Université Constantine 2
جامعة قسنطينة 2

Artificial Vision

– Course 3 –

Chapter 3 : 3D Reconstruction of Scenes (1/1)

Dr. Benaliouche Houda

Faculté des nouvelles technologies

Houda.benaliouche@univ-constantine2.dz

Université Constantine 2
جامعة قسنطينة 2

Artificial Vision

– Course 3 –

Chapter 3 : 3D Reconstruction of Scenes (1/1)

Dr. Benaliouche Houda

Faculté des nouvelles technologies

Houda.benaliouche@univ-constantine2.dz

Etudiants concernés

Faculté/Institut	Département	Niveau	Spécialité
Nouvelles technologies	/	Master 2	Sciences de Données et Intelligence Artificielle (SDIA)

Summary

Prerequisites

- Mathematical Notions
- Algorithmic Notions

Course Objective

- A look into how machines see the world in 3 D

OUTLINE

- ✓ Definition of 3D Reconstruction of a scene
- ✓ Depth from Stereo and 3D Stereoscopic Vision
- ✓ Camera Calibration
- ✓ Geometry of Epipolar Lines
- ✓ Fundamental Matrix
- ✓ Projection Techniques
- ✓ Determining 3D Coordinates
- ✓ Interpreting Images from Various Angles
- ✓ Structure from Motion
- ✓ Three Dimensions Object Recognition
  - Method 1: Alignment Method
  - Method 2: Invariant Technique
  - Method 3: method of decomposition into parts
- ✓ Object Positioning from a Single Image
- ✓ Challenges

DEFINITION

What is 3D reconstruction?

Definition:

The process of capturing the shape and appearance of real-world objects or scenes to create a 3D digital model

Examples:

- Augmented Reality apps overlaying 3D furniture in rooms.
- Autonomous driving using LiDAR for street mapping.
- CT scan reconstruction into 3D heart models.

DEFINITION

Imagine taking pictures or videos of an object, a room, or a place from different angles, and then using a computer to "rebuild" that scene in three dimensions.

DEFINITION

Here's a simple way to think about it:

Taking Pictures: Think of it like taking multiple photos of a sculpture from all sides. Each picture shows a little piece of the sculpture.

Connecting the Pieces: A computer analyzes these images to figure out the shape, size, and position of the object in 3D space. It does this by looking at how features like edges and corners align in the different images.

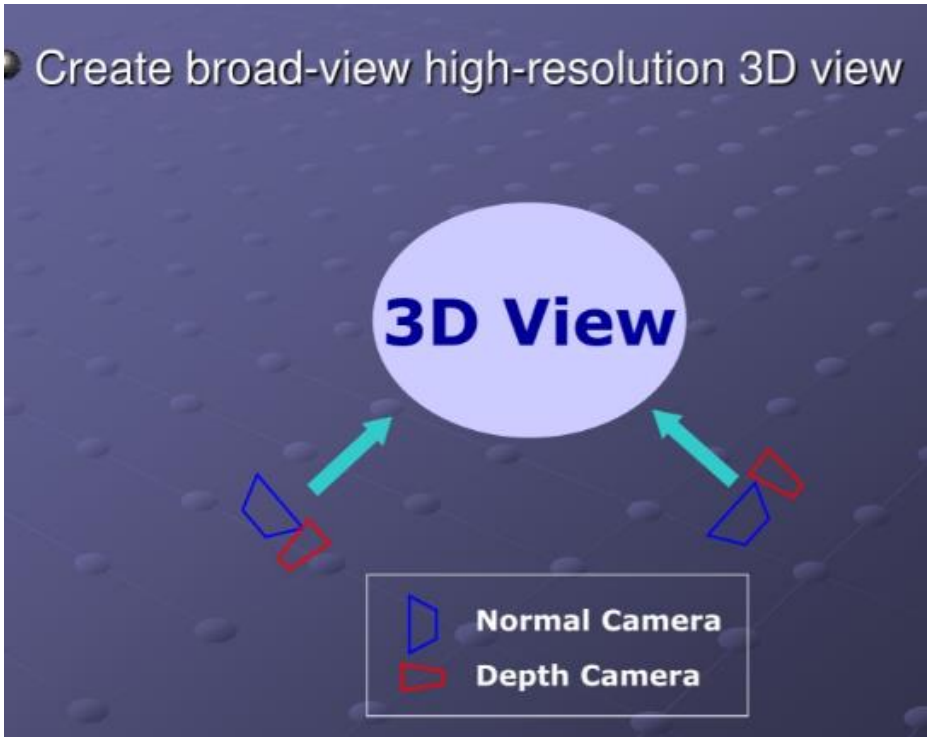
Making the Model: The computer then uses all the information to create a 3D model that you can view from any angle.

Depth from Stereo and 3D Stereoscopic Vision

- Definition:
- • Stereo vision estimates depth by comparing two images from slightly different viewpoints.

- How It Works:
- • Corresponding points in the left and right images are identified.
- • Disparities (differences in position) are used to calculate depth.

- Applications:
- • Robotics, VR/AR, Autonomous Vehicles

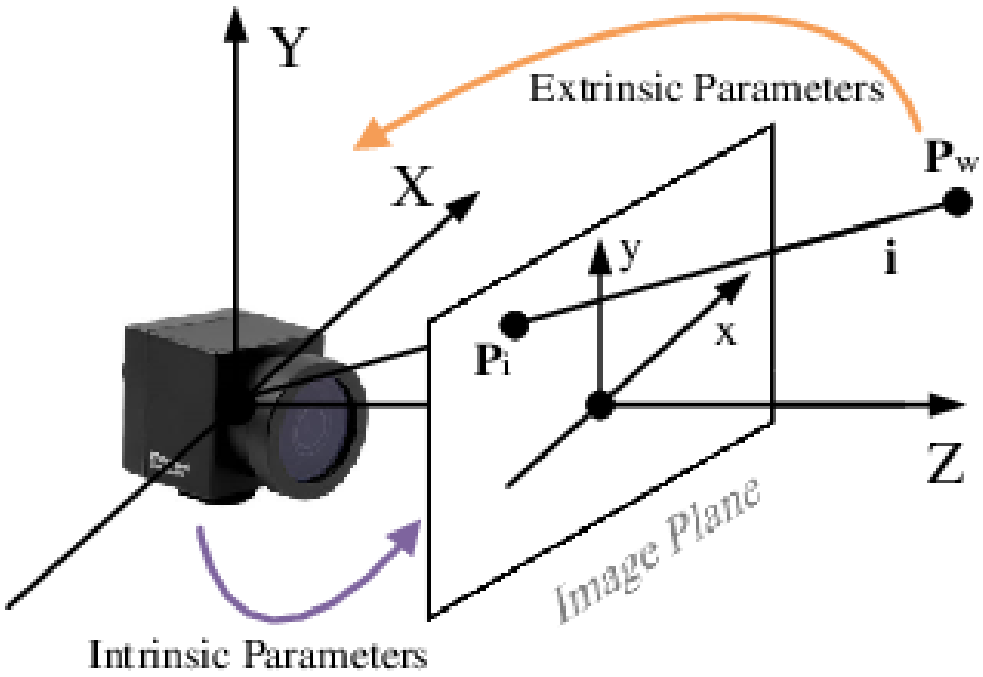


Camera Calibration

- Definition:
- • Camera calibration determines intrinsic and extrinsic parameters.

- Intrinsic Parameters:
- • Focal length, principal point, lens distortion coefficients.

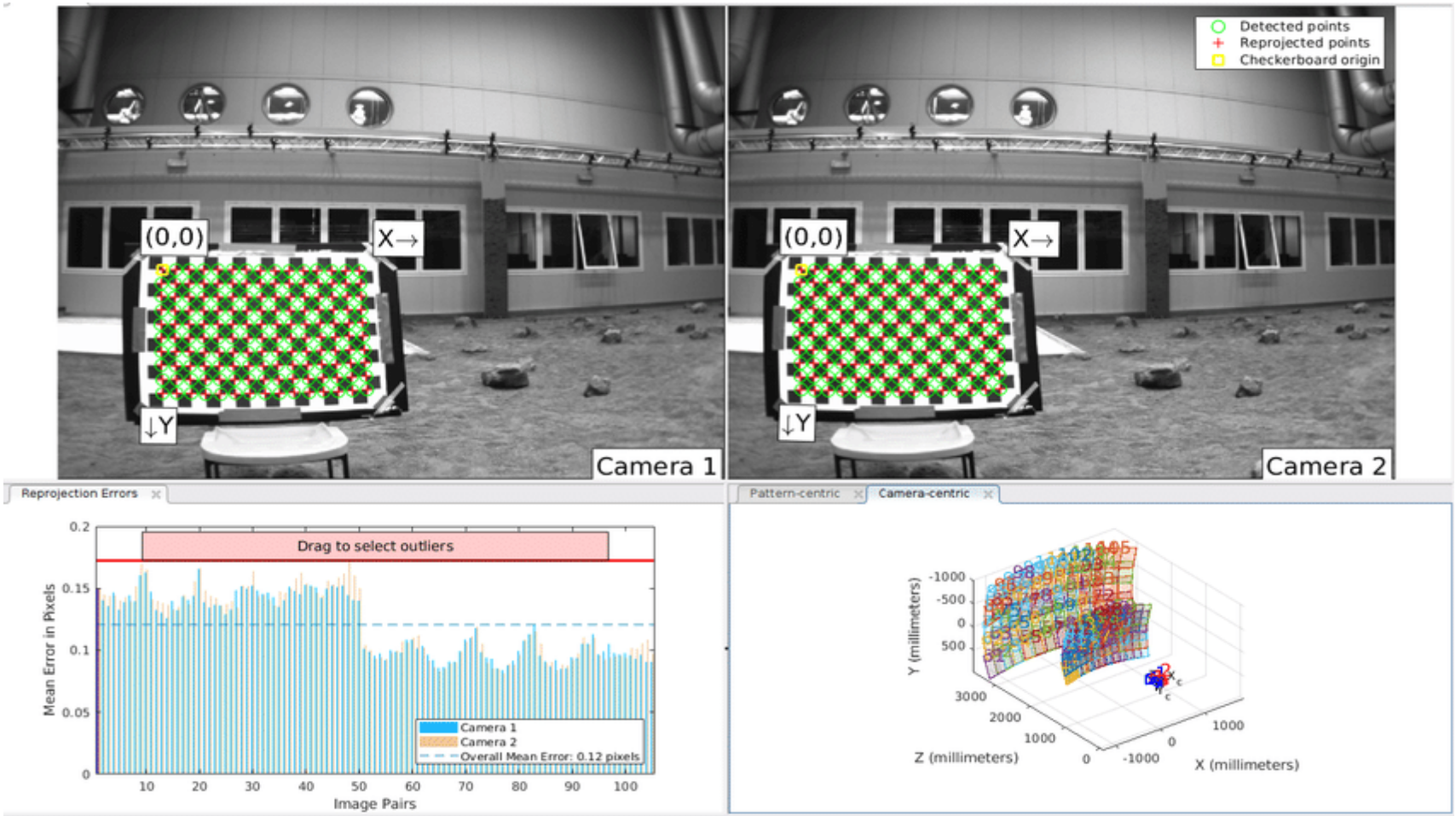
- Extrinsic Parameters:
- • Camera position and orientation relative to the scene.



Standard Camera Model

Camera Calibration

- Steps:
- 1. Use a calibration pattern (e.g., checkerboard).
- 2. Capture multiple images from different angles.
- 3. Estimate parameters using optimization algorithms.



Geometry of Epipolar Lines

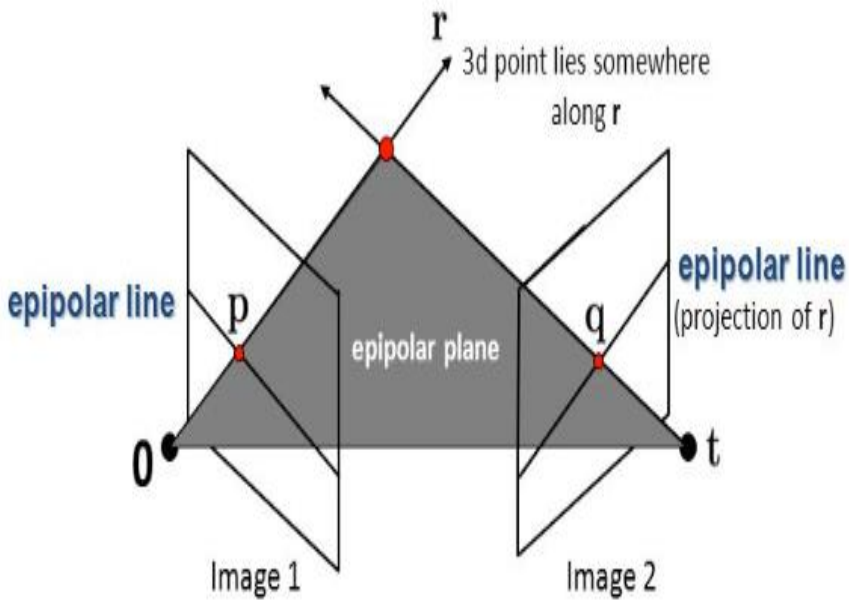
- Definition:
- **Epipolar geometry** describes the relationship between two views of a scene.

- Epipolar Lines:
- Constraints on corresponding points; they lie along epipolar lines.

- Key Concepts:
- Epipole: Intersection of the line connecting camera centers with the image plane.
- Epipolar Plane: Plane containing the two camera centers and a 3D point.

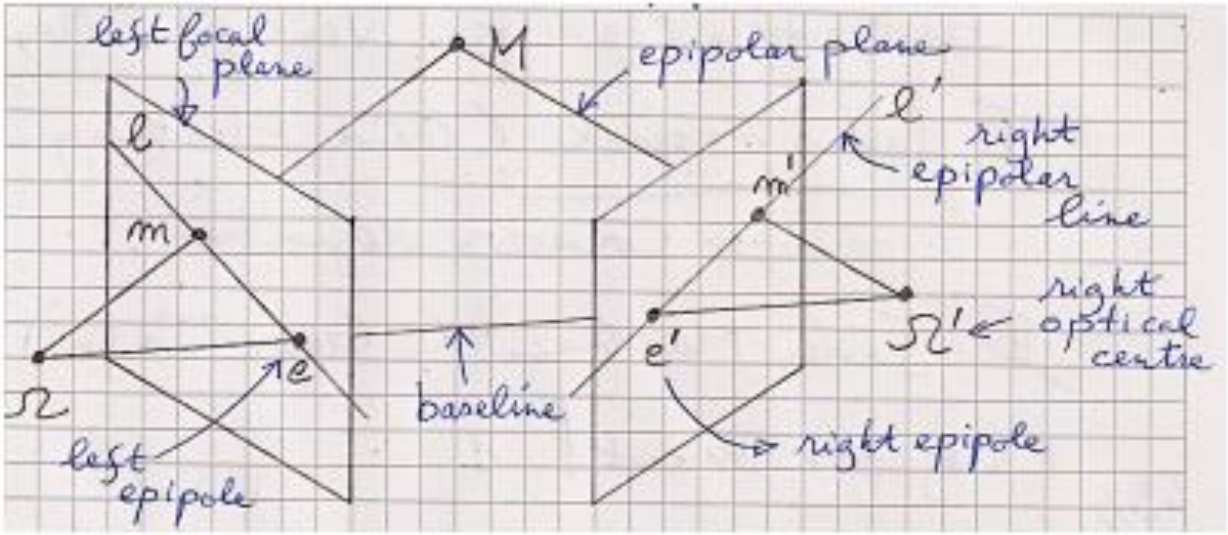
Two-view geometry

- Where do epipolar lines come from?



Geometry of Epipolar Lines

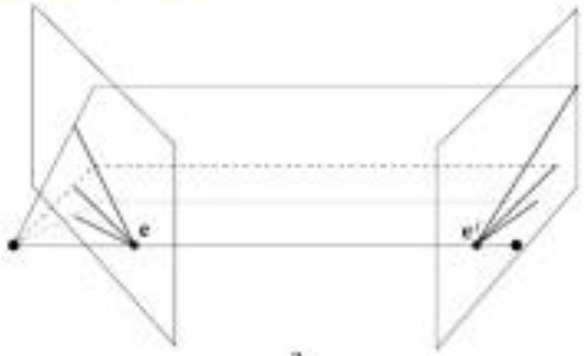
Epipolar Geometry



- Ω, m, M, m' and Ω' are coplanar.
- The epipolar plane cuts each focal plane through the epipolar line.
- Each point M has its own epipolar plane.
- All epipolar planes (epipolar pencil) intersect at the baseline ($\Omega\Omega'$)

Geometry of Epipolar Lines

Example 1: Converging Cameras



a

b

c

Figure from [Hartley and Zissermann 2003]

Geometry of Epipolar Lines

Example 2: In-Focal-Plane Moving Camera

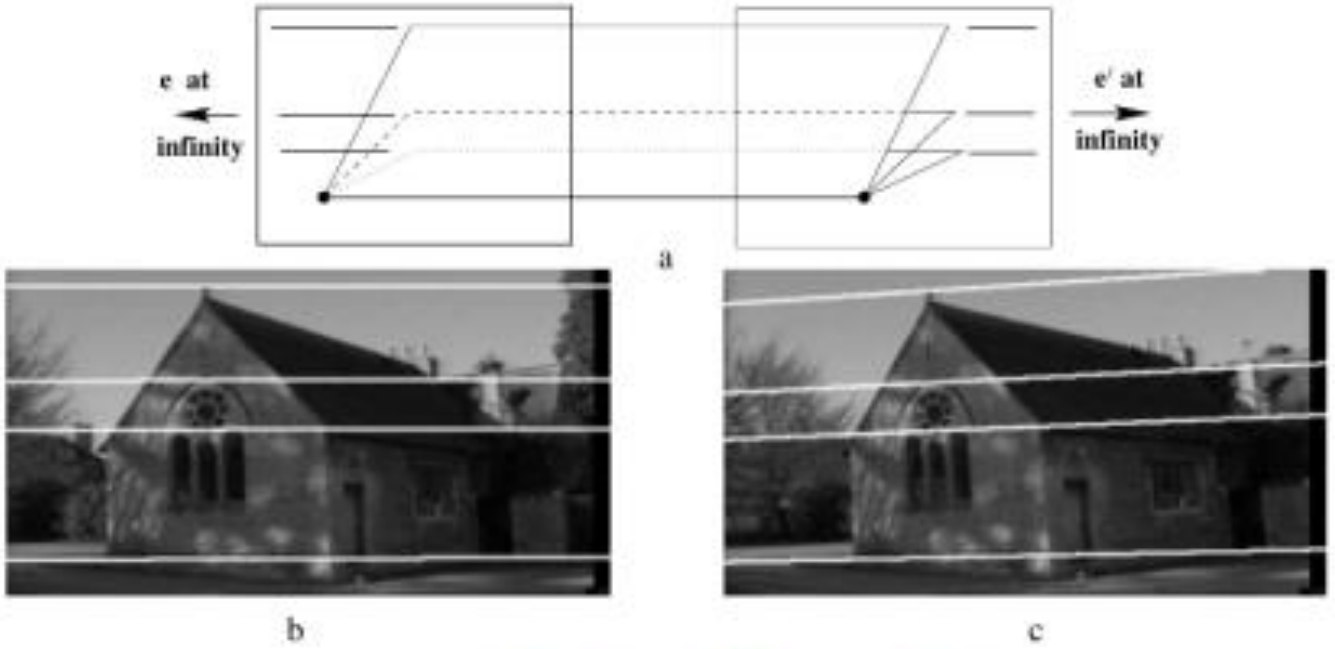


Figure from [Hartley and Zissermann 2003]

Geometry of Epipolar Lines

Example 3: Radially Moving Camera

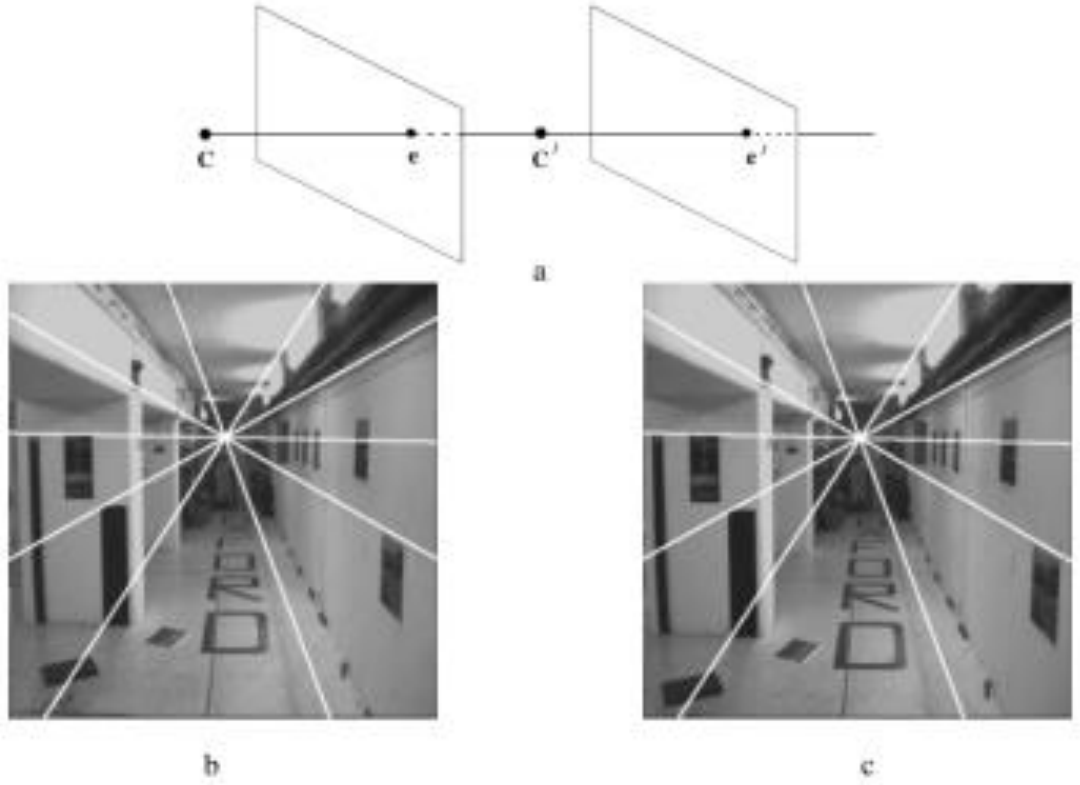
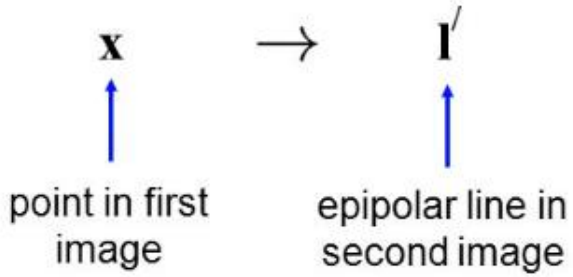


Figure from [Hartley and Zissermann 2003]

Algebraic representation of epipolar geometry

We know that the epipolar geometry defines a mapping

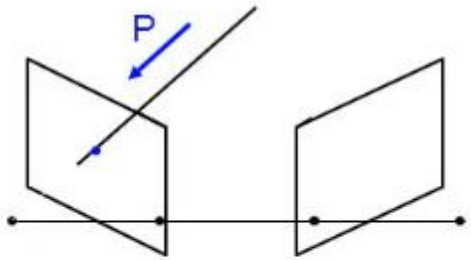


- the map ony depends on the cameras $\mathtt{P}, \mathtt{P}'$ (not on structure)
- it will be shown that the map is linear and can be written as $\mathbf{l}' = \mathtt{F}\mathbf{x}$, where $\mathtt{F}$ is a 3×3 matrix called the fundamental matrix

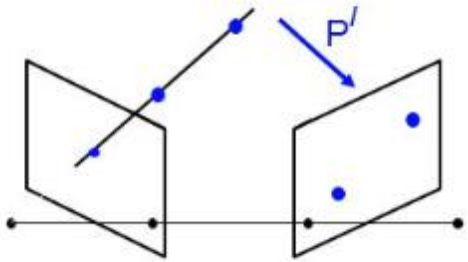
Derivation of Algebraic expression

Outline

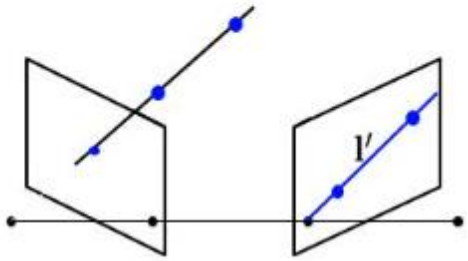
Step 1: for a point x in the first image back project a ray with camera P



Step 2: choose two points on the ray and project into the second image with camera P'



Step 3: compute the line through the two image points using the relation $\mathbf{l}' = \mathbf{p} \times \mathbf{q}$



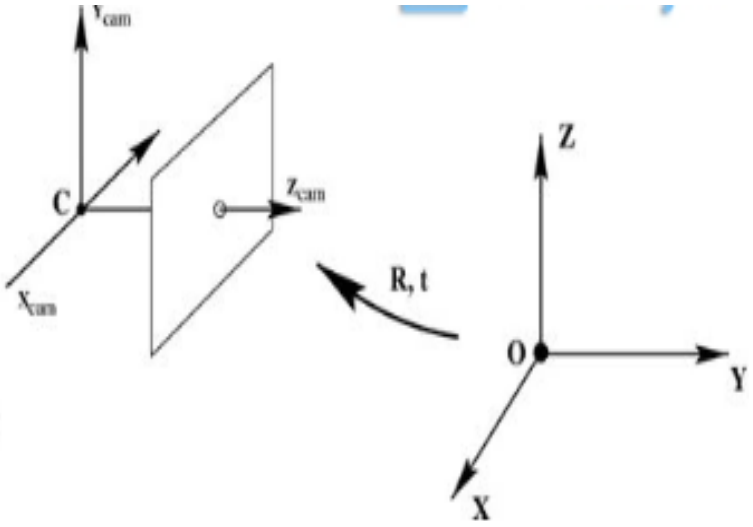
Derivation of Algebraic expression

- choose camera matrices

$$P = K[R \mid t]$$

internal calibration — rotation — translation

from world to camera coordinate frame



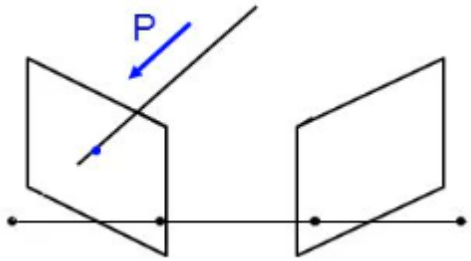
- first camera $P = K[I \mid 0]$

world coordinate frame aligned with first camera

- second camera $P' = K'[R \mid t]$

Derivation of Algebraic expression

Step 1: for a point x in the first image
back project a ray with camera $\mathrm{P} = \mathrm{K}[\mathrm{I} \mid \mathbf{0}]$



A point $\mathbf{x}$ back projects to a ray

$$\begin{pmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{pmatrix} = \mathrm{z}\mathrm{K}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathrm{z}\mathrm{K}^{-1}\mathbf{x}$$

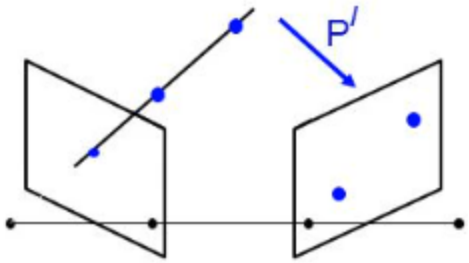
where $\mathbf{Z}$ is the point's depth, since

$$\mathrm{X}(\mathrm{z}) = \begin{pmatrix} \mathrm{z}\mathrm{K}^{-1}\mathbf{x} \\ 1 \end{pmatrix}$$

satisfies

$$\mathrm{P}\mathrm{X}(\mathrm{z}) = \mathrm{K}[\mathrm{I} \mid \mathbf{0}]\mathrm{X}(\mathrm{z}) = \mathbf{x}$$

Derivation of Algebraic expression



Step 2: choose two points on the ray and project into the second image with camera P'

Consider two points on the ray $\mathbf{X}(z) = \begin{pmatrix} zK^{-1}\mathbf{x} \\ 1 \end{pmatrix}$

- $\mathbf{Z} = 0$ is the camera centre $\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$
- $\mathbf{Z} = \infty$ is the point at infinity $\begin{pmatrix} K^{-1}\mathbf{x} \\ 0 \end{pmatrix}$

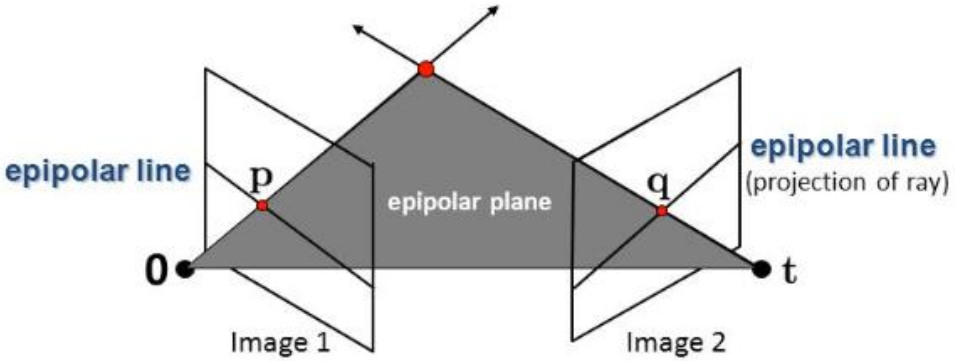
Project these two points into the second view

$$P' \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = K'[R \mid \mathbf{t}] \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = K'\mathbf{t} \qquad P' \begin{pmatrix} K^{-1}\mathbf{x} \\ 0 \end{pmatrix} = K'[R \mid \mathbf{t}] \begin{pmatrix} K^{-1}\mathbf{x} \\ 0 \end{pmatrix} = K'RK^{-1}\mathbf{x}$$

Fundamental Matrix

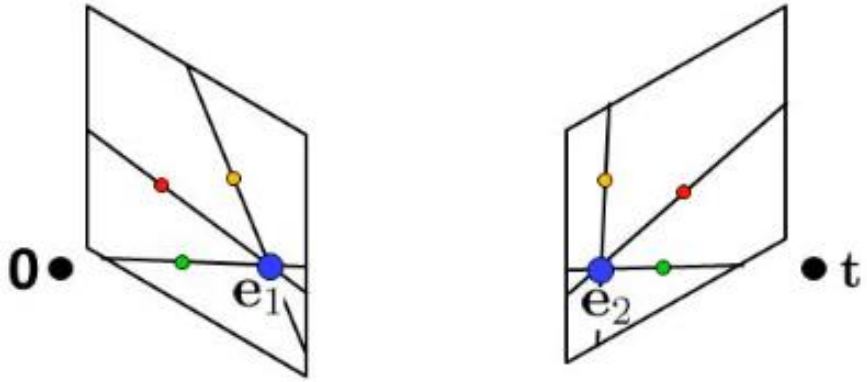
- Definition:• The fundamental matrix encodes the relationship between corresponding points in two views.
- Applications:
- • Essential in stereo vision and structure-from-motion algorithms.

Fundamental matrix



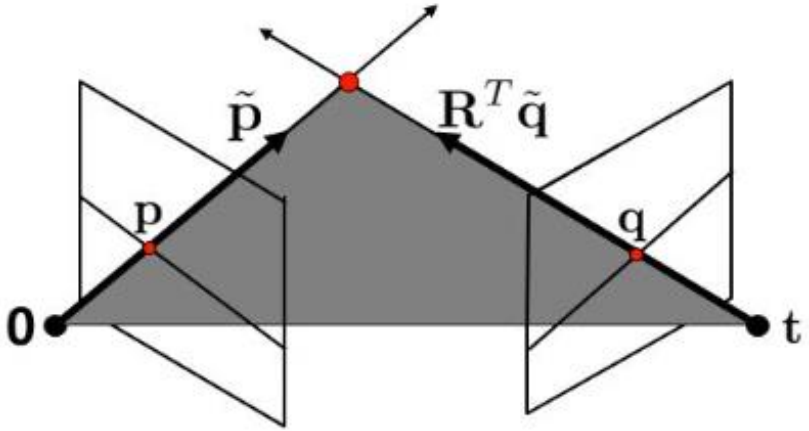
- This *epipolar geometry* of two views is described by a Very Special 3x3 matrix $\mathbf{F}$, called the *fundamental matrix*
- $\mathbf{F}$ maps (homogeneous) *points* in image 1 to *lines* in image 2!
- The epipolar line (in image 2) of point $\mathbf{p}$ is: $\mathbf{Fp}$
- *Epipolar constraint* on corresponding points: $\mathbf{q}^T\mathbf{Fp} = 0$

The epipoles



- Two special points: $\mathbf{e}_1$ and $\mathbf{e}_2$ (the *epipoles*): projection of one camera into the other
- All of the epipolar lines in an image pass through the epipole

Fundamental matrix



$\mathbf{K}_1$: intrinsics of camera 1

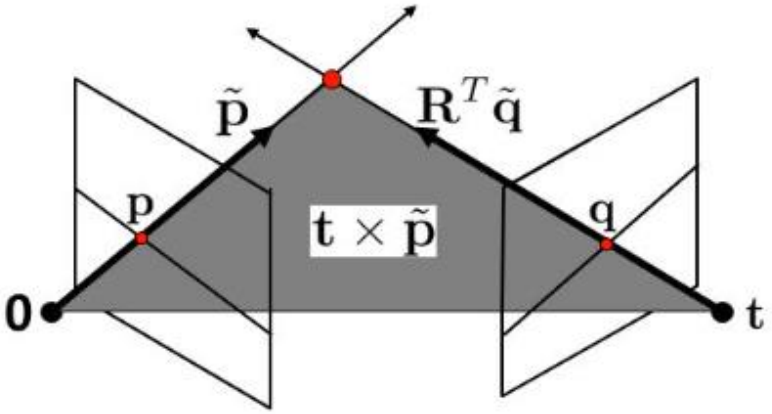
$\mathbf{K}_2$: intrinsics of camera 2

$\mathbf{R}$: rotation of image 2 w.r.t. camera 1

$\tilde{\mathbf{p}} = \mathbf{K}_1^{-1}\mathbf{p}$: ray through **p** in camera 1's (and world) coordinate system

$\tilde{\mathbf{q}} = \mathbf{K}_2^{-1}\mathbf{q}$: ray through **q** in camera 2's coordinate system

Fundamental matrix

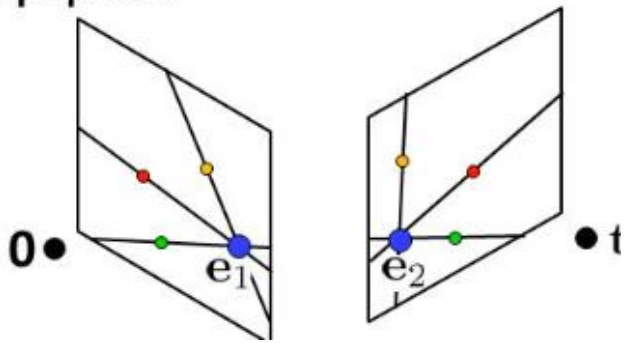


- $\tilde{\mathbf{p}}$, $\mathbf{R}^T\tilde{\mathbf{q}}$, and $\mathbf{t}$ are coplanar
- epipolar plane can be represented as $\mathbf{t} \times \tilde{\mathbf{p}}$

$$(\mathbf{R}^T\tilde{\mathbf{q}})^T(\mathbf{t} \times \tilde{\mathbf{p}}) = 0$$

Properties of the Fundamental Matrix

- $\mathbf{Fp}$ is the epipolar line associated with $\mathbf{p}$
- $\mathbf{F}^T\mathbf{q}$ is the epipolar line associated with $\mathbf{q}$
- $\mathbf{Fe}_1 = \mathbf{0}$ and $\mathbf{F}^T\mathbf{e}_2 = \mathbf{0}$
- All epipolar lines contain epipole



Fundamental Matrix Summary

For 2 images captured by cameras with distinct optical centres, the fundamental matrix is the unique 3×3 rank 2 matrix F that satisfies $m'^t F m = 0$, for all corresponding pairs of points (m, m').

- **Epipolar lines**: $l' = Fm$ and $l = m'^t F$ are the right and left epipolar lines respectively.
- **Epipoles**: Since $e' \in l'$, we have $\forall m$, $e'^t F m = 0$. Then $e'^t F = 0$. Similarly, $Fe = 0$.
- **Rank**: F is an homogeneous (8 DoF) 3×3 matrix, and has rank 2 ($\det F = 0$), so it actually has 7 DoF.

Projection Techniques

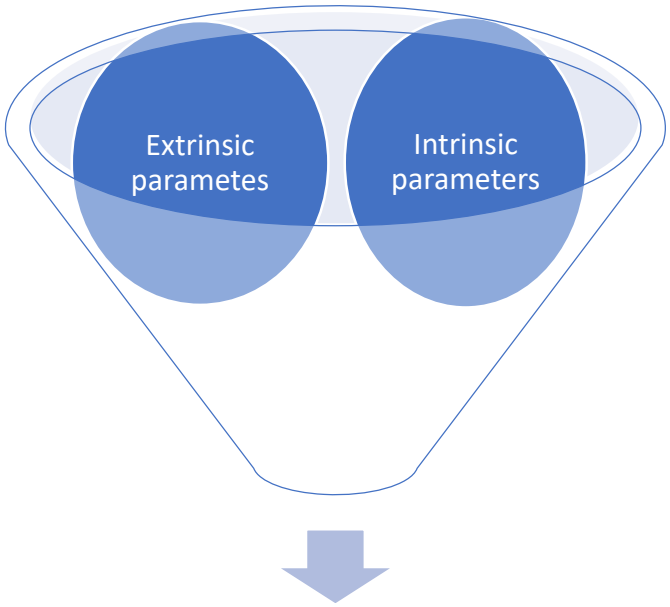
- Definition:
- • Projection maps 3D points to 2D image coordinates using a camera matrix.

- Camera Matrix:
- • Combines intrinsic and extrinsic parameters.

- Applications:
- • Used in rendering 3D models and augmented reality applications.

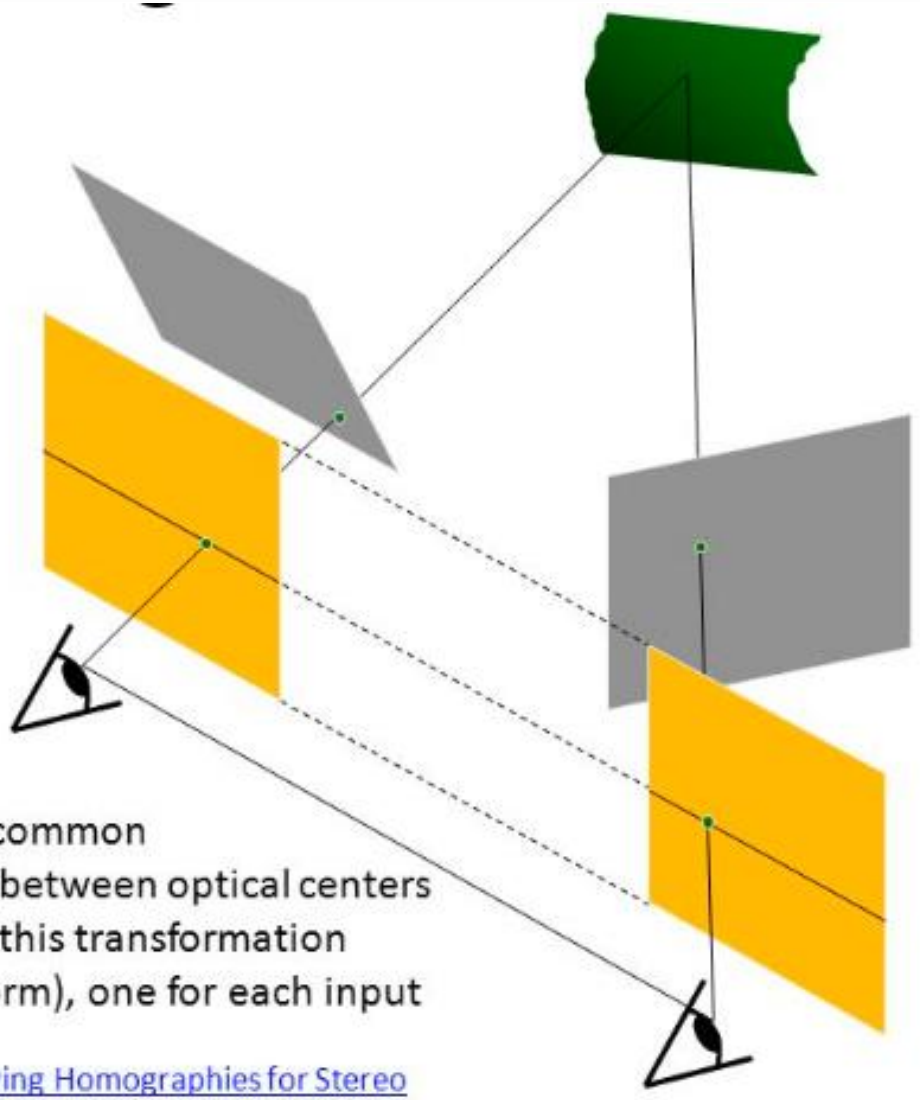
Projection Techniques

- A **camera matrix** is like the brain of a camera. It's a mathematical tool that helps the camera convert 3D points in the real world into 2D points in a photo or video.
- Think of it this way:
- **The real world (3D)** is like a stage with actors.
- **The photo (2D)** is like the snapshot taken by the camera.
  The camera matrix uses information about the camera (like how zoomed in it is and where the center of the image is) to calculate where each actor should appear in the photo.



camera matrix

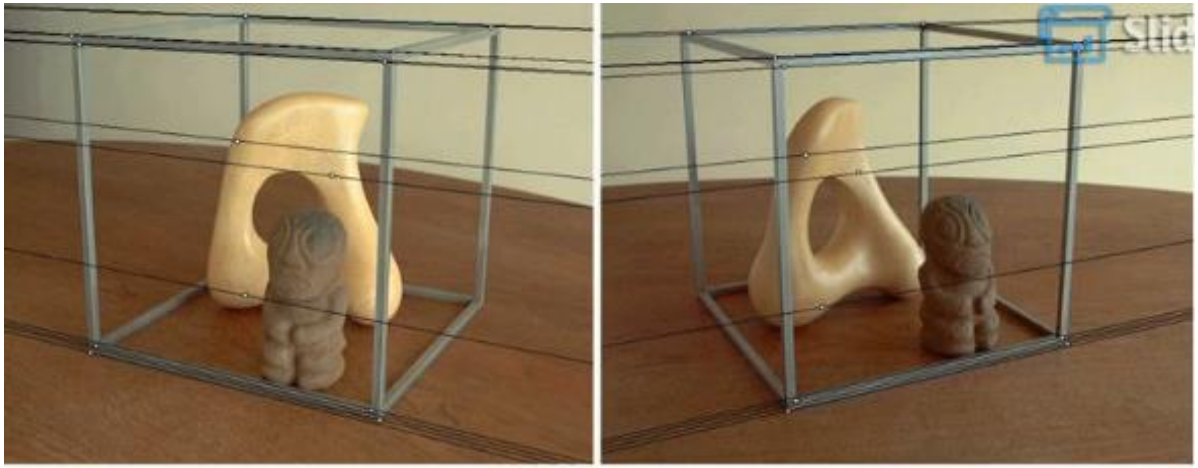
Projection Techniques



- reproject image planes onto a common
- plane parallel to the line between optical centers
- pixel motion is horizontal after this transformation
- two homographies (3x3 transform), one for each input image reprojection

➢ C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Projection Techniques

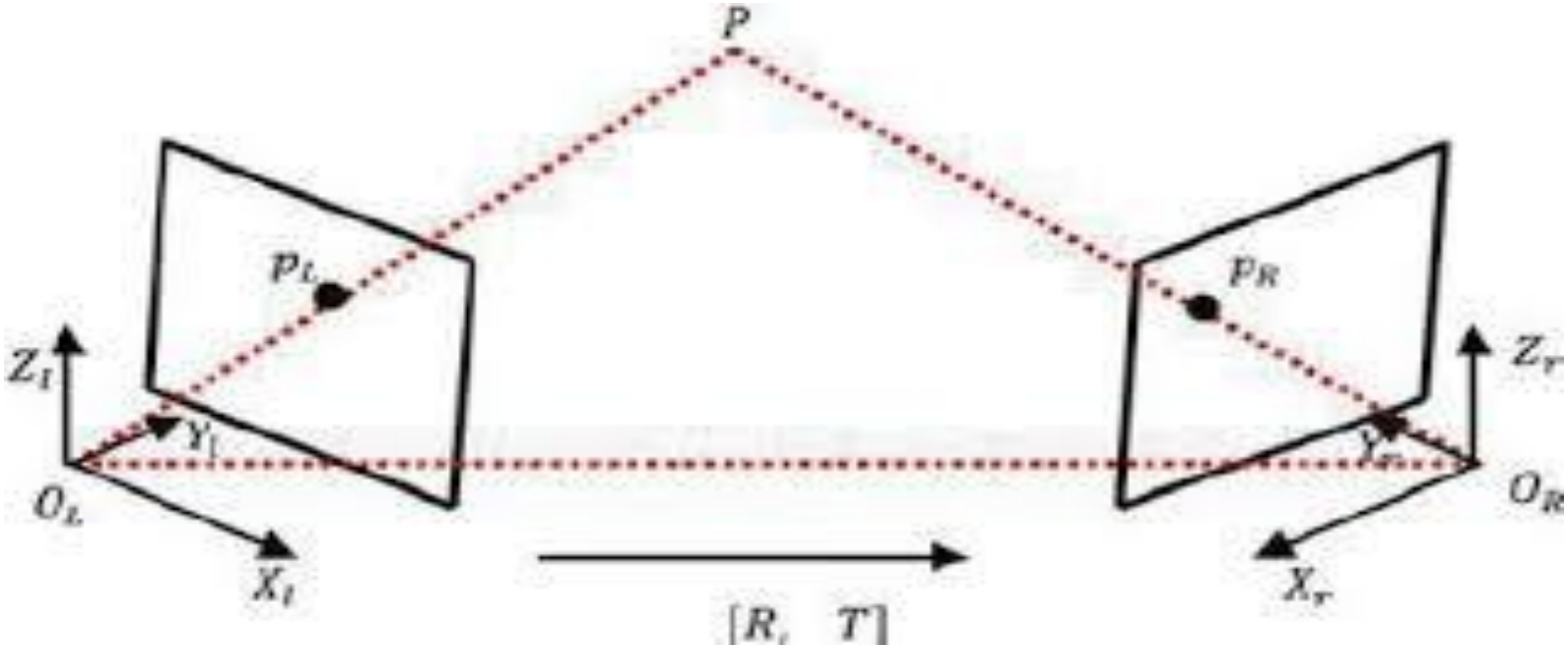


Original stereo pair

After rectification

Determining 3D Coordinates

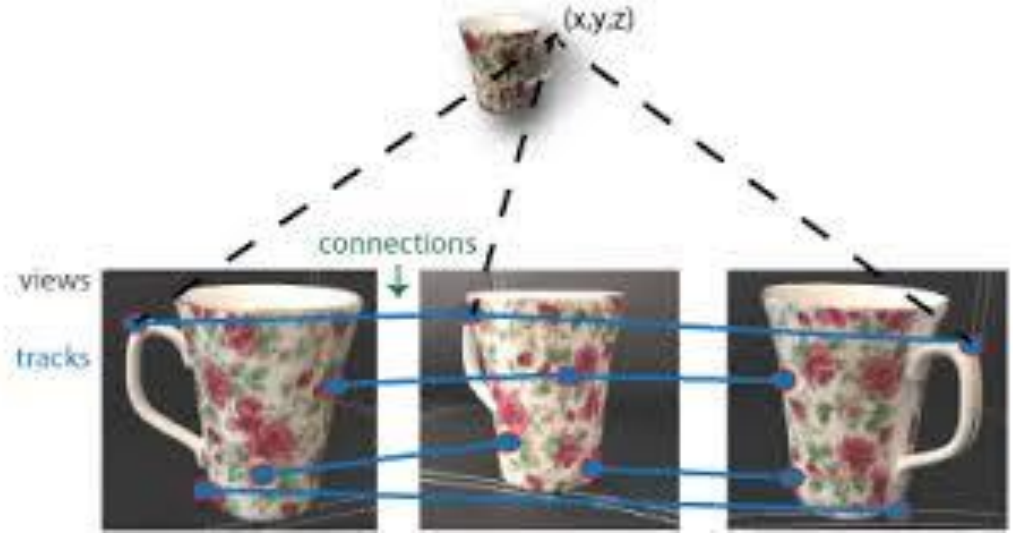
- Definition:
- • **Triangulation** determines 3D points from corresponding 2D points in multiple images.
- Key Techniques:
- • Use known camera matrices and disparities to compute 3D positions.
- Applications:
- • Used in generating point clouds and 3D models from stereo images.



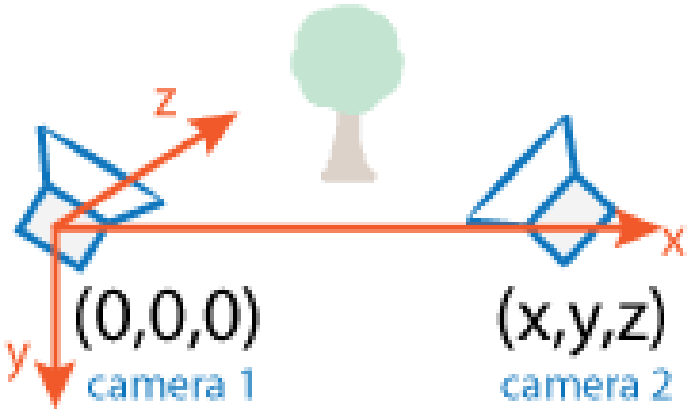
Interpreting Images from Various Angles

- Definition:
- • Multi-view reconstruction combines images from different angles to create a 3D model.

- Key Techniques:
- • Structure from Motion (SfM): Estimates camera poses and reconstructs scenes.
- • Neural Techniques: Use deep learning to refine models from limited views.



Structure from Motion

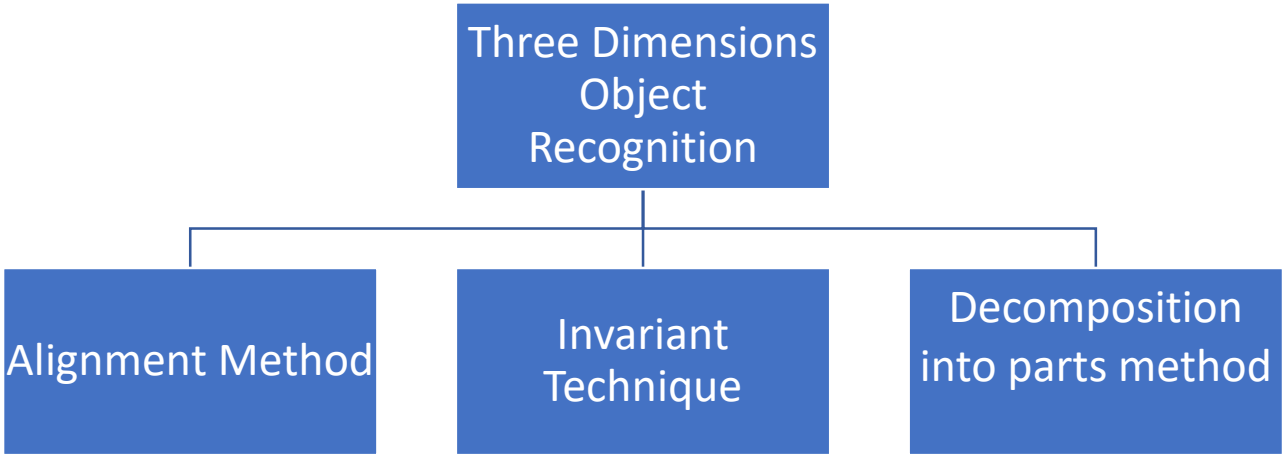


- Definition:
- • Estimates camera motion and 3D structure from overlapping images.

- Steps:
- 1. Feature detection and matching.
- 2. Motion estimation.
- 3. Sparse and dense 3D reconstruction.

- Applications:
- • Used in cultural heritage, urban planning, and animation.

Three Dimensions Object Recognition



Three Dimensions Object Recognition

1/ Alignment Method

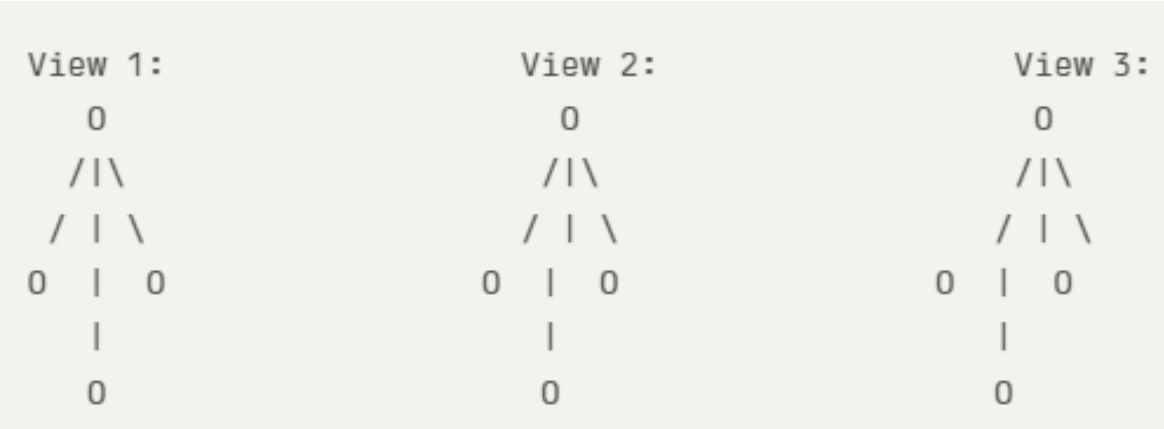
•The alignment method focuses on positioning and orienting 3D objects accurately in space. This can involve techniques like Iterative Closest Point (ICP), which aligns point clouds by minimizing the distance between corresponding points. The process often includes estimating the object's pose iteratively using algorithms such as Newton's method and Levenberg-Marquardt minimization to refine the alignment based on feature correspondences between 2D images and 3D models

In this illustration, the object (represented by "O") starts in an initial position and is aligned to match the reference position through iterative adjustments.

Three Dimensions Object Recognition

2/ Invariant Technique

Invariant techniques aim to recognize objects regardless of variations in viewpoint, scale, or lighting. These methods typically leverage features that remain consistent under different conditions, such as shape descriptors or histogram of oriented gradients (HOG). By focusing on these invariant features, the recognition system can effectively match objects even when they appear differently in various images or environments.

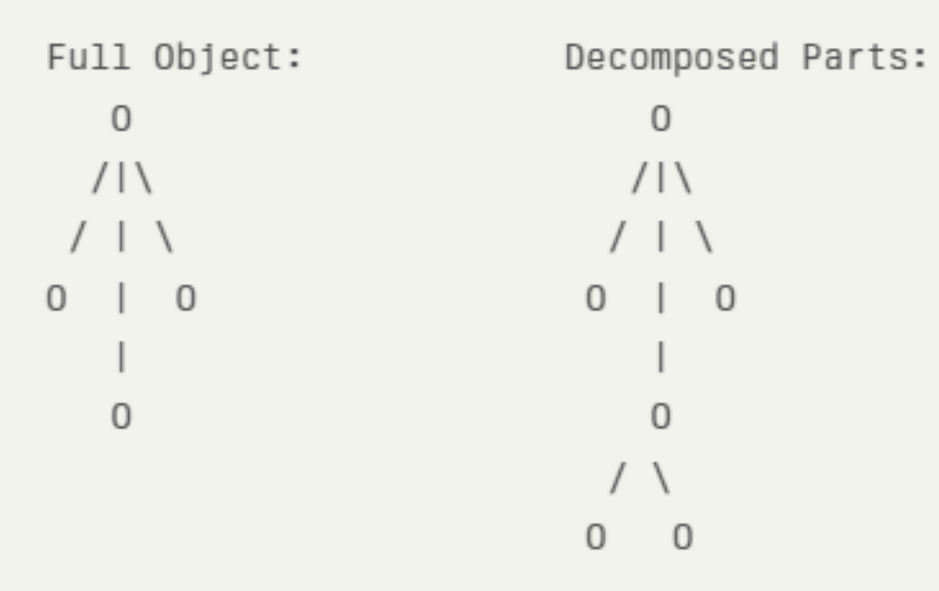


Here, the same object is viewed from different angles (View 1, View 2, View 3). The invariant features (e.g., shape descriptors) allow recognition despite the variations in perspective.

Three Dimensions Object Recognition

3/ Method of Decomposition into Parts

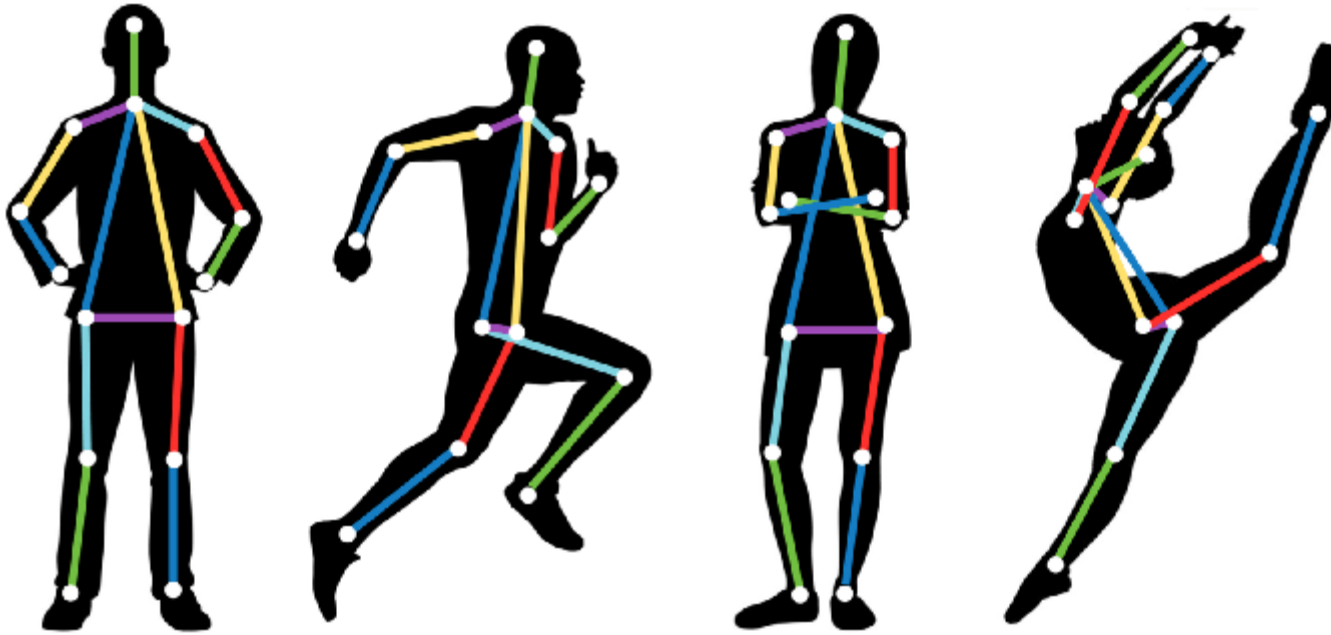
•This approach involves breaking down complex objects into simpler, recognizable parts. The method utilizes a part-based model where each part is identified and localized within the image. By analyzing the spatial relationships and interactions among these parts, the system can categorize and recognize the overall object. This technique is particularly useful for handling occlusions and variations in object appearance



In this illustration, the full object is represented at the top, while below it shows the decomposition into parts. Each part can be recognized independently, and their spatial relationships help identify the overall object.

Object positioning based on a single image.

Pose estimation is a computer vision task that involves determining the **position** and **orientation** of an object or human body in an image or video. This process identifies specific points, known as **keypoints**, which represent various parts of the object, such as joints in the human body.

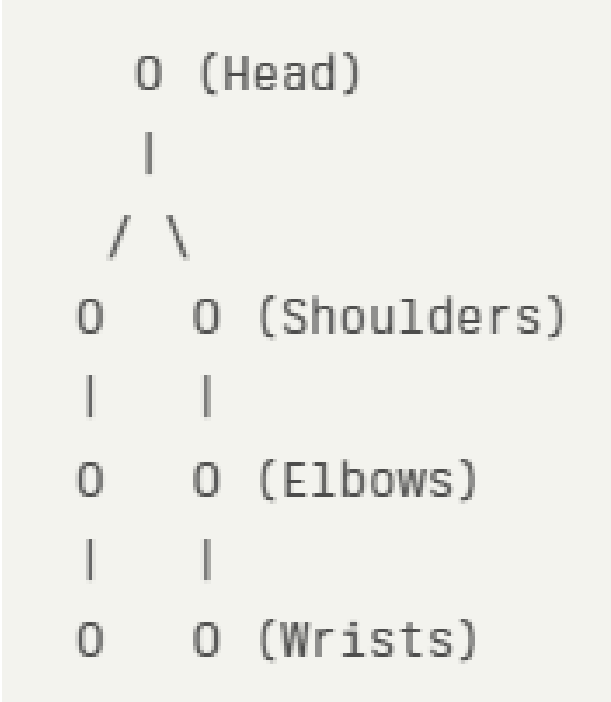


Object positioning based on a single image.

Illustration

In this diagram, each "O" represents a **keypoint** (e.g., head, shoulders, elbows) that pose estimation algorithms would identify and track within an image or video sequence. These keypoints can be used to analyze movement patterns or to interact with digital environments in applications like fitness tracking, gaming, and robotics

Overall, pose estimation is crucial for applications ranging from human-computer interaction to advanced robotics, enabling systems to understand and react to human movements effectively.

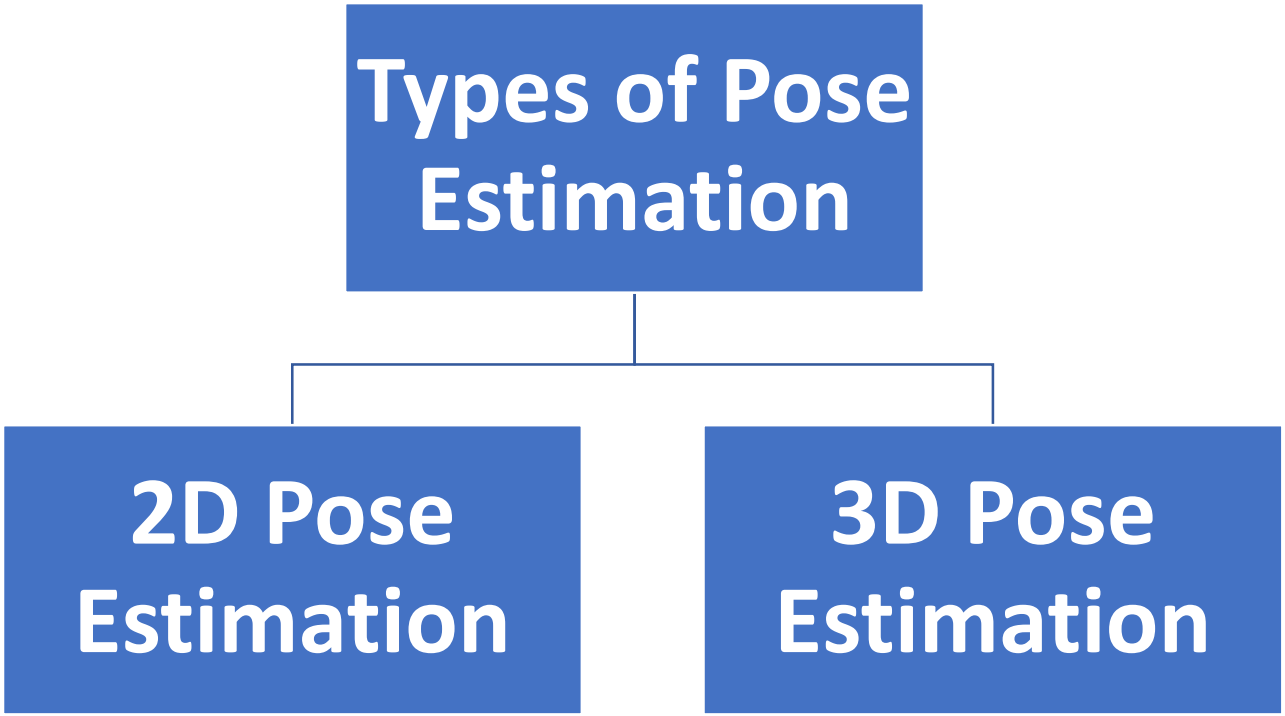


Object positioning based on a single image.

Types of Pose Estimation

2D Pose Estimation: Determines the position of key points in a 2D plane (image coordinates).

3D Pose Estimation: Identifies the key points in 3D space, considering depth information.



Object positioning based on a single image.

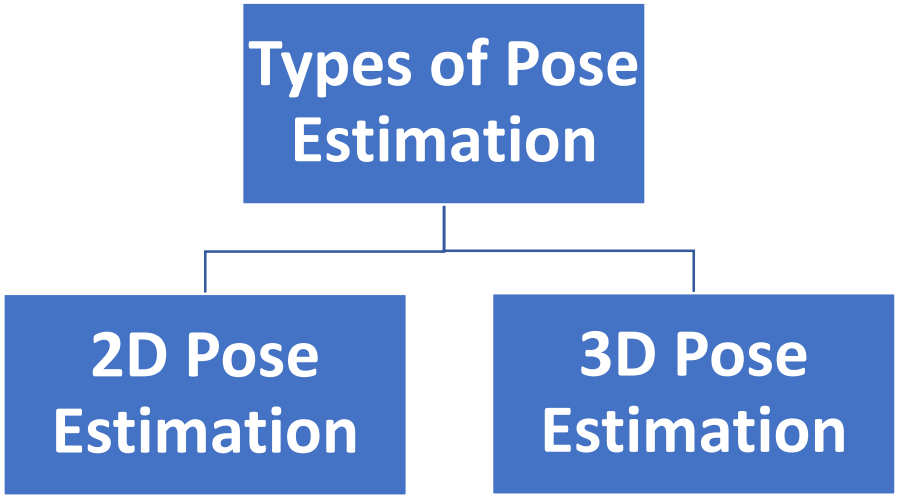


Illustration: 2D Pose Estimation Example
Imagine a person standing with arms outstretched. A pose estimation algorithm would:
•Detect key joints (e.g., head, shoulders, elbows, wrists).
•Represent these as points.
•Connect these points with lines to create a stick figure.

Illustration: Keypoints in 3D Space
For 3D pose estimation, depth information is also computed, which could involve:
Using stereo vision (two cameras).
Leveraging depth sensors like Kinect.

Object positioning based on a single image.

Methods for Pose Estimation

a. Top-Down Approach

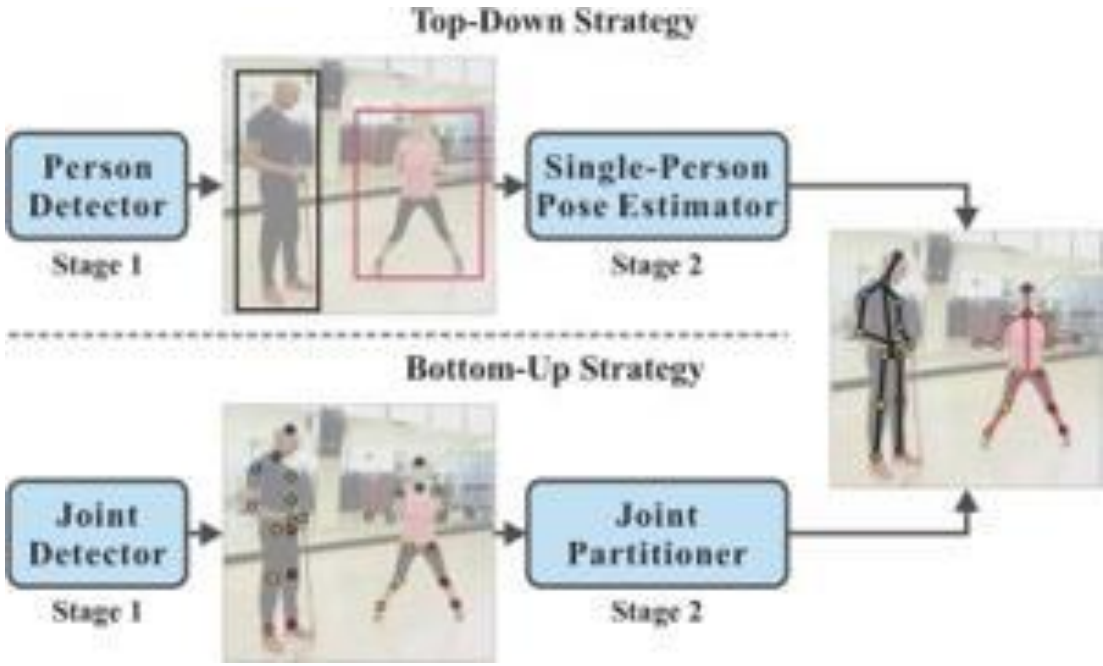
1.Detects the person in the image first using an object detection algorithm.
2.Then applies pose estimation on the detected region.

Example Models: Mask R-CNN, AlphaPose.

b. Bottom-Up Approach

1.Detects all key points in the image irrespective of the number of people.
2.Groups these points to form skeletons for each individual.

Example Models: OpenPose, PifPaf.



Challenges

Challenges in 3D reconstruction:
The main challenges of 3D scene reconstruction include:
Occlusions: Portions of the scene may be hidden from view, making it difficult to capture complete data.
Lighting Variability: Changes in lighting conditions can affect the accuracy of texture and depth perception.
Noisy or Incomplete Data: Sensor data, such as from cameras or LiDAR, may include errors or gaps.
Difficult Surfaces: Reflective, transparent, or textureless surfaces are particularly challenging to reconstruct.
Computational Efficiency: Balancing accuracy and speed is critical, especially for large-scale or real-time applications.

References

1. Dana H. Ballard & Christopher M. Brown. Computer Vision Prentice Hall, Inc, 1982
2. Robert M. Haralick & Linda G. Shapiro. Computer and Robot Vision, Vol-I, Addison-Wesley Publishing Company, 1992
3. Robert M. Haralick & Linda G. Shapiro. Computer and Robot Vision, Vol-II, Addison-Wesley Publishing Company, Inc, 1993
4. Linda Shapiro & Azriel Rosen eld. Computer Vision and Image Processing, Academic Press, Inc, 1992
5. Berthold Klaus Paul Horn. Robot Vision , MIT Press McGraw-Hill Book Company, 1986
6. Robert J. Schalko. Digital Image Processing and Computer Vision, John Wiley & Sons Inc, 1989
7. George Stockman and Linda Shapiro. Three Dimensional Computer Vision. Prentice Hall 2000.
8. David Marr. Vision, W. H Freeman and Company, NY, 1982
9. Rafael C. Gonzalez and Paul Wintz. Digital Image Processing, Third edition, Addison Wesley, MA. (Now with Prentice Hall, eective 1999).
10. Ernest Hall. Computer Image Processing and Recognition, second edition, Academic press 1982.
11. Azriel Rosenfeld and Avinash C. Kak. Digital Picture Processing, Vol. 1 & Vol. 2, Academic Press, 1982.
12. Robert J. Schalko. Digital Image Processing and Computer Vision: An introduction to theory and implementations, John Wiley & Sons, New York, 1989.
13. William K. Pratt. Digital Image Processing, John Wiley & Sons, 1993.