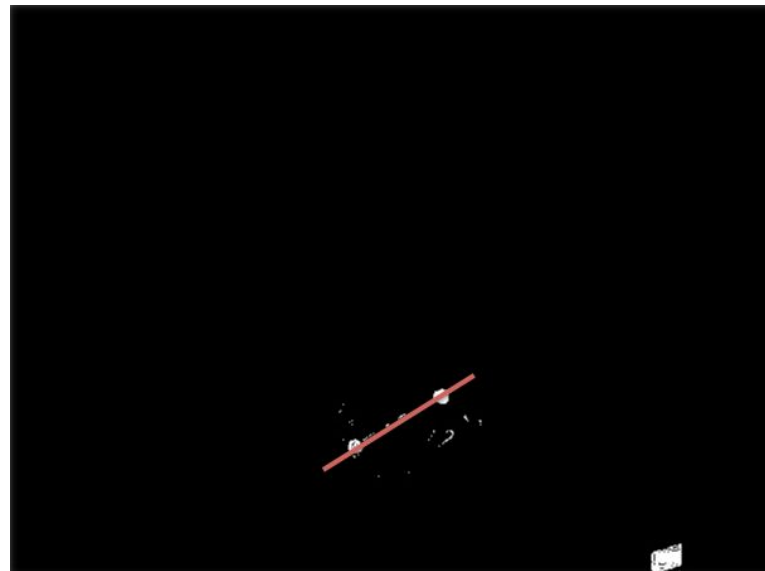




C++ for Computer Vision

OpenCV - Dag 3

Thresholding



Liste over filtre

- Gaussian blur

<https://www.geeksforgeeks.org/opencv-c-program-to-blur-an-image/>

- Box filter
- Median blur
- Normalized block filter
- Bilateral filter

https://docs.opencv.org/4.6.0/dc/dd3/tutorial_gaussian_median_blur_bilateral_filter.html

- Otsu anvendes til automatisk at finde en threshold-værdi samt til sidst at anvende tærskelværdien som et slags baggrundsfiler

Godt eksempel til at anvende Otsu:

<https://learnopencv.com/otsu-thresholding-with-opencv/>

Bonusfilter

Edgepreserving filter fjerner støj, men forsøger at bibeholde objekter der er gode kandidater til kanter



noisy image



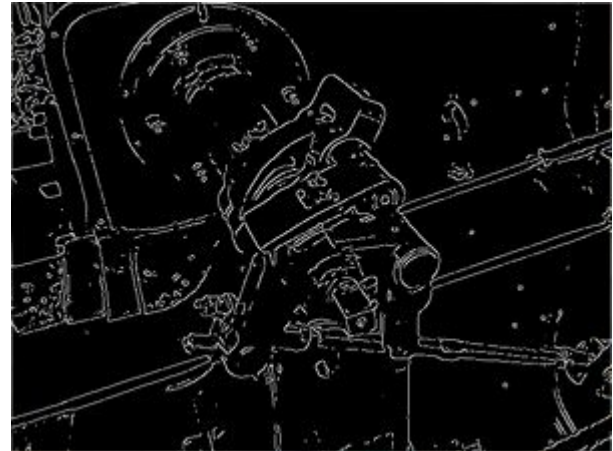
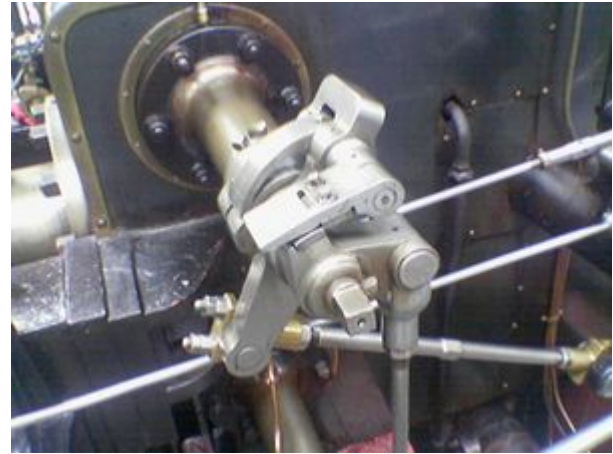
naïve denoising
Gaussian blur



better denoising
edge-preserving filter

Edge detection

- Præcise og/eller generaliserende edge detection kræver oftest filtrering i flere lag
- Edge detection er meget sensitiv overfor støj
- Lettere at gøre på gråtonebilleder
- Ofte udført før vi detekterer konturer
- Canny edge detection er typisk den bedste baseline



Pipelines

Detektion er ofte ikke succesfuldt uden at processere billedet først. Vi kalder typisk en række af processering for en pipeline

Typisk pipeline

- Grayscale konvertering
- Filtre
- Edge detection
 - Hvis det fejler stadig, prøv thresholding
- Dilation til at gøre kanter tykkere
- Evt. erosion til at forbinde kanter

Derefter er billedet normalt klar til at finde konturer og dermed objekter i billedet.

Forskellige blandinger af ovenstående trin kan have varierende effekt, så eksperimenter!