Universitetet i Stavanger

DET TEKNISK-NATURVITSKAPLEGE FAKULTET

MASTEROPPGÅVE

Studieprogram/spesialisering: Informasjonsteknologi, automatisering og signalbehandling	Vårsemester, 2017 Open
Forfattar: Eivind Nordal Inderøy	…… (signatur forfattar)
Fagansvarleg: Ivar Austvoll Rettleiar: Ivar Austvoll	
Tittel på masteroppgåva: Eit litteraturstudie på objektdeteksjon og attkjenning av køyretøy i ei køyrebane Engelsk tittel: A survey of on-road object detection and vehicle recognition.	
Studiepoeng: 30	
Emneord: Kamerasyn, monosyn, stereosyn, bildebehandling, objektdeteksjon, køyretøyattkjenning	Sidetal: 72 Stavanger, 15-06/2017

Universitetet
i Stavanger

Eit litteraturstudie på objektdeteksjon og attkjenning av køyretøy i ei køyrebane

av

Eivind Nordal Inderøy
Juni 2017

MASTEROPPGÅVE

Det teknisk og naturvitenskapelege fakultet
Informasjonsteknologi, automatisering og signalbehandling

Rettleiar: Ivar Austvoll

Samandrag

Ei mykje omtalt problemstilling er overgangen frå bilar styrt av menneske til autonome bilar som må lese trafikkbilde fortløpande. Utviklinga er komen så langt at all form for automasjon og målesystem som trengs for å få nødvendig informasjon allereie er på plass. Spørsmålet framover vil vere kvar det kan kuttast ned på kostnadar men likevel ha eit robust system.

Hovudmålet for denne oppgåva er å gje innsyn i utviklinga innanfor køyretøydeteksjon fram til dagens «state-of-art» med eit hovudfokus på kamerasyn. Teknologien som dei kommersielle bilfabrikantane nyttar er proprietær, og sidan det ikkje er mogleg å få innsyn i dette er det antatt at dei nyttar nokon av dei presenterte metodane. Ei analyse er gjort på bakgrunn av opne rapportar som presentera metodar for å detektere køyretøy i eit køyrefelt. Rapporten presentera ei oversikt over sensorar og metodar som blir brukt for å skilje mellom køyretøy og ulike objekt i eit trafikkbilde. Det er valt å sortere arbeidet inn i monosyn, stereosyn og ein fusjon av sensorar slik som kamera, radar og lidar.

Fokuset i dette feltet er hurtig skiftande, og det har gått ifrå enkle metodar som søk etter køyretøy på bakgrunn av symmetri, til komplekse eigenskapar som blir definert av djupe nevrale nett og punktskyar frå aktive sensorar. Det mest lovande arbeidet for å nytte i eit sjølvstyrt køyretøy basera seg på ein fusjon mellom aktive og passive sensorar som kontinuerleg har eit overblikk over miljøet rundt køyretøyet.

Forkortingar

Forkorting	Forklaring
DOF	Degrees of Freedom
DoG	Difference of Gaussian
GPU	Graphical Processing Unit
HG	Hypotesegenerering
HOG	Histogram of Oriented Gradients
HV	Hypoteseverifisering
IPM	Invers perspektivmodell
Lidar	Laser imaging, detection and ranging
LoG	Laplacian of Gaussian
MMW	Milli Meter Wave
NN/DNN	Nevrale Nett/Djupe Nevrale Nett
PCA	Principal Component analysis
Radar	Radio detection and ranging
ROI	Region of Interest (Region av interesse)
SAD	Sum of Absolute Differences
SAE	Society of Automotive Engineers
SIFT	Scale Invariant Feature Transform
SSD	Sum of Squared Differences
SURF	Speeded Up Robust features
SVM	Support Vektor Maskin
WHO	World Health Organization

Nomenklatur

Utrykk	Forklaring
d	Forskjell mellom bildepunkt - disparitet
f	Brennvidde
H	Homogen transformasjonsmatrise
$\boldsymbol{H_H}$	Hessianmatrise
I	Bilde
I_I	Integralbilde
K	Indre kalibreringsmatrise
M	Kameramatrise
$\boldsymbol{P} = [X, Y, Z]^T$	Punkt i rommet
$\boldsymbol{p} = [x, y]^T$	Punkt i bildeplan
$\widehat{\boldsymbol{p}} = [\hat{x} \quad \hat{y} \quad 1]^T$	Punkt i eit normalisert bildeplan
$\boldsymbol{P'}, \boldsymbol{p'}$	Punkt i rommet og bildeplan i høgre kamera
R	Rotasjonsmatrise
t	Translasjonsvektor

Innhald

1 Innleiing

1.1 Bakgrunn for oppgåva

Denne oppgåva har bakgrunn i dagens problemstilling med overgangen frå bilar styrt av menneske til autonome bilar som må lese trafikkbilde fortløpande. Utviklinga er komen så langt at all form for automasjon og målingar som trengs for å få nødvendig informasjon allereie er på plass. Spørsmålet framover vil vere kvar det kan kuttast ned på kostnadar men likevel ha eit robust system. Eksempelvis så har George Hotz gått ut og sagt at han skal kunne konstruere ein autonom bil som vil fungere betre, og være billigare, enn Tesla sine alternativ[1]. Den har allereie blitt testa i trafikken i San Francisco.

WHO sin globale statusrapport på vegsikkerheit, som dei ga ut i 2015, syner at det er over 1,25 millionar dødsfall i trafikken kvart år[2]. Mange av desse dødsfalla kjem av menneskeleg feil som kan eliminerast ved å assistere føraren med sensorar eller nytte fullt autonome bilar. Eksempelvis så vart Tesla Autopilot etterforska av den amerikanske trafikksikkerheitsadministrasjonen på grunn av ei dødsulykke i 2016. Etterforskinga viste tydeleg at autopiloten ikkje hadde noko å gjere med ulykka, men derimot at teknologien har redusert ulykkesraten med 40%[3].

Objektdeteksjon og attkjenning i trafikk er eit felt som er forska på i mange år. Arbeid som er gjort før 2005 er nøye gjennomgått av Zehang Sun, George Bebis og Ronald Miller i [4] der det er ei tydeleg utvikling i feltet for køyretøyattkjenning. Sayanan Sivaraman og Mohan Manubhai har i [5] ein detaljert rapport om kva som er gjort med dette frå 2013 og tilbake til 2005. Denne rapporten syner ei bratt utvikling innanfor datasyn både med mono -og stereosyn, men også ved å kombinere datasyn med ulike sensorar som radar og lidar for å betre ulike målingar. Desse rapportane er sett på som «state-of-art» av dei fleste nyare rapportar[6][7][8]. Dei seinare åra visar det seg at utviklinga går på å utvikle nye metodar for eigenskapsutrekning og maskinlæring for å betre presisjon og hastigheit på deteksjonane.

I kommersielle bilar, slik som Tesla, Mercedes og Ford, har det også vore ei eksepsjonell utvikling med eit fokus på å få autonome bilar ut i trafikken. Men løysingane til desse bilfabrikantane er dessverre proprietær og ikkje tilgjengeleg for det offentlege. Med eit slikt fokus og utvikling innanfor dette tema så rasar prisane på sensorar og programvare slik at det om ikkje lenge vil være sannsynleg med kommersielle køyretøy som er utstyrt som fullt autonome bilar.

Vegen til fullt autonome bilar på offentlege vegar har eit stykke igjen. Sjølv om teknologien lar bilane operere i dei fleste miljø må det, eksempelvis i Noreg, framleis leggast fram for Stortinget og bli vedtatt i vegtrafikklova. Vinteren 2017 sendte det norske samferdselsdepartementet på høyring eit forslag til lov om utprøving av sjølvkøyrande køyretøy på veg[9]. På bakgrunn av dette forslaget ligg det ei grundig utreiing om forbetring av køyresikkerheit og ulike personlover om datasikkerheit. Poenget her er at statlege instansar byrjar å sjå nytten bak autonome køyretøy og at det faktisk er teknologisk mogleg.

Denne oppgåva gjer eit innsyn i forskinga som er gjort dei siste åra. Teknologien som dei kommersielle bilfabrikantane nyttar er proprietær, og sidan det ikkje er mogleg å få innsyn i dette er det antatt at dei nyttar nokon av dei presenterte metodane. Rapporten gjer ei oversikt over sensorar og metodar som blir brukt for å skilje mellom køyretøy og ulike objekt i eit trafikkbilde med eit hovudfokus på bruk av kamera. Vidare er det valt å sjå nærare på eigenskapspunkt i bilde for å sortere ut køyretøy og heilt til slutt blir det konkludert med kva som ser ut til å være ei god løysing slik som det er i dag.

1.2 Oppgåvebeskriving

Rapporten fylgjer tre hovudspørsmål:

- **Kva er gjort dei siste åra for autonome bilar og køyretøydeteksjon.**
- **Fordelar og ulemper med kamera som sensor.**
- **Kva er dei beste løysingane i dag på køyretøydeteksjon.**

Måla som er nemnt her kan vidare utdjupast. Hovudmålet er å undersøke kva tidlegare forsking presentera i feltet rundt køyretøydeteksjon med kamera, og kva alternative sensorar som blir brukt. Ved å sjå på trend og resultat skal dei beste metodane presenterast på ein slik måte at det kan antakast kvar vidare forsking vil fokusere. Søket vil ha eit hovudfokus på deteksjon med kamera, og det vil da bli undersøkt kva fordelar og ulemper ein slik sensor kan ha.

1.2.1 Avgrensingar

Oppgåva omhandlar køyretøydeteksjon frå statiske bilde som er tatt frå eit køyretøy sett framover på dagtid. «Tracking» og metodar for optisk flyt blir i denne rapporten ikkje diskutert.

1.3 Rapportinndeling

Denne rapporten er delt inn i to hovuddelar, teori og litteratursøk. I kapittel to blir det presentert teorien bak oppgåva med djupare forklaring for ulike algoritmar og verktøy som er brukt i forsking dei siste åra, med eit djupare fokus på kameradelen. I kapittel tre blir det lagt fram tidlegare arbeid som er gjort med køyretøydeteksjon i ei trafikkscene. I denne delen blir det presentert tidlegare arbeid som ledar til ei samanlikning av dei mest føretrekte metodane. Heilt til slutt blir det presentert eit resultat av litteratursøket med forfattarens eigne innspel og oppsummering av kva som er dei beste metodane.

Kapittel 2 - Teori
I dette kapittelet beskrivast teorien bak kamera og dei viktigaste metodane som er brukt dei siste åra. Dette omhandlar grunnprinsippet bak kameraparameter, både for mono -og stereokamera. Vidare blir teorien for den epipolare avgrensinga forklart med omsyn på det å minimere søket etter eit punkt til eit 1D søk. Dei viktigaste metodane for eigenskapsdeteksjon i bilde og metodar for å finne disparitet mellom to stereobilde, samt dei mest brukte verktøya for klassifisering blir presentert.

Kapittel 3 - Tidlegare arbeid
I dette kapittelet blir objektdeteksjonen delt opp i fleire deler basert på tidlegare arbeid. Ulike vinklingar for objektdeteksjon blir presentert, der det i hovudsak er delt inn i rørsle -og utsjåandebaserte metodar. Her blir fordelar og ulemper med ulike sensorar som radar, lidar og kamera presentert, samt tidlegare arbeid med eit hovudfokus på kameradeteksjon.

Kapittel 4 – Resultat frå dei ulike studia
Forfattaren drar ut dei viktigaste arbeida som er gjort for å detektere køyretøy i ei vegbane dei siste åra og presentera viktige funn og resultat som er blitt gjort.

Kapittel 5 – Diskusjon
Forfattaren diskutera rundt forventningane på resultatet frå tidlegare arbeid.

Kapittel 6 – Konklusjon
I dette kapittelet blir det lagt fram ei vurdering på det endelege resultatet av oppgåva i forhold til problemstilling og resultat.

2 Teori

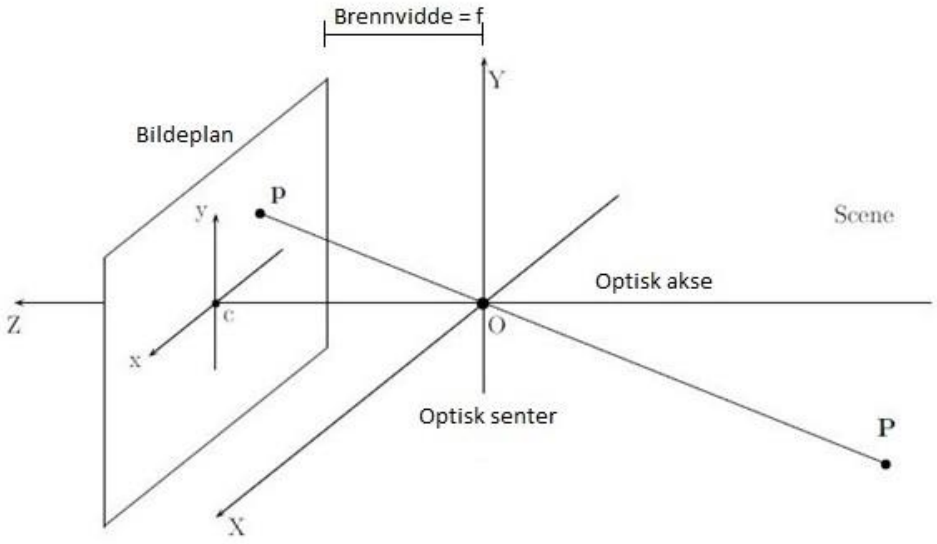
Dette kapittelet går i djupna på dei elementa som er mest aktuelle å kunne ha ei djupare forståing på. Desse elementa ligg som bakgrunn for det arbeidet som er gjort med objektdeteksjon i ei trafikkscene. Først blir teorien bak kameraparameter for eit enkelt kamera presentert, deretter for stereokamera. Nokre av dei mest brukte algoritmane for å finne eigenskapspunkt i bilde presentert, og vidare algoritmar for å finne korrelasjonspunkt mellom stereobilde. Til slutt blir det gått gjennom nokre av dei mest brukte verktøya for klassifisering.

2.1 Monokamera

For å relatere scena i eit køyrefelt til bilde frå kamera må ulike kameraparameter vere kjente. For å enkelt forklare samanhengen mellom verdskoordinatar og bildekoordinatar er eit kamera med ei linse på størrelse med ei nål mykje brukt[10].

Nålehols kameramodell

Figur 1 representera ein enkel kameramodell med kameralinsa, eller det optiske senteret, i punkt **O**. Frå teoremet om nåleholskamera kan det tenkast på kameralinsa som uendeleg liten, då vil lyset frå punktet **P** i ei scene bli representert med ei rett linje til punktet **p** i bildeplanet[10]. Kvart av punkta har koordinatar i kvar sine respektive koordinatsystem, verds -og bildekoordinatar. Denne enkle modellen vil vise objektet opp ned i bildeplanet og tar ikkje høgde for ulike kameraparameter som vil bli diskutert i kapittelet under, kap. 2.1.1. Det er i dei fleste applikasjonar naturleg å tenke på bildeplanet framfor det optiske senteret slik at bildet ikkje blir opp ned.



Figur 1 Nålehols kameramodell: Bilde representera ein enkel kameramodell. Figuren er henta frå *[11]*.

Vidare er punkta representert ved koordinatane $\boldsymbol{P} = [X, Y, Z]^T$ og $\boldsymbol{p} = [x, y]^T$ der samanhengen mellom koordinatsystema blir representert av:

$$\mathrm{x} = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z} \tag{1}$$

Der f er brennvidda til kamera. Desse kan også bli skreve som:

$$\begin{bmatrix}\mathrm{x}\\ y\end{bmatrix} = \frac{f}{Z}\begin{bmatrix}X\\ Y\end{bmatrix} \tag{2}$$

Homogene koordinatar

Koordinatane som representera punkta i bilde -og verdskoordinatar er no oppgitt i Euklidiske koordinatar. Det vil vise seg hensiktsmessig å vidare nytte homogene koordinatar for å representere ulike geometriske transformasjonar ved matriserekning. Eksempelvis så er $\boldsymbol{P} = [X \quad Y \quad Z]^T$ koordinatane til punktet **P** i Euklidiske koordinatar, og $\boldsymbol{P} = [X \quad Y \quad Z \quad 1]^T$ er det same punktet representert av homogene koordinatar[10]. I følgande eksempel så blir samanhengen mellom to Euklidiske koordinatsystem representert av ei rotasjonsmatrise $\boldsymbol{R}$ og ei translasjonsmatrise $\boldsymbol{t}$, desse blir forklart i kap. 2.1.2. Den rigide transformasjonen blir då:

$$\mathbf{P_1} = \boldsymbol{RP_2} + \boldsymbol{t} \tag{3}$$

der $\boldsymbol{P_1}$ og $\boldsymbol{P_2}$ er i euklidiske koordinatar. Ved å utvide med homogene koordinatar kan dette skrivast som:

$$\mathbf{P_1} = \mathbf{TR}\boldsymbol{P_2}, \text{ der } \mathbf{TR} = \begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\ 0^T & 1\end{bmatrix} \tag{4}$$

Punkt i bildeplanet vil no vidare bli representert av dei homogene koordinatane $\boldsymbol{p} = [x \quad y \quad 1]^T$ og punkt i rommet blir representert av dei homogene koordinatane $\boldsymbol{P} = [X \quad Y \quad Z \quad 1]^T$.

2.1.1 Indre kameraparameter

Som nemnt over så har kameramodellen nokre parameter som må tas høgde for. Dette er parameter som fortel noko om brennvidda, skeivheit i bildeplan og det optiske senteret. Ein metode for å finne desse, som er mykje brukt, er å kalibrere kamera med «sjakkbrett-metoden» som blir forklart i kapittel 2.1.3. På bakgrunn av desse parameterane blir det her gjennomgått ei transformasjonsmatrise som transformera eit punkt i scena, gitt ved kamerakoordinatar, til pikslar i bildeplanet. Første steg er å definere det normaliserte bildeplanet der eit punkt i planet er representert ved $\hat{\boldsymbol{p}} = [\hat{x} \quad \hat{y} \quad 1]^T$. Det normaliserte bildeplanet er ein skalert versjon av bildeplanet, og er gitt ved at brennvidda i (2) blir satt til $f = 1$. Likninga kan då skrivast på ny med homogene koordinatar:

$$\hat{\boldsymbol{p}} = \frac{1}{Z}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\boldsymbol{P} \tag{5}$$

der $\hat{\boldsymbol{p}} = [\hat{x} \quad \hat{y} \quad 1]^T$, $\boldsymbol{P} = [X \quad Y \quad Z \quad 1]^T$ og brennvidda no er definert som standard projeksjonsmatrisa $f = \mathbf{\Pi_0} = [\boldsymbol{Id} \quad 0]$ der $\boldsymbol{Id}$ er identitetsmatrisa.

Dei indre kamera parameter blir representert av likning (6) og syner forholdet mellom eit punkt i bildeplanet $\boldsymbol{p} = [x \quad y \quad 1]^T$ og eit punkt i det normaliserte bildeplanet $\hat{\boldsymbol{p}} = [\hat{x} \quad \hat{y} \quad 1]^T$ som er den skalert versjon av bildeplanet.

$$\mathbf{p} = \mathbf{K}\hat{\boldsymbol{p}} = \begin{bmatrix}\alpha & -\alpha\cot\theta & x_0\\ 0 & \dfrac{\beta}{\sin\theta} & y_0\\ 0 & 0 & 1\end{bmatrix}\hat{\boldsymbol{p}} \tag{6}$$

Der den indre kalibreringsmatrisa er definert som:

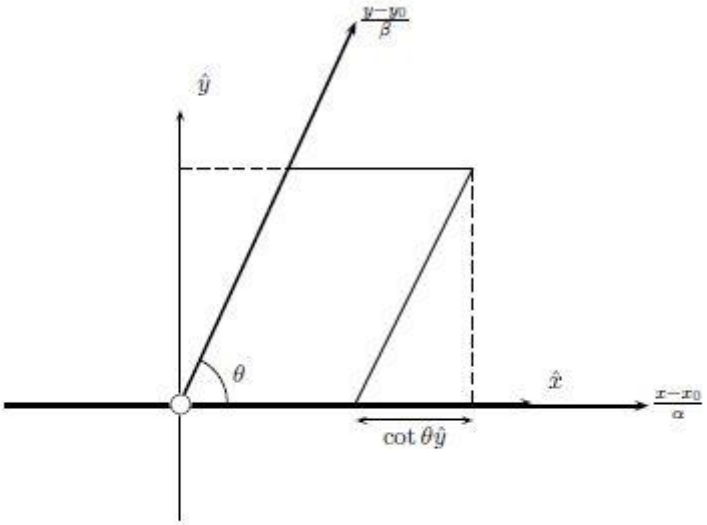
$$\mathbf{K} = \begin{bmatrix} \alpha & -\alpha\cot\theta & x_0 \\ 0 & \frac{\beta}{\sin\theta} & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Dei ulike parameterane i matrisa er:

- $\alpha = kf = \frac{f}{\Delta x}$ som er ei skalering mellom brennvidda og pikselstørrelse i x-retning
- $\beta = lf = \frac{f}{\Delta y}$ som er ei skalering mellom brennvidda og pikselstørrelse i y-retning
- $x_0\ og\ y_0$ er pikselkoordinatar i bildeplanet. Desse er definert som positive heiltal, og har origo i eit hjørne av bildebrikka.
- θ er skeivheit i bildebrikke (meir nøyaktig pixlane) i y-retning. Frå Figur 2 kan bildebrikka da definerast ved dei to likningane:

$$\frac{x - x_0}{\alpha} = \hat{x} - \cot\theta\hat{y} \tag{8}$$

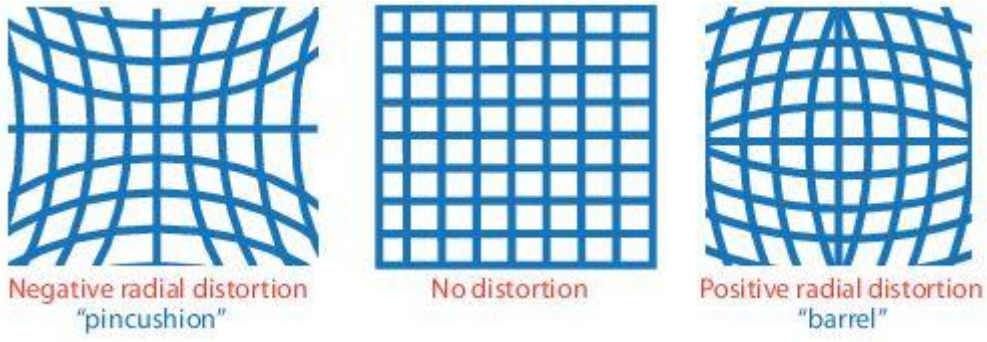
$$\frac{y - y_0}{\beta} = \frac{\hat{y}}{\sin\theta} \tag{9}$$



Figur 2 Skeivheit i bildebrikke: X-aksen er definert som uforandra medan det er ein skeivheit i y aksen som skapar ei skeiv bildebrikke. Figuren er henta frå *[11]*.

Linseforvrenging

Sidan eit reelt kamera har fleire variablar enn den grunnleggande kameramodellen, så må det også takast førehandsreglar med tanke på forvrenging i linsa. Det er i hovudsak tre ulike typar forvrenging[12]. Negativ og positiv radial forvrenging samt tangential forvrenging. Prinsippet for radial forvrenging er at lyset som passera linsa blir vridd og skapar eit forvrenga bilde, medan for tangential forvrenging er det bildebrikka som ligg skeivt i forhold til linsa.



Figur 3 Negativ, nøytral og positiv linseforvrenging. Figuren er henta frå *[13]*.

Figur 3 syner forvrenging som kan oppstå grunna linseoptikken. Negativ forvrenging blir omtalt som nålepute, og positiv forvrenging som tønne. I [12] er det tatt utgangspunkt i følgande modell for å ta høgde for dette etter kamera kalibrering.

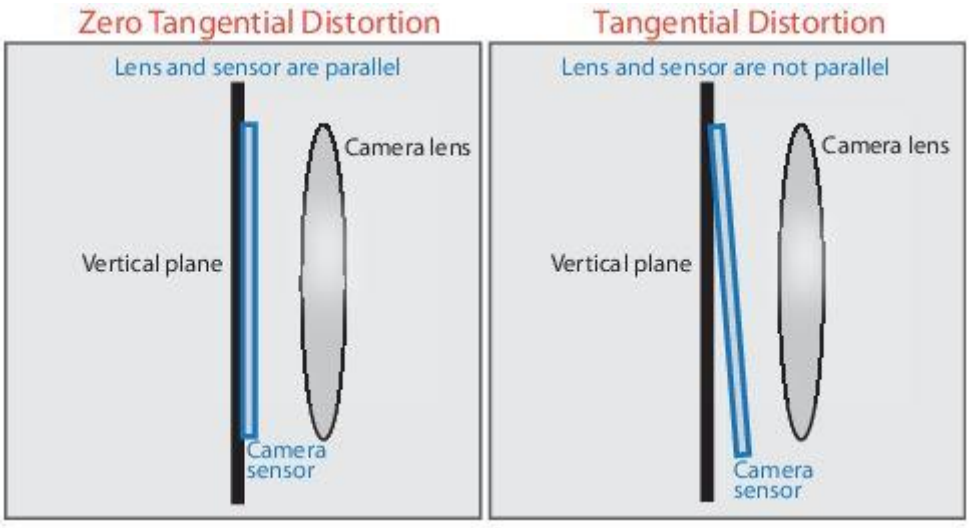
$$x_{dist} = x(1 + k_1 r^2 + k_2 r^4) \tag{10}$$

$$y_{dist} = y(1 + k_1 r^2 + k_2 r^4) \tag{11}$$

Frå likning (10) og (11) har vi følgande parameter:

- x_{dist}, y_{dist}: Pixelkoordinatar med forvrenging
- x, y: Pixelkoordinatar utan forvrenging
- k_1, k_2: Forvrengingskoeffisientar
- r^2: $x^2 + y^2$

I nokre kamera kan også bildebrikka ligge skeivt i forhold til linsa. Dette skapar tangential forvrenging og er synt i Figur 4**.**



Figur 4 Tangential forvrenging der bildebrikka er skeiv i forhold til linsa i kamera. Figuren er henta frå *[13]*.

[12] nyttar følgande modell for å ta høgde for dette etter kamera kalibrering:

$$x_{dist} = x + (2p_1xy + p_2(r^2 + 2x^2) \quad (12)$$

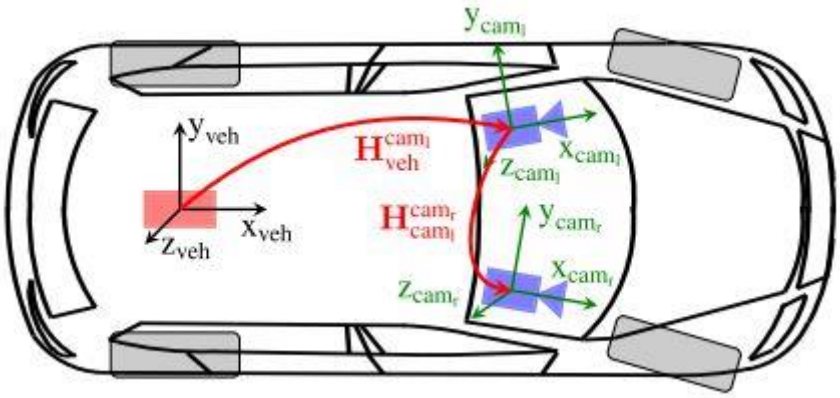
$$y_{dist} = y + (p_1(r^2 + 2y^2) + 2p_2xy) \quad (13)$$

Frå likning (12) og (13) har vi følgande parameter:

- x_{dist}, y_{dist}: Pikselkoordinatar med forvrenging
- x, y: Pikselkoordinatar utan forvrenging
- p_1, p_2: Forvrengingskoeffisientar
- r^2: $x^2 + y^2$

2.1.2 Ytre kameraparameter

For å finne forholda mellom dei ulike kamera og punkt eller objekt i scena, er det nødvendig å vite noko om orienteringa til kamera. Dette er definert som rigid bevegelse og fortel noko om rotasjon og translasjon i X, Y og Z retning i rommet[14].



Figur 5 Ulike koordinatsystem mellom to kamera og køyretøyet. Køyretøyets koordinatsystem er merka med ein raud firkant, og dei to kamera er merka med blått. Rotasjon mellom dei ulike koordinatsystema er definert ved ei homogen transformasjonsmatrise H. Figuren er henta frå *[15]*.

Figur 5 syner to kamera i forhold til verdskoordinatar om bord i eit køyretøy ved homogene transformasjonar. Kamera sine origo har ein translasjon og rotasjon til punktet i verdskoordinatar. For å komme fram til matrisa **H** som fortel noko om rotasjon og translasjon må dei ulike delmatrisene definerast.

Rotasjon
Ei rotasjonsmatrise er definert som $\boldsymbol{R} \in SO(n)$, der SO står for Special Orthogonal Group og n er dimensjonen i rommet. Rotasjonsmatrisa har følgande eigenskapar:

- $\boldsymbol{R^{-1}} = \boldsymbol{R^T}$
- $Det(\boldsymbol{R}) = 1$
- Kolonnar og rader er ortogonale
- Kolonnar og rader er kvar for seg ein einingsvektor

I rommet må rotasjonen rundt kvar av aksane definerast (x, y, z). Dette blir representert av tre rotasjonsmatriser:

$$\boldsymbol{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{14}$$

$$\boldsymbol{R}_{y,\alpha} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \tag{15}$$

$$\boldsymbol{R}_{z,\varphi} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

Avhengig av korleis systemet er satt opp, så er rekkefølga av rotasjonane viktige. Dersom ein rotasjon skal beskrivast i forhold fleire rotasjonar så må matrisene postmultipliserast. Eksempelvis er total rotasjon i eit system som har to rotasjonar: $\boldsymbol{R_{TOT}} = \boldsymbol{R}_{x,\theta}\boldsymbol{R}_{y,\alpha}$. Den totale rotasjonen blir funne ved å først rotere rundt x-aksen, og så rotere rundt y-aksen. Ein meir generell måte å skrive dette på er: $\boldsymbol{R}_3^0 = \boldsymbol{R}_1^0\boldsymbol{R}_2^1\boldsymbol{R}_3^2 = \boldsymbol{R}_1^0 R_3^1$

Translasjon

Translasjonen fortel noko om avstanden mellom origo til eit koordinatsystem i forhold til eit anna langs dei ulike aksane. Det blir skrevet som:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x + x_0 \\ y + y_0 \\ z + z_0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \boldsymbol{t} \tag{17}$$

der $\boldsymbol{t} = [x_0 \quad y_0 \quad z_0]^T$ er ein translasjonsvektor.

Homogene koordinatar

Med homogene koordinatar får vi likning (18) for rotasjon og likning (19) for translasjon[11].

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

der **R** innehar total rotasjon slik som $\boldsymbol{R}_3^0$ over. Den samla transformasjonen kan då bli skriven som i likning (20).

$$\mathbf{H} = \mathbf{TR} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix} \tag{20}$$

Regelen for postmultiplisering gjeld også for homogene transformasjonar. Frå Figur 5 er det definert koordinatsystem for køyretøyet og to ulike kamera. Da er den totale transformasjonen frå køyretøy til det høgre kamera definert som $\boldsymbol{H}_{TOT} = \boldsymbol{H}_{cam_l}^{cam_r}\boldsymbol{H}_{veh}^{cam_l}$.

Kameramatrise

For å forenkle notasjonen vidare så er det hensiktsmessig å definere ei samla matrise for rotasjon, translasjon og dei indre kameraparameterane. Dei indre parameterane, $\boldsymbol{p} = \boldsymbol{K}\widehat{\boldsymbol{p}}$, er definert i kapittel 2.1.1, og dei ytre parameter er definert ovanfor. Vidare så er forholdet mellom det normaliserte punktet og punktet i rommet definert som $\widehat{\boldsymbol{p}} = \frac{1}{Z}\boldsymbol{\Pi}_0\boldsymbol{P}^C$, der $\boldsymbol{\Pi}_0$ er definert som standard projeksjons matrisa og punktet $\boldsymbol{P}^C$ er i kamerakoordinatar. Ved å sette saman likningane får vi:

$$\lambda\boldsymbol{p} = \boldsymbol{K}\boldsymbol{\Pi}_0\boldsymbol{P}^C \tag{21}$$

Der $\lambda = Z$ er ein ukjent skaleringsfaktor sidan djupna er ukjent, og punktet i kamerakoordinatar er definert som:

$$\boldsymbol{P}^C = \boldsymbol{TRP}^W \tag{22}$$

Der $\boldsymbol{P}^W$ er i verdskoordinatar. No er forholda mellom dei ulike punkta kjent og:

$$\lambda \boldsymbol{p} = \boldsymbol{K}\boldsymbol{\Pi}_0\boldsymbol{TRP}^W = \boldsymbol{MP} \tag{23}$$

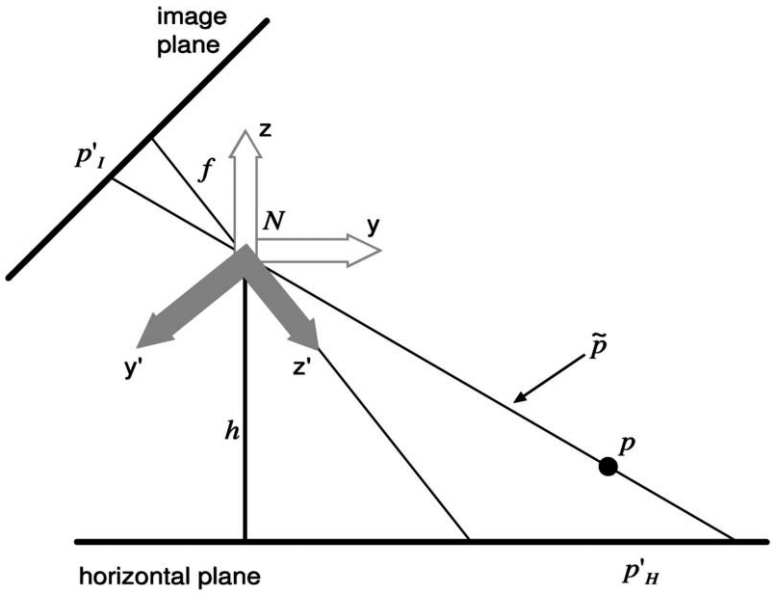
Kameramatrisa er no definert som:

$$\mathbf{M} = \boldsymbol{K}\boldsymbol{\Pi}_0\boldsymbol{TR} \tag{24}$$

Invers perspektivmodell

Frå eit kamera sitt perspektiv så vil eit objekt ha ulik størrelse relativt til avstandar i scena. Dette er kalla perspektiveffekten og gjere til at objekt ser mindre ut jo lenger avstand det er til kamera. Ein metode for å forenkle eller fjerne perspektiveffekten er å innskrenke 3D punkta i ei scene til ei horisontal 2D-flate. Resultatet blir da ein invers perspektivmodell (IPM) [16],[17].

Som forklart i byrjinga av kapittel 2.1 så blir eit punkt i bilde blir danna ved å fylgje eit punkt i rommet tilbake til bildeplanet. Det bildet som no er konstruert i bildeplanet vil vere påverka av perspektiveffekten, og kan bli forenkla ved å nytte IPM. Frå Figur 6 er det synt eit eksempel på geometrien rundt IPM. Frå eit punkt $\mathbf{p'}_\mathbf{I}$ i bildeplanet går det ei linje ut i rommet, gjennom scenepunktet **p**, til punktet $\mathbf{p'}_\mathbf{H}$ i det horisontale planet. Der denne linja kryssar det horisontale planet er resultatet av invers perspektivmodellen for eit punkt i bildeplanet. Ved å utføre den invers perspektivmodellen på kvart bildepunkt, er det mogleg å kartlegge alle punkt $\mathbf{p'}_\mathbf{H}$ i eit bildeplan slik at resultatet blir eit oversiktsbilde som sett ovanfrå. Ergo punkt i 3D-rommet er transformert til punkt i eit nytt 2D-plan. Alle objekt som ligg på overflata vil no bli forvrengt i det nye bildet slik at til dømes køyretøy kan bli detektert.



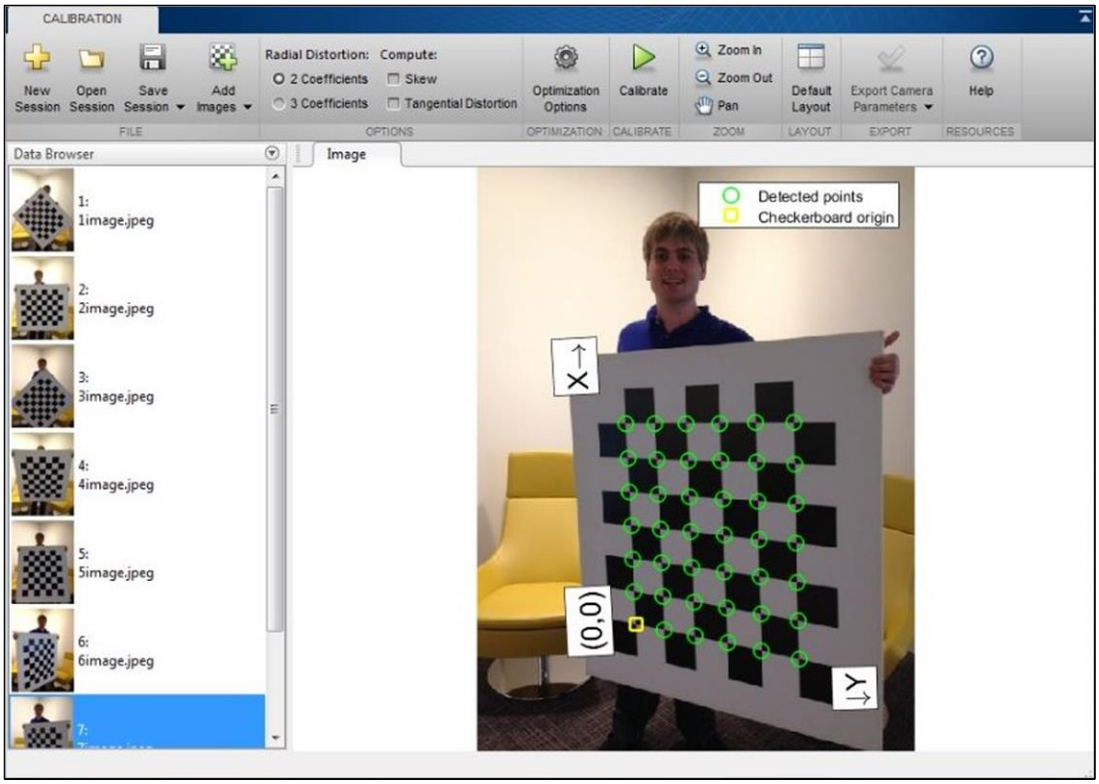
Figur 6 Invers perspektivmodell: Ved å trekke ei rett linje mellom bildeplan og punkt i rommet kan det genererast eit oversiktsbilde over overflata i rommet på den horisontale flata. N er her senter av kamera -og verdskoordinatsystema. y' og z' er kamera-koordinatar, y og z er verdskoordinatar. x-koordinatane, både for kamera og rommet, er vinkelrett inn i arket. **p** er eit punkt i rommet der $\mathbf{p}_\mathbf{I}$ og $\mathbf{p}_\mathbf{h}$ er prosjekteringa av dette punktet i bildeplanet og det horisontale planet. $\tilde{\boldsymbol{p}}$ er den homogene representasjonen av **p'**. f er brennvidda i kamera og h er høgda til kamera over bakken. Figuren er henta frå *[16]*.

I kapittel 2.2.4 blir det gått gjennom triangulering som er ei løysing på å rekonstruere ei 3D scene frå stereosyn. Dersom det ikkje er naudsynt med ein komplett rekonstruksjon av scena så kan IPM nyttast som eit alternativ. For å gje eit betre eksempel så blir det i kapittel 3.4.1 presentert arbeid som er gjort i samband med IPM og køyretøydeteksjon.

2.1.3 Kalibrere kamera

Matrisene frå kap. 2.1.1 og 2.1.2 er ikkje rett fram å finne. Produsenten av kamera kan ofte legge ved detaljar slik som størrelse på bildebrikke, brennvidda og slike ting, men for å få all informasjonen som trengs må kamera kalibrerast. Ulike metodar er blitt presentert i litteraturen, men den som er oftast referert til er «A Flexible New Technique for Camera Calibration»[18] av Zhengyou Zhang som kalibrerer eit kamera ved hjelp av å ta fleire ulike bilde av eit sjakkbrett. Ved å finne eigenskapspunkt i bilde vert kameraparameterane estimert sidan mønsteret på sjakkbrettet er kjent. Denne metoden ledar til ein ganske så nøyaktig estimering av dei indre -og ytre kameraparameterane. Figur 7 syner eit eksempel på ei kalibreringsprosedyre i Matlab. Der er hjørna mellom rutene i sjakkbrettet detektert som eigenskapspunkt, og origo og retning på sjakkbrettet er markert. Ved å nytte ferdiglaga programvare blir dei indre og ytre parameterane estimert og returnert.

For køyretøyattkjenning er det også brukt online-kalibrering, sjå kapittel 3.4.3. Dette har ein samanheng med at dei ytre parameterane er skiftande medan køyretøyet er i bevegelse.



Figur 7 Kamera kalibrering: Eigenskapspunkt frå sjakkbrettet blir detektert frå ulike rotasjonar. Bildet er henta frå [19]

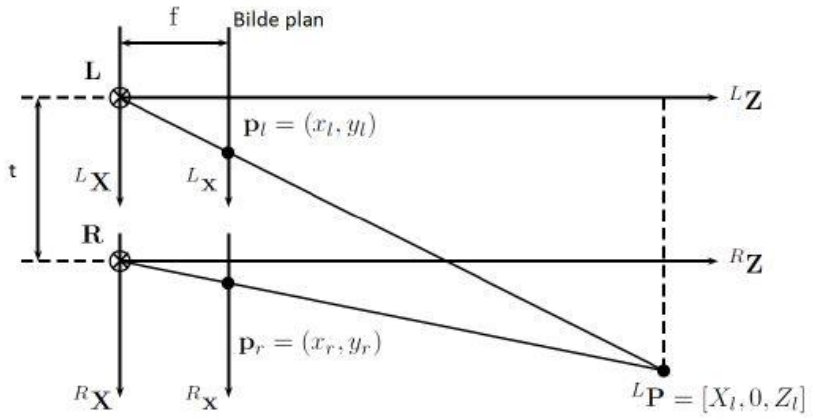
2.2 Stereokamera

Resultatet av to kamera som ser på same objekt tilsvara eit stereosyn, og det blir i litteraturen dratt parallellar til det menneskelege synet. Når dei indre og ytre kameraparameterane mellom to kamera er kjent kan djupna og ein 3D-modell av scena bli konstruert. Det er her essensielt å løyse to hovudproblem; korrespondanse -og rekonstruksjonsproblemet. Dette kapittelet tar for seg geometrien som trengs for å utføre desse utrekningane.

Denne teorien er forska på i mange år, og er i denne rapporten forsøkt forklart i si enkelheit. For ei meir kompleks og djupare lesing anbefalast boka «Computer Vision: A modern approach» og kompendiet til Ivar Austvoll [10], [11].

2.2.1 Enkel modell

Figur 8 syner to kamera, L og R, med ein avstand t ifrå kvarandre, og ein avstand $Z = L_Z = R_Z$ til punktet **P** i verdskoordinatar. Her er L og R det optiske senteret i kvart kamera. Figuren syner translasjon mellom kamera og ingen rotasjon. Da vil rotasjonsmatrisa tilsvara identitetsmatrisa **R = I**. Vidare er begge kamera sine y-retningar inn i arket og satt til null. Til forskjell frå kapittel 2.1 så er bildeplanet her plassert framfor det optiske senter på begge kamera, slik at det resulterande bildets orientering vil vere lik som i scena. Avstanden mellom dei optiske sentera i kvart kamera blir kalla baselinja. Denne er lettare å sjå i 3D modellen i Figur 9. Verdspunktet **P** blir representert som x_l i det venstre bildeplanet og x_r i det høgre bildeplanet.



Figur 8 Geometri mellom kamera, L og R, og punkt P i rommet. Figuren er henta frå *[11]*.

Ved å nytte denne informasjonen og formlikheit i modellen kan vi sette opp forhold mellom punkta slik:

$$\frac{\mathrm{Z}}{f} = \frac{X_l}{x_l} \rightarrow X_l = \frac{Z}{f} x_l \tag{25}$$

$$\frac{\mathrm{Z}}{f} = \frac{X_l - t}{x_r} \rightarrow X_l = \frac{Z}{f} x_r + \boldsymbol{t} \tag{26}$$

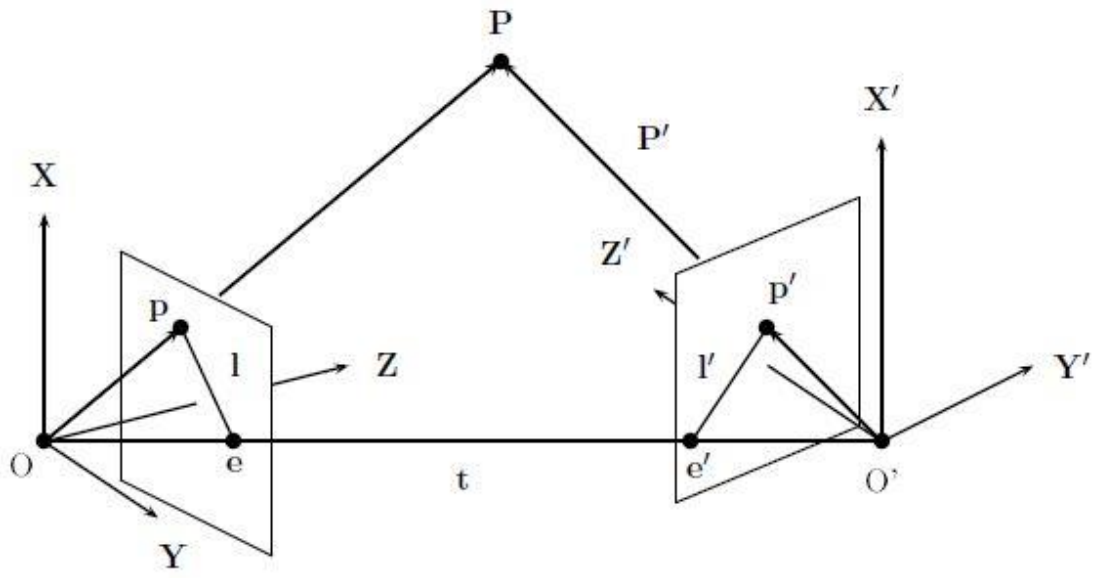
Ved å sette likningane lik kvarandre og sette Z aleine får vi:

$$\frac{\mathrm{Z}}{f}(x_l - x_r) = \boldsymbol{t} \rightarrow Z = \frac{f\boldsymbol{t}}{d} \tag{27}$$

Her er triangulering nytta for å finne differansen i bildeplana $d = x_l - x_r$, også kalla disparitet. Resultatet på dette er avhengig av løysinga av korrespondanseproblemet som blir forklart i kap. 2.2.3.

2.2.2 Epipolar geometri

For å kunne dra nytte av teorien som vart lagt fram i førre kapittel, er det nyttig å sjå på kamerariggen i 3D. Her er det viktig å merke seg at det er nytta normaliserte koordinatar, $\boldsymbol{p} = \hat{\boldsymbol{p}}$, som er gjennomgått i kap. 2.1.1. Figur 9 syner to kamera, **O** og **O**', som har ei ulik orientering i forhold til kvarandre med det venstre kamera, **O**, som referanse. Dei ser på det felles punktet i rommet **P**, der punktet blir representert som **p** i venstre bildeplan og **p'** i høgre bildeplan. Planet som blir danna av punkta **OO'P** blir kalla det epipolare planet og dannar ein trekant i rommet. Det vil da være ulike plan for ulike punkt i rommet. I kvart av bildeplana kan det trekkast ei linje **l,** eller **l'**, i frå punktet i bildet, langs bildeplanet og ned til baselinja. Punktet **e**, eller **e'**, der linja kryssa baselinja er kalla ein epipol, og linja mellom epipolen og punktet i bildet er den epipolare linja.



Figur 9 Her er to kamera med ein translasjon og rotasjon i forhold til kvarandre. Bildeplana er satt framfor det optiske senteret og det er dratt ei linje frå kvart optiske senter til eit felles punkt i rommet. Figuren er henta frå *[11]*.

Den essensielle matrisa

Det antakast vidare at translasjon -og rotasjonsmatrisene mellom kamera kjent. I tillegg så er bildekoordinatane normaliserte, som forklart i kap. 2.1.1. På bakgrunn av den epipolare avgrensinga så er dei tre vektorane $\overrightarrow{\boldsymbol{Op}}$, $\overrightarrow{\boldsymbol{O'p'}}$ og $\overrightarrow{\boldsymbol{OO'}}$, frå Figur 9, i same plan. Vidare så er vektorproduktet $\boldsymbol{t}\times\boldsymbol{Rp'}$ normal til det epipolare planet. Det indre produktet mellom denne vektoren og ein vilkårleg vektor i planet vil då være lik null. Ved å sjå nærmare på punktet **p** i venstre bildeplan blir følgande likning presentert:

$$\boldsymbol{p}^T * [\boldsymbol{t}\times\boldsymbol{Rp'}] = 0 \tag{28}$$

Ut frå denne likninga kan vi definere den essensielle matrisa:

$$\boldsymbol{\varepsilon} \stackrel{\text{def}}{=} \boldsymbol{t}\times\boldsymbol{R} \tag{29}$$

Denne matrisa er forklart her i si enkelheit. Den utrykker detaljar rundt kamera si orientering som translasjon og rotasjon og blir funnet ved hjelp av kalibrering. På bakgrunn av den epipolare avgrensinga og teorien som er blitt lagt fram, så blir den epipolare linja i høgre bildeplan berekna når eit punkt i venstre bildeplan er funnet. Søket er no forenkla frå 2D i heile bildet til 1D langs den epipolare linja. Dette er kalla den epipolare avgrensinga og er i denne samanhengen også kalla den essensielle avgrensinga.

Den fundamentale matrisa

Den fundamentale matrisa representarar koordinatar i pikslar. Denne relasjonen nyttar dei indre kameraparameterane i tillegg til den essensielle matrisa[1]. For å sjå på samanhengen mellom normaliserte -og pikselkoordinatar har vi, frå kap. 2.1.1, $\boldsymbol{p} = \boldsymbol{K}\widehat{\boldsymbol{p}}$ og $\boldsymbol{p}' = \boldsymbol{K}'\widehat{\boldsymbol{p}}'$ der variablar som er merka representera det andre kamera. Ved å snu på likningsetta og sette dei inn i likning (28) får vi:

$$\mathbf{p}\boldsymbol{K}^{-T}\boldsymbol{t}\times\boldsymbol{R}\boldsymbol{K}'^{-1}\boldsymbol{p}' = 0 \tag{30}$$

Ut frå denne likninga definera vi fundamentale matrisa:

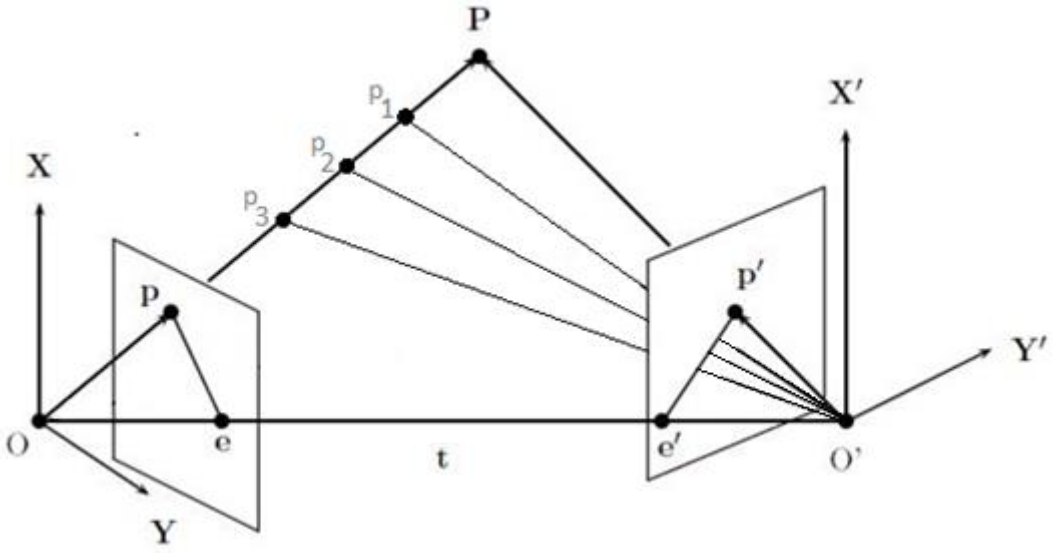
$$\boldsymbol{\mathcal{F}} \stackrel{\text{def}}{=} \boldsymbol{K}^{-T}\boldsymbol{t}\times\boldsymbol{R}\boldsymbol{K}'^{-1} \tag{31}$$

Denne matrisa er også forklart i sin enkelheit. Den utrykker detaljar rundt kamera si orientering som translasjon og rotasjon og blir funnet ved hjelp av kalibrering. Men denne relatera den epipolare avgrensinga i pikselkoordinatar til forskjell frå den essensielle som er i normaliserte rom-koordinatar.

2.2.3 Korrespondanseproblemet

Ved å utnytte dei geometriske eigenskapane til eit stereosystem er det råd å finne felles punkt i kvart av bilda og finne disparitet mellom ulike punkt. Dette er kalla korrespondanseproblemet. Ei løysing på problemstillinga er å nytte seg av teoremet om epipolar avgrensing. Figur 10 syner at punkt som er langs linja **OP** er no å finne i den epipolare linja $\boldsymbol{e}'\boldsymbol{p}'$. Likedan er punkt langs linja **O'P** råd å finne i den epipolare linja linja $\boldsymbol{ep}$. Dette minimera søkefeltet frå heile bilderamma, til eit søkefelt i ei fast linje frå kvart av kamera. Den epipolare avgrensinga blir karakterisert med den *essensielle* og den *fundamentale* matrisa, som er forklart i kap. 2.2.2.

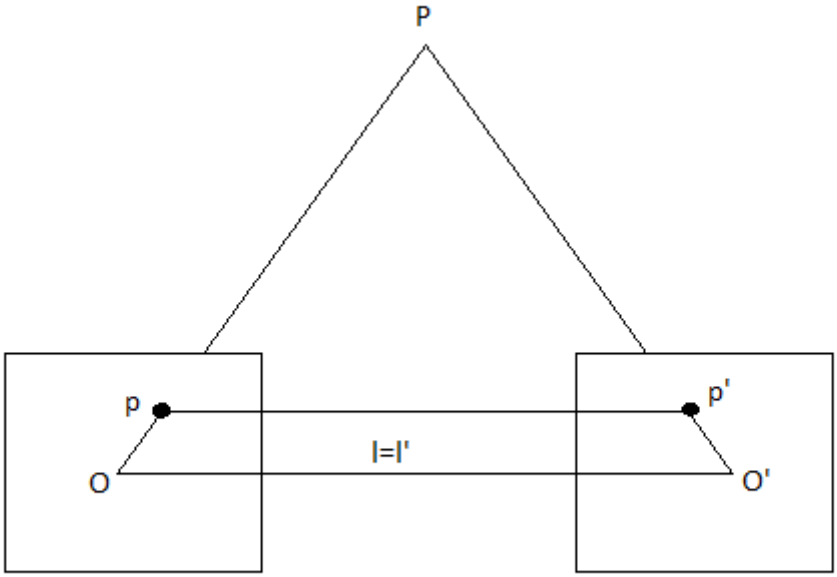
[1] Ved å nytte seg av den essensielle matrisa nyttar den seg også av dei ytre parameterane for rotasjon og translasjon.



Figur 10 Linja OP sine punkt er her lett å finne når søket er i den epipolare linja til det motsette kamera. Søkefeltet er no minimert til å være innanfor den epipolare linja e'p'.

Likeretting av bilde

For å forenkle utrekninga og arbeidet med å finne felles eigenskapar i stereobilde er det ein fordel å likerette bilda. Dette er ein operasjon der dei originale bilda blir erstatta av to bilde med felles bildeplan, og der dei epipolare linjene er parallelle med baselinja. No er dei felles eigenskapane horisontalt i bilda, og søket blir veldig forenkla. Figur 11 syner eit illustrert eksempel på to likeretta bilde.



Figur 11 Bilda er likeretta: Kvart bildeplan har den epipolare linja parallell med baselinja. Søket etter felles punkt i kvart bildeplan er no i x-retning.

Disparitet

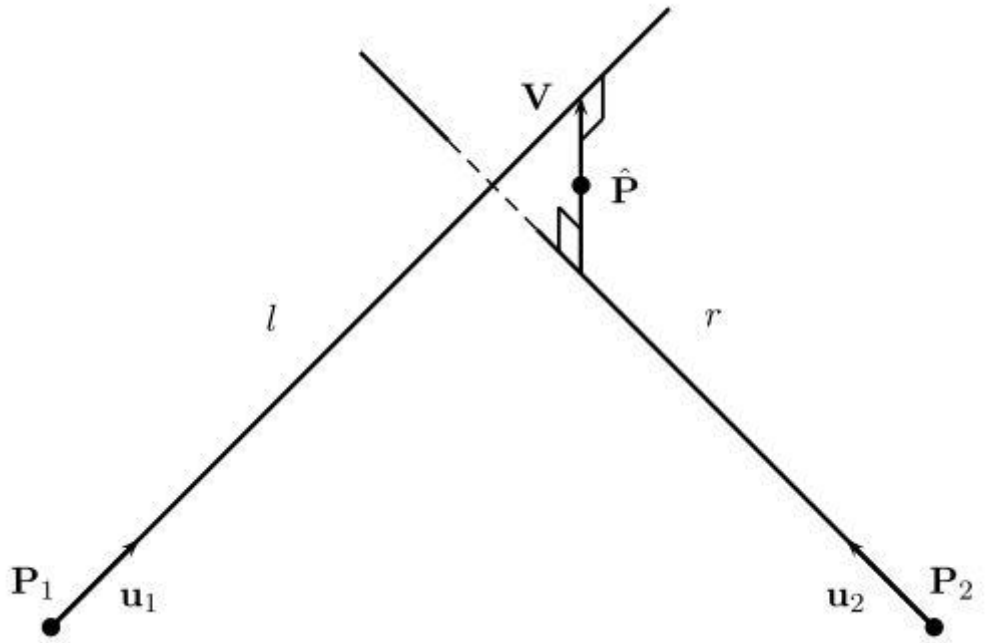
Når bilda er likeretta er neste skritt å finne dispariteten mellom kvart punkt. I kapittel 2.2.1 var det vist metoden for triangulering og dermed finne punkt mellom to kameraaksar. For å triangulere mot eit punkt i 3D-rommet vil det vise seg at 2D-modellen som er vist over ikkje er optimal sidan linjene frå kvart kamera aldri vil møtes. Dette blir forklart i kapittel 2.2.4 som omhandlar rekonstruksjonsproblemet. For å finne dispariteten mellom to bilde blir det nytta ulike kost-funksjonar for å finne like punkt i bilda. Dette blir sett nærmare på i kapittelet 2.3.3. Bildet nedanfor syner korleis disparitetskartet synleggjer forskjellar i avstand mellom dei ulike objekta.

Figur 12 Venstre: Originalbilde av ei scene med ulike køyretøy. Høgre: Disparitetskart over scena. Lys farge representera stor disparitet og kort avstand, mørk farge representera liten disparitet og større avstand. Figuren er henta frå [20].

2.2.4 Rekonstruksjonsproblemet

Når dei felles eigenskapane i dei ulike bilda er funne, er det naturleg å «rekonstruere» scena for å kunne estimere ein 3D-modell og deretter måle avstand til objekt samt størrelse av objekt. På bakgrunn av teorien som er gått gjennom i kap. 2.2.1-2.2.3, så har disparitetskartet informasjon som fortel noko om avstandar i scena. Det er her gått utifrå at kamera er kalibrert slik at dei indre og ytre parameterane er kjent. Då kan problemet bli løyst ved triangulering.

Rekonstruksjon ved triangulering
I prinsippet er det mogleg å rekonstruere scena ved å sjå på dei kryssande linjene frå Figur 10. Men i verkelegheita vil desse linjene aldri krysse grunna støy i dei ulike bildepunkta. Om dei indre og ytre parametrane er kjente, kan dette løysast ved triangulering.



Figur 13 Rekonstruksjon ved triangulering. Kvar linje frå kvart av kamera møtes i verkelegheita aldri, så det må estimerast ei beste løysing. Figuren er henta frå *[11]*.

Figur 13 syner to linjer frå to ulike kamera som kryssar kvarandre, men som ikkje treff same punkt. Dette kan løysast ved å finne den kortaste vegen mellom dei to linjene, og velje det punktet som er på midten av vektoren som skil dei to linjene. Dei to linjene, l og r, er definert av eit punkt og ein vektor. $\boldsymbol{P_1}, \boldsymbol{u_1}$ for l og $\boldsymbol{P_2}, \boldsymbol{u_2}$ for r. Alle punkt og vektorar har same koordinatsystem som referanse. For å finne minimal lengde mellom kvar av linjene må vektoren **V** vere ortogonal til kvar av linjene. Vektoren **V** er vidare definert ved:

$$\mathbf{V} = \boldsymbol{P}_1 + a_1\boldsymbol{u}_1 - (\boldsymbol{P}_2 + a_2\boldsymbol{u}_2) \tag{32}$$

Der a_1 og a_2 er to ukjente positive verdiar. Sidan vektoren V er satt til å vere ortogonal til linjene, kan vi nytte det indre produktet mellom vektoren og kvar av linjene der $\boldsymbol{V}^T\boldsymbol{u}_1 = 0$ og $\boldsymbol{V}^T\boldsymbol{u}_2 = 0$. Ved å sette inn **V** frå likning (32) i kvar av likningane:

$$a_1 - a_2\boldsymbol{u}_2^T\boldsymbol{u}_1 = (\boldsymbol{P}_2 - \boldsymbol{P}_1)^T\boldsymbol{u}_1 \tag{33}$$

$$a_1\boldsymbol{u}_1^T\boldsymbol{u}_2 - a_2 = (\boldsymbol{P}_2 - \boldsymbol{P}_1)^T\boldsymbol{u}_2 \tag{34}$$

Løysinga på kvar av likningane er:

$$a_1 = \alpha(\gamma_1 - \beta\gamma_2 \tag{35}$$

$$a_2 = \alpha(\gamma_1\beta - \gamma_2 \tag{36}$$

Der

$$\alpha = \frac{1}{1-\beta^2} \tag{37}$$

$$\beta = \boldsymbol{u}_1^T\boldsymbol{u}_2 \tag{38}$$

$$\gamma_1 = (\boldsymbol{P}_2 - \boldsymbol{P}_1)^T\boldsymbol{u}_1 \tag{39}$$

$$\gamma_2 = (\boldsymbol{P}_2 - \boldsymbol{P}_1)^T\boldsymbol{u}_2 \tag{40}$$

Det rekonstruerte midtpunktet $\widehat{\boldsymbol{P}}$ er då

$$\widehat{\mathbf{P}} = \frac{1}{2}(\boldsymbol{P}_1 + a_1\boldsymbol{u}_1 + (\boldsymbol{P}_2 + a_2\boldsymbol{u}_2)) \tag{41}$$

Jo større vektoren **V** er, dess større er avstanden mellom linjene. Ved å nytte ei terskling for å velje det rekonstruerte punktet, kan desse feila minimerast. Ved å velje punkt der $\|\boldsymbol{V}\|$ er mindre enn ein valt terskel, blir dei største avstandane valt vekk.

2.3 Bildebehandling

Det er her valt å ha eit eige kapittel for bildebehandling. Dette er operasjonar som blir brukt for å trekke ut ulik informasjon frå bileta. Frå eit kamera sitt perspektiv er det to hovudmetodar som vert nytta; korrelasjon -og eigenskapsbasert[10][21]. Korrelasjon er hovudsakleg nytta i stereobilde for å skape eit disparitetskart, og eigenskapspunkt er hovudsakleg nytta for enkle bilde for å finne dei sterkaste punkta i dei ulike scenene. Desse metodane nyttar ulike eigenskapar slik som hjørne eller kantdeteksjon. I dette kapittelet blir det først presentert nokre grunnleggande bildeoperasjonar som blir brukt i ulike algoritmar som blir presentert i kapittelet om eigenskapspunkt. Til slutt blir det gjennomgått metodar for å finne felles punkt i stereobilde.

2.3.1 Grunnleggande bildeoperasjonar

Det er her valt å sjå nærmare på nokre grunnleggande bildeoperasjonar. Dette er operasjonar som er brukt for å finne kantar, hjørne og eigenskapar i bilde samt forenkle utrekninga i nokre operasjonar. Nokre eigenskapar i bilde som er distinkte er kantar og hjørne. Intensiteten i bilde vil i desse områda ha store gradientar samt gradienten sin orientering vil ha store forandringar i små områder. Det er ulike algoritmar og metodar som nyttar seg av gradienten, og det vil her bli gått gjennom nokre av dei.

Gradienten for kantdeteksjon
Gradienten i bilde er ein vektor med to komponentar. For eit bilde I, er gradienten i x og y-retning

$$\nabla \mathbf{I} = (\boldsymbol{I}_x, \boldsymbol{I}_y)^T = (\frac{\partial \boldsymbol{I}}{\partial x}, \frac{\partial \boldsymbol{I}}{\partial y})^T \tag{42}$$

Sidan gradienten er ein vektor, har den også ei lengde

$$|\nabla I| = \sqrt{\boldsymbol{I}_x^2(m,n) + \boldsymbol{I}_y^2(m,n)} \tag{43}$$

og ein retning

$$\theta = \arctan \frac{\boldsymbol{I}_y}{\boldsymbol{I}_x} \tag{44}$$

Figur 14 Venstre: Originalbilde. Høgre: Gradientane til bildet er synleggjort.

Gradienten framhevar høgfrekvent støy, så det er vanleg å filtrere bildet med eit lineært gaussisk lågpassfilter før gradienten er funnen:

$$\boldsymbol{J}_\sigma = \nabla[\boldsymbol{G} * \boldsymbol{I}] = \nabla[\boldsymbol{G}] * \boldsymbol{I} \tag{45}$$

der

$$\mathbf{G} = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{46}$$

er den gaussiske funksjonen, og

$$\nabla \mathbf{G} = (\frac{\partial \boldsymbol{G}}{\partial x}, \frac{\partial \boldsymbol{G}}{\partial y})^T = [-x - y]\frac{1}{\sigma^3} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{47}$$

Det gaussiske filter er lineært, så det kan nyttast konvolusjon mellom bildet og den horisontale og vertikale deriverte av filteret slik at utrekninga kan bli gjort i ein operasjon.

Laplacian of Gaussian – LoG

Eit alternativ til å nytte gradienten (første ordens deriverte) for kant deteksjon, er å bruke Laplace-operatoren (andre ordens deriverte). Frå definisjonen om gradienten over har vi då

$$\boldsymbol{S}_\sigma = \nabla \cdot \boldsymbol{J}_\sigma = [\nabla^2 \boldsymbol{G}] * \boldsymbol{I} \tag{48}$$

Skalarproduktet mellom gradientoperatoren og gradienten er her kalla Laplace-operatoren. Vidare er da

$$\nabla^2 \boldsymbol{G}_\sigma = \frac{1}{\sigma^3}(2 - \frac{x^2 - y^2}{2\sigma^2}) e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{49}$$

kalla «Laplacian of Gaussian» filterkjerne. Den gaussiske funksjonen er lineær og separabel og kan da skrivast som

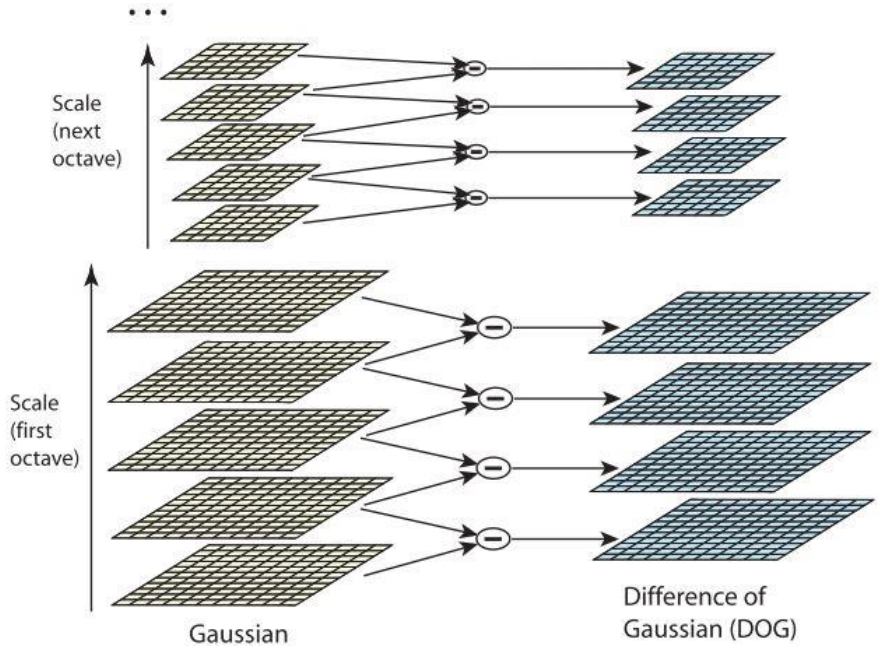
$$\nabla^2 \boldsymbol{G}_\sigma = \frac{1}{\sigma^3}\left(1 - \frac{x^2}{2\sigma^2}\right) G_\sigma(x) G_\sigma(y) + \frac{1}{\sigma^3}\left(1 - \frac{y^2}{2\sigma^2}\right) G_\sigma(y) G_\sigma(x) \tag{50}$$

LoG er nyttig ved ulik skala av bildet **I**, og er ofte nytta i samband med ein gaussisk pyramide for å få fram informasjon ved ulik skala. Dette blir gjort ved å skalere ned bildet og da få fram ulik informasjon ved å konvolvere kvart nedskalerte bilde med eit filter. Den gaussiske pyramiden er illustrert i Figur 25 i teorien om SURF-algoritmen.

Difference of Gaussian - DoG

I nokre tilfelle blir LoG erstatta med ei DoG-kjerne. Årsaka til at dette kan gjerast er at resultat av begge to er tilsvarande likt i frekvensplanet. Differansen mellom to lågpassfiltrerte bilde tilsvara bandpass-filteret som LoG representera i frekvensdomenet. Ved å nytte denne eigenskapen over fleire skale-ringar blir det konstruert ein gaussisk pyramide som framhevar informasjon ved ulik skalering, dette er illustrert i Figur 15. Likning (51) representerar DOG-kjerna:

$$\mathrm{DoG}\{\boldsymbol{I}; \sigma_1, \sigma_2\} = \boldsymbol{G}_{\sigma_1} * \boldsymbol{I} - \boldsymbol{G}_{\sigma_2} * \boldsymbol{I} = \left(\boldsymbol{G}_{\sigma_1} - \boldsymbol{G}_{\sigma_2}\right) * \boldsymbol{I} \tag{51}$$



Figur 15 DOG blir kalkulert ved å finne differansen mellom to gaussiske filter med ulik sigma. Dette blir gjort over ulike skaleringar. Figuren er henta frå [22].

Harris-hjørnedeteksjon

Harris er ei kjent og mykje brukt algoritme, og på same måte som for kantdeteksjon nyttar denne seg også av gradienten i eit bilde. Denne vart presentert av Chris Harris og Mike Stephens i 1988[23], og er framleis ei mykje brukt algoritme. Detektoren er basert på følgande matrise (Hessianmatrise) som er utrekna over eit område i bilde (ei kjerne) med pixlar $i\epsilon\{1,2,3 \dots I\}$:

$$\mathbf{H_H} = \sum_{i\epsilon I} (\nabla \mathbf{I})_i (\nabla \mathbf{I})_i^T = \sum_{i\epsilon I} \begin{bmatrix} I_{xi}^2 & I_{xi}I_{yi} \\ I_{xi}I_{yi} & I_{yi}^2 \end{bmatrix} \tag{52}$$

For denne matrisa er det mogleg å detektere om det er kantar eller hjørne i bilde. På bakgrunn av matrisa $\mathbf{H_H}$, blir eigenverdiane utrekna ved

$$\det(\mathbf{H_H}) - \mathrm{k}(\frac{trace(\mathbf{H_H})}{2})^2 \tag{53}$$

Der k er ein konstant som balansera mellom kant eller hjørneliknande eigenskapar. $\det(\mathbf{H_H}) = \lambda_1\lambda_2$ og $trace(\mathbf{H_H}) = \lambda_1 + \lambda_2$ der λ_1 og λ_2 er eigenverdiane til matrisa H. Denne matrisa blir analysert på vegne av eigenverdiane. Om området har ein homogen intensitet utan kantar vil matrisa $\mathbf{H_H}$ ha rank 0 og ikkje ha nokre eigenverdiar. Om området blir ført over ein kant vil matrisa ha rank 1 og derav vil ein av eigenverdiane vere nærme null. Om området derimot er over eit hjørnepunkt, så vil matrisa ha rank 2 og begge eigenverdiane vil vere ulik null. Fordelen med denne detektoren er at den er invariant for rotasjon og translasjon.

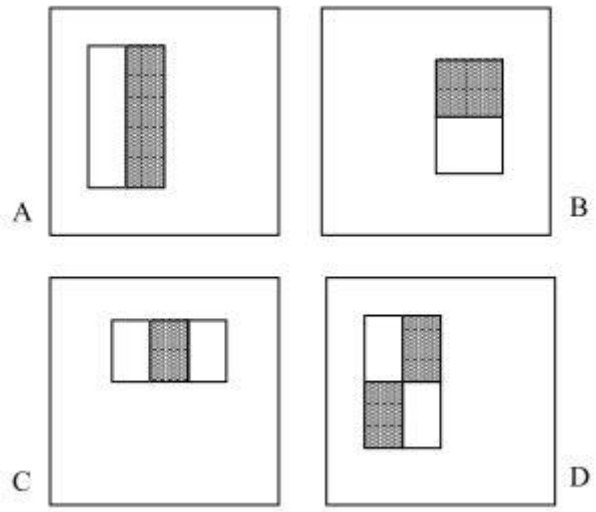
Haar eigenskapar

Haar eigenskapar for eit bilde er basert på Haar-wavelets. Dette er ein av dei enklaste typane av wavelets og er gitt av

$$\varphi(t) = \begin{cases} 1 \; for \; 0 \leq t < \frac{1}{2} \\ -1 \; for \; \frac{1}{2} \leq t < 1 \end{cases} \tag{54}$$

Dette skapar ein rektangulær funksjon og er ein endeleg funksjon. Frå signalbehandling er wavelets nytta for å observere signal i både tids -og frekvensdomenet og kan ofte, på same måte som for DoG, samanliknast med ein pyramide sidan bildet vert delt opp i filterbankar.

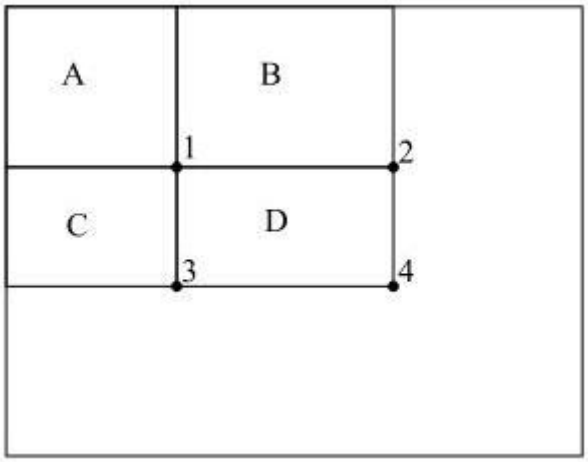
Constantine P. Papageorgiou, Michael Oren og Tomaso Poggio introduserte i 1998 eit rammeverk for objektdeteksjon som var basert på Haar-wavelet representasjon[24]. Paul Viola og Michael Jones tok dette vidare og var dei første som nytta Haar eigenskapar for sanntids ansiktsattkjenning i 2001[25]. I arbeidet til Viola og Jones blir resultatet av det å integrere ein wavelet med ein kernel kalla ein Haar-eigenskap. Denne eigenskapen har vist seg å vere kjapp å integrere i eit system ved hjelp av integralbilde som blir introdusert nedanfor, samtidig som den tydeleg framhevar eigenskapar på køyretøy. Figur 16 syner nokre eksempel på Haar-eigenskapar.



Figur 16 Fire ulike versjonar av rektangel eigenskapar som blir brukt i samband med Haar-eigenskapar. Summen av pixlane som er innanfor dei kvite rektangla blir trekt ifrå summen av pixlane i dei grå rektangla. A og B syner to rektangel. C syner tre rektangel og D syner fire rektangel. Figuren er henta frå [25]

Integralbilde

Dersom eit bilde skal bli filtrert med fleire ulike boksfilter med ulike størrelsar og ved ulike lokasjonar, kan algoritmen bli effektivisert ved hjelp av integralbilde. Viola et al. presenterte i samband med Haar eigenskapane ein metode som er kalla integralbilde, der dei har basert denne på «summed area table»[25]. Forfattarane har her valt å skilje mellom desse metodane sidan dei er brukt i ulike situasjonar. Ved å dele opp eit bilde i fleire rektangel blir integralbildet kalkulert ved $I_I(x,y) = \sum_{i=0}^{i<x}\sum_{j=0}^{j<y} I(i,j)$. Denne metoden minimerer ein del av kalkuleringa i algoritmen. Figur 17 syner korleis integralbildet i eit bilde blir estimert.



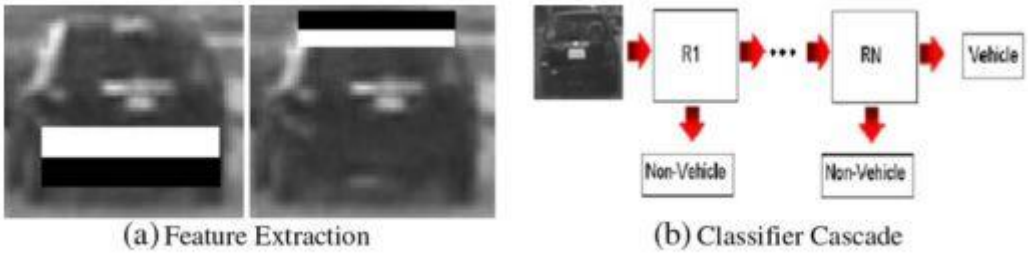
Figur 17 Summen av pixlane i rektangel D kan bli kalkulert med 4 referansar. Summen av integralbildet ved lokasjon 1 er summen av pixlane i rektangel A. Ved lokasjon 2 er summen A+B. Ved lokasjon 4 er summen A+B+C+D. Den totale summen i rektangel D er då lokasjon 4+1-(2+3). Figuren er henta frå [25].

2.3.2 Eigenskapspunkt

For å skilje og kjenne att objekt og punkt i bilde er det ulike metodar som er nytta. Eksempelvis så er det i litteraturen vanleg å finne hjørnepunkt i bilde sidan dei skil seg ut frå andre punkt. Ved å nytte kjente algoritmar, som til dømes Harris, er det mogleg å finne hjørne, men denne algoritmen er ikkje optimal når eit bilde vert skalert, noko som oppstår i ei skiftande scene. Vidare er dei mest brukte algoritmane for å finne eigenskapspunkt i ei trafikkscene forklart.

Viola Jones algoritmen (Haar-eigenskapar)
Haar-eigenskapar blei presentert av Paul Viola og Michael Jones for å detektere objekt i [25], og den er ofte kalla Viola Jones algoritmen. Denne algoritmen vart først presentert for ansiktsdeteksjon, men har blitt adaptert for å kunne brukast i køyretøydeteksjon.

Ved å nytte Haar-eigenskapane frå Figur 16 over eit bilde blir pikslane gitt binære 0 eller 1-verdiar. Det er seinare konstruert utvidingar av desse eigenskapane, med fleire alternative rektangel[26],[27]. Viola et al. nytta eigenskapane til Haar på områder som auger, sidan desse områda var distinkt mørkare enn resten av ansiktet. Eit køyretøy er distinkt i forhold til omgjevnaden og køyrebana og har klare firkanta former som blir framheva av Haar-eigenskapane. Figur 18 syner eit eksempel på arbeid gjort av Sayanan Sivaraman og Mohan Manubhai med Haar-eigenskapar for køyretøyattkjenning[28]. Ved å estimere vektene på kvar eigenskap ved hjelp av integralbilde, er denne algoritmen ganske rask. Noko som fleire av arbeida med køyretøydeteksjon understrekar, sjå kap. 3.4.1.



Figur 18 Haar-eigenskapar: a): Haar boksar er markert over område på køyretøy for å finne eigenskapar. b) Ved å gjere denne operasjonen i ein kaskade, blir fleire falsk-positive klassifiseringa utelukka. Figuren er henta frå [28].

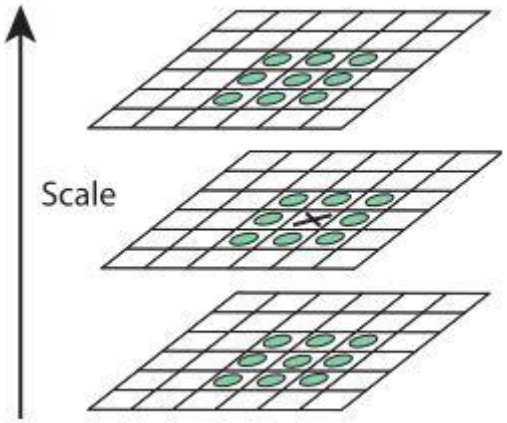
Ved å nytte seg av ei maskinlæringsalgoritme blir dei beste eigenskapane klassifiserte sanne, og falske positive blir kasta. I kapittel 2.4.2 blir det gått gjennom AdaBoost som vart nytta i denne algoritmen. Ved å utføre klassifiseringa i fleire kaskadar vil talet av rett klassifiserte eigenskapar stadig minke, og det vil til slutt bestå av dei sterkaste eigenskapane i eit bilde.

Haar-liknande eigenskapar er nyttige i køyretøydeteksjon sidan dei rektangulære eigenskapane er følsame for horisontale og vertikale kantar samt symmetriske strukturar som er på eit køyretøy. I tillegg er algoritmen rask, noko som er veldig nyttig i sanntids-applikasjonar. I kapittel 3.4.1 vil det bli presentert ulikt arbeid med Haar-eigenskapar.

Scale invariant feature transform (SIFT)

Denne algoritmen blei presentert av David Lowe i 2004[22], og den blir ofte samanlikna med ein kaskade av filter. Ein distinkt fordel med denne algoritmen er at den er robust i søket av punkt sjølv ved ulik skalering og rotering av bilde. Det vil sei at dei same punkta kan bli funne ved ulike syn på objektet. Dette er veldig nyttig i køyretøyattkjenning sidan scena forandrar seg jamt, og gjer bilde av køyretøy i ulike perspektiv. Bakdelen med algoritmen er at den er tung i ei utrekning og ikkje er ideell for ein sanntidsdeteksjon.

Algoritmen finn ekstrempunkt i bilda ved å nytte seg av «Difference-of-Gaussian», som er forklart i kap. 2.3.1. Ved å samanlikne dei ulike resultata av DoG-bilda blir det resultatet med mest tydeleg ekstrempunkt valt som eit eigenskapspunkt i bilde. Figur 19 syner eit eksempel på DoG bilde ved 3 ulike skaleringar.

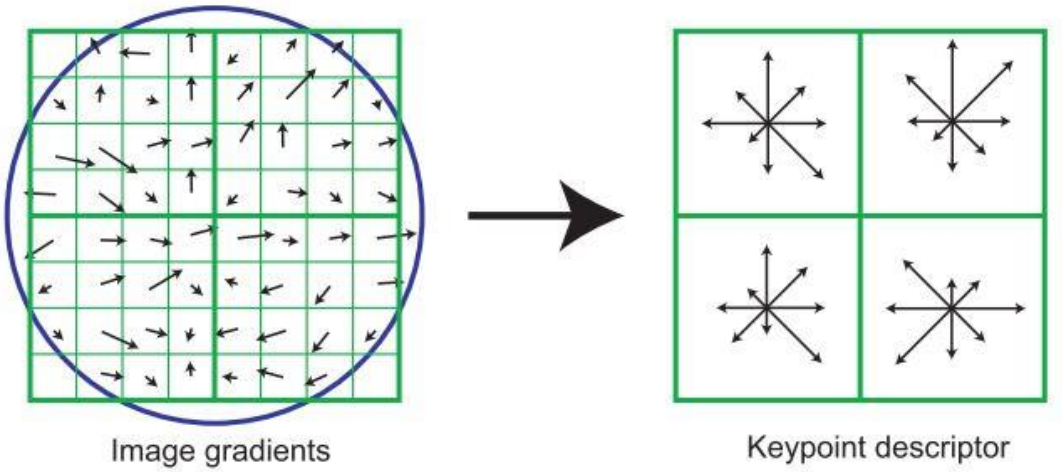


Figur 19 Maximum og minimum av DOG-bilde er detektert ved å samanlikne ein pixel (markert med x) med 26 nabo-pixlar i 3x3 regionar. Dette blir gjort i tre ulike skalaer. Figuren er henta frå [22].

Desse punkta blir no sortert i to steg. Først nyttar dei seg av Taylor-rekker for lokalisering av ekstrempunkt. Her blir punkt sortert vekk på bakgrunn av ein terskel-verdi slik at dei punkt med låg kontrast blir sortert ut. Sidan DOG er følsam for kantar, så er neste steg å nytte seg av ei 2x2 Hessianmatrise for å finne og luke ut kantar i bildet. Denne matrisa er forklart i kap. 2.3.1. Ved å også nytte seg av ei terskling her, blir fleire svake punkt fjerna. Resultatet etter denne filtreringa er dei sterkaste eigenskapspunkta.

Dei ulike eigenskapspunkta er no valt på bakgrunn av skala slik at berekningane blir utført ved ein skaleringsinvariant måte. Vidare blir gradienten og orienteringa i kvart punkt berekna ved å nytte differansen mellom pixlane. Denne informasjonen blir lagra i eit histogram som blir representert av 36 søyler som tilsvara orienteringa 360 grader rundt punktet. Ved å velje den høgste verdien i histogrammet, samt dei verdiane som er over 80% av den valte, blir orienteringa bestemt. Denne metoden fører til at kvart eigenskapspunkt har same skalering og orientering, men ulik retning.

Sidan eigenskapspunkt no er lokalisert blir deskriptoren definert ved å velje ei 16x16 ramme rundt punktet. Vidare blir desse delt inn i 4x4 blokker og området er da representert av 16 blokker. Kvar av blokkene blir vidare representer i eit histogram med 8 verdiar der kvar verdi er vektoren på gradienten. I kvar av blokkene blir vektorane akkumulert og representera samla gradientar i sine områder. Figur 20 syner eit eksempel der deskriptoren er valt frå ei 8x8 ramme.



Figur 20 Venstre: Deskriptoren er konstruert ved å estimere orientering og lengda på gradientane i eit område rundt eigenskapslokasjonen. Høgre: Gradientane er akkumulert frå eit histogram som vidare deler gradientane inn i mindre blokker. Gradientane representarar no summen av gradienten i si retning. Her er det viktig å merke seg at figuren syner ein 2x2 deskriptor som er kalkulert frå eit 8x8 vindauge. Figuren er henta frå[22].

Desse punkta beskriv SIFT-algoritmen i det store bildet. Ved å gje eigenskapspunkta ulike eigenskapar kan dei bli funne igjen ved ulik skala og rotasjon. Ved å samanlikne den minste eukledisk avstanden til naboane til deskriptorane i kvart bilde, blir dei like punkta funne. I denne artikkelen er det føreslått ein metode for å unngå registrering av feil naboar, som kan skje om dei er veldig nærme eller om det er støy i bildet. Ved å samanlikne avstanden mellom den nærmaste og den nest-nærmaste naboen brukar dei avstanden til den nest-næraste som ein indikasjon på feil detektert nabo.

Histogram of oriented gradient(HOG)

HOG er innspirert av den ovannemnte SIFT-algoritmen. Det blir i litteraturen referert til artikkelen til Navneet Dalal og Bill Triggs, der dei brukte den for å kjenne att menneske i bilde[29]. Denne algoritmen er mykje brukt for å skilje ut køyretøy i bilde, og blir her forklart utfrå den tidlegare nemnte artikkelen.

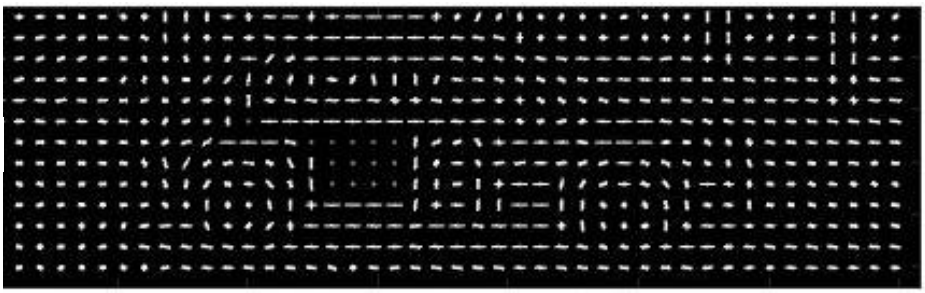
Første steg er å finne vertikale og horisontale gradientar i eit bilde, som forklart i kap. 2.3.1. Når gradientane i kvar retning er funnet, er det elementært å finne retning og storleik på den totale gradienten. For kvar pixel har no gradienten ein storleik og ein retning. For fargebilde blir det sett på gradienten i 3 kanalar, og den gjeldande gradienten er den som har størst storleik og derav blir også retninga valt utifrå den.

HOG-algoritmen ser vidare på lengda og vinkelen for gradienten til kvar piksel. Bildet blir delt opp i celler, der gradienten for kvar piksel i kvar celle er kalkulert. Her kan cellene eksempelvis vere 8x8 pikslar, der kvar piksel gjer ei vekta stemme for ein retningsbasert histogram-kanal. Vidare blir gradientane lagt i eit histogram med 9 søyler, der kvar søyle representera ei orientering på 20 grader og størrelsen representarar summert lengde på gradientane med den gitte orienteringa. Histogrammet representerar da ei orientering på 0-180 grader. Her er det viktig å merke seg at resultatet på 0-180 grader er valt på bakgrunn av ein «usignert gradient», altså gradientar i motsett retning blir sett like.

Ved å vidare normalisere kvart histogram blir effekten av lys og kontrastendringar redusert. Dette blir gjort ved å utvide blokkene til 16x16 slik at det no er 4 histogram for kvar blokk. Desse histogramma kan bli representert av ein 36x1 lang vektor som blir normalisert. For å gjere dette over heile bildet blir 16x16 blokkene flytta ei halv blokk vidare, det blir så konstruert ein ny vektor for kvar forflytning som blir normalisert. HOG-deskriptoren blir den samansette vektoren som blir funne ved å legge alle vektorane frå kvar blokk etter kvarandre til ein lang vektor.

a)

b)



Figur 21 HOG-deskriptor: a) Bildet som skal bli prosessert. b) Kvart av dei kvite objekta viser gradientens retning og lengde i ei 8x8 celle. Rundt køyretøyets kantar er det tydelege forskjellar i orienteringa til gradienten.

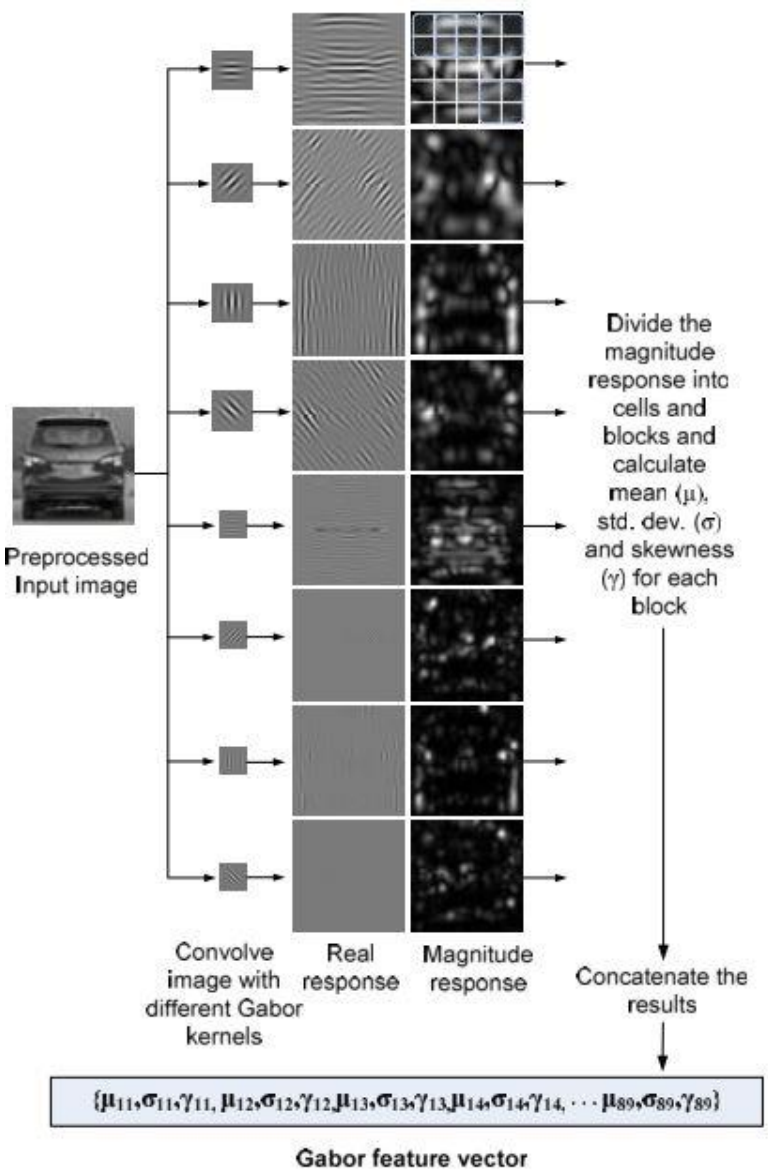
No vil gradientane syne storleik og retning på kontrastane i bilde, og ved kvasse overgangar blir objekt veldig tydelege i forhold til heilt matte bakgrunnar. Frå Figur 21 er det tydeleg at kontrastane frå køy-

retøyet blir framheva av gradientane. Her er kvart histogram frå kvar 8x8 celle visualisert som gradientane i den respektive cella. Denne algoritmen er mykje brukt saman med ulike klassifiseringsverktøy som blir sett på i kapittel 2.4.

Gabor eigenskapar

2D Gaborfilteret vart introdusert av John G. Daugman i 1985[30], og er velkjent i litteraturen for køyretøyattkjenning. Desse eigenskapane er, slik som Haar-eigenskapar, basert på wavelets.

Som forklart om Haar-wavelets, i kapittel 2.3.1, så er vanlegvis wavelets nytta i filterbankar for å få fram informasjon ved ulike frekvensar, og kan bli brukt som ei kjerne konvolvert med eit bilde for å få fram eigenskapar. På same måte er Gaborfilter konvolvert med bildet ved ulike skaleringar og orientasjonar, og ved å nytta prinsippet med cellehistogram frå HOG-eigenskapar kan det utreknast ein Gaborvektor som innehar informasjon om scena. Prinsippet bak denne metoden er illustrert i Figur 22, der Gabor eigenskapsvektoren blir brukt vidare for å klassifisere køyretøy.



Figur 22 Gaboreigenskapar: Frå venstre blir eit bilde konvolvert med eit Gaborfilter ved ulik orientering og skala. Ved å dele responsen inn i eit cellehistogram blir resultatet ein Gaborvektor som kan nyttast vidare i klassifisering. Figuren er henta frå [31].

Speeded up robust features (SURF)

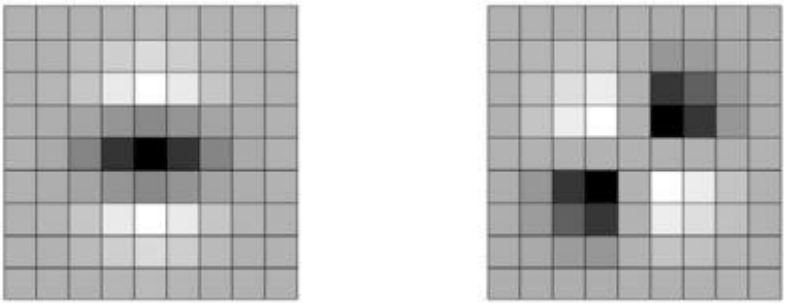
Denne algoritmen er i hovudsak basert på den ovannemnte SIFT algoritmen og vart presentert av Herbert Bay, Tinne Tuytelaars og Luc Van Gool I 2006 [32]. Dei hadde som mål å komme fram til ei raskare algoritme enn det som var tilgjengeleg, og har basert seg på robustheita bak SIFT-algoritmen. For å forbetre hastigheita på algoritmen introduserte dei integralbilde kombinert med ei tilnærming til Hessianmatrisa, som er forklart i kapittel 2.3.1. På bakgrunn av artikkelen blir det her summert opp hovudtrekka med SURF-algoritmen.

Algoritmen basera seg også på arbeid utført i [25] som igjen viser til integralbilde. Ved å dele opp eit bilde i fleire rektangel blir resultatet integralbilde, $I_I(x, y)$. Denne metoden minimera ein del av kalkuleringa i algoritmen.

For kvart punkt $\boldsymbol{X} = (x, y)$ er det ei Hessianmatrise $\mathbf{H}_{\boldsymbol{H}}(\boldsymbol{X}, \sigma)$ med ulik skalering:

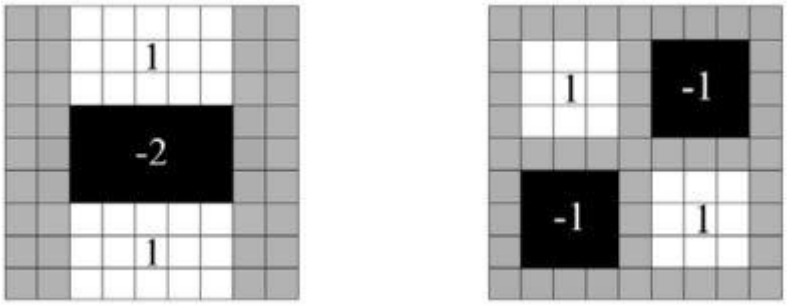
$$\mathbf{H}_{\boldsymbol{H}}(x, \sigma) = \begin{bmatrix} \boldsymbol{L}_{xx}(\boldsymbol{X}, \sigma) & \boldsymbol{L}_{xy}(\boldsymbol{X}, \sigma) \\ \boldsymbol{L}_{xy}(\boldsymbol{X}, \sigma) & \boldsymbol{L}_{yy}(\boldsymbol{X}, \sigma) \end{bmatrix}$$

Her er $\boldsymbol{L}_{xx}(x, \sigma)$ konvolusjonen mellom det gaussiske andreordens filteret $\frac{\partial^2}{\partial x^2} g(\sigma)$ og integralbildet i den same x-koordinaten. Dette er og tilsvarande for dei andre elementa i matrisa.



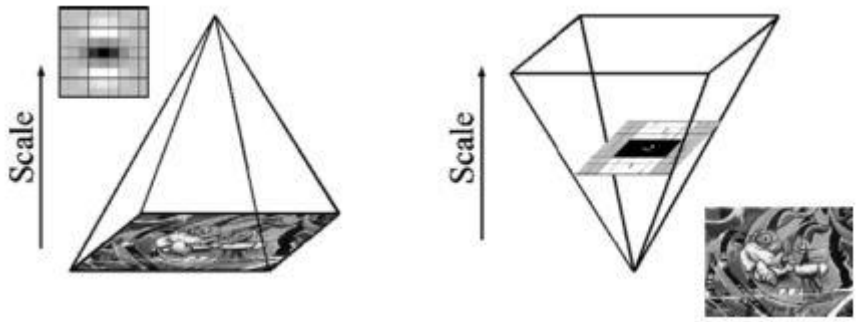
Figur 23 Venstre: Gaussisk 2. ordens deriverte maske i y-retning, L_{yy}. Høgre: Gaussisk 2. ordens deriverte maske i xy-retning, L_{xy}. Områder som er grå tilsvarar verdi 0, svarte fargar er -1 og kvite er 1. Figuren er henta frå [32].

Figur 23 syner den gaussiske 2. ordens deriverte maska som er diskretisert og klippt til. SURF brukar ei tilnærming av dette som er boksfilter, som vist i Figur 24. Konvolusjonen mellom bildet og filtra blir no effektivt berekna sidan det blir brukt boksfilter med integralbildet.



Figur 24 Tilnærming til boksfilter frå den gaussiske funksjonen. Her er boksfiltera 9x9 pixlar med σ=1,2 som representera den lågaste skalaen på filtera. Figuren er henta frå [32].

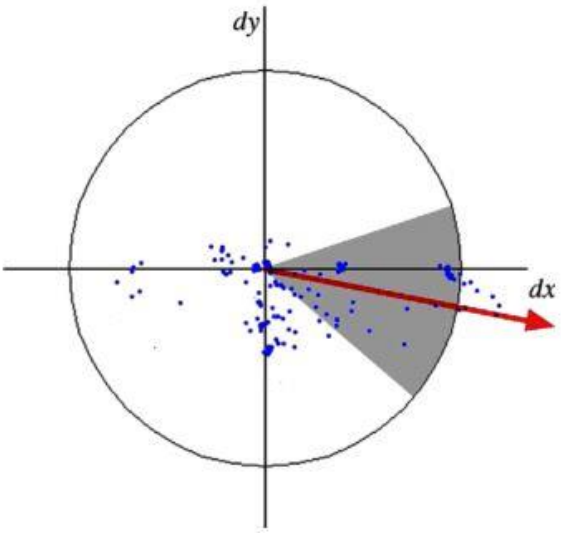
SURF søker gjennom ulike størrelsar ved å endre på dimensjonen av det gaussiske filteret. Vanlegvis blir det nytta ein gaussisk pyramide for bildet slik at det kan filtrerast ved ulike skaleringer, men SURF snur opp ned på dette og skalerer opp filteret i staden. Grunnen for dette er bruken av boksfiltre med integralbilde som gjer ei effektiv utrekning.



Figur 25 Venstre: Gaussisk pyramide som endra skalaen ved å nedsample bildet med 2 for kvar oktav. Høgre: Surf sin metode som endrar filterstørrelsen for kvar oktav. Figuren er henta frå [33].

For å lokalisere eigenskapspunkt over dei ulike skalaene nyttar algoritmen seg av «non-maximum suppression» i eit område for å trekke fram ein lokal maks-verdi. Deretter blir maksimalverdien av Hessianmatrisa sin determinant interpolert i forhold til skala og «image-space» med ein metode som var introdusert av Brown og Lowe[34].

Ved å nytte seg av Haar-wavelet responsar over x og- y retning, som forklart i kapittel 2.3.1, blir summen av responsane ein vektor som representarar orientering for kvart punkt. Dette blir gjort for kvar skala. Figur 26 syner den dominante orienteringa som blir representert av ein vektor.



Figur 26 Dei ulike orienteringane er lagra i eit sektorvindauge. Dette rotera med vinkelen og har ei utstrekning på $\frac{\pi}{3}$. Figuren er henta frå [33].

Deskriptoren blir også generert ved å nytte Haar-wavelets over integralbilde for raskare kalkulering. No bestemmast eit firkanta område rundt kvart punkt som er orientert på bakgrunn av førre steg. Dette området er vidare delt opp i 4x4 blokker for å dele opp informasjonen. Dei har her satt Haar-wavelet i x-retning til d_x og i y-retning d_y der dei ulike retningane er i forhold til eigenskapspunktet sin orientering. For å få informasjon om polariteten for intensiteten i punktet nyttar dei seg av summen av absolutt verdiane, $|d_x|$ og $|d_y|$. Med desse faktorane så dannar dei seg ein 4-dimensjonal vektor for kvart punkt: $v = (\Sigma d_x, \Sigma d_y, \Sigma|d_x|, \Sigma|d_x|)$. Denne vektoren representera no totalt 64 dimensjonar for kvar region.

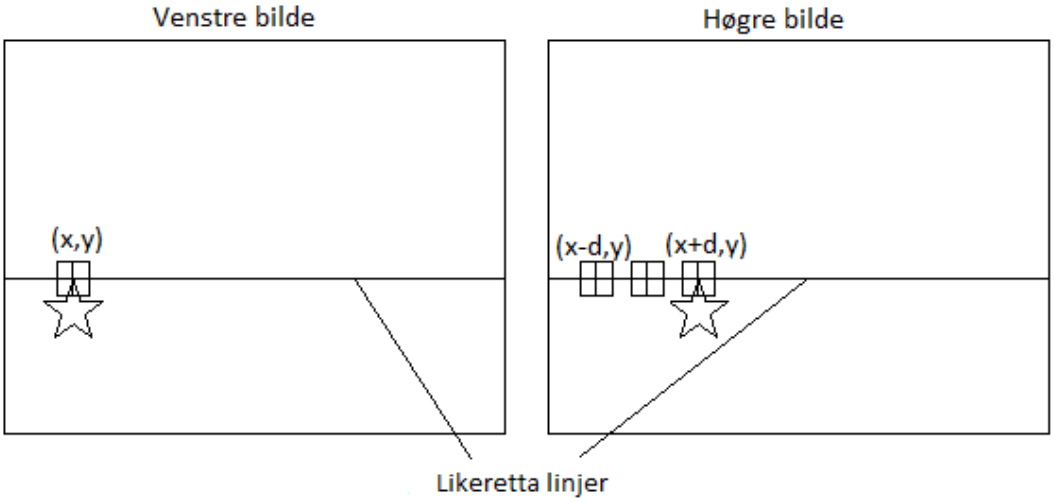
Vidare så samanliknar SURF kvart eigenskapspunkt i eit område slik som SIFT. Hovudforskjellen her er den lettare kalkuleringa som SURF presentera ved hjelp av integralbilde.

2.3.3 Korrelasjonspunkt

I stereosyn er det nødvendig å finne felles punkt mellom kvart av kamerabilda. Som nemnt i kapittel 2.2 så kan dispariteten nyttast til å finne avstanden til objekt frå kamerarigg. Dei første løysingane på problemet var delt i to kategoriar; lokale -og globale metodar[35]. Her viste det seg at dei globale metodane klart leverte betre resultat enn dei lokale, men krevjar desto tyngre utrekning. Som eit alternativ blei den semiglobale metoden presentert som er ein mellomting mellom dei to nemnt over [36][37]. Den er nesten like presis som den globale, og er så kjapp at den kan kalkulerast i ein sanntids applikasjon[5]. Dette er tre hovudmetodar for søk etter disparitet, og i køyretøydeteksjon er fokuset på kjappe algoritmar ein viktig del. Ein nyare studie har sett på kva algoritmar som er akseptable for ein sanntids implementering[38]. Studiet visar til ein total på 184 ulike algoritmar for stereosyn. I dette kapittelet blir det vist dei hovudsaklege ulikheitene mellom lokale, semi-globale og globale metodar der målet er å komme fram til eit disparitetskart.

Lokal metode

Denne metoden nyttar seg av ei blokk langs dei likeretta epipolare linjene, som er forklart i kapittel 2.2.2. Figur 27 viser søket mellom bilda. Når eit punkt er valt i venstre bilde så er søket etter det tilsvarande punktet langs den epipolare linja i det høgre bildet. Ved å nytte seg av ulike kost-funksjonar blir den blokka i det høgre bilde med minst forskjell frå første bilde valt. Dette blir i litteraturen referert som «vinnaren-tar-alt».



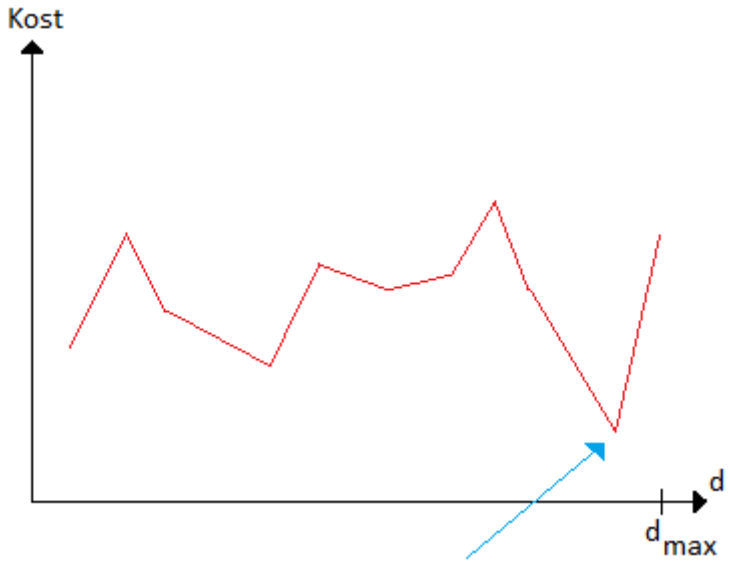
Figur 27 Lokal metode: Punktet (x,y) i venstre bildeplan blir funne igjen ved punktet (x+d,y) i høgre bildeplan ved å samanlikne ei satt blokk frå første punkt med blokker langs den epipolare linja i det andre bildeplanet. Den blokka med den minste kostnaden blir valt som felles punkt.

Denne metoden er i litteraturen nytta mange ulike kostfunksjonar på. Frå Figur 27 har er det eit punkt i venstre bilde som er innkapsla i ei blokk $f(x,y)$ og i det andre bildet er det ei tilsvarande blokk $g(x,y)$ som blir ført langs den epipolare linja. Nokre eksempel på ulike kostfunksjonar er:

$$\mathrm{SAD} = \sum_x \sum_y |f(x,y) - g(x,y)| \tag{55}$$

$$\mathrm{SSD} = \sum_x \sum_y (f(x,y) - g(x,y))^2 \tag{56}$$

Ved å nytte «vinner tar alt» strategien blir det vindauget med lågast kost valt som disparitetspunkt.



Figur 28: Vinnaren tar alt: Det punktet med lågast kost tilsvara punktet som er tilsvarande for begge bildar.

Semi-global metode

Denne metoden var først presentert av Heiko Hirschmuller i 2005[37]. Den har seinare blitt utvikla og testa med ulike kostfunksjonar[36][39], men vil her bli beskriven slik som han la den fram først. Formålet med denne algoritmen er å minimere ein global 2D energi funksjon $E(D)$ ved å løyse fleire 1D minimeringsproblem. Rapporten presentera følgande likning:

$$\mathrm{E(D)} = \sum_p C(p, D_p) + \sum_{q \epsilon N_p} P_1 T(|D_p - D_q| = 1) + \sum_{q \epsilon N_p} P_2 T(|D_p - D_q| > 1) \tag{57}$$

Den første sumeringa frå likning (57) er kosten over eit angitt vindauge langs den epipolare linja. Den andre summasjonen i likninga brukar P_1 som ein faktor dersom funksjonen $T(|D_p - D_q| = 1)$ er sann, altså den straffar små disparitetsforskjellar med ein kost P_1. Den siste sumasjonen påverkar kun store forskjellar i disparitetskartet.

Global metode

Globale algoritmar minimera også ein global energifunksjon, og nyttar ein global glattefunksjon for å tilnærme den samla kosten for dispariteten. Målet er å finne ein disparitetsfunksjon *d* som minimera ein global energi E(d):

$$\mathrm{E(d)} = E_d(d) + \lambda E_s(d) \tag{58}$$

$E_d(d)$ målar kor bra disparitetsfunksjonen d stemmer med dei to stereobilda. Om disparitetskartet er definert ved $M(x, y, d)$, vil da

$$E_d(d) = \sum_{(x,y)} C(x, y, d(x, y)) \tag{59}$$

der C er kosten over disparitetskartet.

$E_s(d)$ er glattefunksjonen i algoritmen. Som regel er denne avgrensa til å måle forskjellane mellom dispariteten mellom nabopixlane:

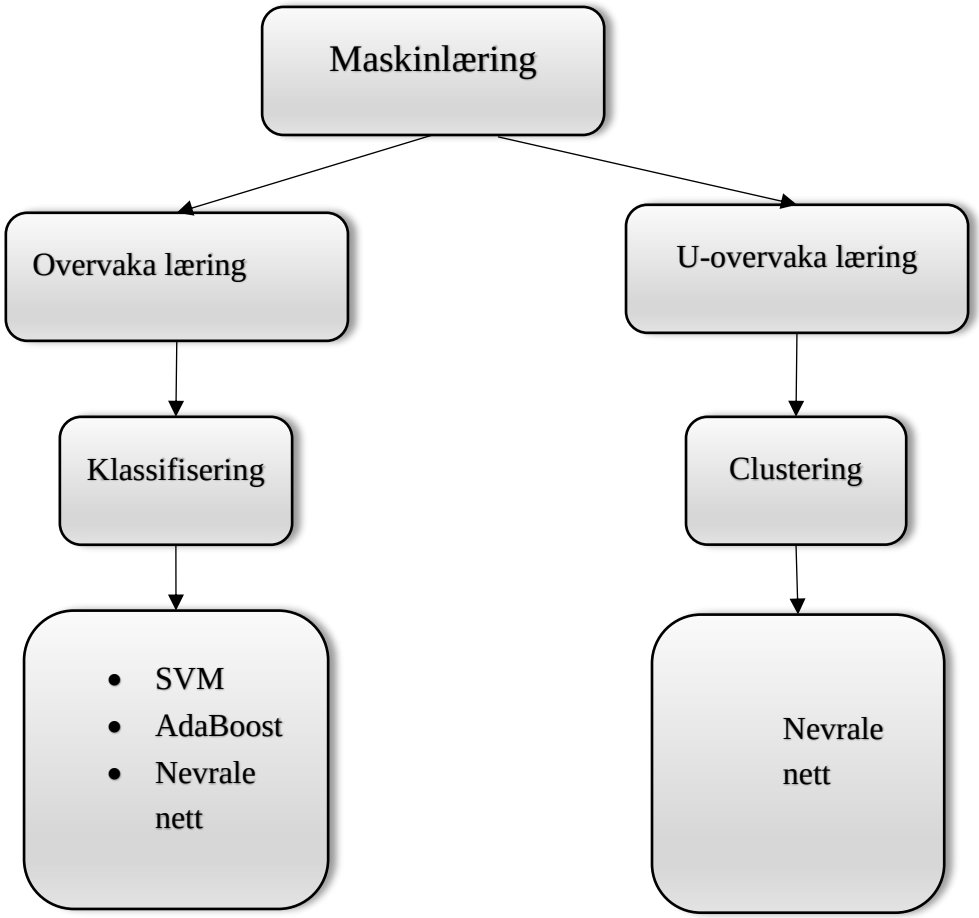
$$E_s(d) = \sum_{(x,y)} \rho(d(x,y)) - d(x+1,y) + \rho\big(d(x,y)\big) - d(x,y+1) \tag{60}$$

der ρ er ein aukande funksjon for forskjellane mellom dispariteten.

2.4 Klassifiserarar

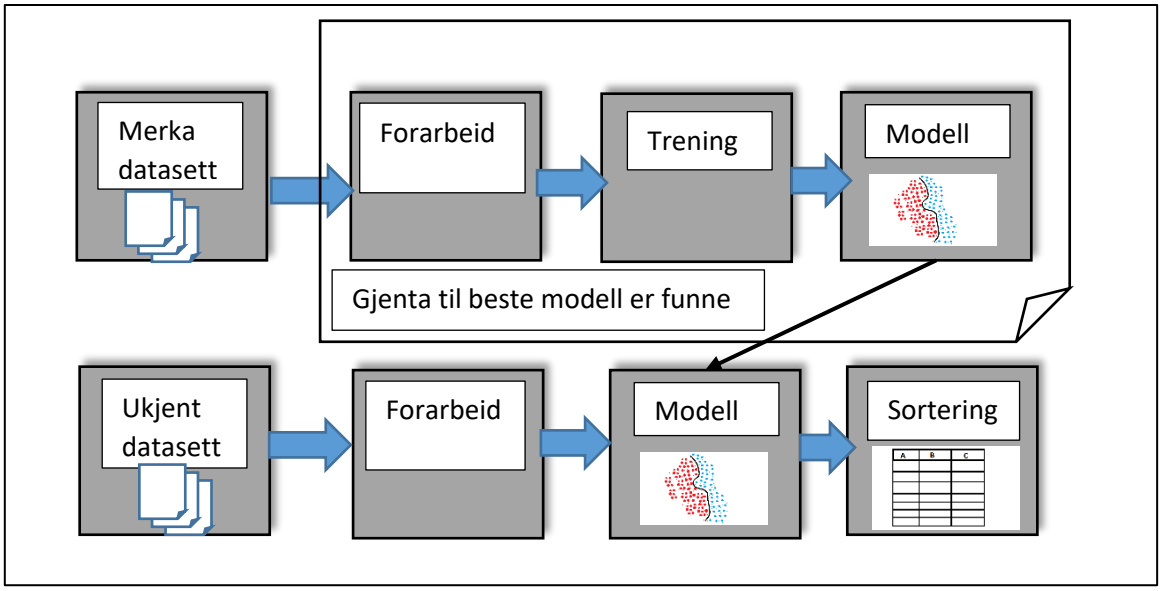
Når eit område i scena er detektert som eit objekt må det klassifiserast som køyretøy eller ikkje køyretøy. Gjennom åra har ulike klassifiseringsverktøy blitt brukt, og det er ikkje sjølvsagt kva verktøy som passar best til dei ulike problemstillingane. I litteraturen rundt køyretøyattkjenning er tre maskinlæringsalgoritmar mest nemnt; SVM, Boosting og Nevrale nett. Dette kapittelet gjer eit teoretisk innsyn i kvar av desse.

Før kvar av desse verktøya blir forklart er det greitt å gå gjennom litt teori om maskinlæring. Maskinlæring kan bli delt opp i to hovuddelar; overvaka -og u-overvaka læring. Desse utrykka fortel noko om måten dei ulike algoritmane blir trena for å klassifisere eller sortere eit gitt datasett, og for køyretøyattkjenning er det hovudsakleg overvaka læring som blir nytta for å klassifisere gitte datasett.



Figur 29 Maskinlæring blir delt opp i to hovuddelar, overvaka og u-overvaka.

Overvaka læring krevjar at dei ulike datasetta sine føretrekte inngangsverdiar er merka med dei ønska klassane (køyretøy eller ikkje køyretøy), og konstruera deretter ein modell basert på desse parameterane. U-overvaka læring lagar ein modell ved å sjå på datasett utan noko form for merking, og vil deretter skape ein modell som passar datasettet best ved å eksempelvis «clustre» ulike datapunkt. Desse konstruerte modellane vil da kjenne att delar i nye datasett og sortere informasjonen basert på tidlegare trening. Avhengig av kva type klassifiserer som er valt, så er det ulike metodar og ulike mengde med data som trengs.



Figur 30 Overvaka læring: Steg 1 er å gjere eit forarbeid på eit merka datasettet. Prefiltrering og eigenskapsutreking er blant anna arbeid som vert gjort her. Vidare vil den valte maskinlæringsalgoritmen trene opp ein modell som kan klassifisere datasettet. Ved iterasjon blir den optimale modellen funnen. Modellen kan vidare bli brukt i ulike applikasjonar for å sortere og klassifisere ukjente datasett.

Figur 30 syner eit enkelt eksempel på hovudtanken bak overvaka maskinlæring. Om datasettet frå figuren innehar nokre bilde av køyretøy frå ei scene i trafikken, så må desse bli pre-prosessert for å trekke fram eigenskapar frå bileta. Vidare er dei ulike bileta i datasettet manuelt merka som køyretøy eller ikkje-køyretøy og ein modell blir generert ved trening. Denne prosessen må gjentakast til modellen er best mogleg. Når modellen passar datasettet best mogleg, så kan den brukast til å klassifisere eit ukjent datasett.

2.4.1 Supportvektormaskin

Supportvektormaskin (SVM) er ein algoritme som lenge har hatt ei favorisering for køyretøyattkjenning. Sjølve idéen bak SVM algoritmen vart presentert av Corinna Cortes og Vladimir Vapnik på 90-talet[40], og har sidan den gang vorte presentert ulike versjonar av[41].

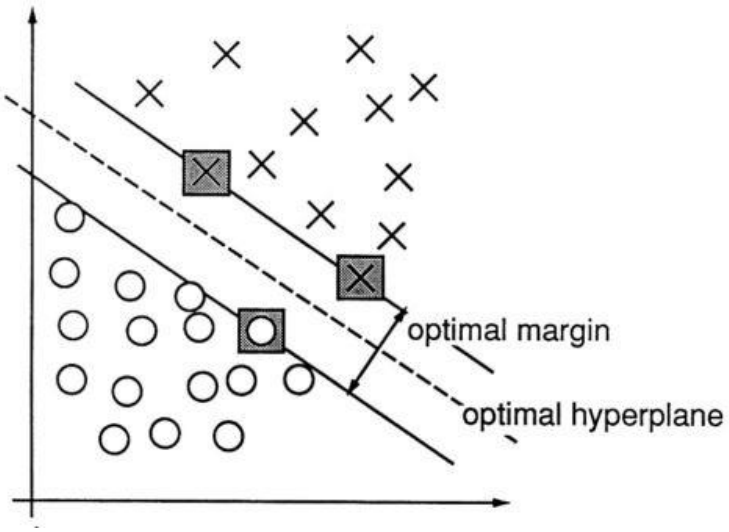
Frå kapittel 2.3.2 var det presentert ulike metodar for å finne eigenskapsvektorar frå eit bilde. Prinsippet med SVM er å skilje mellom desse vektorane med eit hyperplan. Figur 31 syner eit eksempel på dette i det 2-dimensjonale planet. Her vert hyperplanet valt på bakgrunn av ein maksimal margin mellom hyperplanet og dei næraste vektorane frå kvar klasse. Desse vektorane har fått tilnamnet supportvektorar. Det optimale lineære hyperplanet er markert med ei stipla linje der eit punkt på denne er gitt ved den lineære likninga:

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0 \tag{61}$$

der **w** er ein vektor normal til hyperplanet og b er bias frå origo. For eit sett med punkt blir det lineære hyperplanet gitt ved skalarproduktet mellom normalvektoren og punkt i rommet:

$$f(\boldsymbol{X}) = \boldsymbol{w}^T\boldsymbol{X} + b \tag{62}$$

Dei ulike klassane blir no skilde ved å sette $\boldsymbol{w}^T\boldsymbol{X} + b < 0$ til ein klasse, og $\boldsymbol{w}^T\boldsymbol{X} + b > 0$ til ein anna klasse.



Figur 31 Eit eksempel på eit klassifiseringsproblem i 2D-planet. Vektorane som er merka med grå firkantar definera marginen av den største avstanden mellom dei to klassane, og har fått tilnamnet supportvektorar. Figuren er henta frå [40].

For å kunne løyse u-lineære klassifiseringsproblem nyttar SVM seg av ulike kjernar i høgre dimensjonar[42]. Dei ulike kjernane tilsvara det å erstatte skalarproduktet med ein ulineær funksjon, og på den måten flytte datasettet til ein høgre dimensjon. Det er då mogleg å skilje mellom dei ulike klassane ved ei lineær tilnærming.

2.4.2 Boosting

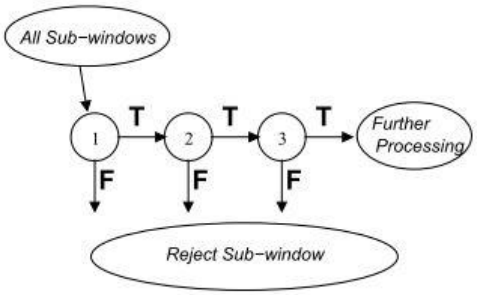
Boosting er ein generell metode som forsøker å forbetre presisjonen i ein læringsalgoritme, der AdaBoost er ein læringsalgoritme som basera seg på dette. Algoritmen AdaBoost vart presentert i 1995 av Yoav Freund og Robert E. Schapire [43], og er blant anna brukt i den kjente algoritmen for ansiktsattkjenning av Viola et al. [25].

Ved å nytte seg av ei generell maskinlæringsalgoritme blir dei beste eigenskapane klassifiserte sanne, og falske positive blir kasta. Den svake klassifiseraren $h_j(x)$ kan eksempelvis bestå av følgande element:

- Eigenskap f_j
- Terskling θ_j
- Polaritet p_j

$$h_j(x) = \begin{cases} 1\,, dersom\ p_j f_j(x) < p_j \theta_j \\ 0\,, \qquad\qquad\qquad ellers \end{cases}$$

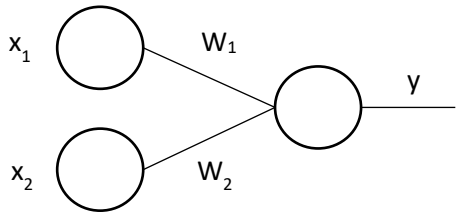
For å vidare auke hastigheit og presisjon på detektoren er det vanleg å nytte ein kaskade av klassifiserarar. Dette er blant anna nytta i algoritmen til Viola et al. og er vist i Figur 32.



Figur 32 Kaskade av klassifiserarar: T representera sanne klassifiseringar og F representera falske. Vektene på dei ulike parameterane blir justert for kvar klassifisering, og dei sterkaste punkta vil da bli med vidare. Figuren er henta frå [25].

2.4.3 Nevrale Nett

I litteraturen for maskinlæring så er nevrale nett (NN) ein måte å estimere matematiske funksjonar basert på den menneskelege hjernens oppbygging. Slike nettverk er satt saman av nevronar som er i kontakt med andre nevronar som igjen blir påverka av ein vekta inngang. Figur 33 syner eit eksempel på eit slikt nettverk i ein liten skala.



Figur 33 Nevralt nettverk med to inngangs-nevronar og ein vekta utgang. Her er x_1 og x_2 inngangsverdiar, w_1 og w_2 er ein vekta verdi til kvar inngang og y er utgangsverdi.

Nettverket frå Figur 33 tar dei to inngangsverdiane, x1 og x2, og vekta dei med w1 g w2, respektivt, til utgangen y. Utgangen er utrekna ved $y = g(x_1 w_1 + x_2 w_2)$ der funksjonen $g(\cdot)$ er ein aktiveringsfunksjon som setter utgangen til ein verdi (ofte 0 eller 1) avhengig av inngangane og vektene. Det er fleire typar aktiveringsfunksjonar å velje mellom, der den tradisjonelle er sigmoid funksjonen. NN må gjere dette i fleire syklusar på treningsdata for å trene seg opp til å kunne klassifisere ukjente datasett. Ved å nytte ein kostfunksjon vil dei ulike vektene forandre seg ved neste treningssyklus, og klassifiseraren vil konvergere mot ein modell.

Utviklinga av NN har dei siste åre fått eit større fokus. Frå konvensjonelle nevrale nett har fokuset no retta seg mot djupe nevrale nett (DNN) med fleire lag av nevronar. Eit omfattande studie utført av Jurgen Schmidhuber syner utviklinga av NN frå 1940-talet fram til dagens DNN[44]. Her blir det blant anna diskutert ImageNet sine konkurransar som har presentert gode resultat med djupe nett og nye aktiveringsfunksjonar. Eksempelvis så vann Alex Krizhevsky, Ilya Sutskever og Geoffrey E. Hinton konkuransen i 2012 med Alexnet som er eit firelags NN med ein ReLu aktiveringsfunksjon[45]. Bakdelen med denne type klassifiserarar er at dei har behov for store mengder med treningsdata for å kunne konvergere mot gode resultat.

3 Tidlegare arbeid

Utvikling av sjølvstyrte bilar er på full fart framover. Eksempelvis så nyttar Tesla seg av sine kommersielle bilar for å samle inn data og samtidig teste ut programvare og sensorar. Dette gjer tilgang på enorme mengder data som blir prosessert av multikjerne grafikkort ved hjelp av djupe neurale nett. Fabrikantar er uansett naudsynte til å halde køyretøya på eit lågt nivå av autonomi grunna vegtrafikklovene rundt om i dei ulike landa. Men nokre firma som Google og Uber testar sine fullt autonome bilar i områder med særskilt personell, slik som ingeniørar, bak rattet i trafikken. I autonome bilar er objektattkjenning basert på eit samarbeid mellom ulike sensorar som radar, lidar og kamera. Figur 34 syner eit bilde frå ein video frå heimesida til Tesla der dei demonstrera ein autonom Tesla modell S. Dei må her ha ein person bak rattet sidan regelverket tilseier dette.

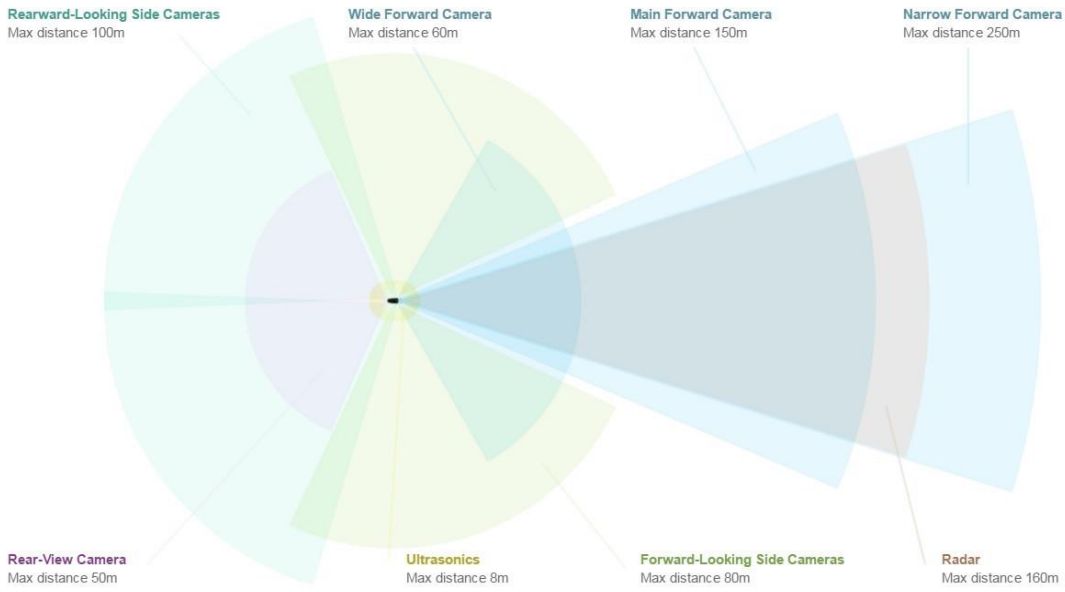
Figur 34 Demonstrasjon av Tesla Autopilot: Venstre: Det er ein person bak rattet sidan regelverket tilseier at det er nødvendig, det var ikkje naudsynt å gripe inn under denne testen. Øverst til høgre: Eit køyretøy er detektert bak Teslaen til venstre. I midten til høgre: Fleire køyretøy er detektert (i blått), og køyretøy som er i same vegbane som Teslaen er registrert som grønt. Nedst til høgre: Eit køyretøy er detektert bak Teslaen til høgre. Bildet er ein skjermdump frå Tesla sine heimesider.

Autonome køyretøy er delt inn i 6 nivå på bakgrunn av Society of Automotive Engineers (SAE) si utarbeiding[46]. Figur 35 syner ei oversikt over nivå som dei kan delast inn i der nivå 0 har ingen automasjon og nivå 5 er fullt autonome køyretøy. Når overvaking av miljøet rundt køyretøyet ikkje er utført av bilføraren lenger, er systemet definert som betinga automatisert (nivå 3).

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	No Automation	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	Conditional Automation	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	High Automation	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	Full Automation	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Figur 35 SAE har delt autonome køyretøy inn i 6 kategoriar. 0 er ingen automasjon og 5 er fullt autonome bilar. Tabellen er henta frå [46].

I denne rapporten blir sensorane delt inn i to grupper, aktive og passive sensorar. Aktive sensorar blir definert som sensorar som avgjer signal og målar tida signala brukar frå sensoren og tilbake frå det reflekterte objektet. Passive sensorar avgjer ingen signal, men målar omgjevnaden ved å ta imot lys og lyd som kjem frå omgivnaden. Figur 36 syner korleis Tesla har løyst problemstillinga med sensorar rundt bilen.

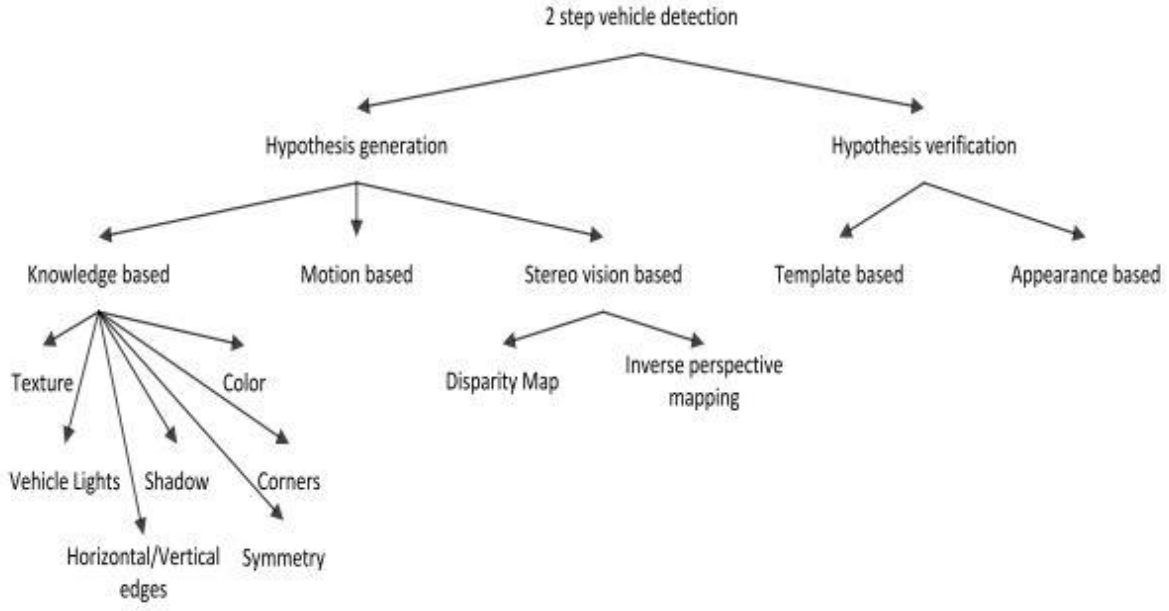


Figur 36 Tesla sine ulike sensorar for å overvake miljøet rund køyretøyet Figuren er henta frå Tesla si heimeside.

Dette kapittelet går inn på kvar enkelt av dei ovannemnte sensorane og metodar som er brukt for attkjenning av objekt i ei scene. Fokuset er ei samanlikning av dei ulike sensorane for å dra ut fordelar og bakdelar med kvar av dei.

3.1 Køyretøyattkjenning i fleire steg

I litteraturen blir køyretøyattkjenning delt inn i fleire steg. Sun et al. presenterte i 2006 to steg der første steg er hypotesegenerering (HG) og steg to er hypoteseverifisering (HV)[4].



Figur 37. Køyretøyattkjenning i to hovudsteg; HG for deteksjon av objekt og generere område av interesse, HV for attkjenning av objekt som køyretøy. Figuren er henta frå *[7]*

Hypotesegenerering(HG)
I HG steget er målet å finne aktuelle kandidatar i ei scene for å avgrense området i bilde. Sun et al. delar HG vidare inn i dei tre ulike metodane kunnskap, stereo -og rørslebaserte metodar. Kunnskapsmetoden nyttar seg av eigenskapane i bildet slik som symmetri, fargar, hjørnepunkt, lys på køyretøy og liknande. Stereometoden skil ut objekt frå eit generert disparitetskart eller ved bruk av invers perspektivkartlegging. Rørslebaserte metodar nyttar seg av optisk flyt for å skilje mellom objekt.

Hypoteseverifisering(HV)
HV steget blir brukt for å verifisere at objektet som er detektert faktisk er eit køyretøy. Sun et al. delar dette steget opp i dei to delane samanlikning og utsjåandebasert verifisering. For å samanlikne må objektet ha visse kriteria for å kunne bli satt som eit køyretøy. Eksempelvis kan dette vere vindaugsrute, skiltplate eller heilt enkelt ei form som passar til malen for samanlikning. Utsjåandebasert verifisering nyttar seg av enten klassifisering for å skilje mellom objekt, eller ved å modellere sannsyns fordelinga av eigenskapar i kvar klasse. Klassifiseringa kan bli gjort med kjente klassifiserarar slik som nevrale nett eller supportvektormaskiner.

Figur 38: Venstre: To køyretøy er synlege i bildet. Midten: Ulike element i bildet er detektert ved HG. Bildet syner eit område som er feildetektert. Høgre: Element i bildet er verifisert som køyretøy ved HV. Bilda er henta frå *[4]*.

Vidareføring av køyretøyattkjenninga
Sivaraman et al. presenterte i 2013 ein rapport som kan bli sett på som ei vidareføring av den ovannemnte rapporten[5]. Den delar opp kamerabasert attkjenning i mono -og stereobaserte metodar. Både mono -og stereobasert attkjenning er vidare delt opp i utsjåande -og rørslebasert attkjenning med eit tillegg i stereomatching for stereosyn. Denne rapporten syner at attkjenning i nokre tilfelle kan bli gjort direkte ifrå bildet, og det er då mogleg å hoppe over HG steget.

Dei to rapportane som er nemnt her blir å rekna som to solide rapportar som syner det som kan kallast «state-of-art» innanfor køyretøyattkjenning. Nyare rapportar basera mykje av sitt arbeid på det som er summert her. Skilnaden mellom dei to rapportane er den tydelege framgangen som er gjort innanfor feltet dei siste åra.

3.2 Aktive sensorar for objektdeteksjon

Utanom kamera så nyttar autonome bilar seg av radar, lidar og ultrasoniske sensorar. Dette er definert som aktive sensorar som kan måle avstand presist og er brukt for å skape eit bilde av miljøet rundt køyretøyet. Oppsett med slike sensorar observera objekt, og klassifisera dei som køyretøy basert på størrelse og bevegelsane til objektet. Dersom objektet blir observert når det står i ro er det vanskeleg å vite om det er eit køyretøy eller ein anna type objekt. I dette kapittelet blir desse sensorane presentert og samanlikna.

Radar

Autonome køyretøy nyttar seg av radar med eit smalt synsfelt[47]. Radarteknologien som blir brukt om bord i køyretøy sender ut radiosignal med millimeter-bølgelengder, og det reflekterte signalet blir så behandla slik at avstand til objektet kan bli utrekna. Ein radar sine målingar er forholdsvis påverka av støy, men er generelt meir robust i ulike vêrforhold enn dei alternative sensorane. Dette gjer betre resultat sjølv om det regnar, snør eller om det er nedsett lyssetting på kveld og natt, sjølv om denne sensoren er meir avhengig av filtrering av reflekterte signal[5]. Den største ulempa med radar er problem med klassifisering av objekt som trafikklys og skilt og at motgåande køyretøy kan påverke målingane til kvarandre og skape falske verdiar.

Lidar

Lidar har ikkje vore mykje brukt før i seinare tid. Figur 39 syner Velodyne sin versjon montert på taket av Uber sin Ford Fusion, der den tar opp mykje plass og forandrar drastisk på køyretøyets utsjåande. Denne typen sensor sender ut lysbølger i eit spektrum på 600-1000nm med ein frekvens på 10-15Hz, og signala som blir reflektert kan vere med på å danne eit 3D-kart over miljøet rundt bilen[5].

Figur 39 Uber sin autonome bil kartlegg gatene med lidar utvikla av Velodyne. Bilde er henta frå [48].

Denne teknologien er førebels ganske dyr. Men utviklinga avansera kjapt, og Velodyne har allereie utvikla mindre og meir brukarvennlege lidarsystem berekna for den autonome marknaden[49]. Bakdelen med denne teknologien er at den er følsam i ulike vêrtypar som tåke, regn og snø[5]. Sidan lidar er dyre i innkjøp og det førebels ikkje skapar noko særskilt flott implementering på køyretøyet er det enno ikkje skikkeleg utbreidd blant dei kommersielle fabrikantane.

Ultrasoniske sensorar

Ultrasoniske sensorar blir hovudsakleg nytta for å sjekke dødsonene rundt køyretøyet. Ved å sende ein høgfrekvent lyd-puls over 20kHz, og så måle tida den brukar på å bli reflektert, så blir avstanden til objekta i nærleik av sensoren kalkulert. Denne type sensorar er brukt for å sjekke perimeteret rundt køyretøyet grunna sin korte måleavstand i forhold til radar og lidar[50].

3.3 Passive sensorar for objektattkjenning

Passive sensorar mottar lys eller lyd utan at dei avgjer noko signal. I litteraturen er det kamera som hovudsakleg blir brukt for attkjenning av køyretøy, men det vil også heilt kort bli gjennomgått arbeid som er gjort med akustiske sensorar for køyretøy attkjenning. Kamera har vore mykje brukt og forska på opp gjennom åra. I problemstillinga rundt autonome bilar har det vore sett på alt frå mono til multiple kamera for deteksjon av objekt i ei scene. Hovudforskjellen mellom mono -og multiple kamera i dette tilfelle er moglegheita for å estimere avstand og størrelsar ifrå bilda. Bilde som blir innhenta av kamera gjer informasjon om omgivnaden som aktive sensorar ikkje kan gje på same måte. Dette vil være seg fargar og kontrastar som kan skilje eit køyretøy frå til dømes ein container ved å sjå på baklys, blinklys o.l.

Vidare så er køyretøyattkjenning i litteraturen hovudsakleg delt opp i to hovuddelar, utsjåande -og rørslebasert. Utsjåandebasert attkjenning finn køyretøy direkte i statiske bilde og klassifisera dei deretter, i.e. direkte frå pikslar til køyretøy. Rørslebasert attkjenning nyttar eigenskapane bak det å sjå på fleire bilderammer i ein sekvens for å skilje ut rørsler i scena og klassifisere objekt utifrå rørsle.

Monokamera
Utsjåandebasert attkjenning er mest brukt i monokamera. Ved å finne eigenskapspunkt i bilde blir køyretøy klassifisert ved ulike kriterium. Dette vere seg symmetri, farge, kantar, hjørne, lys eller andre delar på køyretøyet. Saman med andre sensorar vil dette kunne skilje avstanden mellom dei ulike objekta som er funne i scena.

Multiple kamera
Når det blir lagt til to eller fleire kamera blir det mogleg å estimere djupne i ei scene samt størrelse på objekt. Ved stereosyn og enda fleire kamera er det moglegheit for å estimere eit disparitetskart og skilje mellom ulike objekt ved ulike avstandar. Det er da naturleg å søke etter objekt i eit 3D-miljø, og i litteraturen er det mest vanleg å nytte seg av rørslebasert attkjenning i nettopp eit 3D-miljø.

Mikrofon (akustiske sensorar)
Akustiske målingar er ikkje mykje brukt i konvensjonelle køyretøy, og denne delen kan sjåast på som ein relevant digresjon. Andreas Klausner, Stefan Erb, Allan Tengg og Bernhard Rinner la fram ei hypotese om at køyretøy etterlata seg akustiske avtrykk som kan forbetre robustheita og attkjennings-raten i samarbeid med allereie eksisterande video[51]. Dei har i dette eksperimentet satt to mikrofonar langs ein veg, og ikkje om bord i eit køyretøy da dette ville ha skapt enda fleire utfordringar. Eksperimentet synte at løysinga fungerte best for hastigheiter under 70 km/t og i vegbanar med ei køyreretning. Det er i denne rapporten ikkje diskutert noko vidare om dette.

3.4 Køyretøyattkjenning med kamera

Dette kapittelet tar for seg tidlegare arbeid med kamera for køyretøyattkjenning. Det er gjort mykje forsking på kva type bildebehandling som er blitt brukt opp gjennom åra, og det vil her i hovudsak baserast på rapporten av Sivaraman et al. som delar køyretøyattkjenning i trafikken inn i to hovudkategoriar, utsjåande -og rørslebasert[5]. Rørslebasert objektattkjenning er hovudsakleg brukt for «tracking» av objekt. Dette har etter kvart blitt meir vanleg enn utsjåandebasert sidan det i eit trafikkbilde er interessant å føresjå retninga på hindringar i vegbana. Som nemnt i kapittel 3.3 er utsjåandebasert attkjenning hovudsakleg presentert i monokamera, men er høgst relevant i ein stereorigg for å skilje mellom objekt.

3.4.1 Utsjåandebasert objektattkjenning

Det er her valt å presentere ei utval tidlegare arbeid på utsjåandebasert objektattkjenning. Det er her interessant å sjå på dei ulike løysingane mellom mono -og stereoriggar.

Monosyn
Dei siste åra har detektering gått frå forholdsvis enkle metodar der det blir sett på eigenskapar som symmetri og kantdeteksjon på køyretøy[4], til meir robuste metodar for attkjenning der ulike eigenskapspunkt i bilda blir trekt ut og køyretøy kan bli klassifisert direkte frå bilda[5].

Ein gjengangar i litteraturen er å finne HOG og Haar-liknande eigenskapar i ei trafikkscene. Bakgrunnen for desse algoritmene er forklart i kapittel 2.3.2. HOG-eigenskapar er nytta i fleire studiar [52], [53], og i [54] blir HOG nytta i samband med ein SVM-klassifiserar for å kjenne att køyretøy i urbane omgivnadar. Eksperimentet viser til ein gjennomsnittlig køyretøydeteksjon på 97%. Rapporten konkluderar med at HOG-eigenskapane er tunge i ei utrekning, noko som ikkje er så ideellt for sanntidsapplikasjonar. Dette har gjort til at det er konstruert utvidingar av denne detektoren som pi-HOG[55] og L-HOG[56], samt nye metodar for å finne region av interesse.

Haar-eigenskapar vart nytta i [28] i samband med Adabost som ein aktiv læringsalgoritme. Bakgrunnen for valet av eigenskapsdetektor er den kjappe kalkuleringa denne algoritma oppnår ved å nytte integralbilde. Dette er bakgrunnen for å bruke Haar-eigenskapar i dei fleste studiar[27]. Både HOG og Haar vart nytta i [57] for å detektere køyretøy i ei køyrebane. Det var her eksperimentert med feilraten mellom HOG og Haar, samt ein kombinasjon av begge to der den siste kom ut som ein klar vinnar.

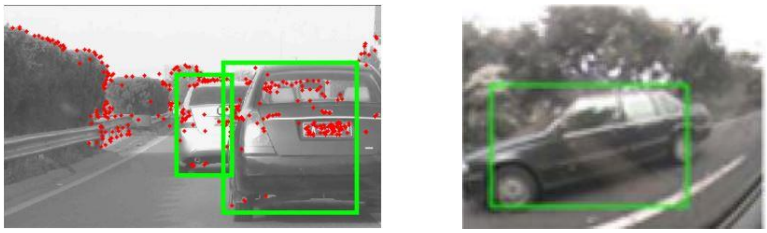
Figur 40 Ulike resultat frå deteksjon av køyretøy: Venstre: HOG-eigenskapar[54]. Midten: Haar-eigenskapar[28]. Høgre: Fusjon av HOG -og Haar-eigenskapar[57].

[58] eksperimenterte med ulike eigenskapar som PCA[2], wavelets og Gabor filter med klassifisererane SVM og nevrale nett. Konklusjonen her var at det ikkje var noko synleg forskjell i tidsbruk for dei ulike kombinasjonane, men ein fusjon av wavelets og Gabor gav det mest nøyaktige resultata. På bakgrunn av at forfattarane av denne rapporten ønskte å konstruere ei algoritme for sanntids-deteksjon, blei det konkludert med at denne metoden var for tung for ei sanntids-kalkulering. PCA kom her dårligast ut,

[2] PCA står for «principal component analysis» og blir brukt ved å konstruere ein eigenskapsvektor basert på eigenvektorane frå eit bilde[98].

rett etter wavelets og Gabor. Systemet dei valte å nytte i ei sanntidsdetektering var med Gabor filter og ein SVM-klassifiserar. Dei oppnådde ein deteksjonsrate på 10Hz som tilsvarar ein deteksjon for kvar 3m i ca. 70mph (112km/t). [31] undersøkte HOG -og Gaboreigenskapar med tre ulike klassifiserara; SVM, multilags perceptron NN, og Mahalanobis avstand. Eksperimentet her gjekk ut på å søke etter den den beste metoden for hypoteseverifisering. Datasetta som vart brukt til trening hadde da allereie blitt markert med eit område av interesse. Resultata her synte at HOG eigenskapar var betre enn Gabor ved alle klassifiserarane, både på presisjon og deteksjonsrate, der HOG med ein SVM-klassifiserar gav best resultat.

SIFT algoritma blei nytta i [59] for å detektere bakdelen av køyretøy i ei køyrebane. Ved å trene på ein database med 1050 bilder av køyrefelt med køyretøy under ulike forhold, var det mulig å identifisere køyretøy som var delvis synlige. Dette syner den sterke eigenskapen bak SIFT algoritma som gjer til at det er ei mykje brukt algoritme. Den like kjente algoritmen SURF vart nytta ilag med Canny kantdeteksjon i [60] for å sjekke for køyretøy i blindsoner. SURF finn først dei eigenskapane som er distinktive og skaleringsinvariante for eit køyretøy. Kantdeteksjon nyttar dei ved å finne det kant-segmente med lengst samanhengande linjer, for så å velje det som ein eigenskap for køyretøyet. Dette kan til dømes være kanten på eit dekk sidan dei er runde og har dermed kontinuerlige linjer.



Figur 41 Venstre: Delvis skjult køyretøy er detektert med SIFT[59]. Høgre: Køyretøy i ei dødsone er detektert med SURF[60].

Det er utført eksperiment med ein kombinasjon av detektorar i fleire rapportar. [61] nyttar ein kombinasjon av Haar og SURF for å verifisere køyretøy. Område av interesse blir her funne ved å detektere køyrefeltet, men kan oppleve feilklassifisering dersom feil køyrefelt blir detektert.

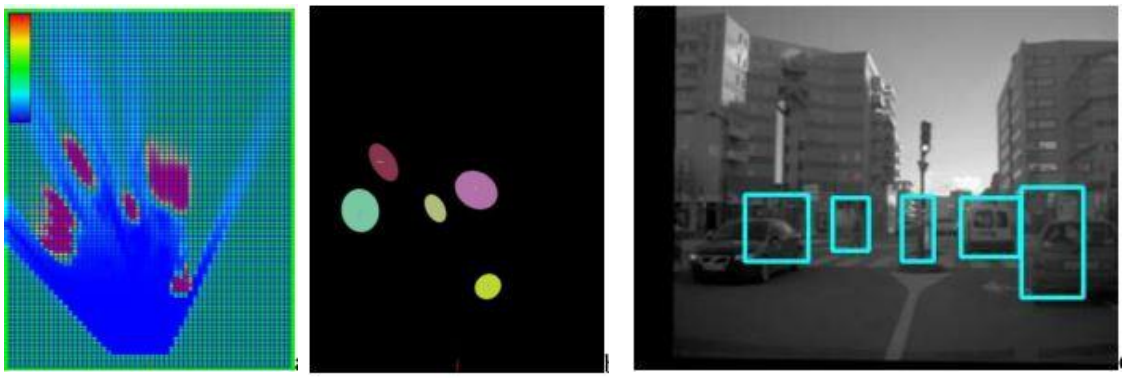
Med fokus på raske deteksjonar og attkjenningar drar [62] ein parallell til dei menneskelege evnene til å raskt oppfatte og prosessere miljøet rundt seg. Her blir det presentert arbeid gjort av Koch og Ullman der dei føreslo konseptet “visual salency”. Dette er eit omgrep som er tatt frå nevroforsking og som omhandlar nettopp menneskets raske evne til å detektere spesielle objekt i ei scene. Forsking på dette området er gjort i fleire år, men denne rapporten drar fram arbeid gjort i 2012 som krevjar låg reknekraft og er rask å estimere[63]. Figur 42 syner resultatet av denne type HG som er eit segmentert bilde som framhevar det objektet som er interessant å verifisere. Denne rapporten framhevar eigenskapane bak djupe neurale nett for klassifisering. Rapporten konkludera med ein deteksjonsrate på 98% og ein hastigheit på 31Hz.

Figur 42 Saliency map: Venstre: Input av eit køyretøy. Midten: gråskalert saliency kart. Høgre: Segmentert bilde som framhevar området der kandidaten for verifisering er i bildet. Bilda er henta frå [62].

Stereosyn

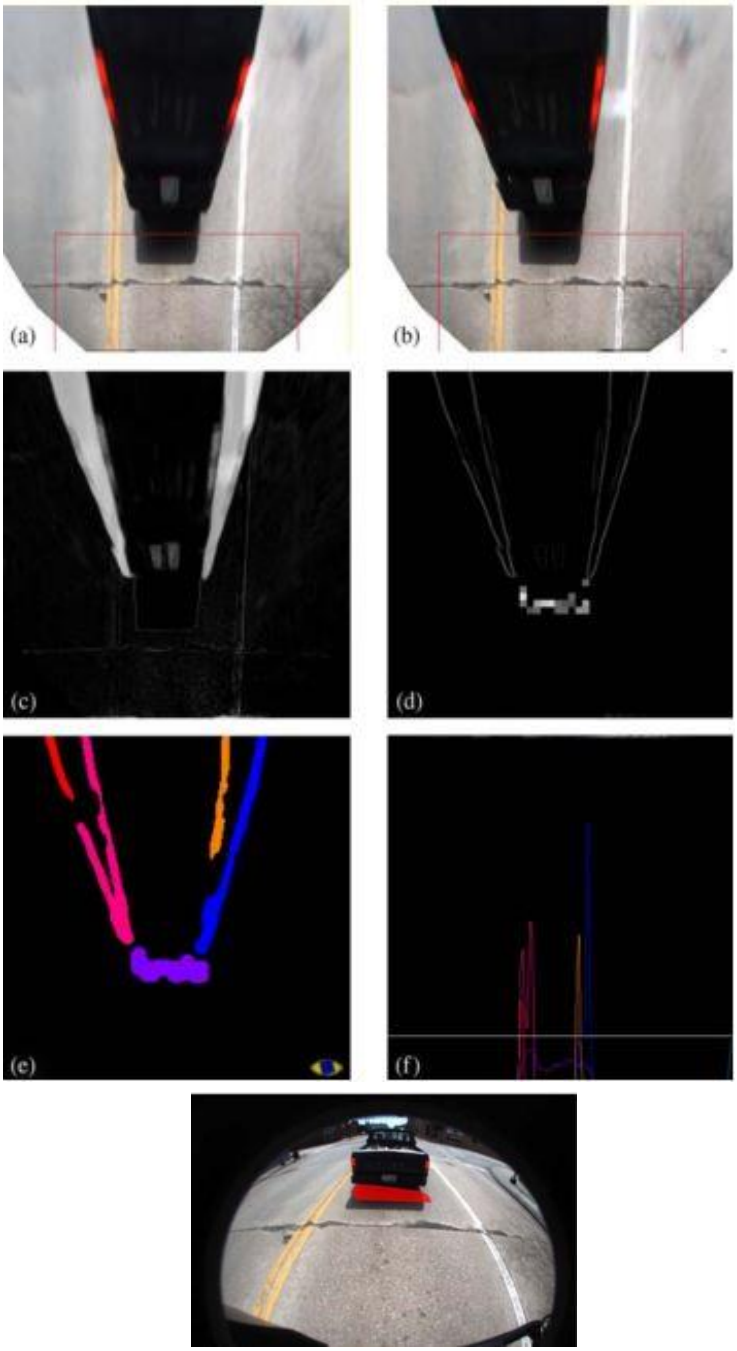
Stereosyn gjer moglegheit for å måle størrelse på objekt og skilje mellom ulike objekt i ei scene ved å sjå på dispariteten, som er forklart i kapittel 2.2.3. [64] nyttar seg av ein stereorigg for å skape eit disparitetskart av omgivnaden med den kjente semi-globale metoden, som er forklart i kap. 2.3.3. For å vidare skilje mellom vegbana og køyretøy er konseptet v-disparitet introdusert. Den blir funnen ved å nytte eit histogram med disparitetsverdiar for kvar piksel-lokasjon med same vertikale bildekoordinat. Det er då mogleg å sortere ut pikslar som er tilhøyrande vegbana ved ulike algoritmar. Denne rapporten nyttar seg av Hough-transformen[3] for å klassifisere pikslar som veg på bakgrunn av ein modell mellom disparitetskartet og v-koordinatane. Når vegen er sortert ut frå scena, kan køyretøyet bli detektert ved å sjå på kontaktpunkt mellom dekk og vegbane. Figur 43 syner resultatet av v-dispariteten frå rapporten. Denne metoden var blant anna nytta av det kjente TerraMax køyretøyet under DARPA Urban Challenge 2007[65]. Ei vidareføring av arbeidet med v-disparitet er u-disparitet[66]. Til forskjell frå v-disparitet så er denne metoden generell for heile bildet. Ved å estimere eit disparitetsbilde så kan avstanden til dei ulike objekta i scena finnast, og det resulterande bildet kan samanliknast med eit fugleperspektiv over den målte scena. Dette bildet blir nytta for å lettare estimere eit oversiktskart over vegbana, og er synt i Figur 44. Til samanlikning så nytta TerraMax IPM, som er forklart i kap. 2.1.2, for å detektere objekt i køyrebana[65]. Denne metoden skapar da eit fugleperspektiv av køyrebana, som synt i Figur 45. Ved å finne gradienten i bilde vil køyretøyet skilje seg ut og bli funnen ved å sjå på polar-histogrammet til bildet.

Figur 43 V-disparitet: Venstre: Bilde av ein kurva veg frå stereokamera. Midten: Vegen er sortert vekk. Her er delar av vegen registrert som hindringar sidan vegen er antatt rett. Høgre: Vegen er antatt kurva og hinder i scena er enda meir tydelige enn for bildet i midten. Bilda er henta frå [64]



Figur 44 U-disparitet: Venstre: Estimert oversiktskart frå u-dispariteten. Midten: Detekterte objekt frå oversiktskartet. Høgre: Dei detekterte objekta er prosjektert over det venstre stereobildet. Bilda er henta frå [66]

[3] Houghtransformen er ein metode som er brukt for å finne linjer i eit bilde[99].

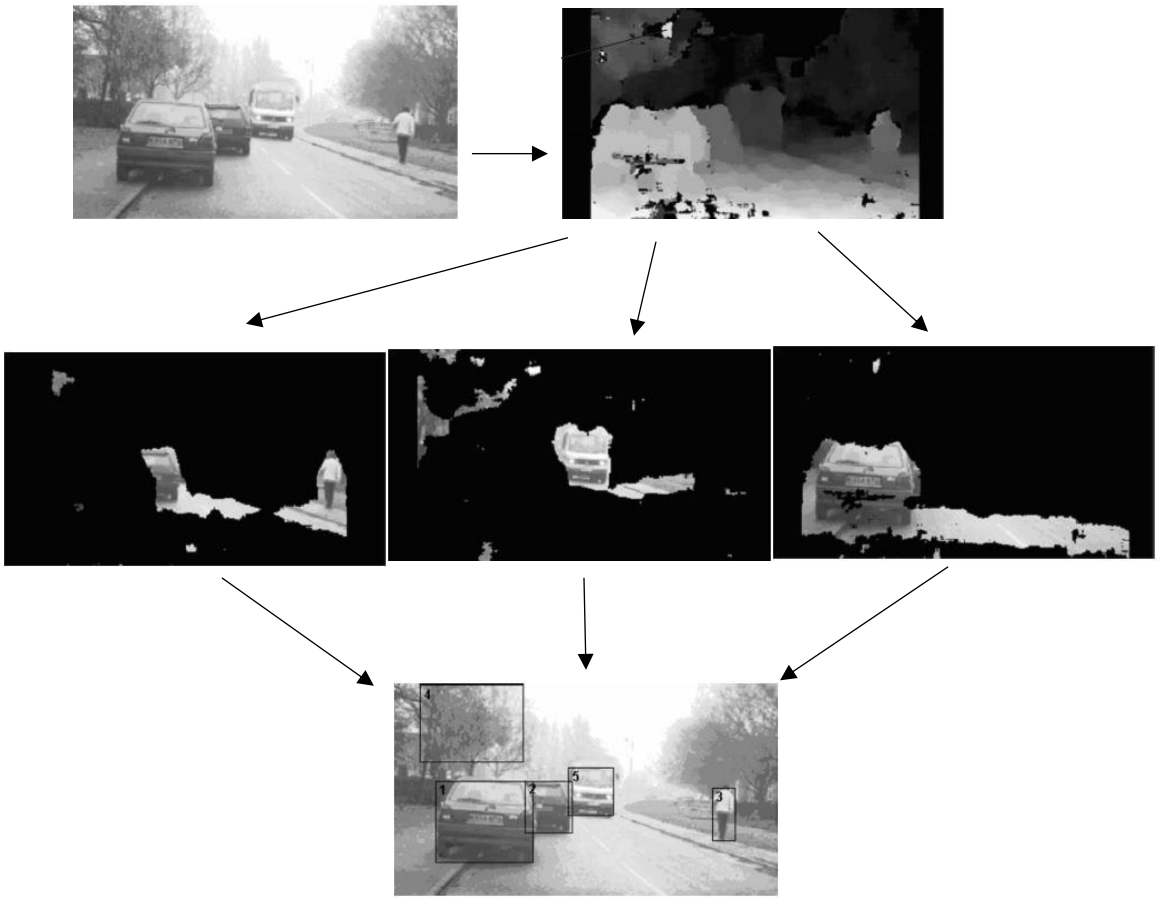


Figur 45 IPM. (a) og (b) syner IPM bilde generert frå stereokamera. Vidare blir det generert eit bilde for å framheve skilnadane mellom kvart av bilda (c). (d) syner resultatet etter bildet er filtrert med Sobel kantdeteksjon. (e) syner ei terskling som framhevar eigenskapar frå køyretøyet. Og til slutt så blir kantane representer i polare koordinatar (f). Det nedste bildet syner det detekterte køyretøyet. Her er det ein ekstrem linseforvrenging som kun blir tatt hensyn til før IPM bildet blir generert. Bilda er henta frå [65].

Kombinasjonar av u -og v-disparitet vart nytta i [67]. Rapporten nyttar først eigenskapane bak DNN for å detektere køyretøy og skape eit område av interesse ved å trene på eit sett med bilde, deretter blir køyretøya verifisert ved å studere scena i form av v -og u-disparitet.

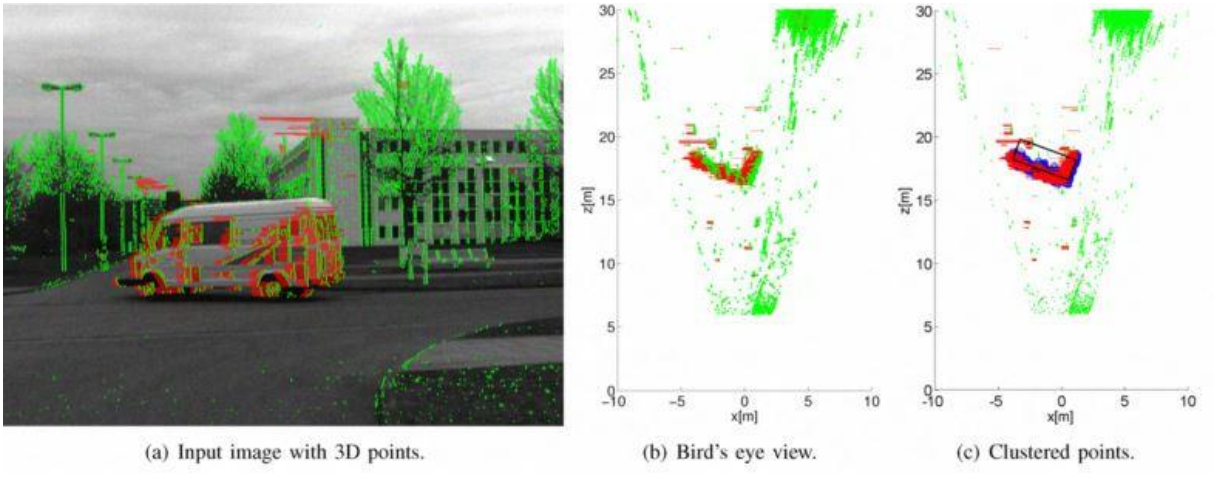
I [20] blei eigenskapane bak stereomatching nytta for å måle avstand til objekt i ei scene og deretter konstruere lagvise bilde, basert på avstand, for å skilje mellom dei ulike objekta. Figur 46 illustrera arbeidet bak stereomatching. Her har dei valt å nytte SAD algoritmen for å finne korrelasjonen mellom

LOG-filtrerte bilde, der bakgrunnen for å pre-filtrere bilda er å forsterke eigenskapspunkta. Målefeil for avstand og størrelsemåling var her under 5% og 10%, respektivt. Ulike eigenskapar som størrelse og bildeintensitet var kombinert i [68] for å detektere objekt i ei scene. Her blir høgde og breidde på objekt målt i både djupne -og intensitetsbilde, samt samanlikna med ein mal som er utarbeidd både for køyretøy og personar i scena.



Figur 46 Lagvis deteksjon. Utifrå stereobilde blir det generert eit disparitetskart som innehar avstandar og størrelsar på objekt i scena. Ved å skilje mellom dei ulike avstandane blir det generert lagvise bilde med informasjon om den ulike avstanden. Siste bilde syner dei detekterte objekta med målt størrelse og avstand. Her er det også ein feildeteksjon (4). Bilda er henta frå [20].

Nokre studiar har og nytta clustering i disparitetskartet for å detektere objekt. [69] detektera køyretøy frå disparitetsbilde, og estimera samtidig køyretøyets orientering i scena ved å nytte ein iterativ næraste-punkt-algoritme på 3D-modellen. Figur 47 illustrera dette. I [70] blir det brukt ulike eigenskapar som er basert på tidlegare arbeid med deteksjon ved hjelp av stereosyn. Kvart punkt på bakdelen av eit køyretøy vil ha lik avstand frå kamera og dermed ein lik disparitetsverdi. På bakgrunn av dette blir søket etter eigenskapar basert på punkt som er samla i disparitetskartet. Sjølve deteksjonen er gjort i fleire steg. Ved å dele scena inn i fleire områder (gates), minimerast søkefeltet og dermed blir berekninga raskare. Eigenskapar i bildet blir definert av kantar på køyretøyet som blir framheva med gradienten for så å bli definert med clustering. Avstanden til kvar eigenskap blir så kalkulert ved å konstruere eit disparitetskart ved ein lokal metode ved hjelp av SAD-kostnad, som er forklart i kapittel 2.3.3. Vidare blir dei falsk-positive klassifiseringane sortert ut ved ulike metodar, blant anna HOG-AdaBoost klassifisering.



(a) Input image with 3D points. (b) Bird's eye view. (c) Clustered points.

Figur 47 Clustering: (a) syner det opprinnelige bildet med plott av 3D-punkt. (b) syner eit fugleperspektiv av den samme scena. (c) syner clustering av køyretøyet i blått. Bilda er henta frå [69]

Dei fleste studiar presentera stereosyn i ein botn-opp-metode, der dispariteten blir kalkulert og objekt blir detektert i ein 3D-modell. [71] foreslår å angripe problemstillinga ved å snu opp-ned på metoden, her kalla stereoassistanse. Her blir dei relevante objekta detektert i eit enkelt kamera, og vidare lokalisert ved å sjå på informasjonen frå to kamera. Bakgrunnen for denne løysinga er å minimere den komplekse kalkuleringa som må til for å generere eit komplett disparitetskart. Rapporten understrekar fleire fordelar med å utføre deteksjonen ved stereoassistanse:

- Metoden er meir robust i søket etter felles punkt sidan det blir gjort i mindre delar av bildet (samanlikna med klassisk stereosyn). Dette fører til at det kan brukast matematisk tyngre metodar for å finne felles punkt på dei mindre delane i bildet, og samtidig ha eit raskt system.
- Sjølv om delar av bildet har forstyrringar, som forvrenging pga. regn på vindaugsruta, sjå Figur 48, klarar systemet å detektere køyretøy.
- Kan detektere objekt ved lengre avstandar. I eit vanleg disparitetskart kan objekt ved lange avstandar smelte saman og sjå ut som eit objekt. Ved stereoassistanse blir kvar av objekta detektert før avstanden blir målt.

Figur 48 Stereoassistanse: Øvst: Venstre og høgre bilde ifrå stereokamera der høgre kamera opplever ei forvrenging pga. vatn på ruta. Nedst: Kvart av køyretøya er detektert i kvart av bilda. Bilda er henta frå [71].

3.4.2 Rørslebasert objektattkjenning

Til forskjell frå utsjåandebasert attkjenning så er dette tema basert på bruk av fleire bilderammer i serie, som gjer moglegheit for rørsleattkjenning basert på ulike algoritmar slik som optisk flyt[5]. Denne type objektattkjenning er, som nemnt i kapittel 3.3, mest brukt for stereosyn. Det vil her bli presentert nokre eksempel på arbeid med rørslebasert attkjenning.

Sivaraman et al.[5] presentera arbeid med både mono og- stereosyn for rørsleattkjenning. Her favoriserast stereosyn sidan det gjev ein moglegheit for å måle objektet i eit 3D-rom. For monosyn blir det presentert ulikt arbeid der det blant anna er brukt adaptive bakgrunnsmodellar som skilja mellom eit køyretøy i bevegelse og den statiske bakgrunnen. I tillegg blir det presentert arbeid der optisk flyt blir nytta for å detektere køyretøy i bevegelse i enkle bilde. For stereosyn drar han fram fordelen med to kamera, der objektet kan bli detektert i bilde frå det eine kamera og deretter bli observert over fleire bilderammer ved å nytte optisk flyt. Fordelen er at køyretøyet kan bli observert i 3D-rommet ved hjelp av disparitetskartet. Tabell 1 syner nokre eksempel på tidlegare arbeid med rørslebasert attkjenning.

Tabell 1 Eksempel på tidlegare arbeid på rørslebasert attkjenning

Monosyn			
Årstal	Studie	Metode	Kommentar
2006	[72]	Dynamisk modellering av bakgrunn for å overvake køyretøy i ei forbikøyring på ein video.	Detektera ca. 100% av køyretøy som utførar ei forbikøyring
2011	[73]	Deteksjon ved optisk flyt, klassifisering ved ein «hidden Markov modell».	Ca. 86,6% av køyretøy blir detektert. Deteksjonen er avgrensa av avstanden i scena.
Stereosyn			
Årstal	Studie	Metode	Kommentar
2005	[74]	Optisk flyt	Interessepunkt er følgt i bildeplanet med optisk flyt. Tilsvarande 3D-punkt og hastigheit er følgt med eit Kalmanfilter.
2012	[75]	Optisk flyt og oversiktsnett over scena.	Effektiv berekning ved å implementere systemet på ein GPU.

3.4.3 Utfordringar ved å bruke kamera som sensor

Passive sensorar som er avhengig av lys(kamera) har fordelen med at dei kan kjenne att detaljar og behandle informasjon i trafikken. Dette vere seg køyrefelt, trafikklys og objekt som menneskjer og køyretøy. Ulempa med kamera vil oppstå når belysning og eksterne parameter forstyrrar målingane. Eksempelvis så er kamera avhengig av tilstrekkeleg med lys for å kunne få bilde som kan prosesserast, så ved nattetid vil det vere avgrensingar. For stereokamera er i tillegg algoritmane for objektdeteksjon basert på at stereokamera sine ytre og indre parameter er kjent, noko som kan bli justert avhengig av køyretøyets posisjon og andre ytre element.

Skiftande kameraparameter
Ei av problemstillingane som kan dukke opp i eit slikt miljø, er den konstante bevegelsen i bilen og deretter kamerariggen. Dette vil igjen føre til at dei eksterne parameterane til kamera vil forandre seg. Det kan stillast spørsmål om dette er ein faktor som gjev stor innverknad sidan dei fleste bilar i dag baserer seg på ein fusjon mellom kamera og andre avstandsmålingar. Det er forska på fleire ulike metodar på sanntids kamerakalibrering dei siste åra, og i [15] blir det presentert ein omfattande kontinuerleg kalibrering av stereokamera for dei ytre kameramatrisene. Her nyttar dei eit Kalman-filter for å predikere dei eksterne parameterane. På bakgrunn av den epipolare avgrensinga, som er gått gjennom i teoridelen, så kan rotasjon og translasjon mellom kamera bli funne ved å finne felles punkt i begge bilda, og deretter estimere den fundamentale matrisa som fortel noko om rotasjon og translasjon mellom kamera. Den endelege løysinga estimerer dei 6-DOF relative transformasjonar mellom både kamera, men også mellom køyretøy og kamerarigg.

Nedsett lyskjelde
Ved ulike lysforhold vil dei konvensjonelle kamera ha vanskeleg for å kunne gje optimale bilde, da spesielt i mørke og skumring da ytre lysforhold er svakare enn ved dagtid. På side 52 blir det presentert ulike kamera som kan erstatte konvensjonelle kamera i slike situasjonar, men vanlegvis er det ynskjeleg å kunne bruke dei ordinere pga. pris. Attkjenning av objekt i mørke er eit problematisk tema sidan kamera er avhengig av køyretøy sine eigne og/eller ytre lyskjelder for å få fram eigenskapane i eit bilde. Ulike løysingar på dette problemet er blitt presentert dei siste åra. [76] nyttar fleire eigenskapar for å kunne detektere køyretøy både under dagslys og mørke. For å skilje mellom køyretøy under mørke scener ser dei på baklys, og klassifisera køyretøy på bakgrunn av avstanden mellom to baklys. [77] har også gått ut ifrå situasjonar som er dårleg opplyst og der køyretøyet manglar direkte attkjennande eigenskapar. Denne rapporten legg og vekt på at nokre u-land ikkje har noko regelverk for at køyretøyet må ha to fungerande baklys, og at tidlegare algoritmar da kan oppleve feildeteksjon. Her blir det presentert ein deteksjon der køyretøy med berre eit fungerande baklys kan bli identifisert som køyretøy, og ved å addere optisk flyt på scena blir køyretøy segmentert og detektert. Tabell 2 viser nokre eksempel på arbeid ved nedsette lysforhold.

Tabell 2 Eksempel på tidlegare arbeid for køyretøydeteksjon ved nedsette lysforhold

Årstal	Studie	Metode
2005	[78]	Nyttar stereosyn for å finne 3D kantar av hindringar, og fargedeteksjon for deteksjon av baklys.
2007	[76]	Deteksjon basert på baklys. Vurderast på bakgrunn av ein fast avstand mellom baklys.

2016	[77]	Deteksjon basert på baklys. Samanlikna resultat med state-of-art.

Tunge kalkulasjonar

Den informasjonen som eit bilde innehar krevjar avanserte algoritmar for å sortere. I kapittel 3.4.1 er arbeid med ulike algoritmar presentert, der særleg SIFT og SURF ikkje levera resultat som er gode nok for eit sanntids system. I nyare tid er produksjon og konstruksjon av meir kraftige datasystem og GPU-ar blitt meir aktuelt. Dette har gjort til at datasyn sine eigenskapar er meir aktuelle i dag enn kva dei var for nokre år sidan[79].

Ulike typar kamera for ulike scener

Som tidlegare nemnt så er det ulike problemstillingar som eit konvensjonelt kamera ikkje fungera så godt i, dette være seg i mørke og ekstremt belyste settingar. I lågt belyste scener må kamera ha lengre lukkartid for å få fram detaljar i bilde, noko som går utover behandlingstida for ein deteksjonsalgoritme. Det er stadig ei utviklinga av kamera og det blir stadig betre oppløysing i kamera som gjer enda betre informasjon om scena. Dilemma framover er om det skal fokuserast meir på ukonvensjonelle kamera som kan lese trafikkbilde i andre spektrum enn vanleg. Dette vil jo gå på kostnad av pris, som er hovudmotivasjonen bak det å faktisk bruke eit kamera som ein sensor. I tillegg så har dei vanlege dagslys kamera mindre energiforbruk enn alternativa. Tabell 3 syner nokre eksempel på arbeid med alternative kamera. I [58] blir Ford sitt proprietær låglyskamera nytta for å kunne detektere køyretøy både under dagtid, men også under forhold med mindre lys. I eit arbeid for å få ein konvoi til å følgje kvarandre (eng: platooning) er det brukt eit sett ulike sensorar for å registrere køyretøy under lågt belyste forhold[80]. Her er primærsensoren eit vanleg kamera, men det vart naudsynt å utvide med eit termisk og eit låglys kamera for å skape eit robust system ved både skumring og natt. I [81] nyttar dei seg av ein fusjon av ulike kamera som består av eit vanleg kamera, eit farge nattsyn kamera og eit termisk kamera.

Tabell 3 Eksempel på tidlegare arbeid med ulike kamera

Årstal	Studie	Kamera
2006	[58]	Proprietær låglys kamera
2015	[80]	Termisk -og låglys kamera
2015	[81]	Nattsyn -og termisk kamera

Desse ulike alternativa skapar betre deteksjonar under dårlegare lyssettingar, men er også dyrare i innkjøp. Denne problematikken må vurderast i forhold til kva scener køyretøyet skal operere i. Til dømes så er det presentert arbeid over der deteksjonen basera seg på andre køyretøy sine ytre lys, slik at konvensjonelle kamera kan nyttast. I tillegg så vil eit køyretøy sine eige lys lyse opp scena framfor seg slik at vegbane og hindringar er synlege.

3.5 Fusjon mellom aktive og passive sensorar

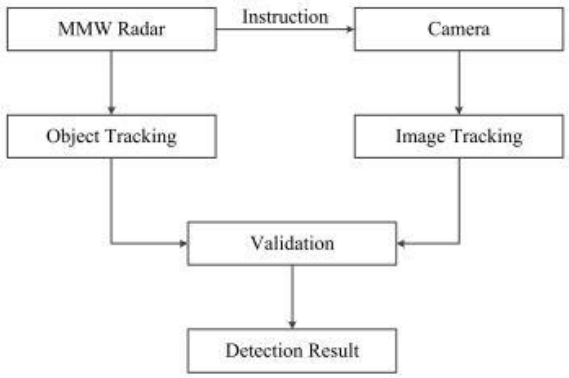
Som forklart i kapittel 3.2 og 3.3 har dei ulike sensorane ulike kvalitetar og eigenskapar. Dei kommersielle køyretøya som fokusera på autonome køyretøy har ulike kombinasjonar av sensorar, da hovudsakleg kamera og radar. Eksempelvis så nyttar Tesla seg av ein radar som primærsensor med kamera som ein sekundersensor. Ford har nyleg lagt fram ei pressemelding der dei seier at arbeidet med autonome bilar vil nå ein høgdepunkt i 2021 med ein fusjon mellom kamera og Velodyne sine lidarsystem[49].

Det er dei siste åra studert på ulike kombinasjonar av sensorar. Bruk av lidar har først tredd fram i dei seinare år pga. kostnaden har vore så høg at det ikkje har vore påtenkt til kommersiell bruk. Lidar har vore mykje brukt i prototypane av autonome bilar sidan det er mogleg å kontinuerleg ha ei oversikt over miljøet rundt køyretøyet, og under testing av køyretøyet er det mogleg å kartlegge områder som kan nyttast på eit seinare tidspunkt. Sun et al. som tar for seg arbeid før 2005 legg fram at fordelane med å nytte fleire sensorar skapar meir presise målingar, men robustheit og pålitelegheit bør forbetrast[4]. Sivaraman et al. som tar føre seg arbeid mellom 2005-2013 kommentera på prisreduksjon rundt radar og lidar, og at retninga framover vil fokusere på ein fusjon mellom ulike sensorar[5].

Det er i litteraturen tre hovudmetodar for sensor-samarbeid[82]:

- Lågnivå: Informasjon er innhenta og nytta direkte til å skape eit nytt informasjonsbilde. Til dømes kombinere vanlege bilde med det infraraude spekteret, eller to stereokamera for å skape eit disparitetskart.
- Mediumnivå: Kombinere eigenskapar frå ulike sensorar for å detektere og klassifisere objekt. Eksempelvis avstand frå radar og eigenskapspunkt frå kamera.
- Høgnivå: Basert på deteksjon og klassifisering mellom dei ulike sensorane blir den endelege avgjersla tatt ved å sjå på den vekta summen basert på kvar av sensorane.

Det som er repeterande i litteraturen er oftast ein mediumnivå fusjon mellom kamera og radar der eit monokamera identifisera køyretøy, og radaren målar avstand. [83] nyttar ein kombinasjon av radar og kamera og drar ein parallell til det menneskelege synet. Her nyttar dei seg av ein mm-radar som skal erstatte «rod-cell» i det menneskelege auget, og kamera som skal erstatte «cone-cell». Bakgrunnen for denne samanlikninga er menneskets raske evne til å oppfatte bevegelse med «rod-cell» for så å definere ein ROI, for så å skilje ut objektet og «tracke» det med «cone-cell». Figur 49 syner ei oversikt over systemet.



Figur 49: Objekt deteksjon ved mediumnivå fusjon: Objekt blir detektert med ein millimeter bølgelengde radar(MMW) og gjev region av interesse til kamera for vidare deteksjon. Det samla resultatet frå begge sensorane er eit detektert køyretøy. Figuren er henta frå [83]

[84] nyttar radar og monokamera for å lese ut informasjon frå scena. Ved å nytte avstandsinformasjonen frå radar blir objekt som er registrert frå radaren plotta i bilde frå kamera. ROI blir definert utifrå punkt som radaren har definert som objekt, og for å finne eigenskapar i bilde er det nytta ein HOG-deskriptor. Deretter blir objekt i scena klassifisert som køyretøy eller ikkje køyretøy. I denne rapporten er det kommentert at deteksjonsraten på 20Hz er avgrensa av radaren. [85] nyttar i staden eit høgnivå samarbeid mellom sensorane. Bakgrunnen er at det er vanskeleg å detektere ein bra nok ROI for deteksjon sidan radaren kan oppleve å berre detektere delar av køyretøyet. Ved å detektere objekt med radar og samtidig detektere køyretøy i frå bilde ved å clustre skuggane under køyretøy og nytte ein SVM-klassifiserer, så blir dei beste samanlikningane detektert som køyretøy. Dei resultata som blir forkasta blir dobbeltsjekka opp mot nye bilde og evaluert på nytt.

[86] nyttar ein lidar kombinert med monokamera for deteksjon av køyretøy. Bakgrunnen er lidar sine presise målingar og moglegheita for å finne eigenskapar i scena med kamera. Her samanliknar dei koordinatane på eigenskapspunkt til objektet, både i bilde og i 3D-punktskya frå lidar. Dei respektive koordinatane kan da transformerast frå bildekoordinatar til 3D-punktskya ved ei transformasjonsmatrise, og på same måte frå skya til bildekoordinatar. Vidare er HOG nytta for køyretøydeteksjon og eit djupt nevralt nett er brukt for å kjenne att bremselys på bremsande køyretøy. [87] utnyttar eigenskapane bak stereosyn i kombinasjon med ein lidar. Ved å utvide disparitetskartet til v-disparitet, som er introdusert på side 46 , blir vegen lokalisert og objekt og delar av scena som ikkje er på vegen blir fjerna. Vidare blir informasjonen frå begge sensorane nytta for å detektere køyretøy i vegen, og vidare tracka med kalman-filter.

3.6 Standarar i litteraturen

Bakgrunnen for at det er så open forsking på dette feltet er den opne delekulturen som er mellom dei ulike forfattarane. Mange databasar for både resultat, bilde og algoritmar er opent for alle slik at det er mogleg å teste og forbetre dei ulike løysingane. Mellom 2005-2012 har Pascal Challenge hatt årlege konkurransar for å skape den beste detektoren. Vidare har ImageNet tatt over stafettpinnen, og er å sjå på som «state-of-art» av generelle databasar.

Sjølv om desse datasetta er offentleg tilgjengelege er det framleis vanleg å konstruere eigne videodata som passar betre til eigne formål. Både infrastruktur og køyretøy kan variere over heile verda. Til samanlikning vil det vere mykje meir køyrefelt med tunellar og ferjer i Noreg enn kva det vil vere i USA.

3.6.1 Databasar

Sjølv om bilfabrikantane sine løysingar er proprietær så er det firma som nyttar seg av offentlege databasar slik at det er mogleg at ulike personar kan forske på same datasett. Nokre rapportar innhentar eigne datasett med eigne kamera, men eit utsnitt av dei mest brukte datasetta er synt i Tabell 4.

Tabell 4 Oversikt over mykje brukte opne datasett med bilde av køyretøy i trafikk.

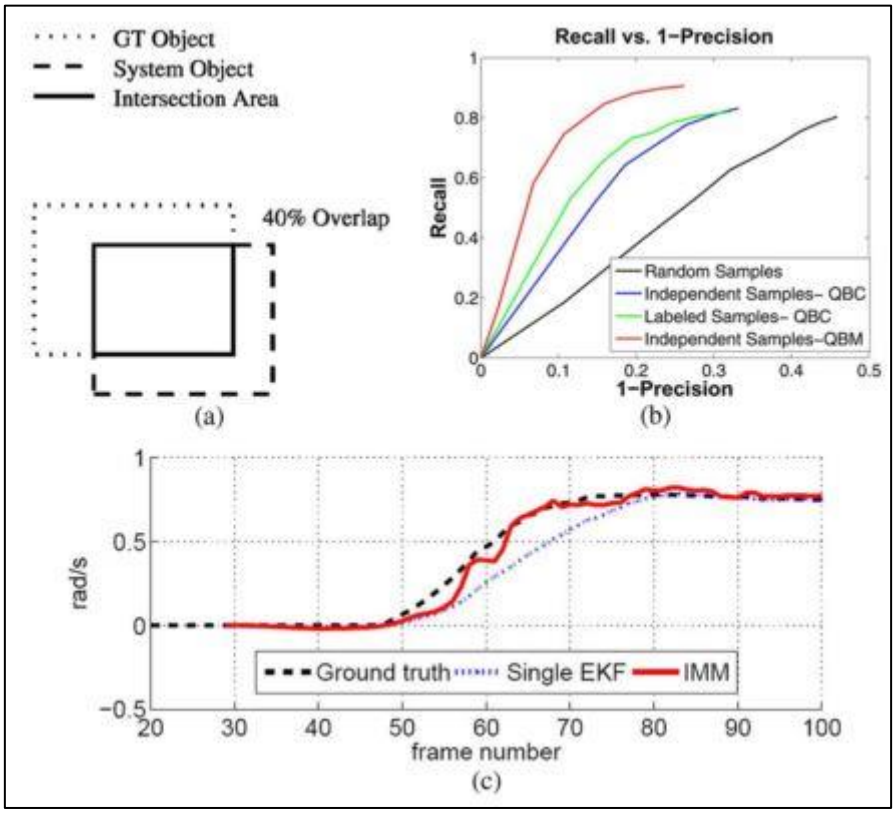
Datasett	**Beskriving**	**Kommentar**
Caltech 1999 & 2001[88]	Bilder av bakdelen av køyretøy i trafikk.	652 bilder i 360x240 pixlar. Bildene er frå ein motorvei i California.
PETS 2000[89]	Bilde av personar og køyretøy utandørs.	Treningssekvens med 3672 bilde og ein testsekvens med 1452 bilde. Bildene er på 768x576 pixlar
PETS 2001[90]	Bilde frå to kamera. Video frå innsida av eit køyretøy framover og bakover.	2867 bilderammer
LISA Vehicle Detection dataset[91]	Tre fargevideoar på motorveg og i urbane omgjevnadar. Videoane er i ulike settingar slik som morgon, kveld, solskinn, overskya, osv.	Utarbeidd av [28].
TME Motorway Dataset[92]	Videoar av ulike køyretøy på motorveien i Nord-Italia. Over 30 000+ bilderammer.	Stereobilde på 1024x768 pixlar. Estimat av ego-motion, køyretøy annotering og kjeldekode. Dette datasettet er spesielt laga for å nyttast til forsking. For å få tilgang må ein kontakte forfattarane i [93].
KITTI, 2015[94]	200 trenings-scener og 200 test-scener.	Resultat på ulike algoritmar.
Pascal VOC[95]	20 ulike klassar og 11 530 bilde.	Årleg konkurranse mellom 2005-2012
ImageNet[96]	1000 klassar og 1 461 406 bilde	Årleg konkurranse som starta opp i 2010.

3.6.2 Målingar

For sann -og falskdeteksjon i køyretøydeteksjon er det ulike standarar som er nytta[5]. For deteksjon av køyretøy er det vanleg å registrere sann-positiv og falsk-positiv måling. Gitt eit område som skal detekterast i ei ramme G_i, og deteksjonsområde D_i, så er deteksjonen sann-positiv dersom overlappinga er over eit fast terskling T:

$$D_i = \begin{cases} Sann\ positiv, dersom\ \dfrac{G_i \cap D_i}{G_i \cup D_i} > T \\ Falsk\ positiv, \quad ellers \end{cases}$$

Figur 50 illustrera ulike standarar som er nytta.



Figur 50 Standarar for måling: (a) Sann eller falsk deteksjon basert på overlapping. (b) Plott av recall og 1-presisjon. (c) orientering og tids-måling for stereosyn, kombinert med sann måling. Figuren er henta frå *[5]*

For stereosyn er det i tillegg fokus på nøyaktigheita på målingane for posisjon, orientering og liknande. Tabell 5 syner typen berekningar som er brukt for deteksjonar ved kamera.

Tabell 5 Berekningsmetodar for køyretøydeteksjon

Berekningar	Definisjon
Deteksjonsrate/Sann-positiv/Recall	$\dfrac{Tal\ Sann\ Positiv}{Tal\ Køyretøy}$
Presisjon	$\dfrac{Tal\ Sann\ positiv}{Sann\ Positive + Falsk\ Positive}$
Falsk positiv	$\dfrac{Antal\ Falsk\ Positive}{Sannsynlege\ deteksjonsboksar}$

Falsk positiv rate/ 1-Precision	$\frac{Antal\ falsk\ positive}{Sann\ Positive + Falsk\ Positive}$
Falsk Positive per Ramme/Falsk Positive per Bilde	$\frac{Falsk\ Positive}{Antal\ Rammer}$
Stereosyn	
Gjennomsnittleg absolutte feil	$\frac{1}{N}\sum\lvert x_{estimert} - x_{sann}\rvert$
Standard fordeling av feil måling	$\sqrt{\frac{1}{N}\sum(x_{estimert} - x_{sann})^2}$

4 Resultat frå dei ulike studia

Dei fleste studiar sett ei nedre grense på deteksjonsraten ved 10Hz, som tilsvara ein deteksjon kvar 3. meter i ca. 112km/t[58][97]. Allereie ved denne responstida er det mogleg å forbetre trafikksikkerheita i trafikken sidan den gjennomsnittlege reaksjonstida for ein bilførar er ca. 1 sekund (1Hz). Sjølv om det menneskelege synet kontinuerleg overvakar og identifisera objekt, er det reaksjonsevna som avgjer utfallet av ein uføresett situasjon. På bakgrunn av litteratursøket som er gjort i denne oppgåva, er det interessant å sette dei ulike studiane opp mot kvarandre. Tabell 6 syner resultat av funna, frå kapitla 3.4.1 og 3.5, i ei kronologisk rekkefølge der resultata er delt inn i monosyn, stereosyn og fusjon av sensorar. Resultat på deteksjonsfrekvens, sann positiv og falsk positiv er for nokre av studia det beste målte resultatet, sidan det er eksperimentert med fleire alternativ i nokre av dei.

Tabell 6 Oversikt over rapportar som har forska på køyretøydeteksjon

Monosyn						
Årstal	**Studie**	**HG**	**HV**	**Deteksjons-frekvens**	**Målingar**	**Kommentar**
2006	[58]	PCA, wavelets, Gabor.	SVM & Neurale Net	10Hz	Sann positiv: 94,6% Falsk positiv: 3,46%	NN. gav dårlegare resultat enn SVM. Beste løysing var Gabor-eigenskapar med SVM.
2008	[57]	HOG & HAAR	Adaboost	2,5Hz	Sann positiv: 93.5% Falske alarmar per bilde: 0,00032	Rapporten testar ulike kombinasjonar av detektorar
2010	[28]	HAAR	Adaboost	n/a	Sann positiv: 99% Falsk positiv: 8,5%	Systemet har blitt testa over fleire datasett ved ulike forhold.
2011	[53]	HOG & HAAR	SVM & Adaboost	n/a	Sann positiv: Ca. 77% 1-presisjon: 0.1%	Denne rapporten har eit fokus på å finne den beste klassifiseringsmetoden basert på to eigenskapsalgoritmar.
2011	[59]	SIFT	Hidden Random Field	n/a	1)Sann positiv: 92,9% 2)Sann positiv:85,9%	1) Total måling 2) Måling for delvis skjulte køyretøy
2012	[52]	Symmetri (Canny)+ clustering	HOG + SVM	12,5Hz	Sann positiv: 94,6% Falsk positiv: 0,1%	Her er eigenskapar som symmetri nytta for å finne ROI. HOG blir nytta for å finne betre eigenskapar i eit mindre område før klassifisering.
2012	[54]	Eigenskapane til ulike delar av køyretøy med HOG	L-SVM	2,8Hz	Sann positiv: Ca. 97% Falsk positiv: Ca. 0,26%	Løysinga med HOG i denne rapporten er for tidkrevjande.
2012	[60]	Kantdeteksjon & SURF	Sannsynsklassifisering	0.48Hz	Sann positiv: 85%	Deteksjon i blindsoner. Omfattande kalkulering gjev veldig dårleg deteksjonsfrekvens.
2015	[27]	HAAR	Utviding av Adaboost	n/a	Sann positiv: 97,4% Falsk-positiv: 3,22%	Her er fokuset på å nytte dei kjappe HAAR-eigenskapane for å få ned tida på

						trening. Samanlikna resultat på felles database med «»state-of-art».
2015	[31]	Ferdig tillaga datasett	HOG+SVM	45Hz	Sann positiv: 98,4% Falsk positiv: Ca. 0,01%	Rapporten testar skilnaden mellom ulike klassifiserer med HOG og Gabor
2015	[55]	Søk med eit glidande vindauge basert på størrelse	Pi-HOG + SVM	4-11Hz	Oppgitt grafisk. Pi-Hog med SVM genera best resultat.	Samanliknar ulike eigenskapsdetektorar med ulike metodar for ROI, på fire ulike databasar
2015	[61]	Deteksjon av veg	HAAR + SURF + AdaBoost	Ca. 47Hz	Sann positiv: 91,2% Falsk positiv: 13,6%	Er avhengig av god lyssetting, og kan feilklassifisere dersom feil vegbane er detektert.
2016	[62]	Visual saliency	Deep learning + SVM	31Hz	Sann positiv: 98,2% Falsk positiv: 0,78%	Rapporten nyttar eigenskapane bak djupe neurale nett for ein kjapp deteksjon.
Stereosyn						
Årstal	**Studie**	**HG**	**HV**	**Deteksjons-frekvens**	**Målingar**	**Kommentar**
2002	[64]	V-disparitet	Leiter etter kontaktpunkt mellom veg og dekk.	25Hz	Kontaktpunkt mellom køyretøy og veg	Nyttar v-disparitet for å minimere søket i ei scene.
2005	[20]	Stereomatching ved lagvise bilde	Objekt segmentering i disparitetskartet	12Hz	Avstand og størrelse på køyretøy	Målar avstand og størrelse på objekt ved å segmentere ulike objekt i fleire lag.
2005	[68]	Samanlikne objekt i scena med ei mal	Bayesian klassifiserer	n/a	Sann positiv: 84% Avstandsmåling	Skil mellom køyretøy og personar i scena.
2009	[69]	Objektdeteksjon i disparitetskartet.	Clustering	n/a	Køyretøyets orientering og avstandsmåling	Estimera køyretøyets orientering vha. clustering i disparitetskartet.
2009	[70]	Stereomatching i disparitetskartet + clustering	HOG-Adaboost	16Hz	Sann positiv: 99% Falsk positiv: 1,77% Avstand til køyretøy	Ved å dele inn scena i fleire regionar, aukar kalkuleringshastigheita.
2010	[66]	U-disparitet	Clustering+ Bayesian filter	n/a	Avstandsmåling	Ikkje oppgitt nokre direkte målingar
2010	[71]	Detektert i eit hovudkamera	Identifisert i kamera nr. to i 3D-rommet.	Rapporten er ikkje spesifikk. Nemner at deteksjon er i sanntid.	Avstandsmåling	Rapporten presentera stereoassistanse
2015	[67]	DCNN	U -og v-disparitet	Ca. 16Hz	Sann positiv: 98,86% Falsk positiv: 0,00015%	Utnyttar eigenskapar frå monosyn i eit DCNN for så å verifisere køyretøyet i disparitetskart.

Fusjon av sensorar						
Årstal	**Studie**	**HG**	**HV**	**Deteksjons-frekvens**	**Målingar**	**Kommentar**
2005	[87]	Stereosyn: v-dis-paritet, lidar	Clustering	25Hz	Sann positiv 100% Falsk positiv: 1,2% Avstand	Objekt blir detektert ved begge sensorane, og falske deteksjonar blir luka ut.
2011	[85]	Kamera: Skugge-detek-sjon + radar	SVM	15Hz	Sann positiv: 96,5% Falsk positiv: 1,1% Avstand	Objekt som er feilklassifisert blir re-klassifisert for å minimere feilklassifisering.
2014	[83]	Radar	Kamera: symmetri	2Hz	Sann positiv: 99,5% Falsk positiv: 0,86%	Samanliknar systemet med den menneskelege synet.
2015	[84]	Radar	Kamera: HOG + clustering	20Hz	Sann positiv: Ca. 82% Falsk positiv per bilde: Ca. 0,12	Rapporten viser til ganske dårleg attkjenning frå ka-mera, dei føreslår ei utvi-ding med deep learning.
2016	[86]	Kamera + lidar	HOG + attkjen-ning av bremselys med DNN.	27Hz	Sann positiv: 99% Falsk positiv: -- Avstand	Rapporten har bra resultat og føreslår å nytte deep learning for å forbetre køy-retøy-deteksjonen enno meir.

5 Diskusjon

Køyretøydeteksjon har hatt eit aukande fokus i fleire år. Arbeidet blant dei kommersielle bilfabrikantane på å utvikle autonome bilar har gjort til at objektdeteksjon og attkjenning av køyretøy har fått eit større fokus dei siste åra. For at autonome bilar skal bli ein realitet må presisjonen på deteksjonen vere optimal i ulike scener og miljø. På bakgrunn av tidlegare arbeid og resultat ifrå dei ulike rapportane blir det her diskutert rundt dei ulike funna og kva som vil vere fokus framover.

5.1 Deteksjon med monosyn

Objektdeteksjon for monokamera har hatt nokre vendepunkt dei siste åra. I kapittel 3.4.1 blir det presentert tidlegare arbeid der enkle metodar for å kjenne at faste mønster i ei scene har utvikla seg til å bli meir robuste og generelle. Tabell 6 syner ei oppsummering av arbeidet frå dette kapittelet med målingar og kommentarar. Tidlegare eigenskapar som symmetri og kantdeteksjon har utvikla seg til meir direkte eigenskapar som HOG og Haar. Dette er ei naturleg utvikling sidan eit køyretøy er lettare å finne i ei scene når det har spesifikke eigenskapar. Ved å sjå på symmetri og kantdeteksjon for seg sjølv, kan det oppstå fleire falsk-positive deteksjonar sidan andre objekt som bygningar og skilt kan ha den type eigenskapar. I verste fall så kan ein enkelt feildeteksjon føre til at køyretøyet bremsar og/eller tar ein uføresett manøver. Hovudbakdelen med monosyn er den manglande eigenskapen for å måle djupne i ei scene som gjere det vanskeleg å nytte aleine.

For sjølve deteksjonen er det i denne rapporten presentert nokre algoritmar som er favorisert i feltet for køyretøydeteksjon ved monosyn. Dei basera seg på lokale og globale eigenskapar i bilde, og krevjar ulik reknekraft:

HOG: Basera seg på globale eigenskapar og dannar ein eigenskapsvektor basert på lengde og orientering av gradientane i eit bilde. Algoritmen er også invariant mot orientering og illuminasjon, men kan vere tung i ei utrekning. I tillegg er eigenskapsvektoren, som kjem frå gradienten i eit bilde, generell og fortel ikkje noko om posisjonen til gradienten. Dette kan føre til at dei ulike cellene i bildet kan ha identiske retningshistogram. I seinare tid har ulike variantar i kombinasjon med ein GPU gjort denne til ein føretrekt metode. Frå kapittel 3.4.1 er den alternative pi-HOG algoritmen presentert[55]. Denne studien presentera ein utvida metode som gjer informasjon om gradientens posisjon og intensitet i eit bilde. Det viser seg at den er tyngre i ei utrekning men gjer betre resultat.

HAAR: Desse eigenskapane har synt seg å ha den kjappaste algoritmen som er presentert her. Haar kjernane er veldig sensitive til horisontale og vertikale intensitetsforandringar og oppnår ei kjapp utrekning ved bruk av integralbilde. Kjernane som blir brukt til å finne eigenskapane i eit bilde kan og bli utrekna ved ulik skala og posisjon. Algoritmen har og vist at den framhevar mange falsk positive målingar, og er derfor heller ein kandidat til hypotesegenerering.

Gabor: Nyttar seg av prinsippet bak wavelets og retningsbasert histogram. Filteret skapar stabile eigenskapar men er tung i ei utrekning. [31] viser at Gabor har dårlegare resultat enn HOG med fleire falsk positive deteksjonar, og er ikkje ideell å bruke aleine.

SIFT: Algoritmen er invariant mot skala, translasjon og illuminasjon som gjere dette til ein veldig sterk og stabil algoritme. [59] syner at ved å trene på ein database med køyretøy, for så å teste på bilde med delvis skjulte køyretøy blir ein stor del av dei delvis skjulte køyretøya detektert. Algoritmen er vidare kompleks i utrekning av eigenskapar, noko som har gjort den dårleg for sanntidsapplikasjonar.

SURF: Denne er basert på trekk i frå Haar og SIFT og var utvikla for å ha ei algoritme som var kjapp i ei utrekning som samtidig skulle finne unike og sterke eigenskapar. Algoritmen er invariant mot skala og rotasjon men er framleis tyngre i ei utrekning i forhold til HOG og Haar.

Det er vidare ein trend i miljøet ved å kombinere ulike eigenskapsdetektorar for å optimalisere deteksjonen og minimere falsk positive deteksjonar, men eigenskapane er framleis vanskelege å detektere ved dårleg lyssetting. Under deteksjon i mørke scener blir køyretøy oftast detektert ved å sjå på lys frå køyretøyet som skal bli detektert.

Dei presenterte algoritmane er nokre av dei mest omtalte i litteraturen. Optimalt bør eit slikt system ha 100% sann positive deteksjonar med ingen falsk positive. Dette er ei veldig vanskeleg problemstilling som er avhengig av mange ulike parameter. Dei siste åra har det blitt sett på alternative metodar for å trekke ut eigenskapar. [62] presentera ei ny løysing som basera seg på konseptet «visual saliency»[63] for hypotesegenerering og eit djupt nevralt nett for å trekke ut eigenskapar for verifisering. Resultatet blir samanlikna med state-of-art arbeid for køyretøydeteksjon[58],[70]. Eigenskapane frå bildet er då funne med det nevrale nettet og klassifisert med ein ulineær SVM.

5.2 Deteksjon med stereosyn

For stereosyn har det vore naturleg å utnytte eigenskapane i disparitetskartet. Tidlegare studiar fokusera på orientering og avstand til objekt i scena, og har ingen direkte resultat på deteksjonsrate før i seinare tid. Tabell 6 syner ei oversikt over arbeid som er diskutert i kapittel 3.4.1.

Sivaraman et al.[5] diskutera rundt det at stereosyn er mest brukt i rørslebasert attkjenning sidan 3D modellen frå systemet gjev gode moglegheita for «tracking» i rommet. Det har i denne rapporten vore hovudfokus på utsjåandebasert attkjenning, men kapittel 3.4.2 presentera nokre eksempel på rørslebasert attkjenning der optisk flyt blir brukt for å estimere bevegelse på objekt.

Hovudproblemet med å nytte stereosyn er dei tunge utrekningane som oppstår når disparitetskartet skal estimerast. I kapittel 2.3.3 blir tre hovudmetodar for utrekning av disparitet presentert. På bakgrunn av dette er det tydeleg ein popularitet for ein semi-global metode for å estimere ein global energifunksjon, som er ein kombinasjon av dei tidlegare lokale og globale metodane. Den presentera eit tydelegare disparitetskart enn den lokale metoden, og er kjappare i ei utrekning enn den globale. Alternativt til disparitetskartet er det presentert arbeid med IPM, som er forklart i kapittel 2.1.2. Denne metoden nyttar informasjonen frå eit 3D rom og legg punkt i rommet horisontalt på eit plan for å skilje ut objekt. Ved å nytte denne metoden er det ikkje naudsynt med eit disparitetskart, men det set krav til at vegbana er forholdsvis rett for å få gode resultat.

Estimering av kameraparameter under køyring er også utfordrande. Ujamn vegbane, vind og liknande kan føre til at stereoparameterane blir forandra. I kapittel 3.4.3 blir det presentert arbeid der kameraparameterane blir kontinuerleg kalibrert og justert. Dette er da ei løysing som vil gå på kostnad av reknekraft.

Ved å nytte stereosyn i utsjåandebasert attkjenning har det vore populært å sortere ut objekt i vegbana ved å gruppere eigenskapar, eller vegen kan bli segmentert for å minimere ROI. Nokre studiar presentera metodar for å skape eit oversiktsbilde ved å nytte IPM som ei alternativ løysing. Felles for desse metodane er fordelen av å detektere køyretøy og i tillegg måle avstand til objekt. Ein metode som er sett på som «state-of-art» innanfor stereosyn er presentert i [70]. Denne er strengt tatt ein fusjon av mono – og stereosyn sidan HG blir generert i disparitetskartet ved å sjå på områder med konstant disparitet, der bakdelen av eit køyretøy hamnar i denne kategorien, og falsk positive blir minimert ved å nytte ein HOG-Adaboost kombinasjon. Men også i stereosyn er det dei siste åra sett på alternative

metodar for å detektere køyretøy. [67] finn HG ved å trene eit djupt nevralt nett til å lokalisere eigenskapspunkt og verifisera køyretøy ved å studere scena i eit disparitetskart. Desse rapportane presentera det hittil beste resultatet for køyretøydeteksjon ved hjelp av stereosyn.

5.3 Deteksjon med ein fusjon av sensorar

I kapittel 3.5 blir det presentert arbeid med sensorfusjon. Ein slik kombinasjon av ulike sensorar har blitt meir vanleg i seinare tid. Auka etterspurnad etter radar -og lidarsensorar har gjeve ein prisreduksjon som gjere dei meir aktuelle. Ved å nytte ein kombinasjon av sensorar vil dei ulike sensorane utfylle kvarandre, og deteksjonen vil bli betre ved hurtig varierande scener.

Resultata frå Tabell 6 syner at deteksjonen frå dei ulike studia også er forhaldsvis bra. Kvart studie presentera at deteksjonen blir vesentleg forbetra ved å kombinere dei ulike sensorane, og at det er ulike svakheiter for kvar av dei:

Radar: Ikkje så påverka av ytre forhold som lidar og kamera, men har generelt mykje støy i sine målingar som kan skape ein del falsk positive målingar. Kan heller ikkje detektere eit køyretøy dersom det står i ro.

Lidar: Skapar nøyaktige punktskyar av omgjevnaden og har gode målingar, men kan ikkje skilje mellom objekt. Er også sensitiv under scener med regn.

Kamera: Er god på å skilje mellom objekt, men kan oppleve ein del falsk positive målingar ved hurtig skiftande lyssettingar, vêrforhold og liknande.

Rapportane syner at system med sensorfusjon er meir stabil for varierande scener sidan det er mogleg å trekke ut dei beste eigenskapane frå kvar sensor. Slike system vil da ikkje ha same avgrensingar som eit system som er basert på kamera aleine. Nokre av dei nyare rapportane diskutera at kombinasjonen mellom radar, lidar og kamera kan forbetrast ved å nytte djupe nevrale nett[84], [86].

5.4 Køyretøyattkjenning i fleire steg

Ein to-stegs prosess ved å generere eit område av interesse (HG) for så å verifisere køyretøy i scena (HV) er ein mykje brukt metode for køyretøydeteksjon[4], [5]. Etter å ha funne eit område av interesse blir søket etter køyretøy i ein mykje mindre del av bildet, og behandlingstida går ned. I kapittel 3.4.1 blir det presentert arbeid med kamerasyn der ulike metodar blir brukt for å finne HG og HV, og i kapittel 3.5 blir det presentert arbeid med ein fusjon av sensorar med ulike metodar for å finne HG og HV.

I monosyn har metodar for å finne HG gått ifrå symmetri og kantdeteksjon til meir generelle algoritmar, som er diskutert i kapittel 5.1. Motivasjonen har vore at dersom eit køyretøy blir mista i HG steget, så blir det heller ikkje detektert i HV steget. Denne faren er minimert dersom køyretøya har distinkte eigenskapar, og gjer det mogleg å hoppe over HG steget.

For stereosyn blir HG steget utført i eit disparitetskart. Ved å segmentere veg eller å «clustre» objekt i disparitetskartet kan køyretøya bli kjent att i scena. «State-of-art» systema viser likevel at å kombinere stereosyn med eigenskapar frå monosyn skapar betre deteksjonar[70].

For fusjon av sensorar er det vanleg å parallelt detektere objekt i dei ulike sensorane, og vidare klassifisere køyretøya på ein vekta sum mellom sensorane, og definere område av interesse for ein sensor med ein anna. Faren ved å miste eit køyretøy i HG steget blir da minimert.

5.5 Sanntidsdeteksjon

Fokuset på sanntids-system vore eit viktig tema. I eit autonomt køyretøy tar ein vekk den menneskelege oppfatningsevna, og må då erstatte det med eit system som overvakar miljøet og reagera på same måte. Litteraturen viser til ein minimum deteksjonsrate på 10Hz som tilsvara ein deteksjon for kvar 3. meter i ca. 110km/t. For å nå dette målet er det mange parameter som må stemme overeins. I litteraturen er no bruken av GPU-ar meir aktuell sidan dei blir billegare og kan utføre utrekningar på parallelle prosessorar samtidig. Samtidig så er deteksjonsraten avhengig av at deteksjonsalgoritmane oppdaterast i ein rate som er kjapp nok.

5.6 Kommentar på resultat

Det som er interessant å sjå på i denne rapporten er presisjon og hastigheit på dei ulike systema. For monosyn og fusjon av sensorar har alle rapportane resultat på sann-detekterte køyretøy, medan rapportar for stereosyn har hovudsakleg fokusert på avstand, størrelse og orientering på objekt. Det er først i nyare tid at rapportar med arbeid på stereosyn viser til ein presisjon på deteksjonar.

Vidare så kan det stillast spørsmål om dei resultata som synar ei oversikt over presisjon og deteksjonsrate eigentleg kan samanliknast sidan det er nytta ulike datasett i kvart studie. Optimalt så burde alle desse metodane ha vorte testa på det same datasettet for å kunne skilje ut kva som er den beste løysinga. Dei ulike studia som er presentert i rapporten har testa dette mykje på sin eigne måtar der nokre av dei går over fleire like datasett.

Rapportens hovudfokus har vore køyretøydeteksjon i køyrefelt framføre køyretøy på dagtid. Det har vore presentert heilt kort ulike løysingar på deteksjon i mørke, men sidan dette ikkje har vore hovudfokus har det ikkje blitt grundig undersøkt. Men til og med ved dagtid vil det oppstå ulike dårlege situasjonar for deteksjon. Tåke, kraftig regn, tunellar og gjenskin i køyrefelt er nokre situasjonar som skapar usikkerheit i trafikken. Deteksjon under slike forhold blir mykje forbetra ved å kombinere sensorar og nytte ulike algoritmar for deteksjon, til dømes klassifisering på bakgrunn av baklys og liknande.

5.7 Vegen framover

I løpet av dei siste åra er det presentert fleire ulike løysingar på køyretøydeteksjon. Sun et al. [4] sitt litteratursøk konkluderte i 2006 med at datidas maskinvare, programvare og algoritmar ikkje var gode nok for at ein autonom bil kunne operere på eiga hand. Det var sannsynleg med køyretøyassistanse, men å fullt erstatte sjåføren var høgst usannsynlig. Sivaraman et al. [5] konkluderte i 2013 med at feltet for køyredeteksjon har hatt ein enorm vekst, men er også eit så utfordrande tema at det er ulike ting som enno må forbetrast. Det blir nemnt at fokuset framover bør gå meir mot maskinlæring for meir effektive klassifiseringar, og i den samanhengen bør også systemet vere meir basert på ein fusjon av sensorar for å heve robustheita. På bakgrunn av litteratursøket som er lagt fram i denne masteroppgåva blir det her presentert løysingar som bør vere med i eit køyretøydeteksjonssystem.

For eit sanntids deteksjonssystem vil det vere naturleg å nytte ein fusjon av monokamera med både radar og lidar for å optimalisere deteksjonen i ulike miljø. Hypotesegenerering ved å nytte mediumnivå fusjon der radar og/eller lidarsensorane detektera objekt i vegbana og gjev informasjon om region av interesse til eit kamera for vidare deteksjon vil skape sikre målingar. For å vidare kontinuerleg oppdatere deteksjonen vil eit høgnivå samarbeid med ei vekta klassifisering mellom sensorane gje ein sikrare deteksjon. I denne samanheng kan bruken av stereokamera sjåast vekk frå sidan avstand vil bli lest ut frå radar -og lidardata. Den tunge utrekninga som bruken av eit stereosystem krevjar kan då bli sett vekk ifrå.

Sidan scena for eit køyretøy er hurtig skiftande må deteksjonsalgoritmen for eit monosyn vere invariant mot rotasjon, skala -og illuminasjon. Det naturlege valet vil vere ein kombinasjon av HOG, Haar, SIFT og/eller Surf liknande eigenskapar med ein kombinasjon av nyare utvikla GPU-ar for utrekningar. Alternativt bør det undersøkast meir i retning av eigenskapsutrekning med DNN. Sjølv om ein radar-lidar fusjon vil skape betre deteksjonar i mørket, så bør også deteksjon under dårlege lyssettingar undersøkast meir. Men løysingane som er presenterte ved å sjå på baklys vil elles vere eit naturleg val.

Tidlegare litteraturstudie viser til at nevrale nett kjem til kort i forhold til andre klassifiserarar som SVM og Adaboost grunna for mange parameter og at resultata har ein tendens til å konvergere mot eit lokalt optimum[5]. I nyare tid har nevrale nett på mange måtar gjenoppstått i ei ny og betre form. Mykje av dette kan visast til eit gjennombrot som vart gjort i regi av ImageNet si konkurranse i 2012[96]. Fokuset har retta seg meir mot arbeid med djupe nevrale nett både for klassifisering, men også for eigenskapsutrekning. Bakdelen med denne angrepsmåten er eit behov for store mengder med data. Eksempelvis så har Tesla latt sine konvensjonelle køyretøy samle data i fleire år, og har da kunne trene opp sine DNN for å klassifisere køyretøy.

Noverande system for køyretøyattkjenning ligg an til å fungere veldig bra på rette motorvegar med tydelege markeringar utan for mykje detaljar i scena. Arbeidet framover kjem til å fokusere på vanskelege scener som eksempelvis vil oppstå på vegbanar i utkanten av urbane strøk, samt den tette trafikken i bymiljø. I utkantane er det ofte ingen vegmerking, mange tunellar, og to køyreretningar i same vegbane. I bykjernane er det ofte mykje køyretøy, bygningar og vegkryss som må takast omsyn til.

5.8 Vidare arbeid

Denne rapporten dannar eit grunnlag for kva retning system for køyretøydeteksjon har hatt dei siste åra. I den forstand er eit vidare litteratursøk ikkje særleg hensiktsmessig. Rapporten kan verte brukt som eit grunnlag for ei oppgåve innanfor deteksjon av køyretøy, der det i kapittel 3.6.1 blir presentert opne databasar som ville vore naturleg å bruke i ei slik oppgåve. I den samanheng blir det presentert nokre alternativ til oppgåver:

Søk etter eigenskapar frå bilde med djupe nevrale nett
Populariteten for djupe nevrale nett har vore aukande dei siste 5 åra. Det er tydeleg at feltet for køyretøyattkjenning har vendt seg etter automatiske metodar for å finne og klassifisere eigenskapar frå bilde. Det vil da være interessant å undersøke om ei slik løysing kan overgå dei tradisjonelle algoritmane i både hastigheit og presisjon.

Optimalisere eigenskapsalgoritmar for deteksjon under dårleg opplyste scener
Deteksjon under dårleg opplyste scener har vore løyst ved å sjå på baklys ved konvensjonelle kamera eller å sjå på scena i ulike spekter. Her vil det være interessant om det hadde vore mogleg å finne eigenskapar frå kamera som er unike for slike scener.

6 Konklusjon

Hovudmålet med denne rapporten var å trekke fram eksempel på dei fremste metodane for køyretøydeteksjon som har vore brukt dei siste åra, med eit hovudfokus på kamera. Frå desse eksempla skal det presenterast den løysinga som ser ut til å vere den beste i dag.

Arbeidet som er presentert viser at eit føretrekt system detektera eigenskapar frå eit køyretøy med ein passiv sensor som kamera, og kontinuerleg overvakar scena med aktive sensorar som radar og lidar. Køyretøyets eigenskapar blir funne med kamera, og avstandsmåling og deteksjon under mørke scener blir forbetra av dei aktive sensorane.

Deteksjon med monokamera har gått frå enkle metodar som nyttar seg av symmetri og kantdeteksjon til meir generelle eigenskapar. Komplekse og hurtig skiftande scener i trafikken gjere til at køyretøya bør ha unike eigenskapar som er enkle å finne. Algoritmane må i ei slik scene vere invariante mot rotasjon, translasjon og illuminasjon, og har i den samanheng utvikla seg til ulike variantar av algoritmar som HOG og Haar. Dei siste åra har trenden gått mot djupe nevrale nett for eigenskapsutrekning. Bakgrunnen er dei eksepsjonelle resultata som denne type nettverk har oppnådd i ulike konkurransar for bildeattkjenning. Dette fokuset på eigenskapar frå monosyn har også gjort til at stereosyn aleine ikkje opplever like gode resultat, og er å føretrekke med ein kombinasjon av monosyn. Til samanlikning med aktive sensorar har stereosyn ikkje like god presisjon på målingar for avstand og krevjar desto tyngre prosessering.

Systema som blir brukt i dag fungera bra på motorvegar som har minimalt med varierande detaljar, men det må framover fokuserast på attkjenning i områder som er å finne både utanfor og innanfor urbane strøk. Der er scenene veldig varierande og køyreretningane er ofte i same vegbane.

Å sette saman eit litteratursøk av dette formatet er ganske omfattande, og det er vanskeleg å dra ein strek for når oppgåva kan avsluttast. Mykje av grunnen for dette er at problemstillinga rundt autonome bilar er i ein fase der utvikling av program -og datavare er i ein enorm vekst, og fokuset på autonome bilar er eit av dei varmaste tema i teknisk litteratur. Denne rapporten viser ikkje til noko nye metodar for objektdeteksjon og klassifisering, men gjer ein peikepinn på kva retning forskinga vil ha framover. Konklusjonen er at måla for oppgåva er nådd, og at rapporten gjev ei oversikt over kva som har vore fokus dei siste åra.

Bibliografi

[1] A. Vance, “The First Person to Hack the iPhone Built a Self-Driving Car. In His Garage,” *Bloomberg businessweek*, 2015. [Online]. Available: https://www.bloomberg.com/features/2015-george-hotz-self-driving-car/. [Accessed: 30-Jan-2017].

[2] World Health Organization, “Global Status Report on Road Safety 2015,” *WHO Libr. Cat. Data Glob.*, p. 340, 2015.

[3] Marius Valle, “Tesla Autopilot frikjent etter dødsulykke,” *Teknisk ukeblad*, 2017. [Online]. Available: https://www.tu.no/artikler/tesla-autopilot-frikjent-etter-dodsulykke/367917. [Accessed: 31-Jan-2017].

[4] Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection: A review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, 2006.

[5] S. Sivaraman and M. M. Trivedi, “Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, 2013.

[6] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, “Overview of Environment Perception for Intelligent Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–18, 2017.

[7] J. J. Antony and M. Suchetha, “Vision Based Vehicle Detection: A Literature Review,” *Int. J. Appl. Eng. Res. ISSN*, vol. 11, no. 5, pp. 973–4562, 2016.

[8] C. Bila, F. Sivrikaya, M. A. Khan, and S. Albayrak, “Vehicles of the Future: A Survey of Research on Safety Issues,” *IEEE Trans. Intell. Transp. Syst.*, vol. PP, no. 99, pp. 1–20, 2016.

[9] Samferdselsdepartementet, “Høring - Forslag til ny lov om utprøving av selvkjørende kjøretøy på veg,” 2017. [Online]. Available: https://www.regjeringen.no/no/dokumenter/horing---forslag-til-ny-lov-om-utproving-av-selvkjorende-kjoretoy-pa-veg/id2523663/. [Accessed: 01-Apr-2017].

[10] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Second edi. Pearson Education M.U.A, 2014.

[11] I. Austvoll, “Machine/Robot Vision Part 1,” *Lect. notes ELE510 Image Process. with Robot Vis.*, p. 113, 2015.

[12] J. Heikkilä and O. Silvén, “A Four-step Camera calibration procedure with Implicit Image Correction,” *Comput. Vis. Pattern Recognition, 1997. Proceedings., 1997 IEEE Comput. Soc. Conf. on. IEEE 1997*, vol. 98, no. 93, pp. 1106–1112, 1997.

[13] The Mathworks Inc, “What is camera calibration?” [Online]. Available: https://se.mathworks.com/help/vision/ug/camera-calibration.html. [Accessed: 01-Feb-2017].

[14] M. W. Spong, S. Hutschinson, and M. Vidyasagar, *Robot modeling and control*. Wiley, 2006.

[15] G. R. Mueller and H. J. Wuensche, “Continuous extrinsic online calibration for stereo cameras,” *IEEE Intell. Veh. Symp.*, no. IV, pp. 966–971, 2016.

[16] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, “Inverse perspective mapping simplifies

optical flow computation and obstacle detection," *Biol. Cybern. 64*, vol. 56, pp. 49–56, 1991.

[17] M. Bertozz, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: theory and applications," *Image Vis. Comput.*, vol. 16, pp. 585–590, 1998.

[18] Z. Zhang, "A Flexible New Technique for Camera Calibration (Technical Report)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[19] The Mathworks Inc, "Single Camera Calibration App." [Online]. Available: https://se.mathworks.com/help/vision/ug/single-camera-calibrator-app.html. [Accessed: 25-May-2017].

[20] C. Thompson, H. Yingping, and S. Fu, "Stereovision-Based Object Segmentation," *AppliedMathematics Comput. Group, Sch. Eng. Cranf. Univ.*, pp. 2322–2329, 2005.

[21] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer-Verlag London Limited 2011, 2011.

[22] D. G. Lowe, "Distinctive image features from scale invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, p. 28, 2004.

[23] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Procedings Alvey Vis. Conf. 1988*, pp. 147–151, 1988.

[24] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," *Int. Conf. Comput. Vision, IEEE*, pp. 555–562, 1998.

[25] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Comput. Vis. Pattern Recognit.*, vol. 1, pp. 511–518, 2001.

[26] P. Menezes, J. C. Barreto, and J. Dias, "Face tracking based on haar-like features and eigenfaces," *IFAC/EURON Symp. Intell. Auton. Veh.*, vol. 500, 2004.

[27] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295. pp. 395–406, 2015.

[28] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 267–276, 2010.

[29] N. Dalal and W. Triggs, "Histograms of Oriented Gradients for Human Detection," *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. CVPR05*, vol. 1, no. 3, pp. 886–893, 2004.

[30] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters.," *J. Opt. Soc. Am. A.*, vol. 2, no. 7, pp. 1160–1169, 1985.

[31] S. S. Teoh and T. Braunl, "Performance evaluation of HOG and Gabor features for vision-based vehicle detection," *Proc. - 5th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2015*, no. November, pp. 66–71, 2015.

[32] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006.

[33] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[34] M. Brown and D. Lowe, "Invariant Features from Interest Point Groups," *Br. Mach. Vis. Conf.*, pp. 656–665, 2002.

[35] D. Scharstein, R. Szeliski, and R. Zabih, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Stereo Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings. IEEE Work.*, 2001.

[36] H. Hirschmüller and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching," *2007 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, 2007.

[37] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," *Comput. Vis. Pattern Recognition, 2005. CVPR 2005. IEEE Comput. Soc. Conf.*, vol. 2, pp. 807–814, 2005.

[38] B. Tippetts, D. J. Lee, K. Lillywhite, and J. Archibald, "Review of stereo vision algorithms and their suitability for resource-limited systems," *J. Real-Time Image Process.*, vol. 11, no. 1, pp. 5–25, Jan. 2016.

[39] I. Haller, C. Pantilie, F. Oniga, and S. Nedevschi, "Real-time semi-global dense stereo solution with improved sub-pixel accuracy," *IEEE Intell. Veh. Symp. Proc.*, pp. 369–376, 2010.

[40] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[41] A. Ben-Hur and J. Weston, "A user's guide to support vector machines.," *Methods Mol. Biol.*, vol. 609, pp. 223–239, 2010.

[42] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, Massachusetts: MIT Press, 2002.

[43] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, pp. 119–139, 1995.

[44] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst. 25*, pp. 1–9, 2012.

[46] S. of A. Engineers, "SAE INTERNATIONAL STANDARD J3016," 2014. [Online]. Available: https://www.sae.org/misc/pdfs/automated_driving.pdf. [Accessed: 01-Feb-2017].

[47] S. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive Radars," *IEEE Signal Process. Mag.*, vol. 34, no. March, pp. 22–35, 2017.

[48] N. Garun, "Here's a first look at Uber's self-driving car," 2017. [Online]. Available: https://thenextweb.com/insider/2016/05/19/heres-first-look-ubers-self-driving-car/#.tnw_vUjWtVQ9. [Accessed: 02-Feb-2017].

[49] C. Palo Alto, "Ford targets fully autonomous vehicle for ride sharing in 2021," 2017. [Online]. Available: https://media.ford.com/content/fordmedia/fna/us/en/news/2016/08/16/ford-targets-fully-autonomous-vehicle-for-ride-sharing-in-2021.html. [Accessed: 22-Feb-2017].

[50] M. P. Hosur, R. B. Shettar, and M. Potdar, "Environmental Awareness Around Vehicle Using Ultrasonic Sensors," pp. 1154–1159, 2016.

[51] A. Klausner, S. Erb, A. Tengg, and B. Rinner, "Dsp Based Acoustic Vehicle Classification for Multi-Sensor Real-Time Traffic Surveillance," *15th Eur. Signal Process. Conf. (EUSIPCO 2007)*, pp. 1916–1920, 2007.

[52] S. S. Teoh and T. Bräunl, "Symmetry-based monocular vehicle detection system," *Mach. Vis. Appl.*, vol. 23, no. 5, pp. 831–842, 2012.

[53] S. Sivaraman and M. M. Trivedi, “Active learning for on-road vehicle detection: a comparative study,” *Mach. Vis. Appl.*, pp. 1–13, 2011.

[54] H. Tehrani Niknejad, A. Takeuchi, S. Mita, and D. McAllester, “On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 748–758, 2012.

[55] J. Kim, J. Baek, and E. Kim, “A Novel On-Road Vehicle Detection Method Using pi-HOG,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3414–3429, 2015.

[56] J. Zhang, K. Huang, Y. Yu, and T. Tan, “Boosted Local Structured HOG-LBP for Object Localization,” *2011 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010.

[57] L. Prevost, P. Negri, X. Clady, and S. M. Hanif, “A cascade of boosted generative and discriminative classifiers for vehicle detection,” *EURASIP J. Adv. Signal Process.*, p. 12, 2008.

[58] Z. Sun, G. Bebis, and R. Miller, “Monocular Pre-crash Vehicle Detection : Features and Classifiers,” vol. 15, no. 7, pp. 2019–2034, 2006.

[59] X. Zhang, N. Zheng, Y. He, and F. Wang, “Vehicle detection using an extended Hidden Random Field model,” *2011 14th Int. IEEE Conf. Intell. Transp. Syst.*, pp. 1555–1559, 2011.

[60] B. F. Lin *et al.*, “Integrating appearance and edge features for sedan vehicle detection in the blind-spot area,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 737–747, 2012.

[61] S. Sun, Z. Xu, X. Wang, G. Huang, W. Wu, and X. De, “Real-time vehicle detection using Haar-SURF mixed features and gentle AdaBoost classifier,” *Proc. 2015 27th Chinese Control Decis. Conf. CCDC 2015*, pp. 1888–1894, 2015.

[62] Y. Cai, H. Wang, X. Chen, L. Gao, and L. Chen, “Vehicle detection based on visual saliency and deep sparse convolution hierarchical model,” *Chinese J. Mech. Eng.*, vol. 29, no. 4, pp. 765–772, 2016.

[63] T. N. Vikram, M. Tscherepanow, and B. Wrede, “A saliency map based on sampling an image into random rectangular regions of interest,” *Pattern Recognition*, vol. 45, no. 9. pp. 3114–3124, 2012.

[64] R. Labayrade, D. Aubert, and J.-P. Tarel, “Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through ‘V-disparity’ Representation,” *Intell. Veh. Symp. IEEE*, vol. 2, no. January, pp. 646–651, 2002.

[65] A. Broggi *et al.*, “TerraMax vision at the Urban challenge 2007,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 194–205, 2010.

[66] M. Perrollaz, A. Spalanzani, and D. Aubert, “Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection,” *IEEE Intell. Veh. Symp. Proc.*, pp. 313–318, 2010.

[67] Y. Cai, H. Wang, X. Chen, and L. Chen, “Deep representation and stereo vision based vehicle detection,” *2015 IEEE Int. Conf. Cyber Technol. Autom. Control. Intell. Syst.*, pp. 305–310, 2015.

[68] P. Chang, D. Hirvonen, T. Camus, and B. Southall, “Stereo-Based Object Detection, Classification, and Quantitative Evaluation with Automotive Applications,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Work. 2005. CVPR Work.*, p. 62, 2005.

[69] B. Barrois, S. Hristova, C. Wohler, F. Kummert, and C. Hermes, “3D Pose Estimation of Vehicles Using a Stereo Camera,” *Intell. Veh. Symp.*, pp. 267–272, 2009.

[70] B. Southall, M. Bansal, and J. Eledath, “Real-time vehicle detection for highway driving,” *2009*

IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009, pp. 541–548, 2009.

[71] G. P. Stein, Y. Gdalyahu, and A. Shashua, "Stereo-Assist : Top-down Stereo for Driver Assistance Systems," *2010 IEEE Intell. Veh. Symp.*, pp. 723–730, 2010.

[72] Y. Zhu, D. Comaniciu, M. Pellkofer, and T. Koehler, "Reliable Detection of Overtaking Vehicles Using Robust Information Fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 401–414, 2006.

[73] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle Detection and Tracking in Car Video Based on Motion Model," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 583–595, 2011.

[74] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-Vision : Fusion of Stereo and Motion for Robust Environment Perception," *Kropatsch W.G., Sablatnig R., Hanbury A. Pattern Recognition. DAGM 2005. Lect. Notes Comput. Sci.*, vol. 3663, 2005.

[75] M. Perrollaz, J.-D. Yoder, A. Nègre, A. Spalanzani, and C. Laugier, "A Visibility-Based Approach for Occupancy Grid Computation in Disparity Space," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1383–1393, 2012.

[76] Y. M. Chan, S. S. Huang, L. C. Fu, and P. Y. Hsiao, "Vehicle detection under various lighting conditions by incorporating particle filter," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 534–539, 2007.

[77] P. Dave, N. M. Gella, N. Saboo, and A. Das, "A Novel Algorithm for Night Time Vehicle Detection Even with One Non-functional Taillight by CIOF (Color Inherited Optical Flow)," *Pattern Recognit. Syst. (ICPRS-16), Int. Conf.*, pp. 2–7, 2016.

[78] I. Cabani, G. Toulminet, and A. Bensrhair, "Color-based Detection of Vehicle Lights," *Intell. Veh. Symp. 2005. Proceedings. IEEE*, pp. 278–283, 2005.

[79] T. Machida and T. Naito, "GPU and CPU Cooperative Accelerated Road Detection," *Proc. IEEE ICCV Work.*, pp. 506–513, 2013.

[80] C. Fries and H. J. Wuensche, "Autonomous convoy driving by night: The vehicle tracking system," *IEEE Conf. Technol. Pract. Robot Appl. TePRA*, 2015.

[81] S. F. X. Bayerl, T. Luettel, and H. Wuensche, "Following Dirt Roads at Night-Time : Sensors and Features for Lane Recognition and Tracking," 2015.

[82] F. Garcia, D. Martin, A. de la Escalera, and J. M. Armingol, "Sensor Fusion Methodology for Vehicle Detection," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 123–133, 2017.

[83] X. Wang, L. Xu, H. Sun, J. Xin, and N. Zheng, "Bionic vision inspired on-road obstacle detection and tracking using radar and visual information," *17th Int. IEEE Conf. Intell. Transp. Syst.*, pp. 39–44, 2014.

[84] F. A. R. Alencar, L. A. Rosero, C. M. Filho, F. S. Osorio, and D. F. Wolf, "Fast Metric Tracking by Detection System: Radar Blob and Camera Fusion," *Proc. - 12th LARS Lat. Am. Robot. Symp. 3rd SBR Brazilian Robot. Symp. LARS-SBR 2015 - Part Robot. Conf. 2015*, no. 174, pp. 120–124, 2015.

[85] X. Liu, Z. Sun, and H. He, "On-road vehicle detection fusing radar and vision," *Proc. 2011 IEEE Int. Conf. Veh. Electron. Safety, ICVES 2011*, pp. 150–154, 2011.

[86] J. Wang *et al.*, "Appearance-Based Brake-Lights Recognition Using Deep Learning and Vehicle Detection," no. Iv, 2016.

[87] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, "Cooperative fusion for multi-obstacles

detection with use of stereovision and laser scanner," *Auton. Robots*, vol. 19, no. 2, pp. 117–140, 2005.

[88] "Caltech computational vision Caltech cars 1999 & 2001." [Online]. Available: https://www.vision.caltech.edu/html-files/archive.html. [Accessed: 03-Apr-2017].

[89] "PETS 2000." [Online]. Available: ftp://ftp.pets.rdg.ac.uk/pub/PETS2000. [Accessed: 03-Apr-2017].

[90] "PETS 2001." [Online]. Available: ftp://ftp.pets.rdg.ac.uk/pub/PETS2001. [Accessed: 03-Apr-2017].

[91] S. Sivaraman and M. M. Trivedi, "LISA Vehicle Detection dataset," *A General Active Learning Framework for On-road Vehicle Recognition and Tracking*, 2010. [Online]. Available: http://cvrr.ucsd.edu/LISA/vehicledetection.html. [Accessed: 03-Apr-2017].

[92] "TME Motorway Dataset." [Online]. Available: http://cmp.felk.cvut.cz/data/motorway/. [Accessed: 03-Apr-2017].

[93] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and M. Jiri, "A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera," *Int. IEEE Conf. Intel. Transp. Syst.*, pp. 975–982, 2012.

[94] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07–12–June, pp. 3061–3070, 2015.

[95] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results." [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html. [Accessed: 01-Apr-2017].

[96] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[97] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," *Proc. IEEE Work. Appl. Comput. Vis.*, vol. 2002, pp. 171–176, 2002.

[98] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cogn. Neurosci.*, vol. 3, no. 71–86, p. 1991.

[99] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, 1972.