

Designspecifikation

Magnus Dahrén

Version 0.22

Granskad		
Godkänd		

PROJEKTIDENTITET

VT2006, AMASE

Linköpings tekniska högskola, ISY

Namn	Ansvar	Telefon	E-post
Mikael Karelid	kundansvarig (KUN)	073-7170360	mikka086@student.liu.se
John Wood	dokumentansvarig (DOK)		johwo077@student.liu.se
Magnus Dahrén	designansvarig (DES)		magda790@student.liu.se
Jon Månsson	testansvarig (TST)		jonma365@student.liu.se
Ove Svensson	kvalitetsansvarig (QS)		ovesv190@student.liu.se
Johan Hallenberg	projektledare (PL)	073-0330203	johha697@student.liu.se

E-postlista för hela gruppen: johha697@student.liu.se**Kund:** FOI**Kontaktperson hos kund:** Morgan Ulvklo, morgan@foi.se**Kursansvarig:** Klas Norberg, klas@isy.liu.se**Handledare:** Johan Wiklund, ISY och Anders Moe, ISY.

Accurate Multipoint Acquisition from Stereovision Equipment

Innehållsförteckning

1 INLEDNING.....	5
1.1 PARTER.....	5
1.2 MÅL.....	6
1.3 ANVÄNDNING.....	6
1.4 DEFINITIONER	6
2 ÖVERSIKT AV SYSTEMET.....	7
2.1 GROV BESKRIVNING AV PRODUKTEN.....	7
2.2 PRODUKTKOMPONENTER.....	8
2.3 BEROENDEN TILL ANDRA SYSTEM.....	8
2.4 INGÅENDE DELSYSTEM.....	8
2.5 DESIGNFILOSOFI.....	8
3 KAMERARIGG.....	9
3.1 KAMEROR.....	9
3.2 TRÖGHETSSYSTEM IMU.....	10
3.3 LASERSCANNER.....	10
3.4 KAMERAPLACERING.....	10
3.4.1 Stereometoder.....	10
3.4.2 Trinokulära metoder.....	11
4 DATABASEHANDLINGSENHET.....	12
4.1 VERKTYGSLÅDA.....	12
4.1.1 Kamerakalibrering.....	13
4.1.2 Skalpyramid.....	13
4.1.3 Stereoalgoritmer.....	14
4.1.3.1 Feature based.....	14
4.1.3.2 Fäsbaserad stereoestimering.....	15
4.1.4 Egomotion.....	15
4.1.5 Punktmolnsoperationer.....	16
4.1.5.1 Kanonisk Korrelationsanalys.....	16
4.1.5.2 Local Template.....	17
4.1.6 Funktioner.....	17
4.1.6.1 harris.m.....	17
4.1.6.2 canny.m.....	17
4.1.6.3 fastradial.m.....	17
4.1.6.4 testfund.m.....	18
4.1.6.5 calib_gui.m.....	18
4.1.6.6 stereo_gui.m.....	18
4.1.6.7 create_pyramid.m.....	18
4.1.6.8 egoMotion.m.....	19
4.2 APPLIKATION.....	20
4.2.1 Funktioner.....	20
4.2.1.1 GetImages.....	20
4.2.1.2 GetPosition.....	21
4.2.1.3 CreateCloud.....	21
4.2.1.4 CreateModel.....	21
4.2.2 Grafiskt gränssnitt GUI.....	21
4.3 SCENESERVER.....	22
4.3.1 SSInit().....	23
4.3.2 SSLoad_model().....	23
4.3.3 SSMakeTrajectory().....	24
4.3.4 BatchKatalog.....	24
4.3.5 SSGetData_online().....	24
4.3.6 SSGetData_batch().....	24

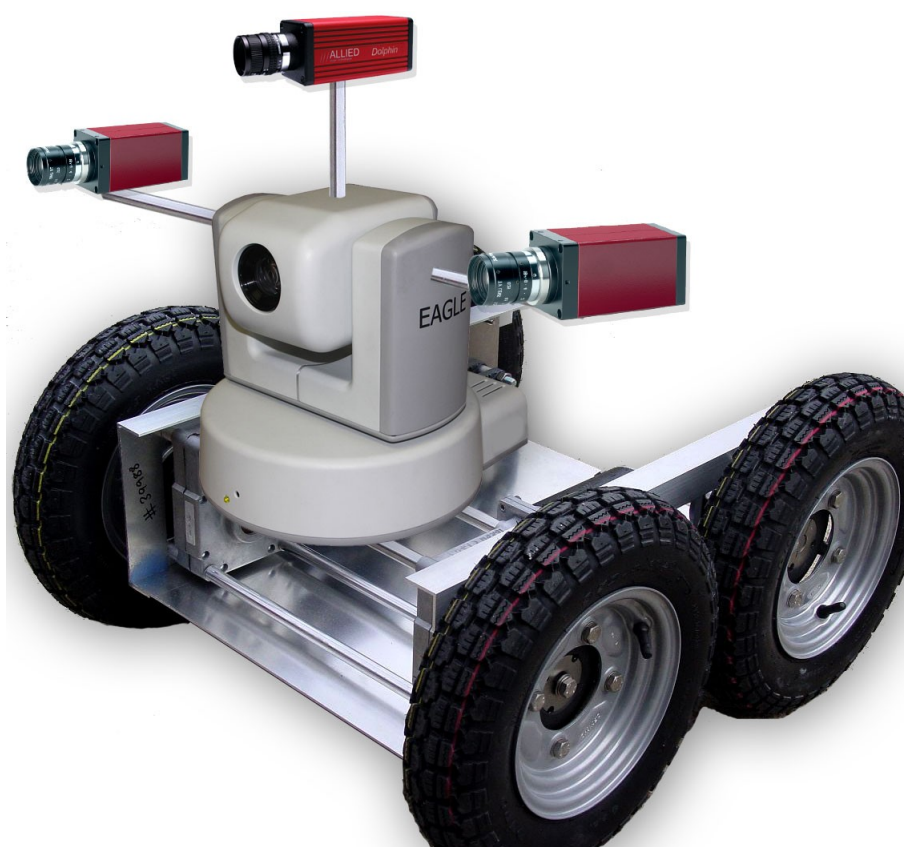
Accurate Multipoint Acquisition from Stereovision Equipment
Dokumenthistorik

version	datum	utförda förändringar	utförda av	granskad
0.1	2006-02-18	Första utkastet	Alla	
0.2	2006-03-02	Andra utkastet	Alla	MD
0.21	2006-03-07	Korrigerat andra utkast	Alla	MD
0.22	2006-03-09	Kompletterat andra utkast	MD	MD

1 Inledning

FOI ska konstruera en obemannad markgående robot (UGV). Vi har fått i uppgift att utveckla ett system som ska ge information som kan hjälpa roboten med t.ex. kollisionshantering. Projektets syfte är att utveckla ett system för estimering av punktmoln som uppdateras med tiden och tar hänsyn till tidigare data. Molnet består av en mängd punkter som beskriver djup i en omgivning. System består av två till tre kameror monterade på en rigg och en dator, se figur nedan.

Detta dokument syftar till att beskriva hur systemet ska realiseras och genomföras. Dokumentet är skrivet utifrån systemskissen och tar upp de idéer och förslag som framkommit under skrivandet av kravspecifikation och systemskiss. Designspecifikationen ligger till grund för implementationen av projektet.



Figur 1: Fotomontage

1.1 Parter

Kund: Morgan Ulvklo, FOI.

Producent: Projektgruppen.

Examinator: Klas Nordberg, ISY.

Handledare: Johan Wiklund, ISY och Anders Moe, ISY.

Accurate Multipoint Acquisition from Stereovision Equipment

1.2 Mål

Att utveckla och implementera ett system för estimering av en omvärldsmodell utifrån punktmoln i en scen med hjälp av två till tre kameror, rigg, tröghetsnavigeringssystem (IMU) och en dator

1.3 Användning

Systemet ska användas av FOI som ett delsystem i en UGV. Systemet ska stödja beslutsfunktioner för t.ex. navigation och kollisionshantering.

1.4 Definitioner

UGV – Unmanned Ground Vehicle

INS – Inertial Navigation System

IMU – Inertial Measurement Unit

GUI – Graphical User Interface

SceneServer – Simuleringsmiljö tillhandahållen av FOI

Fps – frames per second

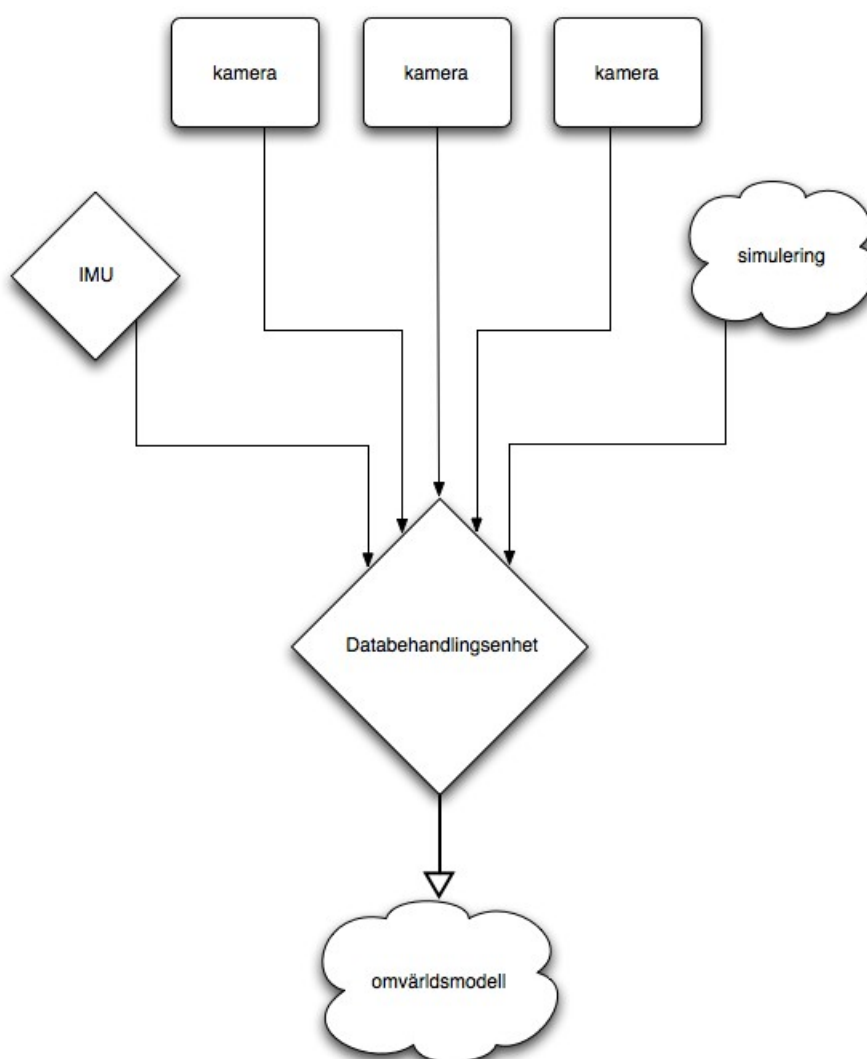
Framerate – datatakt/bildhastighet

Egomotion - egenrörelse

2 Översikt av systemet

2.1 Grov beskrivning av produkten

Systemet ska utgöras av en rigg där två eller tre kameror är monterade, samt en dator för beräkningar och bildanalys. Systemet ska som resultat ge en modell av omgivningen i form av ett punktmoln som uppdateras och utnyttjar tidigare data. Kamerorna kommer att leverera högupplösta stereopar av omgivningen vilka kommer att användas för avståndsberäkning. En tredje kamera kan användas för att öka säkerheten hos avståndsdatat. En IMU kommer att ge data om systemets rörelse vilket kan användas för att ge en snabbare stereoalgoritm och för att skapa ett scenminne. Systemet ska även klara av att göra beräkningarna på simulerade data från programvaran SceneServer.



Figur 2: Skiss över systemet.

Accurate Multipoint Acquisition from Stereovision Equipment

2.2 Produktkomponenter

Implementeringen kommer att ske på en industridator till vilken en kamerarigg kommer att vara kopplad. Programvaran kommer att implementeras i Matlab. På kamerariggen kommer det att finnas tre högupplösta digitalkameror av typen AVT Marlin F-131 B/C. Utöver det kommer även en IMU av typen CCT Crista som levererar positionsdata åt systemet att vara monterad på riggen.

2.3 Beroenden till andra system

Systemet kommer inte vara beroende av andra system, men kravet att det ska gå att köra med simulerade data från SceneServer medför att en fungerande koppling till SceneServer krävs.

2.4 Ingående delsystem

Systemet består av tre delsystem:

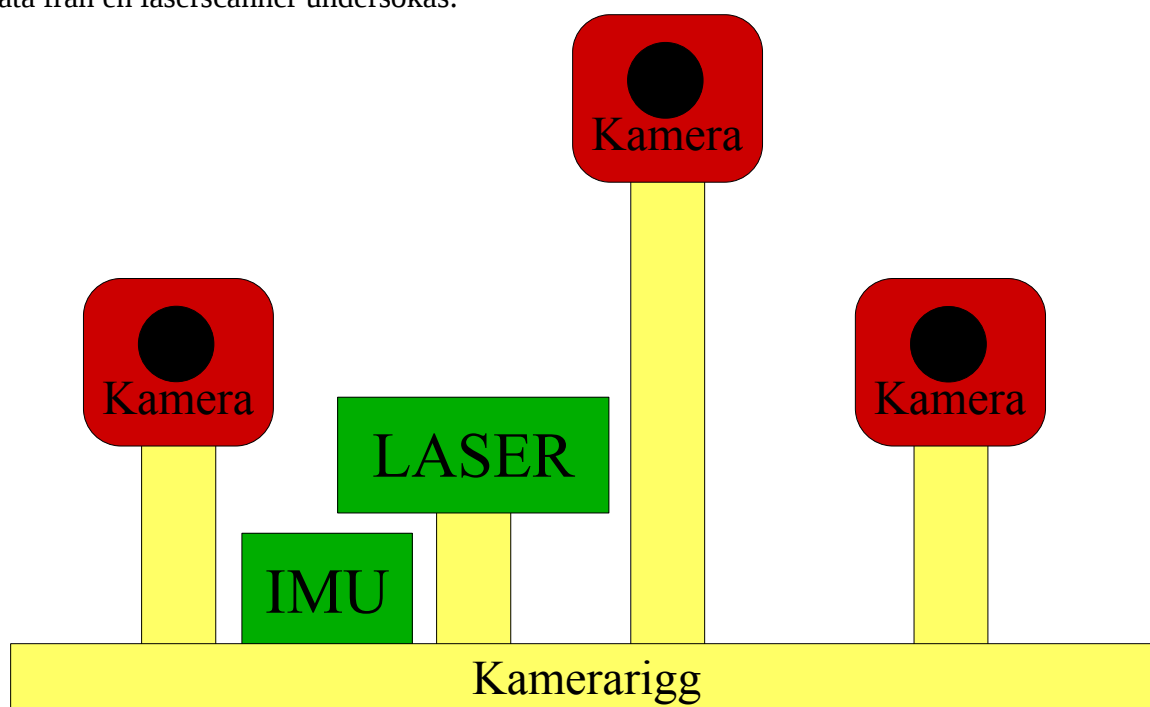
- Kamerarigg
 1. Kameror
 2. Tröghetssystem IMU
 3. Laserscanner
- Databehandlingsenhet
 1. Applikation
 2. Verktygslåda
- SceneServer

2.5 Designfilosofi

Tyngdpunkten vid designen ska läggas på funktionalitet framför prestanda. Prestandan ska dock inte glömmas vid val och implementation av algoritmerna. Gruppen ska fokusera på att få en fungerande applikation i ett tidigt skede, för att sedan utveckla och förbättra applikationen till en nivå som uppfyller ställda krav. Systemet ska också konstrueras på sådant sätt att information från ytterligare sensorer och mätdata ska kunna användas.

3 Kamerarigg

Kamerariggen, vilken utvecklas för att möjliggöra autonom styrning av en UGV, består av tre fastmonterade kameror vilka ska ge information i form av bilder till databehandlingsenheten. Då kamerariggen är tänkt att fästas på en pan-tilt-plattform på UGV-farkosten ska även en IMU ingå, vilken ska förse systemet med tröghetsdata. Vidare ska möjligheten att använda data från en laserscanner undersökas.



Figur 3: Kamerarigg.

3.1 Kameror

För grundläggande funktionalitet ska två kameror användas för att tillgodose systemet med stereodata. Beroende på val av stereoalgoritm ska det undersökas hur en tredje kamera bör konfigureras för att förbättra punktmolnscanningen. Kamerorna som kommer att användas är av märket AVT Marlin F-131 B/C och ansluts till datorn via firewire med 400Mbit/s. Dessa kameror kan leverera bilder i upplösningar upp till 1280x1024. Kamerorna kan ge ifrån sig färgbilder med 12 fps eller gråskalebilder i en hastighet av 25 fps. Dessa kameror stödjer Area of Interest, det vill säga att endast en liten del av bilden kan skickas med högre framerate. Kamerorna drivs av en likspänning på 12 Volt via firewire, och förbrukar en effekt på 3 Watt.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

*Figur 4: AVT Marlin F-131 B/C*

3.2 Tröghetssystem IMU

*Figur 5: CCT Crista IMU*

En IMU från Cloud Cap Technology ska användas för att bidra systemet med 3D-tröghetsnavigationsdata som i första hand ska användas för att skatta systemets egenrörelse. IMU-enheten kommer att kopplas in till datorn via RS-232. Enheten drivs av 4,5-15V och förbrukar en effekt på 260mW. Uppdateringshastigheten för enheten kan styras från medföljande mjukvara.

3.3 Laserscanner

En laserscanner ska kunna kopplas på riggen för att bidra med djupdata längs en horisontallinje. Denna information ska eventuellt användas för att öka säkerheten på utdata.

*Figur 6: Laserscanner från Siemens*

3.4 Kameraplacering

3.4.1 Stereometoder

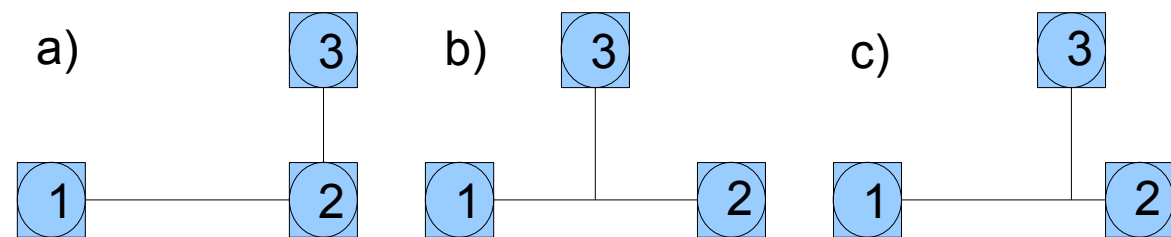
Linjen mellan två kameror i ett stereosystem kallas för baslinje. Fördelarna med en kortare baslinje är att matchningarna av bildpunkter blir lättare på grund av liknande bilder och risken för att vissa punkter i ena bilden inte kan hittas i den andra på grund av att de är skymda minskar. Med en längre baslinje fås däremot ökad säkerhet i djupestimaten eftersom dispariteten blir större.

Accurate Multipoint Acquisition from Stereovision Equipment

3.4.2 Trinokulära metoder

Givet ett system med två kameror kan en tredje läggas till. Allmänt gäller att ju fler utspridda kameror, desto mer information om scenen blir tillgänglig och säkrare djupestimat kan beräknas. Med fler än två kameror där dessa inte ligger längs en gemensam baslinje undviks problemet med kurvor och linjer i samma plan som de två kamerornas baslinje som då är svåra eller omöjliga att estimeras djup från. Trinokulära metoder är alltså att föredra framför stereo om den extra tiden för beräkningar finns. Följande tre sätt att placera ut den tredje kameran på skall utredas under projektets gång:

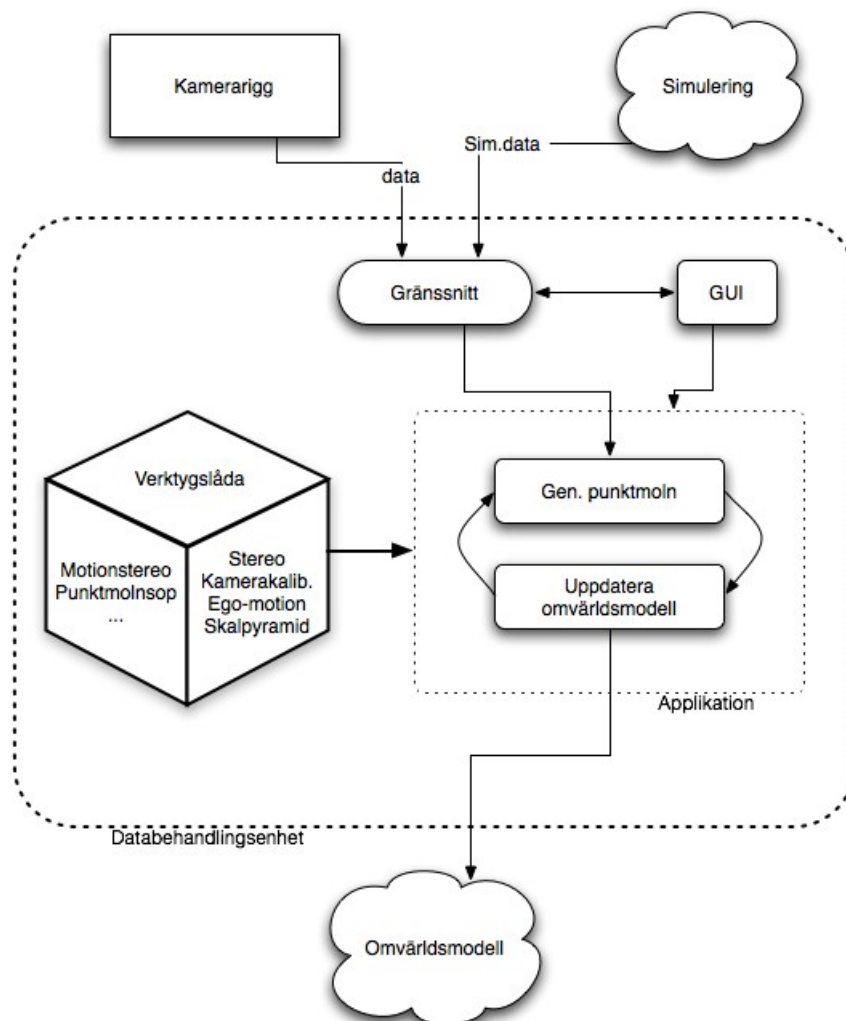
- L-configuration: Den tredje kameran bildar en rät triangel tillsammans med de övriga två, se figur 7. På detta vis blir det lätt att dela upp systemet i två delar som är vinkelräta mot varandra och matcha delresultaten av djupestimaten till ett slutgiltigt.
- Symetric V-configuration: Tack vare ett system enligt figur 7 där tredje kameran placeras mitt emellan de övriga två horisontellt sett fås ytterligare information i denna ledd samtidigt som den extra dimensionen från höjdskillnaden ökar säkerheten.
- Non-symetric V-configuration: Med ett system som det föregående med den skillnaden att det horisontella avståndet mellan kamera 2 och 3 inte är jämnt delbart med avståndet mellan kamera 1 och 3, se figur 7, så fås även fördelen att det blir lättare att estimeras djup vid upprepade mönster i bilden. [1]



Figur 7: Tre olika sätt att placera en tredje kamera till ett stereosystem.

4 Databehandlingsenhet

Databehandlingen kommer ske i en PC utgående från data som den samlar in via sensorerna på kamerariggen. Utdata som eftersöks är en beskrivning av den fysiska omgivningen som systemet befinner sig i, representerat i form av ett punktmoln.



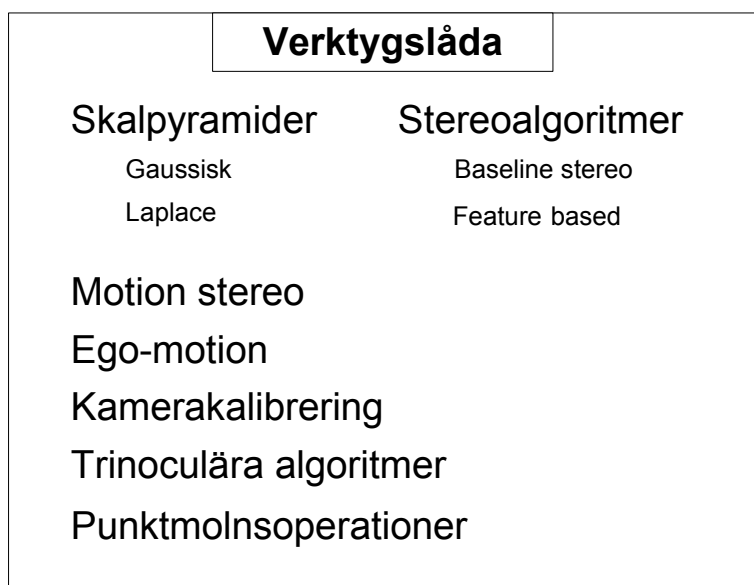
Figur 8: Skiss över databehandlingsenhet.

4.1 Verktygslåda

För att lösa systemets huvuduppgift ska en verktygslåda med diverse funktioner tillverkas. Ett antal av funktionerna ska dessutom kombineras till en applikation som uppfyller målet, omvärldsmodellen. Val av de funktioner som slutligen kommer att användas i applikationen sker genom de utvärdering av prestanda i de test som beskrivs i testplanen.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment



Figur 9: Exempel på verktyslådans innehåll.

Figuren ovan är ett exempel på hur det kan se ut när verktyslådan är färdig. Under projektets gång kommer verktyslådan att utvecklas efter behov.

4.1.1 Kamerakalibrering

Kalibrering av kamerorna är nödvändig för att metrisk data ska kunna erhållas från kamerornas bilder. Kalibreringen innebär att man beräknar de intrinsiska och extrinsiska parametrar för kamerorna. De intrinsiska parametrarna beskriver hur en punkt i 3D-rymden projiceras till bildplanet och utgörs av fokallängd, principialpunkt, skevning och distortion och beskrivs i en kalibreringsmatris. De extrinsiska parametrarna beskriver hur kamerans koordinatsystem förhåller sig till världskoordinatsystemet och beskrivs genom rotation och translation. Våra kameror är kalibrerade med "Camera Calibration Toolbox for Matlab" utvecklat på Caltech[2]. Kalibreringen sker genom att man för kameran visar upp ett plant rutmönster med känd storlek i ett antal vinklar (minst två), rörelsen behöver inte vara känd. Då alla punkter ligger i ett plan ska de också avbildas till ett plan. Detta tillsammans med att man känner storleken på rutorna (avståndet mellan hörnen) kan de intrinsiska och extrinsiska parametrarna beräknas. Toolboxen kan också användas för att kalibrera stereopar av kameror vilket som resultat ger hur de två kameror förhåller sig till varandra. Från kamerornas förhållanden kan man sedan beräkna essentialmatrisen som tillsammans med kalibreringsmatriserna för de båda kamerorna kan användas för att beräkna fundamentalmatrisen. Genom att kalibrera en kamera mot var och en av de två kamerorna och slå samman resultatet kan man dessutom få ut förhållandena mellan alla tre kameror. Då man har sambanden mellan de tre kamerorna kan trifocal tensor beräknas.

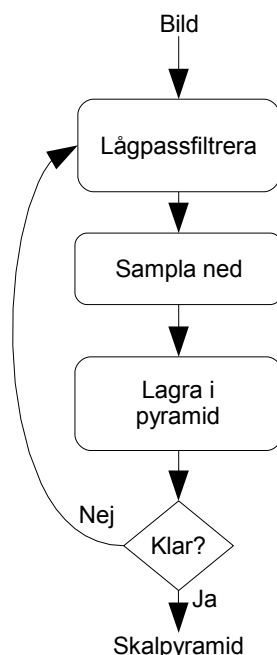
4.1.2 Skalpyramid

För att kunna bearbeta bilder från kamerorna i olika skalor implementeras en funktion som bildar en gaussisk skalpyramid. Bilderna samplas ned och för att undvika störande vinkningsdistorsion lågpasfilteras dessa först. Varje bild nummer n i pyramiden är inbilden nedskalad en faktor 2^n .

Algoritmen arbetar enligt nedanstående flödesschema.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment



Figur 10: Flödesschema för skalpyramidalgoritmen.

Filtreringen görs med ett gaussiskt filter eftersom detta är det enda filter som har egenskapen att en bild högre upp i pyramiden är densamma som resultatet av en direkt lågpasfiltrering och nedsampling av originalbilden till denna storlek. För att öka prestandan implementeras lågpasfiltrering och nedsampling på låg nivå med hjälp av ett bibliotek från Intel som innehåller funktioner optimerade för Intels processorer [3].

Skalpyramiden ska till en början användas för att minska ner storleken på bilderna och snabba upp beräkningar utan att förlora information om hela områden i scenen. Om istället upplösningen på kamerorna ändras, så beskärs bilden. Därför är det önskvärt att låta kamerorna jobba i högsta upplösning för att sedan bilda skalpyramider. En annan idé för användning av skalpyramiden är att om UGV:n rör sig med normal fart så utförs beräkningen av punktmoln på lågupplösta bilder för att hålla en acceptabel uppdateringsfrekvens. Om istället UGV:n står still eller rör sig mycket långsamt så utförs alla beräkningar på högupplösta bilder eftersom lägre uppdateringsfrekvens då är ok.

Funktionen i toolboxen för att bilda skalpyramider kallas `create_pyramid.m`.

4.1.3 Stereoalgoritmer

4.1.3.1 Feature based

Feature based stereo baseras på att man i de två eller tre bilderna söker kännetecknade områden eller punkter (features). Därefter matchas features i de olika bilderna så att man får en lista med matchande koordinater som för punkter i en de olika bilderna. Känner man dessutom till fundamentalmatrisen (två bilder), eller trifocalmatrisen (tre bilder) kan man beräkna epipolarlinjerna i de andra bilderna för en bilds intressanta punkter.

Fundamentalmatrisen samt trifocalmatrisen kan beräknas från kamerakalibreringen.

Sökningen av korresponderande punkter eller områden kan då förenklas och snabbas upp genom att man bara söker efter matchningar längs epipolarlinjerna. När korresponderande punkter eller områden funnits i två eller tre bilder kan punktens 3D-koordinater beräknas

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

m.h.a. fundamentalmatrisen respektive trifocalmatrisen genom triangulering. Trianguleringen fungerar genom att två korresponderande punkter återprojiceras ut i 3D-rymden, om epipolarvilkoret är uppfyllt för punkterna ligger de projicerade linjerna i ett plan och kommer att skära varandra i den sökta punkten. [4]

De funktioner för featuredetectering som finns tillgängliga i toolboxen är

- Canny edge detection
- Harris corner detector
- Loy and Zelinski's fast radial feature detector

Funktionerna finns närmare beskrivna i kapitel 4.1.6.

En funktion för estimering av fundamentalmatrisen finns tillgänglig i toolboxen i funktionen *testfund*. Estimeringen fungerar på så sätt man först beräknar ett stort antal möjliga korrespondanser mellan punkter i två bilder, därefter väljs slumpvis 8 par korrespondanser ut och en fundamentalmatris beräknas genom epipolarvilkoret. Om den funna lösningen stämmer med ett stort antal av de beräknade korrespondanserna bedöms lösningen vara korrekt, i annat fall väljs slumpvis 8 nya punkter och en ny fundamentalmatris beräknas (RANSAC). I de fall då man har kalibrerade kameror behöver man inte göra en estimering istället kan matriserna beräknas från förhållandena mellan kamerorna samt kalibreringsmatriserna. Trianguleringen sker med funktionen *stereo_triangulation* som finns i kamerakalibrerings toolboxen.

4.1.3.2 Fasbaserad stereoestimering

Fasbaserad djupestimering ur ett stereopar bygger på att man analyserar lokal fas i stereoparet, för att sedan beräkna fasskillnaden mellan bilderna. Skillnaden i lokal fas ger ett mått på dispariteten mellan bilderna, och således kan ett djupestimat beräknas. Denna typ av djup-från-stereo-algoritm ger en tät (dense) djupkarta, vilket kan vara fördelaktigt i vissa fall. Observeras bör att denna typ av stereoalgoritm dock kräver att bildplanen är parallella, vilket är precis motsatt krav från den fundamentalmatrisbaserade metoden ovan. Det är någorlunda rättframt att verifiera hur bra djupapproximation som fås ur ett stereopar med denna metod, då visualisering enkelt kan åstadkommas genom en höjdkarta.

4.1.4 Egomotion

Egomotion handlar om att skatta kamerans egenrörelse. En metod för att göra detta utnyttjar optiskt flöde och djup i en statisk scen. Den matematiska beskrivningen kan uttryckas:

$$u(\mathbf{x}) = \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -x_2 \end{bmatrix} \left(\frac{\mathbf{T}}{Z(\mathbf{x})} + \boldsymbol{\Omega} \times \mathbf{x}_h \right)$$

där

$$u(\mathbf{x}) = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix}, \quad \boldsymbol{\Omega} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}, \quad \mathbf{x}_h = \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

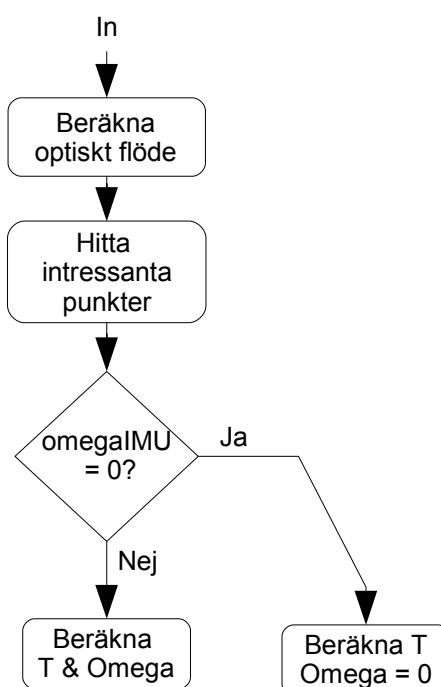
u , Z anger det optiska flödet respektive djupet i punkten $\mathbf{x}$, $\mathbf{T}$ är kamerans translationshastighet och $\boldsymbol{\Omega}$ anger kamerans vinkelhastighet kring koordinataxlarna. Se [5] för en utförligare framställning.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

Djupet har beräknats tidigare medan optiska flödet estimeras enligt en gradientbaserad metod. För att bestämma T och Ω behövs det optiska flödet och djupet för minst 3 punkter eftersom varje punkt ger två ekvationer. Optiska flödet och djupet är bara estimerat och har en viss osäkerhet. Därför väljs ett antal punkter med hög säkerhet på det optiska flödet och djupet för skattning av T och Ω . För att få ett robust estimerat används så många punkter, med hög säkerhet, som möjligt. En annan möjlighet att förbättra skattningen är att använda rotationsdata från IMU:n, om denna visar sig vara tillräckligt bra.

I toolboxen kallas funktionen egoMotion.m, i applikationen används egoMotion.m av getPosition.m.



Figur 11: Flödesschema för egomotionalgoritmen.

4.1.5 Punktmolnsoperationer

Med punktmolnsoperationer menas de operationer som behöver utföras för att matcha den aggregerade modellen med det senast genererade punktmolnet. Det som eftersökes är en rotation och en translation som beskriver förflyttningen mellan två tidpunkter. Här kan IMU-data användas för att med god noggrannhet säkerställa rotationen, vilket reducerar problemet till att hitta den translation och skalning som bäst matchar det nygenererade punktmolnet på den aggregerade modellen. Detta skalnings och translationsproblem är tänkt att angripas genom någon korrelationsbaserad jämförelse, eventuellt kanonisk korrelationsanalys. En annan metod skulle kunna vara att utföra en local template matchning.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

4.1.5.1 Kanonisk Korrelationsanalys

Vid kanonisk korrelationsanalys återfinnes den egenvektor som bäst projicerar datat (punktmolnet) på modellen (aggregerade modellen). Denna egenvektor beskriver då hur punktmolnet skall translateras och skalas för att återfå bästa korrelationen.

4.1.5.2 Local Template

Då IMU-enheten ger en god estimering av rotationen kan en lokal omgivning till en uppsättning featurepunkter i det nya punktmolnet roteras i enlighet med IMU-datat. Därefter kan en vanlig korrelationsanalys utföras mellan dessa lokala omgivningar. Om featurepunkterna väljs på ett sådant sätt att dess lokala omgivningar utgör ett plan med normalen riktad mot kameran, kan en translation i detta plan detekteras.

4.1.6 Funktioner

4.1.6.1 harris.m

Beräknar harrispunkter ur en bild. En gradient tensor beräknas för bilden, de områden med stark förändring i fler än en riktning sparas varefter lokala maximum beräknas och markeras som harrispunkter. Resultatet blir en lista över koordinater för intressanta punkter.[6][7]
Beräknad arbetstid 10 timmar (färdig implementation finns).

Inparametrar

im – bilden som ska bearbetas

sigma – standardavvikelsen för det gaussiska filtret

tresh – tröskel (valfri)

radius – minsta radie för lokalt maximum, typiskt 1-3 (valfri)

disp – flagga för val om punkter ska plottas i originalbilden, 0/1 (valfri)

Utparametrar

cim – binär bild som markerar harrispunkterna (om tröskel och radie ej är angiven blir cim en bild av harrissvaret).

r – radkoordinater för harrispunkterna

c – kolumnkoordinater för harrispunkterna

rsub – subpixel radkoordinater för harrispunkterna

csub – subpixel radkoordinater för harrispunkterna

4.1.6.2 canny.m

Beräknar områden med intensitetsdiskontinuiteter. Funktionen filtrerar först bilden med ett gaussiskt filter, därefter beräknas första ordningens derivata över bilden vilket ger starka utslag vid diskontinuiteter. Slutligen genomförs en tröskling med hysteres så att bara toppen på åsar sparas. I denna implementation genomförs dock ej den sista trösklingen.[7][8] Beräknad arbetstid 10 timmar (färdig implementation finns).

Inparametrar

im – bilden som ska bearbetas

sigma – standardavvikelsen för det gaussiska filtret

Utparametrar

gradient – gradientamplitudbild

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

or – orienteringsbild

4.1.6.3 fastradial.m

Letar efter radiella symmetrier. Funktionen beräknar först två gradientbilder genom faltning med sobelfilter, därefter skapas en orienteringsbild och en magnitudbild. De båda bilderna kombineras och faltas med en gaussfunktion vilket ger medel över symmetrin över en viss radie för varje bildpunkt.[7][9] Beräknad arbetstid 10 timmar (färdig implementation finns).

Inparametrar

im – bilden som ska bearbetas

radii – vektor med heltals radier som ska beräknas

alpha – parameter för symmetrikrav

Utparametrar

S – symmetrikarta. Ljusa punkter med hög symmetri har stora positiva värden, mörka punkter med hög symmetri har stora negativa värden.

4.1.6.4 testfund.m

Beräknar en estimering av fundamentalmatrisen, använder harrispunkter och RANSAC för att beräkna fundamentalmatrisen. Funktionen har modifierats till att returnera fundamentalmatrisen[4][7] Beräknad arbetstid 10 timmar (färdig implementation finns).

Inparametrar

im1 – Den ena stereobilden

im2 – Den andra stereobilden

Utparametrar

F – Den estimerade fundamentalmatrisen.

4.1.6.5 calib_gui.m

Ett grafiskt gränssnitt för kamerakalibrering. För kalibrering krävs ett antal bilder (minst två) av ett kalibreringsmönstret.[2] Beräknad arbetstid 10 timmar (färdig implementation finns).

Inparametrar

Inga, bilderna läses in från det grafiska gränssnittet.

Utparametrar

Inga, kalibreringsresultatet sparas till en fil från det grafiska gränssnittet.

4.1.6.6 stereo_gui.m

Ett grafiskt gränssnitt för stereokalibrering. Funktionen kräver ett antal stereopar bilder över ett kalibreringsmönster, samt kalibreringsdata för de båda kamerorna.[2] Beräknad arbetstid 10 timmar (färdig implementation finns).

Inparametrar

Inga, bilder och kalibreringsdata läses in från det grafiska gränssnittet.

Utparametrar

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

Inga, kalibreringsresultatet sparas till en fil från det grafiska gränssnittet.

4.1.6.7 create_pyramid.m

Beräknar en gaussisk skalpyramid där varje bild nummer n i pyramiden är inbilden lågpasfilterad och nedsamplad en faktor 2^n . Beräknad arbetstid 10 timmar.

Inparametrar

image – bild som ska bearbetas

n – heltal som anger antalet nivåer i skalpyramiden

Utparametrar

pyramid – en cell array där första cellen innehåller inbilden **image** och efterföljande celler innehåller alltmer nedskalade varianter.

4.1.6.8 egoMotion.m

Skattar kamerans egenrörelse utifrån djupdata och optiskt flöde från två bilder. Beräknad arbetstid 20 timmar.

Inparametrar

points – lista innehållande ett antal punkter med tillhörande världskoordinater.

currentImage – nuvarande bild.

currentTime – tid då bilden togs.

previousImage – tidigare bild.

previousTime – tid då tidigare bilden togs.

omegaImu – Rotationshastighetsmatris uträknat från IMU-data.

Utparametrar

T – translationshastighetsmatris.

Omega – rotationshastighetsmatris.

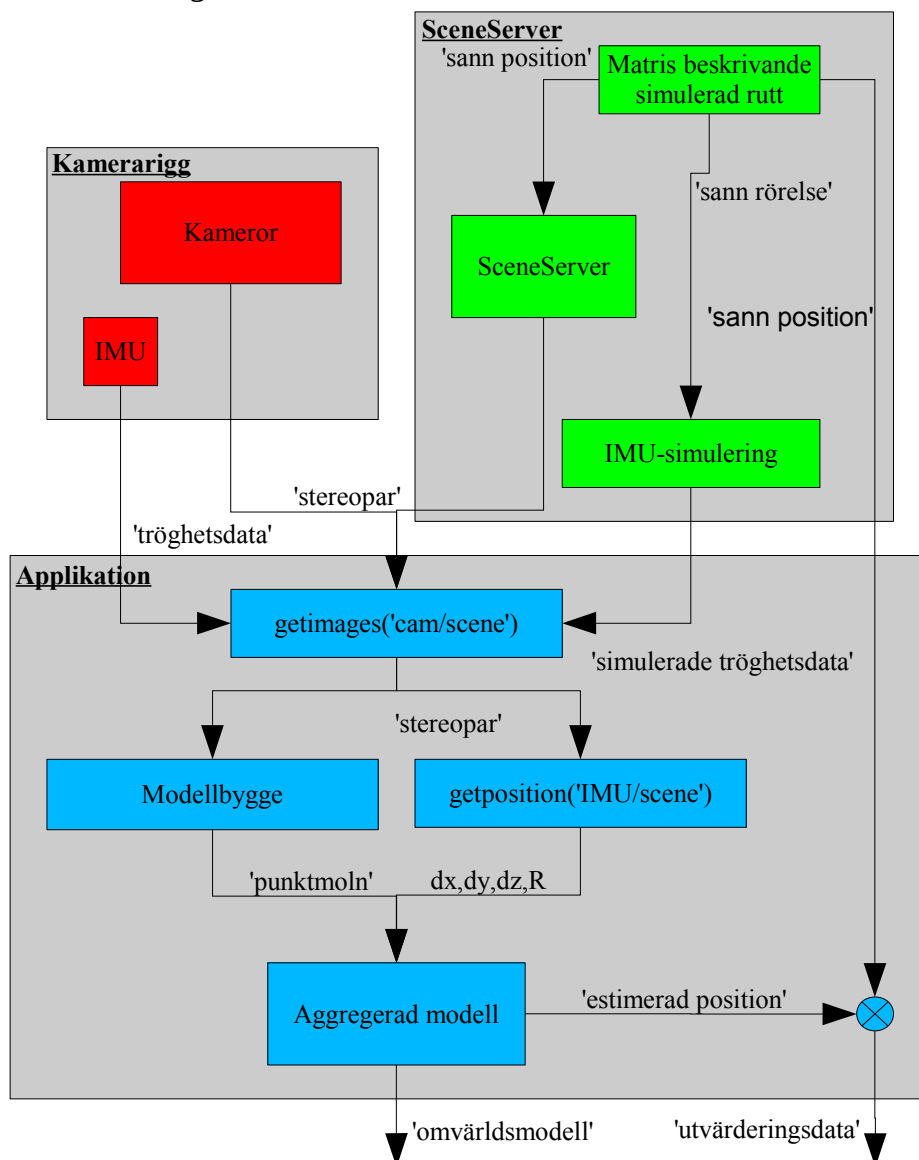
deltaT – tidsskillnad mellan bilderna.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

4.2 Applikation

Utdata från systemet är aktuell representation av omgivande miljö i form av ett punktmoln som uppdateras med tiden. Applikationen använder sig av verktygslådans funktioner för att skatta en modell av omvärlden. I första steget beräknar applikationen ett punktmoln av omgivningen, därefter uppdateras den aktuella omvärldsmodellen med information från det nya punktmolnet och egomotionfunktionerna.



4.2.1 Funktioner

Beskrivning av de funktioner som utgör applikationen.

4.2.1.1 GetImages

Eftersom huvuddelen av utvecklingsarbetet sker med simulerat data bör databehandlingen kunna ske oberoende av om data från kamerariggen eller simuleringen används. Detta är även önskvärt efter utvecklingsfasen, då test av systemet på virtuella miljöer kan vara av intresse. Även styrning av SceneServer och sensorer kopplas via gränssnittet. Gränssnittet ska kunna

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

styras med hjälp av GUI:t för exempelvis styrning av kameror och växla indata. Beräknad arbetstid 20 timmar.

4.2.1.2 GetPosition

GetPosition skattar relativ förflyttning med hjälp av data från egomotionalgoritmen. Beräknad arbetstid 20 timmar.

Inparametrar

T – translationshastighetsmatris.

Omega – rotationshastighetmatris.

deltaT – tidsskillnad mellan bilderna.

Utparametrar

D – en vektor innehållande relativ translation i världen.

4.2.1.3 CreateCloud

CreateCloud läser in ett bildpar från getImages. Därefter beräknas ett punktmoln genom att featurepunkter korreleras och trianguleras. Korrelationen sker genom binär fasbaserad korrelation. Därefter beräknas fundamentalmatrisen och de korrelationer som ej uppfyller epipolarvilkoret väljs bort. Funktionen kräver kalibreringsdata för kamerakonfigurationen. Beräknad arbetstid 20 timmar.

Inparametrar

im1,im2 – de två bilderna

Utparametrar

XL, XR – lista med världskoordinater för featurepunkter utgående från vänster respektive höger kamera.

4.2.1.4 CreateModel

CreateModel roterar det aktuella punktmolnet och använder därefter kanonisk korrelation för att aggregera detta punktmoln till den aktuella modellen (de tidigare aggregerade punktmolnen). Funktionen kräver ett punktmoln i form av listor med världskoordinater från vänster respektive höger kamera från CreateCloud, rotationsdata från IMU-enheten samt den aggregerade modellen. Utdata från funktionen är en ny aggregerad modell. Beräknad arbetstid 20 timmar.

Inparametrar

R – Rotationsmatris från

XL, XR – lista med världskoordinater för featurepunkter utgående från vänster respektive höger kamera.

XL_model, XR_model = nya listor över världskoordinater för featurepunkter.

Utparametrar

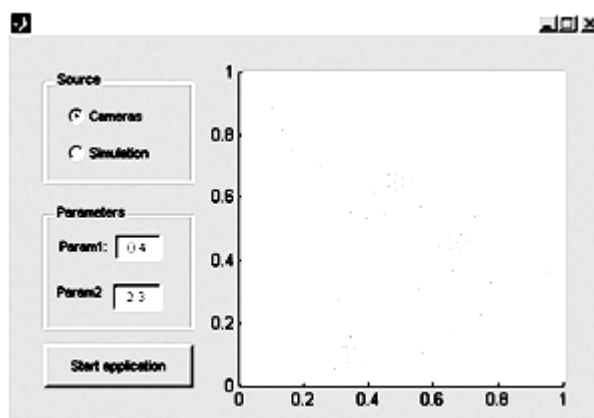
XL_model, XR_model = nya listor över världskoordinater för featurepunkter.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

4.2.2 Grafiskt gränssnitt GUI

Detta delsystem är till för att ge grafisk feedback till användaren för att kunna utvärdera systemet och verifiera dess funktionalitet genom plottar av punktmoln eller estimerad rörelse. Gränssnittet kommer även att möjliggöra justering av vissa parametrar samt styrning av om bilderna som ska behandlas kommer från kameror på riggen eller en simuleringsmiljö (SceneServer). Se figur 8, tidigare i dokumentet, för en visualisering av detta. GUI:t kommer att implementeras i Matlab och visas som ett fönster likt det i figur 12 där användaren lätt kan starta applikationen och justera de valda parametrarna.



Figur 12: Exempel på hur grafiska gränssnittet skulle kunna se ut.

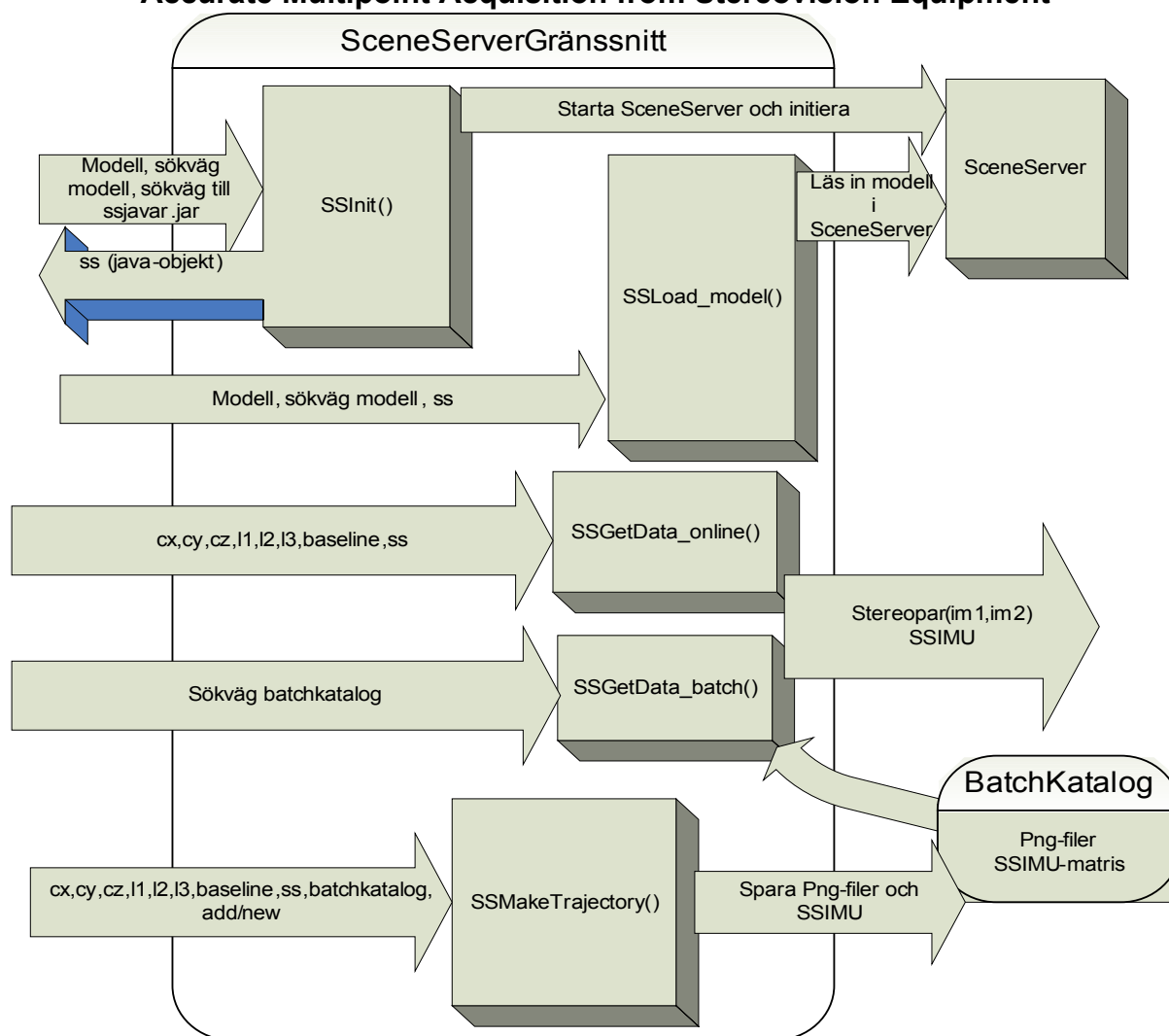
Funktionen för gränssnittet kallas `start_gui.m`. Beräknad arbetstid för implementation av funktionen är 30 timmar.

4.3 SceneServer

SceneServer är ett simuleringsverktyg som kan anropas via matlab utvecklat i java av FOI. Detta verktyg kan användas för att erhålla simulerade stereobilder genom att flytta runt en simulerad nålhåldskamera i en 3D-miljö. Genom att spara förflyttningarna av kameran kan även simulerad IMU-data erhållas. Detta simuleringsverktyg kommer därför vara användbart för att testa och utvärdera de stereoalgoritmer och egomotionalgoritmer som skall implementeras i toolboxen. För att detta skall vara möjligt kommer ett matlabgränssnitt utarbetas för att sköta kommunikation mellan SceneServer och applikationen. SceneServer är således att betrakta som en del av systemet. Beräknad arbetstid för implementation av gränssnittet till SceneServer är 25 timmar.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment



4.3.1 SSInit()

För att överhuvudtaget kunna kommunicera med SceneServer måste SceneServerprogrammet startas och initieras. Detta sker genom att matlab får läsa in javaklassen ssjava.jar som medföljer SceneServerprogramvaran. För att detta ska ske så smidigt som möjligt, kommer klassinläsningen av ssjava.jar ske dynamiskt i matlab (detta får till följd att SceneServerprogramvaran ej behöver vara startad innan matlab startas). Därefter startas SceneServer via matlab och initieringen sker. Då detta är en matlabfunktion kommer SSInit() vara tvungen att returnera javaobjektet (ss i flödesschemat) för att övriga delar av SceneServergränssnittet senare skall kunna anropa SceneServer.

4.3.2 SSSLoad_model()

Denna funktion tar in ett argument (modell) och en sökväg till den katalog där modellen finns. Därefter läses den angivna modellen in bild för bild från modellkatalogen och skickas till SceneServer så att modellen byggs upp.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

4.3.3 SSMakeTrajectory()

Denna funktion tar in kamerans x,y,z position, den position som kameran är riktad mot, baslinjens längd till andra kameran, javaobjektet för kommunikation med SceneServer, sökvägen till batchkatalogen samt ett argument som talar om funktionen om den ska skapa ett nytt dataset eller lägga till det nya datat sist i det aktuella datasetet. Funktionen läser av från SceneServer de nuvarande positionen för kameran samt åt mot vilken punkt kameran är riktad. Utifrån detta och den nya givna positionen kan en korrekt translation och rotation beräknas. Genom att sedan flytta kameran till den nya positionen och spara undan den aktuella bilden till en png fil, för att sedan flytta kameran enligt baselineargumentet till positionen för den andra kameran och spara undan denna bild har ett stereopar erhållits. Bilderna sparas ner i batchkatalogen i png-format med ett unikt nummer som filnamn. Även den simulerade IMU-datan sparas ner i en IMU-data matris (SSIMU-matris) i batchkatalogen.

4.3.4 BatchKatalog

Detta är en godtycklig katalog på hårddisken innehållande ett dataset med png-filer av stereobilder samt en fil innehållande SSIMU-matrisen.

4.3.5 SSGetData_online()

Inargument till denna funktion utgöres av den position kameran önskas flyttas till, den nya riktningen på kamerans samt javaobjektet för kommunikation med SceneServer.

Funktion läser först av kamerans aktuella position från SceneServer och beräknar därefter med hjälp av den nya positionen aktuell translation och rotation. Därefter flyttas kameran till den nya positionen en bild sparas undan (im1), varefter kameran flyttas i enlighet med baseline argumentet till den andra kamerans position. En ny bild sparas undan (im2) och funktionen returnerar stereoparet (im1 och im2) samt den simulerade IMU-datan (SSIMU).

4.3.6 SSGetData_batch()

Denna funktion tar som inargument sökvägen till batchkatalogen. Från denna katalog läses SSIMU-matrisen in (vilken innehåller simulerade IMU-data) samt stereoparbilderna som finns lagrade där i formatet png. Därefter uppdateras de globala stereobildsvariablerna med stereobilderna samtidigt som den globala IMU-data variabeln uppdateras med data från SSIMU matrisen. Detta sker tills all data ur datasetet med stereobilder och simulerad IMU-data liggandes i batchkatalogen har skickats till de globala variablerna.

AMASE

Accurate Multipoint Acquisition from Stereovision Equipment

Referenser

- [1] Todd A. Williamson. *A High-Performance Stereo Vision System for Obstacle Detection*, 1998. <http://www.cs.cmu.edu/~toddw/Thesis.pdf>
- [2] Christopher Mei. *Camera Calibration Toolbox for Matlab*, 2005. http://www.vision.caltech.edu/bouguetj/calib_doc/
- [3] Intel Corporation. *Intel Performance Primitives 5.0*, 2006. <http://www.intel.com/cd/software/products/asmo-na/eng/perflib/238654.htm>
- [4] Richard Hartley, Andrew Zisserman *Multiple View Geometry* 2000
- [5] Anna R. Bruss, Berthold K. P. Horn. *Passive Navigation*, 1983.
- [6] Darya Frolova, Denis Simakov . *Matching with Invariant Features*, 2004. www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/InvariantFeatures.ppt
- [7] Peter Kovesi. *MATLAB and Octave Functions for Computer Vision and Image Processing*, 2006. <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>
- [8] Bob Fisher, Simon Perkins, Ashley Walker and Erik Wolfart. *Canny Edge Detector*, 1994. <http://www.cee.hw.ac.uk/hipr/html/canny.html>
- [9] Gareth Loy, Alexander Zelinsky . *Fast Radial Symmetry for Detecting Points of Interest*, 2003. http://www.nada.kth.se/~gareth/homepage/local_site/papers/pami2003.pdf