



Object Tracking ved Anvendelse af Hovedmonteret Kamera

Projektevaluering d. 22/6-2001

Gruppe 822

01gr822@kom.auc.dk

Program



Introduktion	Tonny Gregersen
Stabilisering	Peter Korsgaard
Segmentering	Lars Chr. Hausmann
Tracking	Jon Frydensbjerg
Systemtest	Lars Jochumsen Kristensen
Demonstration	Lars Jochumsen Kristensen



Introduktion



Introduktion

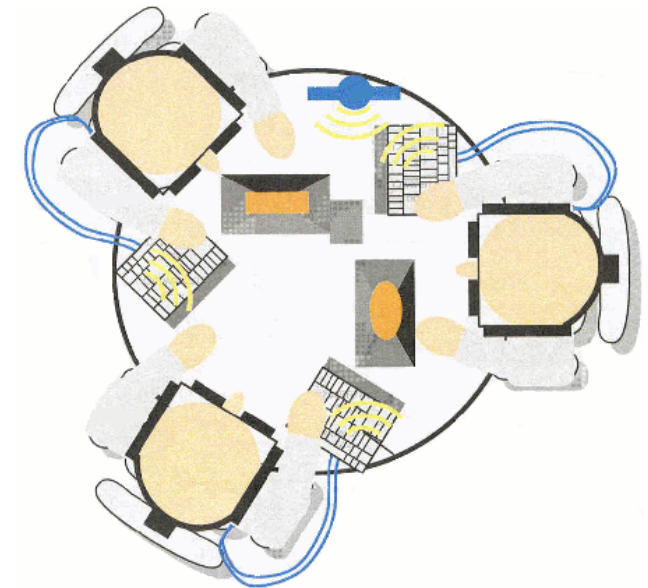
- Motivationen for projektet
- Systembeskrivelse
- Systemarkitektur

Motivation for projektet



Afledt af ARTHUR projektet.

- Et antal brugere sidder om et bord
- Gennemsigtige LCD briller anvendes
- 2 hovedmonterede kameraer som udgangspunkt
- Stereo syn
- Placeholder items på bordet
 - Hverdagsobjekter
 - Må flyttes frit omkring på bordet
- Formål
 - At placere et virtuelt objekt over hver placeholder item
- Minimum 10 billeder per sekund





Systembeskrivelse

Et system skal udvikles, som kan genkende og tracke hverdagsobjekter på et bord vha. billedeanalyse udført på billeder taget af et hovedmonteret kamera.

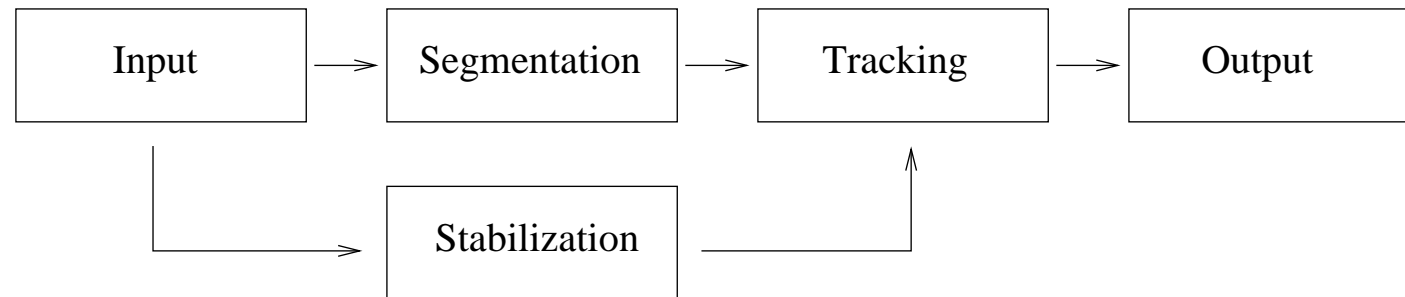
Krav:

- Én bruger
- Ét hovedmonteret kamera
- Minimum 10 billeder per sekund

Begrænsninger:

- Anvender en grøn dug
- Okklusioner fra hænder håndteres ikke
- Begrænset bevægelsesmønster
- Indendørsbelysning

Systemarkitektur



Stabilisering:

- Bestemme kameraets orientering

Segmentering:

- Udsegmentere BLOBs på bordet

Tracking:

- Genkendelse af hverdagsobjekter i udsegmenterede BLOBs



Stabilisering



Motivation

Forbedre robusthed af trackingdelen.

- Bestemme kameraets orientering



Eksempler på ego-motion



Analyse

Miljø:

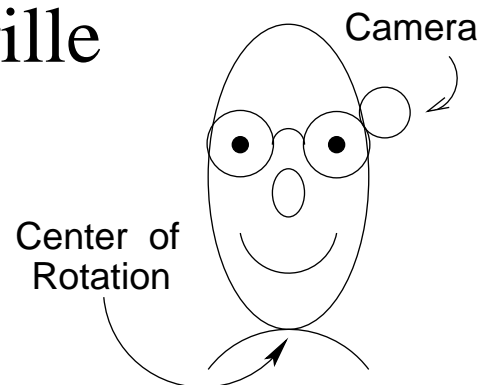
- Bord, objekter og baggrund

Bevægelse:

- Både bevægelse af kameraet (ego-motion) og bevægelse af objekterne (independent motion)
- Brugeren kan bevæge overkroppen og dreje hovedet om alle 3 akser

Kamera:

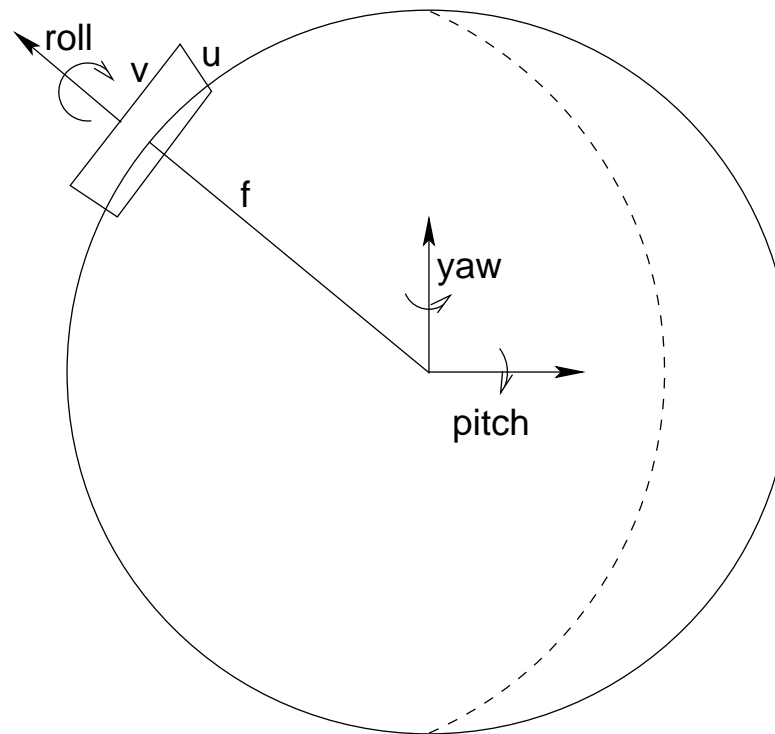
- Monteret på brille





Verdenskoordinatsystem

- Rotation er den primære form for ego-motion
- Sfærisk koordinatsystem med centrum i kameraets rotationspunkt

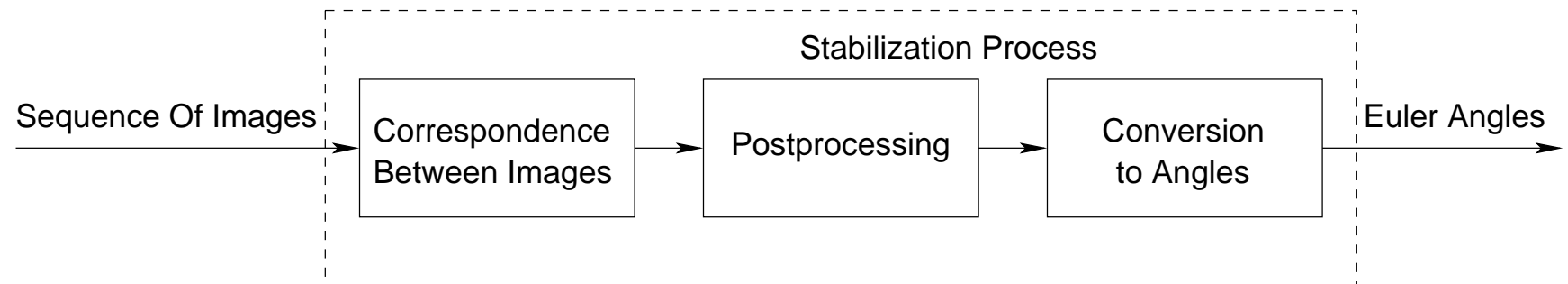




Overordnet design

3 delprocesser:

- Find forskydning af billedepunkter
- Fjern fejl i forskydning (outliers)
- Konverter forskydning til orientering



Input: En sekvens af billeder fra inputdelen

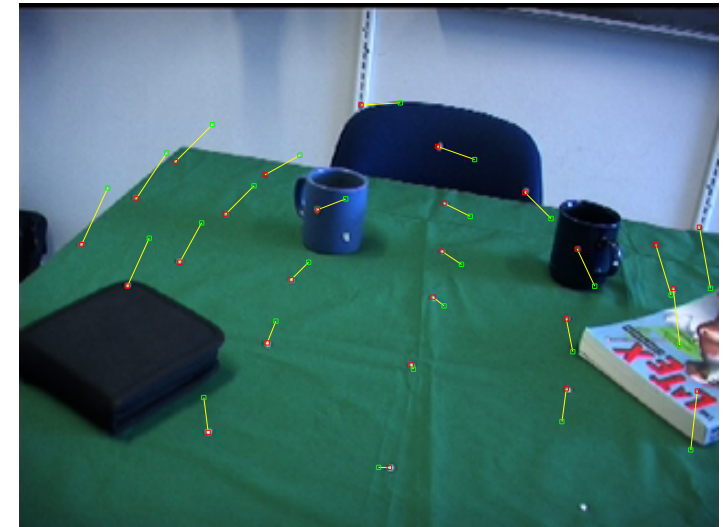
Output: Den nuværende orientering af kameraet relativt til det første billede

Forskydning af billedpunkter #1



3 metoder er undersøgt:

- Feature matching
- Vinduebaseret korrelation
- Gradientbaseret optisk flow



Forskydninger fra rotation

Valg:

	Feature matching	Korrelation	Optisk flow
Begrænsning:	Features	Tekstur	Tekstur, lille bevægelse
Hastighed:	Hurtig	Mellem	Langsom
Tæthed:	Tynd	Mellem	Tæt
Robusthed:	God	Rimelig	Dårlig

Forskydning af billedpunkter #2



Parametre:

- Farverum
- Opløsning
- Vinduesstørrelse
- Metode til udvælgelse af startvinduer
- Repræsentation af vinduer

Test resultater:

- Saturation, 160x120, 16x16 vinduer, threshold på STD på 15, histogrammer med 64 bins



Fjernelse af outliers #1

Outliers:

- Fejl fra beregning af forskydning
- Independent motion af hverdagsobjekter

Forskydninger er kontinuerte i forhold til naboer.

$$e = \sum \frac{1}{dist} \cdot |\overline{diff}|$$

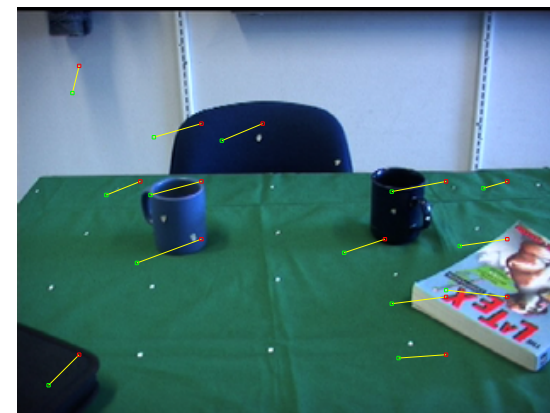
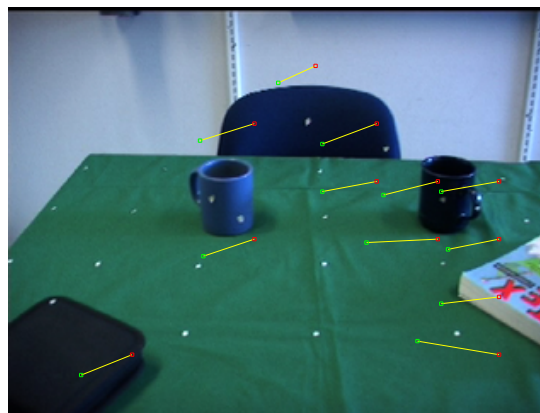
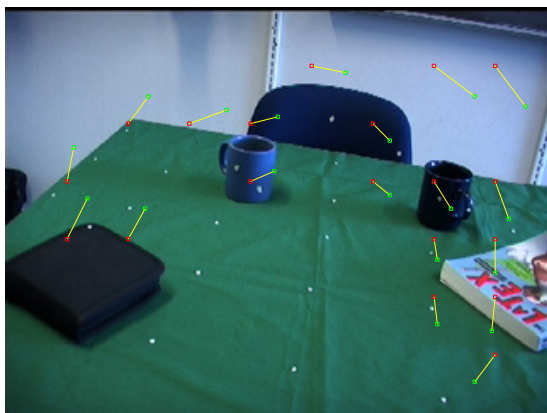
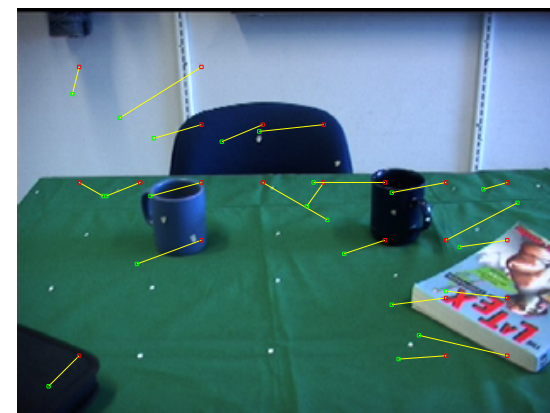
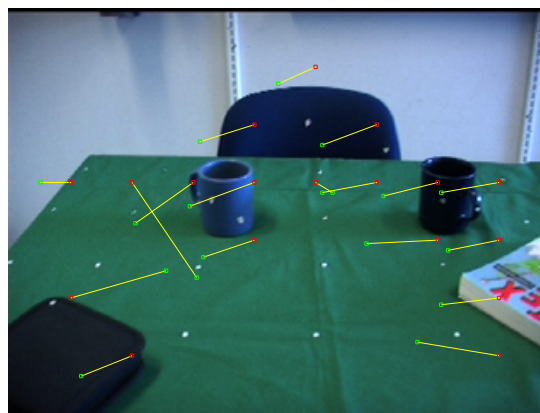
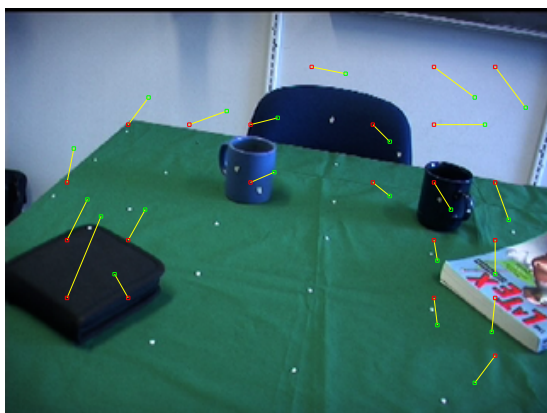
Hvor:

- e fejlen
- $dist$ afstanden fra forskydningsvektor \bar{a} til \bar{b}
- \overline{diff} differensvektor mellem \bar{a} og \bar{b}

Fjernelse af outliers #2



Før fjernelse af outliers



Efter fjernelse af outliers



Konvertering til orientering

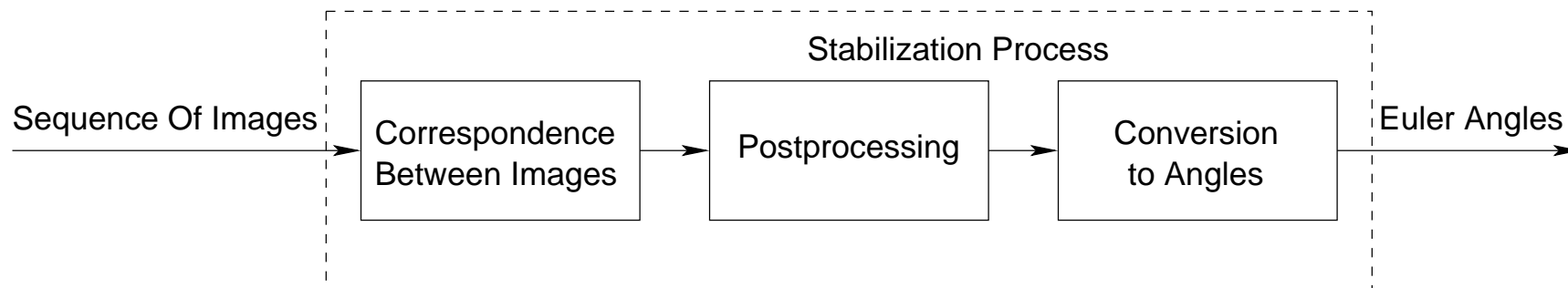
4 metoder er undersøgt:

- 2D affine model
- Hyperbolsk model
- Tensor model
- Mosaik

Valg:

	2D Affine	Hyperbolsk	Tensor	Mosaik
Begrænsning:	Små rotationer, Lille FOV	Ingen roll	Små rotationer	Små rotationer
Hastighed:	Hurtig	Mellem	Mellem	Langsom
Robusthed:	Dårlig	Rimelig	Rimelig	God

Opsummering



Forskydning af billedpunkter:

- Vinduebaseret korrelation med histogrammer

Fjernelse af outliers:

- Lokal kontinuitet

Konvertering til rotationer:

- 2D affine model



Segmentering

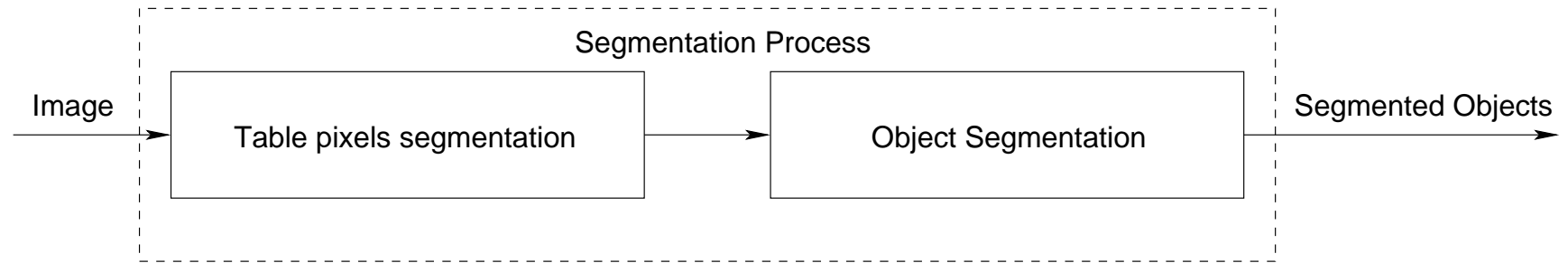


Motivation



- Fjerne uønsket information fra billedet
- At finde objekterne på bordet

Overordnet design



Input: Et billede fra inputkomponenten

Output: Segmenterede objekter (BLOBs) placeret på bordet, som er fundet i billedet



Segmentering af bordet

En hurtig og robust segmentering ønskes.

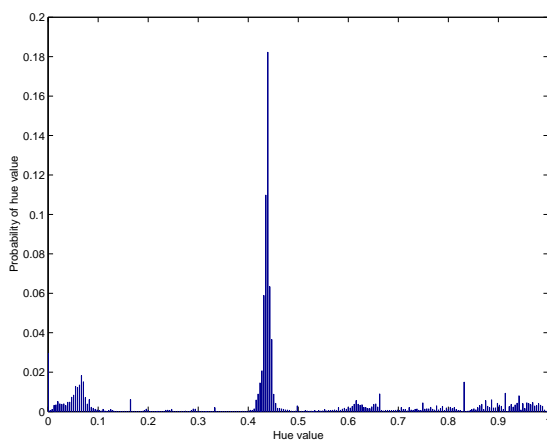
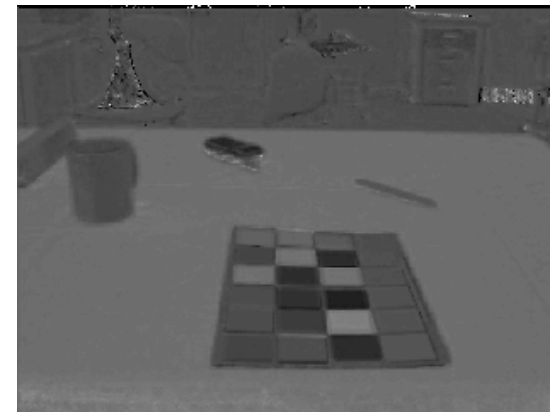
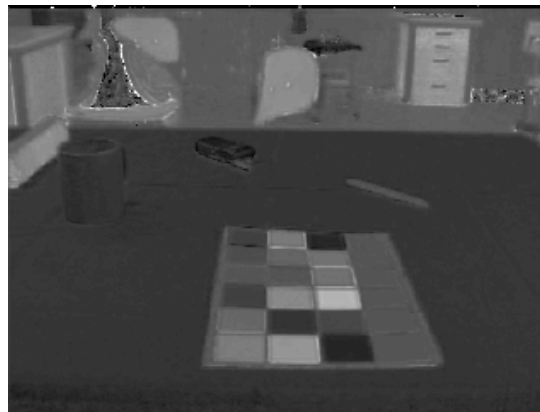
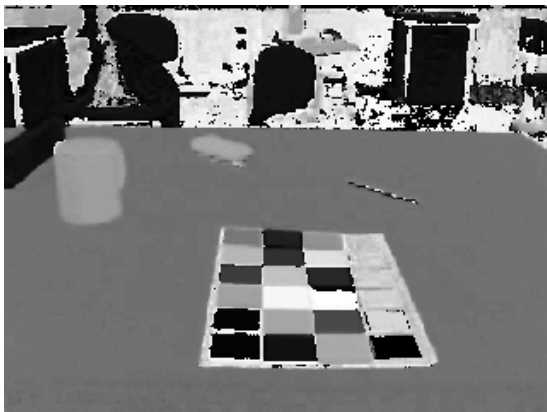
Segmenteringsmetoder:

- Kant detektion
- Global thresholding
- Uniform kvantisering
- Region growing
- Region splitting and merging

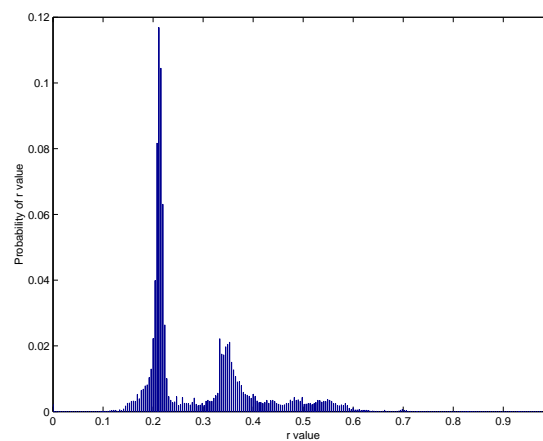
Forudsætninger/krav:

- Udnytte den grønne dug
- Valg af robust farverum
 - Farve og intensitet er separeret
 - HSV eller rgb er mulige valg

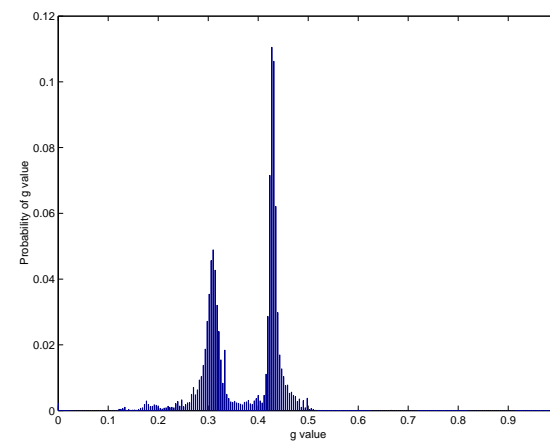
HSV eller rgb?



HSV's H farvekanal



rgb's r farvekanal

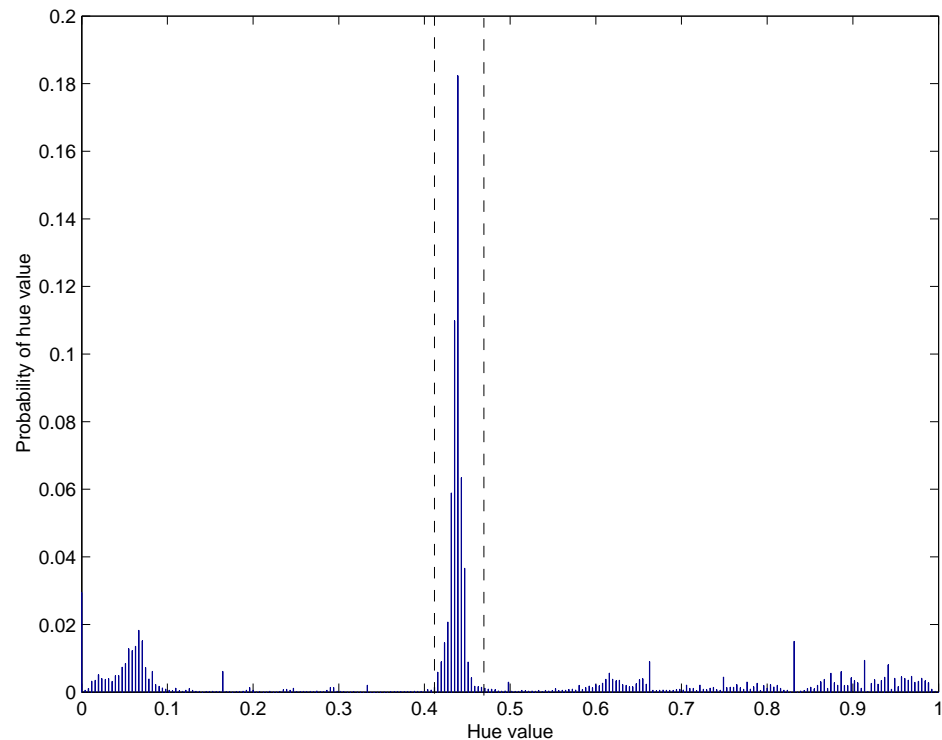


rgb's g farvekanal



Den valgte metode

- Global Thresholding
- Hue-kanalen er valgt
- Threshold bliver fundet ved initialisering af bord





Segmentering af bord #2

Thresholding alene er utilstrækkeligt:

- Global thresholding giver en binær maske
- Billedet indeholder artefakter



Binær morfologi bruges til at fjerne artefakter:

- Opening operation fjerner artefakter
- Der er stadig lukkede områder i nogen objekter





Segmentering af bord #3



For at udvælge bordet bruges “pin pointing”:

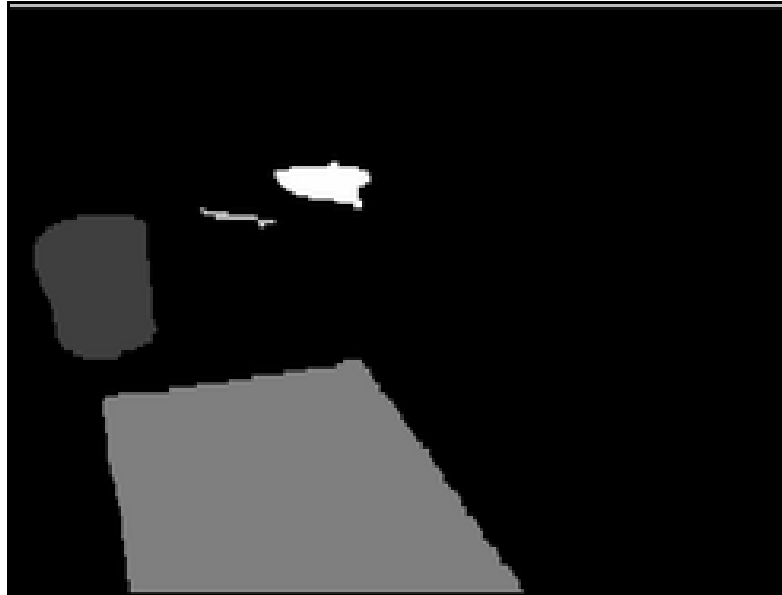
- Alle konturer og deres areal bestemmes
- Konturer hvis areal er mindre end A_{min} pixels fjernes

Segmentering af objekter



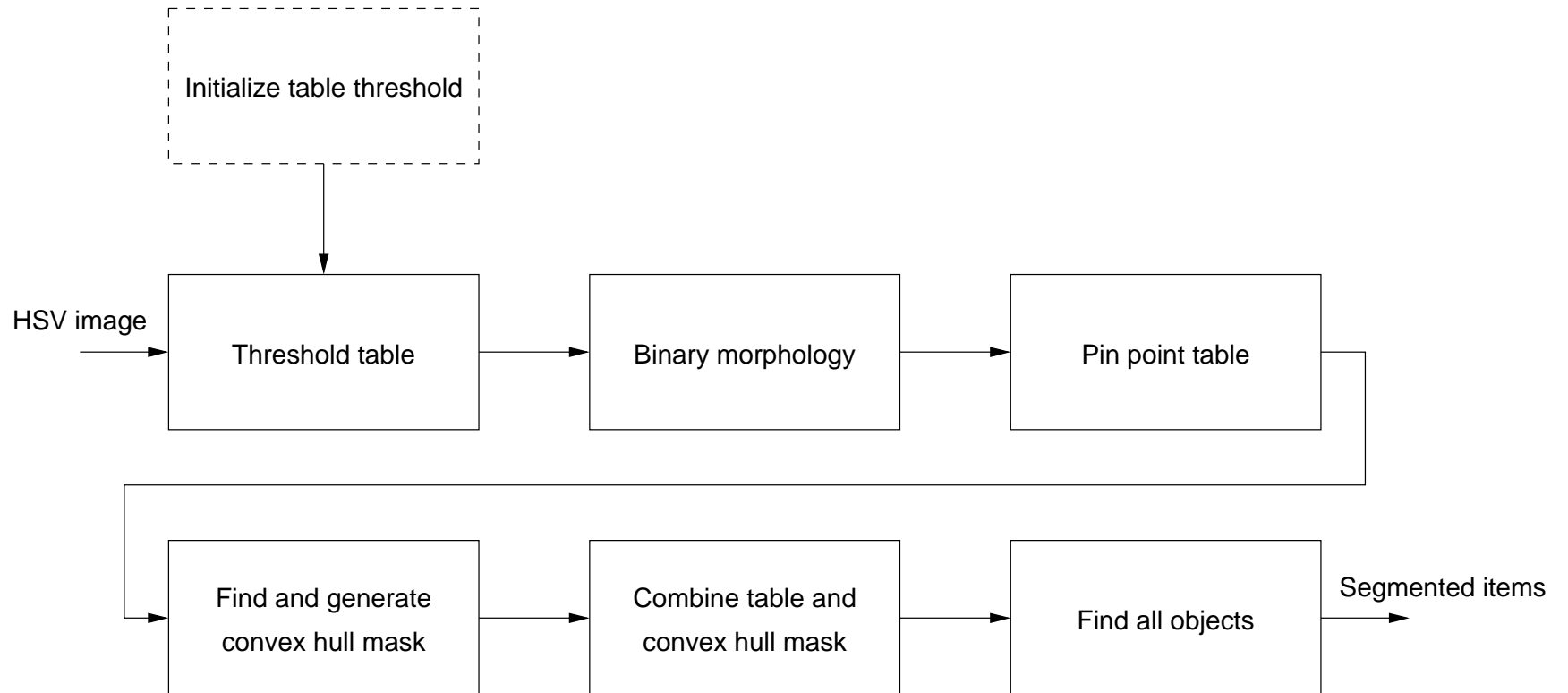
- Udnytte at objekterne ligger på bordet
- Håndtere objekter, som ligger helt ud til kanten
- Lav convex hull maske af bordet

Resultat af segmenteringen



- Objekterne findes ved at lave en *exclusive-or* (XOR)
- Resulterer i een BLOB for hvert af objekterne, som ligger på bordet

Opsummering

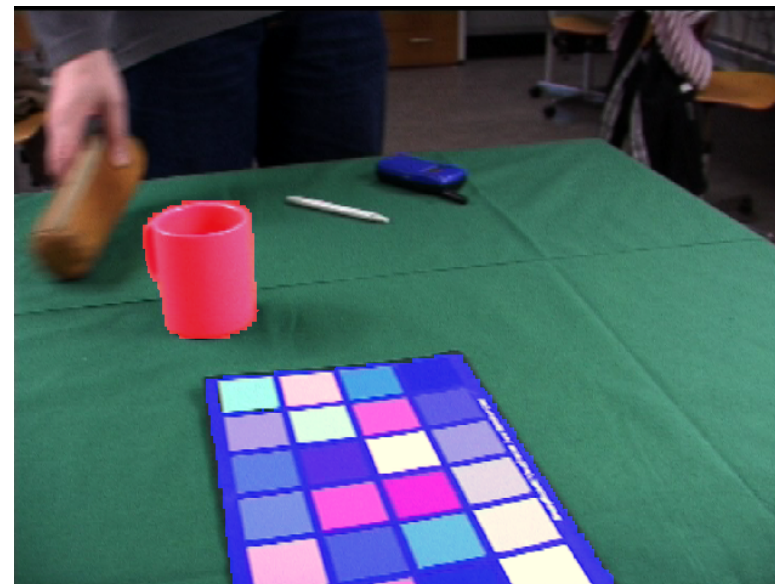
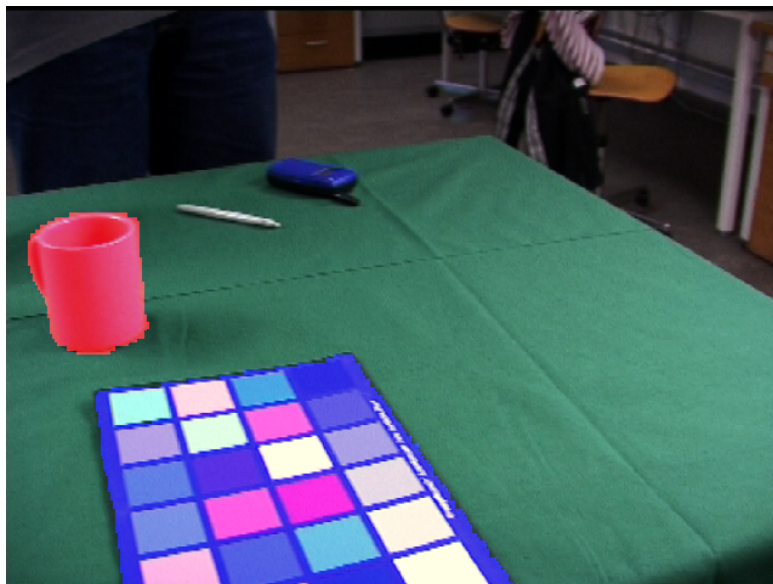




Tracking

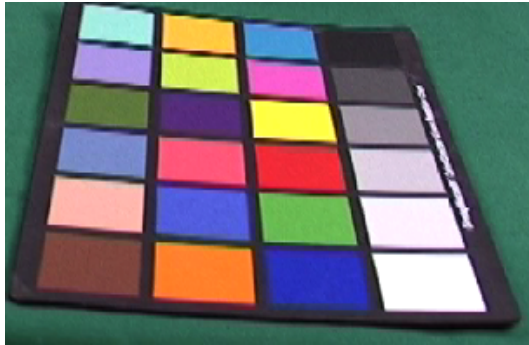


Motivation



Object tracking er en konsistent genkendelse af objekter mellem fortløbende billeder.

BLOB eksempel



(1)



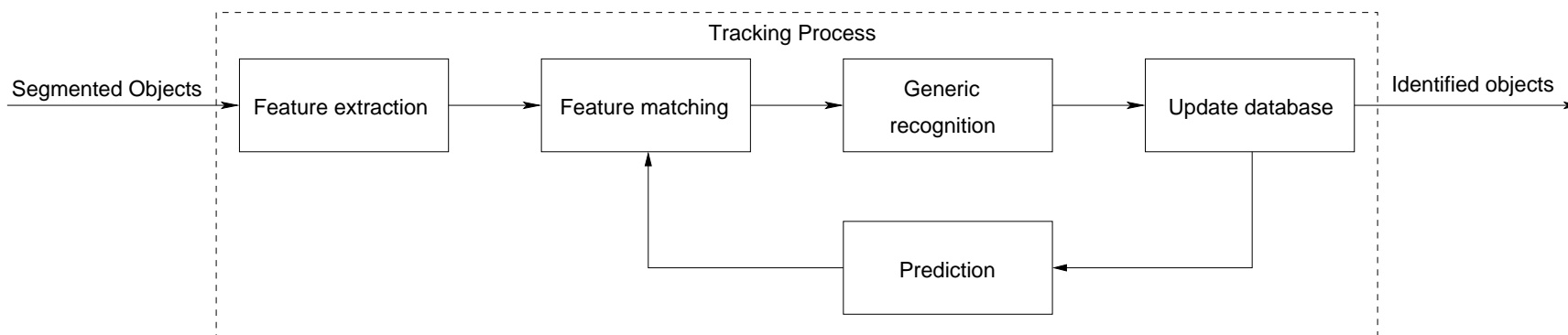
(2)



(3)

1. Rektangulær region indeholdende en BLOB
2. BLOB'ens maske
3. Segmenteret billeddata

Overordnet design



Input: Segmenterede objekter (BLOBs) placeret på bordet, og den øjeblikkelige orientering af kameraet

Output: BLOBs klassificeret som placeholder items fra systemets database



Feature extraction

Invariante og **variante** features samlet i en vektor. Sammenlagt 61 feature-elementer, hvoraf 6 er variante:

- Positionen (2)
- Spatiale størrelse (2)
- Centroiden (2)
- Hu momenter (7)
- Histogrammer af H, S, og V (3×16)

Invariante features er robuste overfor rotation, translation, og skalering, og betragtes som pålidelige gennem hele systemets køretid.



Prædiktion

Lineær ekstrapolation anvendes uden regression.

$$\bar{p}_k = \frac{t_p - t_1}{t_2 - t_1} (\bar{c}_k - \bar{o}_k) + \bar{o}_k$$

Hvor:

- \bar{p}_k prædikeret værdi af feature k
- t_p øjeblikkelig system tid
- t_1 system tid ved næstsidste prædiktion
- t_2 system tid ved sidste prædiktion
- \bar{o}_k værdien af feature k ved næstsidste prædiktion
- \bar{c}_k værdien af feature k ved sidste prædiktion



Feature matching

Korrespondance mellem ekstraherede og prædikterede objekter. Mahalanobis afstand:

$$\mathbf{C}_{ij} = \sqrt{(\bar{a}_i - \bar{b}_j) \mathbf{V}^{-1} (\bar{a}_i - \bar{b}_j)^T}$$

Problem: Udregning af kovariansmatricen, \mathbf{V} .

Faser:

- Find kovariansmatrix til Mahalanobis afstand
- Kovarians vægtning
- Genkendelse



Afstandsmål (kovarians)

Kovariansmatricen, V , estimeres via en sammensat liste af feature-vektorer fra:

1. De m ekstraherede objekter
2. En ringbuffer med de sidste o ekstraherede objekter (max. 250 vektorer)
3. De n objekter i item databasen. Variante og invariante features behandles forskelligt:

Variante: Den prædikterede værdi

Invariante: En vægtet blanding af den prædikterede og originale værdi

V^{-1} findes vha. en low-rank approksimeret SVD.



Kovarians vægtning

Den inverse kovariansmatrix, \mathbf{V}^{-1} , vægtes.

Feature	Vægt
Positionen	2
Spatiale størrelse	2
Centroiden	3
Hu momenter	4
Histogrammer af H, S, og V	1

For hvert objekt i item databasen beregnes en specialudgave af \mathbf{V}^{-1} , hvor variante features nedskaleres over tid.



Genkendelse

Følgende algoritme genkender objekter, startende med det bedste match:

1. Traverser C-matricen, og find den laveste C_{ij} -værdi, m , det bedste match
2. Stop, hvis m er større end C_{match}
3. Marker objekt i fra item databasen som værende det ekstraherede objekt j
4. Sæt alle elementer af C-matricen i række i og kolonne j til ∞
5. Gå til trin 1

$C_{match} = 1.2$ i implementationen.

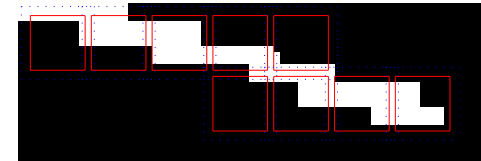
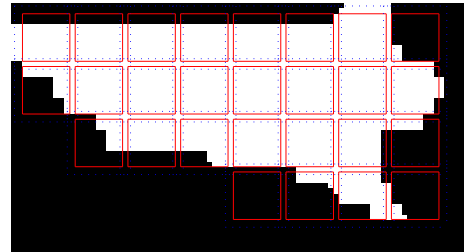
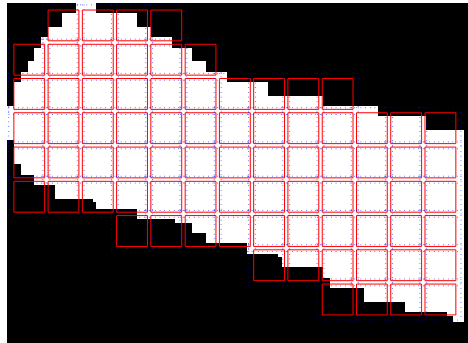
Generisk genkendelse



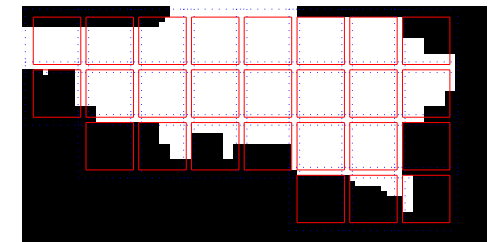
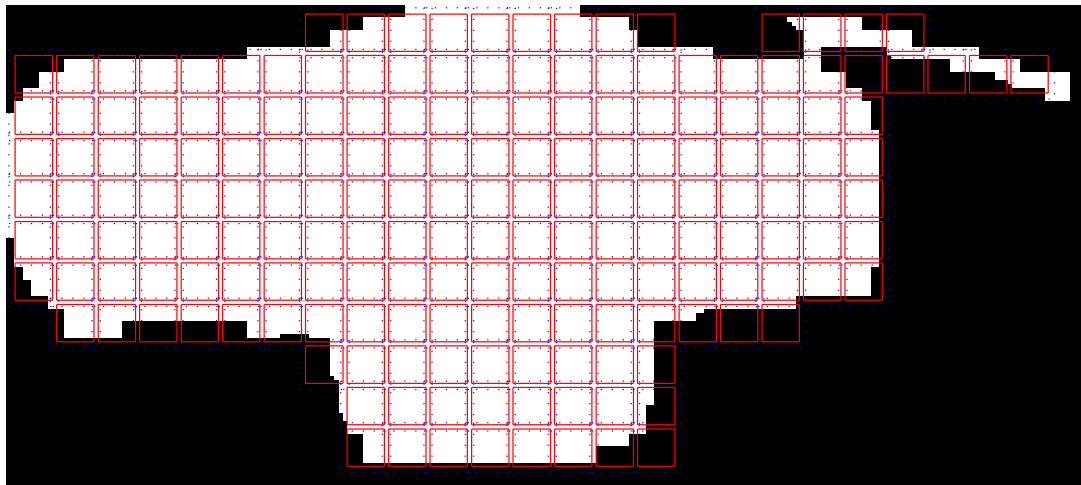
Subregion matching:

- Små regioner anvendes, fx 10×10 eller 12×12 pixels
- Overlap (fx 30%) mellem regionerne øger robustheden
- Histogrammer beregnes for H, S, og V
- Via S-kanalen findes brugbarheden af data i H/V-kanalen
- Genkendelsesalgoritmen fra feature matching anvendes med euklidisk afstand ($C_{match} = 0.1$)

Masker med subregioner



Reference BLOB masker (invariante).

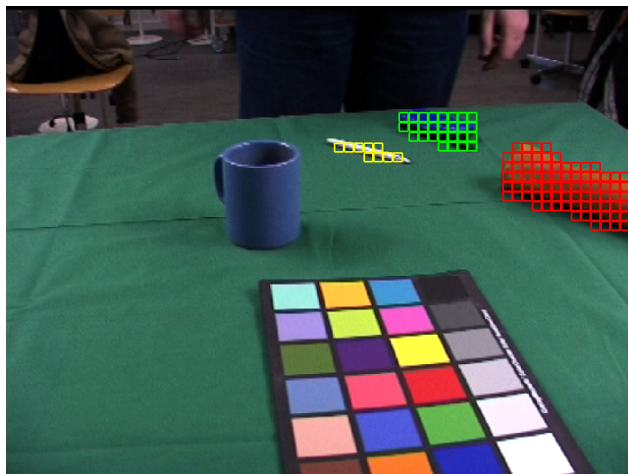


Seneste ekstraherede BLOB masker.

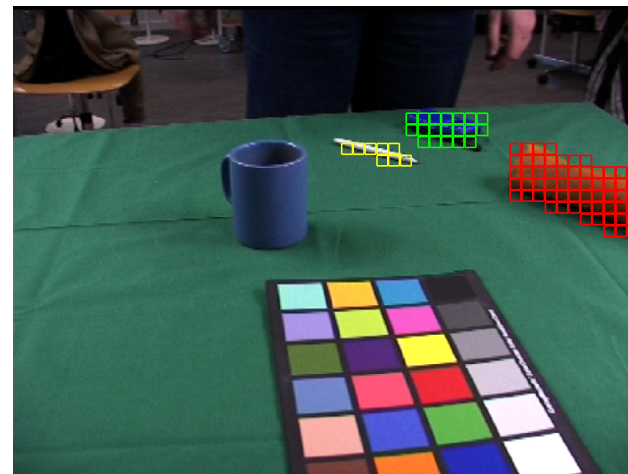
2 subregion eksempler



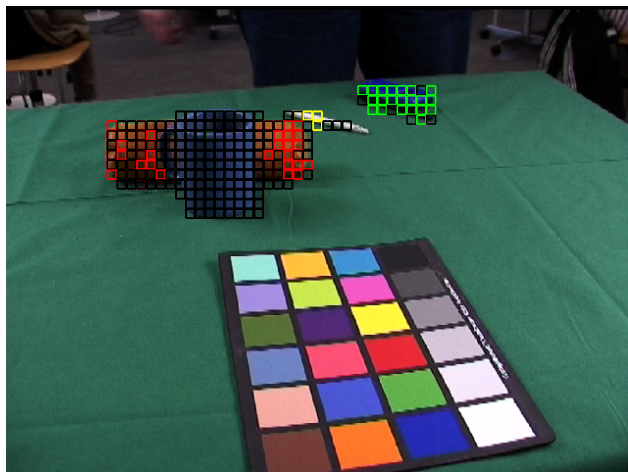
Original subregions.



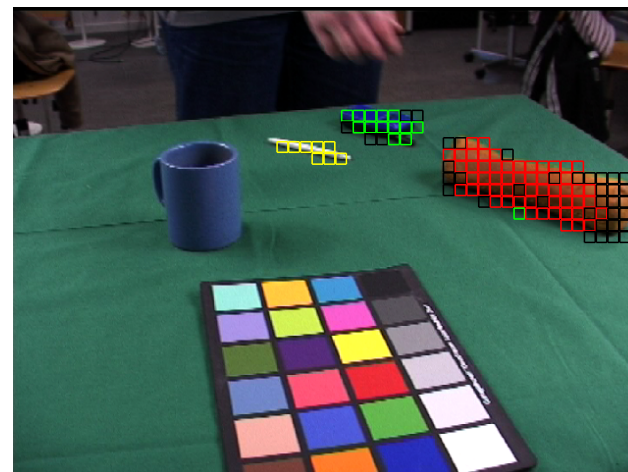
Original subregions.



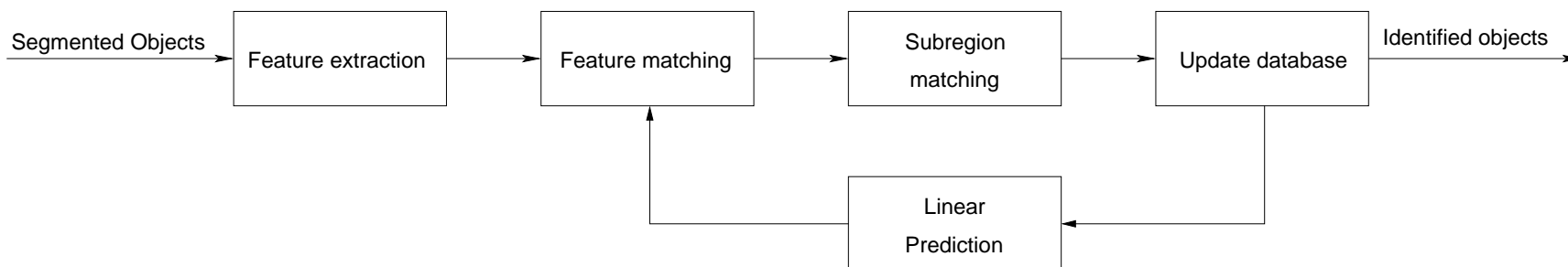
Matched subregions (black regions are unknown).



Matched subregions (black regions are unknown).



Opsummering



Feature extraction: Tilføj formbaserede features, som fx konturbaserede momenter

Prædiktion: Anvendelse af fx et Kalman filter

Feature matching: Brug af mindre komplekst afstandsmål, som stadig inddrager forskelle i det numeriske område for hver feature. χ^2 en mulighed



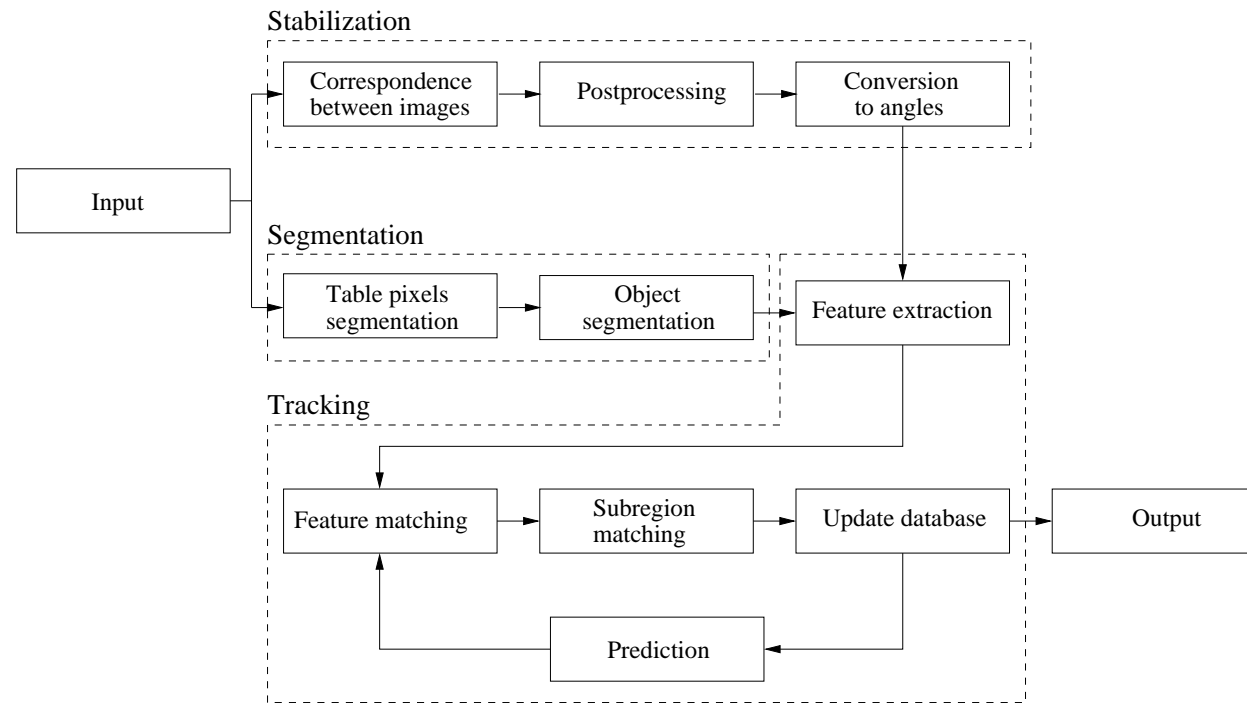
Systemtest og konklusion



Systemtest og konklusion

- Test af det implementerede system
- Test af tidsfordelingen i systemet
- Konklusion
- Diskussion

Det implementerede system

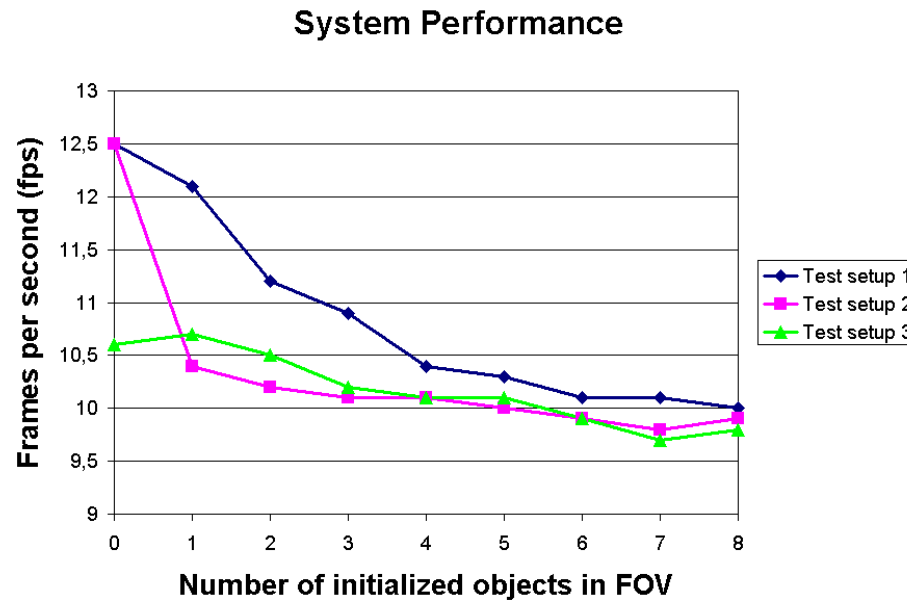


Implementeret i C++

- Input
- Segmentering
- Tracking (uden subregion)
- Output

Implementeret i Matlab

- Stabilisering
- Subregion matching



Test 1: Stigende antal initialiserede objekter

Test 2: Stigende antal initialiserede objekter af de 8 objekter i FOV

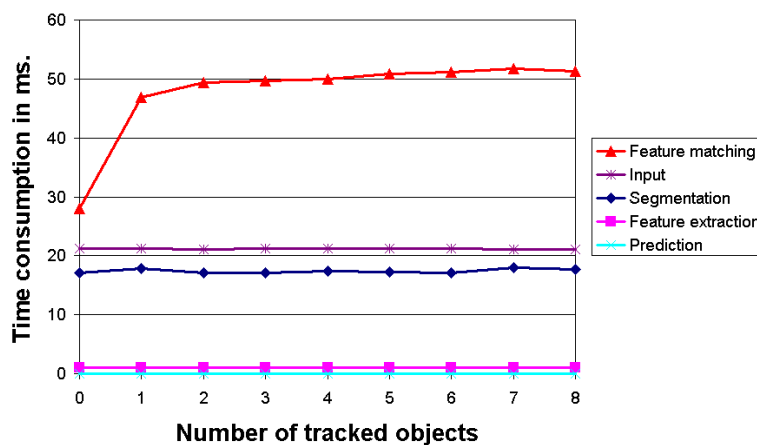
Test 3: Stigende antal i FOV af 8 initialiserede objekter

Tidsfordeling



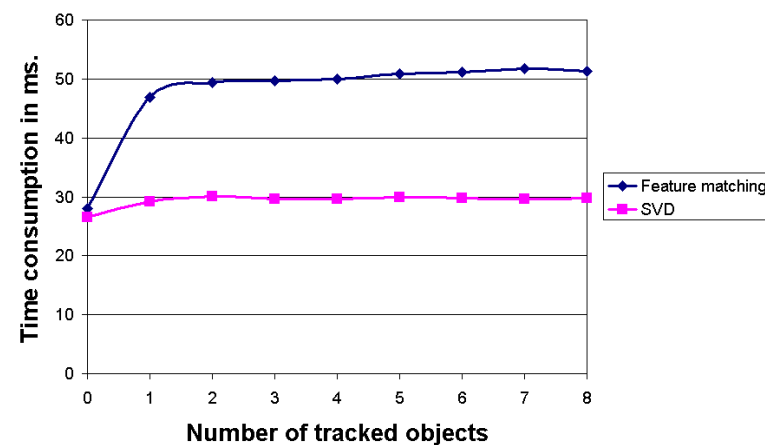
Tidsfordelingen for test setup 2

Time Consumption For Sub Components



- Feature matching bør optimeres
- Segmentering afhænger af antallet af objekter

Time Distribution For Feature Matching



- SVD udgør 60 % af tidsforbruget
- Udregning af kovariansmatricen



Konklusion

Det designede system vs. problemformuleringen

Hvordan kan et system designes, der ved anvendelse af et hovedmonteret kamera, kan tracke hverdagsobjekter på et bord, som er dækket af en uniform, ensfarvet dug, ved en opdateringsfrekvens på 10 billeder/sekund.

Succes

- Op til 6 initialiserede objekter
- Robust overfor skalering, rotation og translation
- Håndterer independent og ego-motion
- Okklusioner

Problemer

- Objekter med identisk farve
- Objekter midlertidigt udenfor FOV



Diskussion

Refleksioner over projektet i forhold til ARTHUR

Stabilisering:

- Udveksling af information mellem brugere
- Stereo vision
- Overblikskamera

Segmentering:

- Ensfarvet bord øger robustheden

Tracking:

- Farvebaserede features velegnede
- Formbaserede features bør tilføjes
- Prædiktør bør forbedres



Demonstration