

Norges teknisk-
naturvitenskapelige universitet
NTNU

Fakultet for ingeniørvitenskap og teknologi
Institutt for bygg, anlegg og transport
Studieretning Geomatikk



Postadresse
Høgskoleringen 7a
7491 Trondheim
Telefon 73 59 45 80
Telefax 73 59 70 21

MASTEROPPGAVE

VÅREN 2005

For

Stud. techn. Anders Hveem Malum

Metoder for automatisert oppmåling av flater i nærfotogrammetri

**Methods for automated surface measurement in close-range
photogrammetry**

Bakgrunn

Fotogrammetri har gjennom mange år vist seg nyttig til oppmåling av kulturminner. Dette har imidlertid ikke ført til rutinemessig bruk av fotogrammetri for oppmåling av kulturminner. Det har vist seg vanskelig å komme fram til metoder som både gir ønsket detaljeringsgrad og nøyaktighet, og som gir resultater raskt og til en rimelig kostnad. Automatisering er en forutsetning for å oppnå dette, dersom nøyaktigheten også kan ivaretas.

Dokumentasjon av helleristningsfelt er ett av mange anvendelsesområder der det er behov for nøyaktig og detaljert oppmåling av flater.

Oppgave

Kandidaten skal studere, beskrive og vurdere algoritmer og metoder som kan egne seg for automatisk måling av mange punkt på tredimensjonale flater. Utgangspunkt er digitale

bilder, og det endelige resultatet skal være en digital flatemodell. I tillegg til stereo-matching, skal det tas med metoder som innebærer multibilde-matching og konvergente opptak.

En aktuell metode skal velges ut. Etter eventuell videreutvikling, skal denne metoden prøves ut på eksisterende bilder fra helleristningsfeltet Lurfald i Stjørdal. Det kan også være aktuelt å prøve metoden på andre bilder.

Forord

Denne rapporten oppsummerer arbeidet med min masteroppgave ved faggruppe Geomatikk våren 2005. Det overordnede temaet for oppgaven er digital nærfotogrammetri og bildematching. Dette er et tema som spenner over flere fagfelt, med særlig vekt på bildebehandling og fotogrammetri. Som geomatikkstudent med bakgrunn fra institutt for datateknikk og informasjonsvitenskap føler jeg at oppgaven inkluderer flere av de viktigste emnene i studiet mitt.

Arbeidet med oppgaven har hatt en fin fordeling mellom praksis og teori. Underveis har jeg selvfølgelig støtt på problemer, og ting som jeg trodde var rimelig opplagt har vist seg å ikke være det. Det har vært særlig utfordrende å velge riktig bakgrunnsmateriale. Prosessen har vært utviklende, og det har vært veldig interessant å få fordype seg i en enkelt problemstilling.

Denne rapporten tar utgangspunkt i at leseren besitter en del kunnskap innenfor matematikk og billedbehandling. Det er imidlertid gitt en kort oversikt over noen benyttede matematiske begreper og symboler i appendix A.

Jeg ønsker å takke professor Knut Ragnar Holm for god veiledning og tilbakemelding, og for at han har stilt digitalt billedmateriale til rådighet.

sign.

Anders Hveem Malum

Trondheim, 8.6.2005

Sammendrag

Denne masteroppgaven beskriver metoder for automatisert måling av mange punkt på 3-dimensjonale overflater. Hovedfokus er på anvendelse innen nærfotogrammetri. Bakgrunnsteori for generelle bildematchingsteknikker presenteres. Dette inkluderer en detaljert beskrivelse av ulike interesseoperatorer for objektbasert matching.

To programmer har blitt utviklet. Det ene inkluderer implementasjon av to ulike interesseoperatorer og en enkel matching av punktobjekter basert på krysskorrelasjon. Det andre programmet beregner modellkoordinater for punktobjekter dersom orienteringsparameterene til de to bildene er kjent. Det gis eksempler på mulig videre arbeid med modellkoordinater, inkludert triangulering av overflaten. En tredje interesseoperator, SIFT-deskriptoren, har også blitt testet. Implementasjonen av SIFT inkluderer matching av punktobjektene ved hjelp av minste kvadraters metode.

Til slutt i rapporten gis konklusjoner i forhold til hvilke matchingsalgoritmer og interesseoperatorer som er best egnet for automatisk flategenerering. Det gis også en oversikt over mulig videre arbeid med programvare og teori. All kildekode og alle testbilder er tilgjengelig på CD-ROM.

Abstract

This thesis describes methods for automated surface measurement in close-range photogrammetry. The theory of image matching fundamentals is given, including a detailed description of various interest operators for feature based matching.

Two programs have been developed. The first one includes implementation of two interest operators and a simple cross correlation technique for matching features. The second program computes model coordinates for extracted features if orientation parameters for the two images is known. A third interest operator called the SIFT-descriptor has also been tested. The implementation of SIFT includes least squares matching of the features. Examples of use includes triangulation of a surface based on computed model coordinates.

Finally, conclusions are given regarding the suitability of different matching algorithms for automatic surface measurement. Possible improvements of the implemented software are also described. All source code and sample pictures are available on CD.

Innhold

1	Innledning	1
1.1	Bakgrunn	1
1.2	Formål	2
1.3	Rapportens oppbygning	2
2	Ulike matchingsmetoder - bakgrunnsteori	3
2.1	Introduksjon	3
2.2	Arealbasert matching	4
2.2.1	Arealbasert matching med krysskorrelasjon	4
2.2.2	Arealbasert matching med minste kvadraters metode	5
2.2.3	Avansert arealbasert matching	8
2.2.4	Arealbasert matching i objektrommet	9
2.3	Objektbasert matching	9
2.3.1	Objektbasert matching med minste kvadraters metode	10
2.3.2	Objektbasert matching med dynamisk programmering	11
2.3.3	Objektbasert matching med klynger (clustering)	12
2.4	Relasjonsbasert matching	13
2.5	Hierarkisk bildematching	13
2.6	Flerstasjonsopptak med konvergent geometri	14
2.7	Valg av matchingsmetode	16
3	Interesseoperatorer	17
3.1	Historikk	17
3.2	Generelle krav til interesseoperatorer	19
3.2.1	Valg av distinkte punkter	19
3.3	Förstners interesseoperator	20
3.4	Harris' interesseoperator	21
3.5	Lowes SIFT-deskriptor	24

3.5.1	Deteksjon av målestokksekstrema	25
3.5.2	Deteksjon av lokale ekstrema	26
3.5.3	Utvalg av nøkkelpunktkandidater	26
3.5.4	Den lokale bildedeskriptoren	27
3.6	Tidligere evaluering av ulike interesseoperatorer	28
3.6.1	Test av interesseoperatorer - Schmid et al.	28
3.6.2	Test av interesseoperatorer - Parks og Gravel	30
4	Praktisk arbeid, eksperimenter og resultater	33
4.1	Implementasjon av interesseoperatorer	34
4.1.1	Implementasjon av Förstneroperatoren	34
4.1.2	Implementasjon av Harrisoperatoren	36
4.2	Praktisk testing av Harris og Förstners interesseoperatorer	38
4.3	Praktisk testing av SIFT-deskriptoren	40
4.4	Implementasjon av matchingsprosessen	47
4.5	Implementasjon av modellkoordinatberegning	48
4.5.1	Kolinearitetslikningene	48
4.5.2	Cramers metode	49
4.6	Modellkoordinatberegning	50
5	Drøfting	52
5.1	Interesseoperatorene	52
5.2	Matching av punkt	53
6	Konklusjoner og videre arbeid	54
6.1	Mulig fremtidig arbeid	55
A	Matematiske definisjoner	59

Figurer

2.1	Prinsippet for Arealbasert matching	4
2.2	Flerstasjonsopptak med konvergent geometri - 4 kamera	14
3.1	4-punkters nabolag	18
3.2	8-punkters nabolag	22
3.3	Diskret gradientberegning	23
3.4	Sift-deskriptoren	28
4.1	Flytskjema program	35
4.2	Testbilde	38
4.3	Gradient g_{xx}	38
4.4	Gradient g_{yy}	38
4.5	Gradient g_{xy}	38
4.6	Oppdagede punkter: Harrisdetektoren	39
4.7	Oppdagede punkter: Förstnerdetektoren	39
4.8	Oppdagede punkter levegg: Harrisdetektoren	39
4.9	Oppdagede punkter levegg: Förstnerdetektoren	39
4.10	Oppdagede punkter lurfald 1: Harrisdetektoren	40
4.11	Oppdagede punkter lurfald 2: Harrisdetektoren	40
4.12	Bildepyramide SIFT	41
4.13	Bildepyramide SIFT, Difference of Gaussian	42
4.14	SIFT-kandidater 1	42
4.15	SIFT-kandidater 2	42
4.16	Endelige SIFT-punkt	43
4.17	SIFT-punkt med størrelse og orientering	43
4.18	Matchingsbilder tømmervegg	44
4.19	SIFT-punkter tømmervegg	44
4.20	Matchede bilder	45

4.21	Levegg konvergente opptak	46
4.22	Lerfald konvergente opptak	47
4.23	Triangulering av modellkoordinater	50

Kapittel 1

Innledning

1.1 Bakgrunn

Nærfotogrammetri har i mange år blitt benyttet blant annet til dokumentasjon av kulturminner. På tross av dette har fagfeltet allikevel ikke blitt rutinemessig benyttet innenfor kulturminnedokumentasjon. Den manglende aksepten av nærfotogrammetriske metoder har flere årsaker. En av disse er at det har vist seg vanskelig å finne frem til metoder som har ønsket detaljeringsgrad og nøyaktighet samtidig som de er raske og rimelige å benytte. Denne oppgaven omhandler metoder for automatisert måling av mange punkt på tredimensjonale flater.

Manglende automatisering har vært et av hovedankepunktene mot nærfotogrammetriske metoder til slik dokumentasjon. De automatiserte metodene som eksisterer har hatt problemer med å ivareta og dokumentere ønsket nøyaktighet på de genererte modellene. Det er allikevel viktig å merke seg at matchingsteknikker er mye brukt i andre deler av fotogrammetrien. Her er noen eksempler på bruksområder, hentet fra Manual of Photogrammetry [20]:

- Indre orientering, relativ orientering og absolutt orientering.
- Generering av digitale høydemodeller fra flybilder og satellittbilder.
- Veikartlegging

1.2 Formål

Masteroppgaven tar sikte på å gi en beskrivelse og vurdering av algoritmer og metoder som kan egne seg for slik automatisk måling av mange punkt på tredimensjonale flater. Det vil bli gitt en historisk oversikt over utviklingen innenfor fagfeltet. Oppgaven inneholder også en oversikt over eksisterende matchingsalgoritmer, og en vurdering av hvilke algoritmer som er best egnet til å løse beskrevet problemstilling.

På bakgrunn av denne vurderingen er det gjort praktiske forsøk med noen av algoritmene, implementert i programmeringsspråket `c++`. Deretter er det gjort tester på ulikt bildemateriale, både enkeltbilder, stereobilder og konvergente flerstasjonsopptak.

Detaljnivået i den historiske gjennomgangen er begrenset. Det er imidlertid henvist til aktuelle kilder, slik at den interesserte leser kan finne tilgjengelig bakgrunnsmateriale. Det blir gitt en grundig teoretisk beskrivelse av implementerte algoritmer og aktuelle testresultater. Rapporten inneholder også en beskrivelse av programmet som er laget. Kildekode samt kjørbare programfiler for windowsplattformen er tilgjengelig på CD-ROM.

1.3 Rapportens oppbygning

Rapportens hoveddel starter med å gi en oversikt og vurdering av de ulike hovedkategoriene matchingsalgoritmer som finnes. Dette gjøres i kapittel 2. En metode velges ut, og denne beskrives i detalj i kapittel 3. Kapittel 4 inneholder en beskrivelse av utført praktisk arbeid, eksperimenter og resultater. Til slutt drøftes resultatene i kapittel 5 før konklusjon gis til slutt i kapittel 6. Dette kapitlet inneholder også en vurdering av hva slags arbeid som kan være aktuelt i en videreføring av oppgaven.

Kapittel 2

Ulike matchingsmetoder - bakgrunnsteori

2.1 Introduksjon

Dette kapitlet gir en oversikt over eksisterende metoder for bildematching. Manual of Photogrammetry [20] definerer matching som det å finne samsvarende deler blant flere teoretiske beskrivelser av et virkelig objekt. Et bilde er et eksempel på en slik beskrivelse. Det overordnede målet med matching er i følge samme kilde automatisert innsamling av meningsfull informasjon fra bilder eller sensordata. Christian Heipke [13] har en liknende definisjon, han sier at matching kan defineres som det å finne korrespondanse mellom ulike datasett. Matching benyttes ikke bare i forhold til bilder, men også i kart, objektmodeller og GIS-data.

Matching i digitale bilder løses ved hjelp av algoritmer (sett av regler for å løse et problem). I følge Heipke [13] ble de første forsøkene knyttet til bildematching gjort sent på 1950-tallet. Han refererer til Hobrough [14], som i 1959 gjorde forsøk med stereomatching. Hobrough brukte imidlertid analoge bilder og prosedyrer. Tradisjonelt har matchingsmetoder blitt delt i tre ulike kategorier. Kategoriene er listet opp nedenfor. De engelske begrepene er også nevnt her, siden disse er mye benyttet også på norsk. I resten av denne rapporten kommer imidlertid de norske begrepene til å bli benyttet.

- Arealbasert (area-based) matching. Her benyttes gråvinduer som matchingsprimitive.

- Objektbasert (feature-based) matching. Her benyttes objekter (for eksempel punkt og linjer) funnet i bildene som matchingsprimitiver.
- Relasjonsbasert (relational) matching.

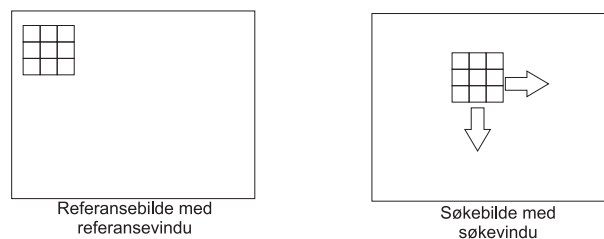
I de påfølgende kapitlene gis en nærmere beskrivelse av disse tre kategoriene. Manual of Photogrammetry [20] er hovedkilde for den informasjonen som gis, der andre kilder er benyttet er dette angitt spesielt.

2.2 Arealbasert matching

Arealbaserte matchingsalgoritmer søker etter områder med samme gråverdimønstre i to eller flere bilder, og benytter denne informasjonen til å finne samsvarende områder. Det finnes to hovedmetoder blant disse algoritmene. Den ene er basert på krysskorrelasjon, den andre bygger på minste kvadraters metode.

2.2.1 Arealbasert matching med krysskorrelasjon

Krysskorrelasjon er en måte å måle samsvar mellom to ulike signaler eller verdier. I dette tilfellet måles korrelasjon mellom to områder i to ulike bilder. Arealbaserte metoder med krysskorrelasjon som grunnlag har en del begrensninger, blant annet er de begrenset til å matche mellom to bilder av gangen. Masken holdes i ro i ett av bildene, dette kalles referansevinduet eller kjernen. I det andre bildet flyttes masken over bildet, denne masken kalles søkevinduet. Figur 2.2.1 viser dette.



Figur 2.1: Prinsippet for Arealbasert matching

I hver posisjon masken plasseres blir krysskorrelasjonskoeffisienten beregnet:

$$\rho(t_r, t_c) = \frac{\sum_{r=1}^{r=R} \sum_{c=1}^{c=C} (g_1(r, c) - \mu_1)(g_2(r + t_r, c + t_c) - \mu_2)}{\sqrt{\sum_{r=1}^{r=R} \sum_{c=1}^{c=C} (g_1(r, c) - \mu_1)^2 \sum_{r=1}^{r=R} \sum_{c=1}^{c=C} (g_2(r + t_r, c + t_c) - \mu_2)^2}} \quad (2.1)$$

Krysskorrelasjonskoeffisienten baserer seg på at kontrasten er den samme i begge bildene. I formelen er $\rho(t_r, t_c)$ korrelasjonen mellom kjernen og søkevinduet når søkevinduet er flyttet t_r rader og t_c kolonner. Kjernen kalles $g_1(r, c)$ og denne har R rader og C kolonner. Søkevinduet er da $g_2(r + t_r, c + t_c)$. Verdiene for forskyvning t_r og t_c vil alltid være et helt antall kolonner og rader.

De fleste arealbaserte matchingsalgoritmer forsøker å ta hensyn til problemer med ulik kontrast i bildene. Dette gjøres ved å normalisere krysskorrelasjonskoeffisienten. Dermed får man en korrelasjonskoeffisient som har verdi mellom -1 og 1. Koeffisienten i likning 2.1 er normalisert. Det er også verdt å merke seg at størrelsen på patchene som benyttes under matchingen har stor innflytelse på hvor gode resultatene blir. Søkevinduets størrelse bør gjenspeile usikkerheten i verdiene t_r og t_c , og det bør være et visst samsvar mellom kjernens og søkevinduets størrelse. De behøver imidlertid ikke være like store. Dersom de to vinduene ikke er like store, må dette tas hensyn til ved beregning av krysskorrelasjonskoeffisienten. Det er også mulig å gi pikslene i vinduene ulik vekt.

2.2.2 Arealbasert matching med minste kvadraters metode

Bruk av minste kvadraters metode(MKM) er en annen måte å gjøre arealbasert matching på. Krysskorrelasjonsmatching er enkel å implementere, men den har mange svakheter som arealbasert MKM-matching ikke har. De viktigste fordelene ved å benytte minste kvadraters metode er:

- Mulighet for å innføre andre geometriske transformasjoner enn translasjon mellom bildene. Se kapittel 2.2.3 for nærmere beskrivelse.
- Mulighet for subpikselberegning av punktenes plassering.
- Statistiske data om antatt feilfordeling er enkelt tilgjengelig.

Den største ulempen med arealbasert MKM-matching er at det stilles større krav til de initielle verdiene for t_r og t_c .

For å beskrive prinsippene bak MKM-matching benytter Manual of Photogrammetry [20] seg først av et eksempel med matching langs epipolare linjer (1-dimensjonal matching). Disse linjene tilsvarer rader i bildene dersom stereoopptak er benyttet. Samme notasjon som i tidligere beskrivelse av krysskorrelasjonsmatching benyttes her.

1-dimensjonal MKM-matching

Dersom de to bildene ikke har noen radiometriske forskjeller seg imellom, forventes det at gråverdiene til samhørende piksler er lik. Dette kan uttrykkes matematisk som en forventningsverdi:

$$E\{g_1(c_1) - g_2(c_2)\} = 0 \quad (2.2)$$

Her er verdien $c_2 = c_1 + t_c$. Som allerede nevnt må man ha tilgjengelig en rimelig god tilnæringsverdi for t_c . Dermed kan det beregnes en antatt plassering for alle piksler i søkevinduet ut fra plasseringen av kjernen. Startverdien t_c^0 brukes i en iterativ løsning av formelen

$$E\{g_1(c_1) - g_2(c_2^0)\} = \frac{\partial g_2(c_2^0)}{\partial t_c} \Delta t_c \quad (2.3)$$

Δt_c er her en liten forandring i verdien t_c^0 . Denne vil som regel ikke være et heltall, og ved å bruke lineær interpolasjon (se appendix A) for beregning av søkevinduet gråverdier oppnås nøyaktighet på subpikselnivå. Når ønsket antall iterasjoner er gjennomført kan et estimat for nøyaktigheten til de observerte gråverdiforskjellene uttrykkes som følger:

$$\hat{\sigma}_0^2 = \frac{\sum_{c_1=0}^{C_1-1} \Delta g(c_1)^2}{C_1} \quad (2.4)$$

Dette uttrykket beskriver følgende operasjon: summer kvadratet av gråverdiforskjellen mellom pikslene i vinduet fra hvert av bildene. Divider deretter på antall piksler C_1 (lengden av søkevinduet). Dette uttrykket for antatt nøyaktighet kan også benyttes for å angi variansen til den estimerte translasjonen t_c :

$$\hat{\sigma}_{t_c}^2 = \hat{\sigma}_0^2 \frac{1}{\sum_{c_1=0}^{C_1-1} (g_2, c(c_1^2))} \quad (2.5)$$

Når det gjelder nøyaktighetsmål i MKM-matching skisserer Förstner [8] fire faktorer:

1. Bildestøy. Dette er en faktor som sjelden medfører store forskjeller mellom bilder, den ligger typisk i området 1-2 gråtoneverdier.

2. Kontrast. Lav kontrast gjør det vanskelig å finne gode og entydige områder for matchingen. Dette nøyaktighetsmålet uttrykkes ved summen av gråverdiene i likning 2.5
3. Signalstørrelse (amplitude). Dersom signalamplituden er stor vil antall piksler med sterke gradienter øke. Signalet i et digitalt bilde er gråverdifunksjonen. Dersom et stort antall piksler har sterke gradienter (se appendix A) i flere retninger vil standardavviket til den estimerte translasjonen mellom vinduene minke.
4. Modellkorrekthet. Det gjøres en del antakelser i matchingsprosessen, der antatt translasjon fra kildebildet til søkebildet er en av disse. Dersom disse parameterene er feilaktige vil resultatene heller ikke være til å stole på.

Av disse nøyaktighetsmålene er det kun modellkorrekthet som ikke er mulig å estimere med utgangspunkt i likning 2.5.

2-dimensjonal MKM-matching

Arealbasert matching med MKM ville ikke være særlig nyttig hvis det det bare kunne benyttes i 1-dimensjonale tilfeller. Prinsippene er imidlertid de samme når vi utvider til 2D. Eksempelet som skisseres i MOP [20] tar utgangspunkt i at transformasjonen kan modelleres som en translasjon, altså en forflytning langs en linje i bildeplanet. Det vil si at det antas en forskyvning i radretning Δt_r og kolonneretning Δt_c . Som i det 1-dimensjonale tilfellet er resultatets kvalitet avhengig av en god startverdi for søkevinduet's plassering. Uttrykket for forventningsverdien utvidet til to dimensjoner blir dermed:

$$E\{\Delta g(r_1, c_1)\} = g_{2,r}(r_2^0, c_2^0)\Delta t_r + g_{2,c}(r_2^0, c_2^0)\Delta t_c \quad (2.6)$$

Verdiene r_2^0 og c_2^0 er startverdiene for translasjon i rad og kolonne-retning. Disse er gitt ved følgende uttrykk:

$$\begin{aligned} r_2^0 &= r_1 + t_r^0 \\ c_2^0 &= c_1 + t_c^0 \end{aligned} \quad (2.7)$$

Minste kvadraters metode er en iterativ prosess, der de estimerte verdiene for Δt_r og Δt_c forbedres slik at $\Delta g(r_1, c_1)$ minimeres. Når denne er minimert har vi funnet den plasseringen i søkebildet som samsvarer mest med referansen. For å få et mål på matchingens

kvalitet benyttes kovariansmatrisen:

$$\begin{pmatrix} \hat{\sigma}_{t_r}^2 & \hat{\sigma}_{t_r t_c}^2 \\ \hat{\sigma}_{t_r t_c}^2 & \hat{\sigma}_{t_c}^2 \end{pmatrix} = \hat{\sigma}_0^2 \begin{pmatrix} \sum_{r_1=0}^{R_1-1} \sum_{c_1=0}^{C_1-1} g_{2,r}^2 & \sum_{r_1=0}^{R_1-1} \sum_{c_1=0}^{C_1-1} g_{2,r} g_{2,c} \\ \sum_{r_1=0}^{R_1-1} \sum_{c_1=0}^{C_1-1} g_{2,r} g_{2,c} & \sum_{r_1=0}^{R_1-1} \sum_{c_1=0}^{C_1-1} g_{2,c}^2 \end{pmatrix}^{-1} \quad (2.8)$$

Verdien for $\hat{\sigma}_0^2$ beregnes på samme måte som i likning 2.4, men er selvfølgelig også utvidet til 2 dimensjoner:

$$\hat{\sigma}_0^2 = \frac{\sum_{r_1=0}^{R_1-1} \sum_{c_1=0}^{C_1-1} \Delta g(r_1, c_1)^2}{R_1 C_1} \quad (2.9)$$

$R_1 C_1$ er størrelsen på søkevinduet i antall piksler. Dersom en maske på 5x5 piksler benyttes får dermed $R_1 C_1$ verdien 25.

2.2.3 Avansert arealbasert matching

Så langt har det blitt fokusert på grunnprinsippene for arealbasert matching. En del forenklinger er gjort, og da særlig med tanke på transformasjonsmodellen mellom de to bildene som skal matches. Det er til en viss grad mulig å modellere andre typer transformasjoner enn translasjon ved arealbasert matching. Særlig er dette tilfelle for algoritmer som benytter minste kvadraters metode. En kort oversikt over noen mulige utvidelser til de allerede nevnte metodene kommer her:

- Forandringer i perspektiv kan modelleres som en affin transformasjon (se appendix A for forklaring). Dermed inkluderes målestokksforandringer, orientering og skjærtransformasjonen i matchingen.
- Radiometriske forskjeller mellom bildene kan også modelleres matematisk. Dette er forskjeller i kontrast og gråtoneintensitet. Radiometriske forskjeller kan for eksempel oppstå dersom det er ulik belysning ved fotografering.

Den affine transformasjonen som er gitt ved seks parametere utvides med to parametere b_1 og b_2 dersom det tas hensyn til radiometriske forskjeller. Dersom det ikke er tatt hensyn til radiometriske forandringer kan restfeilen bli stor i minste kvadraters metode. Uttrykket for forventningsverdien $E\{\Delta g(r_1, c_1)\}$ blir nå:

$$\begin{aligned} E\{\Delta g(r_1, c_1)\} = & r_1 g_{2,r} \Delta a_{rr} + c_1 g_{2,r} \Delta a_{rc} + r_1 g_{2,c} \Delta a_{cr} + c_1 g_{2,c} \Delta a_{cc} \\ & + g_{2,r} \Delta r_t + g_{2,c} \Delta t_c - g_1 \Delta b_1 - \Delta b_2 \end{aligned} \quad (2.10)$$

Manual of Photogrammetry nevner at denne utvidelsen av affin transformasjon har ført til konvergensproblemer i minste kvadraters metode i noen tilfeller. Det foreslås en utjevning av gråverdiene i vinduene som undersøkes for å løse dette problemet.

2.2.4 Arealbasert matching i objektrommet

En mulighet som bør nevnes spesielt er arealbasert matching i objektrommet. I forbindelse med 1-dimensjonal MKM-matching ble epipolare linjer nevnt. Disse kan brukes for å innføre en begrensning på translasjonene t_r og t_c . Dersom det i tillegg er tilgjengelig mer enn to bilder samt noen objektpunkt med kjente koordinater, kan disse objektpunktene benyttes for å sikre korrekthet i matchingsprosessen. Ved å bruke flere bilder i den arealbaserte matchingen oppnås høy redundans, noe som også med på å bedre nøyaktigheten på resultatene. Arealbasert matching har blitt benyttet med stort hell i generering av digitale høydemodeller fra ortofoto. En overflatebeskrivelse er ikke lett å danne med utgangspunkt i kun to bilder, siden beregnede høydeverdier vil være meget usikre. Noen eksempler finnes i Wrobel [31] og Grün [10].

Dette kapitlet har gitt en innføring i to forskjellige former for arealbasert matching, og mulige forbedringer av disse metodene. Arealbasert matching har vist seg å ha en del svakheter, og det har derfor blitt foreslått en annen type matchingsalgoritmer. Objektbasert, eller feature based matching er emnet i neste kapittel.

2.3 Objektbasert matching

Objektbaserte matchingsalgoritmer har en rekke fordeler i forhold til de arealbaserte. De er mer robust med tanke på bildestøy og radiometriske forandringer. I tillegg er de mindre beregningstunge, noe som gir kortere kjøretid. Selv med rask maskinvare er beregningstiden essensiell når store bildefiler skal behandles. Manual of Photogrammetry [20] er hovedkilde for informasjonen også i dette kapitlet.

En objektbasert matchingsalgoritme kan deles i tre trinn:

1. Objektekstraksjon. En interesseoperator benyttes for å finne distinkte punkter, kanter eller objekter i bildene som skal matches mot hverandre. Her finnes det en rekke metoder. Tabell 3.6.2 lister opp noen av disse, og gir en vurdering av deres ytelse. En nærmere undersøkelse av tre ulike operatører gis i kapittel 3. Interesseoperatorene arbeider på et bilde av gangen, og det er ikke nødvendig med forhåndskunnskap om bildets orientering.

2. Parvis matching. Objektene som oppdages i trinn 1 sammenliknes, og det lages en liste over punktpar som kan være match. Kun de parkombinasjonene som ikke under noen omstendighet kan være samsvarende forkastes.
3. Bestemmelse av endelige treff. Fra listen med mulige punktpar i trinn 2 dannes en endelig liste med samsvarende punkt. På samme tid beregnes den geometriske transformasjonen mellom bildene. Transformasjonen bør minimum være affin (se definisjon i appendix A), i noen tilfeller kan det også være nødvendig med mer komplisert modellering av transformasjonen.

Objektbasert matching finnes i en rekke ulike varianter. I tillegg til de typene som beskrives her er det også mulig å benytte krysskorrelasjon på de oppdagede punktene for å matche. Krysskorrelasjon er beskrevet i kapittel 2.2.1, og prinsippene er de samme ved objektbasert matching. Beregning av krysskorrelasjonskoeffisienten blir da bare gjort på punktene som er merket som interessante av interesseoperatoren. Dette vil kun fungere på bilder som er tilnærmet stereoopptak.

2.3.1 Objektbasert matching med minste kvadraters metode

Minste kvadraters metode er benyttet i mange implementasjoner av objektbasert matching. For en generell beskrivelse av robust MKM-matching i to eller flere bilder henvises til kapittel 8 i [1]. Her gis en oversikt over beregningsgangen.

Etter at objekteteksjon er utført i bildene, skal som allerede nevnt en liste over mulige punktpar dannes. Krysskorrelasjon kan benyttes til dette. Krysskorrelasjonskoeffisienten (likning 2.1) beregnes som regel med en vindusstørrelse på 5x5 eller 7x7 piksler. Dette kan imidlertid justeres avhengig av bildenes oppløsning og scenen som skal matches. Dersom krysskorrelasjonskoeffisienten overstiger en grenseverdi er punktparet en mulig match.

Punktparlisten danner så grunnlag for det endelige trinnet i MKM-matchingen. Transformasjonsparameterene mellom bildene estimeres. For en affin transformasjon resulterer dette i to observasjoner med seks ukjente parametere for hvert punktpar. Den matematiske modellen for en slik transformasjon er som følger:

$$E \left\{ \begin{pmatrix} r_2 \\ c_2 \end{pmatrix} \right\} = \begin{pmatrix} a_{rr} & a_{rc} \\ a_{cr} & a_{cc} \end{pmatrix} \begin{pmatrix} r_1 \\ c_1 \end{pmatrix} + \begin{pmatrix} t_r \\ t_c \end{pmatrix}$$

Dette uttrykket settes opp på formen Ax der A inneholder kjente verdier og x de ukjente

transformasjonsparameterene:

$$E \left\{ \begin{pmatrix} r_2 \\ c_2 \end{pmatrix} \right\} = \begin{pmatrix} r_1 & c_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & r_1 & c_1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{rr} \\ a_{rc} \\ a_{cr} \\ a_{cc} \\ t_r \\ t_c \end{pmatrix} \quad (2.11)$$

Under forutsetning av at den initelle listen med mulige punktpar inneholder 3 eller flere mulige kombinasjoner er det mulig å løse dette systemet. Vektene i utjevningen beregnes først med utgangspunkt i krysskorrelasjonskoeffisienten til hver observasjon. Minste kvadraters metode gir residualene til hver enkelt observasjon, og disse benyttes til å forkaste punktpar som med stor sannsynlighet ikke er samsvarende punkt. Etter hver iterasjon beregnes nye vektorer for gjenværende observasjoner, og vekten for observasjoner som antas å være grovfeil settes til 0. Metoden itererer til restfeilene for gjenværende punkt er under en gitt grenseverdi.

Den største fordelene med objektbasert mkm-matching fremfor arealbasert er at gråverdiene til området som skal matches kun benyttes i beregning av krysskorrelasjonskoeffisienten. Det er også lavere krav til initelle verdier for de affine transformasjonsparameterene. Förstner antyder i [9] følgende krav til de initelle verdiene:

- Translasjon: Initell verdi kan avvike med 30 % av størrelsen på vinduet som matches.
- Rotasjon: Initell verdi kan avvike med 20 grader i forhold til det korrekte.
- Skalering: Initiell verdi kan avvike med 30 % av det korrekte.

I kapittel 4.3 gjengis testresultater fra matching av SIFT-punkter. Denne matchingen benytter seg av objektbasert minste kvadraters metode.

2.3.2 Objektbasert matching med dynamisk programmering

Dynamisk programmering er en gammel teknikk, og ble beskrevet allerede i 1957 av Bellmann [5]. Det er en teknikk for å løse problemer som kan deles opp i trinn med flere mulige utfall. Teknikken gir ingen transformasjonsparametere mellom bildene, kun en liste med samsvarende objekter. Den baserer seg derfor på at den epipolare geometrien til bildeparet som skal matches er kjent på forhånd. Matching med dynamisk programmering kan dermed ikke benyttes i tilfeller der transformasjonen mellom bildene er

ukjent. Det er også en begrensning at objektene må ligge i samme rekkefølge langs de epipolare linjene i begge bilder. I tillegg vil en feil i de kjente orienteringsparameterene (kjent epipolar geometri) medføre store feil.

Det er ikke gjort noen forutsetning om kjent epipolar geometri for det praktiske arbeidet i forbindelse med masteroppgaven. Derfor er ikke dynamisk programmering aktuelt å benytte. Det er imidlertid en metode som bør tas i betraktning dersom problemstillingen er matching i bilder med kjent geometri. Det finnes også måter å benytte dynamisk programmering i arealbaserte teknikker, en av de nyere beskrivelsene er gitt av Baillard og Dissard[2].

2.3.3 Objektbasert matching med klynger (clustering)

Clustering er en robust teknikk som kan løse mange forskjellige problemer der parametere skal estimeres. Dette innebærer også estimering av transformasjonsparametere mellom to sett av bilder.

På samme måte som i de allerede nevnte teknikkene bygges det opp en liste med mulige samsvarende objektpar. Denne listen er basert på statistiske likhetsmål. Den skiller seg fra minste kvadraters metode fordi den velger ut flere små subsett av objektpar som matcher hverandre. Disse settene resulterer igjen i flere mulige estimater for transformasjonsparameterene. Det vil ikke være redundans i noen av disse subsettene.

Dersom et subsett kun består av korrekte objektpar, vil de estimerte transformasjonsparameterene også være riktige. I en matchingsprosess vil det være mange slike parkombinasjoner som er korrekte, og for disse vil transformasjonsparameterene også være noenlunde like. For alle andre subsett (de som inneholder feilaktige treff) vil transformasjonsparameterene anta tilfeldige verdier. Formulert på en annen måte kan det sies at metoden gjenkjenner klynger (clustre) med subsett der transformasjonsparameterene antar tilnærmet like verdier. De gjennomsnittlige verdiene for disse subsettene brukes dermed som et estimat for transformasjonen mellom de to bildene.

Størrelsen på subsettene er av avgjørende betydning for om metoden lykkes eller ikke. Hvis subsettene er for store vil de inneholde inkorrekte objektpar sammen med de korrekte. Er subsettene for små vil de riktige parkombinasjonene distribueres over flere klynger.

2.4 Relasjonsbasert matching

Den tredje gruppen matchingsalgoritmer kan sees som en utvidelse av de objektbaserte metodene. Relasjonsbasert matching bygger på likhet i topologiske relasjoner mellom objekter. Topologiske relasjoner er uavhengige av perspektiviske transformasjoner. Dermed er relasjonsbaserte metoder kraftige, og de kan fungere i nokså generelle tilfeller. Problemet med denne typen algoritmer er avhengighet av god segmentering i bildene. Segmentering er en prosess der objekter i bildet skilles ut fra bakgrunnen. I tillegg er relasjonsbaserte algoritmer kostbare. Det medfører lang kjøretid, og i tillegg er kjøretiden ofte uforutsigbar. Innenfor kulturminnedokumentasjon er objektene som skal modelleres ofte komplekse, og de relasjonsbaserte algoritmenes søkestrøm vil få en tilsvarende kompleksitet. På grunn av nevnte begrensninger vil relasjonsbasert matching bli viet lite oppmerksomhet videre i denne rapporten.

2.5 Hierarkisk bildematching

Alle metodene som er beskrevet så langt baserer seg på initielle verdier for transformasjonsparameterene som skal estimeres. Hva så hvis slike verdier ikke er tilgjengelig? Hierarkisk bildematching vil være nyttig i slike tilfeller. Slik matching baserer seg på å resample bildet i flere mindre versjoner, for eksempel til halvparten og en fjerdedel av det originale bildets størrelse. Dersom det originale bildet er veldig stort kan også ytterligere reduksjon være nødvendig. En slik skalering av bildet i flere nivå kalles ofte for en bildepyramide. Manual of Photogrammetry [20] beskriver en mye benyttet variant av slik matching for overflaterestruksjon fra to bilder:

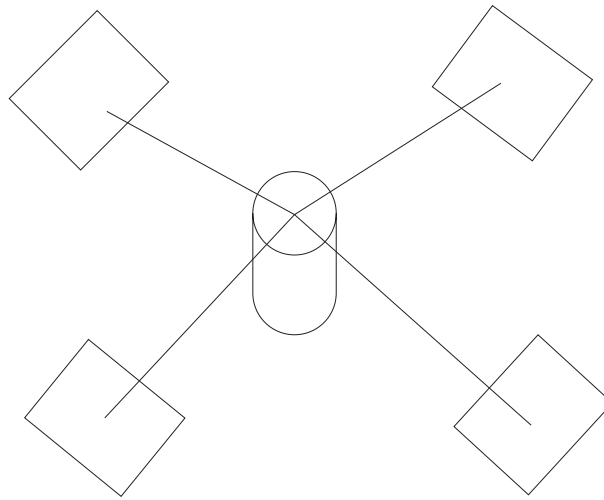
- Det bygges en bildepyramide for hvert av de to bildene.
- Det øverste bildet i venstre pyramide deles i flere små deler, der hver del representerer en liten og delvis glatt del av overflaten.
- Hver del av dette bildet matches mot korresponderede søkevindu i høyre bilde. Dette gir transformasjonsparametere for hver av de små delene i høyre og venstre bilde.
- Hver del av toppbildet i pyramiden splittes i fire nye deler på neste nivå. Her er bildet dobbelt så stort som i øverste nivå. Dermed er hver del i bildet på nivået under like stor som en del fra bildet på øverste nivå. Tilnærmede transformasjonsparametere for hver del beregnes av resultatene på øverste nivå i pyramiden. Til slutt på dette nivået matches venstre og høyre bilde mot hverandre.

- Prosedyren gjentar seg for alle nivå i pyramiden. Når transformasjonen for nederste nivå er beregnet, har man muligheten til å beregne en detaljert beskrivelse av objektets overflate.

En rekke eksisterende programmer benytter seg av bildepyramider, for eksempel Match T i InduSURF. Dette systemet utfører beregning av tredimensjonale flater, automatisk orientering og stereomatching. I det praktiske arbeidet med denne oppgaven har en interesseoperator som benytter seg av bildepyramider blitt testet, mer informasjon om SIFT-operatoren i kapittel 3.5.

2.6 Flerstasjonsopptak med konvergent geometri

I nærfotogrammetrien er det vanlig å benytte mer enn to bilder, og en opptaksmodell med konvergent geometri. Her gjennomgås det matematiske grunnlaget for beregning av terrengkoordinater fra bildekoordinater i slike situasjoner. Kilde for dette kapitlet er kapittel 2.4 i Atkinsons bok om nærfotogrammetri [1]. En mulig opptakssituasjon er vist i figur 2.6.



Figur 2.2: Flerstasjonsopptak med konvergent geometri - 4 kamera

Kolinearitetslikningene er utgangspunktet i en slik beregning. I sin opprinnelige form er

disse likningene:

$$x = x_0 - c \frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (2.12)$$

$$y = y_0 - c \frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (2.13)$$

Her er x og y målte bildekoordinater, x_0 og y_0 er origo for bildekoordinatsystemet, c er kamerakonstanten, r_{ij} koeffisientene i rotasjonsmatrisen, X_0, Y_0, Z_0 er kameraets plassering i terrengekoordinatsystemet og X, Y, Z er punktets terrengekoordinater. Dersom disse likningene utvides for tilfeller med mer enn to bilder antar de følgende form:

$$x_{ij} = x_{0j} - c_j \frac{r_{j,11}(X_i - X_{0j}) + r_{j,12}(Y_i - Y_{0j}) + r_{j,13}(Z_i - Z_{0j})}{r_{j,31}(X_i - X_{0j}) + r_{j,32}(Y_i - Y_{0j}) + r_{j,33}(Z_i - Z_{0j})} \quad (2.14)$$

$$y_{ij} = y_{0j} - c_j \frac{r_{j,21}(X_i - X_{0j}) + r_{j,22}(Y_i - Y_{0j}) + r_{j,23}(Z_i - Z_{0j})}{r_{j,31}(X_i - X_{0j}) + r_{j,32}(Y_i - Y_{0j}) + r_{j,33}(Z_i - Z_{0j})} \quad (2.15)$$

Atkinson nevner også at dersom ulike målestokk innad i bildet antas kan to ulike kamerakonstanter benyttes for å korrigere for dette. I tillegg kan ytterligere parametere inkluderes. Disse vil typisk være korreksjon for radiell og tangentiell linsefortegning. Dersom dette gjøres må høyre side av kolinearitetslikningene utvides. Legg merke til at nå er kamerakoordinatsystemets origo også satt på høyre side av likningssystemet.

$$\begin{aligned} & x_{ij} - x_{0j} + (x_{ij} - x_{0j})d_{ij}^{-1}(K_{1j}d_{ij}^{-3} + K_{2j}d_{ij}^{-5} + K_{3j}d_{ij}^{-7}) \\ & + P_{1j}(d_{ij} * -2 + 2(x_{ij} - x_{0j})^2) + 2P_{2j}(x_{ij} - x_{0j})(y_{ij} - y_{0j}) \end{aligned} \quad (2.16)$$

$$\begin{aligned} & y_{ij} - y_{0j} + (y_{ij} - y_{0j})d_{ij}^{-1}(K_{1j}d_{ij}^{-3} + K_{2j}d_{ij}^{-5} + K_{3j}d_{ij}^{-7}) \\ & + P_{2j}(d_{ij} * -2 + 2(y_{ij} - y_{0j})^2) + 2P_{1j}(x_{ij} - x_{0j})(y_{ij} - y_{0j}) \end{aligned}$$

K og P er parametere for henholdsvis radiell og tangentiell fortegnning. Avhengig av hvilken type linse som er benyttet vil ikke alltid alle disse parameterene være tilgjengelige fra kamerakalibreringen. Det er også mulig å inkludere andre korreksjoner i likningene.

Det totale systemet kan settes opp på matriseform for løsning ved hjelp av minste kvadraters metode:

$$\mathbf{F}(\mathbf{A}, \mathbf{x}, \mathbf{b}) = \mathbf{0} \quad (2.17)$$

Notasjonen her er standard, \mathbf{A} er kjente koeffisienter, \mathbf{x} er de ukjente og \mathbf{b} er målinger.

2.7 Valg av matchingsmetode

Det har nå blitt gitt en beskrivelse av ulike eksisterende matchingsmetoder. Den praktiske delen av denne oppgaven har gått ut på å velge ut en av disse metodene for implementasjon og eksperimenter på et eller flere sett av digitale bilder.

De arealbaserte metodene har flere store begrensninger. ABM basert på korrelasjon kan ikke utvides til å inkludere mer enn to bilder. Dersom det matches i objektrommet kan imidlertid ABM med minste kvadraters metode utvides til flere bilder. Dette forutsetter kjente objektkoordinater for en del punkt, noe som ikke er tilfelle for testbildene. I tillegg er arealbaserte metoder som kjent følsomme for støy, radiometriske forandringer. Det er også mer komplisert å modellere annen geometrisk transformasjon enn translasjon mellom bildene.

Matching basert på dynamisk programmering er ikke aktuelt, siden det krever høy grad av kjennskap til bildenes geometriske begrensninger. Relasjonsbasert matching har sine styrker, men uforutsigbar beregningstid og høy kompleksitet gjør disse uaktuelle.

Den hovedtypen av matchingsalgoritmer som utmerker seg er dermed de objektbaserte. Valget er gjort spesielt med tanke på kjøretid og robusthet i forhold til geometriske og radiometriske endringer mellom bildene. I neste kapittel vil det derfor bli gitt en beskrivelse og sammenlikning av ulike interesseoperatorer. Det falt naturlig å teste mer enn en operator, for å etablere et sammenlikningsgrunnlag. Referanser til eksisterende publikasjoner og programmer innenfor feltet vil gis der dette er tilgjengelig. I det praktiske arbeidet er det gjort implementasjon og testing av to slike interesseoperatorer, og en fritt tilgjengelig MatLabimplementasjon av en tredje operator er testet. Mer om dette i kapittel 4.

Kapittel 3

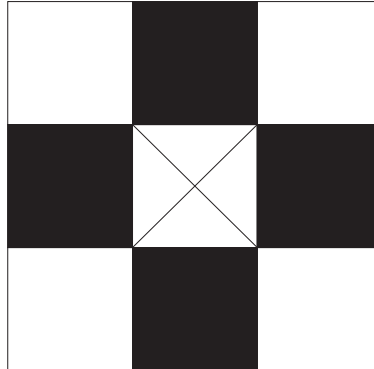
Interesseoperatorer

En interesseoperator er det første trinnet i all objektbasert matching. En slik operator leter etter og markerer punkt i bildene som kan være gode kandidater til matching. Forskning på ulike interesseoperatorer har vært gjort i over 30 år, og det er på sin plass å gi en liten oversikt over utviklingen. Poenget er ikke å gi en inngående beskrivelse av alle nevnte operatorer og bruksområder, for spesielt interesserte henvises til referert litteratur. Denne historiske oversikten i stor grad er hentet fra David Lowes artikkel om SIFT-deskriptoren [19].

3.1 Historikk

Et av de første publiserte arbeidene relatert til objektbasert matching ble gjort av Marsha Jo Hannah [11]. Hun beskrev en objektbasert matchingsmetode basert på krysskorrelasjon (se kapittel 2.2.1). Denne interesseoperatoren var en av inspirasjonskildene til W. Förstner senere [9]. I 1981 publiserte Moravec en artikkel om sin operator [22]. Han arbeidet med stereomatching, og utviklet i den forbindelse en interesseoperator som finner punkt der gråverdiene har største minimum varians i fire retninger som vist i figur 3.1. Moravec selv videreutviklet denne til å inkludere et større antall piksler rundt punktet som undersøkes.

Noe senere på 1980-tallet arbeidet Harris og Stephens med en interesseoperator. De tok da utgangspunkt i Moravecs arbeid. Harrisoperatoren beskrives nærmere i kapittel 3.4. Denne operatoren har hatt stor suksess, og den er fortsatt benyttet til en rekke ulike



Figur 3.1: 4-punkters nabolag

matchingsoppgaver. Den har også vært gjenstand for videreutvikling de senere årene. Et par eksempler er verdt å nevne:

- Zhang et al. [32] fra 1995. Her beskrives muligheten for å matche Harrispunkter ved hjelp av et korrelasjonsvindu. Dette vinduet plasseres over hvert punkt som oppdages av Harrisdetektoren. Dataene fra korrelasjonsvinduet blir så brukt for å finne sannsynlige treff. Grovfeil fjernes ved å benytte kjent kunnskap om avbildningens geometriske betingelser. Det kreves med andre ord noe forkunnskap om bildene som skal matches.
- Schmid og Mohr [25] viste i 1997 at lokal objektmatching kunne utvides til å matche et objekt mot en stor database av bilder. Harrisdetektoren brukes til å velge interessepunkter, men en rotasjonsuavhengig deskriptor for bilderegionen ble benyttet i stedet for korrelasjonsvindu. Dermed kunne objekter matches uavhengig av bildenes orientering. I tillegg viste de at flertydige treff allikevel kunne gi god gjenkjenning selv ved hindringer og støy ved å identifisere konsistente klynger av matchede objekter. Objektbasert matching med klynger (clustering) er kort beskrevet i kapittel 2.3.3.

Noen år tidligere enn Harris publiserte Förstner sin artikkel “A Feature Based Correspondence Algorithm for Image Matching” [9]. Han tar utgangspunkt i en algoritme beskrevet av Barnard og Thompson [3], og som tidligere nevnt i arbeidet til Hannah. Også Förstners deskriptor har vært mye brukt, og en rekke videreutviklinger/implementasjoner finnes. Kapittel 3.3 beskriver denne interesseoperatoren.

Både Harris’ og Förstners operatører er følsomme for målestokksforandringer, og er dermed ikke gode ved matching av bilder med ulik størrelse eller målestokk. I 1999 publis-

erte Lowe [18] en artikkel der han utvider eksisterende metoder for å oppnå uavhengighet av målestokk. Dermed blir det mulig å matche bilder av ulik størrelse og bilder tatt i ulik avstand til scenen som skal modelleres. Testing har vist at denne metoden gir mer distinkte objekter, og den er mindre følsom for forstyrrelser i bildet. Slike forstyrrelser kan være støy og belyningsforandringer.

Det å finne objekter som er uavhengige av affine transformasjoner har vært et av de største problemene innenfor matching. Lowe henviser til en rekke forskjellige metoder som forsøker å oppnå dette. De viktigste (foruten Lowes eget arbeid) er Baumberg [4], Tuytelaars and Van Gool [28], Mikolajczyk and Schmid [21] og Schaffalitzky og Zissermann [24]. Ingen av disse har lyktes fullt ut, men forskningen kommer stadig nærmere en løsning på problemet. I kapittel 3.5 gis en grundig beskrivelse av Lowes SIFT-deskriptor fra 2004. Dette er den nyeste utviklingen innenfor objektbasert matching og interesseoperatorer.

3.2 Generelle krav til interesseoperatorer

3.2.1 Valg av distinkte punkter

Förstner angir i [9] en rekke krav som må oppfylles for ekstraherte objekter. Kort gjengitt er disse:

1. Distinkte. Punktene må være klart forskjellige fra nabopunkter.
2. Uforanderlige. Punktene må forventes å holde seg konstante hva gjelder geometriske og radiometriske egenskaper. Det vil si at det må være mulig å finne igjen punktene i andre bilder uavhengig av opptakspunkt, og uavhengig av belyningsforandringer.
3. Stabile. Punktene må kunne finnes igjen i alle andre bilder av scenen.
4. Sjeldne. Punktene må være mulige å skille i et globalt perspektiv. Særlig gjelder dette i bilder der man har mønstre som gjentas. Dersom bildenes orientering eller andre begrensninger er kjente på forhånd kan det benyttes for å skille punkter globalt.
5. Tolkbare. Utvalgte punkter må være tolkbare på en eller annen måte. Förstner nevner her kanter, hjørner og flekker som eksempler.

Punktene må i følge Förstner representeres i form av sine pikselkoordinater og en beskrivelse. Denne beskrivelsen kan for eksempel være den lokale gråverdifunksjonen. Avanserte

metoder som Lowes SIFT-operator er i mindre grad avhengig av piksellokasjonen til punktet.

Det er i følge Cornelia Schmid [26] vanlig å dele interesseoperatorene inn i tre kategorier:

- Konturbaserte operatører
- Intensitetsbaserte operatører
- Operatører basert på en parametrisk modell

De konturbaserte operatorene er som navnet tilsier basert på plassering av kandidatpunkter for matching der konturene i bildene er særpreget. Kanter er eksempler på slike konturer.

Intensitetsbaserte metoder baserer seg på gråverdifunksjonen. Eksempler på slike er Harrisoperatoren (og dermed også Moravec) og Förstneroperatoren. Hovedtyngden av disse operatorene benytter seg av korrelasjon.

Operatører basert på parametriske modeller inkluderer SIFT-operatoren. Denne beskrives i kapittel 3.5. Disse parameterene beskriver punktobjektets geometriske egenskaper. En slik modell tillater også å benytte kjente begrensninger i bildene og kunnskap om objektets geometri.

3.3 Förstners interesseoperator

Förstners metode for deteksjon og ekstraksjon av kjennetegn [9] ble som nevnt publisert første gang i 1984, og den har blitt stående som et av de store fremskrittene innenfor objektbasert matching. I artikkelen gis ikke bare en beskrivelse av interesseoperatoren, men også en komplett fremgangsmåte for hele matchingsprosessen. Dette kapittelet fokuserer imidlertid på interesseoperatoren.

Styrken til Förstners operator ligger først og fremst i presis plassering av senteret til runde objekter. Den er også rask. Förstner har i tillegg til punktobjektens plassering inkludert en beskrivelse av gråverdifunksjonen til utvalgsvinduet rundt punktet. Han fremhever også at senterpikselen i matchingsvinduet sjelden er det ideelle punktet for matching, siden dette punktet ikke er geometrisk konstant. I stedet for senterpunktet benyttes det vektete gravitasjonscenteret. Dette beregnes for hver piksel i bildet som undersøkes.

Et vindu (ofte 3x3, 5x5 eller 7x7 pixler) beveges over bildet. I hvert vindu beregnes det

vektede gravitasjonssenteret $z = (x, y)^T$. Først settes følgende enkle feillikninger opp for hvert av punktene i vinduet:

$$\begin{aligned}x_i + v_i &= \hat{x} \\ y_i + v_i &= \hat{y}\end{aligned}\tag{3.1}$$

v_i er initielle verdier. Deretter beregnes korrelasjonsmatrisen M :

$$M_i = \begin{bmatrix} g_{xi}^2 & g_{xi} * g_{yi} \\ g_{xi} * g_{yi} & g_{yi}^2 \end{bmatrix}\tag{3.2}$$

Merk at denne kalles W i den originale artikkelen, M benyttes for å være konsistent i forhold til beskrivelse av andre operatører senere i rapporten. Følgende normallikninger kan nå settes opp:

$$\begin{aligned}N_{11} * \hat{x} + N_{12} * \hat{y} &= \sum(v_i * g_x^2) + \sum(v_i * g_x * g_y) \\ N_{21} * \hat{x} + N_{22} * \hat{y} &= \sum(v_i * g_x * g_y) + \sum(v_i * g_y^2)\end{aligned}\tag{3.3}$$

Elementene i normalmatrisen N er som følger:

$$\begin{aligned}N_{11} &= \sum(g_x^2) \\ N_{12} &= N_{21} = \sum(g_x * g_y) \\ N_{22} &= \sum(g_y^2)\end{aligned}\tag{3.4}$$

Løsningen $\hat{z} = (\sum M_i)^{-1} \sum(M_i z_i)$ av normallikningene vil gi det vektete gravitasjonssenteret. Her er $z_i = (x_i, y_i)^T$. Deretter bestemmes “styrken” til hvert kandidatpunkt. Dette kan gjøres ved å kombinere leddene i korrelasjonsmatrisen på følgende måte:

$$punktstyrke = \frac{g_x^2 * g_y^2 - (g_x g_y)^2}{g_x^2 + g_y^2}\tag{3.5}$$

Dette er bare en av flere mulige måter å vurdere om et punkt skal regnes som distinkt. Förstner nevner i sin artikkel at man kan også vurdere hvert punkt opp mot referansemodeller, og kategorisere de som generelt punkt, skjæringspunkt eller liknende på bakgrunn av dette.

3.4 Harris’ interesseoperator

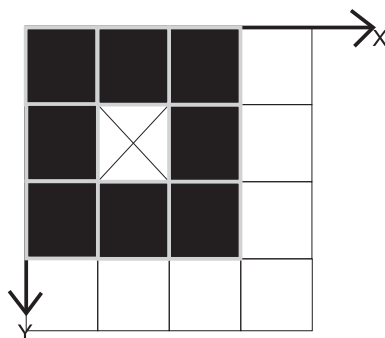
Harrisoperatoren kom noen år senere enn Förstners, og den har en litt annen bakgrunn. Artikkelen den ble publisert i heter “A Combined Corner and Edge Detector.” Som tittelen tilsier omhandles ikke bare enkeltpunkt, men også kanter. Forfatterene ønsket å

oppnå en forståelse av naturlige scener. Selv om kantdeteksjon er interessant, faller det utenfor temaet i denne rapporten. I det følgende blir det derfor fokusert på den delen av Harris & Stephens artikkel som omhandler punkt.

I kapittel 3.1 ble det nevnt at Harrisoperatoren er basert på Moravecs tidligere arbeid. Moravecs objekt-detektor er forholdsvis enkel. Den beregner forandring E i intensitet for en forflytning (u, v) i bildet:

$$E_{u,v}(x, y) = \sum_{u,v} w_{u,v} (I_{x+u, y+v} - I_{x, y})^2 \quad (3.6)$$

Her er w vinduet som flyttes over bildet og I er gråtoneintensitet i aktuell piksel. x og y er koordinatene til senterpunktet i masken, altså punktet som undersøkes. Forflytningene som tas i betraktning er $((1,0), (1,1), (0,1), (-1,1), (-1,0), (-1,-1), (0,-1), (1,-1))$. Se figur 3.2. Det bør bemerkes at disse forflytningene er feilaktig beskrevet i Harris' originale artikkel. Der er kun fire nabopikslers tatt med.

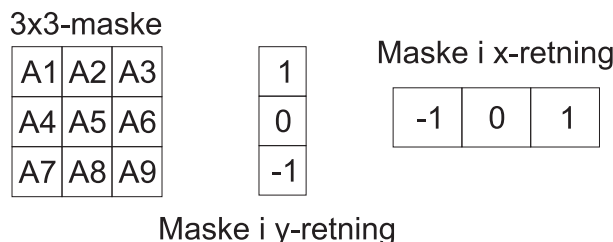


Figur 3.2: 8-punkters nabolag

Moravecs detektor ser altså etter lokale maksima for $\min\{E\}$. Det spesifiseres en grenseverdi for hva som regnes for å være et slikt maksima. Ytelsen til Moravecs operator er ikke særlig god, og Harris gjør en rekke endringer i det matematiske grunnlaget for operatoren. For det første slår han fast at responsen til operatoren er anisotrop (egenskapene varierer avhengig av måleretning), siden det bare blir gjort beregninger for de diskrete pikslene som er markert sort i figur 3.2. Dette løser Harris ved å gjøre en analytisk utvidelse rundt senterpikselen. For et tilfeldig shift (u,v) kan intensitetsvariasjonen uttrykkes som:

$$E_{u,v}(x, y) = \sum_i \left(u \frac{\partial I_i}{\partial x} + v \frac{\partial I_i}{\partial y} \right)^2 \quad (3.7)$$

Indeks i refererer til alle forskyvninger som gjøres. De første gradientene tilnærmes med diskret konvolusjon i henholdsvis x og y -retning som i figur 3.3. Maskene legges over senterpikselen i 3×3 -vinduet som undersøkes.



Figur 3.3: Diskret gradientberegning

Dette tilsvarer å ta den partiellderiverte med hensyn på henholdsvis x og y :

$$X = I \otimes (-1, 0, 1) = (I_{A6} - I_{A4}) = \frac{\partial I}{\partial x} \quad (3.8)$$

$$Y = I \otimes (-1, 0, 1)^T = (I_{A2} - I_{A8}) = \frac{\partial I}{\partial y} \quad (3.9)$$

\otimes er konvolusjon, se appendix A for forklaring. Siden masken som benyttes i Moravec-operatoren er kvadratisk og binær (se figur 3.2) er responsen veldig følsom for støy. Intuitivt forstår man at det bør legges mer vekt på de målingene som ligger nærmest senterpikselen i vinduet. Dette problemet løser Harris ved å bruke et glatt, sirkulært vindu (representert ved en Gaussisk funksjon) i stedet for de diskrete shiftene:

$$w_{u,v} = e^{-(u^2+v^2)/2\sigma^2} \quad (3.10)$$

Dermed utvides uttrykket for E ytterligere, og blir som følger:

$$E_{u,v}(x, y) = \sum_i w_i \left(u \frac{\partial I_i}{\partial x} + v \frac{\partial I_i}{\partial y} \right)^2 \quad (3.11)$$

Her er w_i verdien for uttrykket i likning 3.10 i posisjon i .

I tillegg er det et problem at Moravecs operator reagerer altfor lett på kanter. Dette kommer av at den kun ser etter lokale maksima i $\min\{E\}$. Husk at E uttrykker forandringen i gråtoneintensitet ved en translasjon (x,y) . Harris velger å omformulere uttrykket for E slik at det utnytter variasjonen som oppstår når forskyvningens retning forandres:

$$E_{u,v}(x, y) = Ax^2 + 2Cxy + By^2 \quad (3.12)$$

Her er

$$\begin{aligned} A &= X^2 \otimes w \\ B &= Y^2 \otimes w \\ C &= (XY) \otimes w \end{aligned} \quad (3.13)$$

$$E_{u,v}(x, y) = (x, y) \begin{bmatrix} A & C \\ C & B \end{bmatrix} (x, y)^T \quad (3.14)$$

Den symmetriske matrisen i dette uttrykket kalles M, og tilsvarende autokorrelasjonsfunksjonen. Harrisoperatoren benytter seg av matrisens egenverdier (se appendix A, α og β). For kanter vil α være stor og β liten (ideelt sett 0). I et distinkt punkt vil begge verdier være store, og på en flate vil begge være små. Dette benyttes for å luke ut uønskede objekter langs kanter.

I tillegg til de ovennevnte metodene for å velge punktobjekt trengs et mål på responsen til de valgte punktene. Den endelige responsen til Harrisoperatoren i et punkt kan uttrykkes ved determinanten og sporet til matrisen M som følger:

$$R = \text{Det}(M) - k(\text{Tr}(M))^2 \quad (3.15)$$

Uttrykket k i formelen over er en konstant. Denne er vanligvis i størrelsesordenen 0.4-0.14. I Harris' originale implementasjon er k satt til 0.6. De interessante punktobjektene finner vi der verdien R har et lokalt maksima. Dersom R er liten vil punktet med stor sannsynlighet befinne seg på en flate, og kan lukes vekk fra videre undersøkelser. For nærmere informasjon om implementasjon av Harrisoperatoren, se kapittel 4.1.2.

Harrisoperatoren yter mye bedre enn den originale Moravecoperatoren. Det bør allikevel nevnes at også denne har sine begrensninger. For det første krever den mye regnekraft. Den er også følsom mot støy, selv om den ikke er like følsom som Moravecs. Den er anisotrop (det er en vanlig oppfatning at den ikke er det, men det er feil), og den er ustabil ved T-kryssoverganger i gråtoneverdiene.

3.5 Lowes SIFT-deskriptor

David G. Lowe beskriver i [19] en metode for å finne distinkte, konstante objekter i bilder. Disse objektene kan benyttes til å gjøre matching mellom bilder tatt fra forskjellige opptakspunkt. Objektene skal være uavhengige av bildemålestokk og rotasjon, og robuste i forhold til radiometriske forandringer og støy. Kompleksiteten i denne interesseoperatoren er meget stor i forhold til de to tidligere beskrevne, Harris og Förstner. Fordelen er at i motsetning til tidligere interesseoperatører er objektene mer distinkte,

og et enkelt objekt kan matches mot en stor database av objekter fra mange ulike bilder. Deskriptoren (representasjonen av objektene) er døpt SIFT, som står for Scale Invariant Feature Transform. På norsk: Målestokksuavhengig objekttransformasjon.

Søk etter punkt består av fire ulike trinn:

1. Deteksjon av målestokksekstrema. Det bygges en bildepyramide. Alle målestokker og piksellokasjoner gjennomføres. Dette er effektivt implementert ved hjelp av en funksjon som finner forskjeller i Gaussverdi. Dermed finner algoritmen potensielle interessepunkt som er uavhengige av målestokk og plassering.
2. Nøkkelpunktlokalisering. I hver plassering der et kandidatpunkt ble funnet i første trinn benyttes en detaljert modell for å finne punktets plassering og målestokk.
3. Tilordning av orientering. En eller flere orienteringer tilordnes hvert nøkkelpunkt fra trinn 2. Alle senere operasjoner gjøres på bildedata som er transformert relativt til oppgitt orientering, målestokk og plassering for hvert objekt. Dermed oppnås uavhengighet i forhold til disse transformasjonene.
4. Nøkkelpunktbeskrivelse. Lokale gradienter måles i valgt målestokk (målestokk angis som en verdi for σ) i området rundt hvert nøkkelpunkt. Disse transformeres til en representasjon som tillater relativt store lokale optiske forandringer og forandringer i belysning og kontrast.

I det følgende beskrives hvert enkelt trinn i detalj.

3.5.1 Deteksjon av målestokksekstrema

Målet med dette trinnet i algoritmen er å identifisere plasseringer og målestokker som kan tilordnes forskjellige bilder av samme objekt. Først bygges en bildepyramide opp, der bildet resamples både til dobbel størrelse og til mindre versjoner. Lowe henviser så til tidligere forskning gjort av Könderink [16] og Lindeberg [17]. Disse har funnet at den eneste mulige matematiske funksjonen som kan benyttes til dette er en Gaussisk funksjon.

Målestokksrommet til et bilde er definert som en funksjon $L(x, y, \sigma)$. Konvolusjon gjøres mellom bildet som skal undersøkes og følgende Gaussfunksjon:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

Altså finner man målestokksrommet til et bilde slik:

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y) \tag{3.16}$$

Nå er det etablert en definisjon på målestokksrommet til et bilde. Som nevnt i innledningen til dette kapitlet brukes dette i en beregning av Gaussiske forskjeller (differanser). Først beregnes $L(x, y, \sigma)$ for en rekke ulike målestokker (det vil si forskjellige verdier av σ). Deretter beregnes forskjellen mellom par av nærliggende målestokker som følger:

$$\begin{aligned} D(x, y, \sigma) &= ((G(x, y, k\sigma) - G(x, y, \sigma)) \otimes (x, y)) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (3.17)$$

Faktoren k er en konstant faktor som beskriver denne forskjellen. Kostnaden i kjøretid for beregningen er veldig liten, siden det eneste som faktisk gjøres er en enkel subtraksjon mellom to glattede bilder.

De konvolverte bildene $L(x, y, \sigma)$ grupperes i noe som kalles oktaver. En oktav er definert som en dobling av standardavviket σ . Når man har bestemt seg for hvor mange bilder som trengs i hver oktav, er faktoren k i uttrykket for $D(x, y, \sigma)$ også gitt. Den velges slik at antall bilder blir det samme i alle oktaver.

3.5.2 Deteksjon av lokale ekstrema

Punkter av interesse finnes på bakgrunn av liknende kriterier som i tidligere beskrevne interesseoperatorer. Forskjellen er at med SIFT brukes Gaussforskjellbildene $D(x, y, \sigma)$ i letingen etter maksima og minima. Metoden leter altså etter lokale topper eller bunner i flere bilder samtidig:

- Et vindu, som regel på 3x3 piksler, føres over bildet som undersøkes.
- Hver piksel sammenliknes med sine 8 naboer i samme bilde.
- I tillegg sammenliknes hver piksel med de 9 tilhørende pikslene i de to naborivåene i pyramiden.
- Dersom pikselen på bakgrunn av denne sammenlikningen har en lokal topp eller bunn i gråverdi, velges den som en kandidat til nøkkelpunkt.

Dette kan virke som en kostbar operasjon kjøretidsmessig. Det er imidlertid slik at etter noen få runder med sammenlikning vil de fleste punktene allerede være eliminert bort.

3.5.3 Utvalg av nøkkelpunktkandidater

For hvert punkt som merkes som en kandidat må det gjøres en rekke undersøkelser og beregninger i etterkant:

- Interpolasjon med nærliggende piksler gjøres for å bestemme en nøyaktig posisjon.
- Punkter fjernes dersom de har lav kontrast i forhold til sitt nabolag.
- Unødig respons langs kanter fjernes. Se også diskusjonen av Moravec versus Harris i kapittel 3.4 om dette. Det er et problem for de fleste interesseoperatorer at de velger for mange punkt langs en kant. Slike punkt er ustabile.
- Nøkkelpunktet tilordnes en orientering.

Valg av orientering

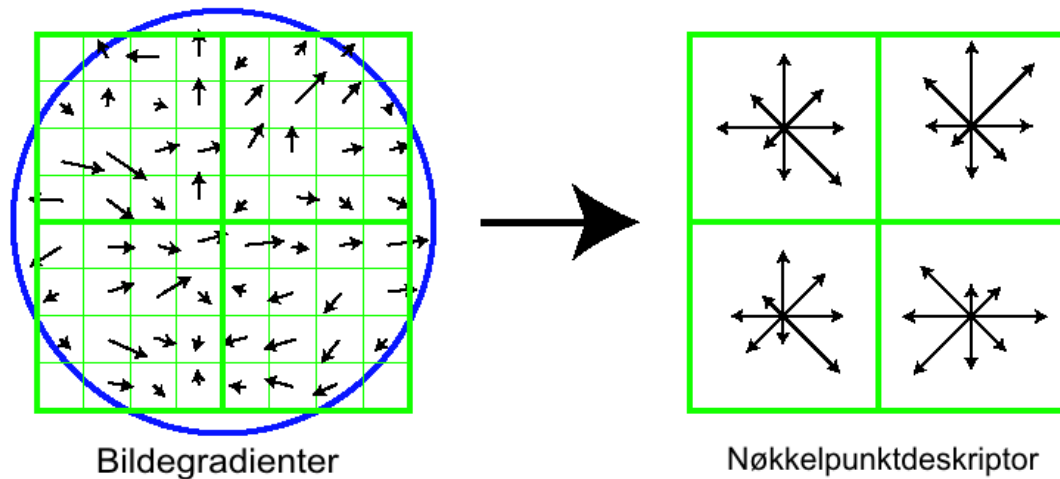
Dersom målet om rotasjonsuavhengighet for punktobjekter skal oppnås, må de tilordnes en orientering. Lowe gjør dette ved å beregne et såkalt gradienthistogram i området rundt nøkkelpunktet. Hver nabopiksels bidrag vektet med størrelsen på gradienten i den retningen, og med et Gaussisk vindu der σ er 1.5 ganger større enn målestokken til nøkkelpunktet.

Toppen i dette histogrammet defineres som en dominerende retning, og et eget nøkkelpunkt lages for denne retningen. Det samme gjøres for hver retning der størrelsen på gradienten er innenfor 80 % av toppens.

Deretter måles alle nøkkelpunktets egenskaper i forhold til nøkkelpunktets orientering. Dette gjør at punktobjektet ikke blir avhengig av bildets rotasjon.

3.5.4 Den lokale bildedeskriptoren

Nå har det blitt etablert en orientering for hver av nøkkelpunkt kandidatene i det aktuelle bildet. Da er det tid for å beregne selve deskriptoren. Dette gjøres i et område på 4×4 piksler rundt senterpunktet. Som i foregående beregning vektet hver piksels bidrag med gradientens størrelse og en Gausskjerne med σ 1.5 ganger større enn målestokkens. Histogrammene inneholder 8 “kasser” (bins) hver seg, og hver deskriptor inneholder et array med 4 histogrammer rundt nøkkelpunktet. Dermed er selve vektoren representert som en $4 \times 4 \times 8$ -vektor med til sammen 128 elementer. Vektoren inneholder nøkkelpunktets posisjon, målestokk(σ), orientering og informasjon om strukturen til området rundt nøkkelpunktet. Denne vektoren normaliseres for å forhindre problemer relatert til belysning og kontrast. Figur 3.4 er et visuelt eksempel på hvordan en SIFT-deskriptor fortoner seg. Figuren er hentet fra Lowes artikkel [19].



Figur 3.4: Sift-deskriptoren

Matching av nøkkelpunkt fra ulike bilder gjøres ved hjelp av minste kvadraters metode, og blir uavhengig av rotasjon fordi deskriptoren inneholder informasjon om orientering. Dermed kan området rundt nøkkelpunktene sammenliknes relativt til denne.

3.6 Tidligere evaluering av ulike interesseoperatorer

Som kapittel 3.1 viser er det utarbeidet en rekke andre interesseoperatorer til bruk i objektbasert matching. Det er også gjort arbeid relatert til å evaluere de ulike operatorene. De fleste av disse undersøkelsene har vært konsentrert om operatorene som finner kanter i bildene, men det finnes også noen få undersøkelser av punktoperatorene. I dette kapitlet gjengis resultater fra to slike undersøkelser.

3.6.1 Test av interesseoperatorer - Schmid et al.

En av de mer omfattende undersøkelsene ble gjort av Cordelia Schmid, Roger Mohr og Christian Bauckhage, og er beskrevet i [26]. De har testet 5 ulike interesseoperatorer mot følgende kriterier:

- Repeterbarhet. Dette er et mål på den geometriske stabiliteten til de ekstraherte punktene, og måles ved å sammenlikne objekter. Et objekt er repetert hvis et punkt

fra det første bildet også er oppdaget i bilde nummer to. Repeterbarhetsraten defineres av Schmid et al. som prosentandelen av det totale antall punkt som merkes i begge bilder.

- Informasjonsinnhold. Dette er en test på hvor distinkte punktene er. Entropien til disse deskriptorene måler informasjonsinnholdet til et sett med interessante punkter.

Kriteriene kan benyttes på en hvilken som helst scene, og baserer seg ikke på en bestemt objekttype eller tolkning.

Det bør nevnes at dette ikke er den eneste måten å teste brukbarheten til interesseoperatører. Tidligere arbeid nevnes også i [26]. Problemet med mange av disse evalueringsmetodene er at de er subjektive, altså avhengig av menneskelig tolkning.

Testede operatører

De testede interesseoperatørene er:

1. Harrisoperatoren (forbedret utgave).
2. Förstneroperatoren.
3. Cottieroperatoren.
4. Heitgeroperatoren.
5. Horaudoperatoren.

Med unntak av Harrisoperatoren har de benyttet forfatterens originale implementasjon av operatoren. I tilfellet med Harrisoperatoren har Schmid et al. forbedret denne ved å endre masken for beregning av den deriverte til autokorrelasjonsfunksjonen. I stedet for å benytte den originale $[-1 \ 0 \ 1]$ -masken, brukes den deriverte av en Gaussisk funksjon. Denne versjonen viste seg å gi bedre geometrisk stabilitet.

Følgende forsøk ble gjennomført for alle fem operatører:

- Bilderotasjon. Denne testen innebar rotasjon av kameraet rundt den optiske akse. Testen er et mål på geometrisk stabilitet for ekstraherte objekter.
- Målestokksforandring. Dette ble gjort ved å variere brennvidden til linsen.

- Belysningsforandring. Testen var todelt, operatorene ble testet både mot en uniform belysningsforandring og en endring av lyskilde (kompleks belysningsforandring).
- Endring av opptakspunkt. Kameraet flyttes rundt i en scene.
- Entropi. Entropi er et mål for uregelmessigheten til en variabel. Schmid et al. sier at entropien beskriver det gjennomsnittlige informasjonsinnholdet i objektene som er funnet.

Av testene faller de fire første inn under kategorien repeterbarhet, mens entropi er relatert til informasjonsinnholdet. Forfatterene begrunner valget av nettopp disse testene med at de kan relateres til et stort utvalg datasynoppgaver. I tillegg til bildematching inkluderer dette blant annet beregning av epipolar geometri.

Testresultater

Forsøkene viste at i alle tilfeller var den modifiserte versjonen av Harrisoperatoren bedre enn eller tilsvarende de andre operatorene. Dette gjaldt både for repeterbarhet og for informasjonsinnholdet. Forfatterene kommenterer også det faktum at i alle tester var interesseoperatorene bedre enn tilfeldig valgte punkt. Det er også verdt å merke seg at Förstners operator ligger som nummer tre eller fire i alle testene relatert til repeterbarhet, og at den yter dårligst når det gjelder informasjonsinnholdet i punktene som blir funnet. Det er bare tilfeldig valgte punkt som yter dårligere i så henseende.

3.6.2 Test av interesseoperatorer - Parks og Gravel

Donovan Parks og Jean-Philippe Gravel ved Faculty of Engineering, McGill University, har gjort en sammenlikning av den teoretiske ytelsen til en rekke interesseoperatorer. Her gjengis en tabell med disse resultatene. For mer informasjon om deres arbeide henvises til websiden deres [23].

Også denne artikkelen er såpass gammel at den ikke har med LOWEs SIFT-operator. Av de to andre operatorene som er valgt ut for testing ser vi at denne testen har merket Förstneroperatoren som god på alle testkriterier unntatt kjøretid. Harrisoperatoren går under navnet Plessey i tabellen. Denne har gitt noe dårligere resultater i testingen til Parks og Gravel. Dersom resultatene fra Schmidts testing sammenliknes med resultatene fra denne testen er det lett å se at de trekker noe ulike konklusjoner. Schmid har rangert Förstneroperatoren som den dårligste av de testede, mens Parks og Gravel mener den

Tabell 3.1: Interesseoperatorers ytelse

Operator	Deteksjonsrate	Lokalisering	Repeterbarhet	Støyrobust	Hastighet
Beaudet	Gjennomsnittlig	Gjennomsnittlig	Dårlig ved skalering, god ved affine transformasjoner	Dårlig	God
Moravec	Gjennomsnittlig	God	Gjennomsnittlig	Gjennomsnittlig	God
Kitchen og Rosenfeld	Gjennomsnittlig	Gjennomsnittlig	Gjennomsnittlig	Gjennomsnittlig	Dårlig
Forstner	God	God	Utmerket ved affine transformasjoner, gjennomsnittlig ved skalering	God	Dårlig
Plessey	God	God ved L-kryss, dårlig ellers	Utmerket for affine transformasjoner hvis isotropisk gradientberegning benyttes, gjennomsnittlig ved skalering	Gjennomsnittlig	Dårlig
Deriche	Gjennomsnittlig	God	God	Dårlig	God
Wang og Brady	God	God	God	Gjennomsnittlig	God
SUSAN	God	Dårlig ved uskarpe bilder, veldig god ellers	God ved skalering, dårlig ved affine transformasjoner	Utmerket	God
CSS	God	God	Utmerket	God	Avhengig av kantdetektoren som benyttes
Trajkovic og Hedley (4-punkters nabolag)	Dårlig	God	Gjennomsnittlig (ikke rotasjon-suavhengig)	Dårlig	Utmerket
Trajkovic og Hedley (8-punkters nabolag)	Gjennomsnittlig	God	Gjennomsnittlig (ikke rotasjon-suavhengig)	God	Utmerket
Zheng og Wang	God	God i L-kryss, gjennomsnittlig ellers	Utmerket ved affine transformasjoner, gjennomsnittlig ved skalering	Gjennomsnittlig	Gjennomsnittlig

er bedre enn Harrisoperatoren. Detaljer om implementasjonene Parks og Gravel har benyttet er ikke kjent, men det er grunn til å anta at de avviker noe fra Schmids. De sprikende testresultatene bekrefter det som generelt gjelder for interesseoperatorer: deres ytelse er alltid avhengig av bruksområdet.

For en sammenlikning mellom egne testresultater og disse tidligere testene henvises til kapittel 5.

Kapittel 4

Praktisk arbeid, eksperimenter og resultater

Det praktiske arbeidet har gått ut på testing av ulike interesseoperatorer og matchingsalgoritmer. Det har blitt utviklet to egne programmer i den forbindelse. Det ene programmet har funksjonalitet for Harris- og Förstners interesseoperatorer samt krysskorrelasjonsbasert matching av resultatene fra disse. Program nummer to beregner modelkoordinater fra bildekoordinater. Dette programmet er avhengig av tilgjengelige orienteringsdata for bildene som matches.

De egenutviklede programmene er kodet i programmeringsspråket `c++`. DevC++ fra Bloodshed software er benyttet under kodingen. Dette programmet finnes fritt tilgjengelig på internett [27], og inkluderer en `c++`-kompilator fra Minimalistic Gnu for Windows, MinGW [7].

DevC++ har basisfunksjonalitet for utvikling av programmer i `c++`. I tillegg har det et pakkehåndteringssystem som tilbyr nedlasting av en rekke utvidelser. Det er selvfølgelig også mulig å benytte alle andre `c++`-bibliotek. For basisfunksjonalitet knyttet til håndtering av bildefiler ble et bibliotek kalt VIGRA benyttet. VIGRA står for “Vision with generic algorithms,” og er laget for bruk i datasyn-applikasjoner. I følge bibliotekets hjemmeside [29] er VIGRA laget for å kunne integreres i alle `c++`-programmer. Algoritmene i VIGRA er maler som må tilpasses etter behov for å gi ønsket funksjonalitet. Den kanskje største fordelen med VIGRA er knyttet til kjøretid. Siden biblioteket består av et sett standard headerfiler som inkluderes etter behov, blir ytelsen den samme som for en tradisjonell, skreddersydd applikasjon. Det finnes også andre slike bibliotek fritt tilgjengelig, blant

annet leptonica [6].

Vigrabiblioteket må kompileres til et statisk bibliotek for å fungere. Dette kan gjøres med en hvilken som helst windows- eller unixkompilator. Som standard støtter VIGRA lesing og skriving av en rekke ulike bildeformater. Merk imidlertid at støtte for tif, jpg og png ikke er med som standard. Disse formatene støttes ved å laste ned libtiff, libjpeg og libpng fra Dev-C++. I tillegg må alle bibliotekene inkluderes under kompilering ved å sende kommandoene -lvigraimpex, -ltiff, -ljpeg og -lpng til kompilatoren. For ytterligere detaljer om implementasjon, kompilering og Vigras virkemåte henvises til kildekoden og nevnte hjemmesider.

4.1 Implementasjon av interesseoperatorer

Kildekoden til programmet finnes i filen Interesseoperator.cpp. Programmet er kommandolinjebasert, og jobber på 2 bilder av gangen. Brukeren velger først hvilken interesseoperator som skal benyttes, og angir fullstendig filnavn for bildefilene som skal undersøkes. For hvert bilde skrives en resultatfil til disk. Deretter gis brukeren mulighet til å velge om resultatene fra de to bildene skal matches mot hverandre. I så fall gjøres en enkel krysskorrelasjonsbasert matching av resultatene.

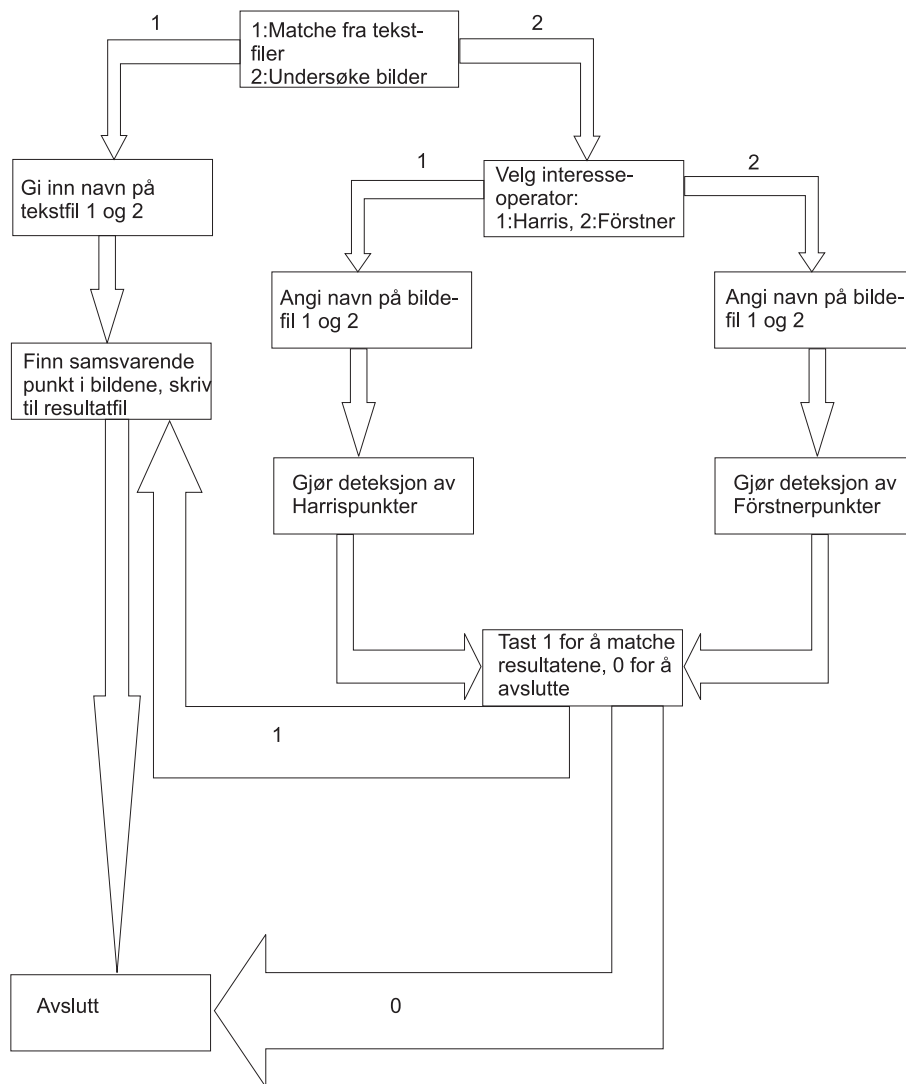
Alle resultater lagres i tabulatorseparerte tekstfiler. Programmet opplyser om navnene på disse filene underveis. Figur 4.1 viser et flytskjema for programmet.

Programmet har en del begrensninger. Det fungerer kun med gråtonebilder. Disse må være 8-bits (256 gråtoner), og det er en fordel om de er i et ukomprimert format. Eksempler på slike bildeformat er bmp, png og tiff. Siden matchingsdelen er basert på krysskorrelasjon, vil denne kun gi gode resultater for tilnærmede stereoopptak.

4.1.1 Implementasjon av Førstneroperatoren

I dette kapitlet gis en beskrivelse av implementasjonen som er gjennomført av Førstners interesseoperator.

Bildet som skal undersøkes lastes inn i programmet. Det opprettes 2 midlertidige bilder, et for mellomlagring av resultater, og et for lagring av endelig operatorrespons. Disse bildene har samme størrelse som det originale bildet. Deretter kalles metoden for selve interesseoperatoren. Denne oppretter tre nye midlertidige bilder, g_{xx} , g_{yy} og g_{xy} . Hvert av disse inneholder en beregning av produktet av bildets gradienter (se Appendix A)



Figur 4.1: Flytskjema program

som angitt av subskript. Hvert bilde er glattet med en Gausskjerne (Appendix A) der σ er satt til 1.0:

$$\text{Strukturtensor}(I) = \begin{pmatrix} G \otimes (I_x I_x) & G \otimes (I_x I_y) \\ G \otimes (I_x I_y) & G \otimes (I_y I_y) \end{pmatrix} \quad (4.1)$$

Her representerer G Gaussisk glatting, I_x og I_y er gradientene, $*$ er konvolusjon og $I_x I_x$ samt $I_y I_y$ er produktene av de gradientbildene. VIGRA kaller disse tre bildene for en strukturtensor. Det blir kun tre bilder siden matrisen er symmetrisk. Operasjonen er den samme som i likning 3.5. Strukturtensoren benyttes også i Harrisoperatoren, som beskrives i neste kapittel.

De tre midlertidige bildene med strukturtensoren må kombineres til ett, og det gjøres piksel for piksel ved å benytte følgende formel:

$$\text{forstnerrespons} = \frac{G_{xx}G_{yy} - G_{xy}G_{xy}}{G_{xx} + G_{yy}} \quad (4.2)$$

Det er dette uttrykket som gir den endelige responsen til selve Förstneroperatoren. Resultatet av denne operasjonen lagres til den første midlertidige filen som ble opprettet ved oppstart av programmet.

Nå gjenstår kun ett trinn i deteksjon av punktobjekter. Det midlertidige bildet med operatorens respons gjennomføres på jakt etter lokale maksima. En 5x5-maske benyttes i søket, og punkt som representerer lokale topper i bildets gråverdier markeres med hvitt i det endelige bildet. Alle andre piksler i bildet får svart markering. Hver gang en topp blir funnet, skrives pikselkoordinatene til en tekstfil i tillegg til at punktet markeres.

Funksjonen som finner lokale maksima er en standard funksjon i VIGRA. Denne måtte modifiseres noe, siden den opprinnelig benyttet en 3x3-maske. Dette viste seg å gi for mange kandidatpunkter for matchingen. Masken ble utvidet til 5x5 piksler.

Resultatene skrives til tekstfiler med prefix *forstner* + bildets filnavn + .txt. Hver linje i filen inneholder informasjon om ett punktobjekt. Den første kolonnen inneholder et løpenummer. Deretter kommer punktets x-koordinat, punktets y-koordinat, 25 kolonner med gråverdiene til 5x5-masken, en kolonne med summen av gråverdiene i masken og en kolonne med gjennomsnittlig gråverdi for masken. Denne informasjonen benyttes i matchingsprosessen senere.

4.1.2 Implementasjon av Harrisoperatoren

Interresseoperatorprogrammets implementasjon av Harrisoperatoren er helt standard, og følger Harris originale beskrivelse i [12]. Det første trinnet i beregningen er det samme som for Förstneroperatoren, beregning av strukturtensoren. Harris kaller denne 2x2-matrisen for M i sin artikkel. Bildene i strukturtensoren kombineres så på en annen måte enn i Förstneroperatoren:

$$R = \text{Det}(M) - k(\text{Tr}(M))^2 \quad (4.3)$$

R uttrykker den endelige responsen til operatoren før deteksjon av lokale maksima. Mer detaljert er denne operasjonen implementert som

$$\begin{aligned}
 R &= \text{Det}(M) - k(\text{Tr}(M))^2 \\
 &= AB - C^2 - k(A + B)^2 \\
 &= G_{xx}G_{yy} - G_{xy}^2 - k(G_{xx} + G_{yy})^2
 \end{aligned} \tag{4.4}$$

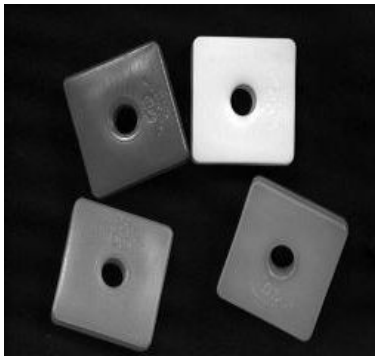
En punktvis beskrivelse av gangen i beregning av Harrisoperatoren er som følger:

- Programmet tar inn gråtonebildet. I programmet er hardkodet verdier for k og T (Treshold), samt Gaussisk varians.
- Det opprettes et midlertidig bilde for operatorrespons og et bilde for lagring av endelig resultat.
- For hver piksel(x,y) i bildet beregnes autokorrelasjonsmatrisen M . Dette gjøres også som i Förstneroperatoren, ved beregning av strukturtensor for bildet og lagring av denne i tre midlertidige bilder.
- De tre midlertidige bildene med gradientproduktene G_{xx} , G_{yy} og G_{xy} kombineres i henhold til likning 4.4, og resultatet lagres i det midlertidige bildet for operatorrespons.
- Punkter som ikke er lokale maksima fjernes fra bildet. Resultatet lagres til filen for endelig resultat.
- Treshold gjøres på det midlertidige bildet med operatorrespons for å fjerne ikke signifikante punkter.
- Resultatet av treshold kombineres med resultatet av lokale maksima, og skrives til filen med endelig resultat.
- Resultatbildet skrives til disk, sammen med en tekstfil inneholdende piksellokasjonene til punktobjektene, gråverdien til hver av pikslene i 5×5 -masken og gjennomsnittlig gråverdien i 5×5 -masken for hvert punktobjekt.

Verdien k er satt til 0.06, dette er den samme verdien som benyttes i Harris' originale implementasjon av operatoren. Strukturen på resultatfilene er den samme som for Förstneroperatoren (se kapittel 4.1.1), med det unntak at filnavnene har prefix harris.

4.2 Praktisk testing av Harris og Förstners interesseoperatorer

Harris- og Förstneroperatoren er testet på flere ulike bilder, her gjengis testbildet som er benyttet i figur 4.2. Bildet følger med Vigna-biblioteket. På grunn av sin enkelhet egner det seg godt til å illustrere virkningen til interesseoperatoren.



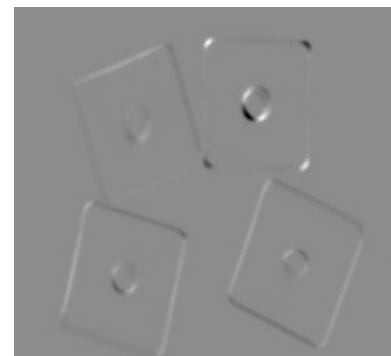
Figur 4.2: Testbilde



Figur 4.3: Gradient g_{xx}

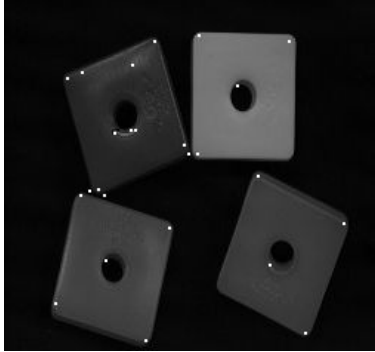


Figur 4.4: Gradient g_{yy}



Figur 4.5: Gradient g_{xy}

De tre gradientbildene g_{xx} , g_{xy} , g_{yy} er vist i figurene 4.3, 4.4 og 4.5. Normalt skrives ikke gradientbildene til disk, men de er tatt med her for å illustrere prosessen i operatorens punktsøk. I figur 4.6 er de punktene som ble markert av Harrisdetektoren lagt på toppen av det originale bildet. På samme måte vises i figur 4.7 hvordan den implementerte Förstneroperatoren yter på testbildet.

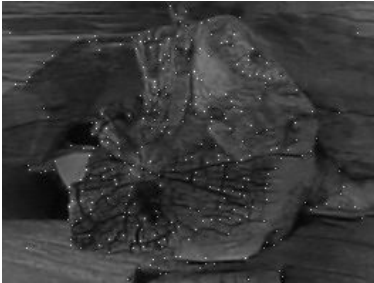


Figur 4.6: Oppdagede punkter: Harrisdetektoren

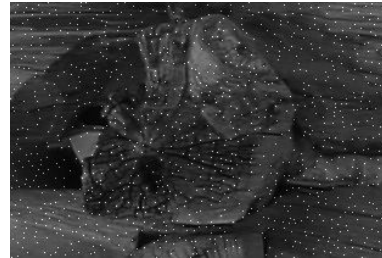


Figur 4.7: Oppdagede punkter: Förstnerdetektoren

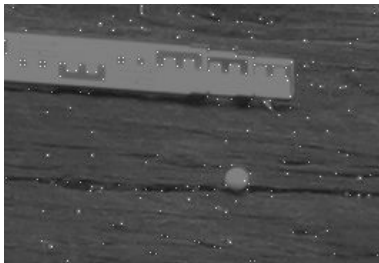
Når det gjelder mer kompliserte bilder, har både bilder fra helleristningsfeltet Lurfald og bilder av en levegg i tømmer blitt benyttet. Figur 4.8 viser resultater fra Harrisoperatoren på leveggen, mens figur 4.9 viser resultater fra Förstneroperatoren på samme bilde. Harrisdetektoren viste seg å være den mest stabile i alle testbildene. Samme område fra Lurfald-feltet markert med Harrispunkter i to ulike bilder er vist i figurene 4.10 og 4.11. Disse bildene er utsnitt. For ytterligere kommentarer til alle testresultatene, se kapittel 5.



Figur 4.8: Oppdagede punkter levegg: Harrisdetektoren



Figur 4.9: Oppdagede punkter levegg: Förstnerdetektoren



Figur 4.10: Oppdagede punkter lersfald 1: Harrisdetektoren



Figur 4.11: Oppdagede punkter lersfald 2: Harrisdetektoren

4.3 Praktisk testing av SIFT-deskriptoren

SIFT-deskriptoren er meget omfattende, og noen egen implementasjon av denne var ikke aktuelt innenfor denne oppgavens tidsrammer. Det finnes imidlertid en implementasjon tilgjengelig på internett, kodet i Matlab. Programmet er laget ved Universitetet i Torontos avdeling for datateknikk [15]. For spesielt interesserte er det mulig å laste ned denne implementasjonen herfra. For å installere programmet ble følgende fremgangsmåte benyttet:

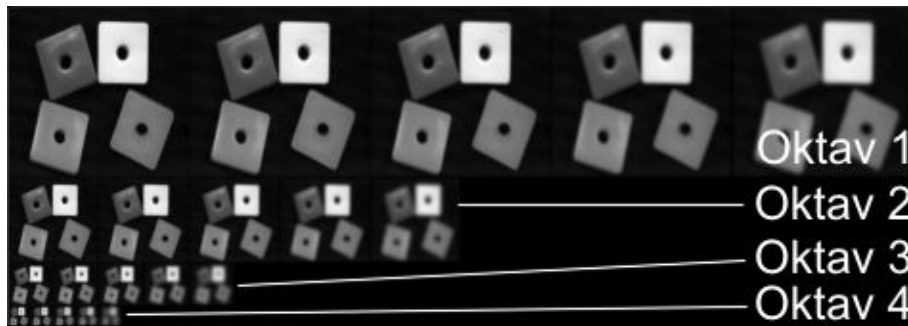
- Matlabillegget `matlabVisTools` ble lastet ned. Dette ligger i en komprimert zip-fil på [15]. Dette tillegget ble pakket ut til Matlabs toolbokskatalog.
- Deretter ble selve programmet `SIFT-tutorial` hentet ned og pakket ut. Siden all testing ble gjort på windowsplattformen var det nødvendig å installere winrar for å få pakket ut denne tar.gz-filen. Alle filene fra denne pakken ble lagt i katalogen (`matlabs toolbokskatalog/utvisToolbox/tutorials`). `UtvisToolbox` er en del av matlabillegget i foregående punkt.
- Matlab ble startet opp, og filen `startup.m` ble lastet inn fra toolbokskatalogen og kjørt. Dette programmet setter opp nødvendige miljøvariabler for `matlabVisTools`, og er essensielt for at `SIFT-tutorial` skal fungere.
- Til slutt ble filen (`matlabs toolbokskatalog/utvisToolbox/tutorials/tutorial.m`) lastet inn. Her ble koden endret noe, slik at riktige bildefiler ble benyttet. Deretter var det bare å kjøre programmet som en vanlig matlab m-fil.

Under den første testingen ble samme bilde som tidligere benyttet. De såkalte oktavene bygges opp. Husk at en oktav inneholder bildet i en viss størrelse, glattet med ulike

gausskjerne. Det originale bildet resamples i denne implementasjonen i 4 ulike utgaver, der bildets opprinnelige størrelse ligger i oktav 2. Den første oktaven inneholder bildet i dobbel størrelse, den tredje er bildet i halv størrelse og den fjerde oktaven inneholder bildet i en fjerdedel av opprinnelig størrelse. Det vil si at dersom det originale bildet har en størrelse på 400x400 piksler, er oktavene bygget med disse bildestørrelsene:

1. 800x800 piksler, σ varierer mellom 1.0 og 2.0.
2. 400x400 piksler, σ varierer mellom 2.0 og 4.0.
3. 200x200 piksler, σ varierer mellom 4.0 og 8.0.
4. 100x100 piksler, σ varierer mellom 8.0 og 16.0.

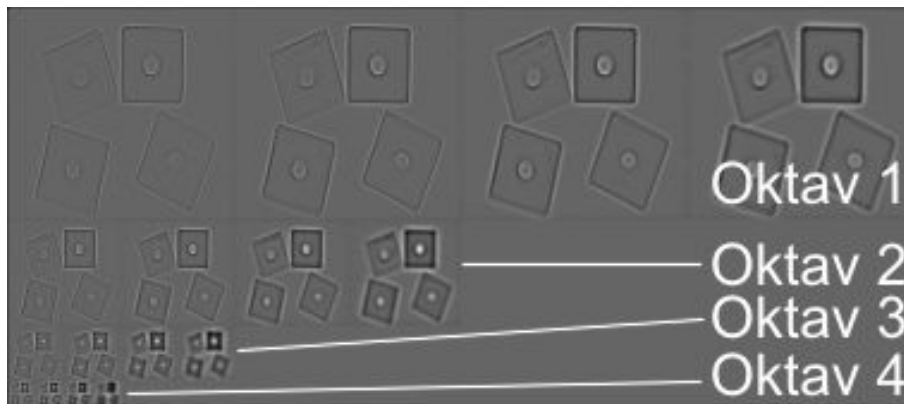
Innenfor hver oktav lages 4 utgaver av bildet i samme størrelse, glattet med gausskjerne der σ varierer. I figur 4.12 vises alle bildene som lages i hver oktav. Figuren er selvfølgelig kraftig forminskert, men illustrerer hva som skjer.



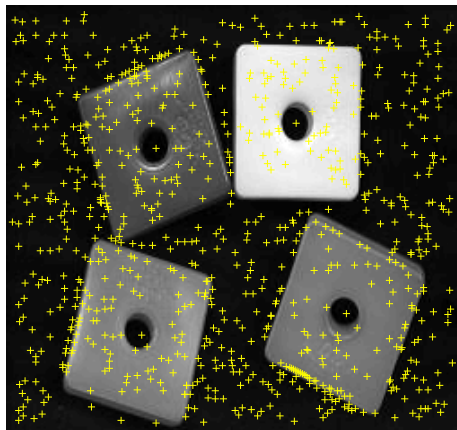
Figur 4.12: Bildepyramide SIFT

Det neste trinnet innebærer å bygge “Gausddifferanse-bildene” i henhold til likning 3.17. Resultatet av denne operasjonen er vist i figur 4.13.

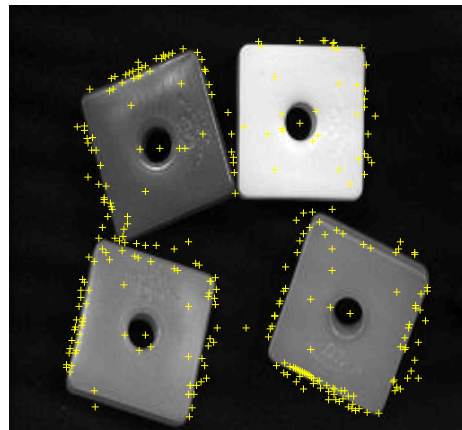
Når FIG-bildene (FIG=Forskjell i Gaussverdi) er generert begynner letingen etter punkt som egner seg for matching. Dette gjøres ved en beregning av lokale ekstrema i to intervaller i hver oktav. Etter første gjennomkjøring ble punkt merket som vist i figur 4.14. Så fjernes punkt med lav kontrast i forhold til omkringliggende piksler. Resultatet av denne operasjonen vises i figur 4.15. Deretter gjøres en tredje runde med fjerning av punkter. Denne gangen er det de punktene som er resultat av for stor kantrespons som forsvinner, se figur 4.16. I siste instans tilordnes de gjenværende punktene en orientering og en størrelse, markert med piler i figur 4.17



Figur 4.13: Bildepyramide SIFT, Difference of Gaussian

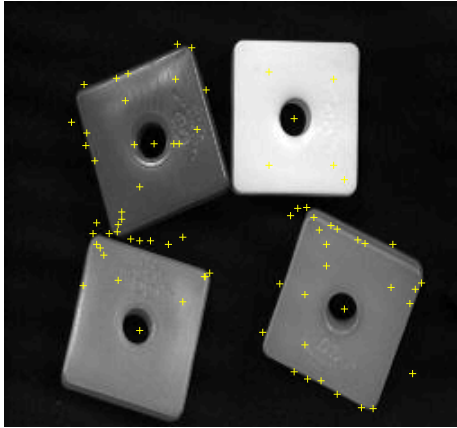


Figur 4.14: SIFT-kandidater 1

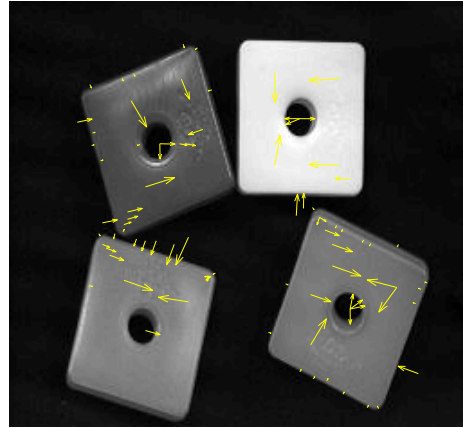


Figur 4.15: SIFT-kandidater 2

Som beskrevet i teoridelen lagres en vektor med 128 elementer for hver av punktbeskrivelsene i en fil. Denne databasen brukes videre i matchingen mot andre avbildninger av samme objekt. For å teste at dette fungerer gjør programmet en rotasjon av bildet, og matcher mot denne. Punktobjekter søkes og lagres i denne roterte versjonen på samme måte som for første bilde, og sammenliknes mot referansen. Deretter beregner programmet rotasjonsparametere for den affine transformasjonen. Disse sammenliknes med fasiten, siden det er kjent nøyaktig hvor mye bildet er rotert. Rotasjonsmatrisen ble for



Figur 4.16: Endelige SIFT-punkt



Figur 4.17: SIFT-punkt med størrelse og orientering

testbildet beregnet til å være

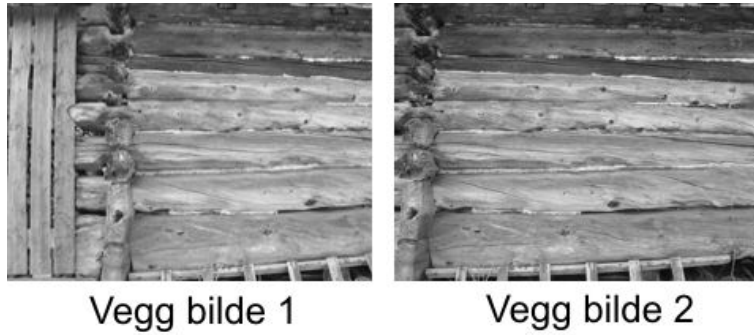
$$\begin{bmatrix} 0.7191 & -0.7134 & 127.2273 \\ 0.7128 & 0.7170 & -130.2423 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

mens den teoretisk sett skulle blitt

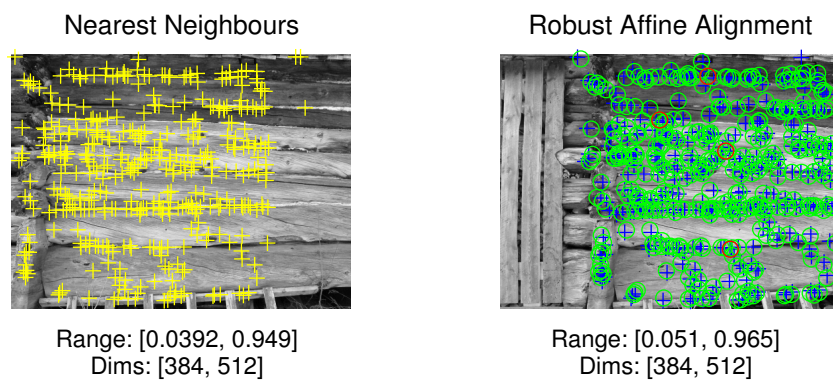
$$\begin{bmatrix} 0.7071 & -0.7071 & 127.2792 \\ 0.7071 & 0.7071 & -127.2792 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

Som man ser er avvikene små. SIFT-deskriptoren yter med andre ord godt på enkle testbilder. En test ble så kjørt på mer kompliserte bilder. Bildene av leveggen er tilnærmede stereoopptak, og burde matches godt. Veggen er vist i figur 4.18.

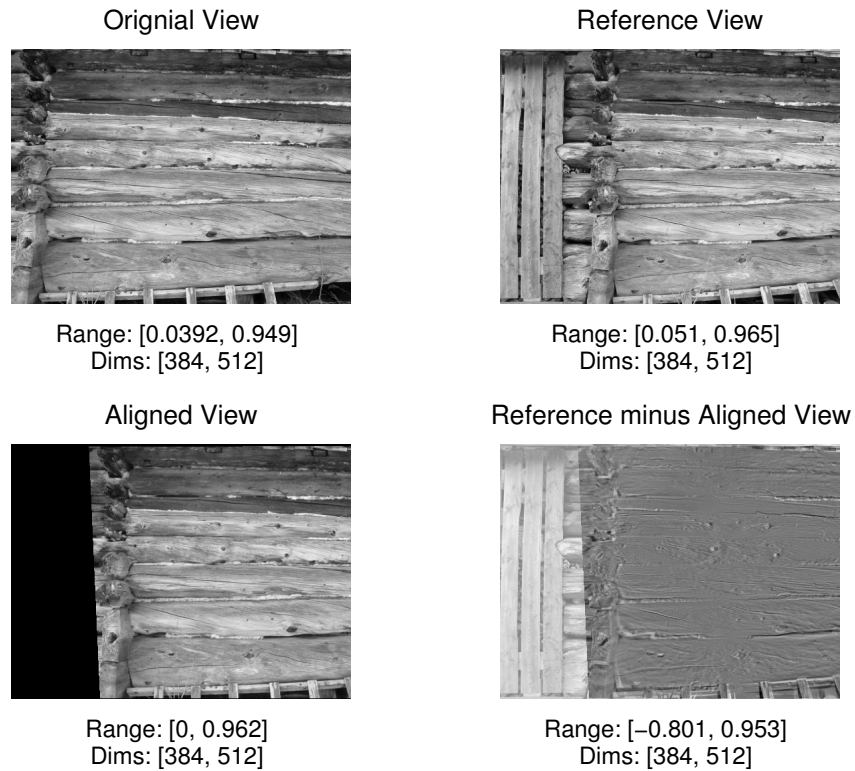
Resultat av SIFT-deteksjon i de to bildene er vist i figur 4.19. Korrekte treff er markert med grønt, grovfeil med rødt. På bakgrunn av disse nøkkelpunktene er bildene matchet mot hverandre som vist i figur 4.20.



Figur 4.18: Matchingsbilder tømmervegg



Figur 4.19: SIFT-punkter tømmervegg



Figur 4.20: Matchede bilder

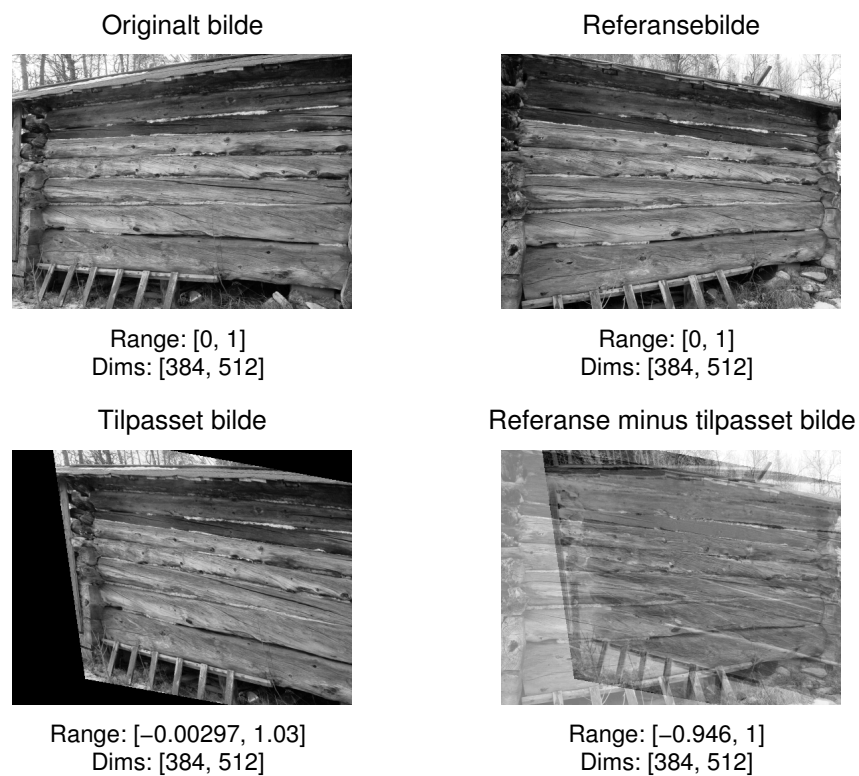
Den beregnede transformasjonsmatrisen mellom de to bildene ble som følger:

$$\begin{bmatrix} 0.9753 & 0.0438 & 116.7001 \\ -0.0002 & 0.9906 & 3.0469 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

Denne er beregnet på bakgrunn av 420 korrekte matcher mellom de to bildene. Det originale bildet var 512*384 piksler, altså relativt lite sett i forhold til den maksimale oppløsningen på moderne digitale speilreflekskamera. Bakgrunnen for å benytte bilder med såvidt dårlig oppløsning er minnekravene til beregningen av SIFT-deskriptoren. Programmet ble kjørt på en kraftig 3GHz Intel Xeon-maskin med 2GB minne. Denne

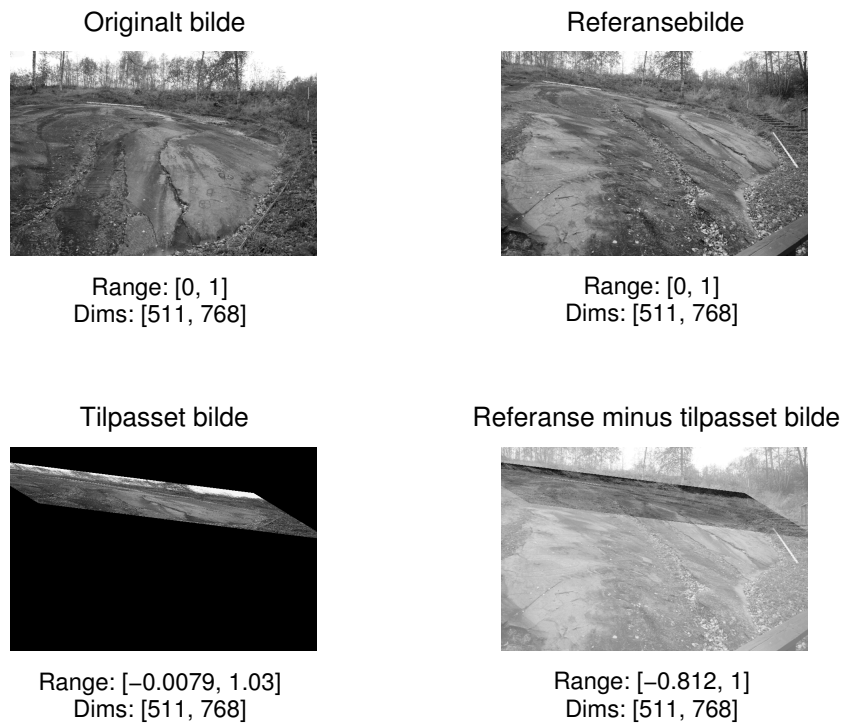
maskinen taklet bilder opp til 1024x768 piksler, men brukte da meget lang tid på beregningen.

Til slutt er det interessant å se på hvordan SIFT-matchingen fungerte for mer utfordrende opptakssituasjoner. Først ble to bilder av den samme leveggen testet. Disse var ikke stereoopptak. Bildene og resultatet av matchingen er vist i figur 4.21.



Figur 4.21: Levegg konvergente opptak

Det er tydelig at også SIFT-deskriptoren begynner å få problemer med så konvergente opptak. I senter av bildet er matchingen brukbar, men mot kantene blir feilprosenten stor. Samme test ble kjørt på 2 konvergente opptak fra helleristningsfeltet. Resultatet av denne matchingen er vist i figur 4.22. Her fant matchingsprogrammet bare 5 gode fellespunkt å matche mot, og den affine transformasjonen ble feilaktig beregnet.



Figur 4.22: Lerfald konvergente opptak

4.4 Implementasjon av matchingsprosessen

For Harris og Förstneroperatoren ble det gjort forsøk med en enkel implementasjon av krysskorrelasjonsmatching. Dette programmet leser inn resultatfilene fra punktdeleksjonen i de to bildene, og beregner krysskorrelasjonskoeffisienten. I filene er lagret gråverdiene for alle piksler i en 5x5-maske rundt punktet. Programmet gir også brukeren mulighet til å begrense søkeområdet dersom det er kjent hvor det overlappende området begynner og slutter i bildene. Da antar programmet stereopar av bilder i orthoprojeksjon. Det vil si at det overlappende området ligger i venstre side av det høyre bildet, og i høyre side av det venstre. I tillegg begrenses søkeområdet i y-retning til referanseobjektets y-koordinat $\pm 5\%$ av bildets høyde. Større parallakse enn dette bør

ikke opptre i stereobilder. Krysskorrelasjon for alle aktuelle kombinasjoner av punktpar beregnes, og de punktparene der korrelasjonskoeffisienten overstiger 0.95 samtidig som punktet er innenfor søkevinduet, beholdes som mulige treff.

Deretter beregner programmet en lineær transformasjon mellom de to stereobildene ut fra de samsvarende punktparene. Tabellen nedenfor viser en sammenlikning mellom beregnede koordinater og målte koordinater (målt i et bildebehandlingsprogram) for et punkt i sentrum av det overlappende området mellom bildene.

Metode	X	Y	Avvik x	Avvik Y
Målt	1411	541	0	0
SIFT	1400	541	-11	0
Krysskorr	1420	538	+9	-3

4.5 Implementasjon av modellkoordinatberegning

Det har blitt utviklet et enkelt program for beregning av modellkoordinater fra SIFT-objektene. Dette programmet er også laget i c++, og benytter seg av eksisterende orienteringsdata og kameradata for de to bildene. Kildekode og kjørbart fil for windows er tilgjengelig.

4.5.1 Kolinearitetslikningene

Modellkoordinatene beregnes med utgangspunkt i kolinearitetslikningene. Fra fotogrammetrien er disse godt kjent:

$$x = x_0 - c \frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (4.5)$$

$$y = y_0 - c \frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (4.6)$$

Med orienteringsdata og kameradata tilgjengelig, er det kun modellkoordinatene (X, Y og Z) til aktuelt punkt som er ukjent i disse likningene. Hvert matchet punkt er målt i to bilder, slik at det finnes fire observasjoner for å løse disse tre ukjente. Beregningen er implementert med en første beregning av terrengkoordinater på grunnlag av 3 likninger med tre ukjente. Her benyttes x og y fra bilde 1 og x fra bilde to. Deretter gjøres

samme beregning igjen, men da med y fra bilde to i stedet for x . Resultatet av de to beregningene sammenliknes deretter, og resultatet midles. En annen mulighet ville vært å benytte minste kvadraters metode i beregningen.

For å kunne gjøre disse beregningene har likningene 4.5 og 4.6 blitt satt opp som et likningssystem på formen $Ax = b$. Det medførte en omskrivning av kolinearitetslikningene slik at alle konstantledd ble samlet på høyre side av likhetstegnet. Resultatet av dette ble:

$$\begin{aligned} & ((x - x_0)r_{31} + cr_{11})X + ((x - x_0)r_{32} + cr_{12})Y + ((x - x_0)r_{33} + cr_{13})Z \\ &= ((x - x_0)r_{31} + cr_{11})X_0 + ((x - x_0)r_{32} + cr_{12})Y_0 + ((x - x_0)r_{33} + cr_{13})Z_0 \end{aligned}$$

$$\begin{aligned} & ((y - y_0)r_{31} + cr_{21})X + ((y - y_0)r_{32} + cr_{22})Y + ((y - y_0)r_{33} + cr_{23})Z \\ &= ((y - y_0)r_{31} + cr_{21})X_0 + ((y - y_0)r_{32} + cr_{22})Y_0 + ((y - y_0)r_{33} + cr_{23})Z_0 \end{aligned}$$

Og tilsvarende for bilde nummer 2. Dette ser noe uoversiktlig ut, men dersom det skrives på matriseform $Ax = b$ blir det lettere å se sammenhengen. Her er $x - x_0$ og $y - y_0$ forkortet til x_d og y_d , d for delta:

$$\begin{bmatrix} x_d r_{31} + cr_{11} & x_d r_{32} + cr_{12} & x_d r_{33} + cr_{13} \\ y_d r_{31} + cr_{21} & y_d r_{32} + cr_{22} & y_d r_{33} + cr_{23} \\ x_d r_{31} + cr_{11} & x_d r_{32} + cr_{12} & x_d r_{33} + cr_{13} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} konst_1 \\ konst_2 \\ konst_3 \end{bmatrix} \quad (4.7)$$

Konstantleddene på høyre side av likhetstegnet er vist i likning 4.7. De to første linjene i A og b -matrisene beregnes fra det første bilde, den siste fra bildet som matches mot. Matrisen x vil ved løsning av systemet inneholde en entydig bestemmelse av terrengkoordinatene til punktet som beregnes.

4.5.2 Cramers metode

Likningssettet med tre likninger og tre ukjente løses ved hjelp av Cramers metode. Metoden er basert på determinanter. Determinanten til en 3x3-matrise regnes ut slik:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{31}a_{12}a_{23} - a_{31}a_{22}a_{13} - a_{11}a_{32}a_{23} - a_{21}a_{12}a_{33}$$

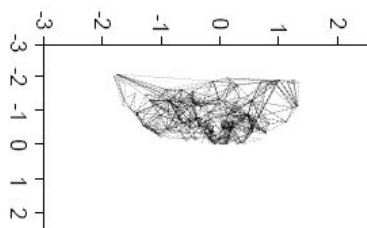
Cramers metode går ut på å erstatte enkeltkolonner i A med verdiene fra b . For beregning av X erstattes første kolonne i A med verdiene fra b . Determinanten til denne nye matrisen regnes ut, og verdien til X finnes ved å dele denne determinanten på determinanten til den opprinnelige A -matrisen. Deretter erstattes andre kolonne i stedet for første ved beregning av Y , og divisjon gjøres på samme måte. Til slutt erstattes tredje kolonne i A for å beregne Z -verdien til punktets modellkoordinater. For å utnytte overbestemmelsen gjøres deretter samme beregning en gang til, men denne gangen erstattes likningen basert på y_1 med en likning basert på y_2 . De beregnede terrengkoordinatene fra hvert likningssystem midles, og et enkelt standardavvik beregnes slik:

$$\sigma_i = \sqrt{(i_1 + i_2)^2} \quad (4.8)$$

der i er X , Y eller Z . Programmet skriver så de midlede terrengkoordinatene med tilhørende standardavvik til en tekstfil på harddisken. Det genereres også en fil med kun de midlede modellkoordinatene. Denne filen kan benyttes videre i et program for triangulering og generering av overflate for punktsky.

4.6 Modellkoordinatberegning

For SIFT-operatoren ble koordinatene til de matchede punktobjektene lagret i en tabulatorseparert tekstfil. Disse koordinatene er tilgjengelige i variabelen `p0` etter kjøring av matlabprogrammet. Den affine transformasjonen ble benyttet til å beregne koordinatene til punktene i referansebildet, og resultatene ble lagret til samme fil. I tillegg var tilnærmede kameradata for bildene tilgjengelig fra tidligere prosjekter med samme kamera og tilsvarende opptaksavstand. Kameradataene ble benyttet til orientering av bildene i programmet PhotoModeler. Kameradata og orienteringsparametere for hvert bilde ble også lagret i tabulatorseparerte tekstfiler. Disse filene ble benyttet for beregning av modellkoordinater for matchede punkter.



Figur 4.23: Triangulering av modellkoordinater

Siden ingen kjente grunnrisspunkt fantes i bildene var det ikke mulig å transformere til terreng. En implementasjon av terrengkoordinatberegning har da heller ikke blitt gjort i denne oppgaven, men programmet vil enkelt kunne utvides med slik funksjonalitet senere. Det ble gjort en test av triangulering av modellkoordinatene til de to bildene av leveggen. Resultatet av denne trianguleringen er vist i figur 4.23. Triangulering er utført ved hjelp av programpakken GMT [30]. Dette er en samling gratisprogrammer for manipulering av geografiske og kartesiske data. Programmet benytter standard Delaunay-triangulering.

Kapittel 5

Drøfting

I dette kapittelet drøftes resultatene fra det praktiske arbeidet. Det gis også antydninger om mulige feilkilder. Kapittelet må sees i relasjon til den informasjonen som er gitt i kapittel 4.

5.1 Interesseoperatorene

Valget av objektbasert matching er allerede begrunnet i kapittel 2.7. De testede interesseoperatorene har vist seg å gi svært forskjellig kvalitet hva utvalg av punktobjekter angår.

Av de to intensitetsbaserte operatorene viste Förstneroperatoren seg som den desidert svakeste. Denne markerte svært mange punkt, og av disse var en stor andel i områder med lav kontrast. Förstneroperatoren hadde også meget ujevn respons langs kanter i bildene. Harrisoperatoren var mer stabil. Som det er mulig å se av figurene 4.10 og 4.11 markeres for en stor del samsvarende punkt i to ulike bilder med Harrisoperatoren. Også denne viste seg imidlertid å ha problemer dersom rotasjonen mellom bildene ble for stor. For begge disse operatorene ble gråverdiene i en 5×5 -matrise rundt senterpikselen benyttet i videre matching. Det finnes også andre kriterier for matching mellom utvalgte punkt, SIFT-operatoren benytter en helt annen tilnærming.

SIFT-operatorens punktobjekter er karakterisert ved en parametrisk fremstilling. Som beskrevet er dette en 128-elementers vektor for hvert enkelt punkt. Dermed benytter ikke punktobjektene gråverdier som beslutningsgrunnlag i videre matching. I stedet har

hvert enkelt punkt både en orientering og en størrelse tilknyttet seg. Størrelsen utregnes på bakgrunn av gradienten, og retningen peker den veien gradienten er brattest (har størst verdi). Testresultatene viser at SIFT-operatoren finner færre kandidatpunkter til matching enn de to andre, men for bilder med god geometri under opptak vil brorparten av disse punktene medføre en korrekt match. Det kan altså konkluderes med at av de testede interesseoperatorene er Lowes SIFT-operator den som gir best resultater i videre matching. Svakheterne med denne er den store beregningskostnaden. Særlig er det kalkulering av 128-elementers deskriptorvektoren for hvert nøkkelpunkt som tar lang tid. I tillegg fungerer den dårlig på store bilder. Årsaken til dette ligger dels i matlabs måte å håndtere bildefiler på, og dels i SIFT-operators bruk av bildepyramider.

5.2 Matching av punkt

Det egenutviklede programmet for matching av punkter fra de to intensitetsbaserte interesseoperatorene er basert på krysskorrelasjon, og er således ikke av de mest avanserte. Det finnes som nevnt i kapittel 2 flere andre metoder som vil gi sikrere beregning av transformasjonen mellom de to bildene. Programmet er i tillegg avhengig av stereoopptak. Dersom rotasjonen blir for stor, vil punkt som egentlig er samhørende ikke bli matchet. Andre problemer knyttet til denne implementasjonen er lang kjøretid. Det gjøres heller ingen sjekk på om et punkt allerede er markert som kandidat for match med et annet punkt. Det viste seg imidlertid at krysskorrelasjonsmatchingen ga overraskende gode resultat for tilnærmede stereoopptak. Dette er vist i tabellen i kapittel 4.4. Det er allikevel en større robusthet i SIFT-operatoren, siden den benytter en affin transformasjon. I tillegg var bildene som SIFT ble testet på bare 1/4 av de originale bildene som Harrisoperatoren benyttet. God oppløsning i bildene er viktig for å oppnå god nøyaktighet. En SIFT-implementasjon som takler store filer bør derfor oppnå mye bedre nøyaktighet enn krysskorrelasjonsmatching med Harrisoperatoren.

De beregnede modellkoordinatene er belemret med stor usikkerhet, siden de kun er beregnet fra målinger i to bilder. Det er også noe usikkerhet i de beregnede kamera- og orienteringsparameterene. Orienteringsdata er hentet fra måling i programmet PhotoModeler fra EOS systems. Figur 4.23 viser resultatene fra triangulering av leveggen. Punktene som ble modellert er markert med grønt i figur 4.19. Det er tydelig at en del av modellkoordinatene ikke er korrekt beregnet. I tillegg til feil i orienteringsdata og transformasjon skyldes dette at kun to bilder benyttes. Det er med andre ord grunn til å anta bedring i resultatene dersom flere bilder inkluderes, og minste kvadraters metode benyttes i beregningen.

Kapittel 6

Konklusjoner og videre arbeid

Denne rapporten har gjengitt teori og resultater fra arbeidet med automatisk måling og matching av mange punkt på 3-dimensjonale flater. De viktigste prinsippene for matching har blitt gjengitt. Det er konkludert med at objektbasert matching er mest optimalt ved 3D-rekonstruksjon, både med tanke på kjøretid og nøyaktighet.

I objektbasert matching er interesseoperatorer et viktig element. Dette er metoder for deteksjon av punkt i bildene som egner seg godt for matching. Kriteriene for valg av slike punkt er at de skal være distinkte, uforanderlige, stabile, sjeldne og tolkbare. Praktiske tester er utført på tre av disse interesseoperatorene. Førstneroperatoren ga lavt nøyaktighetspotensiale i matchingen, og fant få samsvarende punkter i testbildene. Den markerte i tillegg svært mange punkt med dårlig kontrast i forhold til nærliggende piksler. Harrisoperatoren fungerte godt for stereobilder, men får problemer ved store rotasjoner. SIFT-operatoren yter best på store transformasjoner og minst like bra som Harrisoperatoren for stereobilder.

Den største begrensningen i de testene som er gjort er at de kun har blitt utført på par av bilder. Det har ikke blitt tid til implementasjon og testing av flerstasjons konvergente opptak, selv om prinsippene for dette har blitt beskrevet. Programmet som har blitt laget for beregning av modellkoordinater er avhengig av kjent orientering og kjente kameradata for bildene. Det betyr at slike data må være tilgjengelig fra andre programmer. Et program som kan benyttes til dette er PhotoModeler fra EOS systems. Også beregning av modellkoordinater vil få mye bedre nøyaktighet dersom flere bilder benyttes.

6.1 Mulig fremtidig arbeid

Den teoretiske delen av rapporten gir en grundig beskrivelse av eksisterende matchingsteknikker. Dersom videre arbeid med programvaren som er laget blir aktuelt, vil det være naturlig å starte med implementering av funksjonalitet for arbeid på flere bilder samtidig. Det er konkludert med at SIFT-deskriptoren av LOWE er den som egner seg best til nærfotogrammetrisk matching og måling av punkt på 3-dimensjonale flater. Den testede implementasjonen er kodet i matlab. Matlab har mange sterke sider, men kjøretiden blir meget lang og arbeid på store filer er umulig selv på kraftig maskinvare. Minnet fylles rett og slett opp. En ny implementasjon av SIFT-deskriptoren i c++ eller et annet maskinvareoptimalisert språk vil derfor være nødvendig. Dagens digitale kamera har god oppløsning, og denne bør utnyttes for å kunne lage nøyaktige flatemodeller.

Annen funksjonalitet som vil være mulig å bygge inn i systemet er automatisk kamerakalibrering og orientering. Dersom dette gjøres bør det være mulig for brukeren å utføre visuell kontroll med matchingsarbeidet. Automatisk generering av flatemodeller i nærfotogrammetri er et spennende fagområde, og et område som stadig er i utvikling. Automatisert måling og matching bør kunne utvikles med tilfredstillende nøyaktighet. Programvare og teori fra denne rapporten danner grunnlag for en slik videreutvikling.

Bibliografi

- [1] Atkinson. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, 1996.
- [2] C Baillard og O Dissard. A stereo matching algorithm for urban digital elevation models. *Photogrammetric Engineering and Remote Sensing*, 66(9):1119–1128, 2000.
- [3] S T Barnard og W B Thompson. Disparity analysis of images. *IEEE, PAMI* - 2:333–340, 1981.
- [4] A Baumberg. Reliable feature matching across widely separated views. *Conference on Computer Vision and Patern Recognition, Hilton Head, South Carolina*, side 774–781, 2000.
- [5] R Bellmann. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, USA, 1957.
- [6] Dan Bloomberg. Leptonica. <http://www.leptonica.com>.
- [7] MinGW Compiler. Mingw 4.0.5. <http://www.mingw.org>.
- [8] Wolfgang Förstner. On the precision of digital correlation. *Int. Arch. of Photogrammetry and Remote Sensing*, 24-3:176–189, 1982.
- [9] Wolfgang Förstner. A feature based correspondence algorithm for image matching. *Int. Arch. of Photogrammetry*, 26-3/3, 1984.
- [10] A Grün. Adaptive least squares correlation - a powerful image matching technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3):175–187, 1985.
- [11] M. J. Hannah. Computer matching of areas in stereo images. *P.H.D. Thesis, Stanford University*, 1974.

- [12] C Harris og M Stephens. A combined corner and edge detector. *Fourth Alvey Vision Conference, Manchester, UK*, side 147–151, 1988.
- [13] Christian Heipke. An overview of image matching techniques. *OEEPE - Workshop on the Application of Digital Photogrammetric Workstations, OEEPE Official Publications*, 3:173–189, 1996.
- [14] G L Hobrough. Automatic stereo plotting. *PE & RS*, (25)5:763–769, 1959.
- [15] Prof. Allan Jepson og Prof. David Fleet. Csc 487/2503 foundations of computer vision. <http://www.cs.toronto.edu/~jepson/csc2503/>.
- [16] J J Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.
- [17] T Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.
- [18] David G. Lowe. Object recognition from local scale-invariant keypoints. *International Conference on Computer Vision, Corfu, Greece*, side 1150–1157, 1999.
- [19] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [20] J. Chris McGlone. *Manual of Photogrammetry*. American Society for Photogrammetry and Remote Sensing, Bethesda, Maryland, 2004.
- [21] K Mikolajczyk og C Schmid. An affine invariant interest point detector. *European Conference on Computer Vision, Copenhagen, Denmark*, side 128–142, 2002.
- [22] H. Moravec. *Rover Visual Obstacle Avoidance*. International Joint Conference on Artificial Intelligence, Vancouver, Canada, 1981.
- [23] Donovan Parks og Jean-Philippe Gravel. Corner detectors. <http://www.cim.mcgill.ca/~dparks/index.htm>.
- [24] F Schaffalitzky og A Zisserman. Multi-view matching for unordered image sets, or “how do i organize my holiday snaps”. *European Conference on Computer Vision, Copenhagen, Denmark*, side 414–431, 2002.
- [25] C Schmid og R Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.
- [26] Cordelia Schmid, Roger Mohr og Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, side 151–172, 2000.
- [27] Bloodshed software. Dev-c++ version 4.9.9.2. <http://www.bloodshed.net>.

- [28] T Tuytelaars og L Van Gool. Wide baseline stereo based on local, affinely invariant regions. *British Machine Vision Conference, Bristol, UK*, side 412–422, 2000.
- [29] University of Hamburg Ullrich Köthe Cognitive systems group. Vigna 1.3.2. <http://kogs-www.informatik.uni-hamburg.de/~koethe/vigna/>.
- [30] Paul Wessel og Walter H F Smith. Gmt - the generic mapping tools. <http://gmt.soest.hawaii.edu/>.
- [31] B Wrobel. Facet stereo vision (fast vision) . a new approach to computer stereo vision and to digital photogrammetry. *Proceedings ISPRS Intercommision Workshop on Fast Processing of Photogrammetric Data*, side 231–258, 1987.
- [32] Z Zhang, R Deriche, O Faugeras og Q T Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, side 78,87–119, 1995.

Tillegg A

Matematiske definisjoner

Affin transformasjon

En affin transformasjon (6-parameter transformasjon) består av en lineær kombinasjon av translasjon, rotasjon og skalering. Den kan også inkludere skjærtransformasjonen (ikke-uniform skalering i noen retninger) i stedet for skalering. For pikselverdier kan en affin transformasjon uttrykkes slik:

$$\begin{aligned}r_2 &= a_{rr}r_1 + a_{rc}c_1 + t_r \\c_2 &= a_{cr}r_1 + a_{cc}c_1 + t_c\end{aligned}$$

der a_{ij} og t_k er de affine transformasjonsparameterene.

Eigenverdi

En matrisers egenverdi (eigenvalue) kan for små matriser beregnes symbolsk ved å bruke et karakteristisk polynom. For de metodene som er nevnt i artikkelen er det kun små matriser som er aktuelle, for eksempel matrisen M i Harrisoperatoren. For en 2×2 -matrise

beregnes egenverdiene som følger:

$$M = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$
$$p(x) = \det(M - \lambda I) = \det \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} = (\lambda - 1)^2$$

Dette eksempelet er et litt spesielt tilfelle, siden det bare har en egenverdi. Som regel har 2x2-matriser 2 egenverdier. Egenverdien her blir altså 1.

Gaussisk funksjon

En Gaussisk funksjon er en funksjon på formen

$$f(x) = ae^{-(x-b)^2/c^2}$$

For reelle konstanter a, b og c.

Gradient

Gradienter er den enkleste formen for kantdeteksjon i bilder. Det er en diskret beregning av den deriverte til en piksels gråtoneverdi. Når en piksel har en derivert med høy verdi betyr det at pikselen ligger på en kant. Beregning av gradienter er vist i likningene 3.8 og 3.9.

Konvolusjon

En konvolusjon er en matematisk operasjon som tar to funksjoner f og g og produserer en tredje funksjon. Denne tredje funksjonen representerer overlappet mellom f og en translert versjon av g . Den diskrete versjonen av konvolusjon er den mest aktuelle i bildebehandlingssammenheng. Matematisk uttrykk:

$$(f \otimes g)(m) = \sum_n f(n)g(n - m)$$

Lineær interpolasjon

Dette er en metode for tilnærming av en ukjent verdi dersom to verdier er gitt og forandringsraten er konstant. Formelen for lineær interpolasjon i en dimensjon er gitt ved:

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \quad (\text{A.1})$$

$f(x)$ kan for eksempel være gråverdier, x_i er pikselkoordinater.