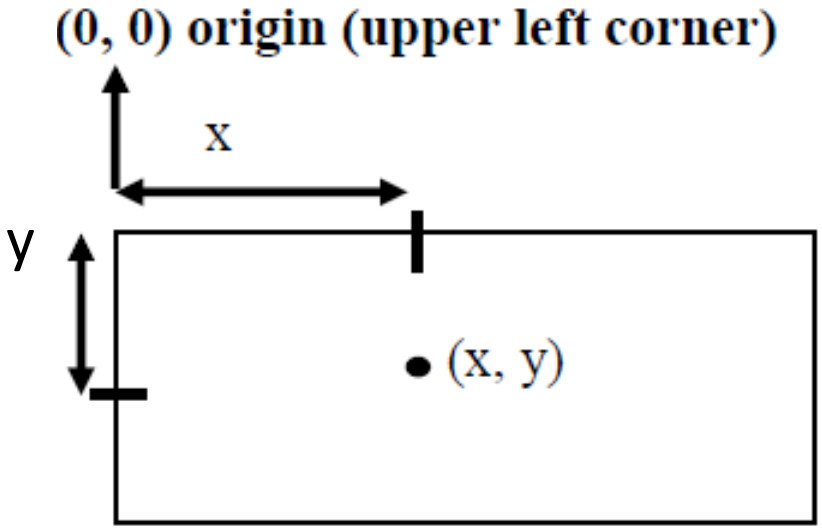
$c \leftrightarrow y$

90° cc rotation

Point in 2D

$\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ or $\begin{bmatrix} x \\ y \end{bmatrix}$

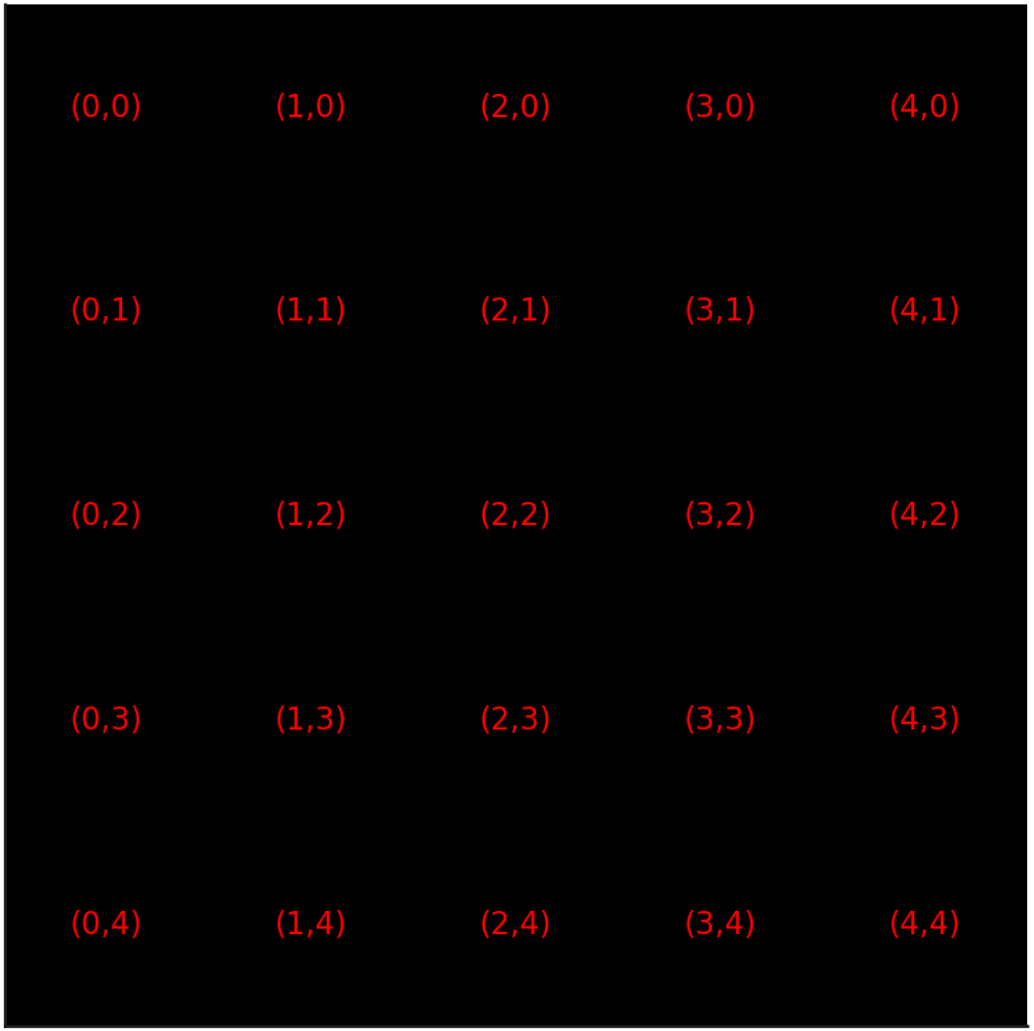
This point $\vec{x}$ is the same as the two Cartesian coordinates x and y. We're doing something funny when we're dealing with images.



Understanding 2D Points in Images

- A point in 2D projective geometry is represented as (x, y), similar to Cartesian coordinates.
- However, in digital images, the **origin (0,0)** is at **the top-left corner**.
- The **x-axis** increases **horizontally to the right**, and the **y-axis increases vertically downward**.
- This is different from the traditional Cartesian system, where the y-axis increases upward.
- Understanding this coordinate system is crucial in computer vision and image processing.

Pixel Coordinate System (Digital Image Processing - Gonzalez & Woods)



Point in 2D

- We can select the **origin of our image coordinate system** whatever we like, but usually, we talk about the **upper left corner** and having the **coordinates (0, 0).**

- Usually, in image processing, when we talk about **a point (x, y),** then we are talking about **displacement to the right by x** and **displacement down by y**. We will write these vectors as **row vectors or column vectors**.

- Generally, we assume that we are modeling **some points** in **the image plane** in most cases.

Key Differences Between Cartesian (Euclidean) and Projective Coordinates

Feature	Cartesian Coordinates	Projective Coordinates
Definition	Standard (x, y) or (x, y, z) coordinates	Homogeneous (x, y, w) or (x, y, z, w)
Extra Dimension	No extra dimension	Includes an extra scale factor w
Usage	Geometry, physics, machine learning	Computer vision, 3D graphics, camera modeling
Transformations	Linear transformations (rotation, translation, scaling)	Affine and projective transformations (perspective, homographies)
Representation	Fixed point locations	Defined up to a scale factor
Perspective Effect	No perspective effects	Can represent perspective transformations