Exercise 03 for MA-INF 2201 Computer Vision WS24/25
27.10.2024
Submission on 3.11.2024

Overview

This assignment consists of implementing three fundamental image processing algorithms:

1. Binary Distance Transform: Computing distance maps from binary images
2. Hough Transform: Line detection in edge images
3. Mean Shift: Color-based image segmentation

Submission Requirements

1. Your implemented functions with comments
2. Visualization files for all results (including PDF/PNG files of plots and visualizations)
3. Analysis for tasks 2 and 3 regarding impact of variation in the respective parameters

Code Structure

Template solutions have been provided in the following Python files:

1. `distance_transform.py`: Implementation of binary distance transform
2. `hough_transform.py`: Implementation of Hough transform for line detection
3. `mean_shift.py`: Implementation of mean shift segmentation

Students only need to implement the respective functions marked with TODO comments. The template code includes automatic result generation and visualization functionality.

1. **Task 1: Distance Transform** *(5 Points)*

   Implement a binary distance transform algorithm that computes the distance from each pixel to the nearest background pixel in a binary image.

   Mathematical Background

   For a binary image $I(x, y)$, the distance transform $D(x, y)$ is defined as:

   $$D(x, y) = \min_{(i,j) \in B} \sqrt{(x - i)^2 + (y - j)^2}$$

   where B is the set of all background pixels.

   Implementation should use a two-pass algorithm:

   (a) Forward pass (top-left to bottom-right):

   $$D_1(x, y) = \min\{D(x + i, y + j) + \sqrt{i^2 + j^2}\}$$

   where $(i, j) \in \{(-1, -1), (-1, 0), (-1, 1), (0, -1)\}$

   (b) Backward pass (bottom-right to top-left):

   $$D_2(x, y) = \min\{D_1(x, y), \min_{(i,j)}\{D_1(x + i, y + j) + \sqrt{i^2 + j^2}\}\}$$

   where $(i, j) \in \{(1, -1), (1, 0), (1, 1), (0, 1)\}$

   Input

   - A 64x64 binary image containing a simple shape (square, circle, or triangle)
   - Background pixels are 0, foreground pixels are 1
   - Three sample images are provided for testing

   Evaluation Basis

   - Correctness of implementation
   - Magnitude of Error

2. **Task 2: Hough Transform for Line Detection** *(8 Points)*

   Mathematical Background

   (a) Line parameterization in normal form:

   $$\rho = x\cos\theta + y\sin\theta$$

   where ρ is the perpendicular distance from origin to the line, and θ is the angle from x-axis.

   (b) Accumulator array $A(\rho, \theta)$ voting:

   $$A(\rho, \theta) = \sum_{(x,y)\in E} \delta(\rho - x\cos\theta - y\sin\theta)$$

   where E is the set of edge pixels and δ is the Dirac delta function.

   (c) Peak detection in accumulator space:

   $$(\rho_i, \theta_i) = \arg\max_{\rho,\theta} A(\rho, \theta)$$

   subject to local maxima conditions and minimum threshold.

   (d) Line reconstruction: For each peak (ρ_i, θ_i), points (x, y) on the line satisfy:

   $$\rho_i = x\cos\theta_i + y\sin\theta_i$$

   Input

   - A 64x64 binary edge image containing 2-4 straight lines
   - Background pixels are 0, edge pixels are 1
   - Four sample images are provided for testing

   Evaluation Basis

   - Quality of line detection (based on visualization)
   - Correctness of accumulator array and peak detection method implementation
   - Analysis on the impact of the *n_peaks* parameter

3. **Task 3: Mean Shift Segmentation** *(7 Points)*

Mathematical Background

Mean shift vector for a point $\mathbf{x}$:

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_i K(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2)\mathbf{x}_i}{\sum_i K(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2)} - \mathbf{x}$$

where K is the Gaussian kernel and h is the bandwidth parameter.

Input

- A 64x64 RGB image containing 3-4 distinct colored regions
- Values normalized to [0,1]
- Three sample images are provided for testing

Evaluation Basis

- Quality of segmentation results (based on visualization)
- Correctness of mean shift implementation
- Analysis on the effect of the *bandwidth* parameter