

Assignment 2 Writeup

Ian Chen

September 19, 2023

1 Short answer problems

1. *Compare the effects of 1) Dilation + Erosion against 2) Erosion + Dilation. Do they have the same effects? Why?*

They have different effects. Erosion then dilation, also called opening, is used to remove small objects while preserving original shape. This effect removes small regions-like objects like lines or noise, while preserving the larger areas. Dilation then erosion, also called closing, fills holes in regions while preserving region sizes. This effect fills in gaps between objects or within the object itself, while preserving large holes and regions.

2. *List two examples of regular texture and two examples of near-regular texture.*

Regular- Brick Wall and Checkboard. Both are exact patterns and identical shapes with no randomness.

Irregular- Stone Floor Tiles and Wood Planks. Both are identical from far away, but when closely examined, have small details unique to each piece.

3. *What are the cases where optical flow is not well-defined? Please give two concrete examples.*

One case is the aperture problem, where the object's motion is not aligned with the direction calculated by the optical flow gradient, such as a spinning barber pole, with the pixels seem to move up constantly, while in reality it's moving either clockwise or counter-clockwise. The occlusion of the object's motion causes a false interpretation of its motion.

Another case is the inconsistent brightness, where the object's brightness changes over time, such as a light source being moved around. The optical flow gradient will be inconsistent and not well-defined. Another case is texture-less regions, where the object has no texture, such as a white wall. The optical flow gradient will not sense motion, since all pixels have the same brightness.

Another case is motion too fast for the frame rate, where the object moves too fast for the camera to capture smooth differences between each frame. This causes the optical flow gradient to be unable to track motion throughout the frames.

4. *What are the advantages of RANSAC when compared with Hough Transform?*

RANSAC is more robust to outliers than Hough Transform. RANSAC is able to detect and identify more types of shapes than Hough, so it's more applicable to a wider variety of images. RANSAC is also more efficient than Hough Transform, since only a subset of the total points are needed to be sampled to have a confident result that's representative of the entire image.

2 Circle Detection

- *Explain your implementation in concise steps (English, not code).*
- *Demonstrate the functions applied to the provided images 'coins.jpg' and 'planets.jpg' and one image of your choosing. Display the images with detected circle(s), labeling the figure with the radius. Note: you only need to select one reasonable radius and display all detected circles (i.e., those with highest votes) under that radius. You are not required to consider circles with a center off the image.¹*
- *For Hough Transform, explain how your implementation post-processes the accumulator array to determine automatically how many circles are present.*

- For RANSAC, explain how you implement circle fitting.
- For one of the images, display and briefly comment on the Hough space accumulator array.
- For one of the images, demonstrate and explain the impact of the vote space quantization (binsize). In other words, alter the bin size and compare and contrast with a brief explanation why what happened makes sense.
- For one of the images, plot the progress of the RANSAC as the number of tries increase. The x axis of the plot should be the number of tries, and the y axis should be the number of inliers that the best model produces.

3 Image segmentation with k-means

1. Given an $h \times w \times 3$ matrix 'Im', where h and w are the height and width of the image, apply k-means clustering to associate pixels with clusters. Return 'labelIm', an $h \times w$ matrix of integers indicating the cluster membership (e.g., from 1 to k) for each pixel. Please use the following form:

```
function [labelIm] = clusterPixels(Im, k)
```

2. Detect cluster boundary pixels from 'labelIm'.

```
function [boundaryIm] = boundaryPixels(labelIm)
```

3. Please test both functions on the provided images 'gumballs.jpg', 'snake.jpg', and 'twins.jpg' and one other image of your choosing, and then displays the results.