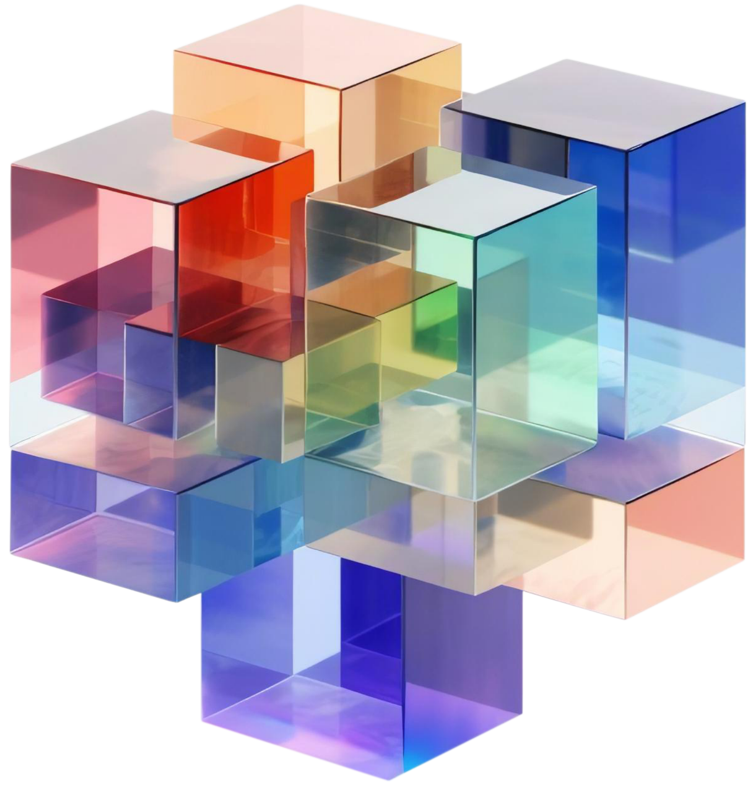


From Pixels to Patterns

Image Processing Techniques for the Real World


Nagasai Vegur

Introduction

Introduction: Image

Image: It is a function $f : X \times Y \rightarrow Z$ (f is a 2D function)

Pixel: $((x, y), z)$ where (x, y) is pixel location and z is pixel value, $(x, y) \in X \times Y$ and $z \in Z$

Types of Image:

- ***Analog Image***: at least one of x or y is non-discrete
  - Film based optical image, film based x-ray image
- ***Discrete Image***: both x and y are discrete
  - CCD Array
- ***Digital Image***: all x, y and z are discrete
  - JPEG, PNG etc..

Introduction: Type of Image Processing

- Low level image processing
  - Image enhancement: Noise removal, smoothening, sharpening, image reconstruction etc..
  - Image compression
  - Image security: Water marking, steganography
- Middle level image processing
  - Edge detection
  - Segmentation
  - Feature engineering
- High level image processing
  - Object representation
  - Object recognition/classification

Introduction: More on Image

- Image display : CRT, CCD, LED, OLED
- Images in memory

$f = (M_R, M_G, M_B)$ where f is sequence of matrices and each M is a tradition image I we discussed earlier

- Types of images based on intensity
  - Color Image, Gray Scale Image, Binary Image (Old School newspapers)
- Color to gray scale
  - $I(x, y) = \frac{1}{3}(r(x, y) + g(x, y) + b(x, y))$
  - $I(x, y) = \frac{1}{2}g(x, y) + \frac{1}{4}[r(x, y) + b(x, y)]$

Basic operations on Images

Basic Operations on Images

- Bilinear interpolation
- Arithmetic Operations
  - Image sum, difference, multiplication and division
- Unary Operations
  - Image negation or complementation
- Set theoretic Operations
  - Union, intersection and complementation
- Geometric Transformations
  - Scaling, translation, rotation and shearing

Spatial domain operations on Images

Spatial domain operations on Images

- Point processing
  - Identity transformation
  - Image negation
  - Thresholding
  - Contrast enhancement
  - Contrast stretching
  - Bit plane slicing
  - Logarithmic transformation
  - Power law/gamma transformation
  - Histogram equalization transformation
- Neighborhood processing
  - Average filter, Median filter, Template matching, Convolution/Correlation
  - Filter design: Average filter, Gaussian filter, Derivative filter, Prewitt filter, Sobel filter, Median filter, Gradient filter, Laplace filter, Image sharpening

Frequency domain operations on Images

Frequency domain operations on Images

- Fourier transform
- Fast Fourier transformation
- Convolution theorem
- Ideal frequency domain filters

Edge detection

Edge Detection: Overview

Definition

- Edge detection identifies sharp changes in intensity to locate boundaries of objects in an image.

Purpose

- Simplifies image data for analysis.
- Helps detect object structure, shape, and boundaries.

Steps

1. **Noise Reduction:** Smooth the image (e.g., Gaussian filter).
2. **Gradient Calculation:** Measure intensity changes in all directions.
3. **Edge Localization:** Highlight areas with significant gradients.

Edge Detection: Canny filter

Steps

1. Apply Gaussian Blur to reduce noise.
2. Compute intensity gradient using Sobel operators.
3. Apply Non-Maximum Suppression to refine edges.
4. Use Double Threshold to detect strong and weak edges.
5. Perform Edge Tracking by Hysteresis to connect edges.

Advantages: Precise and detects edges in noisy images.

Edge Detection: Real world applications

Medical Imaging: Highlight organs/tumors in X-rays or MRIs.

Computer Vision: Object detection and facial recognition.

Robotics: Path planning and obstacle detection.

Industrial Automation: Quality control via object boundary analysis.

Satellite Imaging: Land-use mapping and terrain analysis.

Segmentation

Segmentation: Overview

Definition

- Segmentation is the process of dividing an image into distinct regions or objects for easier analysis.

Purpose

- Identify meaningful structures (e.g., objects, boundaries).
- Simplify image representation for further processing.

Types of Segmentation

1. **Thresholding:** Based on intensity values.
2. **Edge-based:** Detects discontinuities.
3. **Region-based:** Groups pixels with similar properties.

Segmentation: Watershed Algorithm

Concept

- Treats pixel intensities as topographical elevation.
- Finds "catchment basins" and "watershed lines" to separate objects.

Steps

1. Compute gradient of the image to highlight edges.
2. Mark initial seeds (local minima in intensity).
3. Flood regions starting from seeds, filling basins until boundaries meet.

Applications:

- Separates overlapping objects (e.g., cells in microscopy).

Segmentation: Other Segmentation Methods

K-Means Clustering

- Groups pixels into k clusters based on intensity.
- Simple and widely used for color segmentation.

Region Growing

- Starts from seed points and expands based on similarity criteria.
- Useful for smooth and connected regions.

Graph-Based Segmentation

- Represents image as a graph; partitions nodes (pixels) based on edges (similarity).

Deep Learning Segmentation

- Neural networks (e.g., U-Net, Mask R-CNN) for highly accurate segmentation in medical and industrial use cases.

Segmentation: Real world applications

Medical Imaging:

- Tumor detection, organ segmentation in MRIs/CT scans.

Autonomous Vehicles:

- Lane detection, object separation.

Satellite Imaging:

- Land-use classification, agricultural monitoring.

Document Analysis:

- Segmentation of text, tables, and diagrams.

Augmented Reality:

- Object and environment separation for realistic overlays.

Object detection

Object detection: Overview

Definition

- The process of identifying and localizing objects in an image or video.

Purpose

- Detects what objects are present and their positions in the image.

Steps in Object Detection

1. **Feature Extraction:** Identify key points and descriptors.
2. **Feature Matching:** Compare features to known objects.
3. **Localization:** Draw bounding boxes or regions around detected objects.

SIFT (Scale-Invariant Feature Transform)

Overview

- Detects and describes local features in images, invariant to scaling and rotation.

Steps

1. **Scale-space Construction:** Apply Gaussian filters at different scales.
2. **Keypoint Detection:** Identify extrema in Difference of Gaussians (DoG).
3. **Keypoint Orientation:** Assign orientations to ensure rotation invariance.
4. **Descriptor Generation:** Create 128-dimensional descriptors around keypoints.

Advantages:

- Highly accurate for complex objects.
- Robust to rotation, scaling, and partial occlusion.

SIFT: Scale-Space Construction

Purpose

- To detect features invariant to scale changes.

Key Concept

- A scale-space is created by convolving the image with Gaussian functions at different scales.

Equation for Scale-Space Representation:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Where

- $L(x, y, \sigma)$: Scale-Space representation
- $G(x, y, \sigma)$: Gaussion function
- $I(x, y)$: Input image
- σ:Scale (Standard deviation of the Gaussian)

Different of Gaussian

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Where k is a constant scale factor

SIFT: Keypoint Detection

Purpose:
Identify extrema in the scale-space by comparing each pixel with its neighbors in both spatial and scale dimensions.

Key Steps:

- For each pixel in the image: Compare its intensity with 26 neighbors (8 in the current scale, 9 each in adjacent scales).
- Identify maxima and minima in $D(x, y, \sigma)$

Mathematical Representation:

The extrema occur where the gradient is zero:

$$\frac{\partial D(x, y, \sigma)}{\partial x} = 0 \quad \text{and} \quad \frac{\partial D(x, y, \sigma)}{\partial y} = 0$$

Keypoint Refinement:

A quadratic approximation of the scale-space is used to refine keypoint locations:

$$\hat{x} = -\mathbf{H}^{-1}\frac{\partial D}{\partial x}$$

Where:

- H : Hessian matrix of $D(x, y, \sigma)$
- $\frac{\partial D}{\partial x}$: Gradient vector

SIFT: Orientation Assignment and Descriptor Generation

- Orientation Assignment: Ensures rotation invariance by assigning an orientation to each keypoint based on the local image gradient
- Gradient Calculation

$$m(x, y) = \sqrt{\left(\frac{\partial L}{\partial x}\right)^2 + \left(\frac{\partial L}{\partial y}\right)^2}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{\partial L/\partial y}{\partial L/\partial x}\right)$$

- Description Generation
    1. Divide the region around the keypoint into 4x4 subregions.
    2. Compute a histogram of gradient orientations (8 bins) for each subregion.
    3. Form a 128-dimensional feature vector: 4x4x8=128
- Normalization:

  Normalize the descriptor to make it robust to illumination changes:

$$\mathbf{v} \leftarrow \frac{\mathbf{v}}{||\mathbf{v}||}$$

SURF (Speeded-Up Robust Features)

Overview

- An improvement over SIFT for faster computation.
- Uses integral images and approximations of Gaussian derivatives for efficiency.

Steps

1. **Keypoint Detection:** Detect points using Hessian matrix approximation.
2. **Descriptor Generation:** Extract descriptors around keypoints.
3. **Feature Matching:** Match descriptors between images.

Advantages:

- Fast and scale-invariant.
- Robust to rotation and illumination changes.

SURF: Hessian Matrix Approximation

- Hession matrix approximation

$$H(x, y) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}$$

- Keypoints are identified as maxima or minima of the determinant of $H(x, y)$:

$$\text{det}(H) = L_{xx}L_{yy} - (L_{xy})^2$$

- Efficient computation using box filters:
  - Box filters approximate Gaussian derivatives: L_{xx}, L_{xy}, L_{yy}
  - They reduce computational complexity and allow parallel processing.
  - Integral images enable rapid computation of box filter responses at any scale.
- Integral Images:
  - Each pixel in the integral image represents the sum of all pixels above and to the left of it in the original image.
  - This allows summing over rectangular regions in constant time.

SURF: Description Generation and Feature Matching

- Divide the region around the keypoint into subregions.
- Compute Haar wavelet responses indirections.
- Form descriptors by summing weighted responses in subregions.

After descriptor generation, features from different images are matched based on the similarity of their descriptors.

Matching Criteria:

- Typically, the **Euclidean distance** between descriptors is used as a similarity measure.
- The nearest neighbor approach matches each keypoint to its closest counterpart.

Ratio Test:

- A second nearest neighbor is used to verify that the first match is significantly better than the second.
- This helps avoid incorrect matches.

SIFT vs SURF

Feature	SIFT	SURF
Speed	Slowed DoG computation	Faster using integral images
Accuracy	High for complex structures	Slightly less accurate
Robustness	Strong against noise/illumination	Strong but less robust than SIFT
Applications	Detailed object recognition	Real-time applications

Object detection: other methods

HOG (Histogram of Oriented Gradients):

- Computes gradient histograms for object detection (e.g., pedestrians).

YOLO (You Only Look Once):

- Deep learning-based; real-time, bounding-box predictions.

Faster R-CNN:

- Uses region proposal networks (RPNs) for object localization and classification.

Object detection: Real world applications

Autonomous Vehicles:

- Pedestrian and obstacle detection.

Security Systems:

- Face detection and recognition in surveillance.

Retail Analytics:

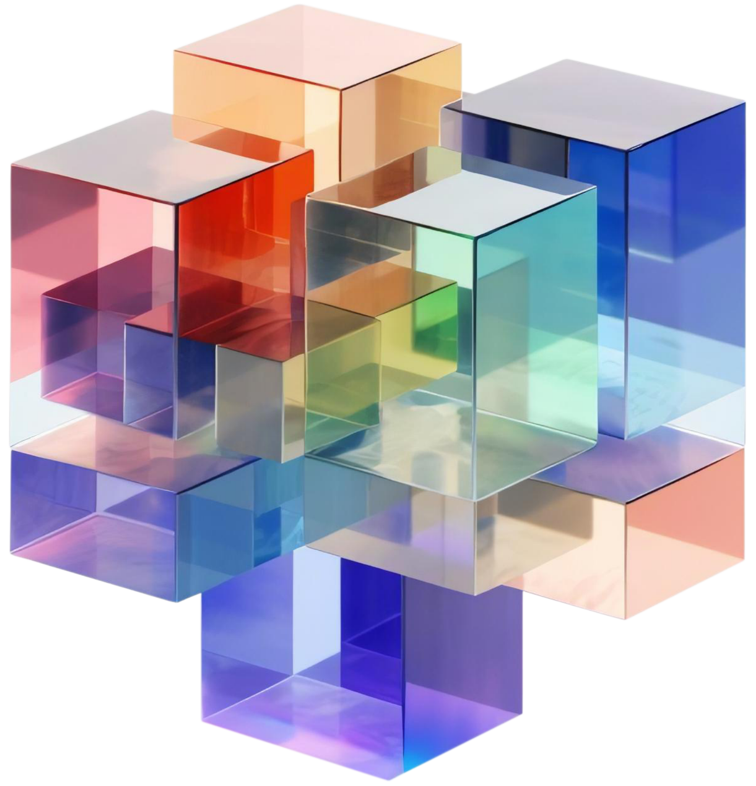
- Customer tracking and product interaction analysis.

Healthcare:

- Detection of anomalies in medical imaging.

Industrial Automation:

- Object detection for quality assurance.



Thank you

Nagasai Vegur