Kandidatavhandling

Kandidatprogrammet i Datavetenskap

Samtidig lokalisering och kartläggning med datorseende

Sebastian Sergelius

5.11.2020

Matematisk-naturvetenskapliga fakulteten

Helsingfors universitetet

Handledare

Doktor Jeremias Berg

Examinatorer

Doktor Patrik Floréen

Kontaktinformation

PB 68 (Pietari Kalm gata 5)
00014 Helsingfors universitetet, Finland

E-postadress:: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Matematisk-naturvetenskapliga fakulteten	Kandidatprogrammet i Datavetenskap

Tekijä — Författare — Author

Sebastian Sergelius

Työn nimi — Arbetets titel — Title

Samtidig lokalisering och kartläggning med datorseende

Ohjaajat — Handledare — Supervisors

Doktor Jeremias Berg

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidatavhandling	5.11.2020	16 sidor

Tiivistelmä — Referat — Abstract

Robotars förmåga att navigera är ett forskningsområde som har gått framåt under de senaste årtionden. En drönare, som är en flygande robot, har använts för att skaffa visuell information av katastrofområden eller övervaka industriolyckor. Att en robot skulle kunna navigera autonomiskt måste den vara medveten om sitt läge, miljö, kursriktning och hastighet. En viktig del i autonomisk navigering för drönaren är att den kan samtidigt kartlägga omgivningen och lokalisera sig själv i omgivningen.

För att kameror är billiga och är inte beroende på utomstående hjälp för att skaffa information av omgivningen har vision baserad navigering fått mera uppmärksamhet än de traditionella metoder som till exempel satellitnavigation. Syftet med denna avhandling är att undersöka visionsbaserad navigering från drönarens perspektiv med fokus på samtidig kartläggning och lokalisering och hur detta problem har angripits.

ACM Computing Classification System (CCS)
Computer systems organization → Embedded and cyber-physical systems
→ Robotics → Robotic autonomy

Avainsanat — Nyckelord — Keywords

drönare, kartläggning, lokalisering, samtidig lokalisering och kartläggning, datorseende

Säilytyspaikka — Förvaringsställe — Where deposited

Helsingfors universitets bibliotek

Muita tietoja — övriga uppgifter — Additional information

Innehåll

1 Inledning

En drönare är ett obemannat luftfordon som kan styras med hjälp av vision baserad-, tröghets- eller satellitnavigation (Lu m. fl., 2018). För att robotar eller drönare skulle kunna navigera autonomiskt måste den vara medveten om sitt läge, omgivning, hastighet, kursriktning samt start- och slutplats. Andra faktorer som borde beaktas för att flyga från start till slutplatsen är hur drönare kan behandla data från inbyggda sensorer i sig, det vill säga kartlägga miljön och sin egen position i denna miljö samt hur man planerar vägen utan att kollidera med hinder.

För att drönare är mer versatila än robotar som inte flyger så finns det mera användningsfall för dem. De kan övervaka markföroreningar, industriolyckor eller växternas behov av vatten och näring (Motlagh m. fl., 2017). Drönare har också använts vid katastrofområden, såsom vid Japans jordbävning för att mäta strålningsvärden i Fukushima och få visuell information om katastrofområdet samt räddning av invandrare vid medelhavet. Mest lovande forskning är inom vision baserad navigering med hjälp av datorseende. I logistik är man också intresserad om autonoma drönaren som skulle leverera paket för kunder, till exempel 86 % av Amazons paket väger mindre än 5 skålpund (ca. 2.26 kg) (Charlie Rose, CBS News, u. å.).

Robotnavigering med hjälp av datorseende är ett aktivt forskningsområde (Desouza och Kak, 2002). För att robotar skulle vara verkligen autonomiska måste de kunna kartlägga sin omgivning och lokalisera sig själv i omgivningen, detta kallas för SLAM (Simultaneous Localization and Mapping). SLAM är ett problem som har fått uppmärksamhet av det vetenskapliga samfundet sedan 80-talet och har sätts som ett rön om det löses. Utmaningar i utvecklingen är att konstruera algoritmer som fungerar i allmänhet, såsom i dynamiska och stora miljön (Durrant-Whyte och Bailey, 2006).

Avhandlingen kommer att vara en översikt om SLAM problemet och vilka metoder det finns för att kartlägga miljön och lokalisera sig själv med hjälp av datorseende. Frågor som behandlas är: Vad är det fundamentala problemet med SLAM? Vilka SLAM metoder skulle kunna användas eller har använts med drönare och vad begränsar användningen av dessa? Varför har vision baserad navigation fått så mycket uppmärksamhet än de traditionella metoderna?

Resten av avhandlingen är strukturerad enligt följande: I kapitel 2 öppnas bakgrund bakom navigering, kartläggning, lokalisering och bildbearbetning som är nödvändiga för SLAM, kapitel 3 innehåller information om samtidig lokalisering och kartläggning och hur SLAM problemet kan grupperas i olika kategorier enligt data som man har.

2 Bakgrund

2.1 Navigering

Med navigering anser vi att en drönare planerar och utför en rutt från startplats till målet (Lu m. fl., 2018). För att kunna navigera till målet måste den vara medveten om sitt läge, miljö, kursriktning och hastighet. Autonomisk navigering kräver att drönare kan undvika hinder, planera sin rutt samt ta omvägar vid behov. I visionsbaserad navigering används visuella sensorer för att få bilder som indata. Bilderna kan sedan behandlas med algoritmer för att få en representation av omgivningen och lokalisera drönare i omgivningen.

Visuella sensorer som kan användas är monokulära, stereo, RGB-D och fisheye-kameror samt kombinationer av dessa (Lu m. fl., 2018). Kamerasensorer är billiga att operera, väger lite och med dem kan man uppfatta färg, textur och former. Traditionella navigeringsmetoder såsom satellit, som använder GPS-signaler (Global Positioning System) för lokalisering, och tröghetsnavigering (Inerital Navigation System), som använder accelerometer och gyroskop, får man inte visuell information av omgivningen. Med kameror kan man navigera i GPS-förnekade områden för att kameror är passiva, det vill säga de kan inte bli störda av utomstående signaler eller upptäckas av utomstående entitet.

Drönare kan använda laser och ultraljudsensorer för att navigera men dessa kräver mycket energi och väger mer än kameror (Fraundorfer m. fl., 2012). För att drönare byggs mindre än förr, och på grund av detta har begränsad bärförmåga samt batterikapacitet, har dessa sensorer uteslutnas. Med hjälp av vision baserad navigering, som använder kameror, kan man få rikligt med information om omgivningen (Lu m. fl., 2018). Visionsbaserad navigering, som är ett aktivt forskningsområde, är den mest lovande metoden inom robotnavigering.

2.2 Datorseende och bildbearbetning

Bildbearbetning i visionsbaserad navigering menar att man gör utdrag ur bilder i form av former, kanter, linjer, djuphet, rörelse, färg eller mönster (Desouza och Kak, 2002). Man kan också utjämna bilder (Se, D. Lowe m. fl., 2002). Detta menar att man blurrar bilder för att få bort brus och då man gör utdrag ur bilder i form av egenskaper får man färre, men bättre träffar. Metoder för att uppskatta djuphet med datorseende är att beräkna binokulära skillnaden eller rörelseparallax (Mansour m. fl., 2019).

Rörelseparallax betyder att objekt rör på sig snabbare då de är närmare observatören och långsammare då de är långt borta. För att räkna binokulära skillnaden behöver man två kameror som är riktade parallelt i samma linje, kamerornas distans från varandra är känd, deras synfält överlappar och av båda kamerornas bilder utdras samma kännetecken (Mansour m. fl., 2019). Från skillnaden ur landmärkets horisontala position i båda av kamerornas bilder kan man estimera djuphet. Med en monokulär kamera kan man uppskatta djuphet av bilder baserat på rörelseparallax. För att bygga djuphetskarta med monokulärkamera och rörelseparallax behöver man veta distansen till ett objekt då navigeringen börjar och som indata får man bilder och rörelseinformation av roboten (Mansour m. fl., 2019). Mansour m. fl., 2019 undersökte detta och märkte att stereokameror fungerar bättre då objekten är nära medan monokulära kameror med hjälp av rörelseparallax kan uppfatta mer precis distans då distansen växer.

För att få former, kanter eller linjer ur bilder finns det olika algoritmer såsom SIFT (Scale-Invariant Feature Transform), HARRIS hörn upptäckare, SURF (Speeded Up Robust Features), ORB (Oriented FAST and Rotated BRIEF), med mera (Rublee m. fl., 2011; Aulinas m. fl., 2008; Se, D. Lowe m. fl., 2002). SIFT, SURF och ORB prövar hitta egenskaper från bilderna som är invarianta för rotation, som menar att samma egenskaper kan hittas från olik synvinkel (Rublee m. fl., 2011). Enligt Rublee m. fl., 2011 är ORB mer energieffektiv än SIFT för att den kräver mindre beräkning för att hitta egenskaper, som menar att den är snabbare färdig utan att tappa effektivitet.

2.3 Lokalisering i omgivningen

Med att en robot lokaliserar sig själv menar vi att den tar reda på sin position i miljön (Desouza och Kak, 2002). Då man har en uppfattning om miljön, det vill säga en karta, så kan en drönare lokalisera sig själv med att ge den landmärken som den hittar då den navigerar eller så kan den observera dessa själv och associera dem med kartan som den har. Från bilderna ur kamerorna i drönaren identifieras landmärken, sedan matchas observerade landmärken med de som finns i kartan och efter det kan man estimera positionen av roboten.

Positionen och orientationen av roboten beräknas med sannolikhetsberäkning baserat på tid och indata av sensorer (Thrun m. fl., 2005). Då vi lokaliserar en robot så är roboten vid början vid någon plats, den rör på sig och sedan observerar landmärken. Då kan vi beteckna att för tidpunkt t, betecknas robotens position och orientation med x_t, rörelseinformation av roboten med u_t och landmärken extraherade från kamerasensorer med z_t. För att x_t är en kontinuerlig stokastisk variabel så kan betingade sannolikhetsfördelningen för x_t skrivas som

följande:

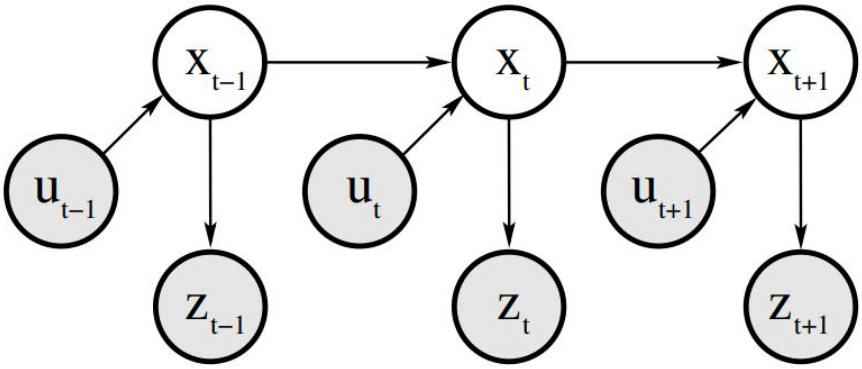
$$P(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) \tag{2.1}$$

Där vi antar att roboten rör på sig u_1 före den skaffar sensorinformation z_1. Med Markovegenskapen, som betyder att en villkorlig sannolikhetsfördelning för en tidpunkt är bara beroende på dens tidigare sannolikhet. Då x_t är den bästa framtidssyn så kan vi begränsa att x_t är beroende bara på dens tidigare position x_{t-1} och rörelseinformation u_t, se figur 2.1. Betingade sannolikhetsfördelningen för x_t kan då betecknas som följande:

$$P(x_t|x_{t-1}u_t) \tag{2.2}$$

Om x_t är den bästa framtidssyn så då är betingade sannolikhetsfördelningen för observerade landmärken z_t:

$$P(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) = P(z_t|x_t) \tag{2.3}$$



Figur 2.1: En graf om lokalisering som visualiserar Markovegenskapen där till exempel x_t är bara beroende av x_{t-1} och u_t (Thrun m. fl., 2005).

$P(x_t|x_{t-1}u_t)$ är rörelseuppfattningssannolikhet och $P(z_t|x_t)$ är landmärkesuppfattningssannolikhet för roboten (Thrun m. fl., 2005).

Positionen och orientationen av roboten kan beräknas med Bayes filter (Thrun m. fl., 2005). Bayes filter är en rekursiv algoritm som beräknar aposteriorisannolikhet för robotens position $bel(x_t)$ baserat på robotens tidigare position $bel(x_{t-1})$, rörelseinformation u_t och observerade

landmärken z_t. Med hjälp av Bayes regler, lagen om total sannolikhet för kontinuerliga variablar och Markovegenskapen kan detta bevisas med matematisk induktion där vi antar at $bel(x_0)$ är sann då $t = 0$, altså roboten vet sin position vid början.

$$
\begin{aligned}
bel(x_t) &= P(x_t|z_{1:t}, u_{1:t}) && \text{(BF1)}\\
&= \eta P(z_t|x_t, z_{1:t-1}, u_{1:t}) P(x_t|z_{1:t-1}, u_{1:t}) && \text{(BF2)}\\
&= \eta \underline{P(z_t|x_t)} P(x_t|z_{1:t-1}, u_{1:t}) && \text{(BF3)}\\
&= \eta P(z_t|x_t) \underline{\int P(x_t|x_{t-1}, z_{1:t-1}, u_{1:t}) P(x_{t-1}|z_{1:t-1}, u_{1:t}) dx_{t-1}} && \text{(BF4)}\\
&= \eta P(z_t|x_t) \int \underline{P(x_t|x_{t-1}, u_t)} P(x_{t-1}|z_{1:t-1}, u_{1:t}) dx_{t-1} && \text{(BF5)}\\
&= \eta P(z_t|x_t) \int P(x_t|x_{t-1}, u_t) P(x_{t-1}|z_{1:t-1}, \underline{u_{1:t-1}}) dx_{t-1} && \text{(BF6)}\\
&= \eta P(z_t|x_t) \int P(x_t|x_{t-1}, u_t) \underline{bel(x_{t-1})} dx_{t-1} && \text{(BF7)}
\end{aligned}
$$

Bayes filters definition är BF1, det vill säga förtroende för roboten att den är vid x_t, som betecknas $bel(x_t)$, är samma som sannolikheten av x_t givet $z_{1:t}$ och $u_{1:t}$. Med Bayes teorem kan detta skrivas om i format BF2. Vid BF3 använder vi Markovegenskapen för x_t så kan betingade sannolikheten för z_t skrivas som landmärkesuppfattningssannolikheten. Med satsen om total sannolikhet för kontinuerliga variablar, som uttrycker sannolikhet för enskilda händelser till betingade sannolikhet, kan formeln skrivas om till BF4. Vid BF5 och BF6 använder vi Markovegenskapen för x_t, se figur 2.1. Till slut får vi BF7, som bevisar med induktion att algoritmen fungerar rekursivt (Thrun m. fl., 2005).

Bayes filters algoritmen fungerar så att för varje x_t beräknas två kritiska steg, som är spådom och korrigering (Thrun m. fl., 2005). Spådomsdelen i algoritmen är delen där vi beräknar integralen för två faktorer: sannolikheten att rörelseinformationen u_t tar oss till x_t och den gamla förtroende av positionen $bel(x_{t-1})$. Korrigeringsdelen är då man multiplicerar spådomsdelen med sannolikhet att z_t skulle observeras för varje möjliga x_t. Oftast är korrigeringsdelen inte en sannolikhet, alltså dens täthetsfunktion summa är inte 1 för varje värde, så därför används det en normering konstant η för att normera sannolikheten. Ett steg av algoritmen är definierat vid 1 som sedan görs rekursivt för resultatet av algoritmen och när man har ny

rörelseinformation och landmärken.

Algorithm BFA: Bayes Filter Algoritm

`BayesFilter(`$bel(x_{t-1})$`, `u_t`, `z_t`):`

forall x_t **do**

$sp\mathring{a}dom = \int P(x_t|x_{t-1}, u_t)bel(x_{t-1})dx_{t-1}$

$bel(x_t) = \eta P(z_t|x_t)\ sp\mathring{a}dom$

end

return $bel(x_t)$

Bayes filter är huvudsakliga algoritmen för att beräkna robotens position och orientation (Thrun m. fl., 2005). Algoritmen baserar sig starkt på Markovegenskapen att det nuvarande läge är oberoende av tidigare data och framtidsdata. I robotnavigering är det inte dock så lätt att anta detta, för det skulle betyda att allt runtomkring roboten borde vara statiskt. Till exempel om en bil rör på sig då roboten navigerar så då man gör landmärkesuppfattning kommer Bayes filter att ge felaktig estimering. Bayes filtern är också basis för många andra lokaliseringsalgoritmer, som till exempel Kalman filter eller Markov localisation.

Positionen av roboten kan beräknas också utan kartan med att räkna distansen till landmärken och distansen mellan landmärken som extraheras ur bilder (Durrant-Whyte och Bailey, 2006). Lokalisering kan delas i global och lokal lokalisering (Desouza och Kak, 2002; Se, D. G. Lowe m. fl., 2005). Med global lokalisering har roboten ingen vetskap om sin position i början av navigeringen. I lokal lokalisering har roboten en ungefärlig eller exakt vetskap om sin plats vid början av navigeringen som den fått som indata. Lokala lokaliseringsmetoder strävar för att korrigera fel som uppstår av robotens rörelse. Globala lokaliseringsmetoder kan återvinna från verkliga misstag då man estimerar robotens position.

2.4 Kartläggning av omgivningen

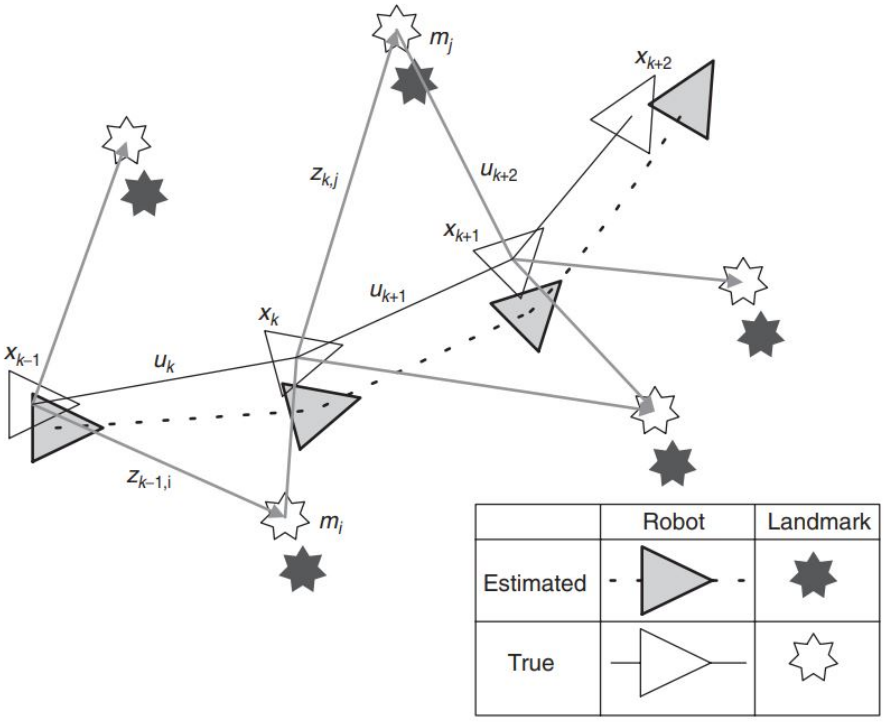
En karta om miljön kan representeras i två (2D) eller tre (3D) dimensioner (Lu m. fl., 2018). Kartor kan sparas i olika format, såsom datorstödd konstruktion (CAD, Computer-Aided Design), beläggningskarta (Occupancy Map), topologisk karta eller en enkel graf om landmärken och deras sammankopplingar (Desouza och Kak, 2002). En datorstödd konstruktion av miljön kan vara en mycket detaljerad representation av omgivningen. En beläggningskarta är en 2D eller 3D-modell av omgivningen som är sparat i ett rutsystem där rutor som är upptagna är någon objekt i miljön (Heng m. fl., 2011; Desouza och Kak, 2002).

Kartan kan vara färdigt sparad för en drönare eller så kan miljön kartläggas från bilderna av sensorerna då den flyger (Lu m. fl., 2018). Med 3D volymetriska sensorer kan man konstruera en 3D modell och spara denna information i en Octree struktur. Med strukturen kan data om

miljön packas i mindre format utan att tappa möjligheten att uppdatera informationen vid behov. En annan metod som tas upp är med stereovisionsalgoritmer göra en djuphetskarta och behandla data till plana ytor som minskar på missvisning som uppstår med användning av stereovision algoritmer när man bygger upp djuphetskartan.

3 Samtidig lokalisering och kartläggning

Samtidig lokalisering och kartläggning (SLAM) är ett av de grundläggande problem i robotnavigering (Durrant-Whyte och Bailey, 2006). SLAM är ett problem där en autonom robot som inte har tidigare information om sin plats eller omgivning skall samtidigt bygga en karta och lokaliserar sig själv till exempel med hjälp av att identifiera landmärken. Detta kan estimeras med sannolikhetsberäkning baserat på tid k, där man tar i beaktande riktningen och positionen av roboten x_k, distansen roboten rör på sig u_k, landmärken som är invariant för rörelse m_i och observationer för varje landmärke i som roboten gör vid varje tidpunkt z_{ik}, se figur 3.1. Några lösningar för SLAM som baserar sig på sannolikheträkning är Extended Kalman Filter (EKF-SLAM) och FastSLAM (Durrant-Whyte och Bailey, 2006).



Figur 3.1: Bild på SLAM-problemet (Durrant-Whyte och Bailey, 2006).

SLAM-problemet har delproblem som borde lösas för att få robotar att navigera autonomiskt (Aulinas m. fl., 2008). Det svåraste problemet i flesta SLAM-lösningar är dataföreningsproblemet som betyder att man identifierar två olika landmärken som en och samma. Detta problem kan uppstå redan vid korta rörelsen av robotar eller då en robot har navigerat och

kommer till en plats som den har redan varit i förr, detta kallas för loopstängning (Loop Closure). Med att förena landmärken fel så uppkommer det misvisningar då man estimerar positionen av roboten.

För autonom navigering behöver en robot veta sitt läge i miljön (Lu m. fl., 2018). Med hjälp av kameror, bildbearbetning och beräkning kan miljön kartläggas, i helhet eller delvis, och drönaren lokalisera sig själv, detta kallas också för visuell lokalisering och kartläggning. Detta kan delas i tre kategorier, som är kartlösa (mapless), kartbaserade (map-based) och kartbyggande (map-building) system.

3.1 Kartlösa system

I system utan karta navigerar drönare bara med hjälp av att observera tydliga egenskaper i miljön (Desouza och Kak, 2002). Dessa kan vara till exempel väggar, dörrar, möbler eller andra landmärken. Metoder som används inom kartlösa system är optiskt flöde (Optical flow) och spårning av egenskaper (Feature tracking).

3.1.1 Optiskt flöde

Santos-Victor et. al har använt optiskt flöde i en robot för att imitera bin (Santos-Victor m. fl., 1993). De placerade två kameror på varsin sida av en robot och beräknade hastigheten från bilderna av båda kamerorna med att ta fem bilder som utjämnas med Gaussisk oskärpa och de två sista utjämnade bilder används för att beräkna medeltalet av optiska flödesvektorer. Om flödesvektorerna var samma på båda sidorna så for roboten rakt framåt, annars så far den mot den sida vilkens hastighet är mindre. Metoden som Santos-Victor et al. använde för roboten fungerar bara om båda kamerorna är symmetriskt riktade från varandra när man tar i beaktande rörelseriktningen av roboten (Desouza och Kak, 2002). Denna teknik fungerar dåligt i texturlösa miljö och det går inte att byta rörelseriktningen. Fastän metoden som Santos-Victor m. fl., 1993 använde tog bara i beaktande horisontala flödesvektorer så har sedan detta användning av optiska flödesmetoder forskats mera. Nuförtiden kan man använda drönare för att uppskatta avstånd, hålla sin höjd, undvika hinder, beräkna hastighet och landa på en plattform som rör på sig med hjälp av optiskt flöde (Chao m. fl., 2013).

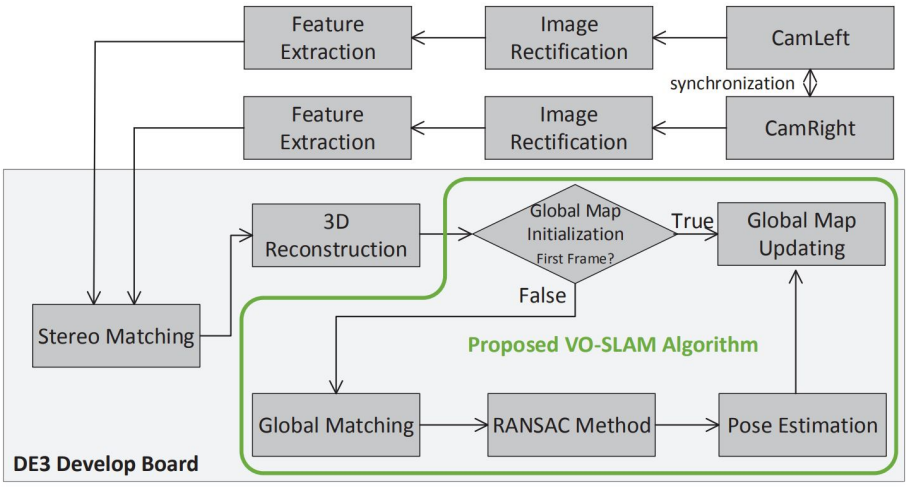
3.1.2 Spårning av egenskaper

Spårning av egenskaper (Feature tracking) används för att skaffa information om objekt, så som linjer, hörn och olika former som är invarianta för rörelse (Lu m. fl., 2018). Med hjälp

av dessa egenskaper av objekten och deras relativa rörelse i sekventiella bilder kan man bestämma robotens position. Då drönare navigerar i området, så kommer den troligtvis att se dessa objekt från olika perspektiv, som hjälper drönare att navigera. Traditionella SLAM-lösningar, såsom EKF-SLAM och FastSLAM faller under denna kategorin (Latif och Saddik, 2019). Dessa SLAM-lösningar fungerar inte då det finns mycket egenskaper att extrahera i miljön. EKF-SLAM nackdel är beräkningsbehovet som växer kvardatiskt och FastSLAM nackdel är bildprocessering.

För att drönaren har begränsad batterikapacitet och bärkraft har Gu m. fl., 2015 föreslagit användning av VO-SLAM (Visual Odometry SLAM) som kan kartlägga omgivningen och lokalisera observatören då det finns upp till 30 000 egenskaper i kartan, med en stereokamera som tar 31 bilder per sekund och en krets som använder tio gånger mindre energi än en Intel i7 3770K-processor (Gu m. fl., 2015). Deras VO-SLAM algoritm implementeras i en FPGA-krets (Field-Programmable Gate Array) vilkens logik är omprogrammerbar, den använder färre energi än vanliga processorer och kan beräkna parallelt.

VO-SLAM-lösningen fungerar så att de extrahera landmärken med SIFT från bilderna (Gu m. fl., 2015). Med hjälp av binokulära skillnaden från bilderna i stereokameran konstruera de en 3D-representation av dessa landmärken som de sparar i en matris. FPGA-kretsen, som är programmerat för matrisberäkning, beräknar vektorvinklarna för varje observerade landmärke från globala kartan och hittar de bästa träffar för lokaliseringsalgoritmen. Före lokaliseringsuppfattningen använder de RANSAC (Random Sample Consensus) för att ta bort avvikande träffar inom landmärken, se figur 3.2.



Figur 3.2: Bild på VO-SLAM processen (Gu m. fl., 2015).

Med hjälp av FPGA-kretsen har de fått dataförening med matriser att beräknas i realtid med upp till 31 bilder i sekund och liten energibehov. Enligt Gu m. fl., 2015 kan VO-SLAM

estimera positionen med 1-2 cm exakthet med upp till 30 000 landmärken i kartan, hantera loopstängning och är mest energieffektiva lösningen (Gu m. fl., 2015). Något som deras lösning tar inte i beaktande är SIFT algoritmens tidkrav. Latif och Saddik, 2019 använder VO-SLAM där de delat upp processen så att indata från stereokameran, bildbearbetning och 3D-representationen som görs med SIFT bearbetas i en egen processor och uppdatering av kartan, matcha observerade landmärken med kartan, lokalisering och uppdatering av kartan hanteras i FPGA-kretsen (Latif och Saddik, 2019).

3.2 Kartbaserade system

Med kartbaserade system har drönare en färdig vetskap om miljön som kan vara i form av geometriska modeller, topologiska kartor eller förhållande mellan landmärken (Desouza och Kak, 2002). Idén är att då roboten navigerar prövar den hitta landmärken från bilder som är lika till de landmärken som roboten vet om. När den hittat dessa så kan roboten beräkna sin position i miljön. Med hjälp av karta kan drönare planera sin rörelse i förhand och ta omvägar där det behövs (Lu m. fl., 2018).

Kartbaserad lokalisering med datorseende kan delas upp i fyra steg, som är följande (Desouza och Kak, 2002):

- skaffa sensorinformation av kamerorna
- upptäcka landmärken från informationen med bildbearbetning
- matcha observationerna med förväntan
- beräkna positionen av roboten

Det svåraste steget av dessa är att matcha observationerna med förväntan. Detta kallas för dataföreningsproblemet. Man kan inte med full säkerhet veta robotens position och då är det svårt att matcha landmärken från rätt synvinkel.

Då kartan finns kan man fokusera på lokalisering av roboten (Desouza och Kak, 2002). I global lokalisering måste man lita på att man kan förena observationerna med informationen man har och ta i beaktande osäkerheten med att observationerna kan matcha flera av de landmärken man vet om. Detta problem kan lösas till exempel med Monte Carlo lokalisation eller Markov lokalisation. Monte Carlo lokalisation fungerar så att en robot antar med lika sannolikhet sin position i kartan vid början av navigeringen (Dellaert m. fl., 1999). Då roboten rör på sig så observerar den nya landmärken och kan beräkna sannolikheten att hur de

landmärken den observerat matchar de landmärken i kartan som den vet om. Markov lokalisationsalgoritmen är nästan identisk till Bayes filtern som presenterades vid 1. Som indata används också kartan som man har för att estimera positionen av roboten.

I lokal lokalisering måste lokaliseringsalgoritmen hålla koll på osäkerheten av robotens position då den rör på sig och vänder (Dellaert m. fl., 1999). Då osäkerhet av sin position är för stor stannar roboten och beräknar sin position med hjälp av visuell information av bilderna, alltså minskar på osäkerheten.

3.3 Kartbyggande system

Kartbyggande system kan användas då det är svårt att navigera med en existerande karta om omgivningen eller om kartan inte finns, som i katastrofområden (Lu m. fl., 2018). Roboten kan vara medveten eller omedveten om sin position vid början av kartbyggandet (Se, D. G. Lowe m. fl., 2005). Kartbyggande systems bildbearbetning kan delas tre kategorier, som är indirekta, direkta och hybrida metoder, som sammanslår indirekta och direkta metoder (Lu m. fl., 2018).

3.3.1 Indirekta metoder

I bildbearbetning som använder indirekta metoder tar man kännetecken ur bilden som är invariant för rotation, synvinkeländringar och rörelseoskärpa, dessa ges som indata som sedan kan användas för rörelseuppfattning och lokalisering (Lu m. fl., 2018).

Ett sätt att konstruera en karta är att beakta robotens rörelse och synvinkel (Se, D. G. Lowe m. fl., 2005). Se, D. Lowe m. fl., 2002 har gjort dessa samt användt indirekta metoder i sin artikel ”Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks” för att bygga en 3D-karta (Se, D. Lowe m. fl., 2002). De tar bilder från stereokameror, utjämnar bilderna och använder SIFT-algoritm för att extrahera egenskaper ur bilderna. Med denna metod har de kunnat konstruera en 3D karta av omgivningen baserat på landmärken utan att spara korrelationsmatris mellan landmärken som minskar häftigt på behov av beräkning. Denna metod har ändå problem då det kommer till loopstängning och uppdatering av kartan (Se, D. G. Lowe m. fl., 2005). Med att använda samma spårning av egenskaper, kartlägga delar av områden och senare bygga en stor global karta fick forskarna loopstängning och kart-uppdateringen löst. De fick dessa problem löst med att kartlägga delar av kartan och från dessa delar som var bredvid varandra kunde de identifiera landmärken och foga ihop kartorna till en stor karta. Indirekta metoder fungerar dåligt i strukturlösa

miljön, för det finns inte egenskaper att extrahera ur bilder (Lu m. fl., 2018).

3.3.2 Direkta metoder

Direkta metoder fungerar bättre i strukturlösa miljön (Engel m. fl., 2014). Som i indirekta metoder där man prövar hitta många små kännetecken ur bilder så i direkta metoder använder man hela bilden för att hitta geometriska egenskaper. Med hjälp av dessa så kan man konstruera en detaljerad karta med extra processorberäkning (Lu m. fl., 2018).

4 Sammafattning

Referenser

Aulinas, J., Petillot, Y., Salvi, J. och Llado, X. (jan. 2008). ”The SLAM problem: a survey”. I: vol. 184, s. 363–371. DOI: 10.3233/978-1-58603-925-7-363.

Chao, H., Gu, Y. och Napolitano, M. (2013). ”A survey of optical flow techniques for UAV navigation applications”. I: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, s. 710–716. DOI: 10.1109/ICUAS.2013.6564752.

Charlie Rose, CBS News (u. å.). *Amazon's Jeff Bezos looks to the future*. https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/, läst 02.11.2020.

Dellaert, F., Fox, D., Burgard, W. och Thrun, S. (1999). ”Monte Carlo localization for mobile robots”. I: *Proceedings of 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. Vol. 2, 1322–1328 vol.2. DOI: 10.1109/ROBOT.1999.772544.

Desouza, G. N. och Kak, A. C. (2002). ”Vision for mobile robot navigation: a survey”. I: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2, s. 237–267. DOI: 10.1109/34.982903.

Durrant-Whyte, H. och Bailey, T. (2006). ”Simultaneous localization and mapping: part I”. I: *IEEE Robotics Automation Magazine* 13.2, s. 99–110. DOI: 10.1109/MRA.2006.1638022.

Engel, J., Schöps, T. och Cremers, D. (2014). ”LSD-SLAM: Large-Scale Direct Monocular SLAM”. I: *ECCV*. ISBN: 978-3-319-10604-5. DOI: 10.1007/978-3-319-10605-2_54.

Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P. och Pollefeys, M. (2012). ”Vision-based autonomous mapping and exploration using a quadrotor MAV”. I: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, s. 4557–4564. DOI: 10.1109/IROS.2012.6385934.

Gu, M., Guo, K., Wang, W., Wang, Y. och Yang, H. (2015). ”An FPGA-based real-time simultaneous localization and mapping system”. I: *2015 International Conference on Field Programmable Technology (FPT)*, s. 200–203. DOI: 10.1109/FPT.2015.7393150.

Heng, L., Lee, G. H., Fraundorfer, F. och Pollefeys, M. (2011). ”Real-time photo-realistic 3D mapping for micro aerial vehicles”. I: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, s. 4012–4019. DOI: 10.1109/IROS.2011.6095058.

Latif, R. och Saddik, A. (2019). ”SLAM algorithms implementation in a UAV, based on a heterogeneous system: A survey”. I: *2019 4th World Conference on Complex Systems (WCCS)*, s. 1–6. DOI: 10.1109/ICoCS.2019.8930783.

Lu, Y., Xue, Z., Xia, G.-S. och Zhang, L. (2018). ”A survey on vision-based UAV navigation”. I: *Geo-spatial Information Science* 21.1, s. 21–32. DOI: 10.1080/10095020.2017.1420509.

Mansour, M., Davidson, P., Stepanov, O. och Piché, R. (2019). ”Relative Importance of Binocular Disparity and Motion Parallax for Depth Estimation: A Computer Vision Approach”. I: *Remote Sensing* 11.17. DOI: https://doi.org/10.3390/rs11171990.

Motlagh, N. H., Bagaa, M. och Taleb, T. (2017). ”UAV-Based IoT Platform: A Crowd Surveillance Use Case”. I: *IEEE Communications Magazine* 55.2, s. 128–134. DOI: 10.1109/MCOM.2017.1600587CM.

Rublee, E., Rabaud, V., Konolige, K. och Bradski, G. (2011). ”ORB: An efficient alternative to SIFT or SURF”. I: *2011 International Conference on Computer Vision*, s. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

Santos-Victor, J., Sandini, G., Curotto, F. och Garibaldi, S. (1993). ”Divergent stereo for robot navigation: learning from bees”. I: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, s. 434–439. DOI: 10.1109/CVPR.1993.341094.

Se, S., Lowe, D. G. och Little, J. J. (2005). ”Vision-based global localization and mapping for mobile robots”. I: *IEEE Transactions on Robotics* 21.3, s. 364–375. DOI: 10.1109/TRO.2004.839228.

Se, S., Lowe, D. och Little, J. (aug. 2002). ”Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks”. I: *The International Journal of Robotics Research* 21, s. 735–760. DOI: 10.1177/027836402761412467. URL: https://www.cs.ubc.ca/~lowe/papers/ijrr02.pdf.

Thrun, S., Burgard, W. och Fox, D. (2005). *Probabilistic robotics*. Intelligent robotics and autonomous agents. MIT Press, s. 1–647. ISBN: 978-0-262-20162-9.