Forelesning 3 Edge detection

* Man kan gjette noe om hvordan formen er kun ut ifra kantene
* Kanter virker fundamentale til menneskelig oppfattning
  + Det er en komprimert versjon av bildet
* Nyttig for kart
* Edges:
  + Grensen mellom «bland»-e regioner av et bilde
  + Regioner: Homogene områder mellom kanter
  + Det er en dualitet mellom kanter og regioner
* Diskontinuiteter
  + Mange ting kan skape disse kantene
    - Depth discontinuity
    - Endring i orienteringen til en overflate (Topp og side av en boks)
    - Skygger
    - Reflective discontinuity.
      * Egenskaper til overflaten, markeringer, teksturer…
  + Cotrast: Grey level er forskjellig på begge sider.
* Edge profiles
  + Step edge: Den idealiserte vi vil ha
    - Fra verdi a til verdi b
  + Line edges a->b->a
  + Roof edge «Går lineært opp» og så «Lineært ned»
* Grey level: Gjennomsnittet av R,G og B-bildet
* Virkelige signaler er veldig støyete
  + Det er ikke faktiske sprang.
    - VI ser heller etter «Gradienter» «slopes» mellom verdiene
  + Illusionary contours:
    - Vi trenger bare å være klar over at det finnes, men vi kommer ikke til å gå over det
* Man vil aldri se faktiske sprang, fordi alle linser gir uklarhet i bildet
  + Convolve with the Gaussian
  + Uklarheten er bra for oss, siden den lar oss se på de deriverte
* 2ND derivative zero crossing
* Kantene står ortagonalt med gradientene
* Edge Descriptions
  + Edge normals (+3 andre jeg gikk glipp av)
* De stedene vi ser kanter er der gradienten er størst
* Vi har lyst til å tilpasse alt med en glatt overflate
* Gradient images
  + Gjør det litt lettere å se endringene
  + Man kan dermed se på de x-derivere og de y-deriverte, eller normen
  + Normen slipper problemet med at dx og dy er avhengig av orientasjonen til kameraet
* Edge operators
  + Difference operaors
  + Convolution operators
* Raster format: Når man representerer et bilde som en 1D-array
* Derivasjon har en lei tendens til å øke støy
  + Om man ser på det i Fourier-domenet ganges det med , slik at man legger mer vekt på høye frekvenser
  + Skal vi løse det ved å gange DFT-en med noe som fjerner/demper de høye fekvensene, og går tilbake igjen
    - Alt dette er en veeeeldig vanskelig måte å si at vi tar og jjører det gjennom et lavpassfilter
  + Convolving with a low-pass filter
    - Vi snakker om faktisk konvolusjon.
  + Gaussian g, og f er signalet. Vi tar og deriverer dette
* Convolution in C
* Noise in 2D
  + Smoothingen hjelper mot hvitt støy
    - Men linjene ender opp med å bli bredere
* Gaussian smoothiening
  + Eliminerer høyfrekvent støy
  + Er rask fordi kernelen er:
    - Liten
    - Separerbar
* Gaussian mask
  + - Bredden er inverst proposjonal med den originale Gaussianen
    - Convolving med en Gaussian fjerneer høyfrekvent støy, men man legger ikke til noe. Man lager ikke «artifacts»
  + Det at gaussian -> Gaussian er bra når man vil gjøre en 2D – konvolusjon
    - En retning av gangen gir fremdeles et gyldig resultat
    - ?
* Det gir en veldig lik effekt som når man har en uklar linse
  + Dette er sannsynligvis grunnen til at man approksomerer feilen av en linse som en konvolusjon med en gaussian
* En stor sigma gir mye smoothening
  + Men det blir vanskeligere å si hvor en grense er
* Gradient Maximization
  + Man har bildet sitt, og hvor kameraet var i den forrige farmen
* Cannyedge detector
  + Vi vil ha den nøyaktige pressisjonen av bildene
    - Vi vil ha tynne konturer
  + Convolution
    - Gir oss gradient strength og gradient direction
  + Threshholding
    - Non-maxima suppression
    - Man lager linjer som er en pixel vree
    - Hysterese(?)
  + Non-maxima supression
  + Hysterisis thresholding
    - Man tar inn to thresholds, høy og lav
    - Man får linjer
      * Noen fra over high thresholds
      * Og noen kun fra den lave
      * Vi beholder bare de som er knyttet til sterke linjer
  + Man kan variere
    - Sigma (bredden av thresholden)
    - ….
* Scales pace
  + Det å øke fjerner bare detaljer, det legger aldri til dem
* I en del tilfeller finnes det ingen verdier av sigma soom gir deg hele tingen du ser etter
  + F.eks når man ser etter veier fra luft-foto
* Surface fitting
  + Ingen har fått til dette
  + Når man prøver å passe overflaten til et 3D-objekt bassert på det man ser
  + Det er ikke alltid at kanter er den samme typen overgang
* Flere typer grenser
  + Intensitet
  + Lysstyrke
  + Farge
  + Tekstur
* Machine learning
  + Man trenger en databse, og de vil bare fungere på de tingene man har trent dem på
  + Man kan få litt rare egenskaper på grunn av hva de er trent på
    - F.eks at man ignorerer mønster på en skjorte som er tydlig svart-hvit