

TSBB18 – Föreläsning 1

Lets Automate!

Mårten Wadenbäck

(Originalversion: Mikael Persson)

Department of Electrical Engineering
Computer Vision Laboratory
Linköping University

2022-01-17

Idag:

- Kort om delprojekten
- Seminarier
- Bilder, kameror, rektifiering
- Detektion, morfologi
- Tips för Projekt 1

Ändringar

2021:

- Ökat inslag av utveckling på distans (git, arbetsfördelning, ...)
 - *mycket* användbar erfarenhet, även efter covid
- Tyvärr mindre hands-on problemlösning
- Robotsimulator till projekt 2 och 3
 - buzzwords: *digital tvilling*, virtuell miljö, proof-of-concept, ...

2022:

- Projekten är på campus igen – lättare att ordna med handledningen
- Projekt 1: mer hands-on för *hela* gruppen
- Projekt 2: vi börjar med simulatorn, men ...
- Projekt 3: ... senast här kör vi på riktig robot

Sorteringsrobot i tre delar

- Projekt 1 – Livedetektion av lego i video.
- Seminarie 1 – Video och detektion (idag)

Demos – projekten presenteras på korta seminarier.

Sorteringsrobot i tre delar

- Projekt 1 – Livedetektion av lego i video.
 - Projekt 2 – Inverskinematik och styrning av robot.
-
- Seminarie 1 – Video och detektion (idag)
 - Seminarie 2 – 3D Geometri, transformer, och inverskinematik

Demos – projekten presenteras på korta seminarier.

Sorteringsrobot i tre delar

- Projekt 1 – Livedetektion av lego i video.
 - Projekt 2 – Inverskinematik och styrning av robot.
 - Projekt 3 – Legosortering.
-
- Seminarie 1 – Video och detektion (idag)
 - Seminarie 2 – 3D Geometri, transformer, och inverskinematik
 - Seminarie 3 – TBA, (homografier, kamerakalibrering, linux, ...)

Demos – projekten presenteras på korta seminarier.

Livedetektion av lego i video

Varje grupp får tillgång till

- Raspberry Pi och kamera
- Skärm + mus + tangentbord
- Handledning, ca 30 min per vecka

Boka tid med Fredrik Viksten för utlämning av hårdvaran!

Boka tid med mig för handledning (jag kommer även spontanbesöka Akvariet).

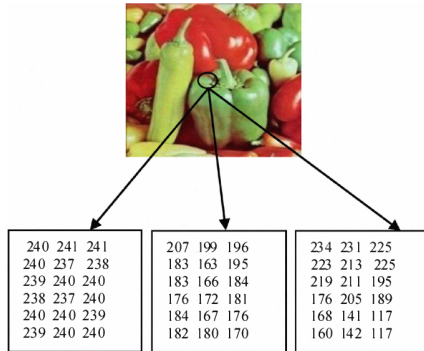
Mål

- Kompilera OpenCV
- Detektera och rita centrum för röda, gröna och blåa legobitar i videoströmmen
- Systemet skall kunna köra i realtid på Raspberry Pi

Projekt 1 skall funka på demo-seminarie 1 (4 februari).

Vad är en digital bild?

- En 2D-funktion/signal $f(x, y)$, ...
- ...ibland till 1D (gråskala), ibland till 3D (färg)
- x och y är pixel-koordinater



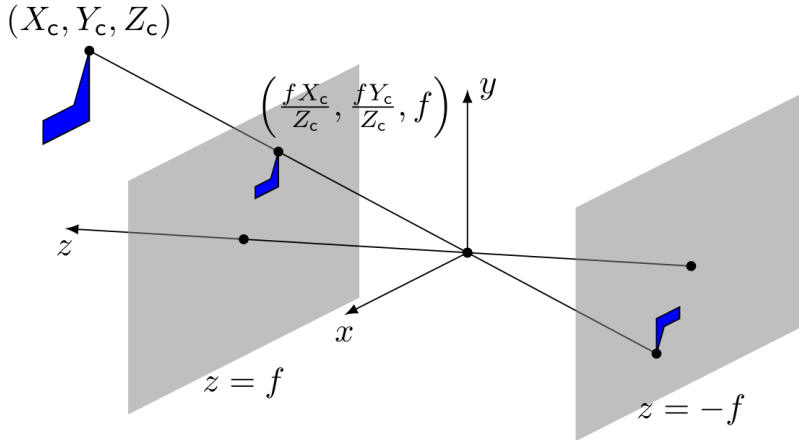
Video – Sekvens eller ström av bilder

- En serie stillbilder $f_1(x, y), f_2(x, y), \dots$
- Alternativt en *tidsberoende* bild $f(x, y, t)$ – för varje fixt t blir det en *video frame*
- Normalt sett varierar bilden måttligt mellan efterföljande frames

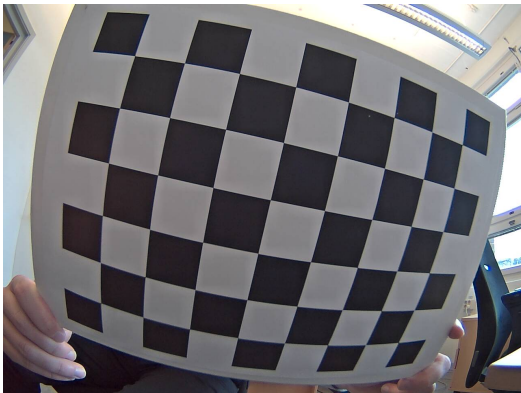
Video eller liveström:

- I tillämpningar: ofta liveström (backkamera, videoövervakning, diverse styrning, ...)
- Under utveckling: ofta förinspelad video (reproducerbarhet, kräver mindre uppställning, ...)
- *Alltid* en fördröjning (överföring, bildprocessering, ...) – realtid är *svårt*!

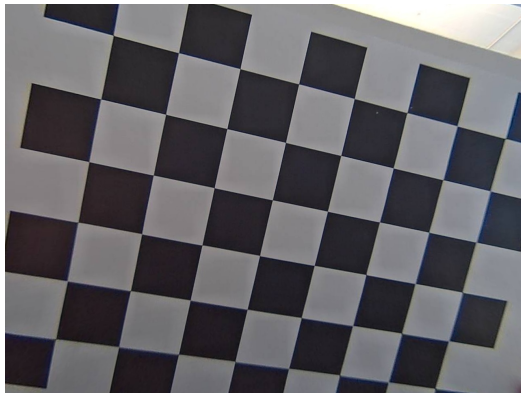
The Perspective Pinhole Camera



Verkliga kameror har olika typer av distortion



Barrel distortion



No distortion

Kamerakalibrering

Vi kan ta bort distortion (rektifiera), om distortionskoefficienterna är kända.

- Distortionsparameterskattning mha schackrutor finns i OpenCV.
- Rektifiering mha distortionsparameterskattning finns i OpenCV.
- Nödvändigt för att korrekt kunna översätta mellan bildkoordinater och 3D-koordinater i projekt 3.

Homografier



Originalbild



Transformerad bild

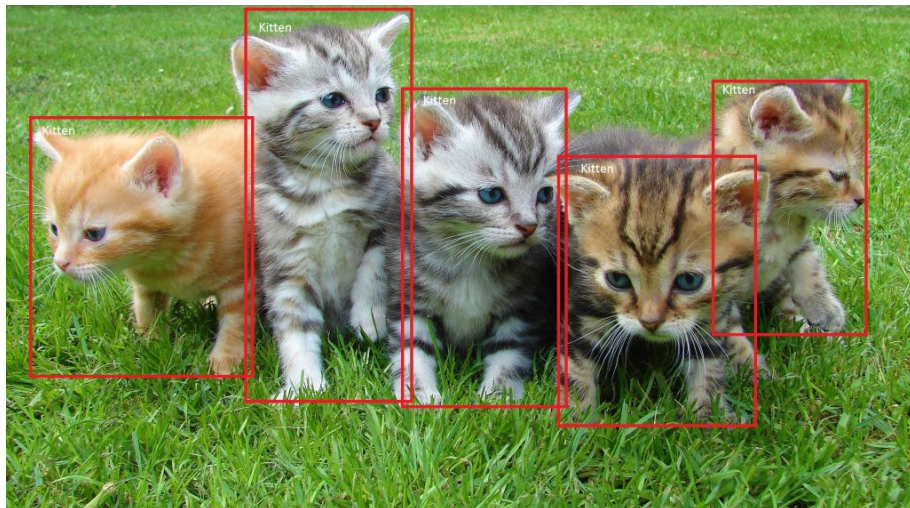
Homografier

En homografi beskriver en perspektivavbildning mellan två plan, exempelvis från ett plan i 3D till *bildplanet* (eller omvänt). De används också för att skapa panoramafoton, eller exempelvis i Google Street View.

Detta kommer bli relevant i projekt 3.

(Djupdyk med CVLs profilkurser.)

Find All Cats!





IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

Metoder

- Deep Convolutional Neural Networks DCNN
- Klassiska metoder (tröskling, morfologi, ...)

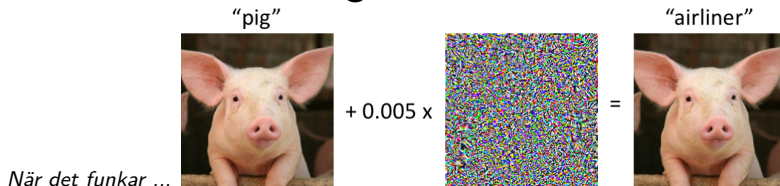
DCNN är överlägset bäst!

När det funkar ...

Metoder

- Deep Convolutional Neural Networks DCNN
- Klassiska metoder (tröskling, morfologi, ...)

DCNN är överlägset bäst!



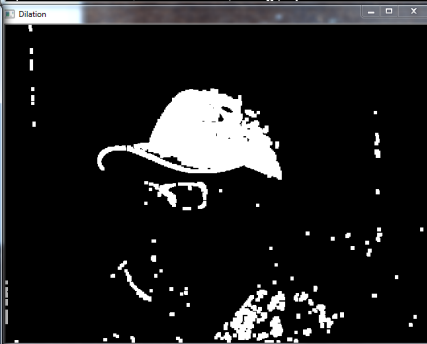
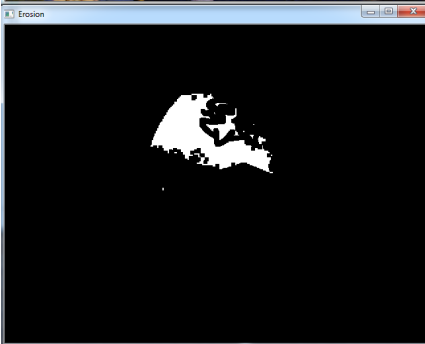
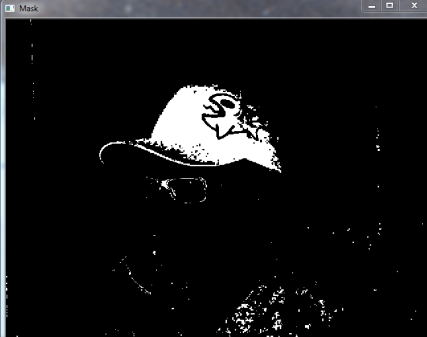
Fundamentalt består både DCNN och morfologiska operationer av faltningar följt av icke-lineariteter.

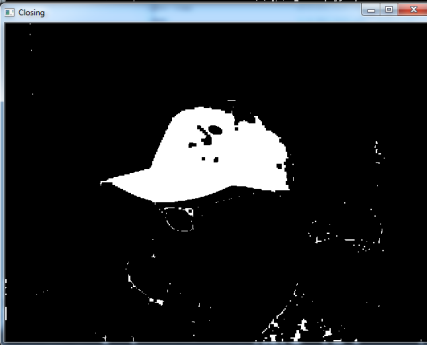
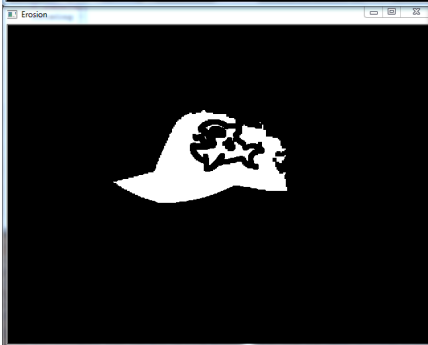
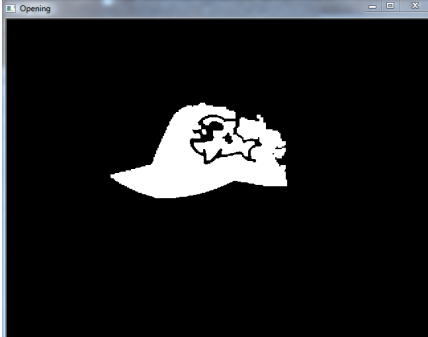
- DCNN är tätare kopplade.
- DCNN kräver massor med data för att träna och tar mycket tid.
- Morfologi kräver att man vet vad man vill göra.

Vanliga Morfologiska operationer

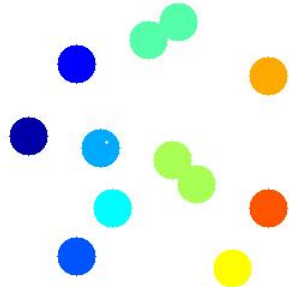
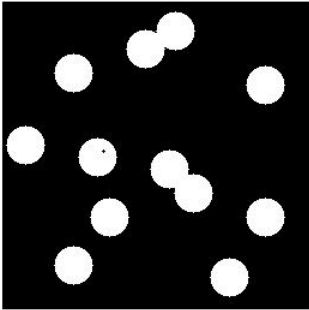
- Tröskling – $IM < \text{threshold}$
- Dilate, erode – denoising
- open/close – denoising
- Identifiera separata object – bwlabel
- Masscentrum
- Finns färdigt i OpenCV

Djupdyk med cvls profilkurser.





Connected Component Labelling



Tips för projekt 1

- Python, Anaconda
- OpenCV – bilder/video, morfologi
- OpenCV – bättre gui med Qt
- gitlab.liu.se

För projekt 2 och 3:

- Koden skall funka i robotsimulatoren innan ni kör på riktig robot.
- Verktyg, skruvar, tejp, snabblim och annat som kan behövas är tillgängligt.
- Roboten är stark nog att slå runt – var försiktiga!