

Automatisk gjenkjenning av rullebaneterskel i flyfoto.

Steinar Romberg Kvaale

Master i teknisk kybernetikk

Oppgaven levert: Juni 2008

Hovedveileder: Tor Engebret Onshus, ITK

Biveileder(e): Jostein Trones, Norwegian Special Mission AS

Oppgavetekst

Ved verifisering av flynavigasjon og landingsystemer er det nødvendig med et referansesystem for måling av posisjon. En metode er treghetsmåling sammen med korrigerende ut fra bilder av rullebanen eller andre objekter med kjent posisjon.

I endene av rullebanen, rullebanetersklene, er det et lett gjenkjennelig mønster. Disse ser ut som et fotgjengerfelt og kalles threshold. Internasjonale standarder tilsier at mellomrommet mellom de midterste stripene typisk er dobbelt så stort som mellom de øvrige stripene. Det er disse stripene som det tas bilder av og som posisjonen beregnes ut fra.

Det rå bildet fra bildesensoren har endel linseforvrengning slik at man kan ha en lett fisheye-effekt. Bildet kan også være vanskelig å bruke direkte grunnet vanskelige lysforhold.

Oppgaven går ut på å se på nødvendigheten av å utføre korrigerende av bildet ut fra kjente linsegenskaper. Det kan også være nødvendig å justere andre parametre (lys, kontrast) i bildet for å få det ønskede mønstret tydelig frem. Deretter kan for eksempel linjer i bildet finnes, slik at man kan danne seg en modell av stripene i bildet. Målet er å finne pixel (x,y) som tilsvarer det ønskede punktet i bildet. Det er ønskelig å utvikle en algoritme som automatisk finner det ønskede punktet i en serie av bilder.

Delmål for oppgaven: Utvikle algoritmer for nødvendig korreksjon av bildet. Finne en eller flere metoder for automatisk gjenkjenning av det ønskede punktet i midten av rullebanen. Se på muligheten for å lage en algoritme som finner det eller de bildene som kan brukes for finne det ønskede punktet i en serie bilder.

Oppgaven gitt: 14. januar 2008

Hovedveileder: Tor Engebret Onshus, ITK

Forord

Denne rapporten er resultat av avsluttende masteroppgave i studiet teknisk kybernetikk ved Norges teknisk-naturvitenskapelige universitet (NTNU) våren 2008.

Bakgrunn for arbeidet er forslag til oppgave gitt av Jostein Trones ved Norwegian Special Mission (NSM). Problemstillingen er basert på NSM sitt produkt for kalibrering av instrumentelle landingssystem for fly. Data kommer fra kontrollflygninger utført med dette systemet.

Jeg hadde lite bakgrunnskunnskap innenfor disse emnene og det har derfor vært mye nytt å sette seg inn i. Dette har vært svært lærerikt og har også gjort arbeidet med oppgaven spennende siden det ikke bare er bruk av tidligere kjent stoff. Oppgaven har gitt meg mye ny kunnskap og erfaring med å jobbe med ukjent stoff.

Ønsker å takke Jostein Trones for presentasjon av bakgrunnen for oppgaven, fremskaffelse av nødvendig data og motivasjon. Vil også takke Øyvind Stavdahl for tips til metoder for å løse oppgaven.

Steinar Romberg Kvaale

Trondheim, 9. juni 2008.

Sammendrag og konklusjon

Ved innflyging og landing av fly under forhold med redusert sikt brukes instrumentlandingsystem for veiledning langs en ønsket innflygningsbane. For å verifisere at instrumentlandingsystemet er innenfor gitte krav er det nødvendig å utføre kontrollflygninger for kalibrering av systemet.

Under kontrollflygingen er det nødvendig med et referansesystem for posisjon til estimering av banen til flyet. Referansesystemet er avhengig av å beregne flyets posisjon over kjente referansepunkter på rullebanen for korrigering av den estimerte banen. Flyets eksakte posisjon beregnes ut fra bilder av referansepunktene tatt fra buken på flyet.

Arbeidet presenterer en algoritme for automatisk beregning av posisjonen til referansepunktene. Ut fra en bildeserie tatt i det flyet passerer punktene velges automatisk bildet for videre beregning ut. Korrigering av linseforvringning ut fra kjente linseegenskaper er også implementert for økt nøyaktighet på beregningen av posisjonen til referansepunktene.

Det er utviklet flere metoder for å løse utvelgelsen av bildet og beregningen av posisjonen til referansepunktene. Basert på en evaluering av hver metode er den endelige løsning implementert. Denne algoritmen er testet ut på bildeserier fra reelle kontrollflygninger.

Resultatet av algoritmen testet på bildeseriene viser at algoritmen gjenkjenner og beregner posisjonen til referansepunktene innenfor gitte krav. Den kan videre implementeres i eksisterende system eller utvikles og testes videre for verifisering av resultatet i dette arbeidet.

Innholdsfortegnelse

1	INNLEDNING	1
1.1	MÅLSETNING	1
1.2	TIDLIGERE ARBEIDER	1
1.3	UTVIKLINGSVERKTØY.....	2
1.4	KOORDINATSYSTEM	2
1.5	RAPPORTENS OPPBYGNING.....	3
2	KALIBRERING AV INSTRUMENTLANDINGSSYSTEM	5
2.1	KALIBRERINGSSYSTEMET	6
2.2	REFERANSEPUNKTENE	6
2.3	BILDESERIENE	8
2.4	KRAV TIL LØSNINGEN.....	9
2.4.1	<i>Krav til valg av bilde.....</i>	<i>9</i>
2.4.2	<i>Krav til beregning av posisjonen til referansepunktene.....</i>	<i>9</i>
3	BILDEANALYSE OG MØNSTERGJENKJENNING.....	11
3.1	MODELLERING AV KAMERA	11
3.1.1	<i>Knapphullsmodellen</i>	<i>11</i>
3.1.2	<i>Linseforvrengning</i>	<i>12</i>
3.2	BILDEFORBEDRING.....	14
3.3	BILDESEGMENTERING.....	15
3.3.1	<i>Terskling.....</i>	<i>15</i>
3.3.2	<i>Kantdeteksjon.....</i>	<i>15</i>
3.4	MØNSTERREPRESENTASJONER	16
3.4.1	<i>Hough-transformen for å finne linjer.....</i>	<i>16</i>
3.4.2	<i>Samforekomstmatrise for mønsterrepresentasjon.....</i>	<i>19</i>
3.5	COSINUS SIMILARITETSTRANSFORM.....	20
4	VALG AV BILDE	21
4.1	METODE 1: UTVELGELSE BASERT PÅ INTENSITET OG LINJER I BILDET	21
4.1.1	<i>Utvelgelse på bakgrunn av intensitet</i>	<i>21</i>
4.1.2	<i>Hough-transform for å finne den sterkeste linjen i hvert bilde.....</i>	<i>21</i>
4.1.3	<i>Velger bildet med horisontal linje nærmest senter.....</i>	<i>22</i>
4.1.4	<i>Evaluering.....</i>	<i>22</i>
4.2	METODE 2: MØNSTERSAMMENLIGNING VED SAMFOREKOMSTMATRISER	23
4.2.1	<i>Forbehandling.....</i>	<i>23</i>
4.2.2	<i>Samforekomstmatriser</i>	<i>24</i>
4.2.3	<i>Sammenligne mønster.....</i>	<i>24</i>
4.2.4	<i>Utvelgelse</i>	<i>27</i>
4.2.5	<i>Evaluering.....</i>	<i>27</i>
4.3	METODE 3: MØNSTERSAMMENLIGNING AV VERTIKAL GJENNOMSNIITSPROFIL	28
4.3.1	<i>Beregne gjennomsnittsprofil.....</i>	<i>29</i>
4.3.2	<i>Sammenligne mønster.....</i>	<i>29</i>
4.3.3	<i>Utvelgelse</i>	<i>29</i>

4.3.4	<i>Evaluering</i>	30
5	KORRIGERE LINSEFORVRENGNING	31
5.1.1	<i>Beregne koordinatmatriser</i>	31
5.1.2	<i>Transformere bildet på bakgrunn av koordinatmatrisene</i>	32
6	BEREGNING AV REFERANSEPUNKTETS POSISJON	35
6.1	METODE 1: KANTDETEKSJON OG HOUGH-TRANSFORM	35
6.1.1	<i>Horisontal linje</i>	36
6.1.2	<i>Vertikale kanter</i>	37
6.1.3	<i>Beregne koordinatene til punktet</i>	38
6.1.4	<i>Evaluering</i>	39
6.2	METODE 2: PARAMETRISERING AV KANTER VIA HOUGHTANSFORM	39
6.2.1	<i>Horisontal linje</i>	39
6.2.2	<i>Vertikale linjer</i>	40
6.2.3	<i>Beregne koordinatene til punktet</i>	41
6.2.4	<i>Evaluering</i>	41
7	TOTALLØSNING	43
7.1	VALG AV BILDE	43
7.2	BEREGNING AV REFERANSEPUNKTETS POSISJON	43
7.3	IMPLEMENTASJON AV TOTALLØSNING	44
7.4	OPERATØRKOMMUNIKASJON	44
8	RESULTAT	47
9	DISKUSJON	49
9.1	RESULTAT	49
9.2	METODER	50
10	VIDERE ARBEID	51
REFERANSELISTE		53
LITTERATURREFERANSER		53
INTERNETTEREFERANSER		54
BILDEREFERANSER		54
REFERANSEPERSONER		54
VEDLEGG A	KAMERAKALIBRERING	55
VEDLEGG B	DIGITALE VEDLEGG	57

1 Innledning

Ved innflyging og landing av fly i redusert sikt er det viktigste systemet for veiledning instrumentlandingsystemet (ILS). For å opprettholde sikkerheten er det nødvendig med periodiske kontroller av ILS, for å verifisere at det er innenfor gjeldene krav.

For å kontrollere ILS brukes et fly med et eget kalibreringssystem som estimerer posisjonen til flyet under innflyging og sammenligner denne med ønsket innflygingsbane. For å kontrollere at ILS er innenfor aksepterte toleranser må det brukes et posisjonssystem som har betydelig større nøyaktighet en ILS [1][2].

Bakgrunnen for oppgaven er et kalibreringssystem med posisjonsmåling basert på inertnavigasjon med korrigerende via flyfoto av kjente referansepunkter. Dette består i hovedsak av et gyroskop som logger flyets posisjon under kontrollflygingen. Når flyet kommer inn over rullebanen beregnes flyets eksakte posisjon ut fra bilder tatt av de kjente referansepunktene. Flyets kjente posisjoner brukes til å korrigere innflygingsbanen estimert med gyroskopet.

1.1 Målsetning

I det eksisterende systemet blir posisjonen til referansepunktene markert manuelt av operatøren i bildet. Oppgaven går ut på å utvikle en algoritme som beregner posisjon til referansepunktet i bildene automatisk. Det blir tatt en bildeserie i det flyet passerer over hvert av referansepunktene og algoritmen må da først finne det bildet som er ønsket brukt før posisjonen til selve punktet kan beregnes. Bildene har forvrengning som følge av vidvinkelinnse i kameraet og dette skal korrigeres for å øke nøyaktigheten på beregningene.

Målet er å utvikle en robust løsning som finner referansepunktet innenfor en gitt toleranse. Siden bildeseriene kan være svært ulike er det nødvendig å ha en løsning som er dynamisk slik at endringer i datasettet ikke påvirker algoritmens evne til å løse problemet.

1.2 Tidligere arbeider

Det er ikke funnet noen direkte kilder til tidligere arbeid med samme problemstilling som i dette arbeidet. Men det er funnet arbeider som beskriver kort teorien bak korreksjon av estimert innflygingsbane ved bruk av flyfoto [1][2]. En presentasjon under International Flight Inspection Symposium 2004 viser hvordan et automatisk system for å finne referansepunktene med et kamerasystem fungerer [3]. Ingen av arbeidene går inn på

metodene brukt for å løse problemet med å beregne posisjonen til referansepunktene automatisk.

Når det gjelder bruk av ulike teknikker i bildebehandling er det sett på to arbeider som begge går på å finne egenskaper i bilder for bruk til navigering. Disse er bruk av bilder i autonom styring av mobil robot [4] og rullebanedeteksjon for selvstyrt landing av ubemannet fly [5].

Disse to arbeidene bruker også bildeanalyse til å beregne posisjon for henholdsvis robot og fly. Arbeidene ser på forbehandling og analyse av bildene for å finne de nødvendige egenskapene for videre å beregne objektets oppførsel.

1.3 Utviklingsverktøy

For utvikling og testing av algoritmen blir MathWorks Matlab 7.5.0 (R2007b) brukt. Dette er et mye brukt verktøy for simulering og utvikling av matematiske algoritmer. Matlab er et numerisk beregningsmiljø og programmeringsspråk.

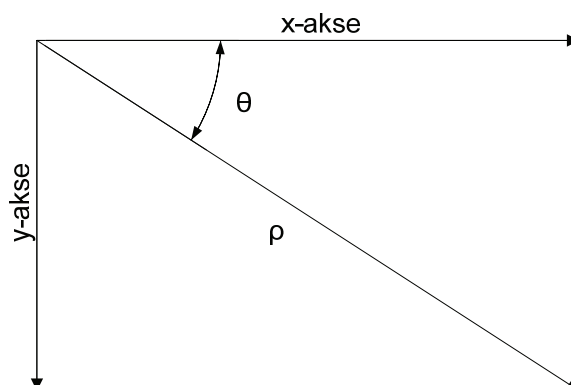
Fordelen med Matlab er at det har mange innebygde funksjoner som forenkler arbeidet og at det er spesielt tilpasset numeriske operasjoner på matriser. Det er også enkelt å skrive egne rutiner og funksjoner for raskt utvikling av prototyper.

Matlab kan utvides med tilleggspakker og det er i dette arbeidet brukt en pakke for bildebehandling, Image Processing Toolbox (IPT). Denne utvider funksjonaliteten i Matlab med mange funksjoner for bildebehandling.

1.4 Koordinatsystem

Ved bruk av Matlab i bildebehandling blir koordinatsystemet definert som i figur 1-1. Positiv retning på y-aksen (vertikalakse) er definert nedover med origo i øvre venstre hjørne. Positiv retning på x-aksen (horisontalaksen) er mot høyre.

I figur 1-1 er det også vist to andre variable som brukes når bildet blir beskrevet i polare koordinater, ρ og θ . Variabelen ρ beskriver lengden til et gitt punkt fra origo og θ er vinkelen mellom x-aksen og vektoren med positiv retning med klokka.



Figur 1-1: Koordinatsystemet brukt i arbeidet.

1.5 Rapportens oppbygning

Rapporten er bygd opp i tre hoveddeler med. Først beskrives bakgrunn og teori brukt i arbeidet. Deretter presenteres de ulike metodene som er utviklet for å løse delproblemene. Evalueringer av disse gir opphav til totalløsningen ved en algoritme. Totalløsningen blir så analysert og diskutert før forslag til videre arbeid blir presentert.

I kapittel 4 og 5 blir de ulike metodene for å løse oppgaven beskrevet. Det er valgt å ha en evaluering av disse metodene i hvert kapittel og en mer generell diskusjon av resultatet av arbeidet i kapittel 7.

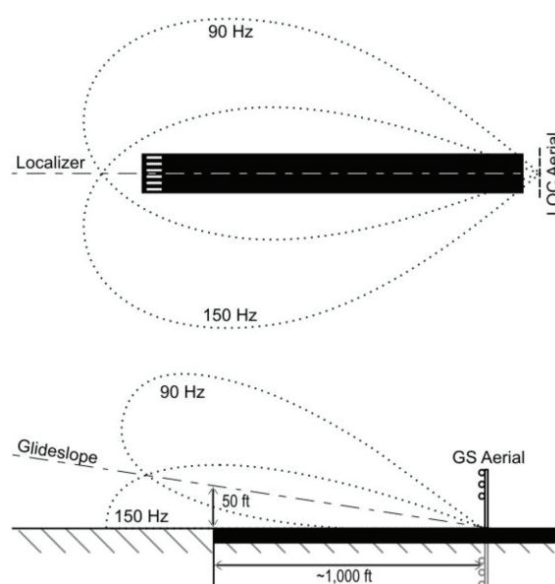
- **Kapittel 1** er en innledning til arbeidet med mål, tidligere arbeid og veiledning til leseren.
- **Kapittel 2** beskriver bakgrunn for oppgaven med definisjon av krav til løsningen.
- **Kapittel 3** presenterer teorien bak metoden brukt i arbeidet.
- **Kapittel 4** består av utviklede metoder for å velge bildet i bildeserien for videre bruk i beregningen av posisjonen til referansepunktet.
- **Kapittel 5** beskriver algoritmen utviklet for korrigerende av linseforvrengning.
- **Kapittel 6** består av utviklede metoder for å beregne referansepunktens posisjon i det valgte bildet.
- **Kapittel 7** presenterer forslag til totalløsning som resulterer i implementering av denne som en algoritme i Matlab.
- **Kapittel 8** viser resultatet av algoritmen testet på bildeseriene.
- **Kapittel 9** diskuterer resultatet av arbeidet og metodene som er brukt.
- **Kapittel 10** inneholder forslag til videre arbeid.

Rapporten inkluderer også to vedlegg med følgende innhold:

- **Vedlegg A** er kameraparametre brukt i arbeidet.
- **Vedlegg B** gir en oversikt over digitale vedlegg.

2 Kalibrering av instrumentlandingsystem

Instrumentlandingsystem (ILS) er et bakkebasert system for veiledning under innflygning og landing av fly. ILS veileder flyet i ønsket innflygningbane inn mot rullebanen. Systemets to hovedkomponenter er localizer og glideslope [1][i1]. Localizer veileder innflygningen i horisontalplanet, som oftest ved å markere midtlinjen til rullebanen, og glideslope veileder ønsket bane i vertikalplanet. Til sammen veileder localizer og glideslope flyet i den ønskede banen inn mot landing på rullebanen. Systemet består av radiosignaler som sendes ut av retningsstyrte antenner. Figur 2-1 viser en illustrasjon av hvordan dette fungerer.



Figur 2-1: Illustrasjon av de to støttesystemene i ILS [i1].

Flyet har instrumenter som fanger opp disse signalene og presenterer flyets posisjon relativt til localizer og glideslope. Basert på denne visningen får flygeren informasjon om hvordan flyet ligger i forhold til anbefalt innflygning til rullebanen.

Siden ILS er basert på radiosignaler vil det derfor være utsatt for forstyrrelser, refleksjoner og andre forhold som påvirker nøyaktigheten til systemet. Forstyrrelser kan komme fra støykilder i nærheten av rullebanen som forstyrrer signalet sendt fra ILS. Refleksjoner kan komme fra bygninger, terrengformasjoner, snø, is og andre naturfenomener.

Det er ulike kategorier som definerer i hvilke siktførhold det er tillatt for fly å lande ved bruk av ILS. For de ulike kategoriene er det krav til nøyaktighet og testintervall. For å kunne verifisere at ILS er innefor kravene til den gjeldende kategorien er det derfor nødvendig med et system for kalibrering av ILS.

2.1 Kalibreringssystemet

For å verifisere at ILS er innenfor kravene til gjeldende kategori må systemet kontrolleres. Dette gjøres ved å utføre en kontrollflyging hvor kontrollflyet følger ILS-anvisningen og kalibreringssystemet i flyet estimerer flyets bane. Ut fra kjent informasjon om ønsket veiledningsbane kan avviket beregnes mellom ønsket og estimert innflygingsbane. For logging av flyets posisjon under innflygning er det nødvendig med et referansesystem for posisjon som er mer nøyaktig enn ILS. Referansesystemet kan for eksempel være basert på satellitnavigasjon Differential Global Positioning System (DGPS) eller som i systemet som er bakgrunn for denne oppgaven, inertnavigasjon med Inertial Reference Unit (IRU).

Kalibreringssystemet basert på bruk av inertnavigasjon har et gyroskop (IRU) for å estimere banen til flyet. Når flyet starter kontrollinnflygningen nullstilles IRU i et startpunkt. Når flyet nærmer seg rullebanen tas det bilder av et referansepunkt i starten av rullebanen og i enden av rullebanen. Samtidig blir flyets høyde over rullebanen målt med en lasermåleren.

Bildene av disse to referansepunktene brukes til å beregne flyets eksakte posisjon i forhold til rullebanen. Flyets oppførsel i lufta, definert av roll, pitch og yaw, er kjent i det bildet blir tatt. Sammen med referansepunktene i bildet og kjent høyde over disse kan flyets eksakte posisjon over rullebanen beregnes. De to kjente posisjonene brukes igjen til å korrigere estimert bane fra IRU for økt nøyaktighet. Korrigering og beregninger blir utført etter at kontrollflygningen er gjennomført.

Har nå en måling av flyets posisjon under kontrollflygingen som sammenlignes med den ønskede innflygningen. Størrelsen på eventuelt avvik må være innefor gjelden krav.

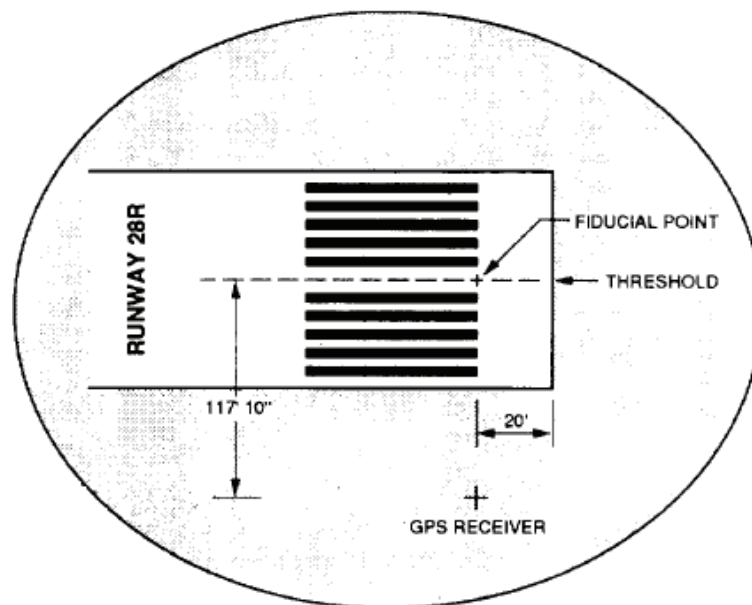
2.2 Referansepunktene

Referansepunktene som blir brukt for å korrigere den estimert banen fra IRU er et punkt i hver ende av rullebanen. Punktene har kjent posisjon relativt til rullebanen. I hver ende av rullebanen er det langsgående striper som marker start og stopp av rullebanen, kalt rullebaneterskel eller threshold. Disse stripene har et karakteristisk mønster som er definert i internasjonale standarder [p1]. Mellomrommet mellom de to midterste stripene er typisk dobbelt så stort som mellomrommet mellom de øvrige stripene. Dette mellomrommet markerer midtlinjen av rullebanen. Figur 2-2 viser en rullebaneterskel med disse stripene rett under flyet.



Figur 2-2: Fly på vei inn for landing over stripene på rullebaneterskelen [b1].

Posisjonen til de to punktene er referert til stripene og ligger midt mellom de midterste stripene i bredderetningen og langs enden av stripene ut av rullebanen i lengderetningen. Figur 2-3 viser en skisse av dette med referansepunktet (fiducial point) markert. Punktene er ikke markert i virkeligheten og posisjonen må beregnes via stripene.



Figur 2-3: Illustrasjon av referansepunktet (fiducial point) med kjent posisjon [2].

Videre i arbeidet vil referansepunktet i starten av rullebanen i fartsretningen til flyet bli kalt terskelpunkt og punktet i enden av rullebane kalt endepunkt. Grunnen til at disse må skilles er fordi bildene vil være ulike med tanke på hvor i bildet stripene befinner seg. Løsningen må derfor tilpasses for beregning av begge disse to punktene.

2.3 Bildeseriene

Bildeseriene som er brukt til utvikling og uttesting av algoritmen kommer fra Norwegian Special Mission (NSM) [p1]. De utvikler og leverer systemer for kalibrering av blant annet ILS basert på metoden beskrevet i kapittel 2.1. Bildeseriene er tatt under ulike kontrollflygninger utført med systemet til NSM.

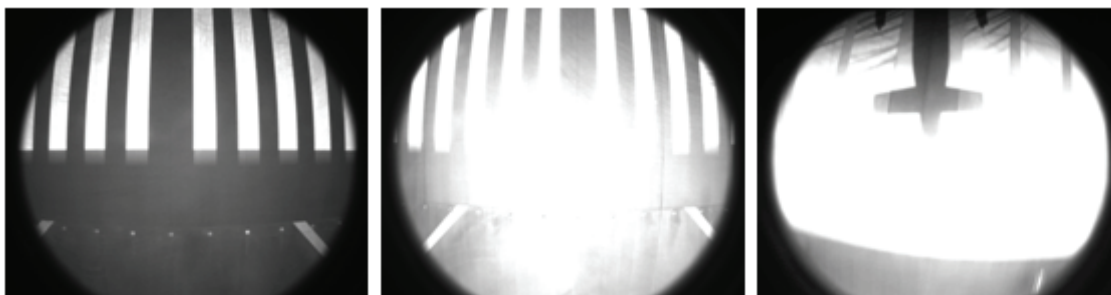
Bildeseriene er av området rundt referansepunktene og blir tatt i det flyet passerer over under kontrollflygingen. Bildene er av størrelsen 1280 x 1024 piksler og har 8-bits gråskala spekter. Dette gir et spekter på 256 nyanser fra svart (0) til hvitt (255).

Antall bilder i en bildeserie kan varierer men de nyeste seriene har normalt 25. Bildene blir tatt av et kamera som er montert i buken på flyet. Bildene har en del forvrengning som kommer av en linse med stor synsvinkel. For å få nok lys i bildet er det brukt en lukkertid som fører til noe uskarphet i fartsretningen. Dette fører til at de horisontale kantene til stripene blir noe utydelige.

Det er totalt 15 bildeserier. Tabell 2-1 viser en oversikt over disse. Ved manuell vurdering av bildeseriene er det bestemt om bildeserien kan brukes til automatisk deteksjon eller ikke. Dette er markert i tabellen. Figur 2-4 viser tre eksempler på bilder av rullebaneterskelen i tre ulike bildeserier. Ser at det kun er ett bilde som er bra nok til å bruke for beregning av posisjonen til det kjente punktet. De to andre bildene har forstyrrelser som gjør at det ikke kan forventes at algoritmen skal kunne produsere resultat.

Tabell 2-1: Oversikt over bildeseriene som er tilgjengelige for uttesting.

Bildeserie nr	Mappe	Startnr	Antall bilder	Punkt type	Egnet til automatisk deteksjon
1	\serie 1 bmp\	526	60	1	Nei
2	\serie 1 bmp\	587	60	2	Nei
3	\serie 2 bmp\	527	60	1	Ja
4	\serie 2 bmp\	587	60	2	Ja
5	\serie 3 jpg\	8531	55	1	Nei
6	\serie 3 jpg\	8586	55	2	Nei
7	\serie 3 jpg\	8641	55	1	Nei
8	\serie 3 jpg\	8696	55	2	Nei
9	\serie 4 jpg\	13256	8	1	Ja
10	\serie 5 bmp\	13031	25	1	Ja
11	\serie 5 bmp\	13056	25	2	Ja
12	\serie 5 bmp\	13081	25	1	Ja
13	\serie 5 bmp\	13106	25	2	Ja
14	\serie 5 bmp\	13181	25	1	Ja
15	\serie 5 bmp\	13206	25	2	Ja



Figur 2-4: Eksempel på bilde av samme objekt i tre ulike bildeserier. Bra bilde fra serie 3 (venstre), blendet bilde fra serie 1 (senter) og blendet og forstyrret bilde fra serie 7 (høyre).

2.4 Krav til løsningen

Som beskrevet i innledningen skal det utvikles en algoritme som automatisk beregner posisjonen til referansepunktet i bildet. Bildet skal velges ut fra bildeserien av algoritmen på bakgrunn av kravene til utvelgelse. Algoritmen skal også korrigere linseforvringningen i bildet før posisjonen til referansepunktet beregnes.

Videre følger krav til utvelgelsen og nøyaktigheten på beregningen av posisjonen til referansepunktet.

2.4.1 Krav til valg av bilde

Siden bildene inneholder forvringning som følge av linsen er det satt som krav at bildet som velges ut har referansepunktet nærmest senter av bildet. Dette vil si at bildet som velges ut er det bildet i serien med de horisontale kantene til stripene nærmest senter av bildet. Bildet til venstre i figur 2-4 viser et eksempel på et bilde som oppfyller kravet.

Unngår da områdene i bildet med størst forvringning og nøyaktigheten på beregningen av posisjonen til punktet blir mer nøyaktig. Dessuten vil beregningen av flyets posisjon ut fra referansepunktet også bli mest nøyaktig i senter av bildet da vinkelen mellom senterlinjen i kameraet og punktet blir liten.

2.4.2 Krav til beregning av posisjonen til referansepunktene

Referansepunktene brukes til å korrigere estimert bane under kontrollflygingen. Beregningen av punktet bør derfor ligge innenfor 10-20 cm horisontalt, og 1-2 meter vertikal. Et bilde fra 20 meters høyde har ca 3 cm per piksel og det er derfor ikke ønskelig å ha et avvik på mer enn 3-5 piksler i horisontal retning og 30-50 i vertikal retning [p1].

Siden bildene har noe uskarphet i vertikal retning er det en viss usikkerhet rundt hvor posisjonen til punktet i realiteten er. Men videre arbeid går ut fra at de horisontale kantene til stripene er der kantene er tydeligst. Antar da for videre bruk at referansepunktet har vertikal posisjon på høyde med disse.

3 Bildeanalyse og mønstergjenkjenning

Bildeanalyse og mønstergjenkjenning er et område med mange ulike metoder og ingen fasit. ”Prøv og feil”-metoden er en mye anvendt teknikk for å løse problemer [6]. For å løse oppgaven er det derfor nødvendig å få et overblikk over hvilke metoder som kan fungere og deretter prøve disse ut på problemet for å se hvordan resultatet blir.

Det vil i dette kapittelet bli beskrevet modeller og metoder som blir brukt videre i arbeidet. I første omgang blir en modell for kamera og linseforvringning presentert. Videre blir presentert ulike metoder for bildeforbedring, bildeanalyse og mønstergjenkjenning.

3.1 Modellering av kamera

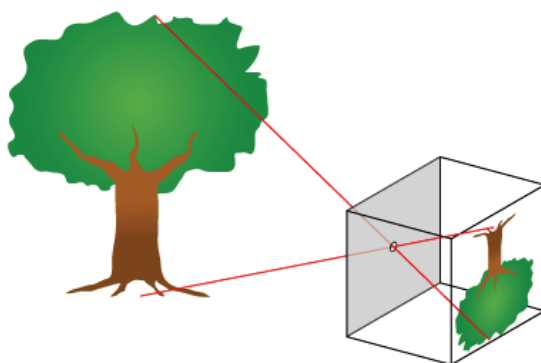
For å få med en større del av rullebanen er det brukt en vidvinkellinse, også kalt fisheye, i kameraet. Problemet med denne type linser er at det fører til forvringning i bildene. Dette sees tydelig i bildene siden rette linjer blir buet og det blir forholdsvis stor forvringning ut mot kanten av bildet. Det er derfor nødvendig med modeller for hvordan forvringningen oppstår for videre bruk til korrigering av bildene.

To modeller presenteres for videre bruk:

- Knapphullsmodellen for modellering av kamera.
- Modell av linseforvringning for videre bruk i korreksjon.

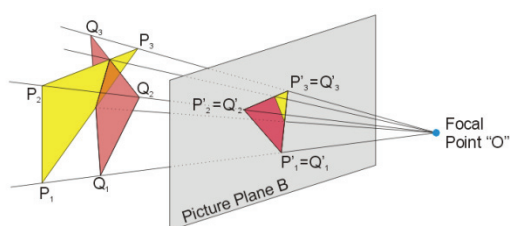
3.1.1 Knapphullsmodellen

For å kunne bruke bilder i geometriske sammenhenger er det nødvendig med en representasjon for kameraet. En mye brukt metode er knapphullsmodellen [7][8]. Denne baserer seg på at kameraet lager en todimensjonal perspektivprosjeksjon av en tredimensjonal ”verden”. Knapphullsmodellen tilnærmer seg kameraet ved at all lys passerer gjennom knapphullet. Knapphullet er det samme som brennpunktet i virkelig kamera. Modellen antar at linsen fører til en lineær projeksjon og tar ikke hensyn til ulike geometriske egenskaper i linsen. Figur 3-1 viser hvordan knapphullsmodellen fungerer.



Figur 3-1: Illustrasjon av knapphullsmodellen [b2].

En mer geometrisk representasjon av knapphullsmodellen vises i figur 3-2. Her er bildeplanet flyttet mellom objektet og brennpunktet for å unngå speiling av bildet. Dette er ekvivalent med å ha det bak som i figur 3-1. Ser også bedre sammenhengen mellom avstanden mellom bildeplanet og brennpunktet som har innvirkning på størrelsen på det avbildede objektet.

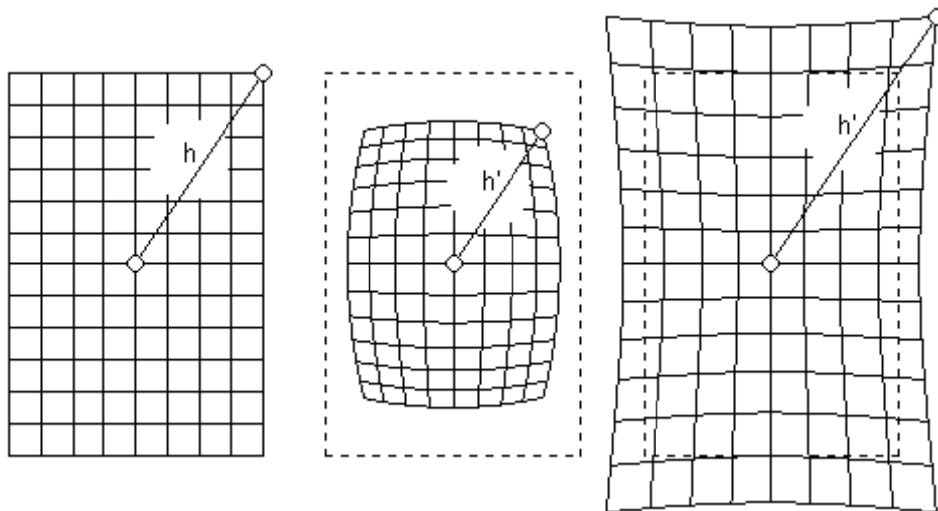


Figur 3-2: Matematisk framstilling av knapphullsmodellen [b3].

3.1.2 Linseforvringning

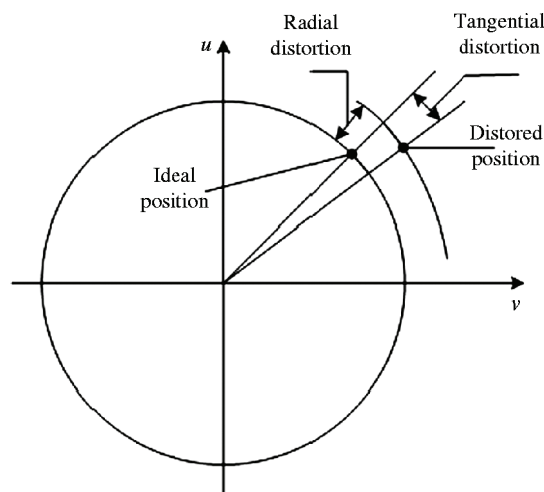
Knapphullsmodellen er en tilnærming til virkeligheten. Ved å anta at linsen i kameraet fungerer som en ideell knapphullsmodell tas det ikke hensyn til geometriske forvringning som påføres av optikken i kameraet. Som figur 3-2 viser antas det at alle punktene i objektet blir projisert lineært på bildeplanet. Dette er stort sett ikke tilfelle i virkeligheten siden optikken vil tilføre ulike typer forvringning i mer eller mindre grad.

Ut fra knapphullsmodellen har alle punkt i objektet en forventet plassering i bildeplanet basert på den lineære projeksjonen. Forvringning er når punktenes posisjon i bildet ikke stemmer overens med posisjonene som er forventet basert på knapphullsmodellen. Figur 3-3 viser en illustrasjon av dette med ulik radiell forvringning.



Figur 3-3: Radiell forvrengning av bilde, uforvrent bilde (venstre), negativ radiell forvrengning (barrel distortion) (senter), positiv radiell forvrengning (pincushion distortion) (høyre) [b4].

Det er i hovedsak to typer forvrengning som kan oppstå som følge av optikken i kameraet, radiell og tangentiell forvrengning [8]. Radiell forvrengning oppstår som følge av ulik brytning i linsen slik at punktene får en annen posisjon i bildet enn uten forvrengning. Tangentiell forvrengning er en forskyvning av punktene langs bildeplanet og kan komme av skeivstilling av komponenter i optikken. Figur 3-4 viser hvordan bildepunktene får endret posisjon som følge av de to forvrengningstypene.



Figur 3-4: Figuren viser hvordan bildepunktene får forvrent posisjon som følge av linseforvrengning [b5].

Radiell og tangentiell forvrengning med knapphullsmodellen som projeksjon har kjente likninger. Sammenhengen mellom bildepunktene (x, y) uten forvrengning og de forvrengte bildepunktene (x_d, y_d) er gitt av (3-1). Koordinatene er normaliserte med origo i senter av bildet gitt av parameteren cc (true center) [i2][8].

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + kc(1)r^2 + kc(2)r^4 + kc(5)r^6) \begin{bmatrix} x \\ y \end{bmatrix} + dx \quad (3-1)$$

Det første uttrykket er radiell forvrengning og dx er tangentiell forvrengning gitt av (3-2). Avstanden fra senter r er i polare koordinater og gitt av (3-3).

$$dx = \begin{bmatrix} 2kc(3)xy + kc(4)(r^2 + 2x^2) \\ kc(3)(r^2 + 2y^2) + 2kc(4)xy \end{bmatrix} \quad (3-2)$$

$$r^2 = x^2 + y^2 \quad (3-3)$$

Vektoren kc inneholder forvrengningsparametrene til linsen.

$$kc = \{kc(1), kc(2), kc(3), kc(4), kc(5)\} \quad (3-4)$$

Har nå en modell for hvordan forvrengningen i bildene oppstår. Parametrene kc og cc finnes som oftest ved å utføre en kalibrering av kameraet.

3.2 Bildeforbedring

Målet med bildeforbedring er å utføre operasjoner på bildet slik at resultatet er bedre egnet enn det originale bildet for en spesifikk applikasjon. Ordet spesifikk er viktig siden de fleste metodene innen bildeforbedring og bildebehandling generelt, er svært problemorienterte. En metode som fungerer bra til en applikasjon kan i et annet tilfelle ikke brukes med tilfredsstillende resultat. Målet med bildeforbedring er å forbedre og finne de egenskapene i bildet som er interessante for videre bruk og framgangsmåten for dette vil være ulikt fra problem til problem [6].

Bildeforbedring brukes om operasjoner som blir utført på bildene for å forbedre og fremheve egenskaper for videre bruk i bildeanalyse. Skillet mellom bildeforbedring og segmentering er litt uklart siden litteraturen [6] [7] bruker metoder fra bildeforbedring også i segmentering. Men bildeforbedring er ofte det første steget i en bildeanalyse.

Bildeforbedring kan være:

- Filtrere bildet for å fjerne støy og redusere detaljgrad.
- Filtrere bildet for å utheve egenskaper i bildet som for eksempel kanter og linjer.
- Justere lys- og kontrastforhold for å bedre dynamikken i bildet og fremheve egenskaper for videre bruk.

3.3 Bildesegmentering

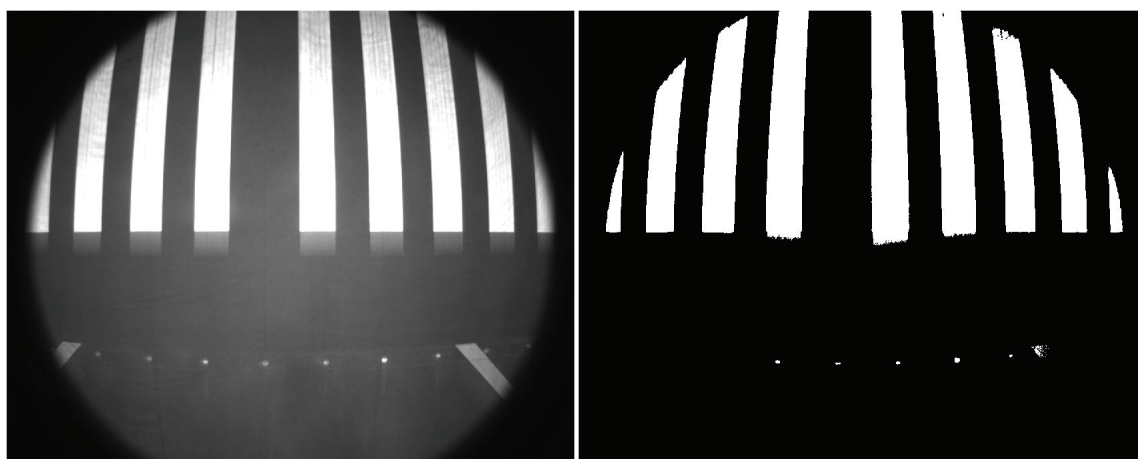
For å bruke bilder videre i analyse og mønstergjenkjenning er det nødvendig å få presentert de ønskede egenskapene i bildet på en hensiktsmessig måte. Hvilke egenskaper som er interessante vil avhenge fra problem til problem. Segmentering er metoder for å samle områder i bildet med samme egenskap.

Segmenteringsmetoder som blir forklart videre er:

- Terskling av bildet for å skille egenskaper i bildet basert på pikselverdier.
- Deteksjon av kanter i bildet.

3.3.1 Terskling

En mye brukt metode i bildebehandling er konvertering av bilder til binære bilder via terskling. Terskling utføres ved å definere en grense der alle pikselverdier under grensen blir satt til 0 (svart) og alle verdier over grensen blir satt til 1 (hvitt). Målet med dette er at egenskaper i bildet skal bli fremhevet til videre bruk. Dette fungerer bra på bilder der det er stor forskjell i intensitet på de egenskapene som ønskes fremhevet og resten av bildet. Figur 3-5 viser et eksempel på terskling.



Figur 3-5: Eksempel på terskling av bilde til binært bilde.

Vanskeligheten med denne metoden er for det første å finne en passende grense for skillet mellom 0 og 1. Feil grense kan føre til at resultatet av tersklingen ikke fremhever egenskapene som ønsket og resultatet er uegnet til videre bruk. Dessuten er det vanskelig å velge en statisk grenseverdi siden bilder er ulike og grensen derfor ikke passer for alle. Terskling er også vanskelig å bruke på bilder med ujevn belysning.

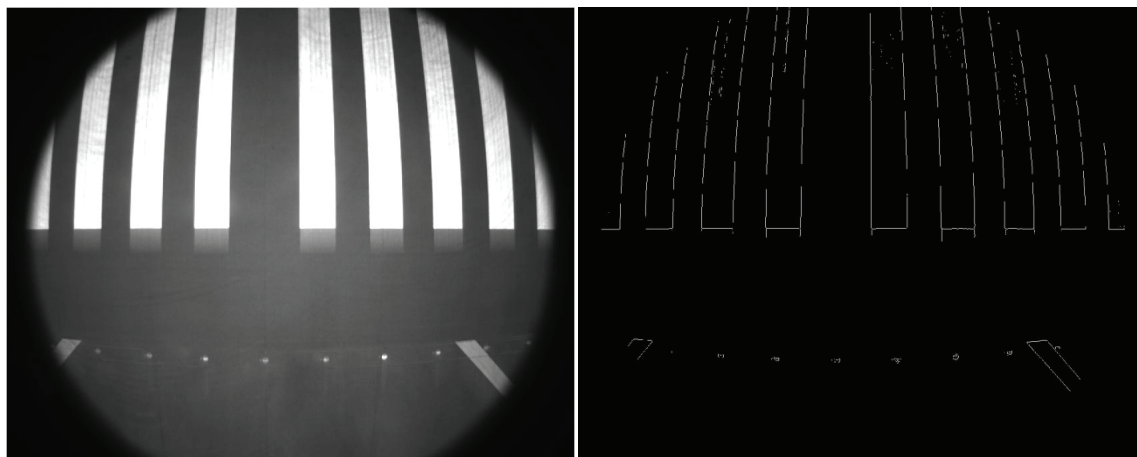
3.3.2 Kantdeteksjon

En annen metode for å fremheve egenskaper i bildet er ved kantdeteksjon. Kantdeteksjon baserer seg på å finne diskontinuiteter i bildet som tilsvarer kanter.

Det finnes mange ulike kantdeteksjonsmetoder som gir ulike resultater. Noen baserer seg kun på filtrering mens andre baserer seg på kalkulasjon av gradienter. To metoder som

ofte er brukt er Sobel og Canny. De ulike metodene har ulike egenskaper og hvilken som blir brukt avhenger av det aktuelle problemet.

Figur 3-6 viser resultatet av Sobel-kantdeteksjon på et bilde. Denne er utført med funksjonen *edge* i Matlab. Denne funksjonen terskler og tynner ut de detekterte kantene slik at resultatet blir kanter markert i et binært bilde med kun en piksels bredde.



Figur 3-6: Eksempel på kantdeteksjon på bilde.

3.4 Mønsterrepresentasjoner

For å kunne gjøre analyser basert på mønsteret i bildet er det ofte nødvendig å bruke metoder for å representere og registrere egenskaper i bildet.

I videre arbeid blir to metoder for brukt for å representere egenskaper i bildet:

- Hough-transformen for å finne og representere rette linjer i bildet.
- Samforekomstmatrise for å registrere mønsteret i bildet.

3.4.1 Hough-transformen for å finne linjer

En mye brukt metode innefor segmentering av bilder er bruk av Hough-transformen for å finne rette linjer i et bilde. Denne metoden ble beskrevet av Hough i 1962[9]. Fordelen med denne metoden er at rette linjer blir representert via parametre som senere er enkle å bruke videre i analyse og mønstergjenkjenning.

Hough-transformen baserer seg på representasjonen av en linje i bildeplanet, likning (3-5).

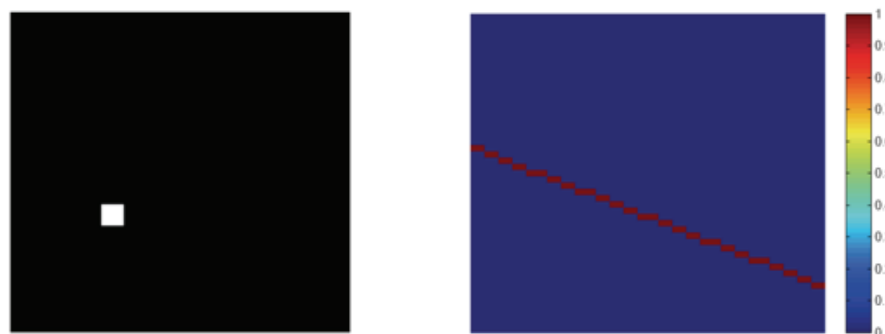
$$y = ax + b \quad (3-5)$$

Parameteren a svarer til stigningstallet og b er skjæringspunktet med y-aksen. En linje kan defineres som en rett kurve med uendelig mange punkter der alle punktene vil ha samme parametre (a, b) mens koordinatene (x, y) vil variere i henhold til (3-5)

Siden et bilde er en matrise med bestemte koordinater er derfor x - og y -verdiene gitt og det er derfor parametrene (a, b) til eventuelle linjer som er ukjent. Disse må da estimeres og Hough foreslår å skrive om (3-5) til (3-6) [6]

$$b = y - ax \quad (3-6)$$

Nå kan karakteristikken til linjen representeres i parameterplanet (a, b) istedenfor i bildekoordinater (x, y) . Linjen kan nå representeres som et punkt i parameterrommet til (a, b) . Gjennom et punkt vil det kunne gå et uendelig antall linjer med tilhørende parametersett. Figur 3-7 viser et punkt i bildet og Hough-transformasjonen med de ulike parametrene (a, b) som tilsvarer "alle" teoretiske linjer som kan gå gjennom punktet.



Figur 3-7 Et punkt i bildeplanet (høyre) og tilsvarende transformasjon i parameterplanet (høyre) [8].

Metoden som brukes er å beregne parametrene (a, b) til "alle" linjene som kan gå gjennom hvert punkt. Hvert parameterpar blir registrert i en akkumulatormatrise med a og b som koordinater. Når punkt i bildet ligger på linje vil de ha ett parameterpar som er like og akkumulatorverdien for dette parameterparet vil øke. Elementet i akkumulatormatrisen som er størst når alle punktene er prosessert vil være den mest fremtredende linjen i bildet. Figur 3-8 viser en linje og tilsvarende transformasjon for alle punktene til linjen. Ser at det sterkeste punktet er der hvor alle linjene krysser hverandre og dette er da parametrene til linjen.

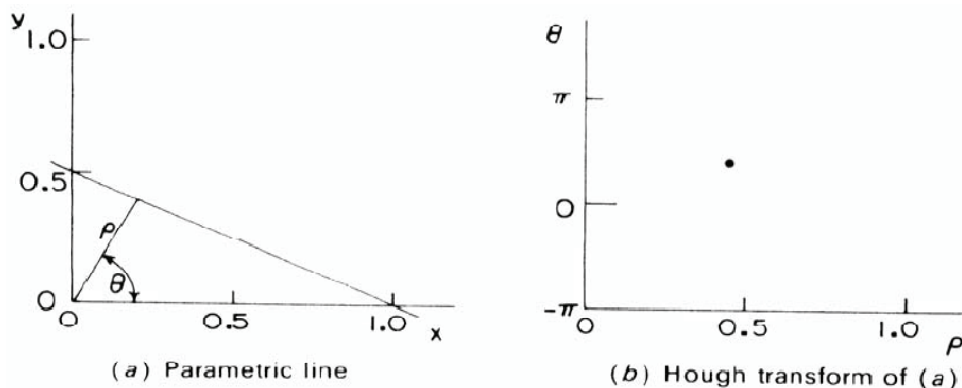


Figur 3-8 En linje i bildeplanet (venstre) og transformasjonen i parameterplanet (høyre)[8].

Men denne metoden har et problem. Parameterrommet vil være ubegrenset siden vertikale linjer vil ha $a = \infty$. Duda og Hart presenterte i 1972 en løsning på dette problemet[10]. De foreslo å bruke en ny beskrivelse for linjen, som består av normalvektoren til linjen gjennom origo, likning (3-7) [6][10].

$$\rho = x \cos \theta + y \sin \theta \quad (3-7)$$

Her er ρ lik lengden av linjenormalen fra origo til linjen og θ er vinkelen til vektoren. Disse parametrene kan nå brukes til å danne parameterplanet (ρ, θ) som vil være begrenset. Figur 3-9 viser normalparametriseringen og resultatet av transformasjonen.



Figur 3-9: Normalparametriseringen av en linje (a) og resultatet av transformasjonen (b) [10].

Ved å ta utgangspunkt i et punkt i bildet og beregne ρ som funksjon av ulike verdier for θ , vil dette gi en parameterkurve som er unik for punktet. Hvert punkt vil gi en unik parameterkurve og der disse skjærer hverandre vil det være punkter som ligger på linje.

Parameterkurvene akkumuleres på samme måte i en matrise som videre kan undersøkes for å finne hvor det er rette linjer i bildet. Dette vil vise seg som topper eller matriseelementer med høyest verdi innenfor et område. Figur 3-10 viser eksempel på transformasjon av en linje med normalparametriseringen.

Ved omskriving av (3-5) kan linjene beskrives på følgende måte [11].

$$y = -\frac{\cos \theta}{\sin \theta} x + \frac{\rho}{\sin \theta} \quad (3-8)$$

$$y = -\frac{1}{\tan \theta} x + \frac{\rho}{\sin \theta} \quad (3-9)$$

Ser at (3-9) har samme form som (3-5) med sammenhengen gitt av (3-10).

$$a = -\frac{1}{\tan \theta}, b = \frac{\rho}{\sin \theta} \quad (3-10)$$

Det er nå enkelt å plote de ønskede linjene med (3-5) og (3-10) med parameterparene funnet i akkumulatormatrisen.

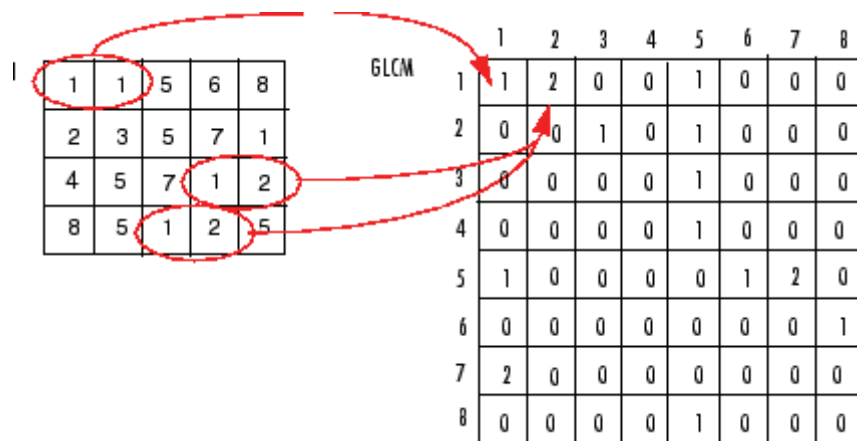


Figur 3-10: Eksempel på resultatet av linjen (venstre) transformert med normalparametriseringen til parameterplanet (høyre) [8].

3.4.2 Samforekomstmatrise for mønsterrepresentasjon

For å registrere til mønsteret i et bilde kan det brukes en samforekomstmatrise (co-occurrence matrix). Denne baserer seg på å se på samforekomsten av ulike pikselverdier i bildet [6]

Bilder er oppbygd av piksler med ulike verdier som til sammen lager et mønster. Ulike typer mønster vil ha en ulik sammensetning av naboforhold mellom pikselverdier i bildet. Denne egenskapen kan brukes til å registrere mønsteret i bildet. Figur 3-11 viser et eksempel på et bilde og tilhørende samforekomstmatrise.



Figur 3-11: Hvordan naboverdiene i bildet blir registrert i samforekomstmatrisen [b6].

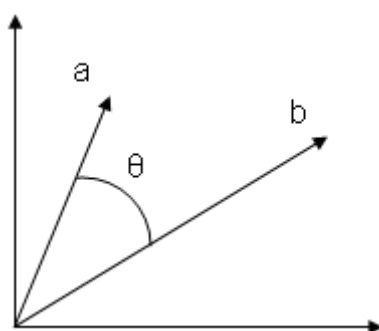
Det er ofte nødvendig å telle flere naboforhold enn kun til høyre. Ved å telle kun i en retning vil resultatet bli følsomt for rotasjon. Derfor kan det også være hensiktsmessig å telle naboforholdet i vertikal og diagonal retning. For å forenkle matrisen til videre bruk kan en registrere (1,2) også som (2,1). Fordelen med dette er at det da kun er nødvendig å benytte halve matrisen siden denne da blir symmetrisk om diagonalen.

For et 8-bits bilde vil størrelsen på matrisen bli $2^8 \times 2^8$. Siden denne matrisene blir store kan det være hensiktsmessig å slå sammen flere verdier slik at størrelsen på matrisen blir redusert. Egenskapene til samforekomstmatrisen kan videre brukes i ulike former for mønstergjenkjenning.

3.5 Cosinus similaritetstransform

Vurdering av likhet mellom mønster er en viktig del av mønstergjennkjenning. En mye brukt metode er cosinus similaritetstransform. Denne er blant annet brukt i arbeider med ansiktsgjennkjenning [12]. Metoden gir et mål på likheten mellom det to mønstrene som sammenlignes. Det er normalt at det ene mønsteret er referansemønsteret som ønskes gjenkjent i et datasett. Resultatet er et mål på likheten mellom referansemønsteret og mønsteret i datasettet. Likhetsverdiene kan så brukes videre i det ønskede formålet for videre klassifisering.

Metoden baserer seg på å sammenligne vektorer. Opphavet for metoden er definisjonen av prikkproduktet mellom to vektorer. Figur 3-12 viser to vektorer (a og b) med en avstand definert av vinkelen θ .



Figur 3-12: To vektorer med avstand gitt av vinkel θ .

Prikkproduktet mellom to vektorer er definert av likning (3-11). Får definisjonen av cosinus similaritetstransformen ved å skrive (3-11) om slik at $\cos \theta$ står alene, likning (3-12).

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}| \cos \theta \quad (3-11)$$

$$\cos \theta = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|} \quad (3-12)$$

Siden cosinus til en vinkel er mellom -1 og 1 er dette området som resultatet av sammenligningen ligger mellom i. Det er normalt å bruke positive vektorverdier og får da likhetsverdier mellom 0 og 1 hvor 0 er ingen likhet og 1 er helt lik.

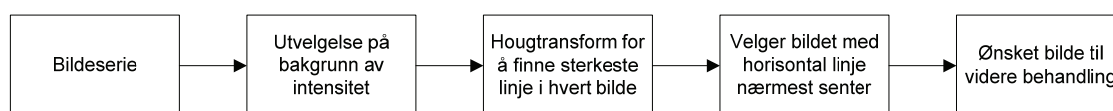
Denne sammenligningsmetoden tar ikke hensyn forskjell i lengde på vektorene. Det er kun sammenligning av formen slik at vektorene kan ha en innbyrdes faktorforskjell uten at dette påvirker resultatet.

4 Valg av bilde

For å velge bildet i bildeserien som svarer til kravet satt for utvelgelsen er det utviklet tre ulike metoder. Oppgaven til disse metodene er å gjenkjenne det bildet som svarer til kravet. Metodene er implementert i Matlab og testet ut på bildeserien. På bakgrunn av dette er hver metode evaluert.

4.1 Metode 1: Utvelgelse basert på intensitet og linjer i bildet

En viktig del av arbeidet med bilder er å gjøre nytte av egenskapene i bildene som forkunnskap for å lage en løsning tilpasset problemet. Bildene det arbeides med har i hovedsak to egenskaper som er tydelige. Det ene er stripene som har markerte kanter og består av tilnærmet rette linjer. Det andre er at stripene har høyere intensitet enn områdene rundt. Denne metoden bruker disse to egenskapene. Figur 4-1 viser hovedtrinnene i metoden.



Figur 4-1: Hovedstegene i metode 1 for valg av bilde.

4.1.1 Utvelgelse på bakgrunn av intensitet

Bildet som skal brukes inneholder striper. Siden stripene er lyse har bilder med striper høyere gjennomsnittsintensitet enn bilder uten striper. Denne egenskapen kan da brukes til en første utvelgelse av bilder fra bildeserien.

Måten dette kan gjøres på er å sette en grenseverdi der alle bilder men gjennomsnittsverdi under grensen blir forkastet. Denne grensen kan settes på flere måter. Noen måter å sette grenseverdien på er som gjennomsnittsverdien av alle bildene, som medianen av alle bildene, eller som en faktor av den høyeste gjennomsnittsintensiteten i serien. Bilder som inneholder bare striper har intensitet høyere enn det bildet som er interessant og det kan derfor også settes en grenseverdi for å forkaste bilder med intensitet over denne grensen. Det er viktig at grenseverdien er dynamisk slik at den tilpasser seg bildeserien.

4.1.2 Hough-transform for å finne den sterkeste linjen i hvert bilde

Om grenseverdien er satt riktig skal en nå kun sitte igjen med bilder som inneholder stripene. Det neste trinnet bruker det kravet at det er ønskelig å ha de horisontale kantene til stripene nærmest senter av bildet. For å finne bildet som oppfyller dette kravet må det

først finnes linjer i bildene. Det blir først utført en kantdeteksjon på bildene med funksjonen *edge* og Sobel som metode. På resultatet av kantdeteksjonen blir det utført en Hough-transform med funksjonen *hough*. Deretter blir parametrene til den sterkeste linje i hvert bilde funnet i de respektive akkumulatormatrisene med funksjonen *houghpeaks*. Parametrene til den sterkeste linjen i hvert av bildene lagres i en tabell for bruk i neste steg.

4.1.3 Velger bildet med horisontal linje nærmest senter

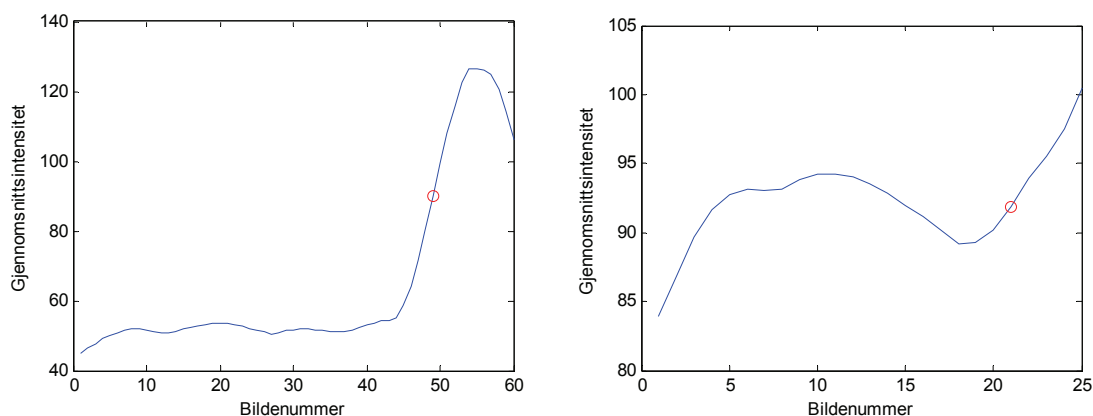
Siden bildet som er ønskelig å bruke videre har den horisontale linjen nærmest midten av senter brukes dette kravet i utvelgelsen. Først blir alle bilder som har linje med vinkel utenfor ett gitt område rundt horisontallinjen forkastet. Dette er fordi den sterkeste linjen i mange av bildene er vertikal som følge av de vertikale kantene til stripene. Sitter nå igjen med bilder som kun har horisontale linjer. Avstanden fra linjen til senter av bildet blir beregnet for hvert bilde og bildet med linje nærmest senter blir valgt til videre bruk.

Sitter nå igjen med bildet i bildeserien der den horisontale kanten til stripene er nærmest senter av bildet.

4.1.4 Evaluering

Ved uttesting av metoden på bildeseriene viser det seg at metoden har varierende resultat. Det ble fort klart at den ikke fungerte som forventet. Hovedproblemet er den først utvelgelsen av bildene. Det er vanskelig å finne en måte å bestemme grenseverdien på som fungerer for alle bildeserier. Ved å bruke medianen av gjennomsnittsverdiene for hele bildeserien fungerer det på bildeserier med tilnærmet likt antall bilder med og uten striper. Når bildeserien inneholder mange bilder som ikke har striper vil mange bilder uten striper bli valgt ut og problemer oppstår. Det er derfor nødvendig å finne en metode for å velge grenseverdi som fungerer for ulike bildeserier.

Figur 4-2 viser et annet problem som også oppstår. Plottet viser gjennomsnittintensiteten til bildene i serie 3 og 10. Det røde punktet markerer bildet som er best å bruke. Ser av figurene at for serie 3 er det stor forskjell på gjennomsnittintensiteten i bildene. For serie 10 er det ikke like tydelig. Ser her at en stor del av bildene før det ønskede bildet har høyere gjennomsnittsverdi og utvelgelsen av bildet kan bli feil. Dette kommer av at de tidlige bildene i serie 10 inneholder andre striper.



Figur 4-2: Plot av gjennomsnittintensitet for bildene i en serie 3 (venstre) og serie 10 (høyre). Rødt punkt markerer bildet som ønskes valgt ut.

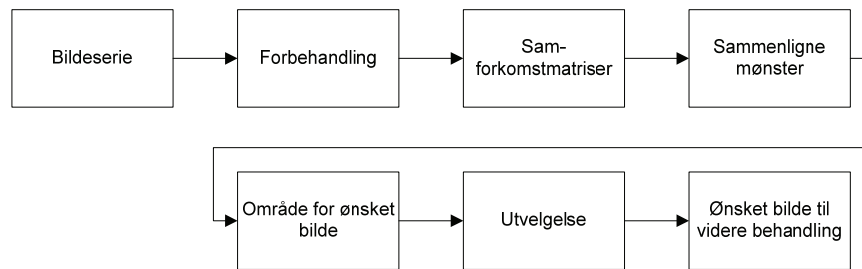
Problemer med utvelgelsen basert på gjennomsnittintensitet oppstår også når bildeserien har liten innbyrdes forskjell i intensitet mellom bildene. Effekter av andre objekter i bildet kan føre til at bilder uten striper kan ha høyere intensitet enn bilder med striper og lignende. Dette er vanskelig å forutse og problemer kan derfor oppstå.

Det er viktig at bildene som blir valgt ut på bakgrunn av gjennomsnittintensitet inneholder stripene eller så vil resultatet av Hough-transformen også kunne føre til feil resultat. Grunnen til dette er at kantdeteksjonen, som brukes for å forberede bildet til Hough-transformen, vil finne mange utydelige kanter i en tekstur som for eksempel gress. Et eksempel på dette kan være at et bilde har en overgang mellom gress og asfalt som er nærmest midten av bildet. Dette bildet vil da bli valgt fordi om det ikke inneholder stripene. Det er derfor stor sannsynlighet for at Hough-transformen vil finne linjer som kan føre til at feil bilde velges om datasettet inn er feil.

4.2 Metode 2: Mønstersammenligning ved samforekomstmatrise.

En metode for å registrere mønster i bilder er samforekomstmatrisen, se kapittel 3.4.2 . Ved å se på samforekomsten av pikselverdier i bildet og registrere dette i en matrise kalt samforekomstmatrisen kan mønsteret i bildet registreres. Ulike mønster i bildet vil gi ulike samforekomster av pikselverdier og dermed ulike samforekomstmatriser.

Samforekomstmatriser blir satt opp for hvert bilde i bildeserien og denne sammenlignes mot en samforekomstmatrise av et referansemønster. Resultatet av sammenligningen brukes for å velge ut bildet. Figur 4-3 viser hovedstegene i metoden.



Figur 4-3: Hovedstegene i metode 2 for valg av bilde.

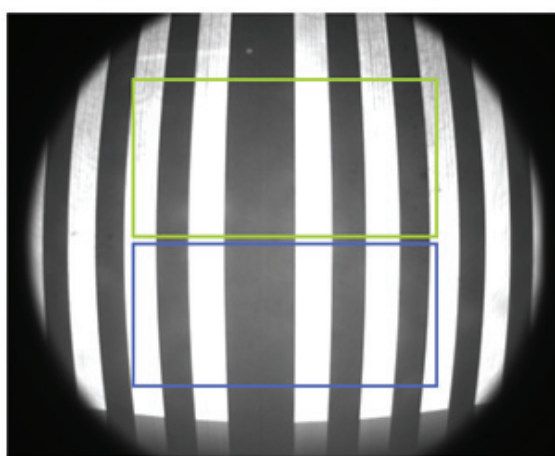
4.2.1 Forbehandling

Siden bildeseriene har ulike lys og kontrastforhold er det nødvendig å gjøre justeringer på referansebildet og bildene i bildeseriene. Dette gjøres ved å strekke histogrammet til bildene slik at den mørkeste pikselverdien blir 0 og den lyseste pikselverdien blir 255. På denne måten blir alle verdiene i gråskalaspekteret brukt. Hensikten er at bildene skal bli likere slik at representasjonen via samforekomstmatrisen blir best egnet til sammenligningen.

4.2.2 Samforekomstmatriser

Det er valgt å se på to utsnitt av bildet. Ett utsnitt over senter og ett under senter av bildet. Grunne til dette er at bilder som har striper i det øverste utsnittet og ikke i det nederste er kandidater til valg av bilde for terskelpunktet. Det motsatte vil være tilfelle for bildet som brukes for endepunktet. Figur 4-4 viser referansebildet med utsnittene markert. Bildet som blir brukt som referanse er nummer 579 i serie 3. De samme utsnittene brukes i både referansebildet og testbildene. Samforekomstmatrisene settes opp for de to utsnittene med funksjonen *graycomatrix*. Det er valgt å telle samforekomstene kun i horisontal retning siden det er begrenset hvor mye rotasjon som oppstår i bildene.

Det er også sett på to ulike størrelser på samforekomstmatrisen. Den ene er på 256x256 som betyr at hver verdi har eget element i matrisen. Den andre er på 32x32 og har da fire verdier i bildet per element i matrisen.



Figur 4-4: Utsnitt som blir brukt for å sette opp samforekomstmatrisene, vist i valgt referansebilde fra serie 3.

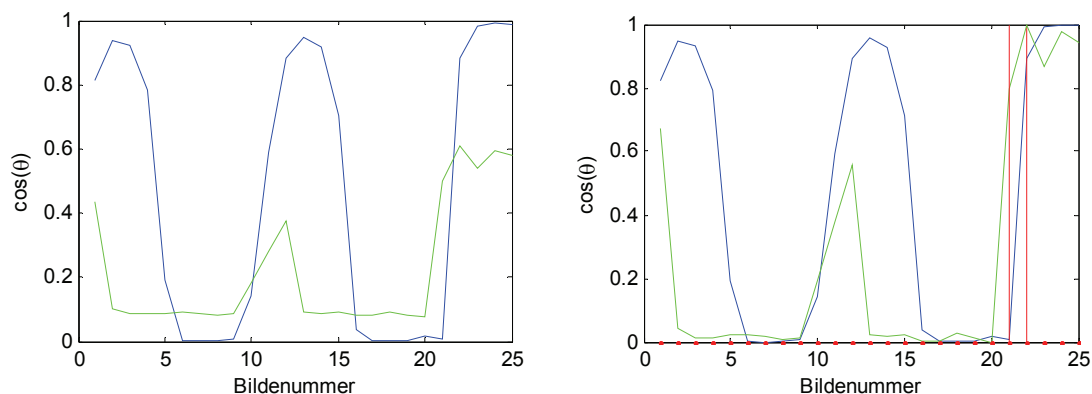
4.2.3 Sammenligne mønster

Sammenforekomstmatrisene sammenlignes med cosinus similaritetstransformen. For å få til dette må de ønskede elementene i matrisene stables i vektorer. Det er valgt å bruke symmetriske matriser og da er det kun nødvendig å se på den ene halvdel av matrisen. Det er sett på to ulike valg av verdier i vektoren. Begge to består av den øvre triangulærhalvdelen men forskjellen er at det ene oppsettet tar med seg diagonalverdien og den andre ikke. Dette viser seg å ha påvirkning på resultatet.

Vektorene sammenlignes med respektive vektorer fra referansemønsteret. Plottet til venstre i figur 4-5 viser metoden brukt på bildeserie 10 som er av terskelpunktet. Kurvene svarer til likheten mellom referansemønster og mønsteret i bildet. Grønn kurve er for grønt utsnitt og det samme med blått. Her er diagonalverdiene i samforekomstmatrisen tatt med i beregningen og størrelsen på samforekomstmatrisen er 256x256. Ser at kurven for det grønne utsnittet viser tydelig hvor mønsteret er mest likt, fra bilde 21 og utover. Den blå kurven er noe vanskeligere å tyde siden den har tre topper. De to første toppene svarer til andre typer striper i bildeserien som opptrer i området før rullebanen. Ser også at ved bilde 22 går verdien opp og holder seg stabil som viser at stripene er i utsnittet.

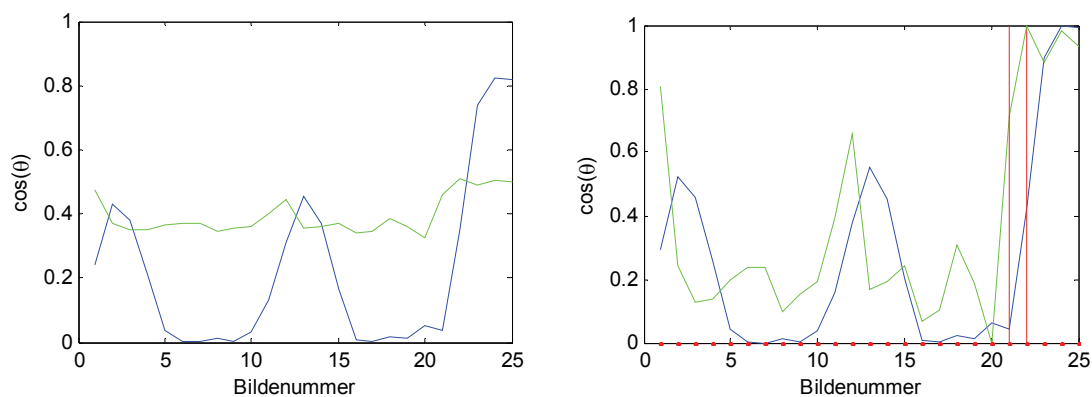
Høyre plot i figur 4-5 viser samme kurver men er her er kurvene normalisert. I tillegg vises resultatet av utført kantdeteksjon på de to likhetsvektorene som røde streker. Ser at

den har funnet en kant ved bilde 21 som svarer til økningen i verdi på den grønne kurven, og en kant ved bilde 22 som svarer til økningen i verdi for blå kurve. Det er da naturlig å tro at det ønskede bilde ligger i området avgrenset av disse to verdiene. Dette viser seg å stemme bra. Bildet som er best å bruke her er enten nummer 21 eller 22 og begge ligger innenfor intervallet gitt av de to linjene.



Figur 4-5: Resultat av sammenligningen brukt på bildeserie 10 med samforekomstmatrise på 256x256 elementer (venstre), normalisering av kurvene, samt resultat av kantdeteksjon vist med røde linjer (høyre).

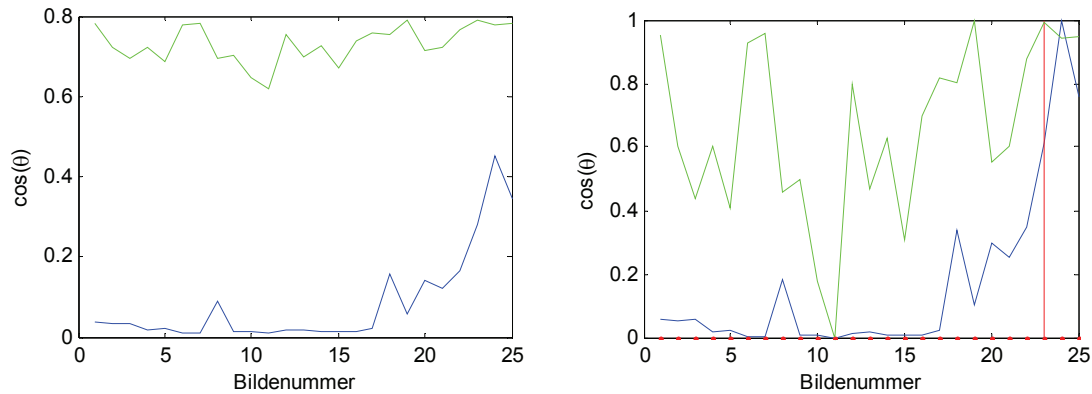
Figur 4-6 viser sammenligning på samme bildeserie som i figur 4-5 men her er størrelsen på samforekomstmatrisen redusert til 32x32. Ser at kurvene blir annerledes men resultatet rundt bildene 21 og 22 er relativt likt.



Figur 4-6: Resultat av metoden brukt på serie 10 med samforekomstmatrise på 32x32 elementer (venstre), normalisering av kurvene, samt resultat av kantdeteksjon vist med røde linjer (høyre).

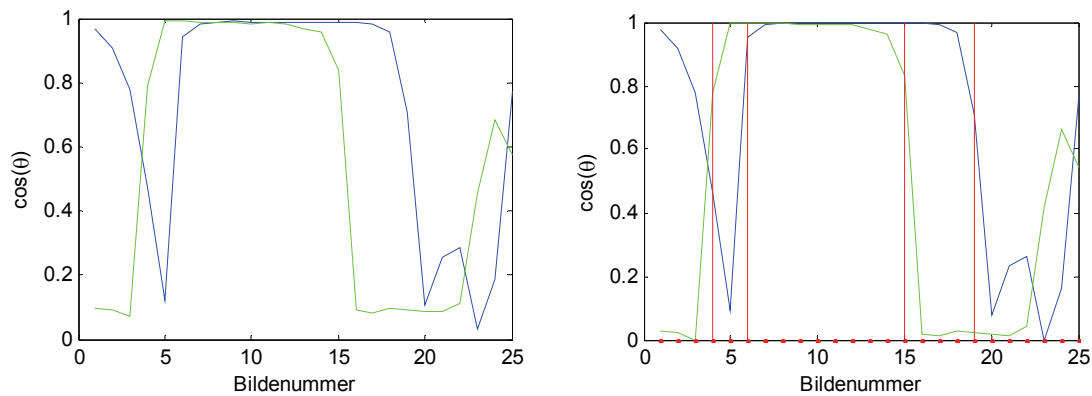
Velges vektorene å settes opp uten å bruke diagonalverdiene vil en ikke ta hensyn til antall samforekomster av like pikselverdier. Dette skulle i teorien bety at sammenligningen blir mindre følsom for blant annet ulik størrelse på stripene, antall striper i utsnittet og lignende. Høyre plot i figur 4-7 viser resultatet av sammenligningen og ser at det er store forskjeller i forhold til i figur 4-5. Den grønne kurven har liten variasjon og har høyest verdi for bilde nummer 1. Den blå kurven får lave likhetsverdier, men har en tydelig økning der mønsteret kommer inn i utsnittet. Venstre plot i figur 4-7 viser de samme kurvene etter normalisering og igjen er det kjørt en kantdeteksjon på vektorene. Ser at det bare er en rød linje og denne svarer til den sterke økningen i verdi for den blå kurven. Den grønne kurven har ingen markant kant og har derfor ikke gitt noe

resultat etter kantdeteksjon. Det viser seg derfor at metoden fungerer best når diagonalverdiene er med i beregningen.

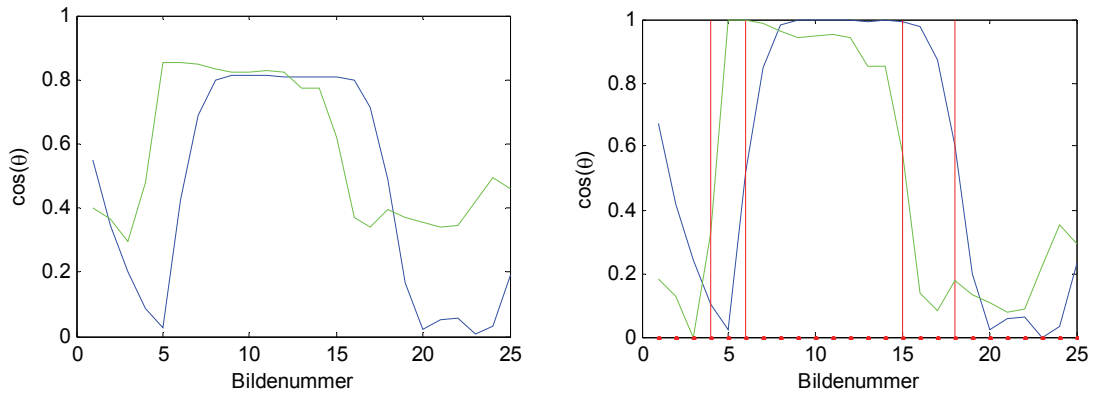


Figur 4-7: Resultat av sammenligning mellom referansene og bildeserie 10 uten diagonalverdier (venstre), samt normaliserte kurver med resultat av kantdeteksjon vist med rød linje (høyre).

For å finne endepunktet, det vil si threshold ved utflyging fra rullebanen blir resultatet noe annerledes. I bildeseriene av dette området vil en detektere kantene til stripene i begge ender. Figur 4-8 viser metoden brukt på serie 11 med en samforekomstmatrise med størrelse 256x256. Figur 4-9 viser samme serie men med samforekomstmatrise med størrelse 32x32. I figurene ser en at stripene kommer inn i bildet ved bilde nummer 3. Dette er enden til stripene inn mot rullebanen og er ikke av interesse. Ved bilde nummer 15 forsvinner stripene i det grønne utsnittet og dette markerer den enden av stripene som er av interesse. Bildet som anbefales brukt her er bilde nummer 16. Ser at dette stemmer bra med der den grønne kurven har fått lav likhetsverdi. Ser også at det normaliserte kurvene har 4 resultater av kantdeteksjonen. Det er de to siste som er interessante, ved 15 og 18, for videre utvelgelse.



Figur 4-8: Resultat av sammenligningen brukt på serie 11 med samforekomstmatrise på 256 verdier (venstre), samt normaliserte kurver med resultat av kantdeteksjon vist med rød linje (høyre).



Figur 4-9: Resultat av sammenligningen brukt på serie 11 med samforekomstmatrise på 32x32 (venstre), samt normaliserte kurver med resultat av kantdeteksjon vist med rød linje (høyre).

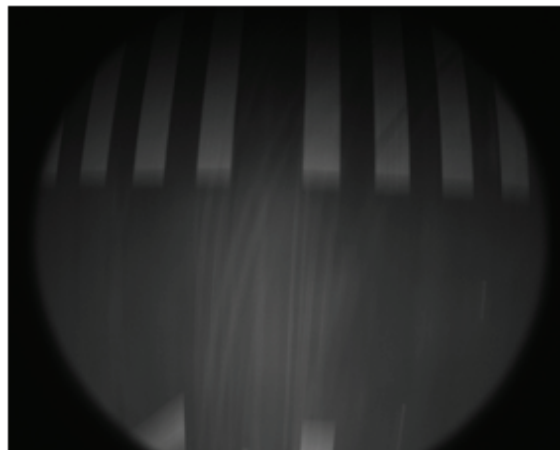
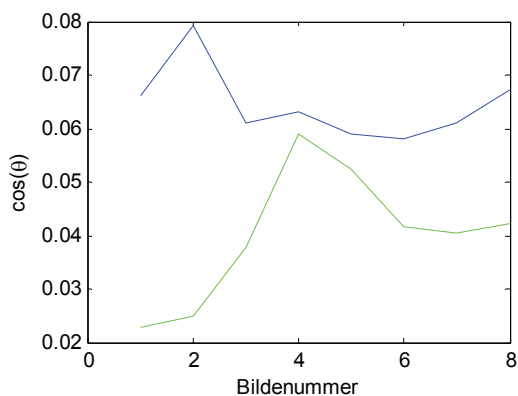
4.2.4 Utvelgelse

Området for hvor det ønskede bildet befinner seg er funnet i resultatvektorene med kantdeteksjon. Resultatet av kantdeteksjonen på likhetsvektoren er en markering av senter av de raske endringene av verdier i sammenligningsvektorene. Bildet som er av interesse ligger midt mellom disse to kantene og velges ut fra dette. Om antall bilder i området er partall må det bestemmes om det skal rundes opp eller ned. Dette vil ikke ha så mye å si siden de horisontale kantene til stripene enten kommer litt over eller litt under senter i det valgte bildet.

4.2.5 Evaluering

Ut fra resultatene som er vist i figurene ser en at metoden fungerer bra for å finne de to referansepunktene. Sammenligningsverdiene er enkle å tyde og resultatet av kantdeteksjonen finner de ønskede områdene i bildeseriene der det ønskede bildet velges ut. Det er også vist at det er nødvendig å ha med diagonalverdiene for å få et resultat av sammenligningen som kan brukes videre. Ved å slå sammen verdier i matrisen kan størrelsen på matrisen reduseres. Ut fra sammenligning av figurene kan det ikke sees noen uønskede konsekvenser med å redusere størrelsen fra 256x256 til 32x32.

Men det er likevel noen problemer med denne metoden. Mønsteret i bildet blir registrert i samforekomstmatrisen på bakgrunn av pikselverdier i bildet. Her oppstår det problemer når egenskaper som lys og kontrastforhold i bildeserien er veldig forskjellig fra referansebildet. Fordi om det blir utført noen justeringer blir resultatet likevel feil. Figur 4-10 viser resultatet av metoden brukt på serie 9 samt bildet med høyest verdi fra den grønne kurven. Ser at resultatet har så lave verdier at mønsteret i praksis ikke er tilstede.

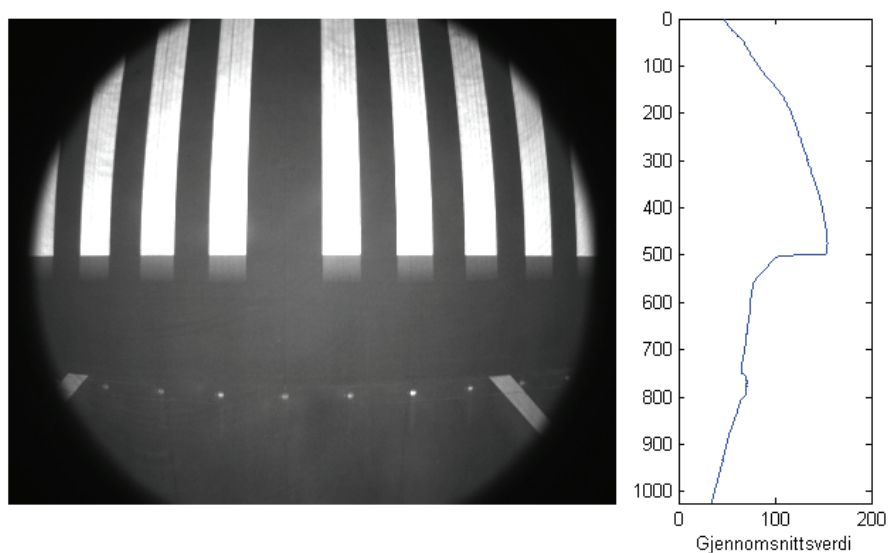


Figur 4-10: Resultat av sammenligning på serie X (venstre), og bildet med høyest verdi fra grønn kurve (høyre).

For å bedre dette problemet må det eventuelt gjøres flere operasjoner på bildeserien for at denne skal bli mer lik referansebildet. Dette vil uansett kreve bruk av flere metoder og vil føre til flere potensielle feilkilder. Denne metoden består allerede av flere trinn der det må settes parametre og dette kan føre til feil.

4.3 Metode 3: Mønstersammenligning av vertikal gjennomsnittsprofil

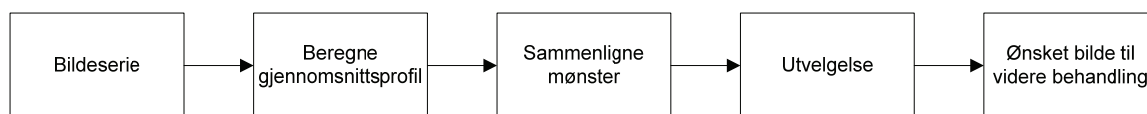
Kravet som blir lagt til grunn for utvelgelsen er at bildet skal ha de horisontale kantene til stripene nærmest midten av bildet. Det er derfor utviklet en metode som direkte bruker denne egenskapen i utvelgelsen. Ved å se på den vertikale gjennomsnittsprofilen til bildet, langs y-aksen, vil kanten av stripene vises som en stor endring i verdiene. Figur 4-11 viser et bilde med kanten i senter og den vertikale gjennomsnittsprofilen til bildet.



Figur 4-11: Gjennomsnittsprofil langs vertikalaksen.

Metoden går ut på å bruke gjennomsnittsprofilen fra et referansebilde og sammenligne denne med gjennomsnittsprofilen til alle bildene i bildeserien. Ut fra resultatet av

sammenligningen kan bildet som svarer til kravet velges ut. Figur 4-12 viser hovedstegene i metoden.



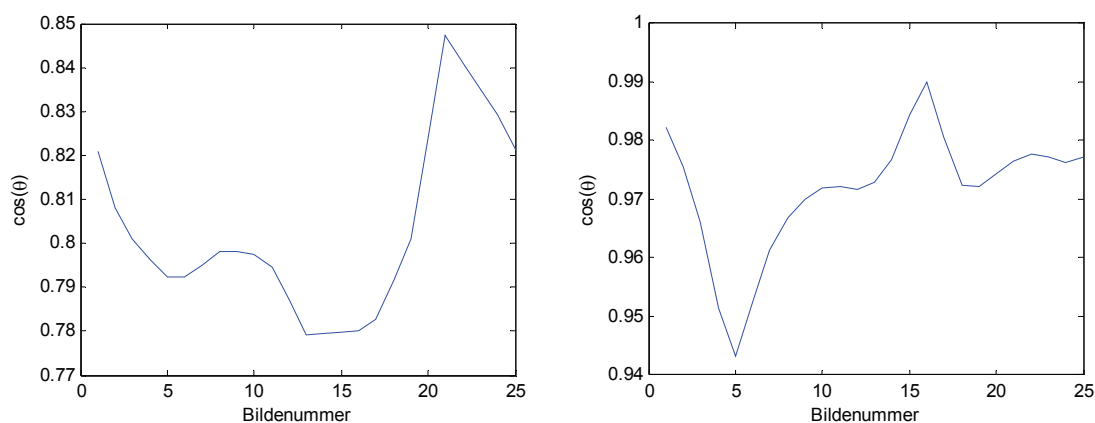
Figur 4-12: Hovedstegene i metode 3 for valg av bilde.

4.3.1 Beregne gjennomsnittsprofil

For hvert bilde i bildeserien beregnes gjennomsnittsprofilen langs vertikalaksen. Dette blir en vektor der hvert element er gjennomsnittsverdien til den respektive raden i bildet.

4.3.2 Sammenligne mønster

Hvert bilde blir sammenlignet med referansebildet via gjennomsnittsprofilen. Sammenligningen utføres med cosinus similaritetstransformen. Referansebildet for terskelpunktet er bilde 575 i serie 3 og for endepunktet bilde 635 i serie 4. Ved å utføre en sammenligning mot respektiv referanse på alle bildene i bildeserien blir resultatet som vist i figur 4-13.

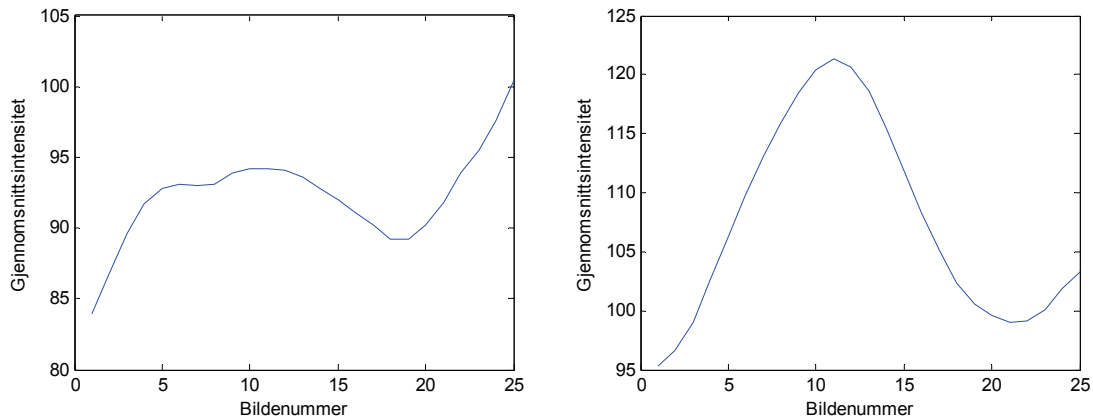


Figur 4-13: Resultat av sammenligning mellom referanse og serie 10 for å finne terskelpunkt (venstre) og serie 11 for å finne endepunkt (høyre).

4.3.3 Utvalgelse

Utvalgelsen er basert på å bruke bildet med størst likhetsverdi. Dette vil være det bildet som er mest lik referanse og som svarer til kravene for videre bruk. Eksemplene i figur 4-13 viser henholdsvis bilde 21 og bilde 16. Dette stemmer bra med visuell vurdering av bildeserien.

En tilleggsverifikasjon for denne metoden kan være å bruke et krav om at bildet skal finnes innenfor en gitt avstand fra bildet med størst intensitet. Bildet med størst intensitet vil normalt være det bildet som har størst andel striper. Figur 4-14 viser gjennomsnittintensiteten for bildene i de to bildeseriene.



Figur 4-14: Gjennomsnittintensitet for bildene i serie 10 (venstre) og serie 11 (høyre).

4.3.4 Evaluering

Det viser seg ved uttesting på bildeserien at metoden fungerer bra. Metoden velger ut bildet som har størst likhet med referansen og dette gir utvelgelse som svarer til kravet. Ved testing på bildeseriene ble riktig bilde valgt ut. Den fungerer også på samme måte for å velge bilde for begge punktene ved å ha en referanse for hver av de to punktene.

Fordelen med denne metoden er at vektorens form er bestemt ut fra bildet og ikke ut fra en transformasjon basert på pikselverdiene som i for eksempel for samforekomstmatrisen. Dette gjør at vektorene til bilder med ulik intensitetsfordeling men samme mønster vil få en vektor med tilnærmet samme form. Forskjellen vil være størrelsen på verdiene. Endringen i gjennomsnittsprofilen der stripene slutter er tydelig og er derfor en egenskap som er lett å gjenkjenne.

Ser at denne metoden løser oppgaven på en enkel og nøyaktig måte. Det er relativt lite beregning og resultatet av sammenligningen kan brukes direkte til å velge ut bildet som svarer til kravet.

5 Korrigerer linseforvringning

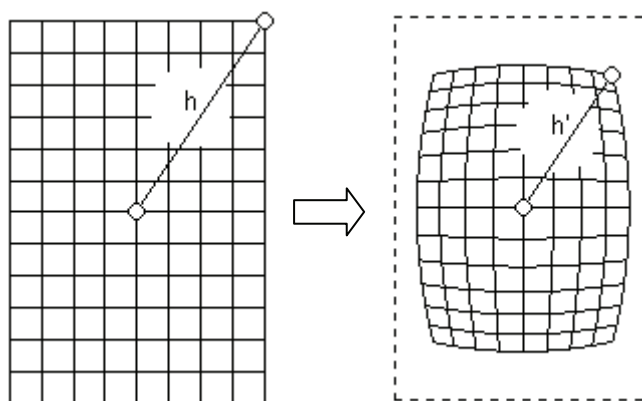
Bildet har forvringning som følge av linsen som medfører at rette linjer blir buet og punkter får en annen posisjon i bildet enn de ville gjort for en linse uten forvringning. Ved beregning av referansepunktets posisjon vil dette føre til at nøyaktigheten blir mer og mer unøyaktig ettersom punktets avstand fra senter øker. For å fjerne linseforvringningen er det nødvendig å utføre en korrigerende på bildet slik at bildepunktene blir flyttet til posisjonene de ville hatt om ikke linseforvringning var tilstede.

I kapittel 3.1.2 ble modellen for linseforvringning satt opp. Kameraet som er brukt til å ta bildene har blitt kalibrert og forvringningsparametrene er derfor kjent. Se vedlegg A.

Det er utviklet en løsning for å fjerne linseforvringning. Denne består av to deler der del en er å beregne hvordan linseforvringningen påvirker posisjonen til bildepunktene. Del to bruker denne beregningen til å korrigerer forvringningen i bildene.

5.1.1 Beregne koordinatmatriser

Ved å bruke modellen for linseforvringning på et bilde vil hvert bildepunkt få en ny posisjon i det forvringte bildet. Denne posisjonen bestemmes av modellen og parametrene til kameraet funnet ved kalibrering. Ved å utføre denne transformasjonen på et bilde med samme størrelse som bildene brukt i bildeserien vil bildepunktene forvringte posisjon beregnes. Figur 5-1 viser hvordan punkter får nye posisjoner som følge av forvringningen. Har nå både de opprinnelige koordinatene for punktene i originalbildet og de forvringte koordinatene til samme punkter beregnet via forvringningsmodellen. Hvert punkt i bildet har da en kjent posisjon i både det opprinnelige og det forvringte bildet.



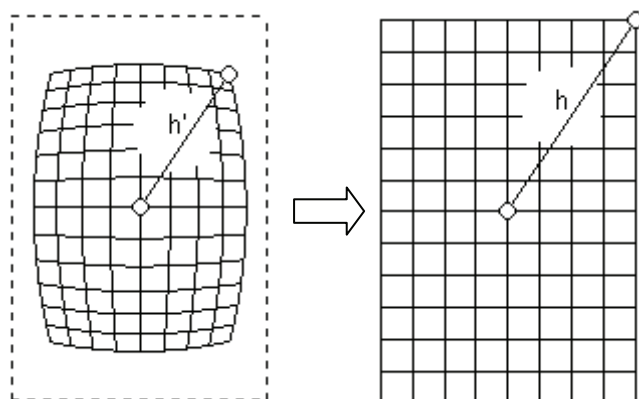
Figur 5-1. Pikslene i originalbildet blir transformert til nye koordinater som følge av modellen for linsekorreksjon.

For å beregne de forvrengte koordinatene til bildepunktene er en algoritme utviklet. Algoritmen *distort.m* er implementert og basert på modellen for linseforvrengning beregner den de to koordinatmatrisene for punktene i det forvrengte bildet.

5.1.2 Transformere bildet på bakgrunn av koordinatmatrisene

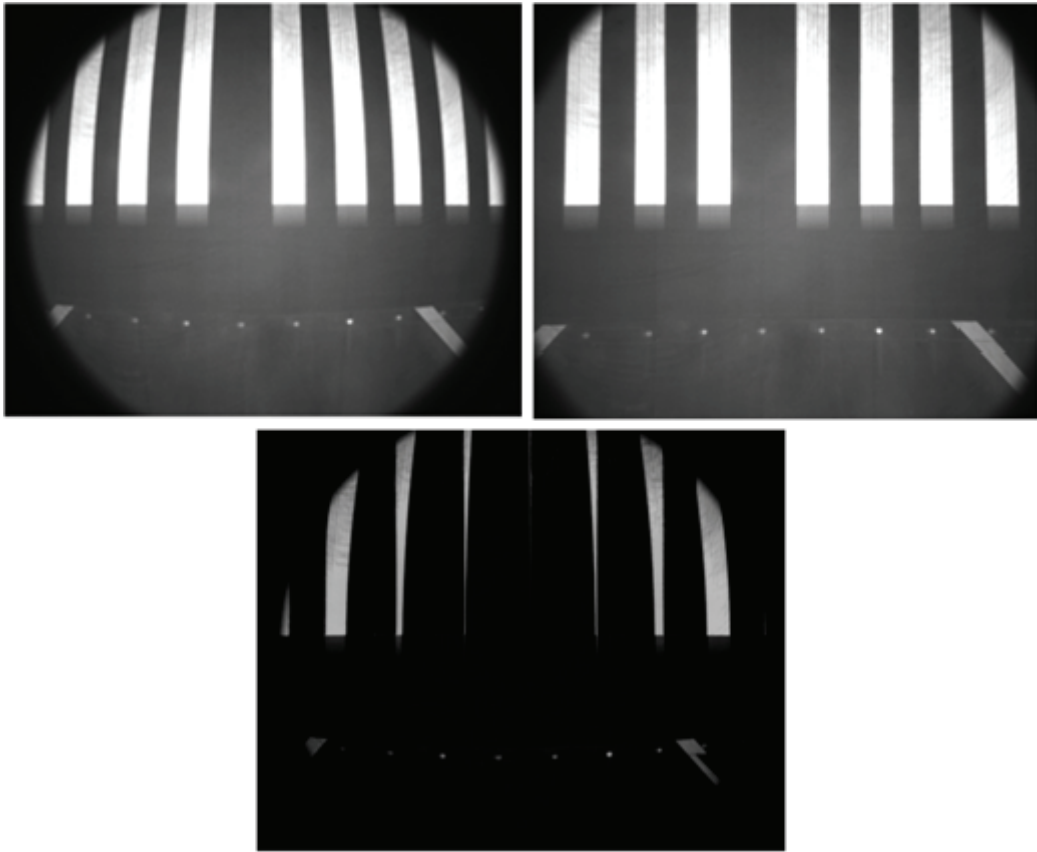
Siden både uforvrengte og forvrengte koordinater er kjent er det nå bare å gå motsatt vei av figur 5-1. Dette gjøres ved transformere det forvrengte bildet basert på sammenhengen mellom forvrengte og uforvrengte koordinatmatriser. Figur 5-2 viser hvordan punkter i bildet gitt av de forvrengte koordinatene blir transformert til respektiv posisjon i det uforvrengte bilde. Ser av figuren at de forvrengte koordinatmatrisene fører til at kun et utsnitt av det forvrengte bildet blir med i det uforvrengte bildet. Dette er logisk siden vidvinkellinsen får med mer informasjon i bildet en det som hadde vært mulig uten forvrengning. For å få med hele det forvrengte bildet kunne størrelsen på det uforvrengte bildet blitt økt. Dette er ikke gjort siden informasjonen ut mot kanten ikke er interessant for videre arbeid.

Punktene som velges ut av det forvrengte bilde vil være en kombinasjon av flere piksler. For å finne pikselens verdi i det uforvrengte bildet brukes da interpolering slik at pikselens verdi blir mest nøyaktig.



Figur 5-2: Utsnitt av forvrengt bilde som blir transformert til uforvrengt bilde.

Ved å utføre denne transformasjonen på bildene i bildeserien vil linseforvrengningen bli korrigert. Figur 5-3 viser et forvrengt bilde og resultatet av å fjerne linseforvrengningen. Figuren viser også forskjellen mellom de to bildene.



Figur 5-3: Originalt bilde (ø. v.), fjernet linseforvringning (ø. h.) og forskjellen mellom de to (nederst).

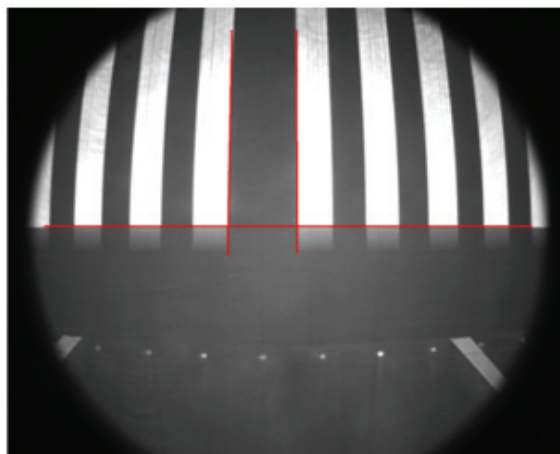
Ser at resultatet av transformasjonen er et bilde uten forvringning. Ser også at differansen mellom de to bildene blir større med avstanden fra senter. Dette stemmer overens med modellen for radiell forvringning. Så lenge punktet er i nærheten av senter i bildet vil beregningen av punktets posisjon kunne utføres på forvrent bilde. Ettersom avstanden øker fra senter vil feilen i beregningen øke og det er derfor nødvendig å fjerne linseforvringningen siden det ikke kan garanteres at punktet alltid er i senter av bilde.

Funksjonen *invdistort* er implementert for å fjerne forvringningen. Denne tar som argument det forvrengte bildet og returnerer det uforvrengt bilde basert på de to posisjonsmatrisene beregnet med *distort.m*.

6 Beregning av referansepunktets posisjon

I forrige kapittel ble det vist ulike metoder for å velge ut bildet for å beregne posisjonen til de to referansepunktene. Dette kapitlet skal ta for seg to ulike metoder som er utviklet for å beregne posisjonen til punktet i det valgte bildet. Metodene i dette kapitlet baserer seg i stor grad på bruk av Hough-transformen for å representere de ønskede egenskapene i bildet.

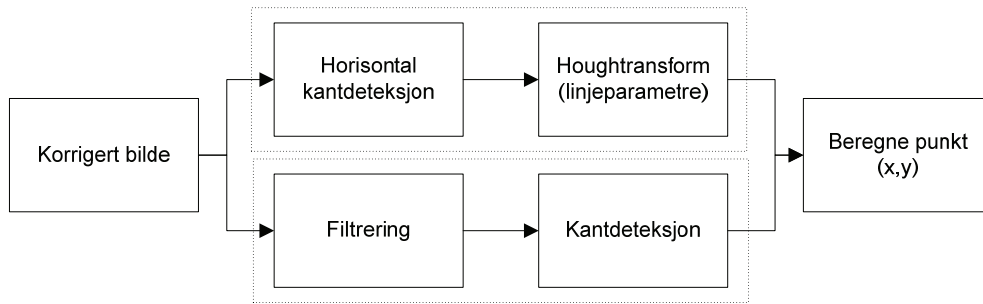
Bildet inneholder mye informasjon og målet er å trekke ut de egenskapene som er nødvendige for å finne punktet. Det er kun tre egenskaper som er nødvendige for å finne punktet. Dette er de to vertikale kantene på stripene inn mot midten av rullebanene og de horisontale kantene som svarer til enden av stripene. Figur 6-1 viser disse tre egenskapene markert med røde linjer. Når disse egenskapene er funnet kan posisjonen til punktet beregnes.



Figur 6-1: De tre egenskapene som er nødvendige for å beregne posisjonen til punktet.

6.1 Metode 1: Kantdeteksjon og Hough-transform

Denne metode for å beregne punktet baserer seg på å finne en linje som tilsvarer de horisontale kantene til stripene ved bruk av Hough-transformen. Deretter beregnes hjørnene til stripene ved å beregne skjæringspunktene mellom de vertikale kantene og linjen. Til slutt beregnes posisjonen til punktet ut fra de to nabo-hjørnene med størst avstand. Figur 6-2 viser hvordan metoden er bygd opp. De ulike modulene blir beskrevet videre i kapitlet.



Figur 6-2: Prinsippskisse for metode 1 for beregning av referansepunktets posisjon.

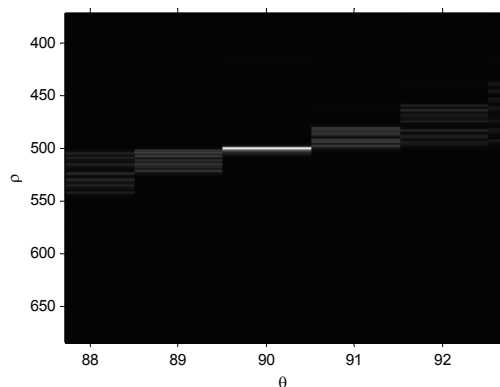
6.1.1 Horisontal linje

For å finne parametrene til en linje som tilsvarer enden på stripene brukes Hough-transformen. Før denne blir brukt på bildet utføres en kantdeteksjon for å finne de horisontale kantene. Dette gjøres ved å bruke funksjonen *edge*. Typen kantdeteksjon som brukes er Sobel med kun deteksjon av horisontale kanter. Siden det er noe uskarphet i bildene i vertikal retning er det horisontale kantene uskarpe og derfor vanskeligere å detektere. Derfor er det nødvendig å detektere disse for seg selv.



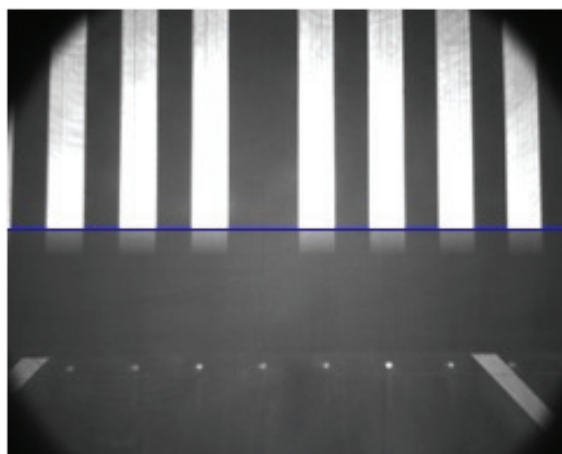
Figur 6-3: Utsnitt av resultatet av horisontal kantdeteksjon utført på bildet i figur 6-1.

På resultatet av kantdeteksjonen brukes den egenutviklede funksjonen *hline* for å utføre Hough-transformen. Denne er implementert slik at det kan bestemmes hvilket område i bildet som skal brukes og for hvilke vinkler det skal letes etter linjer. Fordelen med denne funksjonen er at det ikke er nødvendig å se på områder og vinkler som er uten betydning for resultatet. Siden bildet er valgt på bakgrunn av at det er bildet i bildeserien med enden til stripene er nærmest midten, er det kun nødvendig å utføre Hough-transformen på et område rundt senter av bildet. Dessuten vil vinkelen på linjen være begrenset siden flyet stort sett har lite utslag i forhold til senterlinjen av rullebanen. Det er forventet en vinkel på mellom -10 og +10 grader om horisontalaksen i bildet. Dermed er både inndata og utdata fra Hough-transformen betraktelig redusert. Ut fra definisjonen av koordinatsystemet i kapittel 1.4 vil en horisontal linje i bildet ha en normalvektor med vinkel på 90 grader. Figur 6-4 viser et utsnitt rundt det sterkeste punktet i akkumulatormatrisen for å finne parametrene til den horisontale linjen.



Figur 6-4: Resultat av Hough-transformen i *hline*. Ser tydelig det sterkeste punktet i matrisen som svarer til parametrene til den horisontale linjen.

Siden området er begrenset vil resultatet av Hough-transformen være en akkumulatormatrise med kun en topp som svarer til den horisontale linjen. Funksjonen *hline* returnerer parametrene til denne linjen. Figur 6-5 viser denne linjen plottet sammen med bildet.



Figur 6-5: Bildet med den horisontale linjen som tilsvare enden av stripene.

6.1.2 Vertikale kanter

For å finne de vertikale kantene til stripene er det også brukt kantdeteksjon. Også her er funksjonen *edge* brukt med metoden Sobel. Før kantdeteksjonen utføres en filtrering av bildet med et medianfilter på med størrelse på 15x15. Dette utføres for å redusere detaljgraden slik at kun kantene til stripene skal detekteres.

Kantdeteksjonen blir nå utført i både horisontal og vertikal retning samtidig. Resultatet av dette blir tydelige linjer som tilsvare de vertikale kantene til stripene. Figur 6-6 viser resultatet av kantdeteksjonen på samme bilde som i figur 6-5. Ser at også de horisontale kantene her er detektert men for de fleste bilder vil ikke dette være tilfelle.

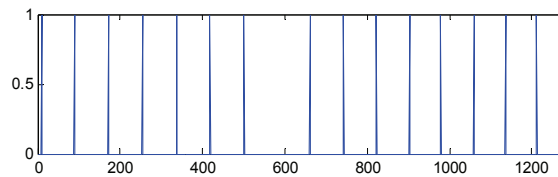


Figur 6-6: Resultat av kantdeteksjon på bildet (utsnitt).

Har nå et bilde der kantene til stripene er markert med verdien 1 og resten av bildet er svart med verdi 0.

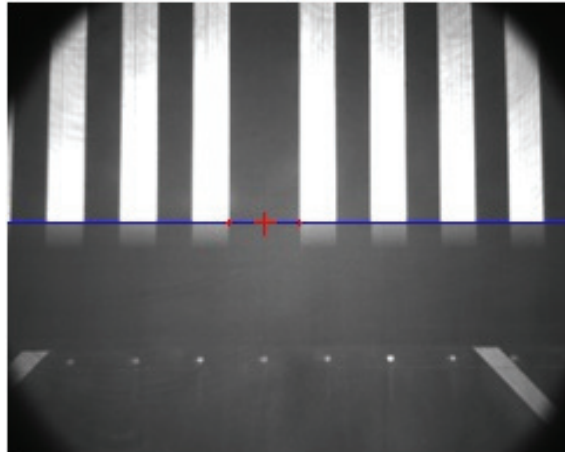
6.1.3 Beregne koordinatene til punktet

For å beregne posisjonen til punktet blir først skjæringspunktene mellom den horisontale linjen og de vertikale kantene funnet ved kantdeteksjonen beregnet. Dette blir en vektor med samme bredde som bildet der skjæringspunktene blir markert. Figur 6-7 viser denne vektoren plottet. Ser tydelig formen til stripene.



Figur 6-7: Plot av vektoren med skjæringspunktene mellom horisontal linje og de vertikale kantene markert med verdien "1".

Siden det er gitt at avstanden mellom stripene om senter av rullebanen er størst er det bare å søke gjennom vektoren med skjæringspunktene og finne de to nabopunktene som har størst avstand. Midt mellom disse to punktene vil da det ønskede punktet ligge og koordinatene til dette er gitt av linjen. Resultatet av beregningen vises i **Feil! Fant ikke referansekilden.** der begge skjæringspunktene og posisjonen til punktet er markert.



Figur 6-8: Terskelpunkt beregnet marker med stort kryss.

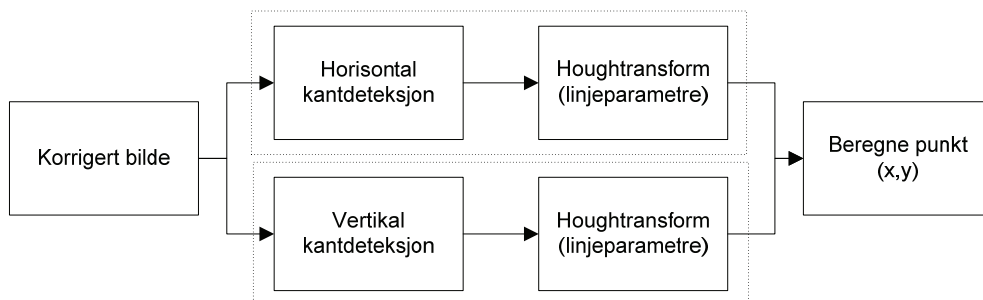
6.1.4 Evaluering

Denne metoden for å finne punktet fungerer bra, men er avhengig av resultatet av kantdeteksjonen for å finne de vertikale kantene. Det er nødvendig at resultatet av kantdeteksjonen for de vertikale stripene er støyfri i området rundt punktet. Mer spesifikt er det nødvendig at det ikke detekteres andre kanter enn kantene tilhørende stripene i området rundt punktet. Dette er nødvendig for at beregningen av posisjonen til punktet skal bli riktig.

6.2 Metode 2: Parametrisering av kanter via Houghtransform

Hough-transformen er en effektiv metode for å finne rette linjer i bildet og denne metoden baserer seg på å representere kantene til stripene kun via Hough-transformen.

Figur 6-9 viser hvordan metoden fungerer. Ser at metoden er lik metode 1 foruten den ekstra Hough-transformasjonen for å finne parametrene til de vertikale kantene. Parametrene funnet fra Hough-transformasjonene brukes videre til å beregne posisjonen til punktet.



Figur 6-9: Prinsippskisse for metode 2 for beregning av referansepunktets posisjon.

6.2.1 Horisontal linje

Den horisontale linjen som tilsvarer enden på stripene må først finnes. Her blir samme fremgangsmåte som i metode 1 brukt. Først blir kantdeteksjon utført på bildet for å finne

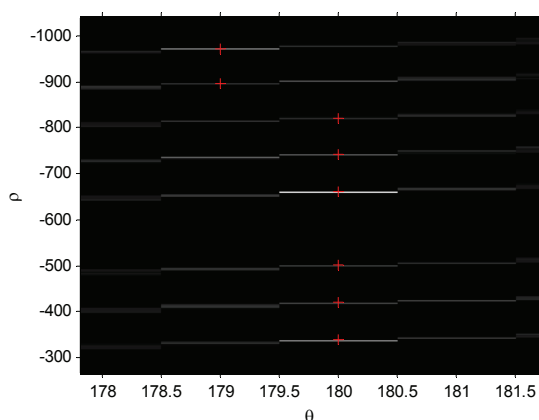
de horisontale kantene og deretter blir Hough-transform utført på resultatet av kantdeteksjonen med funksjonen *hline*. For en mer detaljert beskrivelse se kapittel 6.1.1.

6.2.2 Vertikale linjer

Når de vertikale linjene som tilsvarer kantene skal finnes brukes også samme kantdeteksjon som i metode 1. På resultatet av denne utføres en Hough-transform. Dette gjøres med den egenutviklede funksjonen *vline*. Denne er tilpasset formålet på samme måte som funksjonen *hline*.

Funksjonen *vline* har mulighet for å definere hvilket område av bildet som skal brukes og for hvilket vinkelområde som linjene skal letes etter. Siden de vertikale kantene også vil ha en begrenset vinkel, er det kun nødvendig å se på et begrenset område for vinkelparametrene. Dette er typisk fra -10 til +10 grader om horisontalaksen.

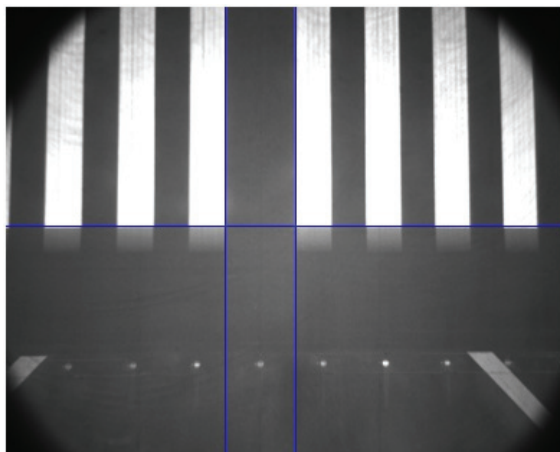
Siden Hough-transformen finner parametrene til normalvektoren til linjene vil også vinkelen til vektorene ligge i området -10 til +10 grader. For å unngå problemet som oppstår med negative vinkler kan parametrene til vektorene gjøres om. Ved å legge til 180 grader på vinklene blir resultatet negative vektorlengder. Resultatet blir derfor det samme og slipper da å jobbe i området rundt 0 grader. Figur 6-10 viser akkumulatormatrisen for *vline* kjørt på bildet og ser at linjene har vinkel rundt 180 grader og negative vektorlengder. Dette stemmer overens med omskrivingen og betyr at linjene har normalvektorer med vinkel 0 grader.



Figur 6-10: Akkumulatormatrisen brukt i *vline* for å finne parametrene til de to ønskede kantene.

Når akkumulatormatrisen er satt opp blir de sterkeste elementene i matrisen funnet ved funksjonen *houghpeaks*. Disse punktene tilsvarer det sterkeste elementet for hver linje som gir parametrene til den respektive linjen.

Siden det kun er to vertikale linjer som er av interesse, de to som tilsvarer kantene om senterlinjen, kan disse to velges ut direkte i akkumulatormatrisen. Dette vil være de to nabopunktene med størst avstand i ρ -verdi. Figur 6-10 viser akkumulatormatrisen med punktene for hver linje markert. Funksjonen *vline* finner de to punktene tilsvarende de ønskede linjene og returnerer parametrene. Figur 6-11 viser de to linjene funnet med *vline* sammen med bildet og den horisontale linjen. Har nå de tre egenskapene for å beregne punktets posisjon.

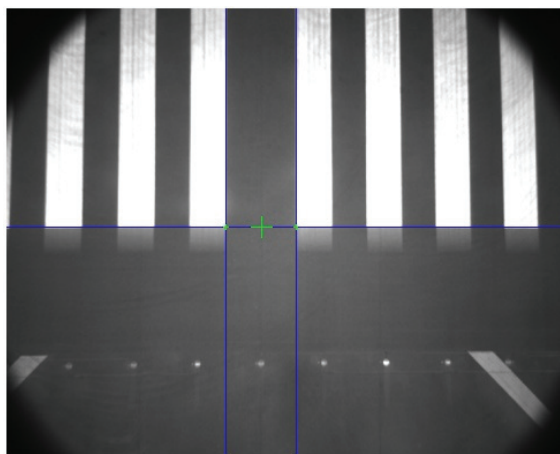


Figur 6-11: Plot av horisontal linje samt linjene tilsvarende de to ønskede kantene funnet med *vline*.

6.2.3 Beregne koordinatene til punktet

De to funksjonene *vline* og *hline* har til sammen returnert tre parameterpar i polare koordinater. Ved å bruke likning (3-9) kan disse uttrykkes som parametre for en linje i planet med stigningstall a og forskyvning langs y -aksen b .

Siden posisjonen til det ønskede punktet er midt mellom de to vertikale linjene langs den horisontale linjen, kan posisjonen til punktet beregnes ut fra dette. Det gjøres ved å finne skjæringspunktene mellom de vertikale linjene og den horisontale linjen. Figur 6-12 viser de tre linjene og skjæringspunktene. Posisjonene til det ønskede punktet er midt mellom disse skjæringspunktene.



Figur 6-12: Bildet med posisjonen til punktet, markert med stort kryss, beregnet ut fra de to skjæringspunktene mellom linjene.

6.2.4 Evaluering

Denne metoden fungerer i hovedsak på samme måte som metode 1. Forskjellen er at istedenfor å bruke de vertikale kantene direkte som i metode1, så transformeres disse med Hough-transformen for å få de representert på parameterform. Dette fungerer bra, men metoden har en svakhet.

Når akkumulatormatrisen er satt opp må de sterkeste elementene for hver linje finnes. Problemet her er at noen linjer kan ha veldig høye verdier i akkumulatormatrisen, mens andre kan ha lave verdier. Problemet blir da å klare å finne et punkt for hver linje som svarer til parametrene for denne linjen. Om utvelgelsen er satt opp feil kan det hende at ikke alle punktene blir funnet som igjen kan føre til feil resultat.

7 Totalløsning

I kapittel 4 og 6 er ulike metoder for å løse de to delproblemene beskrevet. Basert på resultatene av arbeidet med disse metodene kan forslag til totalløsning settes opp. Totalløsningen er implementert i Matlab og testet på bildeseriene som er gode nok for bruk i automatisk deteksjon. Til slutt blir noen forslag til operatørkommunikasjon beskrevet.

7.1 Valg av bilde

Det er utviklet tre ulike metoder for å velge ut bildet. Ut fra uttesting og evaluering av disse ble det klart at metode 3 fungerer best. Dette er en robust metode der sammenligningen baserer seg på en transformasjon basert på koordinater og ikke pikselverdier som de to andre. Metode 2 har også tilfredsstillende resultat men her oppstår det problemer når bildeserien har lav kontrast og ulike lysforhold i forhold til referansen. Metode 1 viste seg under utvikling å være lite egnet og er derfor ikke vurdert som kandidat.

Metode 3 er den enkleste og mest nøyaktige måten for å finne det ønskede bildet. Sammenligningen er direkte på den egenskapen som er satt som krav og dette fører til et nøyaktig resultat som ikke trenger videre behandling. Det er i forslag til løsning ikke brukt noen tilleggsverifisering av resultatet som nevnt i kapittel 4.3.

7.2 Beregning av referansepunktets posisjon

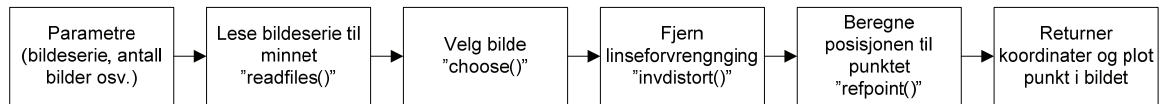
To ulike metoder for å beregne posisjonen til det kjente punktet er også utviklet. Disse fungerer stort sett likt men metode 2 krever en ekstra Hough-transform og utvelgelse av de to linjene om senterlinjen. Her kan det oppstå problemer siden det er nødvendig at det blir funnet maksimumselementer for hver linje akkumulatormatrisen for at resultatet skal bli riktig. Dette fører til flere operasjoner som kan føre til feil.

Metode 1 har den fordelen at den beregner skjæringspunktene direkte ut fra resultatet av den vertikale kantdeteksjonen og den horisontale linjen. Dette betyr mindre beregning men krever også at resultatet av kantdeteksjonen er nøyaktig i områdene rundt skjæringspunktene.

Ved testing av de to metodene viste det seg at metode 1 hadde bedre nøyaktighet enn metode 2. Dette kommer hovedsakelig av problemet med utvelgelse av parametrene i akkumulatormatrisen for de vertikale linjene som nevnt tidligere. Det er derfor valgt å implementere metode 1 i totalløsningen.

7.3 Implementasjon av totalløsning

Totalløsningen er implementert i Matlab og testet ut på dataseriene. Løsningen er implementert med en hovedfil og metodene er implementert som funksjoner i egne filer som hovedfilen kaller opp. Hovedfilen styrer dataflyten mellom funksjonene. Figur 7-1 viser hvordan den implementerte algoritmen fungerer.



Figur 7-1: Flytskjema for algoritmen styrt av hovedfilen *main.m*.

Parametrene som settes bestemmer hvilken bildeserie som skal prosesseres og hvor mange bilder den består av. Det må også defineres hvilket av de to punktene som skal finnes. Referansebildene for utvelgelse settes direkte i funksjonen for utvelgelse.

Hovedfunksjonene i algoritmen er følgende utviklede funksjoner:

- **readfiles()** leser bildeserien som skal behandles inn i minnet slik at utførelsen av de andre funksjonene går raskere.
- **choose()** er den implementerte metoden for å velge ut bildet. Denne er implementert med metode 3. Referansebildene som skal brukes i utvelgelsen settes internt i denne funksjonen.
- **invdistort()** korrigerer for linseforvrengningen som beskrevet i kapittel 5. Denne er avhengig av koordinatmatrisene beregnet i *distort.m*.
- **refpoint()** finner posisjonen til referansepunktet i det valgte bildet, implementert med metoden 1. Resultatet av denne er koordinatene til punktet.

Resultatet av algoritmen er plot av valgt bilde med punktet markert. Algoritmen returnerer også bildenummer og referansepunktets koordinater i bildet.

Funksjonene beskrevet ovenfor er kun hovedfunksjonene i arbeidet. Hver funksjon bruker igjen egenutviklede funksjoner som *vline*, samt Matlab-funksjoner som *edge*, *medfilt2*, *houghpeaks*, *graycomatrix* osv. Til sammen blir funksjonene den totale algoritmen som løser problemet.

7.4 Operatørkommunikasjon

Operatørkommunikasjon er ikke en definert del av oppgaven men det er likevel valgt å se på hvordan dette kan løses.

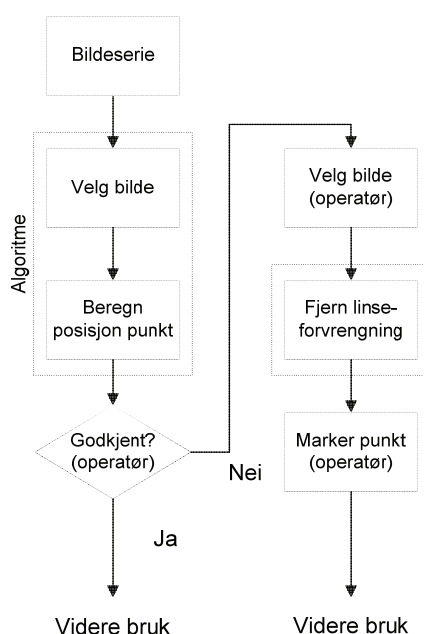
Målet med arbeidet er å lage en algoritme som automatisk finner punktet i bildeserien. Dette kan da brukes videre til posisjonsberegning av flyet. Det enkleste vil da være å holde operatøren ute fra gjenkjenningen av punktet og overlate alt til programmet. Men siden bildeseriene kan være veldig ulike grunnet lysforhold, rullebane, vær og flyets oppførsel i lufta kan det ikke garanteres at algoritmen returnerer det ønskede resultatet.

Det er derfor nødvendig å involvere operatøren slik at han kan godkjenne resultatet og oppdage feil.

Om systemet har beregnet feil punkt må operatøren ta over oppgaven med å finne punktet. Dette kan da for eksempel gjøres ved at han får vist hele bildeserien, velger bildet han vil bruke videre, systemet fjerner linseforvrengning, og operatøren markerer punktet. Figur 7-2 viser ett forslag til flytskjema for operatørinteraksjonen.

I de fleste tilfeller med bra bildeserie vil algoritmen klare å finne punktet og da vil det bare være for operatøren å godkjenne resultatet. I tilfellen der resultatet blir feil vil det ofte også være vanskelig for operatøren å markere punktet.

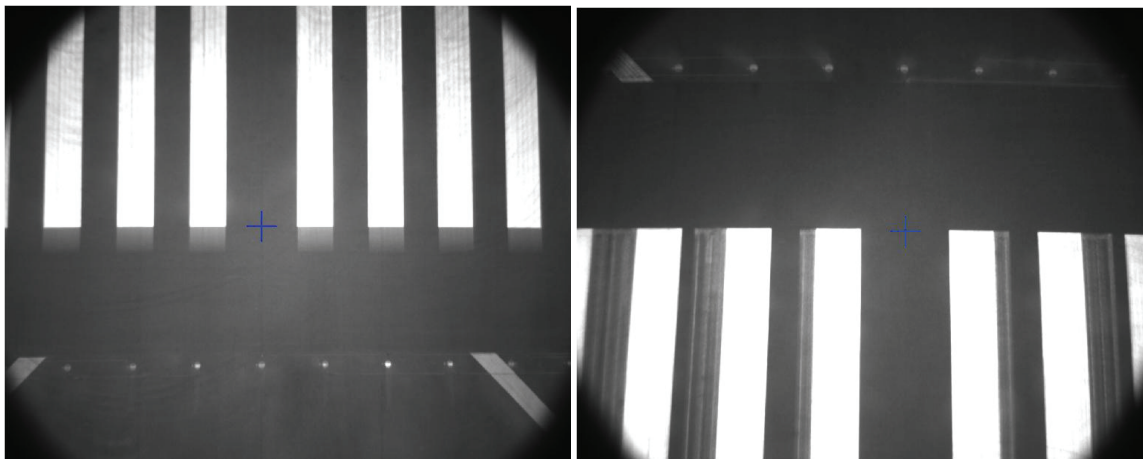
Et ferdig system burde også ha mulighet for å sette kameraparametre, referansebilder og andre parametre som bør kunne endres i algoritmen.



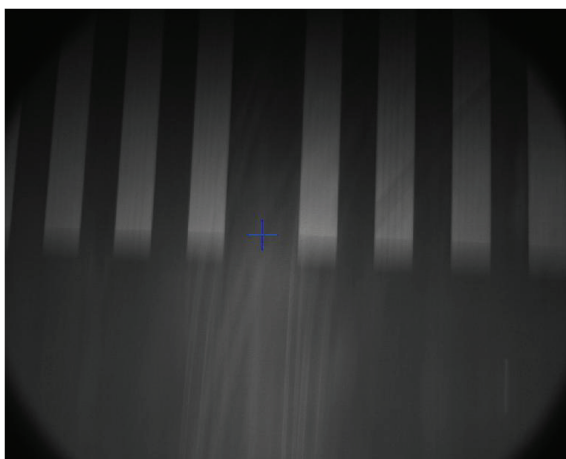
Figur 7-2: Forslag til operatørinteraksjon for implementert algoritme i eksisterende system.

8 Resultat

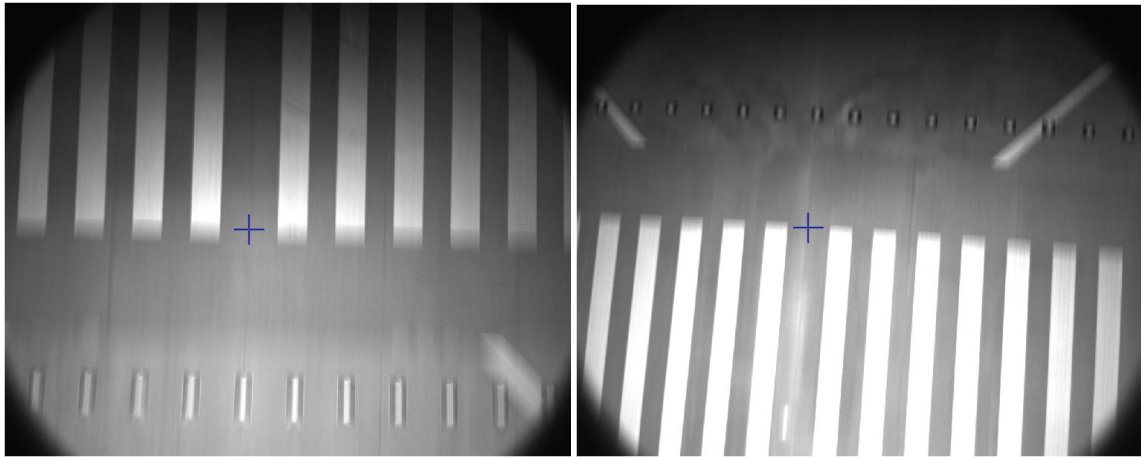
Den implementerte algoritmen i kapittel 7 testes ut på de 9 bildeseriene i tabell 2-1 som har god nok kvalitet for bruk til automatisk gjenkjenning. Referansebildene for utvelgelsen er de samme som ble brukt under utvikling og uttesting av metode 3 i kapittel 4.3. Figur 8-1 til figur 8-5 viser resultatet av testing på de 9 bildeseriene. De to bildene i hver figur kommer fra samme kontrollflygning og den beregnede posisjonen til referansepunktene er markert med blått kryss. Diskusjon av resultatet følger i kapittel 9.



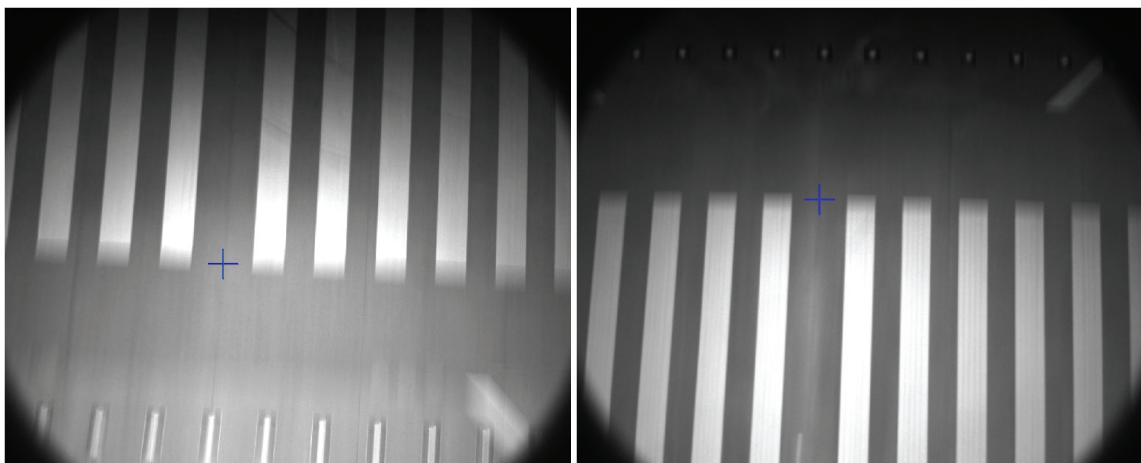
Figur 8-1: Utvalgt bilde og beregnet referansepunkt markert med blått kryss i bildeserie 3 og 4.



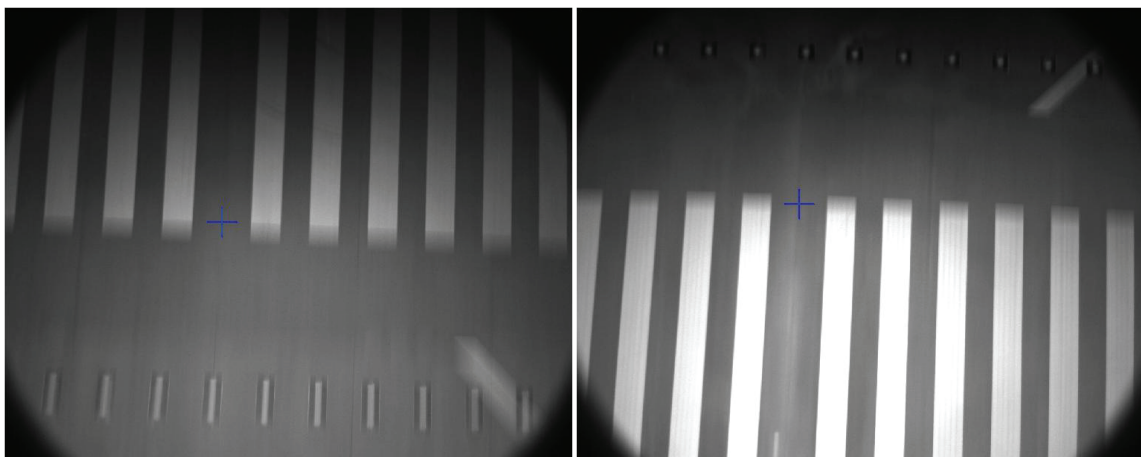
Figur 8-2: Utvalgt bilde og beregnet referansepunkt markert med blått kryss i bildeserie 9.



Figur 8-3: Utvalgt bilde og beregnet referansepunkt markert med blått kryss i bildeserie 10 og 11.



Figur 8-4: Utvalgt bilde og beregnet referansepunkt markert med blått kryss i bildeserie 12 og 13.



Figur 8-5: Utvalgt bilde og beregnet referansepunkt markert med blått kryss i bildeserie 14 og 15.

9 Diskusjon

I dette kapitlet vil arbeidet presentert i rapporten bli vurdert. Det blir først utført en vurdering av resultatene fra uttesting av algoritmen i kapittel 8. Deretter blir de ulike metodene utviklet for å løse problemet vurdert. Forslag til videre arbeid følger i kapittel 10.

9.1 Resultat

Uttesting av totalløsningen på bildeseriene viser et resultat som svarer til forventningene. Algoritmen finner både riktig bilde og beregner posisjonen til punktet for alle bildeseriene den er testet på. Utvelgelsen av bildet skjer etter kravet om at bildet skal ha punktet nærmest senter i bildet. Punktets posisjon beregnes innenfor kravene på et avvik på 3-5 piksler horisontal retning. Hvorvidt dette kravet er oppfylt kan ikke beregnes eksakt siden den virkelige posisjonen til punktet ikke er markert i bildet. Det er derfor kun utført manuell analyse av resultatet og dette antyder at kravet i horisontal retning er oppfylt.

Kravet til nøyaktighet i vertikal retning er på 30-50 piksler. Det er noe uskarphet vertikal retning i bildene og i et bilde fra 20 meters høyde tilsvarer størrelsen på det uskarpe området ca 40 piksler, avhengig av flyets fart. Alle punktene er beregnet innenfor det uskarpe området. Dette betyr at beregningene oppfyller kravet til nøyaktighet i vertikal retning. Posisjonen beregnet i vertikal retning vil også ha noe å si for posisjonen horisontalt når bildet har mye rotasjon. Men virkningen av dette vil begrense seg til mellom 0-2 piksler i de fleste tilfeller.

Resultatet av metoden for å beregne posisjonen til punktet er en tilnærming til mønsteret i bildet. Hvert trinn inneholder tilnærminger og kan føre til feil i resultatet. Hovedkildene til feilberegning kommer som følge av uskarphet i bildet og utvelgelse av parametre for den horisontale linjen. Uskarpheten i vertikal retning fører til at kantdeteksjonen ofte ikke finner noen tydelig kant og resultatet er at den horisontale linjen får noe avvik. I metoden er det en del avrundinger siden alle indeksene til bildeelementene er heltall. Dette kan føre til noe avvik men det påvirker resultatet i mindre grad.

Algoritmen er testet på for få bildeserier til å kunne utføre statistisk analyse på nøyaktigheten av resultatene. Men egenskapene til de ulike bildeseriene er så forskjellige at de er representative for hvilke bildeserier som kan forventes at algoritmen skal kunne fungere på. Det er derfor stor sannsynlighet for at resultatet av uttestingen er representativt for algoritmens evne til å produsere resultater innefor kravene til toleranse.

9.2 Metoder

For beregning av referansepunktets posisjon i bildet ble metode 1 valgt. Denne produserer nøyaktige svar og beregner posisjonen til punktet som ønsket. Denne metoden ble valgt å bruke i totalløsningen siden den har bedre resultat enn metode 2.

Hovedutfordringen med denne metode 2 er ved utvelgelsen av parametrene til de to vertikale linjene. Dette kan føre til feil om ikke parametrene til alle linjene blir funnet i akkumulatormatrisen. Metode 1 har derimot ikke dette problemet. Her brukes resultatet av kantdeteksjonen direkte i beregningen av punktet. Fordelen med dette er at en slipper den ekstra utvelgelsen av parametre for de to vertikale linjene. Ulempen er at metode 1 avhenger av at resultatet av kantdeteksjonen er nøyaktig der hvor den horisontale linjen skjærer kantene. Hvis det er diskontinuitet eller forstyrrelser i kantdeteksjonen kan dette føre til at skjæringspunktet ikke blir funnet eller registrert feil. Dette vil føre til avvik i posisjonsberegningen til punktet. Det viser seg under testing på bildeserien at avviket mellom de to metodene som oftest ligger innenfor 0-2 piksler.

Når det gjelder utvelgelsen av bilde er metode 3 mest eksakt. Denne metoden velger bilde direkte ut fra kravet om at kanten skal være nærmest senter av bildet. Den fungerer på samme måte for begge de to referansepunktene og resultatet av uttesting viser at denne gir riktig valg av bilde for alle bildeseriene. Metode 2 fungerer også bra, men er mer følsom for forskjeller mellom referansebildet og bildeseriene. Selv om noen av egenskapene i bildene er forsøkt justert før sammenligning fungerer ikke denne like bra som metode 3. Dette kommer av at transformasjonen til samforekomstmatrisen utføres på bakgrunn av pikselverdiene. Ulike bilder gir da ulike samforekomstmatriser. Metode 1 viser seg å være lite egnet til å velge ut bildet og fører ofte til feil resultat.

I arbeidet er det benyttet kjente metoder og fremgangsmåter. For mønstergjenkjenning i valg av bilde er cosinus similaritetstransform brukt. Dette er en mye brukt metode innenfor mønstergjenkjenning. Ved beregning av posisjonen til referansepunktene er fremgangsmåten å forbedre bildet, segmentere for å få frem egenskapene og så analysere resultatet for videre beregninger.

Selv om fremgangsmåtene og de underliggende metodene er velkjente er utfordringen å tilpasse de kjente metodene til det spesifikke problemet. Arbeidet vist i denne rapporten har resultert i en algoritme som løser problemet som ønsket. Funksjonaliteten og metodene brukt i løsningen kan brukes direkte videre i eksisterende system for automatisk beregning av posisjonen til referansepunktene. Det må likevel antas at problemer vil kunne oppstå og at justeringer på algoritmen må utføres.

10 Videre arbeid

Resultatet av arbeidet er en algoritme som løser problemet som ble satt innledningsvis. I diskusjonen ble det nevnt at antall bildeserier som algoritmen er testet på er for få til å si noe om algoritmens globale prestasjon. Videre arbeid vil derfor kunne bestå av to hovedretninger. Enten implementering og uttesting i eksisterende system for ILS-kalibrering, eller videre uttesting og utvikling på flere datasett for å undersøke algoritmens nøyaktighet videre.

Resultatet av algoritmen testet på bildeseriene viser at metodene valgt for å løse problemet fungerer. Siden en kan se på bildeseriene som et representativt utvalg kan videre arbeid gå ut på å implementere algoritmen i det eksisterende system. De fleste metodene som brukes i algoritmen er enkle nok til å kunne implementeres manuelt. Det kan også vurderes om det er enklere å kjøpe pakker med ferdige funksjoner for programmeringsspråket brukt i det eksisterende systemet. Når algoritmen er implementert kan den testes ut enten ved simulering eller ved kontrollflygninger.

Videre arbeid kan også gå ut på å gjøre videre testing og utvikling av algoritmen i Matlab. Eventuelle feilkilder og problemer med metodene kan da fjernes før videre implementering i det eksisterende systemet. Under uttesting kan eventuelt begge metodene for å beregne posisjonen til referansepunktene testes for å sammenligne nøyaktigheten på de to.

Bildeseriene har noe varierende kvalitet og videre arbeid kan da også gå ut på å forbedre kvaliteten på bildene. Dette kan også være med på å forbedre resultatet siden uskarpheten som oppstår i fartsretningen som følge av lukkertiden medfører uskarpe kanter. Dette fører igjen til unøyaktigheter i resultatet. Målet må da være å finne et optimalt forhold mellom lukkertid og kvalitet på bildene. Det kan også vurderes om det skal brukes fargekamera slik at ulike kombinasjoner av fargekanalene kan mikses for å få et tydeligere mønster [4].

Referanseliste

Litteraturreferanser

- [1] Kim E, Walter T, Powell J.D. (2007, mai). Standalone GPS-Based Flight Inspection System, Stanford University, Stanford. Paper presentert på European Navigation Conference GNSS/TimeNav 2007, Geneve, Sveits.
- [2] Feit CM, Bates MR. (1994, april). Accurate positioning in an Inertial-Based Automatic Flight Inspection System Using Differential Global Navigation Satellite Systems, Res. Div., Sierra Technologies Inc., Buffalo, NY. Paper presentert på Position Location and Navigation Symposium 1994, IEEE, Las Vegas, NV, USA.
- [3] Dybek J, Hähndel T, Jagieniak S. (2004, juni). Position Reference by Automatic Threshold Detection with a Camera System. Presentasjon på International Flight Inspection Symposium (IFIS 2004), Montreal, Canada.
- [4] Bacha AR. Line Detection and Lane Following for an Autonomous Mobile Robot [Hovedoppgave]. Blacksburg: Virginia Polytechnic Institute and State University; 2005.
- [5] Shang J, Shi Z. Vision-based Runway Recognition for UAV Autonomous Landing. International Journal of Computer Science and Network Security. 2003; 7(3):112-117.
- [6] Gonzalez RC, Woods RE. Digital Image Processing. 2. utgave. Upper Saddle River: Prentice Hall; 2002.
- [7] Sonka M, Hlavac V, Boyle R. Image Processing, Analysis, and Machine Vision. 2. Utgave. Pacific Grove: Brooks/Cole Publishing Company; 1999.
- [8] Fjerdingen SA. Optisk positurestimering for 5 frihetsgrader [Hovedoppgave] Trondheim: Norges teknisk-naturvitenskapelige universitet; 2007.
- [9] Hough PVC. Method and means for recognizing complex patterns. US Patent No. 3,069,654; 1962.
- [10] Duda RO, Hart PE. Use of the Hough Transformation To Detect Lines and Curves in Pictures. Communications of the ACM. 1972; 15(1):11-15.

- [11] Kyewook L. Application of the Hough Transform [Paper]. Lowell: University of Massachusetts; 2006.
- [12] Liu C. The Bayes Decision Rule Induced Similarity Measures. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2007; 29(6):1086-1090.

Internettreferanser

- [i1] Wikipedia. Instrument landing system. Hentet 15. april 2008 fra http://en.wikipedia.org/wiki/Instrument_Landing_System
- [i2] California Institute of Technology. Camera Calibration Toolbox for Matlab. Hentet 17. mars 2008 fra http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

Bildereferanser

- [b1] ATC and Aviation Glossary. Hentet 23. april fra <http://phlairline.com/aviationglossary.html>
- [b2] Pinhole Camera. Wikipedia. Hentet den 5. mai 2008 fra http://en.wikipedia.org/wiki/Pinhole_camera
- [b3] Illustrasjon av perspektiv projeksjon. Hentet 5. mai fra http://upload.wikimedia.org/wikipedia/commons/9/90/Perspective_Projection_Principle.jpg
- [b4] Distortion. Hentet 12. mai 2008 fra <http://www.vanwalree.com/optics/distortion.html>
- [b5] Zhao J, Yu S, Cai H. A precise method for correcting stereo pairs collected by a new binocular device. Industrial Robot: An International Journal. 2007; 34(6):512-519. Figur hentet 22. mai 2008 fra <http://www.emeraldinsight.com/fig/0490340608022.png>
- [b6] Hentet fra Matlab hjelpebibliotek for funksjonen *graycomatrix* i pakken Image Processing Toolbox.

Referansepersoner

- [b7] Jostein Trones, Norwegian Special Mission AS

Vedlegg A Kamerakalibrering

Resultater av kamerakalibrering oppgitt av NSM [p1].

Field of view (FOV)

$$FOV = 2 \arctan \frac{w}{2f}$$

Her er $f=5 \text{ mm}$ som er brennvidden. w er størrelsen på filmen. Siden filmen her er en digital sensorbrikke er det størrelsen på den som er interessant. Brikken er på 1280×1024 piksler og avstanden mellom hver piksel (*pixel-spacing*) er på $6,7 \mu\text{m}$.

FOV i x-retning:

$$FOV = 2 \arctan \frac{1280 \cdot 6,7}{2 \cdot 5} = 81,2^\circ$$

FOV i y-retning:

$$FOV = 2 \arctan \frac{1024 \cdot 6,7}{2 \cdot 5} = 68,9^\circ$$

Effektiv brennvidde f_c : Parameteren f_c er effektiv brennvidde uttrykt i antall piksler horisontalt og vertikalt. For kameraet er disse verdiene målt til 748,70710 og 748,30147. Dette gir en effektiv brennvidde på $748.70710 \cdot 6.7 \mu\text{m} = 5.016 \text{mm}$ horisontalt og $748.30147 \cdot 6.7 \mu\text{m} = 5.014 \text{mm}$ vertikalt. Dette er svært nært linsens oppgitte brennvidde på $f=5 \text{ mm}$.

Linseforvrengning k_c : Radiell og tangentiell forvrengning oppgis i form av vektoren k_c .

$$k_c = \{-0.23693, 0.09255, -0.00444, 0.00182, 0.0000\}$$

Pikselsymmetri α_c : Parametren α_c beskriver pikselsymmetrien. Denne er for dette kameraet 0,0 som betyr at avstanden mellom pikselen i horisontal og vertikal retning er lik.

Ekte senter c_c : Bildesensorens teoretiske senter vil være (640, 512). Ved kalibrering av kameraet viste denne seg å være (633.41, 514.23).

Vedlegg B Digitale vedlegg

Vedlagt ligger CD med følgende innhold:

- Kildekode i Matlab-format.
- Bildeseriene brukt i arbeidet.
- Rapporten i digitalt format.
- Diverse referanser.