

# AUTOMATISK 3D-MODELLERING VED HJELP AV STILLBILDER

AUTOMATIC 3D MODELING USING STILLS

JØRGEN BASMO BJØRNSTAD OG STEIN ARNE FEDREHEIM

UNIVERSITETET FOR MILJØ- OG BIOVITENSKAP

INSTITUTT FOR MATEMATISKE REALFAG OG TEKNOLOGI  
MASTEROPPGAVE 30 STP. 2012



## **FORORD**

Denne masteroppgaven inngår som avsluttende del av et 5-årig masterstudium i Fotogrammetri på linjen Geomatikk, kart, satellitter og 3D-modellering ved Institutt for Matematiske Realfag og Teknologi (IMT), Universitetet for Miljø- og Biovitenskap (UMB). Oppgaven er på 30 studiepoeng. Oppgaven er skrevet på oppdrag fra Forsvarets Forskningsinstitutt (FFI). Hos FFI inngår oppgaven som del av et større prosjekt.

Hovedveileder for oppgaven har vært Ivar Maalen-Johansen ved seksjonen for Geomatikk, IMT. Vi takker for hjelp og veiledning underveis.

Vi takker Lorns Harald Bakstad og Trym Haavardsholm ved FFI for et bra samarbeid og ikke minst for rask hjelp og respons på henvendelser fra oss.

Takk til familie og alle andre som har vært tålmodige og til støtte for oss under de mest intensive periodene.

Universitetet for Miljø- og Biovitenskap, Mai 2012

Jørgen Basmo Bjørnstad

Stein Arne Fedreheim

## **SAMMENDRAG**

Både militært og sivilt har man stor tilgang til bilde og videomateriale fra enkle og mere komplekse sensorer. Med enkle sensorer mener man f.eks. speilreflekskamera eller mobilkamera. De mer komplekse sensorene kan være bemannede og ubemannede farkoster som fly og satellitter. Altså sensorer som kan operere både på land, til sjøs, i lufta eller i verdensrommet. Forsvaret har fram til nå benyttet disse bildene stort sett i 2D-format for bildeanalyse av ulike karakterer. Slikt materiale har stor verdi som planleggings- og kartleggingsmateriale og man kan ved hjelp av 3D-modellering utnytte potensialet i slike bilder enda bedre. I den forbindelse har Forsvarets Forskningsinstitutt (FFI) startet en innledende studie på 3D-rekonstruksjon fra stillbilder og video. Vår oppgave har til hensikt å støtte denne studien ved å kartlegge mulighetene for å gjøre dette så automatisert som mulig og å kunne si noe om forventet nøyaktighet og kvalitet på visualisering av modellene. Samtidig skal vi kunne si noe om hvilken software som kan være aktuell for formålet.

For å finne ut hvilke programmer som er tilgjengelige og i hvor stor grad de er automatiserte, har vi gjort studier av software som er open source og noen som er lisensbelagt. FFI ønsket et program som var så automatisert at man kun behøver å laste opp bildene og få ut en ferdig modell som det er mulig å gjøre gode avstandsmålinger i og som kan GEO refereres. De fire programmene vi endte opp med å sammenligne med hverandre, var Autodesk 123D Catch, SFM Toolkit, Visual SFM og PhotoModeler Scanner. Vi fant ut at ingen av disse programmene er 100 % automatisert siden alle er avhengig av at man manuelt går inn og gjør en eller flere av stegene i prosessen. Autodesks 123D-Catch er det programmet som framstår som mest automatisert. De andre programmene er avhengig av tilleggsprogram for å etterprosessere punktskyene for å få ferdige modeller. Til dette har vi benyttet MeshLab som i og for seg er ganske brukervennlig.

PhotoModeler Scanner er det eneste programmet som har det vi anser som komplett støtte for bruk av geografisk datum, mens Visual SFM kan benytte geografisk data i en begrenset grad. Ingen av de andre programmene innehar en slik funksjon.

Bildesettene vi har benyttet er fotografert med sensorer man normalt kommer i befatning med i militær sammenheng, altså speilreflekskamera. Det vil si at vi i all hovedsak har gjort undersøkelser med stillbilder. Bildene ble tatt fra helikopter og objektene som primært skulle modelleres, var ulike bygninger. Vi har også lyktes med å modellere landskapet rundt bygningene.

Med tanke på de visuelle 3D-modellene man kan få ut gir alle programmene stabilt pene modeller med unntak av PhotoModeler Scanner som krever mye av brukeren for å få til dette. Det programmet som ga penest tekstur, var Autodesk123D-Catch, men modeller generert i MeshLab (SFM Toolkit og Visual SFM) fulgte like bak. Her er våre antakelser at Autodesk 123D-Catch benytter seg av meget bra algoritmer for hull-fylling der punktskyen ikke er fullstendig. Det samme gjelder for tekstureringen av modellen.

Ambisjonene til Forsvaret var at en ferdig skalert modell burde gi målefeil som ikke var særlig større enn 10cm. Nøyaktigheten for avstandsmålinger ble målt ved sammenligning av avstander i modellene, målt med programmenes egne måleverktøy, og referanseavstander målt fysisk på bygningene. Våre resultater gir målefeil mellom 5 cm og 10 cm, der det beste programmet var SFM Toolkit med 5 cm. Vi antar at målenøyaktigheten vil være avhengig av metode som er benyttet for skalering og vi ser også muligheter for at en bruker vil gjøre ulike feil under måling. Dette tilfredsstiller FFIs ambisjoner i stor grad. For å kunne forklare eventuelt ulik nøyaktighet for programmene seg i mellom har vi gjort en sammenligning av kameraposisjoner beregnet under relativorientering av bildene. Her kom vi fram til at det var både små og store forskjeller mellom programmene, avhengig av funksjonalitet.

Vår sluttkonklusjon er at det finnes ferdige programmer som tilfredsstiller deler av Forsvarets behov, men at disse bør raffineres ytterligere for å være optimale avhengig av brukere og etterspurt funksjonalitet.

## SUMMARY

Huge amounts of imagery and full-motion video are generated on a daily basis, both in the military and in the civil society. This imagery is derived from complex sensors as satellites or UAVs, but primarily from noncomplex sensors as single-lens reflex cameras (SLRs) or cell-phone cameras. This could be done from land, sea, air or space. The military have until now primarily used the imagery as it is (2-D) for different purposes. Images or video material derives a lot of valuable information for combat preparations and intelligence in many aspects. Therefore the military has a great opportunity to extend their capacities. This may be done by taking the advantage of the unused potentials of 3D-modeling from imagery. The Norwegian Defence Research Establishment (FFI) has initiated a research project on 3D-reconstructions from pictures and full-motion video. The purpose of our master thesis is to support this project, by identifying the existing possibilities of solving this as automated as possible. Next we will describe the expected accuracy and quality of the final models. By tests of already existing software, we will try to find out which software may satisfy the criteria.

We have used both accessible open-source and licensed software to find out to what extend the tested programs are automated. FFI's ambitions is primarily a software which is automated enough to let a user simply load up images and get a complete 3D-model with good accuracy for length measurements as output. The user should also be able to geo-reference the model if geo-data is accessible.

The four programs we have compared are Autodesk 123D Catch, SFM Toolkit, Visual SFM and PhotoModeler Scanner. Our tests revealed that none of these are 100 % automated. Each of the tested programs are depending that the user does one or more steps of the process semi-manual. Either a main step (i.e. generate TIN or texturing) or a sub step (i.e. start the next operation or scaling). Autodesk 123D-Catch is the most automated program we have tested. Visual SFM is assessed to have a great potential in the future but doesn't generate a complete model (depended on MeshLab). The other programs are also depending of an extra program or manual interference for post-processing of the dense point clouds, to generate a scaled and textured TIN-surfaced model. For this purpose we have used MeshLab, an open-source program which is quite easy to use.

PhotoModeler Scanner is the only program with a more or less complete module implemented for handling geodetic datum and data from geo-tagged images. Visual SFM is the second

program with such a module. Despite the fact that it works very well and is very easy to use, it's a little bit premature at the moment. None of the other programs possess such a function.

The image sets used in our test-projects are stills, taken with SLRs from a helicopter. They were taken from a helicopter and our objects planned for modeling were four different buildings. We have been able to model the terrain surrounding the buildings.

Considering the visual impressions, all the software tested generates stable and good looking 3D-models. The only exception is PhotoModeler Scanner which requires too much manual operations and knowledge of the different functions of the program. Autodesk 123D-Catch gives the most authentic textures, but models textured in MeshLab (SFM Toolkit and Visual SFM) are very good as well. Our assessment is that Autodesk 123D Catch uses very good texturing and hole-filling algorithms which give the models a very nice and smooth surface even for areas described by very few points in the dense point cloud.

The ambition of FFI was that a scaled model should lead to measurement errors that were not particularly greater than 10 cm. The accuracy of distance measurements were derived by comparing the distances in the models, measured by the application's own measuring tools, and reference distances measured physically on the buildings. Our results provide errors from 5 cm to 10 cm. The best program was SFM Toolkit by 5 cm. We assume that the accuracy depends on the method used for scaling and we also find it most likely that errors caused by the user will affect the accuracy. This satisfies the FFI's ambitions to a large extent. In order to explain the varying accuracy of the programs, we have made a comparison of camera positions calculated by the relative orientation of the images.

Our final conclusion is that there are existing programs which will provide valuable products as they are at the moment, but in order to be as optimal as possible they should be further refined. There are both small and large differences between the programs depending on functionality and in order to produce the optimal application for military purposes, FFI should combine functionalities from each program.

# INNHALDSFORTEGNELSE

FORORD .....	I
SAMMENDRAG .....	II
SUMMARY .....	IV
INNHALDSFORTEGNELSE .....	VI
FIGURLISTE .....	IX
TABELLISTE .....	X
<b>1. INNLEDNING .....</b>	<b>1</b>
1.1. BAKGRUNN FOR OPPGAVEN.....	1
1.2. TIDLIGERE RELEVANTE UNDERSØKELSER .....	3
1.3. FORUTSETNINGER/AVGRENSNINGER .....	4
<b>2. 3D REKONSTRUKSJON .....</b>	<b>5</b>
2.1. DETEKSJON AV SAMMENBINDINGSPUNKT .....	8
2.2. ORIENTERING AV BILDER/"STRUCTURE FROM MOTION" .....	11
2.3. DENSE IMAGE MATCHING .....	12
2.3.1. SEMI-GLOBAL MATCHING.....	14
2.3.2. PATCH BASED METHODS .....	15
2.3.3. OPTIMAL FLOW ALGORITHMS .....	18
2.4. FAKTORER SOM PÅVIRKER EN 3D-REKONSTRUKSJON .....	19
<b>3. METODE.....</b>	<b>22</b>
3.1. FOTOGRAFERING .....	22
3.2. SOFTWARE .....	22
3.3. REFERANSEMÅLING .....	24
3.4. PROSJEKTER/MODELLERING.....	24
3.5. SKALERING .....	26
3.6. MÅLING .....	27
3.7. ANALYSE AV MÅLINGER .....	27
3.8. ANDRE DATA OG BRUKSOMRÅDER .....	28

4.	BESKRIVELSE AV SOFTWARE .....	29
4.1.	SFM TOOLKIT 3/PHOTOSYNTH TOOLKIT 7/VISUAL SFM.....	29
4.1.1.	SFM TOOLKIT 3 .....	29
4.1.2.	BUNDLER .....	30
4.1.3.	CMVS/PMVS2.....	31
4.1.4.	PHOTOSYNTH TOOLKIT 7 .....	34
4.1.5.	VISUAL SFM .....	35
4.2.	AUTODESK 123D CATCH .....	37
4.3.	PHOTOMODELER .....	39
4.3.1.	AUTOMATISK PROSESS/SMARTMATCH .....	40
4.3.2.	DENSE SURFACE.....	42
4.3.3.	EXTERNAL GEOMETRY .....	43
5.	RESULTATER.....	44
5.1.	GRAD AV AUTOMATISERING .....	44
5.1.1.	SFM TOOLKIT.....	44
5.1.2.	VISUAL SFM .....	45
5.1.3.	AUTODESK 123D CATCH .....	46
5.1.4.	PHOTOMODELER .....	47
5.1.5.	DELKONKLUSJON .....	48
5.2.	IVARETAKELSE AV GEOMETRI OG TEKSTUR .....	48
5.2.1.	SFM TOOLKIT .....	49
5.2.2.	VISUAL SFM .....	50
5.2.3.	AUTODESK 123D CATCH .....	50
5.2.4.	PHOTOMODELER .....	52
5.2.5.	VISUALISERING AV MODELLENE .....	54
5.2.6.	SAMMENLIGNING AV RELATIV ORIENTERING VS GPS-KOORDINATER .....	55
5.2.7.	DELKONKLUSJON .....	56
5.3.	SKALERING AV 3D-MODELL .....	57



5.3.1.	SFM TOOLKIT .....	57
5.3.2.	VISUAL SFM .....	58
5.3.3.	AUTODESK 123D CATCH .....	59
5.3.4.	PHOTOMODELER .....	60
5.3.5.	DELKONKLUSJON .....	61
5.4.	UTNYTTELSE AV OBSERVERTE STØRRELSER .....	61
5.4.1.	SFM TOOLKIT .....	61
5.4.2.	VISUAL SFM .....	61
5.4.3.	AUTODESK 123D CATCH .....	63
5.4.4.	PHOTOMODELER .....	63
5.4.5.	DELKONKLUSJON .....	63
5.5.	HVILKEN NØYAKTIGHET KAN OPPNÅS .....	64
5.5.1.	RESULTATER .....	64
5.5.2.	FEILKILDER .....	65
5.5.3.	DELKONKLUSJON .....	67
5.6.	ANDRE DATA OG BRUKSOMRÅDER .....	68
5.6.1.	KAMERA KALIBRERING .....	68
5.6.2.	SAMMENLIGNING AV <i>FEATURE DETECTION</i> OG <i>MATCHINGSALGORITME</i> .....	69
5.6.3.	SFM TOOLKIT .....	69
5.6.4.	VISUAL SFM .....	70
5.6.5.	AUTODESK 123D CATCH .....	71
5.6.6.	PHOTOMODELER .....	71
5.6.7.	DELKONKLUSJON .....	72
5.7.	OPPSUMMERING AV RESULTATER.....	72
6.	KONKLUSJON .....	74
7.	VEIEN VIDERE .....	76
	REFERANSELISTE .....	77
	VEDLEGG .....	79

## FIGURLISTE

FIGUR 1 EKSEMPLER PÅ <i>FEATURE DETECTION</i> [7].....	9
FIGUR 2 TF/IMT PROSESSERT MED SIFTGPU, TO BILDER MED FELLES SAMMENBINDINGSPUNKT. MERK AT NOEN ER TYDELIG FEIL MATCHET.....	11
FIGUR 3 VISUALISERING AV CELLER OG PATCHER[14] .....	16
FIGUR 4 EKSEMPEL PÅ EKSPANSJON[14].....	17
FIGUR 5 PHOTOMODELERS FAKTORER FOR NØYAKTIGHET .....	20
FIGUR 6 EKSEMPEL PÅ REFERANSEBILDE MED AVSTANDENE SOM SKULLE MÅLES FOR DELER AV "FLØY 3" .....	24
FIGUR 7 SPREDT PUNKTSKY GENERERT AV BUNDLER.....	31
FIGUR 8 TETT SKY GENERERT AV PMVS2 VIST I MESHLAB (KAMERAPOSISJONER OVER SELVE MODELLEN).....	33
FIGUR 9 SPREDT PUNKTSKY I PHOTOSYNTH AV TF/IMT LAGET FRA 82 BILDER .....	34
FIGUR 10 2D-VIEWEREN I PHOTOSYNTH .....	35
FIGUR 11 VISUAL SFM GUI MED ET PROSJEKT UNDER ORIENTERING.....	36
FIGUR 12 VERKTØYLINJEN I VISUAL SFM.....	36
FIGUR 13 TETT PUNKTSKY AV HAVNEOMRÅDE VED SJURSØYA, LAGET MED 211 BILDER.....	37
FIGUR 14 SMART POINTS GENERERT AV PHOTOMODELER .....	41
FIGUR 15 MULIGHETER I PM CREATE DENSE SURFACE .....	42
FIGUR 16 CHECK POINTS I EXTERNAL GEOMETRY .....	43
FIGUR 17 STEGENE FOR Å GJENNOMFØRE EN 3D-REKONSTRUKSJON I VISUAL SFM .....	46
FIGUR 18 PUNKTSKY AV TF GENERERT AUTOMATISK AV PHOTOMODELER.....	47
FIGUR 19 3D MODELL AV MEIERIKRYSSET GENERERT AV AGISOFT PHOTOSCAN .....	48
FIGUR 20 EKSEMPEL PÅ "PLEN-PUNKT" FESTET TIL TAK KANTEN.....	49
FIGUR 21 NORMAL TIN FRA MESHLAB .....	50
FIGUR 22 QUADNETTVERK AV SAMME MODELL.....	50
FIGUR 23 3D MODELL GENERERT AV AUTODESK 123D CATCH MED 3 BILDER .....	51
FIGUR 24 3D MODELL GENERERT AV PHOTOMODELER .....	52
FIGUR 25 KAMERAPOSISJONER VS GPS- POSISJONER .....	53
FIGUR 26 MÅLEMENY AUTODESK 123D CATCH .....	59
FIGUR 27 MÅLINGER I AUTODESK 123D CATCH .....	60
FIGUR 28 AVVIK MELLOM RELATIV KAMERAPOSISJON(RØD) OG GPS-POSISJON(BLÅ).....	62
FIGUR 29 SAMMENLIGNING AV KOORDINATER HENTET FRA MODELL (HØYRE), KONTROLLERT (PUNKT 2) I GOOGLE EARTH (VENSTRE).....	62
FIGUR 30 CHECK POINT MED HØY FEIL FOR TF/IMT .....	65
FIGUR 31 MÅLINGER PÅ MEIERIBYGGET I PHOTOMODELER .....	66
FIGUR 32 SAMMENLIGNING AV RMSD FOR 3M-INTERVALLER OG TOTALT FOR ALLE PROGRAM .....	67
FIGUR 33 PUNKTSKY OVER LØRENSKOG.....	70
FIGUR 34 POISSONMODELL AV LØRENSKOG .....	70
FIGUR 35 RÅDHUSET PÅ LØRENSKOG MED AUTODESK 123D CATCH.....	71

## TABELLISTE

TABELL 1 PROSEDYRER FOR AUTOMATISK 3D-MODELLERING .....	7
TABELL 2 PROSJEKTER .....	25
TABELL 3 GENERERTE 3D MODELLER .....	54
TABELL 4 RESULTATER AV 7-PARAMETER TRANSFORMASJON .....	55
TABELL 5 RESULTATER FRA 7-PARAMETER TRANSFORMASJON MED SKALERTE KOORDINATER FOR SØRHELLINGA. RMS I METER. ....	55
TABELL 7 VERDIER FOR SKALERING VED BRUK AV SM4 .....	58
TABELL 8 RMSD (METER) AV ALLE MÅLINGER (SKALERINGSMETODE).....	64
TABELL 9 RMSD (METER) AV MÅLINGER MED SFM TOOLKIT MED OG UTEN GPS SKALERING .....	64
TABELL 11 SAMMENLIGNING AV KALIBRERINGSDATA.....	68
TABELL 12 OPPSUMMERING AV RESULTATER .....	72
TABELL 13 FORKLARING PÅ SYMBOLER .....	72

# 1. INNLEDNING

## 1.1. Bakgrunn for oppgaven

Bilder av flere forskjellige typer nyttes av Forsvaret både til trening og i forbindelse med operasjoner. Dette kan være bilder tatt fra både eleverte- og bakkebaserte sensorer. Med eleverte sensorer menes alt fra speilreflekskamera fra helikopter til bilder fra satellitter og med bakkebaserte systemer menes ulike sensorer på land eller sjø. Kamerakvaliteten kan være svært varierende og posisjonen kan være ukjent. Utnyttelsen av disse bildene har vært varierende. De fleste av disse nyttes i liten grad til noe annet enn bare å studeres rent visuelt. FFI vil prøve å utvide bruksområdet for dette bildematerialet, og ser for seg en full 3D-rekonstruksjon. Ideelt sett ønsker FFI et program hvor brukeren selv kan laste opp bildene for deretter å få ut en 3D-modell. Med andre ord en mer automatisert og enklere tilnærming enn de programmer man har hatt tilgang til fram til nå. Denne 3D-modellen skal i tillegg kunne skaleres slik at det er mulig å gjøre enkle avstandsmålinger. For eksempel skal det være mulig å måle bredden på en dør eller et vindu med relativt god nøyaktighet. Ambisjonsnivået var at et absolutt avvik på ca. 10 cm var godt nok. Ideelt sett skal observerte størrelser (GPS/INS) i sensoren også kunne utnyttes i programmet, slik at en eventuell modell kan plasseres, skaleres og roteres riktig i et geografisk datum, med tilfredsstillende grad av nøyaktighet.

Det er viktig å være klar over et par grunnleggende begrensninger mtp. Forsvarets bruk av et slikt system. Forsvaret må ivareta operasjonssikkerheten rundt sine oppdrag. Derfor vil det ikke være aktuelt for Forsvaret å ta i bruk Web-baserte løsninger. Vi vil likevel prøve ut disse for å få oversikt over hvor langt teknologien har kommet. I tillegg vil det i mange tilfeller ikke være mulig for en bruker å være fysisk tilstede på objektet i forbindelse med en fotografering. Med dette mener vi ulik preparering for fotografering og modellering, for eksempel plassering og innmåling av standardiserte referansepunkt (targets). Man ønsker også i mange sammenhenger å holde en viss avstand til objektet samt at det nødvendigvis ikke er mulig å ta bilder fra alle vinkler.

De innledende ideene til løsning av oppgaven fikk vi primært fra vår hovedveileder som tilhører UMB. Dette var informasjon og demonstrasjoner han hadde fått ved deltagelse på diverse fotogrammetriseminar, altså den sivile delen av bransjen. Dette ga oss indikasjoner på at fotogrammetri som utgangspunkt for 3D-modellering var på full fart tilbake og faktisk i stand til å konkurrere med laserskanning som har dominert de siste årene. Dette er sannsynligvis en følge av at maskinvare og algoritmer har hatt stor utvikling og at kostnadene ved bruk av bilder og film har blitt mindre.

Ut ifra bakgrunnen for oppgaven har vi kommet fram til følgende problemstilling:

- Er det mulig å lage 3D-modeller automatisert med utgangspunkt i bilder tatt med enkle sensorer som f.eks. mobilkamera og speilrefleks?
  - Hvilken eksisterende software er i stand til å gjøre dette?
  - Geometri og visualisering
    - Hvilke krav stilles til bildesekvensen for at en rekonstruksjon skal være gjennomførbar?
  - Hvilke metoder og input kan benyttes for skalering og orientering av modellene
  - Hvordan er de ulike softwarene tilrettelagt for å gjøre målinger og hvilken nøyaktighet gir disse?

Fordelingen av arbeidet oss imellom er beskrevet i Vedlegg A.

## 1.2. Tidligere relevante undersøkelser

Det er selvfølgelig gjort en del lignende undersøkelser som til en viss grad samsvarer med våre problemstillinger, og vi vil gi en generell beskrivelse av dette arbeidet.

To publikasjoner har gjort lignende tester på de samme programmene som oss.

Bruksområdene er derimot noe annerledes da man er mer fokusert på kulturarv og arkeologi.

Dette innebærer at testene primært er gjort på mindre objekter enn det vi har sett på, f.eks.

krukker, statuer, enkle fasader på hus osv. I den ene artikkelen som ligner mest på våre

undersøkelser, benytter man laserscanning for å lage en referansemodell av objektene.

Deretter legger man denne og modellen generert av hver av de gjeldende programmene oppå hverandre og ser på forskyvning av detaljer og geometri. Publikasjonen konsentrerer seg også primært om tidsbruk for modellering, nøyaktighet og kvalitet på tekstur. Altså et mye mer snevert innhold enn det vår oppgave vil dekke. Dette skyldes selvsagt at vi har en annen type brukere som målgruppe. Vi har ikke klart å lese noen entydig konklusjon mtp hvilket program som presterer best av de som er testet. De nevner også at grad av nøyaktighet vil være gjenstand for videre undersøkelser. [1]

I den andre publikasjonen sammenligner de ett av de programmene vi har testet, samt noen andre som vi ikke har vært i befatning med eller forkastet i de innledende undersøkelsene. Det ene av disse er Project Photofly som er en tidlig forgjenger av Autodesk 123D-Catch. Her ble det kun testet på tetthet i punkttskyer, dvs. hvor fullstendig modellen blir gjenskapt i punktskyen og hvor mange punkt den inneholdt. De vurderte også prosesseringstid. Her ble det ikke gjort noen endringer i parametere som påvirker resultatene, kun defaultverdier ble brukt. Dette blir nevnt som et eget kriterium for hvordan et ideelt program skal fungere for deres bruk. De fleste programmene som ble testet var nettbaserte. I publikasjonen kommer det fram at de primært er ute etter å utvikle et program som ikke prosesserer på brukerens maskin, men via internett. I denne artikkelen fant vi heller ingen entydig konklusjon som svarer på våre problemstillinger.[2]

Vår generelle vurdering etter å ha lest gjennom ulik litteratur er at det forgår mye forskning og utvikling innenfor dette fagfeltet. Imidlertid anser vi begge disse testene som ufullstendige i forhold til hva Forsvaret ønsker at vi skal finne ut og de bruksområder Forsvaret ønsker at slikt program skal dekke. Årsakene til dette vil vi belyse her. Undersøkelsene beskriver blant annet ikke mulighetene for skalering og måling av modellene eller bruk av observerte størrelser fra f.eks. GPS/INS. Dette er et viktig testmoment for oss, da Forsvaret vil ha tilgang

til bildemateriale fra ulike sensorer som er i stand til å gi slike data. Måling og målenøyaktighet er en viktig del av undersøkelsene da en 3D-modell blant annet vil brukes til planlegging av operasjoner der avstandsdata vil kunne være essensielt. Operasjonssikkerhet er heller ingen faktor i disse publikasjonene og en web-løsning trekkes fram som en ideell løsning. Dette er sannsynligvis ikke noe Forsvaret vil ønske, derfor har vi i større grad gjort tester på programmer som ikke er internettavhengige. Videre er man ikke åpen for at en bruker ved spesielle behov skal kunne justere parametere for hvordan prosessen skal løses. En av våre oppgaver er å finne ut hvor automatisert man kan gjøre en modellering som samtidig tilfredsstillende en del kriterier som at modellen er riktig både geometrisk og visuelt. Dette gjør at det for oss er naturlig å prøve ut eventuelle muligheter for justering av parametere. Aktuelle brukere og bruksområde som publikasjonene er tilpasset til gjør at vi anser det som sannsynlig at våre undersøkelser vil være med å dekke områder som de nevnte publikasjonene ikke vil kunne belyse.

### **1.3.Forutsetninger/Avgrensninger**

Vi forutsetter at de grunnleggende teoriene bak begrepet fotogrammetri er kjent for leseren.

Vi vil ikke gå inn på teorien som omhandler dette. Med dette mener vi f.eks.:

- Sentralprojeksjon
- Kolinearitetsprinsippet
- Fram- og tilbakeskjæring
- Indreorientering
- Ytre orientering
- Grunnleggende bildematching
- Triangulated Irregular Network (TIN)

Vi vil ikke gjøre vurderinger av nøyaktigheten på GPS-data vi har benyttet oss av med unntak av de kvalitetsdata FFI har oppgitt. (Vedlegg C)

Teori og algoritmer for TIN, hulltetting og teksturering vil ikke bli omhandlet i vesentlig grad da dette er noe man har ambisjoner om at skal gjøres automatisk i et ideelt program.

## 2. 3D REKONSTRUKSJON

Fra gammelt av var kart hovedproduktet fra fotogrammetri. Det ble brukt analoge kameraer og instrumenter for å lage produktene man ønsket fra fotogrammetrien. Vi kaller denne typen for flyfotogrammetri. Utviklingen av datateknologien og innføringen av digitale kameraer har endret faget betraktelig, noe som igjen har gitt oss flere produkter. Et av disse er digitale 3D-modeller av terreng, men like gjerne av menneskelagde objekter som hus, bygninger eller byer. Dette er en voksende bransje. Teknikken som nyttes til dette, kalles nærfotogrammetri. En definisjon av begrepet er som følger:

*Gren av fotogrammetrien der fotograferingsavstanden er mindre enn 300 m. Ingen prinsipiell begrensning for valg av kameraorientering og oppstillingssted. [3]*

Det som skiller nærfotogrammetri fra vanlig flyfotogrammetri, er med andre ord avstand fra objektet, kameraets plassering og objektets form. Objektet er nært, bildene er tatt fra forskjellige posisjoner (ikke bare loddrett) og objektene er mye mindre. Definisjonen av nærfotogrammetri sier at avstanden skal være mindre enn 300 m. Med dagens kvalitet på kamerasystemer vil vi påstå at denne kan økes betraktelig.

Likhetene mellom flyfotogrammetri og nærfotogrammetri er mange. Bildene må orienteres i forhold til hverandre. Ytre orienteringsparametere må finnes. Bildekoordinater må etableres. Flyfotogrammetri, med loddrette bilder, er på enkelte måter enklere enn nærfotogrammetri i og med at orienteringsparameterne gir seg selv<sup>1</sup>. Slik er det ikke i nærfotogrammetri. I tillegg har gjerne nærfotogrammetri varierende bildestørrelser og store variasjoner i geometri og radiometri.

Den matematiske modellen er den samme for begge typene. Det er sentralprojeksjonen og kolinearitetsprinsippet som gjelder for begge. Posisjon og form på et objekt bestemmes ved at strålebunten fra objektet gjenskapes. Fra hvert punkt i bildet kan det trekkes en rett linje gjennom projeksjonssenteret til det samme punktet i terrenget eller på objektet. Hvis vi har et kalibrert kamera og vet posisjonen på kameraet i forhold til objektet, kan strålebunten plasseres i forhold til objektet. For å klare å plassere et punkt i rommet trenger man 2 bilder (stereofotogrammetri). I prinsippet kan man bruke ubegrenset antall bilder (multi-image photogrammetry). [4]

---

<sup>1</sup> Gode foreløpige verdier for posisjon og rotasjon. Flyet flyr tilnærmet plant med ellipsoiden. Dette gir en dreining på  $\approx 0^\circ$  for phi og omega. Kappa kan beregnes ut i fra koordinatene på flyet under fotograferingen.



Utviklingen innen maskinvare har vært enorm de siste 10 årene. Dette har influert på fotogrammetribransjen ved at algoritmer og teknikker som muliggjør automatisering utvikles raskt. I dag er det gjennom gode algoritmer for automatisert bildematching, at det er mest å hente hva angår nøyaktighet og tidsbruk. Disse er under stadig utvikling. Mengden punkter kan være enorm, men dagens maskinvare er i stand til å håndtere en slik datamengde. Hver eneste piksel i et bilde kan matches mot piksler i andre bilder.

Fotogrammetribransjen og databransjen (computer vision) har begge jobbet med 3D-modellering. Dette har skjedd uavhengig av hverandre. Innen fotogrammetri brukte man analoge apparater og bilder med mye menneskelig interaksjon i prosessen. Målet var absolutt bestemmelse av punkter i bildene. Databransjen har samtidig jobbet med bilder med lav oppløsning, med fokus på relativ posisjonering av objekter. De siste 10-15 årene har vi hatt tilgang på billige og enkle kameraer med høy kvalitet. Tilgangen på kameraer med bra bildekvalitet, mulig for fotogrammetrisk bruk, ble nå mye bedre. Ideen om å ta det beste fra både fotogrammetri- og databransjen ble født. Man begynte å utvikle verktøy som automatisk og med høy nøyaktighet ville kunne gjennomføre 3D-modellering med bilder tatt av enkle kameraer.[5]

Utviklingen av digitale kameraer gav en enorm mengde bilder med stadig større oppløsning. Rundt år 2000 var ikke bransjen i stand til å håndtere slike store mengder data, noe som medførte at nye algoritmer måtte utvikles og etableres. Det var i hovedtrekk 3 utfordringer som måtte løses. Disse var som følger [5]:

1. Automatisk klare å finne sammenbindingspunkt mellom 2 bilder. Dette måtte gjøres med stor nok tetthet og bra nok fordeling slik at disse kunne nyttes til en orientering av bildene.
2. Orientering av store blokker med bilder med automatisk genererte sammenbindingspunkt. Store deler av disse algoritmene var kjent, men utfordringen var å få dette til i en tilfredsstillende software.
3. Forbedringen av algoritmene som gjennomførte en *dense image matching* av orienterte bilder.

Proessen fra bildene av et objekt er tatt til en ferdig 3D modell er etablert, kalles en 3D rekonstruksjon. Det er mange steg som skal gjennomføres i en slik prosess. Basert på det vi har funnet i litteratur og nærmere studier av programmene har vi kommet fram til følgende liste:

1. Innlesning av bilder
2. Deteksjon av sammenbindingspunkt
3. Matching av sammenbindingspunkt
4. Orientering av bildene med spredt punktsky
5. Generering av tett punktsky
6. TIN- Triangulated Irregular Network
7. Legge på tekstur

Institut Géographique National (IGN) har beskrevet prosessen med 3 moduler. Disse er som følger [6]:

1. Beregning av sammenbindingspunkt.
2. Beregner ytre og indre orienteringsparametere for kameraene. Dette kan også inkludere data fra GPS/INS og bakkepunkter (Ground Control Point-GCP).
3. Generering av nøyaktig 3D modell basert på modul 2.

M. Pierrot Deseilligny har gjennomført forsøk på 3D rekonstruksjon med disse modulene. Han bruker SIFT til modul 1, APERO til modul 2 og MICMAC til modul 3. [5] Vi kommer tilbake til SIFT senere.

Tabell 1 under viser sammenhengen mellom ”vår” prosess og IGN’s.

Innlesning av bilder	Beregning av sammenbindingspunkt
Deteksjon av sammenbindingspunkt	
Matching av sammenbindingspunkt	
Orientering av bildene med spredt punktsky	Beregner ytre og indre orienteringsparametere for kameraene
Generering av tett punktsky	Generering av nøyaktig 3D modell
TIN- Triangulated Irregular Network	
Legge på tekstur	

**Tabell 1** Prosedyrer for automatisk 3D-modellering

Med bakgrunn i den stadige utviklingen som er innenfor dette faget, ser vi oss nødt til å gå i dybden på noen utvalgte deler av prosessen der den normale skolelitteraturen i dag kan synes noe utdatert og mangelfull. Dette skyldes at det hele tiden stilles større krav til ytelse og hastighet fra brukerne og at metoder og algoritmer konstant utvikles og forbedres. Når det er

sagt, gjelder selvsagt de generelle grunnprinsippene fremdeles. De tre hovedemnene vi tar for oss er:

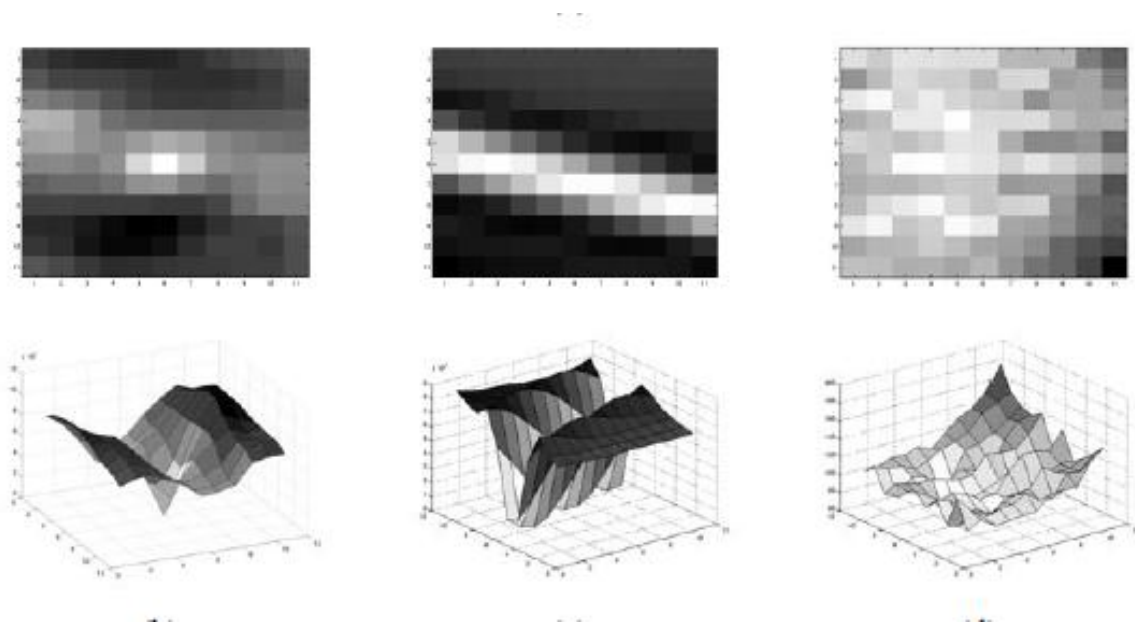
- Deteksjon av sammenbindingspunkt (key-features)
- Structure from Motion (SFM, orientering av bilder)
- Bildematching

Dette er de emnene der det er mest framtreddende forskjeller og utvikling i forhold til hva vi har gått gjennom i løpet av vår studietid. I tillegg er det disse emnene som utgjør hovedessensen i automatisk 3D-rekonstruksjon.

## 2.1. Deteksjon av sammenbindingspunkt

*Feature detection* er en svært viktig del av 3D-modelleringen. Objektene som finnes nyttes både til orientering av bildene og til å sy sammen bildene til en mosaikk i ettertid. Slik vi forstår dette så er resultatet av en *feature detection* et sett med mulige sammenbindingspunkt. I beskrivelsen videre av denne teorien bruker vi også begrepet *keypoints*.

De første objektene vi gjerne ser etter, er de med spesiell plassering i bildene. Dette kan for eksempel være fjelltopper, hushjørner, vinduer eller dører, men mest sannsynlig punkter hvor kontrastforskjellene rundt punktet er unikt. Det er viktig å være klar over at dette er objekter som skal finnes automatisk av et program og ikke manuelt av et menneske. Dette kommer godt fram på Figur 14 side 41. Objektene på denne er funnet av en matematisk algoritme, ikke et menneske. Likevel er disse unike slik at det kan gjennomføres en matching mellom bilder. Objektene må være lette å finne igjen i flere bilder. Slike objekter kan gjerne kalles *keypoint features* [7].



Figur 1 Eksempler på *feature detection* [7]

Det er 2 forskjellige metoder å finne *keypoints* på. Den første finner punkter i det ene bildet og bruker deretter algoritmer som korrelasjon og minste kvadraters metode og klarer ved hjelp av disse å plassere punktene nøyaktig i forhold til hverandre. Metoden er mest anvendbar når vinklene er små mellom bildene. Den andre metoden finner *keypoints* i begge bildene og så matches disse i forhold til hverandre. Denne er mest anvendbar når det er stor forskjell mellom bildene, for eksempel *stitching* til panoramabilder eller i objekt gjenkjenning. [7]

R. Szeliski beskriver 4 steg i prosessen:

1. *Feature detection*. Det gjennomføres et søk gjennom hvert bilde etter punkter som det er sannsynlig at man vil finne igjen i andre bilder. Dette søket vurderer endringer i gråtoneverdier inne et søkevindu. Er endringen (gradienten) stor, kan det være mulig å finne igjen punktet i et annet bilde. Er overflaten homogen, vil det være vanskelig å finne punkter i noen av bildene. [7]
2. *Feature description*. *Keypoints* vil ikke være helt like i et bildepar. I et bildepar vil det kanskje være endringer i både orientering og skalering. Det kan også ha gjennomgått en affin transformasjon. Det vil derfor være nødvendig å uttrykke disse parameterne og endre innholdet i søkevinduet slik at det passer med de andre punktene. Områdene rundt hvert punkt blir med andre ord konvertert til kompakte og stabile benevnelser som senere kan matches med andre *keypoints*. [7]

Det finnes flere typer *descriptors*. Vi vil ikke beskrive alle disse, men programmet SFM Toolkit bruker en som heter *Scale invariant feature transform* (SIFT).

Algoritmen er beskrevet av David G. Lowe. Etter at punktene er funnet, gjennomføres en filtrering av dårlige *keypoints*. Deretter fjernes effektene av rotasjon og skalering før selve beskrivelsen lages. Dette gjøres ved at de lokale gradientene beregnes i området rundt punktet. Lengden på disse legges så til i et histogram, som blir en del av beskrivelsen. Et punkt beskrives av nabopunktens gradienter i form av plassering, orientering og skalering. Denne informasjonen lagres i en database. Videre brukes denne databasen som en referanseliste slik at når man tar inn et nytt bilde der den samme prosessen gjennomføres, vil man i databasen kunne finne punkt med beskrivelse som ligner. Her nyttes også korrelasjonsverdier med en viss terskelverdi. [8]

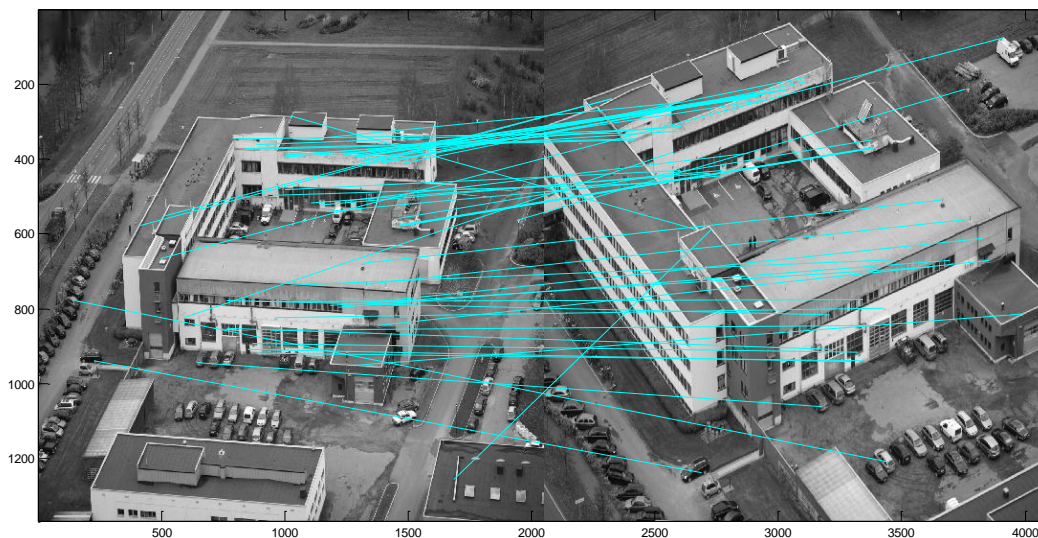
3. *Feature matching*. Når det er funnet *keypoints* i 2 eller flere bilder, starter matchingen. Først må metode velges, deretter en effektiv algoritme. Metode velges gjerne avhengig av hva som skal oppnås. Forskjellen dreier seg gjerne om hvor mange punkter man antar vil matche mellom bildene. I en stiching vil det være relativt mange som matcher, hvis man skal finne et objekt i et stort bilde, vil det være få punkter som matcher. Vi vil ikke beskrive disse algoritmene her.

Etter at matchingsmetode er valgt må man finne en effektiv metode for søket. Den enkleste metoden er å sammenligne hvert eneste punkt med alle andre. Men antallet søk blir da kvadratisk i forhold til antallet punkt. Det finnes flere datastrukturer og algoritmer for å gjøre dette søket enklere og mere effektivt. Hensikten med disse er å begrense søketiden så mye som mulig. [7]

4. *Feature tracking*. Dette er et alternativ til steg 3. Forskjellen er at det søkes i et lite område rundt hvert *keypoint*. Dette gjør prosessen raskere. Passer bra i forbindelse med bruk av videobilder. [7]

Kantdeteksjon er en annen teknikk som nyttes som en del av *feature detection*. Kantene, for eksempel taket på et hus eller fjellprofilen mot himmelen, kan brukes til matching på grunn av deres orientering og lokale profil. Disse kan også nyttes til å finne objektene avgrensninger. Der linjen stopper, stopper gjerne objektet også, for eksempel et hustak. [4]

Når sammenbindingspunkt er funnet i de forskjellige bildene, matches disse mot hverandre. På denne måten finner man igjen samme punktet i flere bilder og man har da et utgangspunkt før bildene skal orienteres.



Figur 2 TF/IMT prosessert med SIFTGPU, to bilder med felles sammenbindingspunkt. Merk at noen er tydelig feil matchet.

## 2.2. Orientering av bilder / "Structure from Motion"

Innen "Computer-vision"-bransjen benytter man "Structure from motion" som uttrykk for orienterings og selvkalibreringsprosessen. "Structure" betegner 3D-egenskapene til et 3D-punkt, mens "motion" kommer av kameraposisjonene og deres egenskaper [7]. I de tilfellene vi har vært mest innom i vår oppgave, er både punktposisjonene og kameraposisjonene ukjente, altså en relativ orientering.

Utgangspunktet for orienteringen av bildene, er de mulige sammenbindingspunktene som er beskrevet i forrige kapittel. Som vi ser av figur 2 så er det en del av disse som er feil. Disse punktene vil få store residualer under prosessen og vil bli filtrert bort.

I de algoritmene vi har sett på benyttes det en vinkel mellom bildene som er mye mindre enn det som har vært vanlig i de sammenhenger vi har erfaring fra. Dette er de mer manuelle metodene vi har erfaring fra. Samtidig benyttes det opp mot 80 % overlapp. De små vinklene

vil kunne gi en dårlig geometri hvis man prøver å orientere bildene på tradisjonelt vis. Dette vil ikke gjøres gjeldende siden vi benytter en såpass stor overlapp og algoritmer som finner ekstremt mange sammenbindingspunkt, godt spredt ut i bildene. I tillegg matches hvert bilde mot bilder med både større og mindre vinkelforskjeller.

En viktig del av fotogrammetrien er å finne orienteringsparametere på sensoren, ytre og indre orientering. Indre orientering gjennomføres i et testfelt eller som selvkalibrering. De fleste programmene vi har brukt, har egne moduler som gjennomfører selvkalibrering og benytter parameterne i de videre prosessene. I vanlig flyfotogrammetri brukte man tidligere kjente signalerte punkter på bakken som utgangspunkt for beregningen av de ytre orienteringsparameterne. Ved innføringen av GPS og IMU har dette ikke vært nødvendig i like stor grad selv om slike punkter fortsatt nyttes for kontroll.

I nærfotogrammetri er det tradisjonelt større variasjon i kvaliteten på utstyret som nyttes. Kamerasystemene kan være kostbare med svært liten deformasjon eller billige vanlige kameraer med stor deformasjon. Enkelte prosjekter gjennomføres med GPS/IMU, men langt i fra alle. Kvaliteten på disse er som regel heller ikke på samme nivå som ved flyfotogrammetri. Dette gjør orienteringsprosessen i nærfotogrammetri svært viktig.

Utgangspunktet for orientering av sensoren er sentralprojeksjonen.

### **2.3.Dense image matching**

Bildematching nyttes flere ganger i en 3D-rekonstruksjon. Innledningsvis brukes denne for å finne bildenes ytre og kameraenes indre orienteringsparametere. Dette resulterer i en spredt punktsky. Deteksjon av sammenbindingspunkt er en viktig del av dette. Deretter nyttes bildematchingen når den tette punktskyen skal genereres. Dette er det som defineres som *Dense image matching*.

I utgangspunktet skulle det være enkelt å matche 1 punkt som samsvarer mellom to bilder. Men så enkelt er det gjerne ikke. Hellinger og dybdeforskjeller i bildene kan i tillegg til kameraets indre orientering gi geometriske feil i modellen/matchingen. I tillegg vil vi kunne få radiometriske feil som skyldes ulik belysning og refleksjoner. Disse vil videre resultere i feil i geometrien. Støy knyttet til sensoren og sensorens GSD vil gi både radiometriske og geometriske feil. [9]

I utgangspunktet finnes det flere metoder å gjennomføre bildematching på. I litteraturen vi har hatt befatning med beskrives primært to typer:

- Arealbasert metode
- Objektbasert metode

Den arealbaserte metoden sammenligner et areal (filter/vindu) i et referansebilde med areal av samme størrelse i andre bilder. Filteret/vinduet bruker gråtoneverdiene i bildene og sammenligner informasjonen for å finne samsvarende områder i bildene. Til å gjøre dette brukes to forskjellige algoritmer, den første baserer seg på Minste Kvadraters Metode, den andre på korrelasjon. [4]

Den objektbaserte metoden er den mest brukte metoden i en automatisert 3D-rekonstruksjon. På grunn av at teorien bak denne er svært lik deteksjon av sammenbindingspunkt i Kapittel 2.1, så vil vi ikke beskrive denne mer her.

Når vi nå har et sett med orienterte bilder samt kamerakalibreringsdata, er det mulig å gjenskape en scene ved hjelp av nye automatiske bildematchingsteknikker. Punktskyene har tilstrekkelig tetthet (tett punktsky) til å beskrive overflaten med uregelmessigheter. Punktskyen må tilpasses slik at tettheten er stor der det er nødvendig (uregelmessigheter) og liten der det er flater. De beste teknikkene i dag er følgende: [10]

- *Multi image matching technique*. Også kalt *Semi-global matching (SGM)*.
- *Patch-based methods*. Implementert i programvaren PMVS2.
- *Optimal flow algorithms*. Implementert i programvaren MICMAC.

I de neste kapitlene vil vi beskrive disse metodene. Vi har prioritert Patch-based methods da det er denne metoden som er inkludert i SFM Toolkit og Visual SFM, to av de programmene vi har fokusert på. Vi vil ikke vurdere metodene opp mot hverandre, da dette ikke er relevant for vår problemstilling.



### 2.3.1. Semi-Global Matching

Et navn på algoritmer som tar hensyn til kostnadene og begrensningene ved en matching, og prøver å minimalisere disse, er *Global Image Matching*. Problemet med slike algoritmer er ytelsen. For å løse dette ble det utviklet en ny algoritme; *Semi-Global Matching* (SGM). SGM baserer seg på piksel mot piksel matching hvor gjensidig informasjon vurderes. SGM bruker teknikken *Mutual Information* for å minimalisere kostnadene ved matchingen. *Mutual Information* ble i utgangspunktet oppfunnet for å måle likheter mellom bilder. Målet var å finne en transformasjon mellom 2 bilder som maksimerte den gjensidige informasjonen mellom bildene. Fordelen med *Mutual Information* er at algoritmen håndterer intensitetsforskjeller mellom bilder. [11] [12] [13]

SGM gjennomfører først en form for segmentering av bildene der entropi dvs variasjon i tekstur eller pikselverdier har stor betydning. Her deles det inn i segmenter/felt som inneholder en viss entropi. Dette er for å kunne få opp hastigheten ved at man i et bildepar kun kjører matching mellom segmenter med en viss likhet i entropi. Man kan si at vi benytter en slags vektning eller en koeffisient som beskriver graden av entropi. Det kan være vanskelig å si nøyaktig hvor kantene i disse segmentene befinner seg. Derfor legger man på en form for ”padding” eller overlapp slik at disse kantsonene blir med gjennom flere segmentsdelte matchinger. [11, 13]

Algoritmen benytter *disparity maps*, det vil si et bilde hvor forskjellene mellom 2 matchede bilder kommer fram etter en piksel mot piksel matching. Dette bildet benyttes av algoritmen i en pyramideprosessering der man starter på f.eks. 8:1 i oppløsning og jobber seg opp til full oppløsning. I 1:1 oppløsning gjennomfører man en piksel per piksel matching. [11]

### 2.3.2. Patch based methods

Vi velger å beskrive hvordan programmet PMVS2 benytter denne metoden da PMVS2 er et program som inngår i to av hovedprogrammene som vi har vært i befatning med gjennom våre undersøkelser. PMVS2 har som oppgave å generere den tette punktskyen. Beskrivelsen tar utgangspunkt i en artikkel [14] skrevet av utviklerne av PMVS2. Det som er viktig å ha i bakhodet når man skal sette seg inn i hvordan dette fungerer, er at PMVS2 starter med et bildesett som allerede er orientert i forhold til hverandre. I tillegg er det gjort en oppretting av radiellfortegning slik at bildene ikke innehar deformasjoner, forårsaket av optikk.

PMVS står for Patch-based Multi View Stereo og typen algoritme som ligger bak kan derfor leses av navnet.

Første steg er key feature detection der framgangsmåten er tilnærmet lik SIFT. Deretter gjennomføres en matching av disse, der man som resultat av en vellykket matching prøver å generere en patch der dette punktet/pikselet utgjør senter av patchen. Algoritmen som gjør disse to trinnene, er Harris i tillegg til en "Difference-of-Gaussian"-operator.

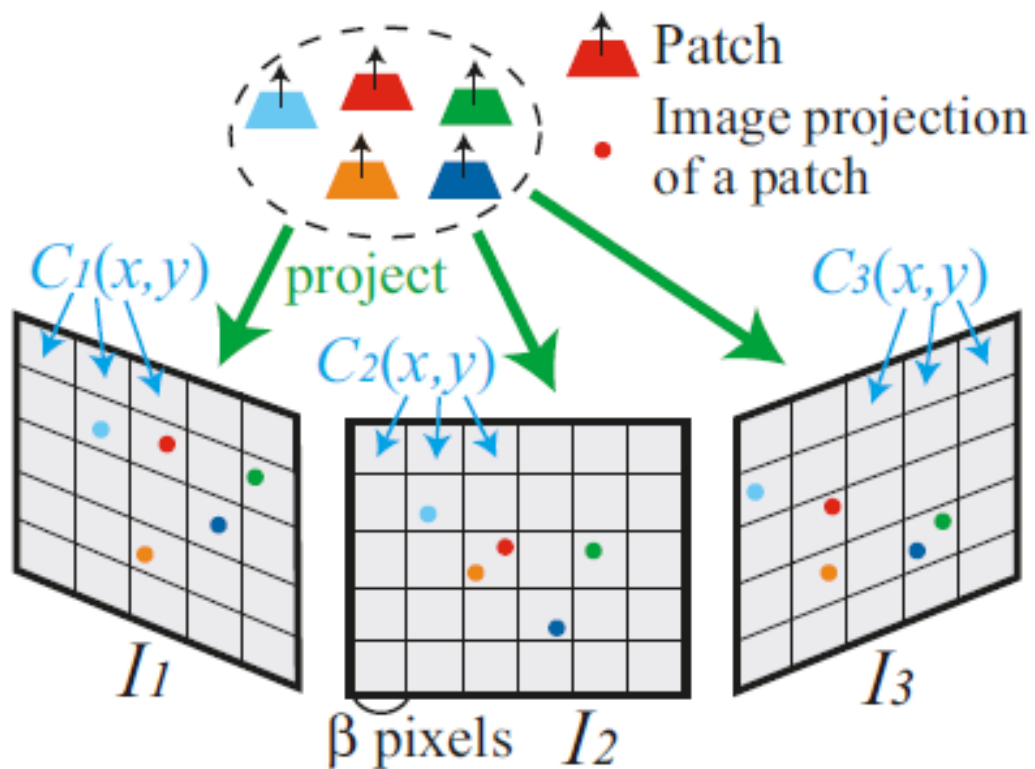
Harrisdetektoren, ofte kalt hjørnedetektor, benytter seg av gradientene rundt en feature for å beskrive denne. [14]

Hvis man tenker at en feature fra referansebildet skal finnes igjen i et annet bilde, finner man lignende features som ligger maksimalt to piksler fra epipolarlinjen og gjør en triangulering av 3D punktene som knyttes til hver av disse. Når algoritmen finner en kombinasjon som tilfredsstillende terskelverdier gjøres dette punktet til senterverdien i en patch. For å sikre seg at dette ikke er en falsk match, benytter man synlighets parametere, dvs. en slags sannsynlighet for at det punktet som er matchet, faktisk er synlig mellom bildene som har bidratt til å generere punktet. I PMVS2 benytter man en thresholdvinkel (terskel) mellom patchens enhetsnormalvektor og retningen fra patchens senterpunkt og til kameraet som skal matches mot denne. Denne vinkelen er i PMVS2 satt til  $\pi/3$  og gjør at kun vinkler under denne verdien godkjennes. Kort oppsummert etableres en patch der man har en fremtredende variasjon i tekstur, altså markerte forskjeller i fargeintensitet og man har klart å finne match mellom flere bilder der terskelverdien for synlighet tilfredsstilles. [14]

En patch er en rektangulær 3D-flate som på en måte tangerer punktet som beskrives og kan ses fra alle kamera som er matchet til denne. Patchens størrelse i PMVS2 er 7x5 piksler. Slik kan man sample fargeverdier som ligger innenfor patchen for å finne en verdi for senterpunktet som er så riktig som mulig. Enhetsnormalvektoren for patchen peker mot det

som algoritmen setter som referansekamera. I hvert bilde vil det genereres en stor mengde patcher. Disse er tett plassert utover alle flater som er synbare i bildene og gjenspeiler pikselintensiteter i framtrede regioner i bildene. [14]

For hvert bilde som er med i prosessen deles disse inn i et kvadratisk rutenett med celler av størrelse  $n \times n$ -piksler. Dette rutenettet vil være orientert likt som bildet. Målet er å generere minst en patch i alle celler for alle bilder. Patcher som er synlig for et bilde lagres i dette rutenettet og tas med videre i prosessen for å gjøre de neste stegene mindre tidkrevende.

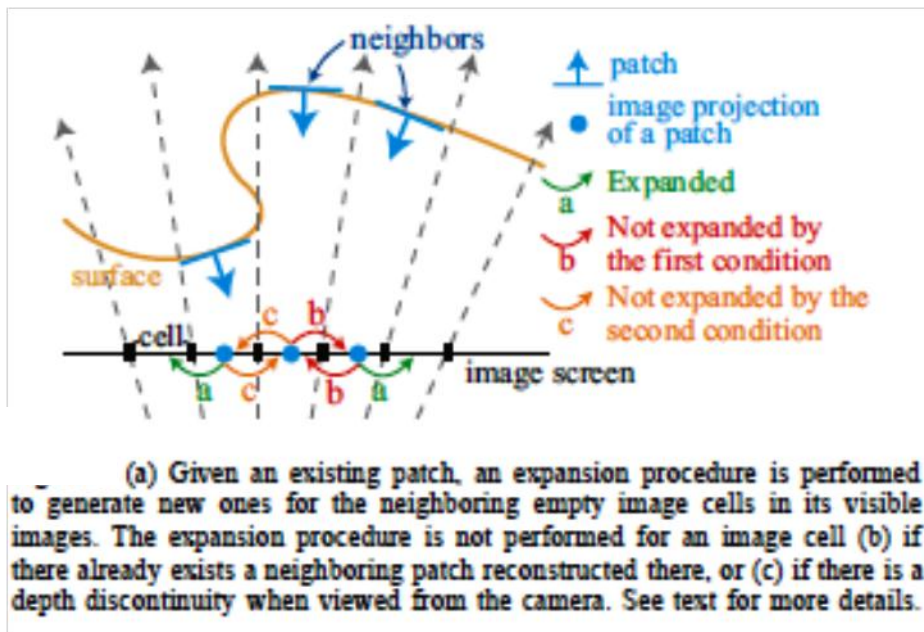


Figur 3 Visualisering av celler og patcher[14]

Patcher som matcher hverandre tas med videre og uttrykkes som punkter i modellen. Dette gir en sky som initielt framstår som en spredt punktsky. Man har mest sannsynlig ikke funnet en patch per celle etter engangs gjennomkjøring, derfor må prosessen itereres. Samtidig kan man finne en del falske match her til tross for bruken av synlighetsparametere. Denne ”skyen” av patcher gir grunnlaget for de to neste stegene som er ekspansjon og deretter filtrering av patchene, noe som vil fjerne det meste av falske punkt. [14]

Ekspansjonen har som oppgave å generere patcher for nabocellene til de allerede eksisterende patchene eller flere innenfor den samme cellen. Hvis korresponderende celler mellom to

bilder har en dybdeforskjell som gjør det unaturlig at det finnes et felles punkt, vil det ikke gjøres en ekspansjon av en ny patch her. Da går man over til neste nabocelle. Hvis ekspansjonen genererer en patch som er lik en allerede eksisterende patch i nabocellen, vil denne ikke bli tatt med. Hvis patchen er av en annen type/beskrivelse, vil den kunne tas med. Slik vil man kunne gjengi små detaljer i objektets overflate. Algoritmen benytter også et dybdekart for å kunne ekspandere patchene riktigst mulig i modellen. Det vil si at man ikke ekspanderer til en nabocelle hvis overflaten som er projisert der har en stor dybdeforskjell i forhold til nabocellen.[14] Vår vurdering er at dette er fornuftig, da faktorer som bl.a. lysforhold kan gjøre at en slik flate gjengis forskjellig eller feil ved dybdeforskjeller i det virkelige objektets overflate (Figur 4). Ved at man benytter disse funksjonene, vil man også kunne gjengi homogene flater med en høy grad av presisjon. Artikkelen [14] inneholder en meget god illustrasjon med forklaring hvordan ekspansjonen fungerer.



Figur 4 Eksempel på ekspansjon[14]

Filtreringen har som oppgave å fjerne patcher som blir matchet uten å være korrekte og å fjerne uteliggere (outliers). Slik vil man sikre bedre bestemmelse av punktet ved at geometrien blir bedre ivaretatt. Algoritmen benytter tre filtre. To som går på synlighetsinformasjon der den første av dem filtrerer vekk patcher som ligger i samme celle som en annen hvis denne er av en annen type (ikke nabopatch) og er synlig av et lite antall bilder. Den andre fjerner patcher som er generert av et for lite antall bilder enn det algoritmen setter som krav. Det siste filteret detekterer og fjerner patcher som har et mindre antall nabopatcher enn en viss terskelverdi. [14] Slik fjerner filtrene det som vil oppfattes som f.eks.

et enslig grått punkt midt på en helt hvit vegg eller enslige punkter som henger midt i luften osv.

Ekspansjonen og filtreringen itereres tre ganger slik at det dannes flest mulig patcher som til slutt vil utgjøre 3D-punktene i skyen. På denne måten er programmet i stand til å gjenskape veldig små detaljer uten noen særlig grad av støy. [14]

### **2.3.3. Optimal flow algorithms**

Vi har slitt med å finne en litteratur som beskriver denne algoritmen, men velger likevel å ta med noe av det vi har funnet.

Slik vi forstår algoritmen så er det en matematisk metode hvor en eller flere størrelser skal minimaliseres. I matchingssammenheng er det funksjonen for energi som skal minimaliseres. Algoritmen kan håndtere forskjellige bildestørrelser. Den bruker en pyramideprosessering som gjør at prosessen går raskt. Mellom hvert steg forsikrer man seg om at de matchede punkter er de samme. Brukeren må plukke ut et «master» bilde som nyttes videre. I dette bildet blir det plukket ut segmenter eller patcher som finnes i nabobildene. Videre nyttes en minimaliseringsfunksjon for å håndtere endringer i bildet. Algoritmen er inkludert i programmet MICMAC. [15]

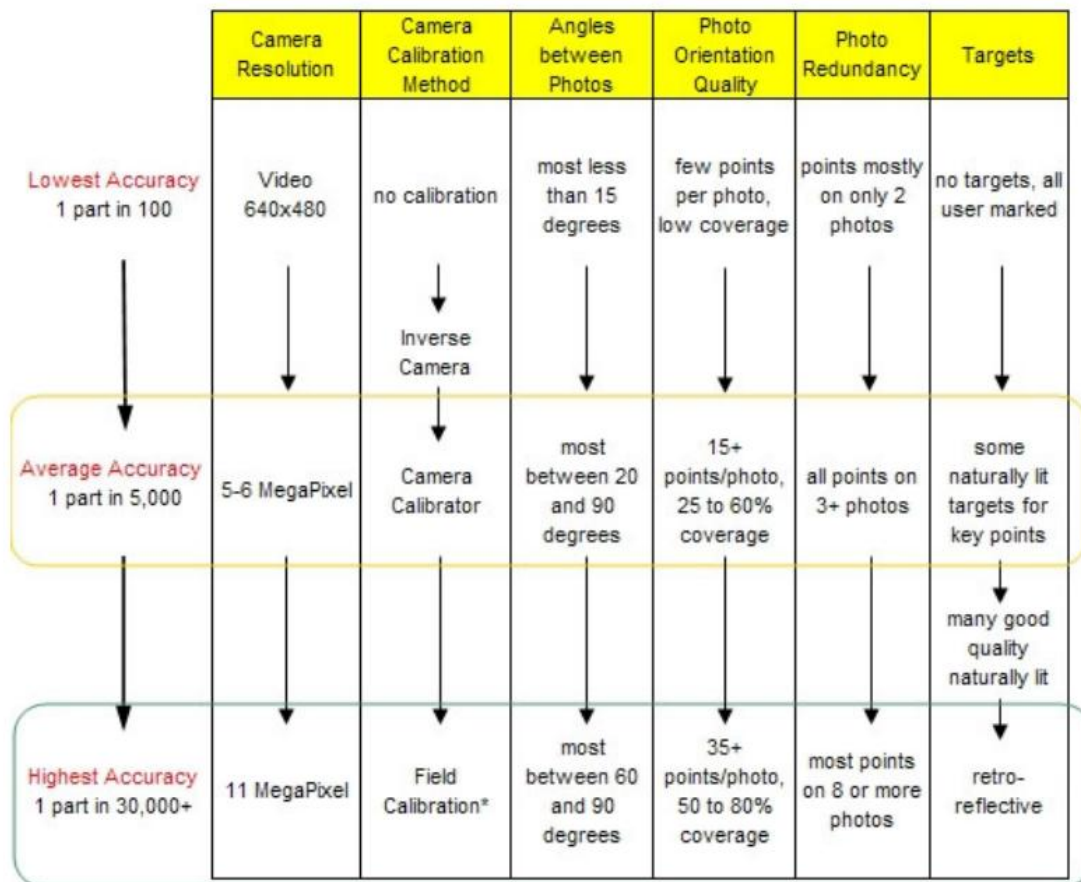
Algoritmen har vært prøvd sammen med SIFT, med stor suksess. [5]

## 2.4. Faktorer som påvirker en 3D-rekonstruksjon

Det er flere faktorer som innvirker på hvor bra en 3D-rekonstruksjon blir gjennomført. For vår oppdragsgiver vil det være viktig å være klar over disse faktorene før fotograferingen gjennomføres. PhotoModeler har beskrevet faktorer som påvirker nøyaktigheten på modellene. Dette er faktorer som vil være gjeldende også for andre programmer [16]:

- **Bildets oppløsning.** Ved høyere oppløsning er det større sjans for å oppnå en nøyaktig plassering av objekter. Dermed også bedre nøyaktighet. Ved dårlig oppløsning vil det være vanskeligere å finne gode sammenbindingspunkter samtidig som modellens oppløsning blir tilsvarende lav og små detaljer utelates.
- **Kamerakalibrering.** Deformasjoner i bildet vil gjøre det vanskelig for orienteringsalgoritmen å finne riktig plassering i bildene som matches samtidig som punkter kan få unøyaktig plassering i modellen.
- **Vinkel mellom bilder.** Ved for stor vinkel mellom bildene vil man få en dårlig romlig bestemmelse av hvert punkt. Dette er på grunn av at vinkelen gir få sammenbindingspunkter som igjen gir dårlig orientering av bildene. I tillegg gir en slik vinkel dårlig forutsetning når den tette punktskyen skal genereres. Er vinkelen for liten, får man tilsvarende dårlig geometri for bestemmelse av punktenes posisjon. Dette vil i tillegg påvirke selve orienteringsprosessen negativt.
- **Bildets orienteringsnøyaktighet.** Hvis orienteringen av bildene ikke er riktig, vil dette bidra til dårlig nøyaktighet av modellen. Orienteringen blir bra når det er et stort antall av nøyaktige sammenbindingspunkt mellom bildene. Disse bør i tillegg være spredt ut over bildene slik at størst mulig del er dekket. Man vil også oppleve å få lite punkter eller deformasjoner i modellen hvis den relative orienteringen er dårlig. Rett og slett fordi punktene vil plasseres feil i forhold til hverandre og derfor filtreres bort.
- **Antall bilder i beregningen.** Jo flere bilder et punkt er markert i, jo mer nøyaktig vil dette punktet plasseres. 2 bilder er et minimum, men det vil i utgangspunktet gi dårligere nøyaktighet enn ved bruk av flere.
- **Mål (targets).** Nøyaktigheten på hvert punkt avhenger av presisjonen på punktets plassering i bildene. Ved å plassere targets på objektet før fotografering vil algoritmen få hjelp til å finne sammenbindingspunkt mellom bildene. Samtidig er disse godt definerte både i bilder og modell og vil derfor ha andre bruksområder. For Forsvaret vil bruk av Targets ikke være mulig i de aller fleste tilfellene.

PhotoModeler har laget en skjematisk framstilling av denne.



Figur 5 PhotoModelers faktorerer for nøyaktighet

Ved å forflytte seg ned en rad vil det gi økt nøyaktighet.

Vi mener variasjon i tekstur i bildene i tillegg er en faktor som må tas med. Hvis teksturen er svært homogen vil det være vanskelig å finne bra sammenbindingspunkt i det området.

Det er viktig å skille mellom nøyaktighet i plasseringen av punktene og kvalitet på en 3D-modell. Et av de mest ønskelige egenskapene med en 3D-modell er full dekning, dvs. at denne ikke har hull i overflaten. For å oppnå dette, må man ha en tett punkttsky, som igjen krever liten vinkel mellom bildene.

Med bakgrunn i egne erfaringer vil vi beskrive de faktorer som vi har identifisert, som må tas med i betraktning når man skal gjennomføre fotografering eller velge ut bilder før en 3D-modellering. Det første er at punkter eller områder på et objekt som ikke er avbildet i to eller flere bilder, ikke vil bli med i modellen. Samtidig må det være nok overlapp mellom bildene for å få orientert dem godt nok, relativt til hverandre. Videre må man tenke at man skal gjøre

det mulig og ”knytte” bilder som avbilder områder utenfor senterområdene til de to første bildene, sammen med minst et av de to første. Dette er spesielt en utfordring der man runder hjørner. Slik fortsetter fotograferingen til man har dekt det området man ønsker å modellere. Det gir også bedre resultater hvis man tar bilder fra varierende avstander slik at man har noen oversiktsbilder som dekker hele objektet. Disse vil kunne benyttes for å orientere resten av bildene bedre da de dekker områder som er avbildet i alle bildene. Bildene må også ha så like radiometriske forutsetninger som mulig. Dette kan man etterjustere i en fotoeditor. Videre er det også viktig at man benytter seg av så kort lukkertid som mulig, evt. stativ, slik at man ikke får bevegelse i bildet. Bevegelse gir støy i bildet som videre gir dårligere resultat da detaljer i bildet blir dårligere definert.

Både PhotoModeler og PhotoSynth har beskrevet hvordan fotograferingen bør gjennomføres for å få best mulig 3D-modell. [17] [18]



### **3. METODE**

Ut fra bakgrunnen for oppgaven og vår problemstilling kan vi trekke ut følgende testkriterier for programmene.

1. Grad av automatisering
2. Ivaretagelse av geometri og tekstur
3. Skalering av 3D-modell
4. Utnyttelse av observerte størrelser
5. Hvilken nøyaktighet kan oppnås

Punkt 1 til 4 vil vi primært evaluere ut fra erfaringer gjort underveis. I punkt 2 vil vi i tillegg vurdere kameraposisjonene numerisk. Punkt 5 vil bli målt og beregnet numerisk.

Vi vil også beskrive andre observasjoner og erfaringer som vil kunne være interessante for vår oppdragsgiver.

#### **3.1.Fotografering**

Vi gjennomførte en fotograferings økt ved hjelp av helikopter, der vi fløy over UMB-Campus. Vi hadde hovedfokus på tre fremtredende bygg, men fotograferte også en del generelt på området. Totalt ble det brukt tre speilreflekskamera av typen Canon EOS 5D MKII, Canon EOS 400D og Sony DSLR A300. Det ble flydd og fotografert rundt hvert objekt til vi var sikker på at vi hadde nok bilder. Fotograferingen ble det ikke gjennomført slik at den tok hensyn til militære faktorer. Det vil si operasjonelle hensyn av taktiske årsaker.

Vi har også benyttet oss av noen bildesett som er hentet fra FFIs eget materiale. Disse settene dekker et område på Lørenskog og et ved Sjursøya.

#### **3.2.Software**

Vi benyttet internett til å lete opp eksisterende software som kunne dekke vårt behov samt at vi fikk en del tips om aktuell software (Bundler) via UMB.

Vi endte opp med ”SFM Toolkit”, ”Autodesk 123D-Catch” og en del andre småprogram. Fra før hadde vi litt kjennskap til ”PhotoModeler” og fikk anskaffet lisens på ”PhotoModeler Scanner”.

Vi prøvde oss litt fram med alle programmene, men delte etter hvert programmene oss i mellom. Jørgen fokuserte primært på SFM Toolkit (SIFT, Bundler og PMVS2) og Stein Arne

på PhotoModeler Scanner og Autodesk 123D-Catch, da disse innledningsvis i prosessen framsto som de mest aktuelle.

Samtidig lette vi begge etter eventuelt andre programmer og disse ble testet i kortere perioder før de enten ble forkastet eller satt på vent. Disse var

- 3DSOM Pro
- TOPCON Image Master
- Agisoft Photoscan
- Smart3Dcapture

Noen av de som ble satt på vent, var lisensbelagte og ville derfor ikke være tilgjengelige for oss innenfor vår tidsplan. I disse tilfellene er nødvendig data for videre undersøkelser av disse beskrevet helt til slutt i rapporten. Andre program hadde en såpass kompleks installeringsprosedyre at vi ikke fikk dem til å fungere. Dette gjaldt blant annet MICMAC og Visual SFM. Visual SFM fikk vi omsider til å fungere helt mot slutten av tiden som var avsatt for undersøkelsene. Dette programmet valgte vi å avsette litt tid til fordi det er en variant der programmet CMVS/PMVS2 inngår, men her med GUI. Vi har gjort begrensede tester av Visual SFM i den betydning at vi ikke har gjort testmålinger av modeller. Derfor vil vi ikke kunne si noe om nøyaktighet på avstandsmålinger. Vi anser det som fornuftig å se videre på Visual SFM etter at våre undersøkelser er avsluttet. Nærmere beskrivelse finnes i kapittel 4.1.5.

De påfølgende seks delkapitlene beskriver metodene vi har brukt for å gjøre det mulig å teste hvert enkelt program opp i mot testkriteriene.

### 3.3.Referansemåling

Disse målene ble gjort med målebånd på selve bygningene. Vi målte opp forhåndsplanlagte detaljer som vindusbredde og – høyde, dører og fasader. Dette ble gjort med målebånd. Lengdemålene ble brukt som ”fasit” under analysedelen. Vi opererte ikke med et standard antall lengder for modellene. For modellene våre varierte antallet fra 10 til 43 stk, med lengder fra 25 cm til ca. 30 m. Hovedtyngden av disse lå på ca. 1 m. Referanseavstandene ble tegnet inn i 2D-bilder som ble benyttet som oppslagsverk under det videre arbeidet (Figur 66).







Figur 6 Eksempel på referansebilde med avstandene som skulle måles for deler av "Fløy 3"

Før fotograferingen signalerte vi flere punkter på og ved siden av to av objektene. Disse ble målt inn med CPOS. Punktene brukte vi til skalering og kontroll av målinger.

### 3.4.Prosjekter/Modellering

For å ha et grunnlag for testing og analyse, definerte vi fire prosjekter, henholdsvis ”Sørhellinga”, ”TF/IMT”, ”Fløy 3” og ”Meierikrysset”. Her plukket vi ut bildesett for hvert prosjekt, som skulle være et felles utgangspunkt for testing. Settene besto av bilder fra alle tre kameraene vi benyttet. Vi gjennomførte en del modelleringer per prosjekt for å ende opp med en modell vi var fornøyd med. Dette ga oss også en dypere forståelse for hvordan programmene virket, da ulike innstillinger av parameterne ga ulik kvalitet og grad av støy.

Navn / Beskrivelse	Bilde
<p>Sørhellinga</p> <p>17 bilder</p> <p>Canon EOS 5D MK II</p>	
<p>TF/IMT</p> <p>35 bilder (22 brukt i noen forsøk)</p> <p>Canon EOS 5D MK II</p>	
<p>Fløy 3</p> <p>32 Bilder</p> <p>Sony A-300</p>	
<p>Meierikrysset</p> <p>3 Bilder</p> <p>Canon EOS 400D</p>	

Tabell 2 Prosjekter

### 3.5.Skalering

Med skalering av 3D-modellen vil det si hvordan modellen skaleres for å få ut riktige avstander når målinger skal gjøres. Er det mulig å gjøre dette direkte ved hjelp av kjente kameraposisjoner eller må vi bruke en avstand målt ved hjelp av andre kilder? Andre kilder kan for eksempel være [www.norgebilder.no](http://www.norgebilder.no), tachymetri, bygningstegninger eller avstander mellom signalerte punkter

Vi har 4 metoder for å skalere modellene. Disse er som følger:

- SM 1. Skalering vha. GPS-data som er knyttet til bildene. Dette er GPS-posisjoner for selve kameraet. Disse ligger i bildenes EXIF-data, men kan også benyttes fra en tekstfil.
- SM 2. Skalering vha. for eksempel [www.norgebilder.no](http://www.norgebilder.no). Her henter man ut en referanseavstand som brukes som skalar for modellen. Enten ved å sette den direkte inn i modellen, eller ved å dividere den med verdiene for samme avstand i den uskalerte modellen. Slik får man en skaleringsfaktor som multipliseres med hele modellen.
- SM 3. Skalering vha. målte punkter i terrenget. Dette kan gjøres vha. f.eks. GPS eller ved å ta ut best mulige koordinater fra et bra kart/ortofoto. Punktene koordinater plottes i programmet.
- SM 4. Gjennomsnittlig skalar vha. tilhørende GPS-data. Gjøres ved å finne avstanden mellom alle mulige kombinasjoner av kameraposisjoner i modellkoordinater og det samme for GPS-posisjonene seg i mellom. Dividerer deretter GPS-avstandene på modellavstandene og får en skaleringsfaktor. Disse faktorene midles. Selve posisjonene vil ikke inngå i modellen i form av GEO referering.

Et annet og ofte tilgjengelig alternativ er å benytte ofte standardiserte størrelser fra kjente objekter f.eks. høyde på dører, lengden av en fotballbane osv. Dette vil alltid medføre en høy grad av usikkerhet da disse kan variere i virkeligheten. Metoden ligner i stor grad på SM2.

Vi gjennomførte modellering med hver våre program og testet skalering vha. av metodene over. Innen hver metode var det små forskjeller avhengig av hvilket program vi nyttet. Dette beskrives under kapittel 0.

Målefunksjonen vurderes sammen med skaleringen. Dette er fordi disse henger tett sammen under selve utførelsen.

### 3.6.Måling

De ulike programmene har individuelle metoder for å gjennomføre målinger (beskrives i kapittel 4 og 5). For å fange opp eventuelle målefeil som skyldes utydelige detaljer i modellen (MeshLab), gjennomførte vi fem målinger per detalj og midlet disse. Dette var ikke hensiktsmessig for PhotoModeler og Autodesk 123D-Catch.

Vi har gjennomført målinger av modellene der metode for skalering har vært forskjellig. Dette kommer vi tilbake til i kapittel 0).

### 3.7.Analyse av målinger

Alle lengdemålinger ble satt inn i et regneark der vi fra før hadde fasitmålene liggende. Her regnet vi ut differanse mellom fasit og målt verdi og kunne på denne måten beregne et absolutt gjennomsnittlig avvik fra fasit. Til dette har vi benyttet oss av *Root-mean-square deviation* (RMSD). RMSD benyttes ofte for å beskrive avstandsforskjeller i et utvalg. Slik vil vi være i stand til å beskrive forventet usikkerhet for våre lengdemålinger i form av centimeter, uavhengig av lengden på detaljene som skal måles. Dette vil være en verdi som forteller en bruker hva han kan forvente av modellen. Vi har beregnet RMSD for alle modeller og totalt for alle program samtidig som vi har sammenlignet modeller som er skalert på ulik måte.

Standardavvik for GPS-koordinatene <sup>2</sup> som er generert av navigasjonsprosesseringen under fotograferingen, finnes i Vedlegg C. Det eneste som var å utsette på disse dataene var standardavviket for høyden. Disse ligger på underkant av 5 m <sup>3</sup>. Disse koordinatene vurderte vi opp i mot kameraposisjonene hvert program beregnet (modellkoordinater). Dette gjorde vi for å illustrere en metode som sammenligner modellkoordinatene fra de forskjellige programmene med GPS-koordinatene. Dette gjorde vi med en Helmert 7-parameter transformasjon ved hjelp av en shareware-versjon av programmet Sevenpar2.10. [19]. Vi brukte samme metode for å vurdere modellkoordinatene opp mot hverandre. I Visual SFM er det, mulig å transformere fra modellkoordinater til ECEF ved hjelp av tilhørende GPS-koordinater. Denne funksjonen beskrives nærmere i kapittel 4.1.5. Dette ble gjennomført for å kontrollere resultater fra Sevenpar 2.10.

---

<sup>2</sup> Latitude, longitude og høyde

<sup>3</sup> En endring på 5 meter i høyde vil gi en forskyvning på 6 meter (ECEF)

### **3.8.Andre data og bruksområder**

Som det kommer fram fra teorien, blir det gjennomført en orientering av bildene. I denne prosessen finnes bildenes indre- og ytreorientering. Disse dataene er programmene helt avhengige av for å være i stand til å rekonstruere en modell. Vi har gått i dybden på alle programmene for å se hva de gir ut av data som er tilgjengelige for oss. Vi har gjort en enkel evaluering på kvaliteten av de ytre orienteringsparameterne. Vi belyser kun forskjeller på data generert for like bildesett, mellom de ulike programmene.

Noen programmer gir indre orienteringsparametere i piksler og andre i mm. Disse dataene kan konverteres fram og tilbake. For sammenligning av disse verdiene har vi fokusert på et bildesett av ”Sørhellinga”.

Vi har også sammenlignet mengden aktuelle sammenbindingspunkter som programmene finner og antallet match mellom de ulike kombinasjonene av bildene. Som grunnlag for beregningene brukes bildene nedskalert til 2048 piksler bredde slik at alle programmene har et likt utgangspunkt.

For Forsvaret vil et valg av et eventuelt program også kunne avhenge av andre funksjoner og bruksområder. Dette kan for eksempel være generering av lokale høydemodeller, objekt gjenkjenning og muligheter til å eksportere modellen til andre verktøy.

## 4. BESKRIVELSE AV SOFTWARE

### 4.1.SFM Toolkit 3/Photosynth Toolkit 7/Visual SFM

Disse programmene er i virkeligheten satt sammen av flere små program som gir en "Toolkit". De tre vi har sett på, er Photosynth Toolkit 3, SFM Toolkit 7 og Visual SFM. De to første er satt sammen i den form de er av Henri Astre for å gjøre det enklere og mere automatisert å gjennomføre en slik rekonstruksjon/modellering som vi vil oppnå, ved å benytte alle småprogrammene som ellers må kjøres hver for seg i primært Linux eller ved hjelp av f.eks. Cygwin for å kunne benytte Windows. Han har satt programmene sammen slik at de kjøres vha. to exe-filer. [20]

Visual SFM er laget av Changchang Wu ved University of Washington at Seattle [21] og grunnlaget for dette programmet er veldig likt de to første. Derfor er det greit å beskrive disse til dels sammen.

Alle tre programmene benytter seg av tre hovedsteg i prosessen slik vi har beskrevet i kapittel 2.

- Deteksjon og matching av sammenbindingspunkt
- Indre- og ytreorientering
- Generering av tett punktsky

Teorien bak disse stegene er gått gjennom i de foregående kapitlene. Dette gjelder også SIFT (kapittel 2.1)

#### 4.1.1. SFM Toolkit 3

Dette er en sammensetning av SIFT, Bundler og CMVS/PMVS2 (beskrives i kapittel 4.1.3) som er ferdig kompilert for å kjøre stegvis fra windows utforsker. Vi har benyttet versjon 3. Dette gjøres ved hjelp av to exe-filer der den første setter i gang SIFT og Bundler, og den andre setter i gang CMVS/PMVS2 som genererer den tette punktskyen. For hver operasjon man setter i gang, må man peke på mappen med de aktuelle bildene. Samtidig får man opp et vindu der man kan justere parametere både for Bundler og CMVS/PMVS2. Man kan imidlertid velge å beholde defaultverdiene. Ved spesielle tilfeller kan man gå inn i en egen tekstfil, hvor disse parameterne lagres, og etterjustere disse og noen tilleggsparametere. Dette beskrives nærmere under kapittel 4.1.3. Se eksempel i vedlegg J.



Underveis kan man monitorere hvordan prosessene går i det DOS-baserte commandvinduet. Utover dette har ikke programmet noen GUI. [20]

Etter at disse to delprogrammene er kjørt gjennom, sitter man igjen med en tett punktsky som kan videre prosesseres med f.eks. MeshLab [22]. Det er ikke mulig å skalere eller måle i punktskyen i toolkitet. Man får heller ikke ut en ferdig teksturert modell med trekantnett (TIN). Derfor kan man ikke si at programmet er fullt ut automatisk. Grunnen til at vi likevel har valgt å teste programmet, er at egenskapene til Bundler og CMVS/PMVS2 er mye omtalt i positive ordelag, og vi tror det har potensiale til å kunne dekke de behov vi har som rammer for våre undersøkelser. En meget viktig tilleggsfaktor er at man ikke er avhengig av tilgang til internett.

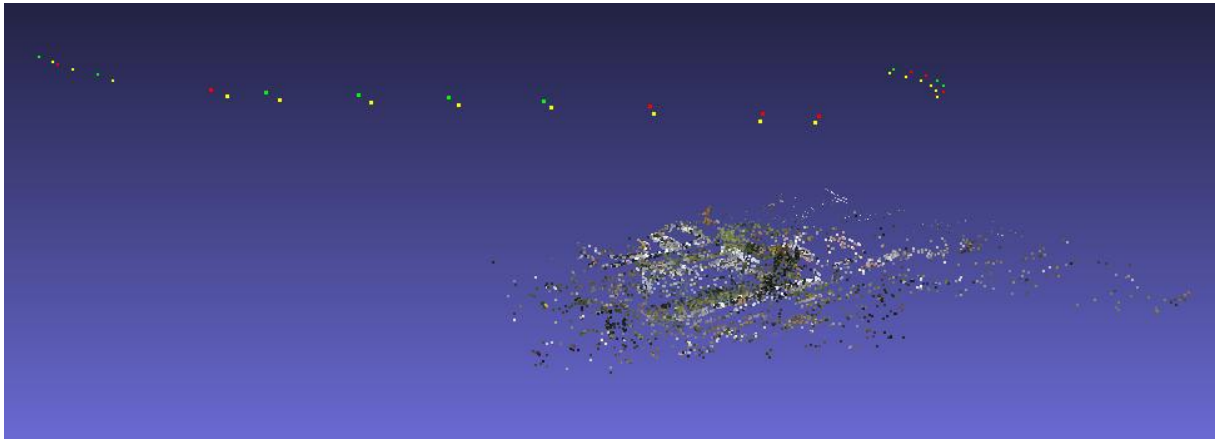
Vi vil beskrive de viktigste komponentene i toolkiten hver for seg i de neste kapitlene med unntak av SIFT (beskrevet i kapittel 2.1). Dette gjør vi for at man her samtidig vil konkretisere en god del av det tidligere gjennomgåtte teoristoffet.

#### **4.1.2. Bundler**

Dette programmet er laget av Noah Snavely og genererer en spredt punktsky med indre- og ytre orienteringsdata. Bundler er skrevet i C++ og finnes nå i versjon 4. Programmet benyttes til å skaffe gode midlertidige verdier for brennvidde og radiellfortegning for hvert enkelt bilde i tillegg til rotasjonsmatrise og kameraposisjon [23]. Bundler benytter SIFT [8] til å detektere og gjennomføre matching av sammenbindingspunkt. Dette gjør ikke Bundler selv. Bundler står for "Structure from Motion"-delen der man gjenskaper strålebunten. Disse sammenbindingspunktene utgjør en spredt punktsky og er i seg selv ikke tett nok til å kunne brukes for modellering og gjenskapning av modeller, men er derimot mer enn god nok for å gjøre fram- og tilbakeskjæring slik at programmet finner orienteringsparameterne.

Bundler har per nå noen begrensninger da man er avhengig av at man downsampler bildene til maks 2048 pikslers bredde, noe som skyldes begrensninger i SIFT-modulen som Bundler benytter for identifisering av sammenbindingspunktene. Begrensningene skyldes at Bundler benytter seg av en demoutgave av SIFT. For å kunne benytte den originale bildestørrelsen til full rekonstruksjon i PMVS2 kan man sette inn disse bildene, istedenfor de med redusert størrelse, etter å ha kjørt Bundler. For at dette skal fungere, må man da dividere bildebredden for det store bildet med det reduserte og bruke dette forholdstallet som skaleringsfaktor på

verdien for brennviddene i ”bundler.out”-filen. Fortegningskoeffisientene må stå som før.



**Figur 7** Spredt punktsky generert av Bundler

#### **4.1.3. CMVS/PMVS2**

CMVS står for Clustering Views for Multi-View Stereo og er på en måte rammen rundt PMVS2. CMVS gjør at man kan ta inn mange bilder uten at maskinen låser seg på grunn av minneproblematikk. Programmet deler opp bildene i mindre grupper og lar PMVS2 gjøre rekonstruksjon på hver av disse gruppene uten at noen detaljer, som er mulig å rekonstruere med den opprinnelige mengden bilder, går tapt. Det gjøres selvfølgelig på en slik måte at resultatene fra hver gruppe passer sammen. På denne måten får man en raskere prosess som gjør at man ikke trenger den dyreste og mest kompliserte maskinvaren for å benytte seg av store bildemengder. [24]

PMVS2 står for ”Patch-based Multi View Stereo”, 2. utgave og er utviklet av Yasutaka Furukawa- University of Washington og Jean Ponce - Ecole Normale Supérieure.[14] Dette programmet er det som gjør hovedjobben. Den dypere teorien rundt ”patch-based matching” er gått gjennom i kapittel 2.3.2. Her genereres det en tett punktsky i tillegg til at programmet renser punkter som kan regnes som støy. Det vil likevel kunne finnes støy i punktskyen etterpå. Pikslar som tilhører f.eks. plenen bak bygget vil kunne feste seg til den tilstøtende tak kanten og flater kan inneholde flere lag med punkt. Dette vil begrenses i meget stor grad, ved å regulere innstillingene for prosessen i option-fila i PMVS2-folderen. Denne fila gir programmet de nødvendige input-verdier for hvordan det skal kjøre rekonstruksjonen. Innstillingene står beskrevet i Vedlegg I. Innstillingene styrer alt fra hvor store patcher som skal brukes i matchingen til om man skal benytte kantdeteksjon og i hvor stor grad. Den kanskje viktigste verdien er den som heter ”Treshold” og styrer hvor god kvalitet det må være

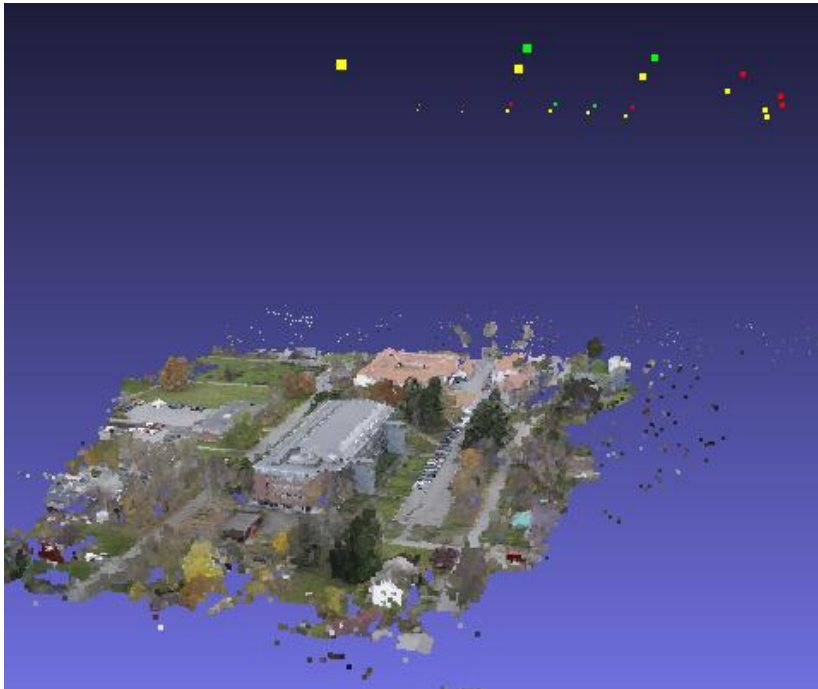
på et 3D-punkt for at det skal bli med i den endelige punktskyen. Likevel vil man kunne ha behov for å rense litt ekstra i punktskyen sin. Vi har brukt MeshLab [22] til dette.

Punktskyen inneholder mye data i form av punkter, men her må man huske at all informasjon programmet har klart å få ut av bildene, er med. Ved modellering vil man kanskje ha behov for bare en spesiell fasade eller område av modellen og man klipper derfor bare ut den ”biten” man behøver, før man genererer TIN og tekstur.

For at prosessen skal gå raskest mulig kan man benytte såkalte ”masker”. Det vil si at man klipper ut det interessante område i et bilde og gir det en hvit farge. Dette kaller vi ”forgrunn”. Resten av bildet farges helt svart og vi kaller dette området for ”bakgrunn”. Bildet lagres til slutt som et gråtonebilde og legges med i tillegg til det originale bildet. Bakgrunnen vil da ikke bli tatt med i matchingen. Man sparer derfor mye prosesseringstid og kan benytte ekstremt mange bilder for en modell, noe som igjen vil øke mulighetene for å gjenskape små detaljer i bildene innenfor det området man har avgrenset.

Ved å benytte strenge verdier for kantdeteksjon, vil man gjenskape for eksempel kantene av vinduer og dører meget godt. Disse kantene inneholder veldig lite støy og kan benyttes for å avgrense flater som ellers er vanskelige å rekonstruere.

Dette programmet kom best ut i en sammenligning av fire av seks modeller i en test for Multi-View Stereo algoritmer. Her ble den genererte modellen sammenlignet med det de definerer som ”ground-truth” og man bruker fullstendighet i prosent og nøyaktighet i mm som evalueringsfaktorer. [25] Vi fant ikke noen informasjon om hvordan ground-truth var oppnådd.



**Figur 8** Tett sky generert av PMVS2 vist i MeshLab (kameraposisjoner over selve modellen).

Vi har utvekslet noen e-mailer med Yasutaka Furukawa for å få supplement til teorien rundt softwaren. Et viktig punkt var å finne ut om softwaren er implementert i noen form for kommersiell software, da dette vil kunne spare Forsvaret for tid, penger og ressurser i forhold til å utvikle et eget system.

Svaret på dette var som følger: -” Unfortunately, a company usually does not want others to know what kind of technologies are used behind their products. Therefore, I cannot disclose that information. But with a few exceptions. I am allowed to say that Industrial Light & Magic uses our software for their commercial use.”

«Dessverre vil ikke et firma at andre skal vite hvilken type teknologi som utgjør grunnlaget for deres produkter. Derfor kan jeg ikke si noe om dette bortsett fra noen unntak. Jeg har lov til å si at Industrial Light and Magic bruker vår software kommersielt.»

Man kan gå ut fra at PMVS2 også er implementert i annen kommersiell software. Industrial Light & Magic er et filmselskap som har spesialisert seg på spesialeffekter i film og har blant annet bidratt i ”Star Wars”-filmene og en rekke av de andre tidløse filmklassikerne. På deres nettside er det ikke skrevet noe om PMVS2 i klartekst, men det tyder på at man benytter denne softwaren for scanne små fysiske modeller som deretter tekstureres og skaleres opp for videre å kunne klippes inn i filmer. [26]

#### 4.1.4. PhotoSynth Toolkit 7

Denne toolkiten er forskjellig fra SFM Toolkit ved at man benytter Photosynth for deteksjon og matching av sammenbindingspunkter og orienteringsprosessene. Ellers benyttes CMVS/PMVS2 på samme måte. Vi har ikke lyktes i å finne ut hvilke algoritmer som blir brukt i Photosynth.

PhotoSynth er en web-applikasjon som fungerer ved at man etablerer en brukerkonto med plass til 20GB. Deretter laster man ned et lite program som brukes til å laste opp bilder til brukerkontoen. Her prosesseres bildene i web-applikasjonen. PhotoSynth genererer indre og ytre orienteringsparametere, i tillegg til at man kan se resultatet i en meget praktisk viewer. Funksjonen med at bildene ligger «oppå» hverandre er særdeles interessant da man til en viss grad vil oppnå en slags 3D-effekt selv om bildene ligger i 2D. Man kan også se dataene som en spredt punktsky. Denne punktskyen kan lastes ned og benyttes i andre applikasjoner i form av en ply-fil med kameraorienteringsparametere og koordinater for punktene.

Proessen med å laste opp bilder og få disse prosessert er meget rask. Her har vi bla. gjort et forsøk med 82 bilder av et objekt.

En stor svakhet med PhotoSynth er at man er avhengig av å laste opp bildene på nettet, noe som i mange sammenhenger, for Forsvaret, ikke vil være ønskelig eller mulig, på grunn av manglende internett-tilgang eller operasjonelle hensyn. Programmet gjør tilsynelatende ikke så mye mer enn det som SFM Toolkit gjør. Av disse to årsakene valgte vi å ikke gjøre noen tyngre testing av denne varianten.



Figur 9 Spredt punktsky i Photosynth av TF/IMT laget fra 82 bilder



Figur 10 2D-vieweren i Photosynth

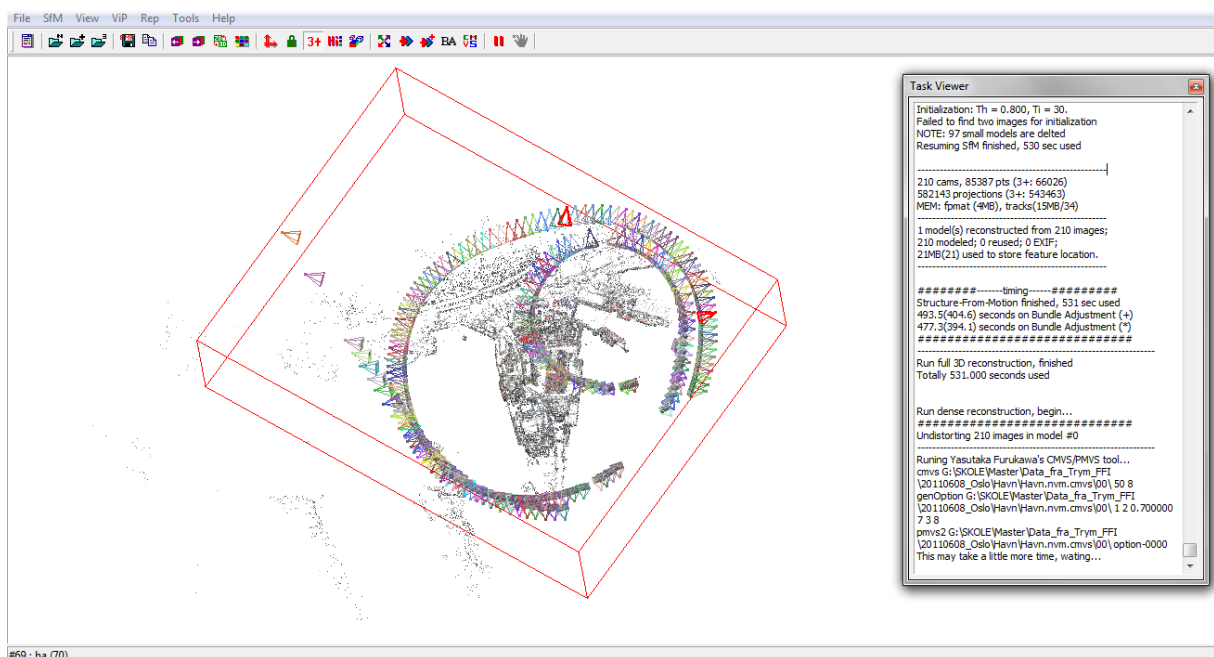
#### 4.1.5. Visual SFM

Dette programmet var et av de første vi så på som benytter PMVS2 for å generere 3D-modeller, men vi fikk det ikke til å fungere i starten fordi det var behov for en del triksing med filer og filstruktur for å få det til å fungere. Derfor ble testing av dette lagt på is. Mot slutten av perioden vi hadde avsatt for testing, fikk vi det som sagt i gang og en del begrenset testing ble gjort. Dette gikk stort sett på funksjonalitet.

Som navnet sier, er dette visuelt/synlig structure from motion (SFM) ved at man har en GUI der man kan monitorere alle steg i prosessen fram til og med en tett punktsky og en Task Viewer som tillater brukeren å monitorere prosessene (Figur 11, side 36).

Programmet benytter SIFT for å finne aktuelle sammenbindingspunkter i bildene, videre en egen orienteringsmodul som heter Multicore Bundle Adjustment (MBA) og til slutt PMVS2. MBA er et program som er utviklet for å sette ned prosesseringstiden og bruk av minne. Som navnet sier er programmet designet for å utnytte alle kjerner som er tilgjengelige på en datamaskin. Utvikleren hevder programmet gjør en 3D-rekonstruksjon 30 ganger raskere enn det han kaller eksisterende "state of the art" software. Dette med samme nøyaktighet som disse programmene. [27] Dette er et oppsett som er relativt likt SFM Toolkit, men Visual SFM benytter tydeligvis en annen utgave av SIFT som lar oss benytte bilder med mere oppløsning enn den utgaven Bundler benytter seg av. Programmet gir ikke en ferdig teksturert overflatemodell, dette må man i likhet med SFM Toolkit gjøre selv vha. MeshLab. I GUI'en

finner man en tekstmeny der man kan gjøre forskjellige trinn i prosessen. Man kan lese inn enkeltbilder, mange bilder, tekstfil med bildeliste som viser til en bestemt plassering eller bildesekvens fra video. Videre kan man gjennomføre matching mellom bildepar, alle bilder, eller et visst antall bilder i sekvensen. Deretter kan man gjøre selve orienteringen av bilder mens man ser på at punktskyen genereres med tilhørende kameraposisjoner. Når dette er gjort, kjører man PMVS2 for å konstruere den tette punktskyen. De fleste funksjoner finnes både i tekstmenyen og som hurtigknapper i GUI. I tekstmenyen kan man gjøre en del ekstra operasjoner som ikke finnes som hurtigknapper. Her kan man også gjøre innstillinger som går på ren ytelse.



**Figur 11 Visual SFM GUI med et prosjekt under orientering**

Under tekstmenyen ”SfM->More Functions->Estimate Georeference” legges det til rette for å ta inn GPS-data for bildene, enten i form av EXIF eller en tekstfil. Dette er en funksjon som i følge utvikleren er under stadig utvikling. Vi har ikke testet disse funksjonene i noen stor grad. Man kan også gå inn og manuelt gi utvalgte punkter nye koordinater (f.eks. kartesiske) og gjennomføre en transformasjon slik at man endrer hele punktskyen til det nye systemet. Vi har testet denne ved å endre kameraposisjonene for Sørhellingaprojektet.



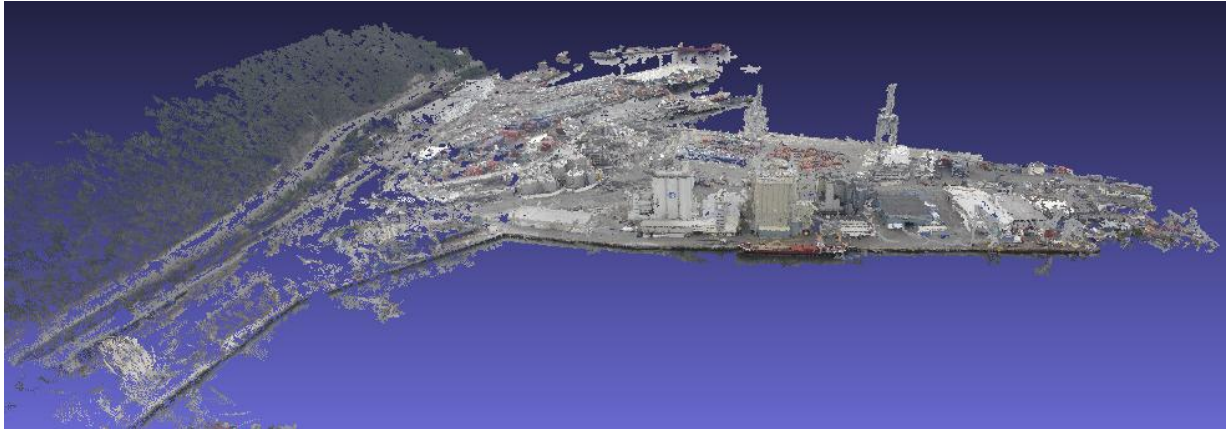
**Figur 12 Verktøylinjen i Visual SFM**



Outputformatet på både den spredte punktskyen og den tette, generert med PMVS2 er i .ply-format, noe som gjør det anvendelig for videreprosessering i andre og kanskje bedre egnede programmer, f.eks. MeshLab.

En positiv egenskap er at man kan benytte full størrelse på bildene selv om utvikleren ikke anbefaler større enn 3200 pikslers bredde. [21]

Programmet håndterer store mengder bilder. Se figur 13.



**Figur 13** Tett punktsky av havneområde ved Sjursøya, laget med 211 bilder

Programmet har også en del ekstra funksjoner som kan komme godt med hva angår bildeanalyse. Her kan man nevne analyse av repeterende objekter i enkeltbilder.

## **4.2. Autodesk 123D Catch**

123D Catch er en software (Beta-versjon) levert av Autodesk. Dette genererer 3D-modeller automatisk. Nettsiden beskriver følgende om produktet:

*Turn ordinary photos into extraordinary 3D models with Autodesk® 123D™ Catch. Take photos of your favorite people, places and things and let the power of the cloud magically transform them into detailed 3D models. [28]*

Det kan virke på oss som om programmet er mest beregnet som underholdning, ikke som et rent fotogrammetrisk produkt. Dette er basert på programmets særdeles enkle oppbygning samt nettsidene til programmet,

Programmet kan lastes ned fra nettsiden, installeres og brukes umiddelbart. Bildene må lastes opp på nettet, prosesseres på en ekstern server, før modellen lastes ned igjen og kan brukes. Programmet innehar flere valgmuligheter, for eksempel muligheten for å legge til flere bilder,



gjøre enkle målinger samt eksport til forskjellige formater. Det er blant annet mulig å eksportere til ipm.formatet som egner seg spesielt til bruk på mobile enheter som iPhone og iPad. Programmet kan også lage videofilmer av modellen. I denne er det mulig å «fly» rundt i modellen. Dette kan gjøres automatisk, men det er også mulig å legge til egne ruter og skjermbilder for å visualisere det som ønskes.

Hvis programmet ikke klarer å orientere og stiche enkelte bilder, innehar programmet en mulighet for å manuelt orientere disse til orienterte bilder. Dette gjøres ved at 4 punkter markeres i bildene, deretter markeres punktene i 2 orienterte bilder. På denne måten får programmet hjelp under orienteringen av bildene.

Det er ikke mulig å endre parameterne for prosesseringen før denne starter. Eksempler på disse kan finnes i SFM Toolkit. I enkelt tilfeller kan det være ønskelig med muligheten til å endre disse, men det vil gjøre kravet til brukeren større.

Gjennom forum på internett har vi fått bekreftet at programmet gjennomfører en kamerakalibrering (både K1 og K2) som en del av prosessen. Men det er ikke mulig å få resultater fra kalibreringen.

Programmet gir noe informasjon om modellen. Ved å eksportere ut en fbx.fil, er det mulig å få ut informasjon om kameraenes modellkoordinater samt noe informasjon om kamerainnstillinger. Vi brukte bildenes modellkoordinater og *focal length* når vi sammenlignet programmer.

Teknologien bak Autodesk 123D Catch er levert av Acute 3D. Vi har prøvd å komme i kontakt med *Acute 3D* for å få informasjon om hva som ligger bak 123D Catch, men vi har ikke mottatt noe svar. Dette gjør det vanskelig for oss å beskrive hvilken teknologi som ligger bak. Nettsiden til *Acute 3D* beskriver følgende om sitt program *smart3Dcapture* :

*smart3Dcapture™ is an industrial software solution allowing to automatically reconstruct true-to-life 3D models from a set of simple photographs. Stemming from over 25 years of research, smart3Dcapture™ is based on the highest-performance photogrammetry, computer vision and computational geometry algorithms. It fulfills industrial-quality requirements in terms of usage, robustness, performance, interoperability and portability. [29]*

Det er vanskelig å si noe om hvilke algoritmer som ligger bak. Men vi er ganske sikre på at en av de nye moderne bildematchingsalgoritmene ligger til grunn for dette. Vi har funnet en

artikkel som hevder at PMVS er inkludert i programmet. I denne er det også henvist til tester på nøyaktigheten til Autodesk 123D Catch. [30] Vi vil referere kort til denne undersøkelsen i kapittel 5.5.

### **4.3.Photomodeler**

PhotoModeler er kanskje det mest brukte og kjente program som nyttes til 3D modellering i dag. I tidligere versjoner har kravet til brukeren vært stort og graden av automatisering svært liten. Brukeren selv har vært nødt til å orientere bildene ved å finne samsvarende referansepunkter i bildene. I tillegg måtte formene tegnes inn ved hjelp av bildene. Programmet har vært imidlertid vært gjennom en stor utvikling, og det er på grunn av denne at vi har inkludert i våre undersøkelser.

I den nyeste versjonen, PhotoModeler Scanner (referert som PhotoModeler videre i oppgaven), er det mulig å gjennomføre prosesser som er svært interessante for vår oppgave. [31]

- Større hastighet på grunn av nye algoritmer.
- Automatisk matching av naturlige objekter i bildene.
- Geografisk koordinat system/datum.
- Målefunksjonalitet

Nettsidene sier ingenting om modellering av større bygninger eller kvartaler, med 360° dekning, som kan være interessante for vår oppgave. Det virker på oss som om programmet er beregnet på mindre objekter med bilder kun fra den ene siden av objektet.

Utfordringen for oss, med et slikt program som PhotoModeler, er at vi ikke får innsikt i hvilke algoritmer som ligger til grunn for det som gjøres i prosessene. Vi har prøvd å komme i dialog med PM for å få en dypere innsikt i programmets algoritmer, men dette var ikke noe de ville gi ut. Vi ble henvist til programmets websider når det gjelder dette. [32]

Vi velger å beskrive de delene av PM som er aktuelle for vår problemstilling og våre testkriterier.

### 4.3.1. Automatisk prosess/SmartMatch

Under oppstarten av programmet er det mulig å velge et automatisert prosjekt. Dette prosjektet bruker enten ferdigkodete mål (targets) plassert i objektet før fotograferingen, ukodete mål i objektet eller SmartMatch. SmartMatch er en av de nye viktigste modulene i PhotoModeler Skanner. SmartMatch finner naturlige objekter i bildene og matcher disse med hverandre mellom de forskjellige bildene. [31]

Bildene som nyttes i et SmartMatch prosjekt må ha god overlapping, ingen områder uten dekning og liten vinkel mellom bildene. I tillegg bør ikke overflatene i bildet være homogene eller glatte. Jo mer variasjon som finnes i bildet jo lettere gjennomføres matchingen. [33]

PM gjennomfører den automatiserte prosessen på følgende måte:

- Registrering av bilder og kamera.
- *Feature detection / Key point detection.*
- Bildematching
- Orientering av bilder

Før denne prosessen starter kan man velge tetthet på matchingen. Valgene er enten høy eller lav. I tillegg kan man velge hvilke bilder hvert enkelt bilde skal matches med. Hvis bildene er store, endrer PM størrelsen til mindre bilder. Bildene våre på 5616x3744 ble endret til 1404x936, altså med en faktor på 4. På oss virker det som om en av forskjellene på lav og høy tetthet er i hvor stor grad bildene nedskaleres. Ved lav tetthet ble bildene skalert mere ned i forhold til høy.

Deretter finner programmet *keypoints*. Alle punktene benyttes ikke, de beste plukkes ut til den videre prosessen. Nå starter den parvise matchingen av bildene med bakgrunn i disse punktene. Hvis PM finner over 500 punkter som matcher i et par, nyttes kun de 500 beste til orienteringen videre. Til slutt starter orienteringen av bildene med bakgrunn i de matchede punktene som er funnet.

Figur 14 viser punktene som ble automatisk generert av PM. Tettheten er forholdsvis stor, men vi ser at i de største homogene flatene (tak, asfalt og gress) har en mye mindre tetthet av punkter. Disse områdene har en radiometrisk likhet som gjør det vanskelig å gjennomføre en matching.

Prosesen gjennomføres relativt raskt, avhengig av antall bilder. Resultatet etter at det automatiske prosjektet er gjennomført er en spredt punktsky. Deretter er det mulig å gjennomføre en prosessering hvor blant annet kamerakalibrering kan gjennomføres med utgangspunkt i den spredte punktskyen generert tidligere. Hvis et kalibrert kamera nyttes tas, kalibreringsdataen hensyn til i prosesseringen. Bildene orienteres deretter på nytt. Resultatet av prosessen får vi ved å trykke på *Project Status report*. Et eksempel på en slik rapport finnes i Vedlegg L. I denne rapporten finner vi også kamerakoordinater og rotasjoner.



**Figur 14** Smart Points generert av PhotoModeler

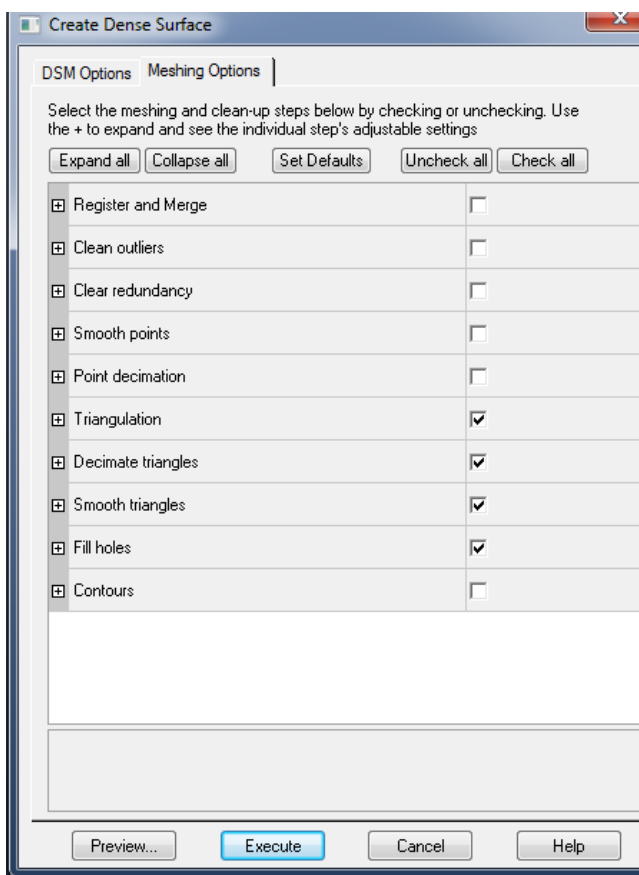
Som sagt tidligere har det ikke vært mulig for oss å si noe om hvilke algoritmer som ligger bak PhotoModelers Smart Points. På forum på internett har vi imidlertid funnet noen som mener at PM bruker SIFT til sin *Feature Detection*. [34]

### 4.3.2. Dense Surface

Den automatiske prosessen med SmartMatch gir som sagt brukeren en spredt punktsky av objektene. Den videre genereringen av en tettere punktsky, TIN og teksturering skjer i modulen *Dense Surface*. Litteraturen vi har sett på i forbindelse med PM sier ikke noe annet enn følgende om hvordan programmet løser genereringen av den tette punktskyen.

*Eos' PhotoModeler Scanner product introduces an advanced correlation-based image scanner/matcher. This technology is capable of extracting accurate dense point clouds from textured surfaces without a laser and without any surface modification. [32]*

Modulen *Dense Surface* gir brukeren svært mange valg, både når det gjelder punktskyen og teksturen. Mulighetene er svært mange.



**Figur 15** Muligheter i PM Create Dense Surface

Vi har ikke satt av mye tid til å få til tekstur på modellene til PhotoModeler. Dette er et bevisst valg i og med at vi er ute etter et automatisert program samt at brukerne til vår oppdragsgiver ikke skal være avhengig av mye opplæring.

### 4.3.3. External Geometry

En av de nye modulene i PhotoModeler Skanner er *External Geometry*. I denne kan man legge inn alle typer skaleringer, rotasjoner og transformasjoner.

I modulen er det mulig å utnytte GPS-data som ligger i bildenes EXIF-fil. På denne måten kan modellen skaleres, roteres og plasseres riktig i et geografisk referansesystem. Det legges til *geographical points*, først *dummy points*, samme antall som bilder i serien. Deretter kan man, via 3D vieweren, tilegne kameraer til punktene. Da spør PM om vi vil bruke modellkoordinatene eller de observerte størrelsene fra GPS (lagret i EXIF). Men dette skjer etter at den automatiske matchingen er gjennomført. Det virker ikke som om PM utnytter denne dataen (koordinatene) som en del av orienteringsprosessen. PM trenger 3 kjente punkter for å gjennomføre en transformasjon. PM setter ikke kameraposisjonene fast, men disse beregnes fortløpende som en del av transformasjonen etter hvert som nye posisjoner legges til. Det er ikke mulig å vekte på noe vis i PM.

Det er også mulig å transformere modellen ved hjelp av innmålte punkter på bakken. Dette gjøres ved at punktene markeres i flere bilder. Deretter legges det til *geographical points*, som tilknyttes punktene på bakken. Ved å trykke på *edit* kan koordinatene legges til for punktene.

I denne modulen er det i tillegg mulig å legge inn kjente punkter med koordinater, samt punkter kun til kontroll (*check points*). Koordinatene som må legges inn i kontrollpunktene må legges inn i jordsentriske koordinater. Det vil si at brukeren selv må gjennomføre en transformasjon mellom de forskjellige koordinatsystemer og kartprojeksjoner. Figur 16 viser resultatet av en slik kontroll<sup>4</sup>.

Det er mulig å legge inn punkter direkte eller å laste de ned via en fil.

A..	V...	Name	Id	X (m)	Y (m)	Z (m)	Delta m
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Check point 8	Pt-132	3172399.27	603801.60	5481852.98	3.37
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Check point 5	Pt-153	3172380.92	603926.76	5481843.41	4.39
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Check point 3	Pt-116	3172412.41	603784.24	5481851.97	3.30
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Check point 12	Pt-100	3172443.24	603802.39	5481835.55	2.67
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Check point 11	Pt-84	3172453.00	603806.06	5481828.01	2.59
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Check point 10	Pt-145	3172446.35	603792.34	5481831.56	2.66

Figur 16 Check Points i External Geometry

<sup>4</sup> Dårlige resultater på grunn av mye støy i GPS-observasjonene.

## 5. RESULTATER

I kapittel 3 beskrev vi metoden for vår oppgave. Vi vil som sagt vurdere hvert enkelt program opp i mot våre testkriterier. Disse er som følger:

1. Grad av automatisering
2. Ivaretagelse av geometri og tekstur
3. Skalering av 3D-modell
4. Utnyttelse av observerte størrelser
5. Hvilken nøyaktighet kan oppnås

Testing av Visual SFM er i stor grad basert på å teste funksjonaliteten på programmet og ikke nøyaktigheten (-modellkoordinater) på dataene som genereres, da disse med høy sannsynlighet vil være nokså like som for SFM Toolkit.

### 5.1. Grad av automatisering

Grad av automatisering vil si i hvor stor grad brukeren selv må bruke tid på å få programmet til å fungere og på å få ut fornuftige modeller. Med dette mener vi også skalering, måling, generering av TIN og teksturering.

Vi har også vurdert ytelsen på hvert program som en del av denne testen.

#### 5.1.1. SFM Toolkit

SFM Toolkit er i hovedsak et trinnvis automatisk system der man kan velge å benytte defaultverdiene som ligger inne. Her har det vist seg at det for vårt bruk kanskje bør ligge andre verdier som default. Med trinnvis menes at man først setter i gang Bundler. Neste steg, når denne prosessen er over, er å sette i gang PMVS2<sup>5</sup>. Deretter har man en tett punktsky som vil være grunnlaget for å generere en TIN med tekstur. Til dette må man se etter en godt egnet punkteditor. Vi har brukt MeshLab fordi den er gratis på nettet, men antar at det finnes bedre egnede systemer å få kjøpt.

Ved å overstyre de mulige innstillingene for Bundler og PMVS2, kan man veldig enkelt få meget gode resultater ut av få, små og relativt dårlige bilder.

Muligheten i CMVS for å clustre bilder gjør at et ekstremt høyt antall bilder kan benyttes.

---

<sup>5</sup> Detaljert beskrivelse av prosedyre i Vedlegg H

For å kunne måle modellene generert av SFM Toolkit benyttet vi MeshLab. Målefunksjonen her fungerer ved at man med et klikk setter startpunkt for målingen og klikker en gang til i endepunktet. Dette gjøres direkte i modellen og fungerer til sammenligning veldig likt som i f.eks. ”Norge i bilder”. Med andre ord en veldig enkel funksjon.

For å generere en tett punktsky med bilder skalert til 2048 pikslers bredde opplevde vi at prosesseringstiden var alt fra 3 minutter til ca. en time. Bundler bruker alt fra et par minutter til 15 minutter på å orientere bildene, avhengig av antall. For videre å generere den tette punktskyen i PMVS2 kan man oppleve litt lengere tid. Vi har gjort forsøk med både lite og stort antall bilder. Tre bilder, som i prosjektet *Meierikrysset*, tar ca. 3 minutter, mens prosjekt med 80 bilder kan ta opp mot en time. Ved å benytte mulighetene for å stille inn hvor mange bilder som skal prosesseres på en gang i CMVS(valg som kommer opp under oppstart), vil man spare tid ved at maskinen får utnytte all prosessorkraft og minne tilgjengelig. Hvis man prøver å prosessere et stort antall bilder på en gang (f.eks. 80), vil det totalt sett gå mye mere tid da man får minneproblematikk. 10 bilder i hver prosessering er et godt utgangspunkt.

For å oppsummere så er SFM Toolkit en trinnvis prosess som genererer en tett punktsky, altså ikke en ferdig teksturert modell. Man kan gjøre innstillinger etter behov, men også velge å bruke defaultverdiene. Disse bør tilpasses en standard i forhold til bruksområde.

### **5.1.2. Visual SFM**

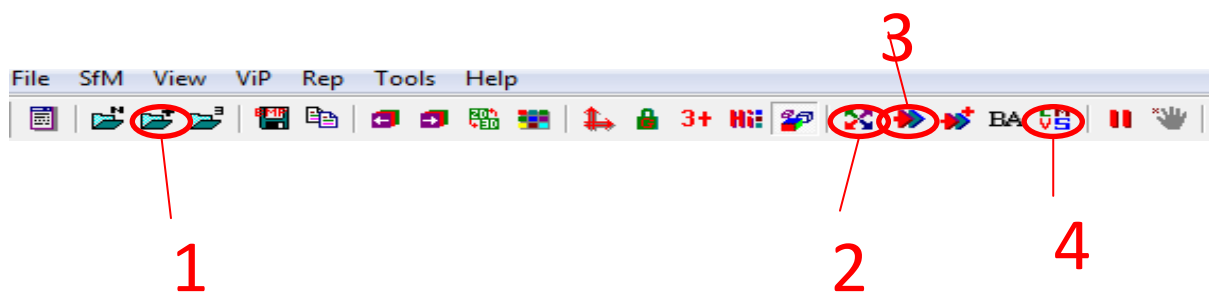
Visual SFM gir denne ”pakken” en mye høyere brukervennlighet og grad av automatisering da man har en egen GUI med både hurtigknapper og tekstmeny. Her kan man samtidig overstyre en del parametere som styrer ytelse, men også mengden bilder som skal matches med hverandre og mengden punkter som skal genereres. En meget bra egenskap er at det er mulig å benytte et meget stort antall bilder uten at programmet begynner å slite. For å lage en tett punktsky benytter man 4 knapper. En for å laste inn de ønskede bildene, en for å matche bildene mot hverandre, en for å gjennomføre orientering og til slutt en egen for å kjøre PMVS2. Fordelen med at man gjør de forskjellige trinnene hver for seg, er at man på et hvilket som helst tidspunkt, kan gå tilbake og legge til eller ta bort bilder. For å justere parameterne for PMVS2 må man gå inn i en scriptfil som heter ”nv.ini”. Denne er veldig intuitiv, men likevel litt tungvint for en normalbruker.

Visual SFM er i stor grad automatisert fram til den tette punktskyen er etablert og eventuelt skalert ved hjelp av GPS-data. Innlesing av GPS-data og videre transformasjon er meget



enkelt. Dette gjøres med to tastetrykk, ett for innlesing fra bildenes EXIF og et for å transformere. Programmet transformerer da både den spredte og den tette punktskyen. Her er det viktig å få med at man gjør dette til slutt. Man kan også lese inn GPS-data fra en tekstfil på like enkelt vis.

I forhold til hastigheten på en prosessering så er Visual SFM noe raskere enn SFM Toolkit fordi innlesing av bilder og orientering av disse går en god del raskere. Utviklerens utsagn om at *Multicore Bundle Adjustment*-programmet gjør dette mellom 30-40 % raskere enn andre tilsvarende programmer høres fornuftig ut. [27] Ellers er tiden som PMVS bruker, likt som for *SFM Toolkit*.



Figur 17 Stegene for å gjennomføre en 3D-rekonstruksjon i Visual SFM

Figur 16 viser hvilke knapper man benytter for å gjennomføre en 3D-rekonstruksjon i Visual SFM. Dette gjøres ved å: 1) Laste bilder, 2) Matching av bilder, 3) Orientering(SFM), 4) Generere tett punktsky.

### 5.1.3. Autodesk 123D Catch

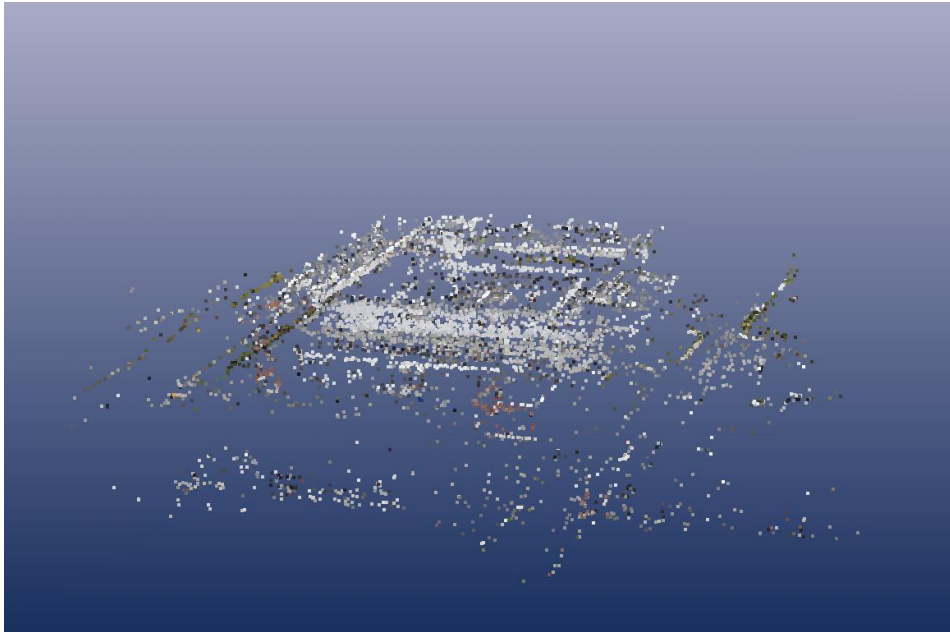
I Autodesk 123D Catch kreves det svært lite for å klare å lage 3D modeller. Programmet er selvforklarende og enkelt og bruke. Dette gjelder i stor grad alle funksjonene i programmet også. Disse er enkle og bruke og krever lite forhåndskunnskaper.

Som sagt i Kapittel 4.2 er det ikke mulig å endre parameterne før prosessen startes. Med denne muligheten ville en eventuell bruker med noe erfaring klare å inkludere bilder som ikke går gjennom automatisk, samt få utnyttet ”dårlige” bildesett i større grad.

Programmet klarer å gjennomføre prosesseringen relativt raskt, 3 til 20 minutter avhengig av antall bilder. Men slik programmet fungerer nå er det mer avhengig av netthastighet i stedet for maskinvare.

#### 5.1.4. PhotoModeler

PhotoModeler kan generere en spredt punktsky automatisk på en svært enkel måte. Brukeren må laste inn bildene og gjøre noen enkle valg før prosessen starter. Figur 18 viser en spredt punktsky generert automatisk.



Figur 18 Punktsky av TF generert automatisk av PhotoModeler

Utfordringen videre i PM er å generere en modell med tekstur. Det vil si en tett punktsky, TIN og tekstur. Dette er en vanskelig prosess som krever mye av brukeren. Selve resultatet av dette kommer vi tilbake til i kapittel 5.2. Programmet i seg selv krever mye av brukeren sett i forhold til de andre vi har testet. Brukeren må bruke en del tid på å sette seg inn i programmet før dette kan brukes.

I modulen *External Geometry* er det mulig å utnytte GPS-posisjonene i bildenes EXIF data. Denne prosessen kan ikke gjøres automatisk, men bilde for bilde.

Programmet klarer relativt raskt å prosessere mange bilder, men dette er fram til den spredte punktskyen er etablert. Å generere en tett punktsky tar lengre tid, 10 til 30 minutter avhengig av antall bilder.

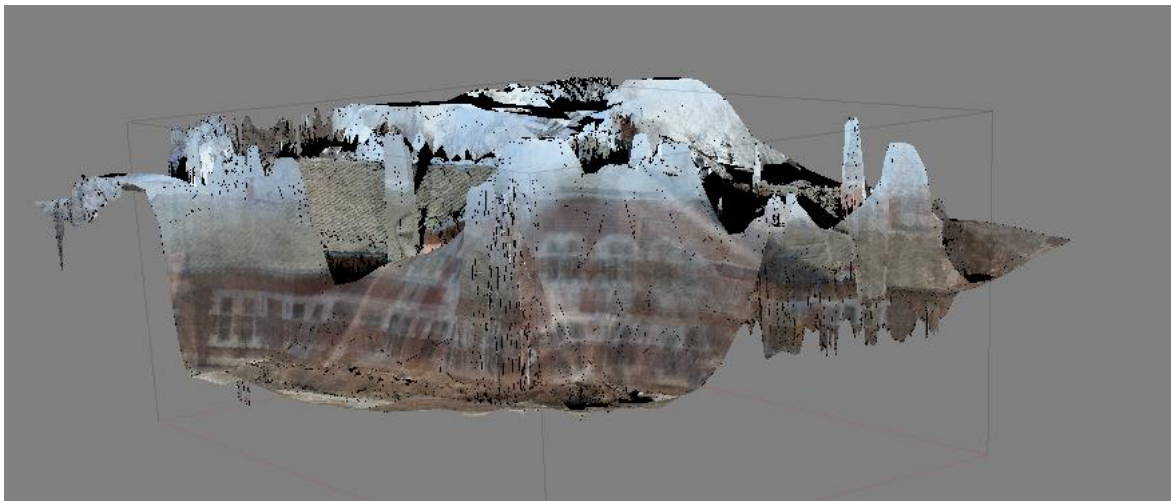
### 5.1.5. Delkonklusjon

Autodesk 123D Catch peker seg ut som det klart mest automatiserte programmet. Det er enkelt å bruke samt at det er det eneste programmet som kjører prosessen fullstendig automatisert fram til og med teksturering. Imidlertid må det gjøres en avveining angående fordeler og ulemper med justering av parametere underveis i prosessen. SFM Toolkit og Visual SFM gjennomfører begge en automatisert prosess helt fram til den tette punktskyen er etablert. Steget derfra til en modell med full tekstur gjennomføres ikke automatisk. Alle programmene klarer relativt raskt å etablere bra modeller av objektene.

### 5.2. Ivaretagelse av geometri og tekstur

Prosessene i programmene er ikke ufeilbare. Det er mange steg som skal gjøres før en modell er klar. Underveis kan algoritmene gi resultater som ikke er riktige. Dette kan være dårlige eller få sammenbindingspunkter, eller feil i orienteringsdata (indre og ytre). Dette vil forplante seg videre i prosessen og i en del tilfeller gi fordreide modeller. Faktorer som blant annet påvirker dette har vi beskrevet i kapittel 2.4.

Ikke alle modeller blir bra. Figur 19 viser et eksempel på en dårlig modell.



Figur 19 3D modell av Meierikrysset generert av Agisoft Photoscan

### 5.2.1. SFM Toolkit

Alle bygg og objekter blir gjengitt med riktige former i den forstand at vi ikke har kontrollert vinkler, men rent visuelt.

Langs kanter som grenser mot friluft, hender det at man kan få med piksler som f.eks. tilhører gressplenen som ligger bak kanten, sett fra hovedretningen til bildene som er matchet. Disse vil være godt synlige i den tette punktskyen, men gjør lite eller ingenting av seg i den endelige modellen. Dette forekommer mest nært bildekantene, dvs. lengst vekk fra kameraet og rammer stort sett objekter som sannsynligvis ikke er det vi primært er interessert i å modellere.



**Figur 20** Eksempel på "plen-punkt" festet til tak kanten

Softwaren er i stand til å rekonstruere en scene vha. kun to bilder, men dette vil kunne gi en del støy hvis vinkelen mellom dem er for stor eller liten. Liten regnes som ca  $2^\circ$  og stor vinkel er større enn  $10^\circ$ . Likevel anbefaler vi å benytte minst tre bilder. Dette er gjennomført med relativt gode visuelle resultater for prosjektet *Meierikrysset*.

Under prøver og forsøk med prosjektet *Meierikrysset*, hadde vi 2 forskjellige bildesett med stor vinkelforskjell mellom settene (luft og bakke). Programmet klarte å inkludere 1 bilde fra luft sammen med bildene fra bakke.

For å generere teksturert modell fra SFM Toolkit, benyttes MeshLab. Her genererer vi en TIN vha. en Poisson-algoritme, som prøver å tette modellen mest mulig. Best resultat fikk vi med denne varianten. Enda bedre ble det når vi gjorde om trekantene til firkanter vha. et eget filter i MeshLab. Likevel hadde det vært fint med en gjengivelse av tekstur med kvaliteten fra bildene. Man kan også benytte en "Ball Pivot"-algoritme som bare lager trekanter mellom

punkter med en viss avstand. Deretter tekstureres hver trekant. Dette gir veldig mye ”blurry” overganger mellom trekantene og gjør at en naturlig kant, som f.eks. kanten på et vindu, kan bli noe støyete og lite definert. Dette kommer av at trekantnettet ikke jevnes ut på noe vis.

I tillegg har vi oppdaget en vesentlighet ved tekstureringen som gir mere markerte skiller i teksturen ved f.eks. dør- eller vinduskanter etc. Dette kan man manipulere i MeshLab ved å benytte filteret ”Tri to Quad by 4-8 Subdivision”. Dette filteret gjør om trekantene, i TIN, til firkanter og det kan virke som om man på denne måten får utnyttet teksturen bedre.



**Figur 21 Normal TIN fra MeshLab**



**Figur 22 Quadnettverk av samme modell**

### **5.2.2. Visual SFM**

Med tanke på punktskyene og modellene man får ut av programmet har vi rent visuelt ikke sett noen forskjeller fra SFM Toolkit.

Metodene for teksturering er de samme som for SFM Toolkit.

### **5.2.3. Autodesk 123D Catch**

Vi opplever at Autodesk 123D Catch klarer å lage svært autentiske 3D modeller. Modellene er svært detaljerte og har liten grad av fortegning. Programmet klarer å gjengi objektene veldig bra, både hva kanter og tekstur gjelder. Spesielt teksturen klarer programmet å gjengi

svært bra. Tabell 3 side 54 viser dette. Når vi imidlertid zoomer inn i modellen, får vi ikke tekstur som er av bildekvalitet.

De utvendige trappehusene har homogene overflater, men likevel klarer programmet å håndtere disse. Trappene har vært en utfordring for både SFM Toolkit og PM. Dette er et eksempel på at programmet fyller alle mulige hull i punktskyen veldig bra.

Programmet krever 3 bilder for å klare å lage en modell. Figur 23 viser at resultatet av en slik prosess nødvendigvis ikke blir bra. Se spesielt på søppelkonteineren som er plassert på 2 steder i bildet.

Vi har testet flere variasjoner av bilder. Programmet klarer de fleste av disse. Vi har blant annet lastet opp bilder tatt fra kamera med forskjellig kvalitet. Disse har programmet klart å bruke i 3D modellen. Kvaliteten på modellen kan imidlertid bli litt dårligere når vi har hatt for mange forskjellige typer bilder. Dette vises ved at teksturen blir litt mer diffus når vi har bilder fra forskjellige kamera.

Vi har ikke funnet beskrivelser som sier noe om vinkelen mellom bildene. Vi har testet dette og det virker som om programmet kan klare bilder med opp til ca. 40 grader mellomrom. Da kan imidlertid resultatet bli som i Figur 23.



**Figur 23 3D modell generert av Autodesk 123D Catch med 3 bilder**

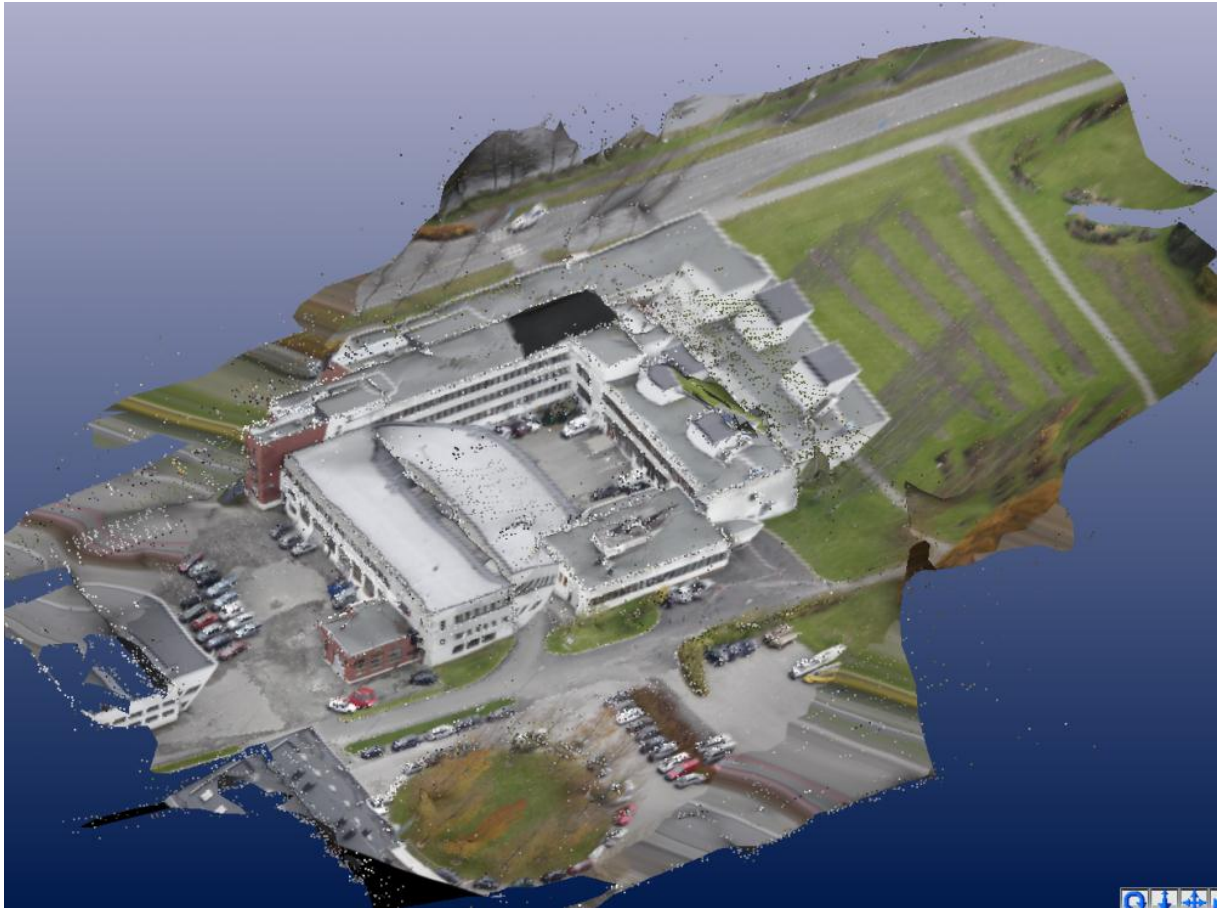
Vi har også prøvd å kombinere bilder tatt fra bakke og fra luft (Meierikrysset). Dette lot seg ikke gjøre, selv om vi manuelt prøvde å binde sammen bildene i Autodesk 123D Catch.

Sannsynligvis er dette fordi det er for stor vinkel mellom bildene. I tillegg var bildene tatt på forskjellige tidspunkter på året, høst og vår, med forskjellig radiometriske forhold.



#### 5.2.4. PhotoModeler

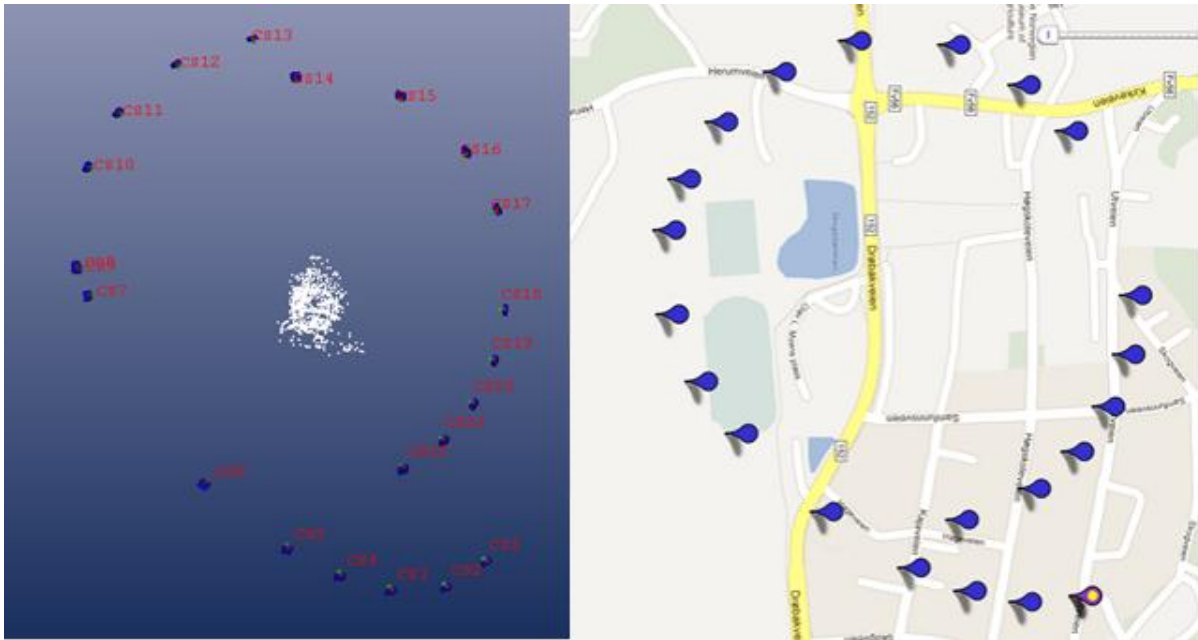
I PM er det som sagt modulen *Dense Surface* som genererer selve 3D modellen. Vi opplever modulen som vanskelig å bruke. Den inneholder svært mange valgmuligheter, for mange etter vår mening. Brukernivået skal være svært høyt for at modellene skal bli bra.



Figur 24 3D modell generert av Photomodeler

Figur 24 viser klart at det kan oppstå problemer under genereringen av modellene. Sannsynligvis er det bilde(r) i settet som er orientert feil. Denne modellen er dog en av de tetteste modellene vi har fått til i PM. De andre har hatt mange hull.

Vi har ikke fått bra resultater under disse prosessene, og det er vanskelig å si hvorfor det blir slik. Vi har testet flere forskjellige innstillinger på parameterne uten at dette hjelper. Det er med andre ord svært vanskelig å si hva vi gjør feil i denne prosessen siden resultatene ikke blir bra. Spesielt takflatene er vanskelig for programmet. Dette skyldes nok de store homogene flatene. Det kan virke på oss som om det er begrensninger i programmet. Om det er algoritmene som behandler punktskyen dårlig, eller om det er orienteringen, er ikke lett for oss å si. Men at orienteringen av kameraene har noe å si er det ikke tvil om.



**Figur 25 Kameraposisjoner vs GPS- posisjoner**

Figur 25 viser hvordan modellkoordinater kan avvike fra de absolutte posisjonene.

Modellkoordinatene er generert av PM gjennom et automatisk prosjekt. Dette er kjørt med lav tetthet. Av figuren kan vi lett se at posisjonene ikke er riktige. Se spesielt CS7, CS8 og CS9.






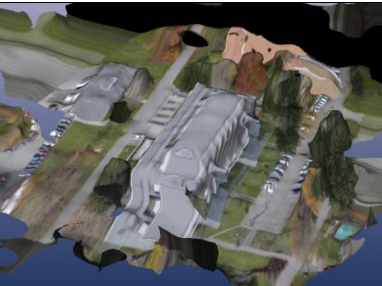




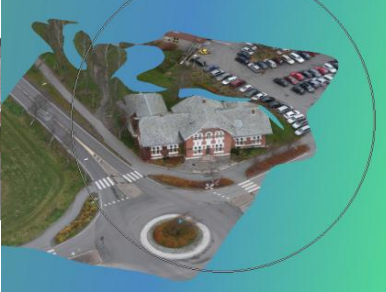

Disse ligger for tett. Se også CS13 og CS14<sup>6</sup>. Alle disse bildene er tatt fra helikopter under samme runde. Med slik orientering vil ikke resultatet i andre enden bli bra heller.

<sup>6</sup> Henvisning til bildefil: Alle bildene er fra serien TF-1 med prefiks 20111026\_Rygge\_XXXX. CS7=0196.jpg. CS8=0198.jpg. CS9=0200.jpg. CS13=0208.jpg. CS14=0210.jpg



### 5.2.5. Visualisering av modellene

Tabell 3 viser genererte 3D modeller av prosjektene. Denne viser forskjellene programmene i mellom. En oversikt over hvilke kamera som er nyttet samt antall bilder finnes i tabell 2 og vedlegg B.

	SFM Toolkit	Autodesk 123D Catch	PhotoModeler
TF			
Sh			
F3			
M			

Tabell 3 Genererte 3D modeller

### 5.2.6. Sammenligning av relativ orientering vs GPS-koordinater

Vi testet kameraposisjonene opp i mot GPS-dataene. Med perfekt orientering og perfekte GPS-koordinater ville en transformasjon mellom modell- og GPS- koordinater ikke gi residualer når transformasjonen ble gjennomført. Vi valgte prosjektet Sørhellinga på grunn av at vår oppdragsgiver mente GPS-dataene var best her. Vi gjennomførte en Helmert 7-parameter transformasjon fra modellkoordinatene til GPS-koordinatene. Det vi si at vi prøvde å transformere modellkoordinatene slik at disse passet overens med GPS-koordinatene. Denne ga oss kvalitetsdata på transformasjonen. Vi gjennomførte en transformasjon med alle 17 bildene som utgangspunkt. Vi satte ikke noen posisjoner som ”fasit”. Resultatene av denne transformasjonen ble som følger:

	Maximum spatial residual (m)	Average spatial residual (m)	Root mean Square (RMS) of residuals (m)
SFM Toolkit	5.9567	2.3334	1.7862
Visual SFM	44.9074	21.2593	15.3710
Autodesk 123D Catch	5.5619	2.1240	1.6458
PhotoModeler	4.6038	2.5550	1.7941

Tabell 4 Resultater av 7-parameter transformasjon

Når vi gjør dette, må vi være klar over mange begrensninger i denne undersøkelsen. Blant annet vil nøyaktigheten på GPS-posisjonene ha mye å si for en resultatet av undersøkelsen. Nøyaktigheten på disse er av normal kvalitet. Beskrivelse finnes i Vedlegg C.

I tillegg gjennomførte vi en test med skalerte modellkoordinater. Koordinatene er skalert opp med felles referanseavstand slik at disse skal bli så lik GPS-koordinatene som mulig. På denne måten kan vi i tillegg teste disse opp i mot hverandre. Med denne metoden kan vi antyde om det er et program som skiller seg fra de andre. Resultatene ble som følger for Sørhellinga:

Fra\Til	SFM Toolkit	Visual SFM	Autodesk 123	PhotoModeler
SFM Toolkit		15.5715	0.3514	0.7109
Visual SFM	15.5546		15.0411	13.6094
Autodesk 123	0.3549	15.1574		1.3936
PhotoModeler	0.8029	15.3770	1.5602	

Tabell 5 Resultater fra 7-parameter transformasjon med skalerte koordinater for Sørhellinga. RMS i meter.

Vi gjennomførte flere undersøkelser med andre bildesett, med varierende resultat. Vi har med bakgrunn i de forskjellige resultatene, vært usikre på kvaliteten på transformasjonen som ble gjennomført. Skal denne testen gjennomføres ved en senere anledning må man være 100 % sikre på at transformasjonen gjennomføres riktig, samt at modellkoordinatene er riktige. I vårt tilfelle har vi tvilt på både resultatet fra shareware-versjonen Sevenpar 2.10 og på modellkoordinatene fra Visual SFM. For Visual SFM kan dette dreie seg om vanskeligheter med å finne modellkoordinatene (se kapittel 5.3.2). Transformasjonen gjennomført i Visual SFM med hensikt å transformere modellkoordinatene til ECEF (benyttet GPS-koordinater som i Sevenpar 2.10) ble kontrollert ved å ta ut koordinater på punkt i den tette punktskyen og deretter plassert i Google Earth (se Figur 29). Disse ser ut til å stemme. Derfor er våre antagelser at det er deler av prosessen i Sevenpar 2.10 som ikke fungerer etter våre ønsker for undersøkelsen.

### **5.2.7. Delkonklusjon**

Våre observasjoner av de visuelle punktskyene og modellene er at samtlige programmer er i stand til å gjenskape modeller (tett punktsky) som ser veldig virkelighetsnære ut. Autodesk 123D Catch gir en fullstendig modell med veldig bra tekstur. Hvert fall så lenge vinkelen mellom bildene ikke er for stor. Gjennom en enkel prosess i MeshLab gjenskapes modeller med tekstur fra de tette punktskyene generert av SFM Toolkit og Visual SFM.

I utgangspunktet får vi god geometri av objektene vi har modellert. Punktskyene blir tette og fine, med unntak av takflatene i PhotoModeler. Et generelt problem er de områdene hvor 2 objekter står plassert nære hverandre. For eksempel hvis et tre står plassert inntil veggen. Hvis veggen er beskrevet med få punkter vil treet bli tatt med som en del av veggen ved generering av TIN. Veggen mister i disse tilfellene sin virkelige form. Dette er noe vi anser som et generelt problem innenfor alle rekonstruksjoner. Dette er ikke et problem når programmene klarer å generere nok punkter av hvert objekt.

Vi vil ikke konkludere med noe angående våre undersøkelser av modellkoordinatene. Dette er fordi undersøkelsen i seg selv er svært liten, og fordi vår erfaring med Helmert 7-parameter transformasjon er for tynn.

### 5.3.Skalering av 3D-modell

Som beskrevet i kapittel 3.5 har vi 4 metoder for skalering. Disse er som følger:

- SM 1. Skalering vha GPS-data som ligger i bildenes EXIF-fil.
- SM 2. Skalering vha for eksempel [www.norgeibilder.no](http://www.norgeibilder.no)
- SM 3. Skalering vha målte punkter i terrenget
- SM 4. Gjennomsnittlig skalar

Vi vil beskrive hvordan disse metodene kan benyttes, og ikke, i de forskjellige programmene.

#### 5.3.1. SFM Toolkit

Vi fikk til å benytte metodene SM2 og SM4. For SM2 benyttet vi "Norge i bilder" og fikk veldig bra resultater vha. av disse. Selve skaleringen er veldig enkel da man måler den ønskede avstanden i MeshLab direkte i modellen. Deretter dividerer man referanseavstanden på den målte og ender opp med en skaleringsfaktor. Denne settes så inn i en egen funksjon i MeshLab og modellen er skalert.

SM4 ble benyttet på modellene av TF/IMT og Sørhellinga på følgende måte: Vi bygde opp et relativt enkelt MATLAB-script (Vedlegg på CD) som leser inn en liste med GPS-posisjonene i latitude, longitude og høyde. Lat/lon gis i grader. Deretter regnes disse om til kartesiske koordinater.

I PMVS2's resultat katalog finner man en liste med kameraposisjonene angitt i modellkoordinater. Denne leses også inn og benyttes slik den er. Her er det viktig å forvise seg om at rekkefølgen i modell- og GPS-listene er like. PMVS2 har en tendens til å gi bildene en ny nummerering uavhengig av bildenes originale nummerering. Hvis bildene ikke ligger i samme rekkefølge som originalbildene, må koordinatlisten sorteres riktig før man gjør beregningene. Dette er arbeid som tar ganske lang tid.

I MATLAB beregnes avstander mellom alle de mulige kombinasjonene av kameraposisjoner i henholdsvis modell- og GPS-koordinater, hver for seg. Disse havner i hver sin  $n \times 1$ -matrise slik at rekkefølgen på avstandene er like disse imellom. Deretter finner vi skaleringsfaktoren for hver avstand ved å dividere GPS- på modellavstander. Dette gir en ny vektor med alle skaleringsfaktorene og vi kan deretter trekke ut en gjennomsnittsverdi fra denne, med et standardavvik. Dette gjennomsnittet er da skaleringen vi setter inn i MeshLab, som vi benyttet i forlengelsen av SFM Toolkit med PMVS2 for teksturering og måling. Standardavviket

benyttet vi for å avdekke eventuelle sorteringsfeil fra PMVS2's kamera liste. Terskel for standardavviket må ses i forhold til størrelse på skalar og naturlig standardavvik for denne verdien. Altså med en stor grad av skjønn.

Etter at dette scriptet ble laget, gikk tidsbruken ned mye, men en funksjon/script som henter ut kameraposisjoner fra PMVS2-filen og sorterer dem automatisk, vil gjøre dette til en veldig effektiv metode. Selve skaleringen av modellen fungerer til slutt akkurat som beskrevet for SM2.

Når vi beregnet skalar med standardavvik for hver av modellene TF/IMT og Sørhellinga, fikk vi et høyere standardavvik for TF/IMT (Tabell 6). Denne forskjellen kommer sannsynligvis av noe så enkelt som at noen få posisjoner hadde større unøyaktighet enn resten av settet enten i settet av GPS-posisjoner eller i relativ posisjonene til kameraene. Eventuelle ustabiliteter i GPS-dataene vil, ved metoden vi her har benyttet, jevnes godt ut siden vi har midlet over et stort antall kombinasjoner.

Prosjekt	Antall kombinasjoner	Skalar	Standardavvik skalar
<b>TF/IMT</b>	595	71.596	9.254
<b>Sørhellinga</b>	136	96.093	0.586

Tabell 6 Verdier for skalering ved bruk av SM4

### 5.3.2. Visual SFM

Visual SFM kan benytte alle de forskjellige metodene for skalering.

Skalering ved hjelp av SM4 fungerer på samme måte som for SFM Toolkit, men her er det enda vanskeligere å finne ut hvordan programmet har nummerert om bildene slik at de havner i en annen rekkefølge. Dette har i noen tilfeller tatt så lang tid at vi har sett oss nødt til å droppe det.

Man kan gjøre en transformasjon av koordinatsystemet i programmet. Dette menes at man kan sette inn kartesiske koordinater for kameraposisjonene eller punkt i modellen for så å gjennomføre en transformasjon av hele punktskyen. Dette gjøres da også med den tette punktskyen. Vi har testet dette veldig lite, men det ser ut som en lovende funksjon selv om vår test ikke ga gode koordinater for punktene i selve modellen. Dette kan skyldes dårlig geometri, fordi objektet kun var fotografert fra 180°. Vi antar at koordinatene ville vært mere riktig hvis man hadde posisjoner 360° rundt objektet eller at vi hadde satt inn koordinater for bakkepunkter i tillegg.

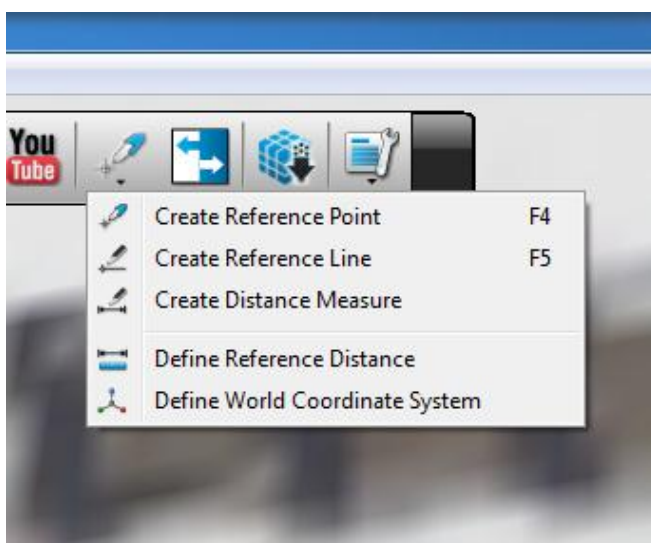
Ved at man har støtte for å ta inn GPS-data, vil man kunne benytte disse til å skalere og rotere modellen slik at det er mulig å kunne måle både koordinater og avstander. Denne funksjonen har vi prøvd ut og sett at den fungerer veldig bra. Et stort minus er at programmet flytter på hvert enkelt punkt i den tette punktskyen slik at man får en modell som til en viss grad blir deformert. Her hadde det vært ønskelig at transformasjonen av modellen ble utført etter at modellen var ferdig med TIN og tekstur. Dette fenomenet ga også store problemer for målefunksjonen i MeshLab da selve punktene i punktskyen så ut til å bli skalert opp slik at målefunksjonen låste seg til senterpunktet på selve punktet man skulle måle fra. På denne måten kan miste opptil flere desimeter på målenøyaktigheten.

Et minus er at man ikke har funksjon for å gjøre avstandsmålinger i programmet. Dette må gjøres i f.eks. MeshLab.

### 5.3.3. Autodesk 123D Catch

Det er fullt mulig å skalere modellen slik at målestokken blir riktig i Autodesk 123D Catch. Dette kan gjøres ved hjelp av SM2 og SM4. Ved hjelp av SM2 så er dette svært enkelt. SM4 krever mere jobb. Her må man eksportere en fbx.fil, åpne denne i en teksteditor, sammenligne kamerakoordinatene til modellen med de absolutte posisjonene, for til slutt å skalere en avstand med gjennomsnittlig skalar.

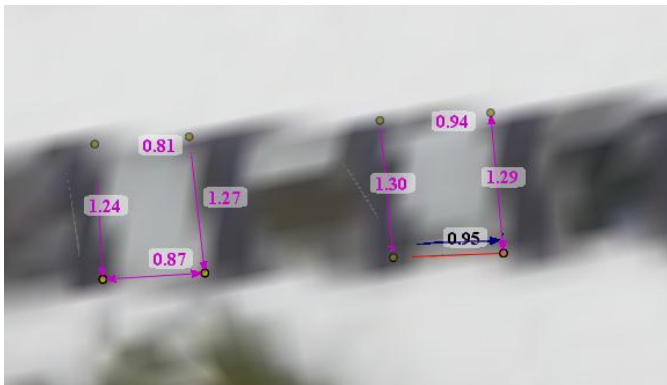
For å overføre denne til modellen må det markeres 2 referansepunkter. Deretter kan målestokken defineres ved å bruke kommandoen *Define Reference Distance*.



Figur 26 Måle meny Autodesk 123D Catch

En positiv side med Autodesk 123D Catch er nøyaktigheten som oppnås når referansepunkter skal settes. Dette gjøres i de enkelte bildene og ikke i selve modellen. Dette er selvfølgelig avhengig av orientering og oppløsning i bildene.

Framgangsmåten er som følger: Punktet defineres i et bilde. Deretter kan man klikke på andre bilder for å se og flytte punktet. Da får man se punktet fra en annen vinkel og man får da flyttet dette til det riktige stedet. Epipolarlinjene fra de andre bildene vises også, noe som gjør det enda enklere. Vår erfaring er at det er en stor fordel og definere punkter som minst finnes i 3 bilder, gjerne med bra vinkel mellom hvert bilde. Selv om dette gjøres, kan det være vanskelig å plassere punktene helt riktig i bildene. Dette er noe en uerfaren bruker må være klar over. Figur 27 viser resultatet av dette. Målingene er gjort av vinduer med samme størrelse.



Figur 27 Målinger i Autodesk 123D Catch

#### 5.3.4. PhotoModeler

PM kan benytte alle de forskjellige metodene for skalering. Det som er spesielt med PM er modulen *External geometry*. I denne er det mulig å legge til punkter med kjente koordinater. Vi la til de kjente punktene vi målte inn med CPOS. Når dette var gjort, fikk vi en modell som var ferdig rotert, skalert og plassert, uten at modellen endret seg visuelt. Det er også mulig å legge inn koordinatene i forskjellige koordinatsystemer og kartprojeksjoner. Vi la inn våre i WGS 84 UTM. Koordinatene regnes deretter om til jordsentriske koordinater.

Kameraposisjonene beregnes i jordsentriske koordinater (ECEF).

PhotoModeler har samme målefunksjonalitet som i Autodesk 123D Catch. Først må det etableres referansepunkter (i flere bilder), deretter kan avstanden måles.

### **5.3.5. Delkonklusjon**

Visual SFM og PhotoModeler er begge i stand til å skalere modellene med alle 4 metodene. SFM Toolkit og Autodesk 123D Catch var kun i stand til å benytte 2 av metodene. Men alle programmene var i stand til å skalere modellene på en enkel måte.

Vi har sett at måleverktøyet i både MeshLab, Autodesk 123D Catch og PhotoModeler framstår som fornuftige og enkle i bruk. En kombinasjon av disse vil være ønskelig for et komplett program, spesielt i tilfeller der teksturen er utydelig i modellen slik at det blir vanskelig å definere detaljer. Da er det en stor fordel å kunne sette målene ved hjelp av 2D-bildene. Måling direkte i modellen er derimot en raskere metode. Både Autodesk 123D Catch og PhotoModeler lar lengdemålinger som er gjennomført lagres i både modellen og bildene. Dette er en funksjon som vi savner i MeshLab, da dette vil være gunstig i tilfeller der man skalerer modellen på nytt, eller bare for å visualisere ferdige mål i modellen eller i bilder.

## **5.4. Utnyttelse av observerte størrelser**

Med utnyttelse av observerte størrelser, mener vi om det er mulig å bruke GPS/INS fra sensoren i programmet og om modellen plasseres, skaleres og roteres riktig hvis disse observerte størrelsene utnyttes i programmet? Med dette mener vi data som kan importeres inn manuelt eller data som finnes i bildenes EXIF-fil.

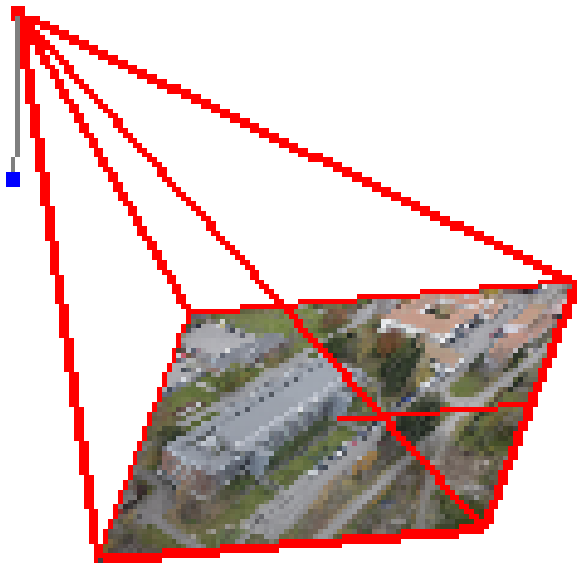
### **5.4.1. SFM Toolkit**

SFM Toolkit gir ingen annen mulighet for bruk av observerte størrelser enn ved å benytte metoden med å beregne en gjennomsnittlig skalar (beskrevet under "Metode-Skalering").

### **5.4.2. Visual SFM**

Som nevnt tidligere kan man i Visual SFM ta inn GPS-data via EXIF-dataene til bildene eller en tilhørende tekstfil med lat/lon og høyde. Dette fungerer veldig greit spesielt med bruk av EXIF der programmet leser inn alle posisjonene på en gang før de blir gjort om til ECEF-koordinater. Estimert avvik mellom den relative kameraposisjonen og GPS-posisjonen visualiseres i programmet (Figur 28). Deretter kan man gjøre en transformasjon av hele modellen.





**Figur 28** Avvik mellom relativ kameraposisjon(rød) og GPS-posisjon(blå).

Man kan også tilordne nye koordinater til punkter i modellen manuelt (kapittel 5.3.2, side 58). Disse angis da som XYZ-koordinater. Per nå kan man ikke velge hvilket format man kan lese ut koordinater fra modellen. Disse angis kun i ECEF. Disse funksjonene virker lovende og vi anbefaler å følge med på utviklingen av dem.



**Figur 29** Sammenligning av koordinater hentet fra modell (høyre), kontrollert (punkt 2) i Google Earth (venstre)

En kontroll av transformasjonsfunksjonen i programmet ble gjennomført ved å hente ut koordinater fra modellen, transformert til ECEF. Disse konverterte vi over til lat/lon og satte de inn i Google Earth. Disse ser visuelt ut til å stemme meget bra, men vi har ikke gjort undersøkelser som gir mere detaljerte data på nøyaktigheten. Vi mener at denne funksjonen er meget bra, men bør raffineres videre for å bli optimal.

### **5.4.3. Autodesk 123D Catch**

Autodesk 123D Catch klarer ikke å utnytte kjente størrelser som posisjon eller rotasjon. Dette er ikke en del av menyene i programmet. Det er heller ikke mulig å endre disse i en teksteditor for så å laste inn modellen senere.

### **5.4.4. PhotoModeler**

PhotoModeler klarer å utnytte kjente størrelser på en svært bra måte. Modulen *External Geometry* er enkel å bruke, oversiktlig og med muligheter til å utnytte bildenes EXIF-data. Det eneste negative er at dette ikke kan gjennomføres automatisk. Se kapittel 5.1.4.

Vi opplevde denne modulen som svært nyttig i vårt arbeid.

### **5.4.5. Delkonklusjon**

Vi antar at nøyaktigheten på GPS/INS, på enkle luft- og bakkeplattformer, vil øke betraktelig i framtiden. Dette gjør at slike størrelser i framtiden kan utnyttes i stor grad i 3D-modellering. Visual SFM og PhotoModeler er begge i stand til å utnytte observerte størrelser, dog med varierende grad av enkelhet og automatisering. Visual SFM viser at det er mulig å gjøre dette svært enkelt for brukeren.

Hvis alle programmene hadde hatt slike moduler ville vi hatt muligheten til å evaluere nøyaktigheten på en annen måte. Ved hjelp av de signalerte punktene ville dette vært mulig.

## 5.5.Hvilken nøyaktighet kan oppnås

### 5.5.1. Resultater

Tabell 7 viser beregnet RMSD for alle målingene.

Prosjekt \ Program	SFM Toolkit	Autodesk 123D Catch	PhotoModeler
Sørhellinga	0,07 (SM2)	0,09 (SM2)	0,07 (SM1)
Fløy 3	0,02 (SM2)	0,06 (SM2)	0,10 (SM3)
Meierikrysset	0,02 (SM2)	0,07 (SM2)	0,07 (SM2)
TF/IMT	0,06 (SM2)	0,13 (SM2)	0,11 (SM3)
Totalt	0,05	0,10	0,10

Tabell 7 RMSD (meter) av alle målinger (Skaleringsmetode)

I utgangspunktet er alle modellene skalert med SM 2, referansemål fra [www.norgebilder.no](http://www.norgebilder.no). For PhotoModeler sin del testet vi andre muligheter. Lengdene vi målte endret seg ikke selv om vi endret skaleringsmetode. Dette var mulig å kontrollere fordi målingene var tilgjengelig selv om skaleringen endres<sup>7</sup>.






Under målinger med SFM Toolkit gjennomførte vi målinger på TF/IMT- og Sørhellinga modellene basert på GPS-data som grunnlag for skaleringen (SM4). Vi mener å se at vi fikk en bedre nøyaktighet på målingene for Sørhellinga modellen. Dette kan komme av at dataene for kameraposisjonene over Sørhellinga hadde bedre kvalitet, selv om grunnlaget er for lite til å konkludere med dette. Mere sannsynlig er det at det er gjort små forskjeller i målingene på de to modellene.

Prosjekt \ Program	SFM Toolkit med GPS	SFM Toolkit uten GPS
Sørhellinga	0,03	0,07
TF/IMT	0,07	0,06
Totalt	0,06	0,065

Tabell 8 RMSD (meter) av målinger med SFM Toolkit med og uten GPS skalering

<sup>7</sup> Dette ble også sjekket med helt feil skalering for å kontrollere utfallet.

Figur 30 viser at den absolutte posisjoneringen til modellen kan være feil. I dette tilfellet er GPS-data i bildene brukt til å transformere modellen. Skaleringen blir riktig, men posisjonen blir feil.

A..	V...	Name	Id	X (m)	Y (m)	Z (m)	Delta m
<input checked="" type="checkbox"/>		Check point 8	Pt-36	3172399.27	603801.60	5481852.98	7.96
<input checked="" type="checkbox"/>		Check point 9	Pt-33	3172412.41	603784.24	5481851.97	6.83
<input checked="" type="checkbox"/>		Check point 10	Pt-30	3172446.35	603792.34	5481831.56	6.36
<input checked="" type="checkbox"/>		Check point 11	Pt-27	3172453.00	603806.06	5481828.01	6.13
<input checked="" type="checkbox"/>		Check point 12	Pt-24	3172443.24	603802.39	5481835.55	6.18

Figur 30 Check Point med høy feil for TF/IMT

### 5.5.2. Feilkilder

Den viktigste feilkilden er det faktum at vi er 2 som har gjennomført målingene i forskjellige programmer. Ideelt sett burde begge målt i alle 3 programmene, men det lot seg ikke gjennomføre på grunn av tid til disposisjon.

Skaleringen av modellene vil også være en feilkilde i målingene. Vi har brukt samme referanselengde i alle prosjektene vi har gjennomført slik at feilen mellom prosjektene skal være minimalisert. Vi har brukt så lange referanselengder som mulig for å få best mulig skalering. Korte referanselengder vil ha et større potensial for feil i skalering enn lange referanselengder.

Vi erfarte at det å måle direkte i modellen fungerte veldig bra, men det forutsetter litt lett analyse av modellen for å finne ut hvor kanter og andre detaljer faktisk befinner seg hvis teksten er litt dratt utover. Her hadde det vært ønskelig med en funksjon der man i tillegg kan måle i selve bildene. (ref. Autodesk 123D Catch). Måling direkte i modellen i tilfeller der avstanden/detaljen man skal måle er noe udefinert, kan gi noe feil ved plassering av målepunkter. Det har videre vist seg at hvis man skal måle på en detalj med tekstur som er uklar, kan man snu modellen slik at man ser den fra innsiden. Da vil man kunne se tydeligere hvor start og endepunkt må være. Dette har gitt noe nøyaktigere målinger, men ikke i en veldig betydelig grad. Vi har ingen annen forklaring på dette fenomenet enn at man kanskje får se detaljen fra et annet perspektiv.

For å begrense feilen i målingene i størst mulig grad har vi som sagt gjennomført flere målinger der det har vært hensiktsmessig (i all hovedsak SFM Toolkit). I de programmene hvor det må plasseres referansepunkter i bildene før en måling kan gjøres (Autodesk 123D

Catch og PhotoModeler), har vi gjort en vurdering underveis på hvor mange målinger vi gjorde. I de tilfellene hvor vi bare har gjennomført en måling har vi prøvd å knytte referansepunktene opp i flest mulig bilder.

Selv om vi gjennomfører flere målinger så er det ikke alltid at resultatet blir fornuftig likevel.

Figur 31 viser et eksempel på forskjellen i målene som man kan få.



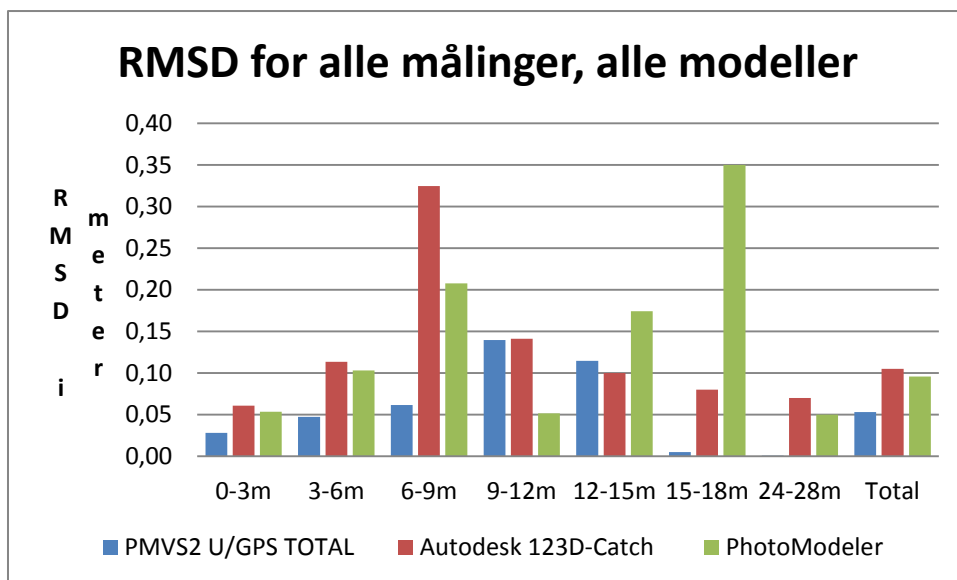
**Figur 31** Målinger på Meieribygget i PhotoModeler

For SFM Toolkit ble nøyaktigheten på målingene veldig mye bedre når vi endret TIN til et firkantnett (kapittel 5.2).

### 5.5.3. Delkonklusjon

Alle programmene gir totalt sett en meget tilfredsstillende nøyaktighet. Dette kan man se i Tabell 7. Statistikk for hver enkelt modell og kombinasjoner av forskjellige skaleringsmetoder kan finnes i Vedlegg på CD.

Tabell 7 viser at SFM Toolkit gir jevnt over mindre feil enn de to andre programmene. For Autodesk 123D Catch og PhotoModeler bedømmer vi likevel nøyaktigheten til å være meget bra. En del av forskjellene mellom programmene kan skyldes at vi har hatt hovedfokus på hver våre program, men vi anser virkningen av dette som at vi begge vil klare å få det maksimale ut av disse og at dette fenomenet er neglisjerbart i så måte. Det som derimot vil ha noe å si er at vi har jobbet lenge med de samme modellene og vil derfor kunne være i stand til å få bedre resultater enn en bruker som gis en kort introduksjon. Dette bør det gjennomføres bredere testing på for å kunne gi en sikrere statistikk.



Figur 32 Sammenligning av RMSD for 3m-intervaller og totalt for alle program

Vi klarer ikke å si sikkert at det er noen klar trend basert på forskjellene mellom de ulike intervallene og programmene. Statistikken for intervallet 0-3 m er det sikreste da brorparten av lengdene lå innenfor dette og man kan se store likheter mellom RMSD for intervallet og for totalen. For intervallene 9-12 m og opp var det få lengder. Dette gjør at en dårlig måling på disse, gir et stort utslag for det aktuelle intervallet.

Vi mener ut fra forsøk med og uten GPS-data for skalering at SFM Toolkit gir bedre resultater ved bruk av slik data. Dette er imidlertid noe som også bør testes mere på siden denne statistikken kun er basert på to modeller.

## 5.6. Andre data og bruksområder

### 5.6.1. Kamera kalibrering

For SFM Toolkits vedkommende får man, vha. av Bundler, ut indre- og ytreorienteringsparametere. De indre orienteringsparameterne finner Bundler vha. selvkalibrering. Disse dataene har flere bruksområder enn for den videre rekonstruksjonen. F.eks. en kalibrering av et kamera tenkt brukt til andre formål. Parameterne man får ut av Bundler er  $f$ ,  $k_1$  og  $k_2$  for hvert enkelt bilde. PMVS2 benytter disse for å gjøre en oppretting av hvert bilde slik at den radielle fortegningen blir lik 0. Dataene er relativt lett tilgjengelige. For hvert bilde angis en  $3 \times 3$ matrise for rotasjon ( $R$ ) og kameraets plassering i modellkoordinater ( $XYZ$ ).

Parameterne fra Bundler ( $f$ ,  $k_1$  og  $k_2$ ) angis i piksler og ikke i mm. [35]

Visual SFM gir omtrent de samme dataene som SFM Toolkit, med unntak av at det angis kun en verdi for radiell fortegning. Full data for disse er i Vedlegg E

I PhotoModeler finner vi, som sagt tidligere, kamerakalibreringsdata i *Project Status Report*. Rapporten gir bare en verdi for de forskjellige parameterne for hele prosjektet, i forhold til SFM Toolkit som gir verdier for hvert bilde. Disse angis i mm. Rapporten for Sørhellinga ligger i vedlegg L.

Autodesk 123D Catch gir kun *focal length*. Denne finnes i prosjektets FBX.fil og er angitt i mm.

For å konvertere brennvidden( $f$ ) mellom piksler og mm har vi tatt utgangspunkt i denne formelen:

$$f_{mm} = \frac{f_{piksel} \cdot \text{Sensorbredde}_{mm}}{\text{Bildebredde}_{mm}}$$

Tabell 9 viser verdiene for Focal length for de forskjellige programmene.

Program		Focal length( $f$ ) i mm
SFM Toolkit	Verdi	51.36
	Standardavvik	0,22
Visual SFM	Verdi	51,48
	Standardavvik	0,11
PhotoModeler	Verdi	51.49
	Standardavvik	0,10
Autodesk 123D Catch	Verdi	51.34

Tabell 9 Sammenligning av kalibreringsdata

Vi viser kun Focal length da vi ikke får data som kan sammenlignes for de andre parameterne. Som man ser av resultatene gir de ulike programmene forskjellige verdier for  $f$ . Dette kan skyldes to faktorer. Det første er at programmene gjør selvkalibrering med varierende kvalitet. Det andre er sammenhengen mellom  $f$  og radiellfortegningsverdier ( $K1$  og  $K2$ ). Hvis  $f$  er forskjellig mellom programmene så vil også  $K1$  og  $K2$  være det. Disse 3 parameterne nyttes sammen i hvert program. Ideelt sett skal de ulike kombinasjonene beskrive det samme.

PhotoModeler er et program som tradisjonelt er kjent for å gi gode kalibreringsdata ved hjelp av flere forskjellige metoder. Det er rimelig å anta at de andre programmene gjør en kalibrering som er basert på å fange opp de groveste deformasjonene.

### 5.6.2. Sammenligning av *Feature detection* og matchingsalgoritme

SFM Toolkit og Visual SFM er de eneste programmene hvor vi med sikkerhet vet hvilken algoritme (SIFT) som gjennomfører *Feature detection*. Programmene gir relativ enkel tilgang til disse dataene. I PhotoModeler gjennomfører som sagt funksjonen SmartMatch *Feature detection*. Tabellen i Vedlegg F en sammenligning av SIFT og SmartMatch. Prosjektet som er brukt som grunnlag er Sørhellinga (De 5 første bildene ble brukt). Resultatene kan tyde på at SmartMatch ikke benytter algoritmen SIFT på grunn av store ulikheter i antall punkter som ble funnet. SIFT fant ca. 40 % færre punkter enn SmartMatch. Det kan være innstillinger i algoritmene som har en effekt på dette, men det er ikke noe vi endrer på.

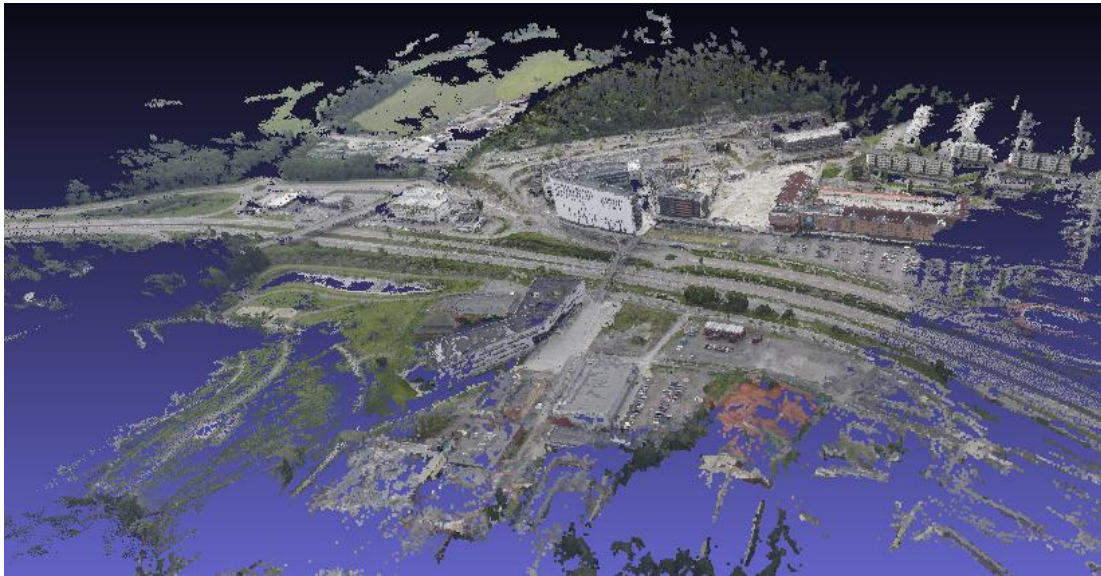
Resultatene fra matchingen mellom sammenbindingspunktene bildene i mellom tyder på at det er forskjellig matchingsalgoritme som ligger til grunn. Ved små vinkler (under 12 bilder), finner SmartMatch flere match enn Visual SFM. Ved større vinkler finner Visual SFM flere match.

### 5.6.3. SFM Toolkit

Programmet kan helt klart benyttes til å generere lokale overflatemodeller, ortofoto eller 3D-modeller over et større område. Dette kan begrunnes i at man kan benytte veldig mange bilder og at punktskyen som genereres inneholder veldig lite støy. Samtidig har man tilgang på den tette punktskyen og denne kan tas inn i software spesialisert for slik bruk. Vi har testet et bildesett på 107 bilder av det nye rådhuset på Lørenskog. Dette ga en punktsky som dekker et relativt stort område. Her antar vi at det kan gi større effekt hvis man benytter ”Fish-eye”-objektiv og tilnærmet loddrett fotografering. Eksempler av hvor stort område man dekker



vha. denne mengden bilder er vist i Figur 33 og Figur 34. Dette er et tilfelle der man har fotografert 360° rundt det store, hvite hesteskoformede huset i senter av eksempelbildene, altså ikke med områdekartlegging som hovedformål, men likevel vil man dekke et stort areal.



Figur 33 Punktsky over Lørenskog



Figur 34 Poissonmodell av Lørenskog

#### 5.6.4. Visual SFM

Programmet har en egen funksjon for å finne repeterende objekter i 2D-bilder. Denne funksjonen kan være et praktisk hjelpemiddel ved analyse av komplekse bilder. Vi har ikke satt oss videre inn i denne.

### 5.6.5. Autodesk 123D Catch

Programmet i seg selv er kun beregnet på å generere automatiske 3D-modeller fra bilder.

Det vil sannsynligvis være lett å kombinere dette med andre programmer fra Autodesk 123D-serien. Dette har vi testet i Autodesk's 3DSMax. Autodesk har i tillegg programmer beregnet på manuell modellering og 3D-print av objekter.

De fotogrammetriske mulighetene er relativt begrenset i programmet. Som sagt tidligere virker det som om programmet er mest beregnet på underholdning i stedet for fotogrammetri. Men vi ser av Figur 35 at det er mulig å få til overflatemodeller og ortofoto av lokale områder ved hjelp av modeller fra programmet.



Figur 35 Rådhuset på Lørenskog med Autodesk 123D Catch

### 5.6.6. PhotoModeler

PhotoModeler er et stort program med mange bruksområder. Disse står beskrevet på nettsidene til PM. Programmet er nok primært tenkt brukt på mindre objekter enn det vår oppdragsgiver ser for seg i tillegg til manuell modellering.




















### 5.6.7. Delkonklusjon

Resultatene fra *Feature detection* og matchingen mellom bildene tyder på at det er forskjellige algoritmer som ligger til grunn for programmene.

SFM Toolkit og Visual SFM gir muligheten til å generere lokale overflatemodeller. Autodesk 123D Catch kan primært nyttes til 3D-modellering, visualisering av modellen og lokale overflatemodeller. PhotoModeler kan nyttes til manuell modellering.

### 5.7. Oppsummering av resultater

Vi har vurdert de enkelte testresultatene og satt de inn i tabellen under.

	Test 1	Test 2	Test 3	Test 4	Test 5
SFM Toolkit					
Visual SFM					
Autodesk 123D Catch					
PhotoModeler					

Tabell 10 Oppsummering av resultater





Test 1 Grad av automatisering

Test 2 Ivaretagelse av geometri og tekstur

Test 3 Skalering av 3D-modellene

Test 4 Utnyttelse av observerte størrelser

Test 5 Hvilken nøyaktighet kan oppnås

Symbol	Forklaring
	Meget bra
	Bra, men noe å utsette
	Mindre bra/Ikke fullstendig funksjon
	Dårlig/fullstendig manglende

Tabell 11 Forklaring på symboler

Tabell 10 viser at totalt sett kommer programmene bra ut i våre undersøkelser. Flere av programmene får høyeste eller nest høyeste vurdering på flere testkriterier. Det er vanskelig å påpeke hvilket program som kommer best ut totalt sett. En kombinasjon ville definitivt vært å foretrekke. Automatiseringen og teksturen til Autodesk 123D Catch, PhotoModelers *External Geometry* og SFM Toolkits nøyaktighet ville vært den perfekte kombinasjon.

En bruker vil spare veldig mye tid på å bruke GPS-data automatisk til skalering av modellen, slik det gjøres i Visual SFM. Graden av enkelhet ved dette vil være svært høy. Lokalt sett vil nok ikke nøyaktigheten på GPS-dataen ha stor innvirkning på skaleringen av modellen. En eventuell absolutt posisjonering av modellen vil nødvendigvis ikke være riktig, selv om lengdemålinger i modellen er det. Tester vi har gjort i PhotoModeler beviser dette.

Alle programmene er i stand til å håndtere bilder med ulik skalering og fra ulike kamera. Dette er en stor fordel da man kan kombinere forskjellige bildesett.

På grunn av at Autodesk 123D Catch er web-basert, er det vanskelig å vurdere hastigheten på programmene mot hverandre. Men vi kan konkludere med at alle klarer å lage modeller relativt raskt.

## 6. KONKLUSJON

I vår problemstilling stilte vi spørsmålet om det er mulig å lage 3D-modeller automatisert med utgangspunkt i bilder tatt med enkle sensorer som f.eks. mobilkamera og speilrefleks. Våre undersøkelser viser at dette er fullt ut mulig å gjøre, relativt raskt, med meget bra resultater for flere programmer. Spesielt programmet Autodesk 123D Catch beviser dette. Programmet gjennomførte en fullt ut automatisert prosess, hvor spesielt teksturen i sluttresultatet var imponerende. De andre programmene vi testet var også automatisert, men ikke i samme grad som Autodesk 123D Catch. De gjennomførte ikke steget fra en tett punktsky til en ferdig teksturert modell automatisk. Dette må per nå gjøres i et eget program. Likevel vil man være i stand til å få ut modeller som er godt brukbare for Forsvarets behov, men det vil kreve mer av en eventuell bruker. Etter vår vurdering har programmene SFM Toolkit og Visual SFM et stort potensiale. Vi antar at de vil komme i en ny og forbedret utgave hvor teksturering kan være inkludert i programmet. Dette er basert på informasjon fra utviklernes nettsider. En stor fordel med disse to programmene er at man kan gå inn og sette andre verdier for hvordan rekonstruksjonen skal gjennomføres. Dette kan i flere tilfeller være en stor fordel der man har bildesett som ikke gir ønskede resultater vha. defaultverdier. Denne muligheten finnes ikke i Autodesk 123D Catch.

I våre studier av prosessen 3D-rekonstruksjon har vi fått et innblikk i den utvikling som har vært de siste 10-20 årene. Denne har etter vår mening vært meget stor og har bidratt til produkter som ikke var mulig tidligere. Utfordringene har i store trekk dreid seg om å håndtere store datamengder, automatisere prosessene, samt å raffinere algoritmene som gjennomfører *dense image matching*. Utfordringen med å automatisk finne ytre og indre orienteringsparametere har man klart å løse. Veien videre, fra en spredt til en tett punktsky og til en teksturert overflatemodell, jobbes det kontinuerlig med.

Våre tester viser at kravene til sensorene ikke behøver å være store. Det er fullt ut mulig å gjennomføre en 3D rekonstruksjon med bilder tatt av kameraer med lav kvalitet. Det en bruker imidlertid må være klar over før fotografering er kravene som stilles til bildesekvensene for at en rekonstruksjon skal være gjennomførbar. Vi tenker da spesielt på vinkler og overlapping mellom bildene.

Skalering, rotasjon og plassering av 3D modeller er en viktig del av vår oppdragsgivers ønske for et program. Programmene vi har testet har alle enkle metoder for å skalere modellene. Den enkleste metoden er å utnytte eventuelle data i bildenes EXIF-fil. Denne egenskapen innehar

kun Visual SFM og PhotoModeler, med forskjellig grad av automatisering og resultat. Begge viser at bruken av et geodetisk datum er fullt mulig i 3D-modeller.

Alle programmene vi testet innehar nødvendig funksjonalitet for å gjennomføre målinger i modellene. Men det er forskjell i måtene dette ble gjort på. Den ene metoden er å måle direkte i modellen, mens den andre er å bruke bildene for å gjennomføre målingene. Det er fordeler og ulemper med begge, og vi ser helst at et eventuelt nytt program innehar muligheter for å gjennomføre målinger på begge måtene.

Ingen av programmene våre innehar alle de kvalitetene vår oppdragsgiver etterspør. Imidlertid har programmene totalt sett alt som ønskes. Selv om vi ikke har funnet et som vi mener er et ideelt program til nå, så er vi sikker på at dette i løpet av noen år vil finnes på markedet.

Til slutt er det viktig å understreke at vi mener alle programmene vil kunne gi verdifulle produkter for Forsvaret, under både skarpe operasjoner og trening. Også med den funksjonalitet de innehar per nå.

Vår oppgave har gitt oss en meget god innsikt i hva som kreves for å gjennomføre en automatisk 3D-modellering. Fordi vi er 2 som har jobbet sammen har vi klart å se på hele prosessen i stedet for små deler av denne.



## 7. VEIEN VIDERE

Denne oppgaven har gitt oss en dyp innsikt i prosessen som må gjennomføres for å få til en 3D-rekonstruksjon. Vi har også gjort oss opp noen tanker om hva som bør utvikles videre i framtiden for eller av vår oppdragsgiver.

Gjennom SFM Toolkit og Visual SFM har vi fått testet bildematchingsalgoritmen PMVS2. Selv om vi ikke har funnet eller fått tilgang til konkret dokumentasjon på de 2 siste programmene vi testet, er det ting som tyder på at disse også bruker samme algoritme. Dette kan bety at vi egentlig har testet samme algoritme opp i mot seg selv, bare med forskjellig innpakning. Vi kunne derfor ønske å teste en av de andre algoritmene. Av det som finnes, virker spesielt programpakken APERO-MICMAC spennende. I artikkelen til M. Pierrot Deseilligny beskrives programmet. Det virker som om kombinasjonen SIFT-APERO-MICMAC kan være verdt å se spesielt nærmere på. Denne softwaren er per nå basert på å bli kjørt ved hjelp av en XML-fil og det opplyses fra utviklerne at det er en komplisert prosess å kjøre programmet. MAP-Gamsau har tatt på seg å utvikle et brukergrensesnitt som skal være mer betjeningsvennlig og er på nettsidene anslått å være ferdig i utgangen av 2011. En slik GUIbasert løsning er per starten av Mai 2012 tilgjengelig for Linux og Mac. Disse ligger på MAP-Gamsau sine nettsider.

Visual SFM gir en mulighet for å oppnå Forsvarets mål hvis man er i stand til å ta inn generering av overflater med tekstur, i tillegg til at man får inn mulighet for måling. Her kan det være verdt å la aktuelle brukere få gjøre noen tester for å se etter eventuelt flere mangler eller ønskede funksjoner.

Acute 3D opplyser på sine nettsider at Autodesk 123D Catch er basert på deres løsning "smart3Dcapture" som er en "stand-alone" software som de også er i stand til å justere og utforme etter brukerens krav. [29] Det som kreves for at Autodesk 123D Catch skal bli svært bra, er nettopp en «stand-alone» løsning samt muligheten for bruk av et geodetisk datum, gjerne gjennom automatisk bruk av bildenes EXIF-fil.

Vi har tidligere beskrevet at en kombinasjon av de forskjellige programmene ville gitt vårt «drømmesoftware». Som sagt så ønsker vi automatiseringen og teksturen som Autodesk 123D Catch gir. Vi ønsker mulighetene til å endre til geografiske referansesystemer som i PhotoModeler, dog med Visual SFM sin automatisering av dette. Vi ønsker punktskyen og nøyaktigheten som SFM Toolkit gir.

## Referanseliste

- [1] M. L. TH. KERSTEN, K. MECHELKE, K. ZOBEL. (2011, 01.05.2012). AUTOMATIC 3D OBJECT RECONSTRUCTION FROM UNSTRUCTURED DIGITAL IMAGERY FOR APPLICATIONS IN ARCHITECTURE, CULTURAL HERITAGE AND ARCAEOLOGY. AVAILABLE: [HTTP://WWW.LC3D.NET/PAPER/2011/KERSTEN\\_HCU\\_FOLIEN.PDF](HTTP://WWW.LC3D.NET/PAPER/2011/KERSTEN_HCU_FOLIEN.PDF)
- [2] Y.-F. WANG. (SEPTEMBER 8, 2011, 01.05.2012). A COMPARISON STUDY OF FIVE 3D MODELING SYSTEMS BASED ON THE SFM PRINCIPLES. AVAILABLE: [HTTP://WWW.VISUALSIZE.COM/TUTORIAL/VS\\_TR.PDF](HTTP://WWW.VISUALSIZE.COM/TUTORIAL/VS_TR.PDF)
- [3] S. KARTVERK, *ORDBOK FOR KART OG OPPMÅLING*, 1989.
- [4] R. LUHMANN, KYLE, HARLEY, *CLOSE RANGE PHOTOGRAMMETRY*: WHITTLES PUBLISHING, 2006.
- [5] M. PIERROT DESEILLIGNY. (2012). *APER0, AN OPEN SOURCE BUNDLE ADJUSTMENT SOFTWARE FOR AUTOMATIC CALIBRATION AND ORIENTATION OF SET OF IMAGES*. AVAILABLE: [HTTP://RECHERCHE.IGN.FR/LABOS/MATIS/PDF/ARTICLES\\_CONF/2011/PIERROT\\_CLERY.PDF](HTTP://RECHERCHE.IGN.FR/LABOS/MATIS/PDF/ARTICLES_CONF/2011/PIERROT_CLERY.PDF)
- [6] P. DESEILLIGNY. (2012, SOME POSSIBLE PROTOCOLS OF ACQUISITION FOR OPTIMAL USE OF THE "APER0" OPEN SOURCE SOFTWARE IN AUTOMATIC ORIENTATION AND CALIBRATION. AVAILABLE: <WWW.MICMAC.IGN.FR>
- [7] R. SZELISKI, "COMPUTER VISION ALGORITHMS AND APPLICATION," SEPTEMBER 3 2010.
- [8] D. LOWE. (2004, DISTINCTIVE IMAGE FEATURES FROM SCALE-INVARIANT KEYPOINTS. AVAILABLE: <HTTP://WWW.CS.UBC.CA/~LOWE/PAPERS/IJCV04.PDF>
- [9] K. B. A. S.F.EL-HAKIM, *CLOSE RANGE PHOTOGRAMMETRY AND MACHINE VISION*, 1996.
- [10] B. REMONDINO, NEX, SCAIONI, SARAZZI. (2011, UAV PHOTOGRAMMETRY FOR MAPPING AND 3D MODELING – CURRENT STATUS AND FUTURE PERSPECTIVES –. VOL. XXXVIII-1/C22, 7. AVAILABLE: <HTTP://WWW.ISPRS.ORG/PROCEEDINGS/XXXVIII/1-C22/PAPERS/REMONDINO.PDF>
- [11] M. GEHRKE, DOWNEY, BOEHRER, FUCHS, "SEMI-GLOBAL MATCHING: AN ALTERNATIVE TO LIDAR FOR DSM GENERATION?."
- [12] J. KIM. (2003, VISUAL CORRESPONDENCE USING ENERGY MINIMIZATION AND MUTUAL INFORMATION. AVAILABLE: <HTTP://WWW.CS.CORNELL.EDU/~RDZ/PAPERS/KKZ-ICCV03.PDF>
- [13] H. HIRSCHMÜLLER, "ACCURATE AND EF CIENT STEREO PROCESSING BY SEMI-GLOBAL MATCHING AND MUTUAL INFORMATION," 2005.
- [14] Y. FURUKAWA AND J. PONCE. (2008, 13.03). ACCURATE, DENSE, AND ROBUST MULTI-VIEW STEREOPSIS. 1. AVAILABLE: <HTTP://WWW.CS.WASHINGTON.EDU/HOMES/FURUKAWA/PAPERS/PAMI08A.PDF>
- [15] M. PIERROT DESEILLIGNY. (2012, SOME POSSIBLE PROTOCOLS OF ACQUISITION FOR OPTIMAL USE OF THE "APER0" OPEN SOURCE SOFTWARE IN AUTOMATIC ORIENTATION AND CALIBRATION. AVAILABLE: <WWW.MICMAC.IGN.FR>
- [16] PHOTOMODELER. *FACTORS AFFECTING ACCURACY IN PHOTOGRAMMETRY*. AVAILABLE: <HTTP://WWW.PHOTOMODELER.COM/KB/ENTRY/63/>
- [17] PHOTOMODELER. (2012). *PHOTOMODELER SCANNER - HOW IT WORKS*. AVAILABLE: [HTTP://WWW.PHOTOMODELER.COM/PRODUCTS/SCANNER/HOW\\_IT\\_WORKS.HTM](HTTP://WWW.PHOTOMODELER.COM/PRODUCTS/SCANNER/HOW_IT_WORKS.HTM)
- [18] PHOTOSYNTH. (2012). *CREATE YOUR OWN PHOTOSYNTHS - PHOTOSYNTH*. AVAILABLE: <HTTP://PHOTOSYNTH.NET/CREATE.ASPX>
- [19] K. SOFT. (2012). *SEVENPAR - CALCULATION OF HELMERT AND MOLODENSKY PARAMETER SETS - BRIEF DESCRIPTION*. AVAILABLE: [HTTP://WWW.KILLETSOFT.DE/P\\_SVPA\\_E.HTM](HTTP://WWW.KILLETSOFT.DE/P_SVPA_E.HTM)
- [20] ASTRE AND HENRI, "SFMTOOLKIT 3," VOL. 3, 3 ED. <WWW.VISUAL-EXPERIMENTS.COM>: ASTRE HENRI, 2011, P. STRUCTURE FROM MOTION TOOL KIT.



- [21] C. WU. (2012, 01.05). *VISUALSFM : A VISUAL STRUCTURE &#X9;&#X9;FROM MOTION SYSTEM*. AVAILABLE: <HTTP://WWW.CS.WASHINGTON.EDU/HOMES/CCWU/VFSM/>
- [22] "MESH LAB," IN *MESH LAB V1.3.0A*, 1.3.0A ED. <HTTP://MESHLAB.SOURCEFORGE.NET/>, 2011, PP. 3D-EDITOR.
- [23] N. SNAVELY, S. M. SEITZ, AND R. SZELISKI. (2012). *PHOTO\_ TOURISM.PDF (APPLICATION/PDF-OBJEKT)*. AVAILABLE: <HTTP://PHOTOTOUR.CS.WASHINGTON.EDU/PHOTO TOURISM.PDF>
- [24] Y. FURUKAWA. (2012, 01.02). *CLUSTERING VIEWS FOR MULTI-VIEW STEREO (CMVS)*. AVAILABLE: <HTTP://GRAIL.CS.WASHINGTON.EDU/SOFTWARE/CMVS/>
- [25] M. COLLEGE. (2009). *THE MIDDLEBURY COMPUTER VISION PAGES*. AVAILABLE: <HTTP://VISION.MIDDLEBURY.EDU/MVIEW/EVAL/>
- [26] I. L. MAGIC. (2012, 28.04). <HTTP://WWW.ILM.COM/>. AVAILABLE: <HTTP://WWW.ILM.COM/>
- [27] C. WU. (2011, 08.05). *MULTICORE BUNDLE ADJUSTMENT*. AVAILABLE: <HTTP://GRAIL.CS.WASHINGTON.EDU/PROJECTS/MCBA/>
- [28] AUTODESK. (2012). *AUTODESK 123D - 123D CATCH TURN PHOTOS INTO 3D MODELS*. AVAILABLE: <HTTP://WWW.123DAPP.COM/CATCH>
- [29] A. 3D. (2011). *SOLUTIONS | ACUTE 3D*. AVAILABLE: <HTTP://WWW.ACUTE3D.COM/SOLUTIONS/>
- [30] J. CHANDLER AND J. FRYER. (2012). *ACCURACY OF AUTODESK 123D CATCH?* AVAILABLE: <HTTP://WWW-STAFF.LBORO.AC.UK/~CVJHC/OTHERFILES/ACCURACY%20OF%20123DCATCH.HTM>
- [31] PHOTOMODELER. (2010). *3D FROM YOUR CAMERA: EOS SYSTEMS ANNOUNCES PHOTOMODELER 2010*. AVAILABLE: [HTTP://WWW.PHOTOMODELER.COM/ABOUT\\_US/NEWSRELEASES/PM2010REL-10-10-28.HTM](HTTP://WWW.PHOTOMODELER.COM/ABOUT_US/NEWSRELEASES/PM2010REL-10-10-28.HTM)
- [32] PHOTOMODELER. (2012). *PHOTOMODELER - PHOTOGRAMMETRY TECHNOLOGY*. AVAILABLE: [HTTP://WWW.PHOTOMODELER.COM/ABOUT\\_US/CORETECHNOLOGY.HTM](HTTP://WWW.PHOTOMODELER.COM/ABOUT_US/CORETECHNOLOGY.HTM)
- [33] PHOTOMODELER, *PHOTOMODELER - QUICK START GUIDE*, 5 ED.: EOS SYSTEMS INC, 2011.
- [34] (2012). *KAP DISCUSSION PAGE - KAP MAPPING GETS EASIER*. AVAILABLE: <HTTP://ARCH.CED.BERKELEY.EDU/KAP/DISCUSS/COMMENTS.PHP?DISCUSSIONID=2991>
- [35] N. SNAVELY, "BUNDLER V0.4 USER'S MANUAL," ED, 2008-2009.

## **VEDLEGG**

## VEDLEGG

A.	FORDELING AV ARBEIDET.....	- 1 -
B.	BILDE LISTE.....	- 2 -
C.	GPS KOORDINATER MED NØYAKTIGHETS DATA.....	- 3 -
D.	KOORDINATER TIL SIGNALERTE POSISJONER .....	- 4 -
E.	KALIBRERINGSDATA.....	- 6 -
F.	ALTERNATIVE SAMMENBINDINGSPUNKT.....	- 8 -
G.	ANTALL MATCH MELLOM BILDEPAR .....	- 8 -
H.	PROSEDYRE FOR Å KJØRE SFM TOOLKIT:.....	- 9 -
I.	VEILEDNING FOR INNSTILLINGER PMVS2 .....	- 11 -
J.	EKSEMPEL PÅ OPTIONS-FIL FRA PMVS2 .....	- 13 -
K.	PHOTODELER SMART MATCH REPORT .....	- 13 -
L.	PHOTODELER PROJECT REPORT(DELER AV DENNE).....	- 17 -
M.	AUTODESK 123D-CATCH FBX.FIL (DELER AV DENNE).....	- 21 -

### A. Fordeling av arbeidet

I henhold til formelle krav ved UMB, må det beskrives i oppgaven hvordan fordelingen av arbeidet har vært mellom 2 studenter, når disse har skrevet Masteroppgave sammen.

Utgangspunktet vårt har vært å gjøre mest mulig sammen, men vi har gjort noen unntak. Disse er som følger:

- Jørgen har beskrevet programmene SFM Toolkit og Visual SFM, og gjennomført testing på disse med beskrivelse av resultater. Dette inbefatter Bundler, CMVS/PMS, og Photosynth Toolkit 7.
- Stein Arne har beskrevet Autodesk 123D Catch og PhotoModeler, og gjennomført testing av disse med beskrivelse av resultater.
- Jørgen har hatt hovedansvar for kap. 2.2 og kap. 2.3.2 når det gjelder teori, men Stein Arne har også fylt inn med en betydelig mengde underveis.
- Stein Arne har hatt hovedansvar for kap. 2.1, kap. 2.3.1 og kap. 2.3.3 når det gjelder teori, men Jørgen har også fylt inn med en betydelig mengde stoff underveis.
- Resterende kapitler er i stor grad skrevet i samarbeid.

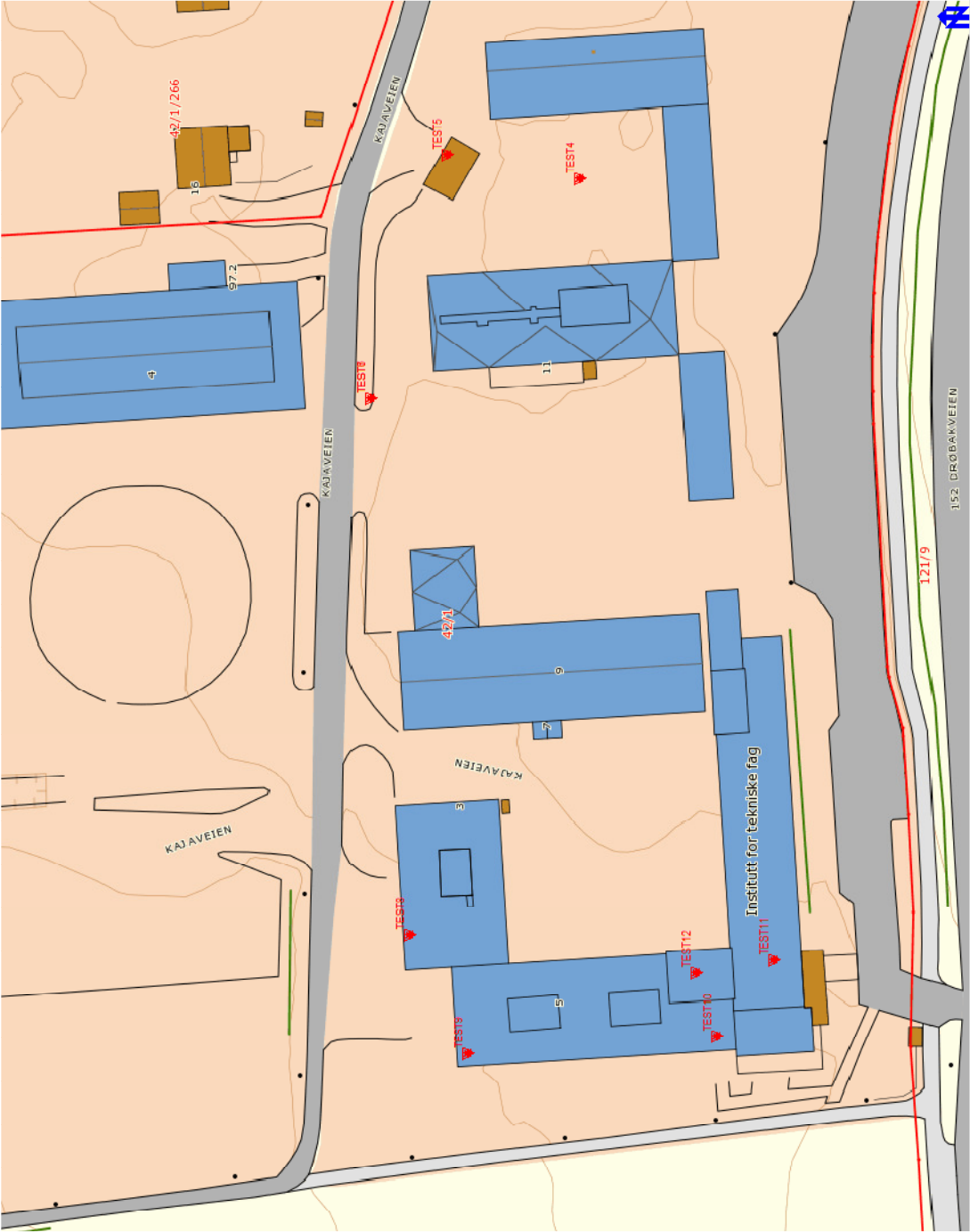
## B. Bilde liste

<b>TF-1</b>	<b>Floy-3</b>	<b>Sørhellinga</b>	<b>Meierikrysset</b>
0184	DSC09410	Rygge_0043	IMG_4545
0186	DSC09411	Rygge_0044	IMG_4546
0188	DSC09412	Rygge_0045	IMG_4547
0190	DSC09413	Rygge_0046	
0192	DSC09414	Rygge_0047	
0194	DSC09415	Rygge_0048	
0196	DSC09416	Rygge_0049	
0198	DSC09417	Rygge_0050	
0200	DSC09418	Rygge_0051	
0202	DSC09419	Rygge_0052	
0204	DSC09420	Rygge_0053	
0206		Rygge_0054	
0208	DSC09470	Rygge_0055	
0210	DSC09471	Rygge_0056	
0212		Rygge_0057	
0214	DSC09552	Rygge_0058	
0216	DSC09553	Rygge_0059	
0218			
0220			
0222			
0224			
0226			
0228			
0230			
0232			
0234			
0236			
0238			
0240			
0242			
0244			
0246			
0248			
0250			
0252			

### C. GPS koordinater med nøyaktighets data

<b>Bildnr.</b>	<b>Lat i grader</b>	<b>Lon i grader</b>	<b>Høyde i m</b>
43	1,5435125E-06	1,5407175E-06	4,849305
44	1,5432675E-06	1,5411375E-06	4,851370
45	1,5428800E-06	1,5416375E-06	4,853180
46	1,5423825E-06	1,5421825E-06	4,854665
47	1,5418625E-06	1,5427400E-06	4,855835
48	1,5414575E-06	1,5433175E-06	4,856705
49	1,5416325E-06	1,5447375E-06	4,856380
50	1,5415400E-06	1,5443950E-06	4,853775
51	1,5413300E-06	1,5441200E-06	4,851590
52	1,5411050E-06	1,5437750E-06	4,846598
53	1,5411100E-06	1,5435875E-06	4,844215
54	1,5411625E-06	1,5433400E-06	4,842005
55	1,5414733E-06	1,5426633E-06	4,838593
56	1,5415900E-06	1,5422175E-06	4,837575
57	1,5416650E-06	1,5410700E-06	4,837265
58	1,5414925E-06	1,5400925E-06	4,838975
59	1,5415725E-06	1,5399425E-06	4,840550

D. Koordinater til signalerte posisjoner



Punkter målt inn ifm. fotografering av TF 26.10.11.

#### UTM-koordinater

TEST4	6615501.274	600184.339	95.878
TEST5	6615522.816	600188.327	96.163
TEST6	6615535.191	600148.759	96.019
TEST7	6615535.189	600148.757	96.018
TEST8	6615528.905	600061.767	101.709
TEST9	6615519.542	600042.504	105.721
TEST10	6615479.216	600045.195	105.706
TEST11	6615469.907	600057.684	107.234
TEST12	6615482.524	600055.563	108.553

#### ECEF-koordinater

TEST4	3172399.80859867580	603925.71178681310	5481832.33465941720
TEST5	3172380.92340307260	603926.76358268887	5481843.40564388410
TEST6	3172376.79842395570	603886.03996283456	5481850.06624882390
TEST7	3172376.79996202000	603886.03816530365	5481850.06440276750
TEST8	3172399.27267786950	603801.59841221711	5481852.97910282020
TEST9	3172412.41246522260	603784.23667867435	5481851.97375676500
TEST10	3172446.35382473240	603792.33723448636	5481831.55921271820
TEST11	3172452.99927115300	603806.06083034817	5481828.00800853870
TEST12	3172443.24208099860	603802.38854013453	5481835.54690567500

## E. Kalibreringsdata

Data for SFM Toolkit (Bundler)

Nr.	f i piksler	k1 i piksler	k2 i piksler
43	2942,91085940000	-0,122487830460000	0,0776618361840000
44	2925,72790500000	-0,146435725930000	0,219452847540000
45	2923,27404110000	-0,136678842800000	0,154158373690000
46	2916,30721380000	-0,129332899910000	0,112717546200000
47	2910,45523130000	-0,130418151040000	0,146319978790000
48	2910,64501900000	-0,124829598970000	0,0759100018760000
49	2905,53809270000	-0,130154029320000	0,166221360590000
50	2902,71601220000	-0,124944862030000	0,0937675505420000
51	2904,32000410000	-0,126267973580000	0,104537083980000
52	2940,35105510000	-0,135208978310000	0,181229374520000
53	2935,94706240000	-0,132459215710000	0,172108208810000
54	2930,58836260000	-0,135735625330000	0,184599439430000
55	2931,97385420000	-0,124654492030000	0,0996318506600000
56	2924,89456230000	-0,121226651500000	0,0757572023970000
57	2921,26175500000	-0,1345514444400000	0,209667443700000
58	2921,90468500000	-0,142959136060000	0,256695385830000
59	2924,82509720000	-0,138694085380000	0,203322621570000

Gjennomsnittsverdier:

f i piksler	k1 i piksler	k2 i piksler
2921.979	-0.134	0.149

Standardavvik for Bundlerverdier

f i piksler	k1 i piksler	k2 i piksler
12.309	0.0072	0.0564



Verdier for Visual SFM:

Bilde	f i piksler	k1 i piksler
43.jpg	2932,67260742000	0,0921429584778000
44.jpg	2926,40576172000	0,0943472563239000
45.jpg	2935,66162109000	0,0992071279365000
46.jpg	2926,12768555000	0,0976729212943000
47.jpg	2937,34423828000	0,0991086523453000
48.jpg	2938,94897461000	0,100043990806000
49.jpg	2935,24926758000	0,0988992251572000
50.jpg	2931,97192383000	0,102827757038000
51.jpg	2927,52099609000	0,113166412453000
52.jpg	2928,76806641000	0,117148889805000
53.jpg	2927,91650391000	0,116038453532000
54.jpg	2925,48706055000	0,116698150866000
55.jpg	2926,53442383000	0,115229826695000
56.jpg	2920,54663086000	0,125793933319000
57.jpg	2918,55200195000	0,131669341079000
58.jpg	2920,93286133000	0,133323246458000
59.jpg	2924,46875000000	0,116596285091000
Snitt:	2928.536	0.110
Std.av:	5.988	0,013

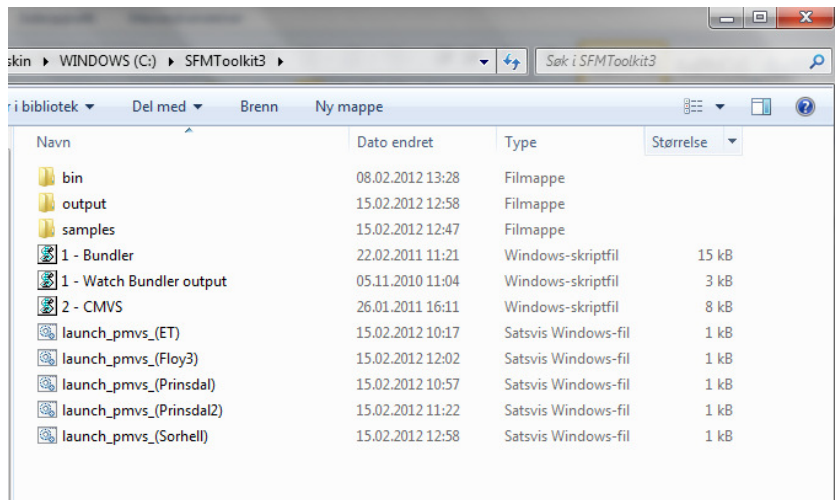
## F. Alternative sammenbindingspunkt

	Visual SFM(SIFT)	PhotoModeller Scanner
	Antall punkt	Antall punkt
Bilde 1	8409	13229
Bilde 2	7771	13330
Bilde 3	7465	13449
Bilde 4	8241	13356
Bilde 5	8599	13298

## G. Antall match mellom bildepar

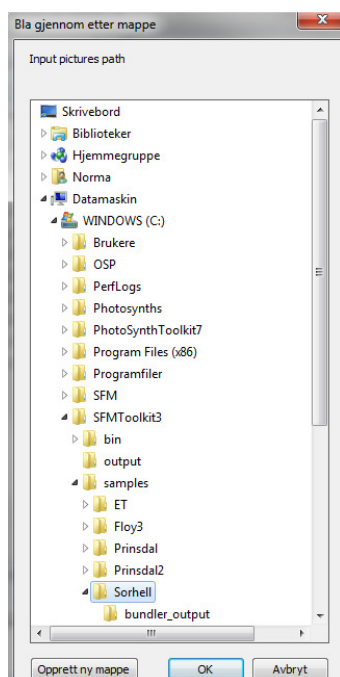
	Visual SFM(SIFT)	PhotoModeler Scanner
Kombinasjon	Antall match	Antall match
1/2	1559	1934 (500 brukt)
1/3	542	359
1/4	218	64
1/5	90	17
2/3	1507	2006 (500 brukt)
2/4	537	368
2/5	193	59
3/4	1568	1989 (500 brukt)
3/5	635	412
4/5	1825	2228 (500 brukt)

## H. Prosedyre for å kjøre SFM Toolkit:



Figur 1 Mappe man kjører SFM Toolkit fra

- Lag en prosjektmappe under samples-katalogen og kall den hva som helst uten å bruke mellomrom i navnet.
- Legg bildene du skal prosessere i denne. NB! Sjekk størrelse og navn på bildene. (kun 8 tegn i bildetittel)
- Dobbeltklikk på Bundler skriptfilen.
- Anvis mappen med bilder



- Deretter følger du instruksjonene som kommer opp (Veiledning for disse følger lengre ned).
- Deretter vil command-vinduet dukke opp og man kan følge med i prosessen. Her må man bare vente til vinduet lukkes av seg selv .
- Hvis man vil, kan man se på punktskyen som Bundler lagde med å trykke på Watch Bundler-output.
- Deretter trykk på CMVS-skriptet og følg programmet. Her er kan man stille inn ønskede parametre for prosessen, som påvirker kvalitet og hastighet på prosessen.
- Når dette skriptet er ferdig vil det komme en melding om at f.eks «Launch Sorhell.bat» er etablert.
- Hvis man vil stille inn andre parametre enn de man får mulighet til gjennom denne prosessen, kan man gå inn i PMVS-mappen, som er generert i bildemappen, og åpne filen(e) som heter "option-000(0-X).fil" i f.eks Notepad. Da kan man endre parametrene direkte.
- Dobbelklikk på launch-skriptet og se at commandvinduet kommer opp igjen.
- Nå er det bare å vente til dette vinduet legger seg ned før du er ferdig. Rekonstruksjonsprosessen itereres tre ganger. Det er mulig å følge stegene i prosessen i cmd-vinduet.
- Nå kan man åpne f.eks. Mesh Lab og ta inn .ply-fila fra PMVS2-folderen.

## I. Veiledning for innstillinger PMVS2

Navn	Default verdi	Forklaring
<i>timages</i>	Inkluderer alle mulige bilder som liste	<p><i>timages</i> står for <i>target images</i>. PMVS2 prøver å generere 3D-punkt vha av alle bildene som er tatt med som target-bilder. Hvis man vil kan man spesifisere disse bildene på to måter:</p> <ul style="list-style-type: none"> <li>• <b>Liste:</b> Etter "<i>timages</i>", angir et positivt tall, antallet bilder I prosjektet etterfulgt av indekseringen av hvert enkelt bilde. Indeksene starter med 0. Eksempel "<i>timages</i> 5 1 3 5 7 9" betyr 5 target-bilder, med indeksene "1 3 5 7 9".</li> <li>• <b>Range:</b> Etter "<i>timages</i>", vil det være tre siffer. Det første må være "-1", for å skille et fra liste og de siste to angir "fra og med-til". Eksempel "<i>timages</i> -1 0 6" betyr at target-bildene er "0, 1, 2, 3, 4 og 5". Bilde "6" er da ikke med.</li> </ul>
<i>oimages</i>	Må spesifiseres	<p><i>oimages</i> står for <i>other images</i>, som også angir hvilke bilder som skal inngå i rekonstruksjonen. Forskjellen fra <i>timages</i> genererer 3D-punkt helt til punktene er dekket av alle target-bildene, men ikke <i>oimages</i>. <i>Brukes</i> brukes til å forbedre nøyaktigheten til rekonstruksjonen. Hvis man ikke har bruk for disse setter man verdien til 0. Ellers lister man dem på lik linje med target-bildene. Denne har vi aldri stilt noe på.</p>
<i>level</i>	1	<p>Software lager bildepyramider. Denne verdien beskriver hvem av pyramidene som skal benyttes.</p> <p>Level 0:- Full oppløsning            Level 1:- Halvert oppløsning (4 ganger færre piksler)            Level 2:- 4 ganger mindre oppløsning (16 ganger færre piksler)            Default:- Level 1</p> <p>Ved å øke level setter man opp prosesseringshastigheten betraktelig, mens punktskyen blir mindre detaljert. Vi benytter konsekvent level 0 siden vi er ute etter mest mulig detaljer.</p>
<i>csize</i>	2	<p><i>csize</i> (cell size) styrer tettheten på rekonstruksjonen. Software prøver å generere en patch(3D-punkt) innenfor <i>csize</i> x <i>csize</i> region i alle målbilder(<i>timages</i>).</p> <p>Ved å øke denne verdien får man med færre detaljer i punktskyen. Vi har stort sett benyttet 1 og 2, men hvis man kjører mange bilder med full størrelse, anbefales det å bruke helt opp til 5.</p>
<i>threshold</i>	0.7	<p>En patch regnes som god nok hvis verdien som angir dens fotogrammetriske konsistenshet ligger over tresholdverdien. Dvs at patchen blir beholdt videre i prosessen. Normalisert krysskorrelasjon blir brukt som målenhet og verdiene ligger</p>

		<p>mellom -1(dårlig) og 1(god). Programmet itererer tre ganger og tresholdverdien senkes med 0.05 etter hver iterasjon. Eks. Hvis vi benytter <i>threshold=0.7</i>, blir verdiene 0.7, 0.65 og 0.6 benyttet.</p> <p>Her fungerer stort sett default 0.7 veldig bra, men hvis punktskyen blir tynn kan man prøve lavere verdier.</p>
<i>wsize</i>	7	<p>F.eks når <i>wsize=7</i>, betyr det at <math>7 \times 7 = 49</math> pikselfarger blir samlet i hvert bilde for å beregne den fotometriske konsistensveriden. Ved å øke denne verdien får man en mer stabil rekonstruksjon, men hastigheten går ned. Denne anbefales å stå på 9 eller 10 for vår type scener. Det gir en renere punktsky.</p>
<i>minImageNum</i>	3	<p>Dette angir hvor mange bilder et 3D-punkt må kunne finnes i. Dette vil ha mye med hvor god teksturen blir til slutt. Ved dårlig bildekvalitet anbefales det å benytte 4 eller 5, men programmet er i stand til å gjennomføre med kun 2. Denne har vi stort sett ikke justert noe på.</p>
<i>CPU</i>	4	<p>Dette brukes for å kunne benytte maskinens virituelle CPU. Denne justerer seg i forhold til maskinen den programmet kjøres på.</p>
<i>useVisData</i>	1	<p>PMVS2 kan bruke Bundler-filenes orienteringsdata til å finne ut hvilke bilder som overlapper hverandre nok til å kunne gjennomføre en fornuftig matching. Ved å sette verdien til 1 gjøres dette og hastigheten på rekonstruksjonen går betraktelig opp. Hvis verdien er 0 gjøres dette i en egen prosess og hastigheten går ned.</p>
<i>sequence</i>	-1	<p>Hvis man har fotografert i serie der bildenes plassering i rommet svarer til indekseringen kan man angi et maks antall nabobilder et bilde skal matches med. Dette kan i mange tilfeller være med å gi en renere model. F.eks. hvis <i>sequence=3</i>, kan bilde 5 matches med 2, 3, 4, 6, 7 og 8. Anbefalt ved bruk av full størrelse på bildene er verdi 1. Dvs. at et bilde matches mot det nærmeste bildet på hver side av seg.</p>
<i>quad</i>	2.5	<p>Programmet fjerner punkt som ikke er consistent med resten av modellen, f.eks. et punkt som ligger langt fra andre. Samtidig vil et punkt som ligger helt likt med nabopunktene også filtreres bort. F.eks i et plan. Dette er en tresholdverdi og fjerner mere jo lavere den er, sammtidig som hastigheten da går ned.</p>
<i>maxAngle</i>	10	<p>Denne verdien er angitt i grader. Verdien angir maksvinkelen mellom to projeksjoner. Ved å senke denne verdien vil man rekonstruere mere av bakgrunnen i bildene, men resultatet har en tendens til å bli mere støybelagt.</p>

[24]

## J. Eksempel på options-fil fra PMVS2

level 1  
csize 2  
threshold 0.7  
wsize 7  
minImageNum 3  
CPU 8  
setEdge 0  
useBound 0  
useVisData 1  
sequence -1  
timages 4 1 2 14 15  
oimages -3

## K. PhotoModeler Smart Match report

Status Report Tree

Project Name: \*\*\* Project has not yet been saved \*\*\*

Problems and Suggestions (2)

Project Problems (1)

Problem: The lowest angle separation between points is lower than 5 degrees and will not be computed accurately.

Suggestion: Points with low angle separation will not solve with good accuracy. If possible add a photo with greater angle separation.

Problems related to most recent processing (1)

Problem: There were more than 10 iterations in the most recent successful processing.

Suggestion: The large number of iterations (18) during processing indicates a problem in the project data: look for misreferenced points, incorrect constraints, or poor camera parameters. If there were green bars in the total error display or if there is some green text above, then these mistake may have already been corrected for you.

Information from most recent processing

Last Processing Attempt: Mon Apr 30 11:03:07 2012

PhotoModeler Version: 2012.0.0.586 - final,full

Status: successful

Processing Options

Orientation: on

Only unoriented photos oriented.

Number of photos oriented: 17

Global Optimization: off

Calibration: off

Constraints: off

Total Error

Number of Processing Iterations: 18

Number of Processing Stages: N/A

First Error: 0.675

Last Error: 1.729

Precisions / Standard Deviations

Quality

Photographs

Total Number: 17

Bad Photos: 0

Weak Photos: 0

OK Photos: 17

Number Oriented: 17

Number with inverse camera flags set: 0

Cameras

Camera1: Canon EOS 5D Mark II [50.00]

Calibration: yes

Number of photos using camera: 17

Average Photo Point Coverage: 84%

Photo Coverage

Referenced points outside of the camera's calibrated coverage region:

Photo1 points outside region:#335, #535, #678, #757, #800, #1476, #1525, #1576, #1601, #1645

Photo2 points outside region: more than 10 (check photo and calibration point coverage)

Photo3 points outside region: more than 10 (check photo and calibration point coverage)

Photo4 points outside region: more than 10 (check photo and calibration point coverage)

Photo5 points outside region: more than 10 (check photo and calibration point coverage)

Photo6 points outside region: more than 10 (check photo and calibration point coverage)

Photo7 points outside region:#4216



Photo8 points outside region:none

Photo9 points outside region:#4626, #5749, #5831, #5883

Photo10 points outside region: more than 10 (check photo and calibration point coverage)

Photo11 points outside region:#6903

Photo12 points outside region:#6903, #7512, #7010

Photo13 points outside region: more than 10 (check photo and calibration point coverage)

Photo14 points outside region: more than 10 (check photo and calibration point coverage)

Photo15 points outside region:#8125

Photo16 points outside region:none

Photo17 points outside region:none

#### Point Marking Residuals

Overall RMS: 0.921 pixels

Maximum: 2.100 pixels

Point 1418 on Photo 3

Minimum: 0.000 pixels

Point 1517 on Photo 1

Maximum RMS: 2.084 pixels

Point 2743

Minimum RMS: 0.000 pixels

Point 1517

#### Point Tightness

Maximum: 0.008

Point 8050

Minimum: 1.2e-007

Point 1517

#### Point Precisions

Overall RMS Vector Length: 0.0407

Maximum Vector Length: 0.188

Point 9167

Minimum Vector Length: 0.00399

Point 3910

Maximum X: 0.0698

Maximum Y: 0.159

Maximum Z: 0.0851

Minimum X: 0.0016

Minimum Y: 0.00228

Minimum Z: 0.00169

Point Angles

Maximum: 89.88 degrees

Point 3910

Minimum: 3.028 degrees

Point 9032

Average: 16.11 degrees

## L. PhotoModeler Project report(deler av denne)

### Status Report Tree

Project Name: SH v4.pmr

Problems and Suggestions (2)

Project Problems (2)

Problem: The largest point residual in your project (Point1068 - 28.12) is greater than 5.00. The largest residual of points used in processing is (Point405 - 19.01)

Suggestion: Your project has very high residuals and is not solving properly. In normal projects, strive to get all point residuals under 5.00 pixels. It is very important that this be fixed. If you have just a few high residual points, study them on each photo to ensure they are marked and referenced correctly. If many of your points have high residuals, make sure the camera stations are solving correctly and that you are using the best camera parameters possible.

Problem: The lowest angle separation between points is lower than 5 degrees and will not be computed accurately.

Suggestion: Points with low angle separation will not solve with good accuracy. If possible add a photo with greater angle separation.

Problems related to most recent processing (0)

Information from most recent processing

Last Processing Attempt: Wed May 02 14:14:53 2012

PhotoModeler Version: 2012.0.0.586 - final,full

Status: successful

Processing Options

Orientation: on

All photos oriented.

Number of photos oriented: 17

Global Optimization: on

Calibration: off

Constraints: on

Total Error

Number of Processing Iterations: 7

Number of Processing Stages: 2

First Error: 33.561

Last Error: 0.977

Precisions / Standard Deviations

Photograph Standard Deviations

Photo 1: 20111026\_rygge\_0043.jpg

Omega

Value: -102.904841 deg

Deviation: Omega: 0.055 deg

Phi

Value: 14.203279 deg

Deviation: Phi: 0.037 deg

Correlations over 95.0%: X:96.3%

Kappa

Value: 169.765618 deg

Deviation: Kappa: 0.047 deg

Xc

Value: 0.072105

Deviation: X: 2.3e-004

Correlations over 95.0%: Phi:96.3%

Yc

Value: 0.331586

Deviation: Y: 3.1e-004

Zc

Value: -0.477698

Deviation: Z: 3.3e-004

Quality

Photographs

Total Number: 17

Bad Photos: 0

Weak Photos: 0

OK Photos: 17

Number Oriented: 17

Number with inverse camera flags set: 0

Cameras

Camera1: Canon EOS 5D Mark II [50.00]

Calibration: yes

Number of photos using camera: 17

Average Photo Point Coverage: 84%

#### Photo Coverage

Referenced points outside of the camera's calibrated coverage region:

Photo1 points outside region:#421, #632, #893, #935, #978

Photo2 points outside region: more than 10 (check photo and calibration point coverage)

Photo3 points outside region: more than 10 (check photo and calibration point coverage)

Photo4 points outside region: more than 10 (check photo and calibration point coverage)

Photo5 points outside region:#1345, #1611, #2796, #3597, #3601, #3720, #3784, #3809, #3405

Photo6 points outside region: more than 10 (check photo and calibration point coverage)

Photo7 points outside region: more than 10 (check photo and calibration point coverage)

Photo8 points outside region: more than 10 (check photo and calibration point coverage)

Photo9 points outside region:#4609, #5008

Photo10 points outside region: more than 10 (check photo and calibration point coverage)

Photo11 points outside region: more than 10 (check photo and calibration point coverage)

Photo12 points outside region: more than 10 (check photo and calibration point coverage)

Photo13 points outside region: more than 10 (check photo and calibration point coverage)

Photo14 points outside region: more than 10 (check photo and calibration point coverage)

Photo15 points outside region:#8095, #8853, #9148

Photo16 points outside region:#8486

Photo17 points outside region:none

#### Point Marking Residuals

Overall RMS: 1.494 pixels

Maximum: 28.124 pixels

Point 1068 on Photo 3

Minimum: 0.000 pixels

Point 9182 on Photo 16

Maximum RMS: 16.279 pixels

Point 1068

Minimum RMS: 0.000 pixels

Point 9182

#### Point Tightness

Maximum: 0.00046

Point 5217

Minimum: 3.8e-008

Point 9182

#### Point Precisions

Overall RMS Vector Length: 0.000646

Maximum Vector Length: 0.00269

Point 8629

Minimum Vector Length: 6.38e-005

Point 4444

Maximum X: 0.00185

Maximum Y: 0.000827

Maximum Z: 0.0022

Minimum X: 3.44e-005

Minimum Y: 2.48e-005

Minimum Z: 4.39e-005

#### Point Angles

Maximum: 89.96 degrees

Point 518

Minimum: 2.402 degrees

Point 8486

Average: 17.24 degrees

## M. Autodesk 123D-Catch fbx.fil (deler av denne)

```
NodeAttribute: 51408824,
"NodeAttribute::/F:/Data_fra_Trym_FFI/20111026_Rygge/20111026_Rygge_0043.jpg", "Camera" {
    Properties70: {
        P: "AspectRatioMode", "enum", "", "",2
        P: "AspectWidth", "double", "Number", "",5616
        P: "AspectHeight", "double", "Number", "",3744
        P: "FilmWidth", "double", "Number", "",1.41732283464567
        P: "FilmHeight", "double", "Number", "",0.94488188976378
        P: "FilmAspectRatio", "double", "Number", "",1.5
        P: "ApertureMode", "enum", "", "",3
        P: "FocalLength", "Number", "", "A",51.3443369936551
        P: "NearPlane", "double", "Number", "",0.001
        P: "FarPlane", "double", "Number", "",1096.63330002665
        P: "FrontPlaneDistance", "Number", "", "A",783.309500019036
    }
    TypeFlags: "Camera"
    GeometryVersion: 124
    Position: 0,0,-50
    Up: 0,1,0
    LookAt: 0,0,0
    ShowInfoOnMoving: 1
    ShowAudio: 0
    AudioColor: 0,1,0
    CameraOrthoZoom: 1
```

Model: 51399848, "Model::\_20111026\_Rygge\_0043\_jpg", "Camera" {

Version: 232

Properties70: {

P: "PostRotation", "Vector3D", "Vector", "",0,-90,0

P: "RotationActive", "bool", "", "",1

P: "ScalingMax", "Vector3D", "Vector", "",0,0,0

P: "DefaultAttributeIndex", "int", "Integer", "",0

P: "Lcl Translation", "Lcl Translation", "", "A",20.0973,-163.583,139.299

P: "Lcl Rotation", "Lcl Rotation", "", "A",49.262741610461,-0.00526339444579944,0

P: "Lcl Scaling", "Lcl Scaling", "",  
"A",0.100000001490116,0.100000001490116,0.100000001490116