Readings

- Klette 3.4 — Hough Transform for Lines & Circles
- Szeliski 4.3.2 — Hough transform
- https://youtu.be/_mGxmZWs9Zw — Video on Generalized Hough Transform

Summary

Motivation

- Many objects are characterized by straight lines – how to find straight lines in an image?
- Edge detectors e.g. Canny gives us *all* edge points but we still don't know which points belong to which line and what the line is
- Hough transform, a voting-based approach can solve this problem

Mathematics of Lines

- Different parameterizations of line equations: slope-intercept form, double-intercept form, normal form; all can be derived from one another
- Resulting fit line depends on the error or objective function being optimized
- Voting is a robust alternative to optimization to handle sensitivity to outliers

Hough Transform for Lines & Circles

- Correspondence between image space vs. parameter space; a point becomes a line, a line becomes a point
- Discretize parameter space into accumulator array, accumulate votes from all points in the image space, local maximum indicates the presence of a line;
- Parameterize lines in normal form instead of slope-intercept for to limit range of parameters

Hough Transform for Circles

- Detecting circles follows same approach as straight lines but using circle equation
- Can limit the parameter space by assuming known radius
- Limit number of votes cast (simplifies max-finding) by leveraging gradient orientation

Generalized Hough Transform

- Detect arbitrary shapes without close-form equations
- Build a model by storing boundary points based on their gradient orientation associated with an displacement vector to a reference point
- For edge points in test image, look up in the table closest gradient orientation and displacement vectors
- Generalize concept of edge points to arbitrary image evidence, e.g. image patches represented by visual codewords for doing object detection, action recognition, etc.