

USN									
-----	--	--	--	--	--	--	--	--	--



Internal Assessment Test 1 – Sept. 2025

Subject/Code: Computer Vision BAI151A				
Date: 26/09/2025	Duration: 90 mins	Max. Marks: 50	Semester: 05	Branch: AI-DS/CS-DS

Sl.	Answer any FIVE FULL Questions	Marks	CO	RBT
1	A) Write a brief history of Computer Vision B) Explain the working of a digital camera with a neat block diagram.	5 + 5	CO1	L2
2	A) Explain histogram equalization with an example. B) Explain separable filtering with a formula.	5 + 5	CO2	L3
3	A) What is non-linear filtering? Give examples. B) Explain bilateral filtering and its applications.	5 + 5	CO2, CO3	L2, L3

4.	Explain applications of Fourier Transform in image sharpening, blurring, and noise removal.	10	CO2	L3
5.	A) What are wavelets? Explain their applications in image analysis. B) Differentiate between interpolation and decimation in pyramids.	5 + 5	CO2	L2
6.	Explain parametric geometric transformations and feature-based morphing with examples.	10	CO2	L2

CI

CCI

HOD

Solution:

Answer Key Computer Vision (Internal Assessment Test 1)

1A) Write a brief history of Computer Vision.

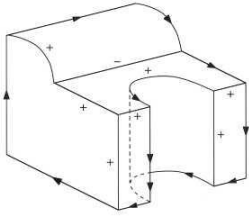
Ans:

1970s: Birth of Computer Vision

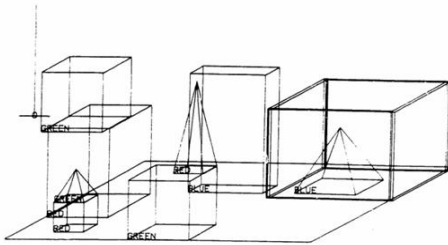
Initially seen as a way to help robots “see” like humans.

Focus was on basic tasks like edge detection and reconstructing 3D structures from 2D images.

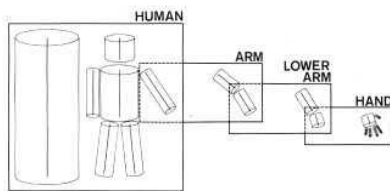
Line Labelling: Identifying which edges belong to which objects.



Blocks World: Using simple cubes and shapes to test vision systems.



Generalized Cylinders: Representing complex objects with simple tube-like parts.

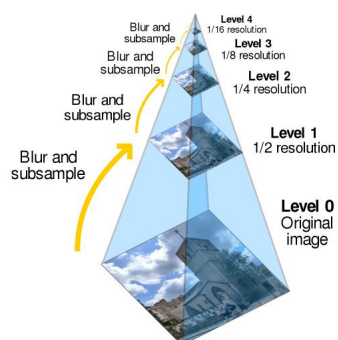


1980s: Math and Models Era

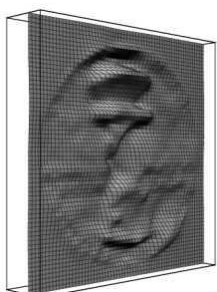
Need for stronger mathematical tools to analyze real images.

Major concepts:

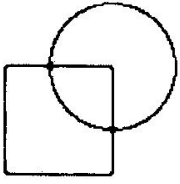
Image Pyramids: Viewing an image at multiple scales.



Shape from X: Inferring 3D shape from shading, texture, or focus.



Edge Detection: Finding the outline of objects in an image. Canny edge detector became a landmark method.



Physically Based Models: Using physics for light, motion, and shape.

Kalman Filters for tracking and *Markov Random Fields (MRF)* for smoothing images.

1990s: Structure and Motion Focus

Shifted towards understanding 3D world and motion.

Key developments:

structure from Motion (SfM): Reconstruct 3D scenes from multiple images/videos.

Eigenfaces: Early face recognition using mathematical patterns.

Image Segmentation: Splitting images into meaningful parts.

Feature Tracking: Following corners or textures across video frames.

Graph Cuts: Mathematical optimization for image segmentation.

Collaboration with computer graphics for realistic 3D scene modeling.

2000s: Graphics + Data + Learning

Faster computers and more visual data enabled new progress.

Highlights:

Image-based Modeling: Building 3D models directly from photos.

Computational Photography: HDR imaging, panorama stitching, inpainting, and texture synthesis.

Object Recognition using Features: Identifying objects based on distinctive parts (corners, curves, textures).

2010s: Deep Learning Revolution

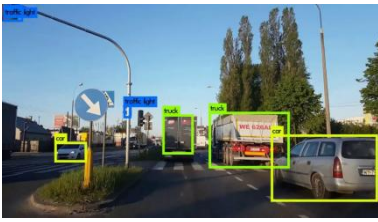
Deep neural networks completely transformed computer vision.

Breakthroughs included:

AlexNet (2012): CNNs proved superior in the ImageNet challenge.

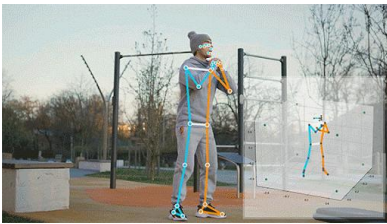
Large datasets (*ImageNet*, *COCO*, *LVIS*) for training models.

Object Detection: Methods like YOLO and R-CNN.



Semantic Segmentation: Labeling every pixel in an image.

Pose Estimation: Detecting human joints in real time.



Depth Estimation and Optical Flow: Understanding 3D structure and motion.

Augmented Reality (AR) and SLAM: Blending virtual content with the real world and enabling self-navigation in unknown environments.

1B) Explain the working of a digital camera with a neat block diagram.

Ans:

Working of a Digital Camera

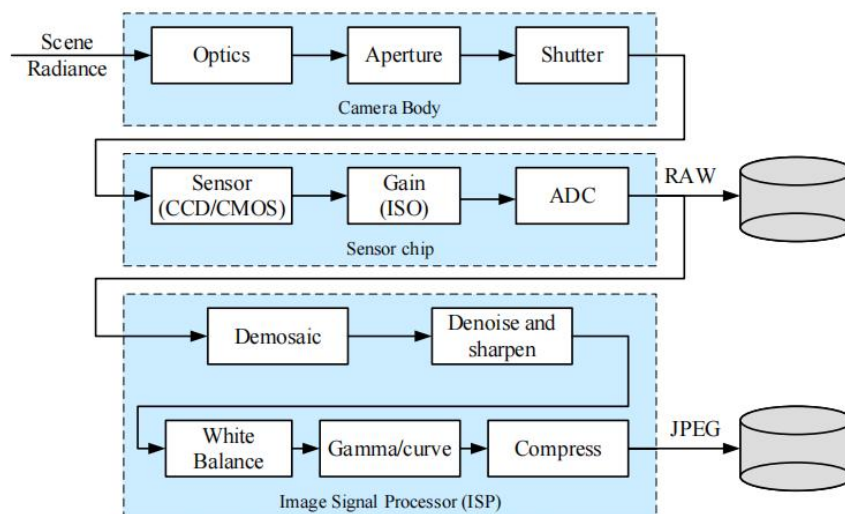


Figure 2.23 Image sensing pipeline, showing the various sources of noise as well as typical digital post-processing steps.

A digital camera captures light and converts it into a digital image through a sequence of steps:

Optics (Lens System)

Light from the scene passes through the camera's lens.

The lens focuses light rays onto the image sensor.

Aperture

The aperture is an adjustable opening in the lens.

It controls how much light enters the camera (like the pupil of the eye).

Shutter

The shutter controls how long the sensor is exposed to light.

A short shutter speed captures fast motion, while a long shutter speed allows more light in.

Sensor (CCD/CMOS)

The sensor contains millions of pixels that convert incoming light (photons) into electrical signals.

Each pixel records light intensity.

Gain

Weak electrical signals from the sensor are amplified using gain.

This improves brightness but may add noise.

ADC (Analog-to-Digital Converter)

Converts the amplified analog signals into digital numbers that a computer can process.

Demosaic

Most sensors use a Bayer filter (RGB color filter).

Demosaicing reconstructs full-color information by combining red, green, and blue pixel data.

Denoise and Sharpen

Noise reduction algorithms remove unwanted grainy patterns.

Sharpening enhances edges and fine details in the image.

White Balance

Adjusts color tones to match the lighting condition (e.g., daylight, tungsten light).

Gamma Curve

The gamma curve adjusts image brightness and contrast for natural-looking images.

Compression

The processed image is compressed (e.g., JPEG) to reduce storage size. Finally, the image is stored on the stored device.

2A) Explain histogram equalization with an example.

Ans:

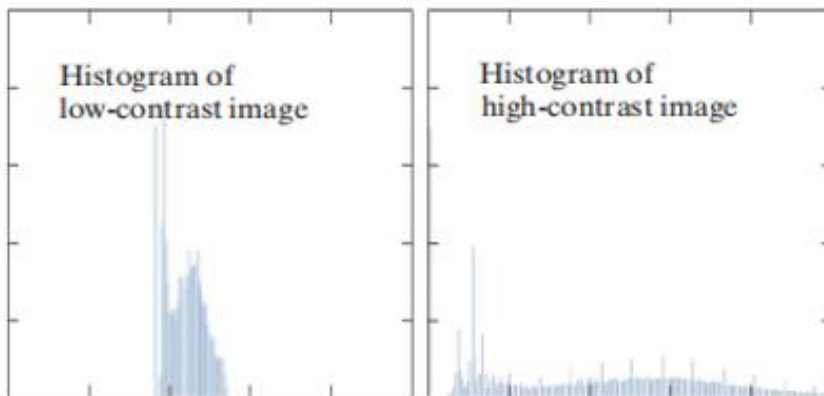
Histogram

A histogram of an image is a plot of frequency of pixel intensity values.

Example: If your image is grayscale (values 0–255), the histogram tells you how many pixels have intensity 0, how many have 1, 2, ... up to 255.

So, histogram = frequency distribution of pixel values.

To check intensity distribution, we draw a histogram = a graph that shows how many pixels have each brightness level (dark → bright).



Histogram Equalization:

Histogram Equalization is an image processing technique used to **improve the contrast** of an image. It works by redistributing the intensity values (pixel brightness levels) so that they cover the entire possible range (0–255 for 8-bit images) more evenly.

To do this, we use something called the cumulative distribution function (CDF).

It basically tells us: "Up to this Intensity value, what fraction of pixels have smaller values?"

For each pixel brightness, find its percentile. Then remap it to a new brightness between 0 and 255.

$$\text{new intensity} = \frac{CDF(\text{current intensity}) - CDF(\min)}{N - CDF(\min)} \times (L - 1)$$

Where:

- **CDF(current intensity)** → cumulative count for that pixel's intensity
- **CDF(min)** → the first non-zero CDF (to avoid mapping dark pixels all to 0)
- **N** → total number of pixels
- **L-1** → max intensity (255 for 8-bit)

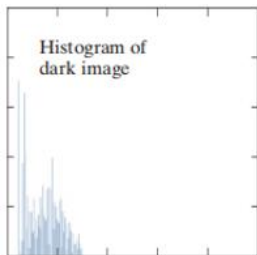
CDF in equalization works like a "percentile mapping": each pixel is assigned a new intensity based on how many pixels are darker than it, then stretched to 0–255 range.

After full equalization, histogram looks nice (flat).

Example:

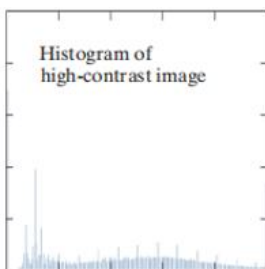
Suppose we have a very dark grayscale image where most pixels are between **50–100** intensity values.

The histogram is clustered in that narrow range.



After histogram equalization, the pixel values are spread across **0–255**.

As a result, dark regions become brighter, and contrast is enhanced.



Drawback: If there's noise, equalization makes it worse (more obvious).

2B) Explain separable filtering with a formula.

Ans:

In image processing, a filter (or kernel) is called **separable** if a 2D filter operation can be broken down into **two 1D filter operations** – one along rows and the other along columns.

This reduces computation, because instead of multiplying with every element in a 2D filter, we perform two 1D operations sequentially.

Formula:

For a 2D filter $H(x,y)$, if it can be written as:

$$H(x,y)=h_1(x).h_2(y)$$

then the filtering operation

$$g(x,y)=f(x,y) * H(x,y)$$

can be computed as:

$$g(x,y)=(f(x,y)* h_1(x))* h_2(y)$$

where

$f(x,y)$ = input image

$g(x,y)$ = filtered output

$h_1(x)$ = 1D filter along rows

$h_2(y)$ = 1D filter along columns

Example:

A Gaussian filter is separable.

A 2D Gaussian kernel can be expressed as the product of two 1D Gaussians:

$$G(x,y)=G(x) \cdot G(y)$$

A 2D Gaussian kernel (3×3) looks like:

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

This is **separable** because it can be written as:

$$\frac{1}{16} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot [1 \quad 2 \quad 1]$$

Instead of applying a 5×5 Gaussian filter (25 operations per pixel), we can apply two 1D filters of size 5 (5 + 5 = 10 operations per pixel).

Not every 2d filter is seperable, to check if a 2d filter is seperable or not , we take its single value decompostion (SVD), $\mathbf{K} = \mathbf{\Sigma} \mathbf{\sigma_i} \mathbf{u_i} \mathbf{v_i^T}$.

If only the first singular value σ_0 is non-zero, the kernel is separable and $\sqrt{\sigma_0} \mathbf{u_0}$ and $\sqrt{\sigma_0} \mathbf{v_0^T}$ provide the vertical and horizontal kernels.

Advantage:

Greatly reduces computation time and memory use.

3A) What is non-linear filtering? Give examples.

Ans:

Non-linear filtering is an image filtering technique in which the output pixel is not obtained by a simple weighted sum of input pixels (unlike linear filters). Instead, it is computed using a non-linear combination of neighboring pixel values.

This makes it especially useful for **removing noise** and **preserving edges** in images.

Linear filters (like Gaussian blur) often blur both noise and edges, while non-linear filters can suppress noise without softening important details.

Examples:

Median Filter

Replaces each pixel with the median value of its neighborhood.

Very effective for removing salt-and-pepper noise (shot noise), where some pixels are much brighter or darker than their neighbors.

Example: If a 3×3 neighborhood has values $\{10, 12, 11, 200, 13, 11, 12, 14, 10\}$, the median = 12, so the noisy pixel (200) is removed.

α -Trimmed Mean Filter

Removes a fraction (α) of the smallest and largest values in the neighborhood and averages the rest.

Works better than the median filter for Gaussian noise.

Weighted Median Filter

Each neighboring pixel is given a weight depending on its distance from the center.

This helps preserve details while reducing noise.

Bilateral Filter

Considers both **spatial distance** and **intensity difference** to smooth images while keeping edges sharp.

Key Property:

Edge-preserving: Unlike linear filters (e.g., Gaussian blur), non-linear filters reduce noise **without strongly blurring edges**.

3B) Explain bilateral filtering and its applications.

Ans:

Bilateral filtering is a non-linear filter used to smooth an image while preserving edges.

- Unlike normal Gaussian blur (which considers only the **spatial closeness** of pixels), bilateral filtering also considers **intensity similarity**.
- This means that pixels that are **close in position** *and* **similar in intensity** have higher influence, while pixels that are very different (possible noise) get down-weighted.
- Median / α -trimmed mean filters reject extreme outliers (bad noisy pixels) completely.
- Bilateral filtering gives less weight (not full rejection, but soft rejection) to pixels that are too different in intensity from the center pixel.

$$g(i, j) = \frac{\sum_{k,l} f(k, l)w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}.$$

$g(i, j)$: output pixel at position (i, j).

$f(k, l)$: input pixel at neighbor (k, l).

$w(i, j, k, l)$: weight of that neighbor.

Formula = weighted average (like Gaussian), **but weights depend on both distance and intensity similarity**.

Two parts decide the weight:

Domain kernel (d) → how far is the neighbor in space? (closer = more weight, like Gaussian blur).

$$d(i, j, k, l) = \exp \left(- \frac{(i - k)^2 + (j - l)^2}{2\sigma_d^2} \right)$$

Range kernel (r) → how similar is the intensity to the center pixel? (more similar = more weight).

$$r(i, j, k, l) = \exp \left(- \frac{(f(i, j) - f(k, l))^2}{2\sigma_r^2} \right)$$

Final weight = (close in space) × (similar in intensity).

Only nearby AND similar pixels strongly influence the result.

$$w(i, j, k, l) = d(i, j, k, l) \cdot r(i, j, k, l)$$

Applications of Bilateral Filtering:

- **Noise reduction while keeping edges sharp** → medical imaging, photo denoising.
- **Edge-preserving smoothing** → keeps object boundaries while removing fine textures.
- **Cartoon and stylized image effect** → smooths flat regions while keeping edges.
- **Pre-processing for segmentation** → enhances edges for better segmentation.
- **High dynamic range (HDR) imaging** → tone mapping while maintaining contrast.

4. Explain applications of Fourier Transform in image sharpening, blurring, and noise removal.

Ans:

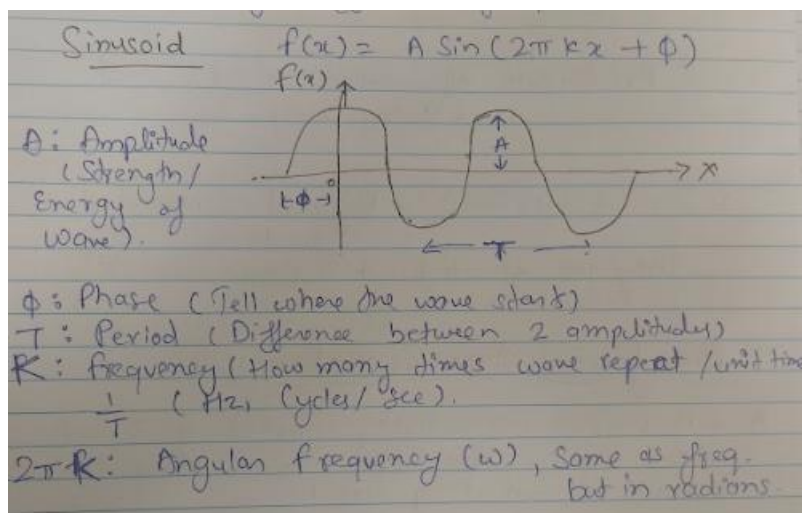
Fourier Transform in Image Processing

Definition:

The Fourier Transform (FT) is a mathematical tool that transforms a signal or image from the **spatial domain** (pixel values) into the **frequency domain** (sinusoidal components).

The **spatial domain** → represents pixel intensities.

The **frequency domain** → represents how quickly pixel values change.



Low frequency = smooth variations (sky, background).

High frequency = sharp changes (edges, noise, fine textures).

By manipulating frequency components, we can perform operations like blurring, sharpening, and noise removal.

2D Fourier Transform Equation:

For a 2D image $f(x, y)$,

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

Inverse Fourier Transform:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

where:

- $(x, y) \rightarrow$ spatial coordinates
- $(u, v) \rightarrow$ frequency coordinates
- $j = \sqrt{-1}$

Applications in Image Processing

Image Blurring (Low-Pass Filtering):

Blurring removes details and smooths the image.

In frequency domain $\rightarrow$ keep **low frequencies** (background) and suppress **high frequencies** (edges, details).

Example: Gaussian Low-Pass Filter (GLPF).

Image Sharpening (High-Pass Filtering):

Sharpening enhances edges and fine details.

In frequency domain $\rightarrow$ keep **high frequencies** (edges) and reduce **low frequencies** (smooth areas).

Example: Laplacian High-Pass Filter.

Noise Removal (Band-Pass / Band-Stop Filtering):

Noise often appears as specific high-frequency components.

Using frequency filters, we can selectively remove noise frequencies while preserving useful image information.

Example: Removing periodic noise using a notch filter in frequency domain.

5A) What are wavelets? Explain their applications in image analysis.

Ans:

Wavelets

A wavelet is a mathematical function that represents data or signals in terms of both time (or space) and frequency.

Unlike the Fourier Transform, which only gives frequency information, the Wavelet Transform provides multi-resolution analysis – meaning it shows which frequencies exist and where they occur in the image.

In images, wavelets allow us to analyze details at different scales:

Approximation (low-frequency) → smooth background.

Details (high-frequency) → edges, fine textures.

For a signal $f(t)$, the continuous wavelet transform (CWT) is:

$$W(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt$$

where:

- $\psi(t)$ = mother wavelet
- a = scale (controls frequency)
- b = translation (controls position)

In images, wavelet transform is applied in 2D (rows & columns) giving:

- LL = approximation (smooth)
- LH, HL, HH = horizontal, vertical, diagonal details

Applications in Image Analysis

Image Compression

JPEG2000 uses wavelets for efficient compression.

Wavelets preserve important features while reducing file size.

Image Denoising

By thresholding high-frequency wavelet coefficients, random noise can be removed while preserving edges.

Edge Detection and Feature Extraction

High-frequency wavelet components highlight edges and fine details.

Useful in medical imaging, fingerprint recognition, and texture analysis.

Image Fusion

Combine features from multiple images (e.g., satellite + infrared) using wavelet decomposition.

Medical Image Analysis

Used in tumor detection, brain scan enhancement, and ECG/EEG signal analysis.

Multiresolution Analysis

Images can be analyzed at different scales (zoom levels), useful in computer vision tasks.

5B) Differentiate between interpolation and decimation in pyramids.

Ans:

Image Pyramids

An image pyramid is a multi-resolution representation of an image, where we repeatedly scale the image up or down.

Difference Between Interpolation and Decimation

Decimation

- 1.Process of reducing the image resolution by removing pixels.
- 2.Achieved by low-pass filtering (to avoid aliasing) followed by subsampling.
- 3.Used in **Gaussian pyramids** when going from a higher level to a lower level.

Example: $512 \times 512 \rightarrow 256 \times 256$.

4.Loss of details, reduced resolution

5.Decimation = shrink the image (downsampling).

Interpolation

- 1.Process of increasing the image resolution by adding new pixels.
- 2.Achieved by inserting pixels and estimating their values using interpolation methods (nearest neighbor, bilinear, bicubic).
- 3.Used in **Laplacian pyramids** to reconstruct higher resolution images from lower ones.

Example: $256 \times 256 \rightarrow 512 \times 512$.

4.Blurred/approximate image, estimated details

5.Interpolation = enlarge the image (upsampling).

6. Explain parametric geometric transformations and feature-based morphing with examples.

Ans:

Parametric Geometric Transformations

Parametric geometric transformations are mathematical mappings that move an image from one coordinate system to another by applying a set of parameters. These parameters control the position, orientation, size, and shape of the image.

They are called parametric because the transformation is fully described by a finite set of parameters (e.g., angle of rotation, scaling factor, translation distance).

Types and Equations

Translation

Shifts the image by moving every pixel by a fixed offset.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix}$$

Example: Moving a satellite image overlay to align with GPS coordinates.

Scaling

$$x' = s_x \cdot x, \quad y' = s_y \cdot y$$

Example: Zooming into a medical scan or enlarging a photo.

Rotation

Rotates the image around a chosen point (usually the origin).

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

Example: Rotating an image taken at a tilted angle.

Shearing

Shifts image rows or columns, creating a slanting effect.

$$x' = x + k_x y, \quad y' = y + k_y x$$

Example: Correcting perspective distortions in scanned documents.

Affine Transformation

Combination of scaling, rotation, translation, and shear.

Preserves parallel lines but not necessarily angles.

Example: Aligning aerial photos from different viewpoints.

Projective (Homography) Transformation

General case that can handle perspective changes.

Used in panorama stitching or projecting 3D scenes onto 2D images.

Applications of Parametric Transformations:

Aligning medical scans taken at different times.

Correcting geometric distortions in satellite images.

Augmented reality (placing virtual objects in real-world coordinates).

Image registration in computer vision.

2. Feature-Based Morphing

Feature-based morphing is an image transformation technique where one image gradually changes into another by aligning and blending corresponding features (e.g., facial landmarks, object edges).

Unlike global transformations, it works locally on features and provides smooth transitions between images.

Steps in Feature-Based Morphing

Feature Identification

Select key points (landmarks) in both images.

Example: Eyes, nose, lips in faces; corners in objects.

Correspondence Mapping

Establish pairs of matching points.

Often done using triangulation (Delaunay triangulation).

Warping

Each triangle/region is warped from the source to match the target shape.

Blending (Cross-Dissolve)

Gradual mixing of pixel intensities from source to target image.

Produces intermediate frames.

Examples of Feature-Based Morphing

Face Morphing (Movies/Apps):

Transforming one actor's face into another smoothly (e.g., Harry Potter transformations, Terminator 2 liquid metal morphing).

Medical Imaging:

Morphing between a healthy and a diseased organ to study disease progression.

Cartoon/Animation:

Turning one object into another in animated sequences.

Image Fusion:

Combining multiple facial images for biometric recognition.

