



Kandidatavhandling

Kandidatprogrammet i Datavetenskap

# Samtidig lokalisering och kartläggning med datorseende

Sebastian Sergelius

12.11.2020

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](mailto:info@cs.helsinki.fi)

URL: <http://www.cs.helsinki.fi/>

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Matematiske-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	12.11.2020	19 sidor	
Tiivistelmä — Referat — Abstract			
<p>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 och att övervaka industriolyckor. Att en robot skulle kunna navigera autonomiskt måste den vara medveten om sitt läge, miljö, kursriktning och hastighet. En viktig del i autonom navigering för drönare är att den kan samtidigt kartlägga omgivningen och lokalisera sig själv i omgivningen.</p> <p>Med kameror kan man extrahera egenskaper ur bilder och kameror är inte beroende på utomstående entitet för att skaffa data om omgivningen. Visionsbaserad navigering med kameror är ett primärt forskningsområde inom robotnavigering. 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.</p>			
<p><b>ACM Computing Classification System (CCS)</b>  Computer systems organization → Embedded and cyber-physical systems  → Robotics → Robotic autonomy</p>			
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

<b>1</b>	<b>Inledning</b>	<b>1</b>
<b>2</b>	<b>Bakgrund</b>	<b>3</b>
2.1	Navigering . . . . .	3
2.2	Datorseende och bildbearbetning . . . . .	3
2.3	Lokalisering i omgivningen . . . . .	5
2.4	Kartläggning av omgivningen . . . . .	8
<b>3</b>	<b>Samtidig lokalisering och kartläggning</b>	<b>10</b>
3.1	Kartlösa system . . . . .	11
3.1.1	Optiskt flöde . . . . .	11
3.1.2	Spårning av egenskaper . . . . .	12
3.2	Kartbaserade system . . . . .	13
3.3	Kartbyggande system . . . . .	15
3.3.1	Indirekta metoder . . . . .	15
3.3.2	Direkta metoder . . . . .	16
<b>4</b>	<b>Sammanfattning</b>	<b>17</b>
	<b>Referenser</b>	<b>18</b>

# 1 Inledning

En drönare är ett obemannat luftfordon som kan styras med hjälp av visionsbaserad-, tröghets- eller satellitnavigation (Lu m. fl., 2018). För att en 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 särskilda 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 jordbävningen nära Japan för att mäta strålningsvärden nära Fukushima och att få visuell information om katastrofområdet samt i räddning av invandrare vid medelhavet. Inom logistikbranschen är man 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) och de har tänkt på att leverera paket inom 30 minuter för människor som bor nära Amazons lager (Charlie Rose, CBS News, u. å.). Lovande forskning för autonoma robotar är inom visionsbaserad navigering med hjälp av datorseende.

Robotnavigering med hjälp av datorseende är ett aktivt forskningsområde (Desouza och Kak, 2002). För att robotar skulle vara verkligen autonoma 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 miljöer (Durrant-Whyte och Bailey, 2006). Sannolikhetsberäkning är tillvägagångssättet för att lösa SLAM-problemet och flesta av de algoritmer som har löst problemet baserar sig på sannolikhetsberäkning (Thrun m. fl., 2005).

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 visionsbaserad navigation fått så mycket uppmärksamhet än de traditionella metoderna?

Resten av avhandlingen är strukturerad enligt följande: I kapitel 2 öppnas bakgrund bakom visionsbaserad navigering. Vad är visionsbaserad navigering och vilka sensorer kan användas samt vilka metoder det finns för att extrahera data ur bilder. Kapitel 2 kommer också att innehålla basen för de flesta lokaliseringsalgoritmer som baserar sig på sannolikhetsberäkning och vad vi menar med kartläggning. Dessa är nödvändiga att förstå SLAM problemet. Kapitel 3 är om samtidig lokalisering och kartläggning och hur SLAM problemet kan grupperas i olika kategorier enligt data som man har. Kapitel 3 kommer också att innehålla användningsfall för några av de metoderna. Kapitel 4 är en sammanfattning om avhandlingen.

# 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. Autonom navigering för drönare kräver att den 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 till exempel färg, textur och former. Med traditionella navigeringsmetoder, såsom satellitnavigering som använder GPS-signaler (Global Positioning System) för lokalisering, och tröghetsnavigering (Inertial 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. Med visionsbaserad navigering som använder kameror så är en drönare inte beroende på utomstående hjälp, därmed kan man beräkna drönarens position i omgivningen inombords (Kehoe m. fl., 2006).

Drönare kan använda laser- och ultraljudsensorer för att navigera, men dessa kräver mer 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 uteslutats. Med hjälp av visionsbaserad navigering som använder sig av kameror kan man få rikligt med information om omgivningen (Lu m. fl., 2018). Visionsbaserad navigering, som är ett aktivt forskningsområde, är en av de mest lovande metod för robotnavigering.

## 2.2 Datorseende och bildbearbetning

Med datorseende kan man få sekventiella bilder ur omgivningen som sedan kan bearbetas med algoritmer (Desouza och Kak, 2002). Bildbearbetning i visionsbaserad navigering betyder att man gör utdrag ur bilder i form av former, kanter, linjer, djuphet, rörelse, färg eller mönster.

Då man extraherar egenskaper ur bilder är det möjligt att utjämna bilder (Se, D. Lowe m. fl., 2002). Detta menar att man gör bilder otydligare för att få bort digitalt brus och att bilden har mindre detalj. Då man extraherar egenskaper med algoritmer ur bilder som är otydligare 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 i synfältet då de är närmare observatören och långsammare då de är långt borta, detta gäller också om observatören rör på sig och objektet står stilla (Mansour m. fl., 2019; Rogers och Graham, 1979). För att räkna binokulära skillnaden för observerade kännetecken behöver man två kameror som är riktade parallellt i samma linje så att deras synfält överlappar, kamerornas distans från varandra är känd och av båda kamerornas bilder utdras samma kännetecken. Från skillnaden ur kännetecknets horisontala position i båda av kamerornas bilder kan man estimeras djuphet. Med en monokulär kamera kan man uppskatta djuphet av bilder baserat på rörelseparallax. För att estimeras djuphet med monokulärkamera och rörelseparallax behöver man veta distansen till ett objekt då navigeringen börjar och som indata får man bilder och rörelseinformation av roboten. Rörelseinformationen i robotar på stagid grund kan man få från vägmätare och tröghetsmätare som är inbyggt i roboten.

Mansour m. fl., 2019 undersökte detta och märkte att stereokameror fungerar bättre då objektet är nära medan monokulära kameror med hjälp av rörelseparallax kan uppfatta mer precis distans då distansen växer (Mansour m. fl., 2019). Med stereokamera kan djupheten estimeras bra upp till cirka 10 meters avstånd och vid 20 meters avstånd så är estimeringarna oanvändbara, efter detta måste man växla över till att använda monokulära kameran. Mansour m. fl., 2019 skriver att i framtiden kommer de att forska hur de kan beräkna rörelseinformationen från stereokamerornas bilder så att de kan avlägsna tröghets- och vägmätaren. Detta ger möjlighet att estimeras djuphet till exempel för drönare.

Fast man har fördelar på data som man kan extrahera ur kameror jämfört med andra sensorer som till exempel laser och ultraljud, så med kameror är förändring i belysning, vädret och årstid samt skuggor ett problem som måste tas i beaktande, för det mesta då man navigerar utomhus (Desouza och Kak, 2002). Dessa problem har lösts med att bearbeta bildernas färgnyanser, färgmättnad (saturation) och ljusstyrka.

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 invariant 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 algoritmen än SIFT för att den kräver mindre beräkning för att extrahera egenskaper ur bilder, som betyder 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 estimeras positionen av roboten.

Då roboten navigerar så beräknas positionen och orientationen av roboten med sannolikhetsberäkning baserat på tid och indata av sensorer (Thrun m. fl., 2005). Under navigeringen vill roboten estimeras sin position efter varje rörelse. Detta gör den med att estimeras vart rörelseorder kommer att ta den och efter att den har rört på sig observerar den landmärken. Med att den observerar landmärken efter rörelsen kan roboten förbättrar sin förtroelse om sin position.

För att estimeras detta så kan man beteckna robotens position och riktning i miljön för varje tidpunkt  $t$  med  $x_t$ . Variabeln  $x_t$  beskriver alltså robotens tillstånd, som i drönarens fall kan vara till exempel drönarens koordinater och orientering i miljön med sex frihetsgrader. Rörelseorder betecknas som  $u_t$  och landmärken extraherade från kameran sensorer med  $z_t$ . För att  $x_t$  är en möjlighet i en kontinuerlig stokastisk variabel  $X$  så kan betingade sannolikhetsfördelningen för  $x_t$  skrivas som följande:

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

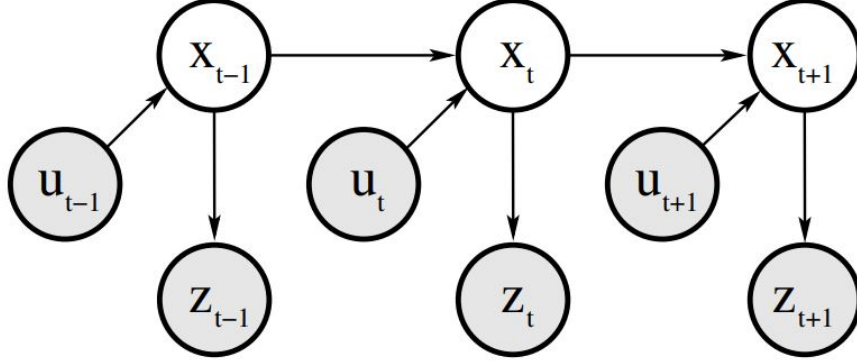
Där vi antar att roboten rör på sig  $u_1$  före den skaffar sensorinformation  $z_1$ . Med Markovegenskapen för  $x_t$ , som betyder att en villkorlig sannolikhetsfördelning för  $x_t$  är bara beroende på den tidigare sannolikhet  $x_{t-1}$  och inte på händelseförloppet som hände före  $x_{t-1}$ .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

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

Då  $x_t$  har Markovegenskapen så är betingade sannolikhetsfördelningen för observerade land-

märken  $z_t$ :

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



**Figur 2.1:** En graf om lokalisering som visualiserar Markovegenskapen där till exempel  $x_t$  är bara beroende av  $x_{t-1}$  och  $u_t$ , detta kallas också för en Markovkedja (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 riktning av roboten kan beräknas med Bayes filter som använder sig av rörelse- och landmärkesuppfattningssannolikhet (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 variabler och Markovegenskapen kan detta bevisas med matematisk induktion där vi antar att  $x_t$  har Markovegenskapen och att  $bel(x_0)$  sannolikhetsfunktion är känd då  $t = 0$ . Med att sannolikhetsfunktionen är känd menar vi att roboten vet sin position vid början med full säkerhet eller så är  $bel(x_0)$  en likformig sannolikhetsfördelning.

$$bel(x_t) = P(x_t|z_{1:t}, u_{1:t}) \quad (BF1)$$

$$= \eta P(z_t|x_t, z_{1:t-1}, u_{1:t}) P(x_t|z_{1:t-1}, u_{1:t}) \quad (BF2)$$

$$= \eta \underline{P(z_t|x_t)} P(x_t|z_{1:t-1}, u_{1:t}) \quad (BF3)$$

$$= \eta P(z_t|x_t) \int \underline{P(x_t|x_{t-1}, z_{1:t-1}, u_{1:t})} P(x_{t-1}|z_{1:t-1}, u_{1:t}) dx_{t-1} \quad (BF4)$$

$$= \eta P(z_t|x_t) \int \underline{P(x_t|x_{t-1}, u_t)} P(x_{t-1}|z_{1:t-1}, u_{1:t-1}) dx_{t-1} \quad (BF5)$$

$$= \eta P(z_t|x_t) \int P(x_t|x_{t-1}, u_t) \underline{bel(x_{t-1})} dx_{t-1} \quad (BF6)$$

$$= \eta P(z_t|x_t) \int P(x_t|x_{t-1}, u_t) \underline{bel(x_{t-1})} dx_{t-1} \quad (BF7)$$

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 variabler, 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 Markovkedjan i 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 täthetsfunktionens summa är inte 1 för varje värde, därför används det en normering konstant  $\eta$  för att normera sannolikheten. Ett steg av algoritmen är definierat nedan, se algoritmen 1, som sedan görs rekursivt för resultatet av algoritmen och när man har ny rörelseinformation och landmärken. Utmaning i algoritmen är integralen i spådomsdelen. Man borde begränsa robotens tillstånd att vara en ändlig variabel så att integralen över spådomsdelen blir en summa.

---

**Algorithm BFA:** Bayes Filter Algorithm

---

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

**forall**  $x_t$  **do**

$spådom = \int P(x_t|x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1}$   
 $bel(x_t) = \eta P(z_t|x_t) smådom$

**end**

**return**  $bel(x_t)$

---

Bayes filter är huvudsakliga algoritmen för att beräkna robotens position och riktning (Thrun m. fl., 2005). Algoritmen baserar sig starkt på Markovegenskapen att det nuvarande läge är oberoende av tidigare data. I robotnavigering är det inte dock så lätt att anta detta, för det skulle betyda att allt runtomkring roboten borde vara statistiskt. 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 lokalisation. Oftast är det inte möjligt att begränsa robotens tillstånd så att integralen i Bayes filtern blir en summa över spådomsdelen. Algoritmer som används för lokalisering måste därför beräkna ett approximativt värde för detta och det är här som olika lokaliseringsalgoritmer skiljer sig.

Med att approximera värdet i lokaliseringsalgoritmer så avstår man oftast från någon egenkap som är viktigt för robotnavigering, såsom beräkningseffektivitet, noggrannhet i approximationen av tillståndet eller enkel implementering av algoritmen för robotar (Thrun m. fl., 2005).

Positionen av roboten kan beräknas också med att räkna distansen till landmärken och distansen mellan landmärken som extraheras ur bilder, alltså uppehålla en korrelationsmatris om landmärken och deras förhållande från robotens position (Durrant-Whyte och Bailey, 2006; Thrun m. fl., 2005). Då man observerat stora mängder av landmärken så kommer beräkningsbehovet att växa kvadratisk enligt observerade landmärken.

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. Global och lokal lokalisering är då täthetsfunktionen av den första förtroende om robotens position i miljön, som till exempel i Bayes filtern vore  $bel(x_0)$  (Thrun m. fl., 2005).

## 2.4 Kartläggning av omgivningen

Med kartläggning av omgivningen anser vi att roboten kartlägger sin omgivning med hjälp av data som den har tillgång till (Thrun m. fl., 2005). Detta kan vara till exempel rörelseinformation eller sensorinformation samt kombinationer av dessa. Kartläggning är ett problem som är beroende av lokalisering. Då man har en uppfattning av robotens läge och åtkomst till rörelseinformation är kartläggning lättare än då robotens position i början är okänd. För att konstruera en karta så måste man ta i beaktande störningar som uppstår av rörelse- och sensorinformation.

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 Grid 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 sparad i ett rutsystem där rutor som är upptagna är någon objekt i miljön (Heng m. fl., 2011; Desouza och Kak, 2002). Att konstruera en 3D representation av omgivningen kräver mer beräkningskapacitet än 2D kartor,

för att man tar i beaktande fler dimensioner (Thrun m. fl., 2005). För robotar som navigerar på stadig grund är oftast en 2D-representation tillräckligt, men för drönaren behöver man kartlägga i 3D.

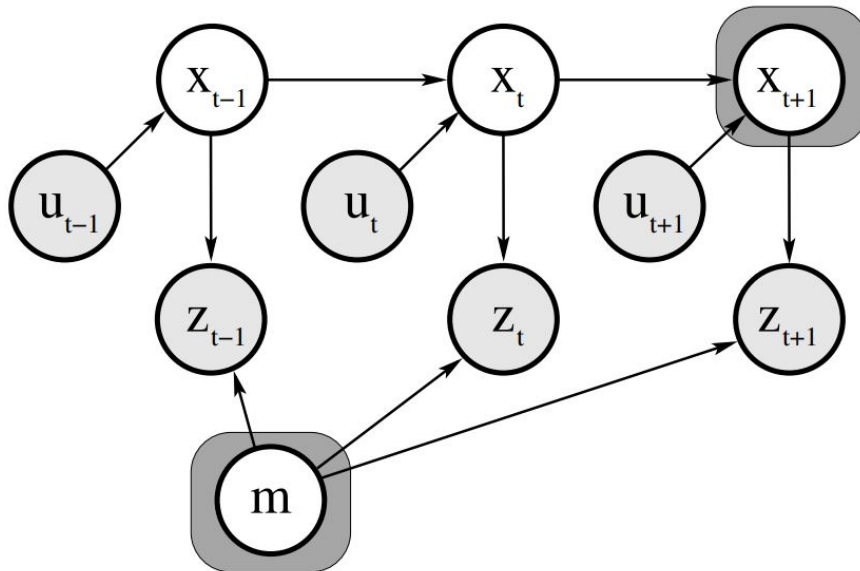
Kartan kan vara färdigt sparad för en drönare eller så kan miljön kartläggas från bilderna av sensorerna då den flyger (Lu m. fl., 2018). Med 3D volymetriska sensorer kan man konstruera en 3D modell och spara denna information i en Octree-struktur. Med strukturen kan data om miljön packas i mindre format 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. Viktigt i kartläggning är att kartan kan lätt uppdateras då man gör nya observationer (Se, D. G. Lowe m. fl., 2005).

Oftast så har robotar inte en färdig karta som de kan använda för att navigera och för att åstadkomma autonoma robotar så måste de kunna själv kartlägga sin omgivning (Thrun m. fl., 2005). Kartläggning för robotar är ett problem som är svårt utan lokalisering, därför refereras kartläggning för robotar som samtidig lokalisering och kartläggning.

### 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  $t$ , där man tar i beaktande riktningen och positionen av roboten  $x_t$ , distansen roboten rör på sig  $u_t$ , kartan  $m$  och observationer för varje landmärke som roboten gör  $z_t$ , se figur 3.1. Att beräkna robotens läge med sannolikhetsberäkning så är formeln  $P(x_t, m | z_{1:t}, u_{1:t})$ .

SLAM problemet kan delas i två delar som är Fullständig SLAM och Online SLAM (Thrun m. fl., 2005). Skillnaden med dessa är hur man beräknar betingade aposteriorsannolikheten för  $x_t$ . I fullständig SLAM beräknas betingade sannolikheten från hela robotens positionskedja  $x_{1:t}$ , det vill säga  $P(x_{1:t}, m | z_{1:t}, u_{1:t})$ , medan i online SLAM använder man bara senaste läge  $x_{t-1}$ , alltså Markovegenskapen. Några lösningar för SLAM som baserar sig på sannolikhetsberäkning är Extended Kalman Filter (EKF-SLAM), som är en Online SLAM-lösning, och FastSLAM, som är en fullständig SLAM-lösning (Durrant-Whyte och Bailey, 2006; Thrun m. fl., 2005).



**Figur 3.1:** Bild på SLAM-problemet (Thrun m. fl., 2005).

SLAM-problemet har delproblem som borde lösas för att få robotar att navigera autonomiskt (Aulinas m. fl., 2008). Ett stort problem i flesta visionsbaserade 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 missvisningar 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 baserat på data som man har i början av navigeringen. Kategorierna är kartlösa (mapless), kartbaserade (map-based) och kartbyggande (map-building) system.

## 3.1 Kartlösa system

Kartlösa system är oftast den realistiska situationen då man navigerar och är den situationen som träffar bäst SLAM definitionen (Thrun m. fl., 2005). 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

Optiskt flöde hänvisar till uppfattad relativ rörelse i sekventiella bilder mellan observatören och till exempel någon fästpunkt i miljön (Kehoe m. fl., 2006). Rörelse uppstår i sekventiella bilder till exempel då man vänder kameran eller om något rör på sig då kameran är stilla.

Santos-Victor m. fl., 1993 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 m. fl., 1993 använde för roboten fungerar bara om båda kamerorna är symmetriskt riktade från varandra när man tar i beaktande rörelseriktningen av roboten. Denna teknik fungerar dåligt i struk-

turlösa miljö och det går inte att byta rörelseriktningen då man navigerar på detta sätt i korridorer.

Fastän metoden som Santos-Victor m. fl., 1993 använde tog bara i beaktande horisontala flödesvektorer så har sedan detta användning av optiska flödesmetoder använts i drönare (Chao m. fl., 2013). Nuförtiden kan drönare uppskatta avstånd, hålla sin höjd, undvika hinder, beräkna hastighet och landa på en plattform som rör på sig med hjälp av optiskt flöde.

### 3.1.2 Spårning av egenskaper

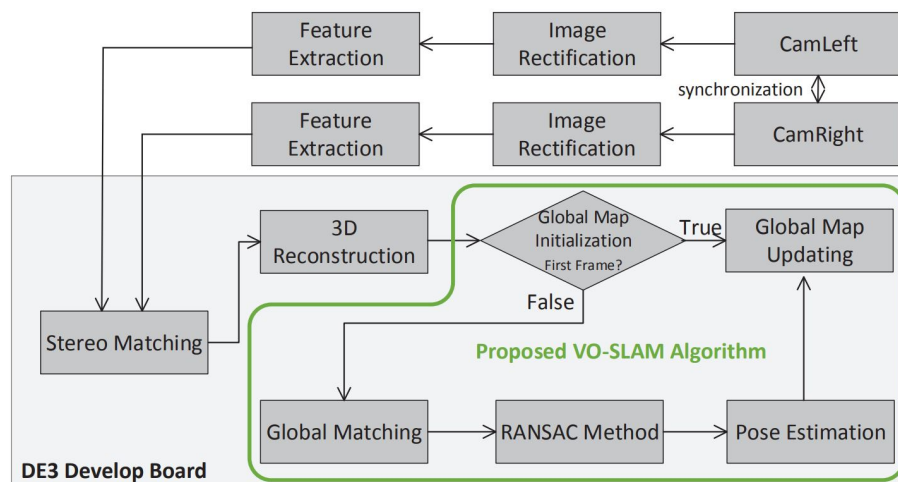
Spårning av egenskaper (Feature tracking) används för att skaffa information om objekt, så som linjer, hörn och olika former som är invariant för rörelse (Lu m. fl., 2018). Med hjälp av dessa egenskaper av objekten och deras relativa rörelse i sekventiella bilder kan man bestämma robotens position. Då 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 kategori (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 kvadratisk av antalet egenskaper och FastSLAM nackdel är bildbearbetning.

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 färre 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 parallellt.

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 byggt och programmerat för matrisberäkning, beräknar vektorvinklarna för varje observerade landmärke från globala kartan och hittar de bästa träffar för lokaliseringsalgoritmen. Före lokaliseringssuppfattningen använder de RANSAC (Random Sample Consensus) för att ta bort avvikande träffar inom landmärken, se figur 3.2.

Med hjälp av FPGA-kretsen har de fått dataförening med matriser att beräknas i realtid med upp till 31 bilder i sekund och litet energibehov. Enligt Gu m. fl., 2015 kan VO-SLAM estimerar positionen med 1–2 cm exakthet med upp till 30 000 landmärken i kartan, hantera





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

loopstängning och enligt dem är troligtvis en av de mest energieffektiva lösningen (Gu m. fl., 2015). Något som deras lösning tar inte i beaktande är SIFT algoritmens tidskrav. Latif och Saddik, 2019 använder VO-SLAM i en drönare 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 vetskaper 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 provar den hitta till exempel landmärken från bilder som är lika till de landmärken som roboten vet om. När den förenat landmärken rätt så kan roboten estimerar sin position i miljön. Med kartbaserad navigering kan en drönare planera sin rutt i förhand och beräkna omvägar under navigeringen då 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 kartan

- estimerar positionen av roboten

Det svåraste steget av dessa är att matcha observationerna med kartan som den har. Detta kallas för dataföreningsproblemet. Roboten kan inte med full säkerhet veta sin position och då är det svårt att matcha landmärken med kartan (Desouza och Kak, 2002).

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. Globala lokaliseringsproblemet kan lösas till exempel med Monte Carlo lokalisation eller Markov lokalisation, som båda är en variant av Bayes filter som presenterades i kapitel Lokalisering i omgivningen (Thrun m. fl., 2005).

Monte Carlo lokalisation fungerar så att en robot antar med lika sannolikhet sin position i kartan vid början av navigeringen (Dellaert m. fl., 1999). Approximerande i Monte Carlo fungerar så att roboten slumpmässigt väljer platser vart den möjligen skulle finnas med den rörelseorder som den får och sedan rör roboten på sig, observerar landmärken, förenar dessa med kartan som den har och på basis av denna information kan den estimerar vilken av dessa slumpmässiga val den gjorde är den sannolika nya position i miljön.

Markov lokalisationsalgoritmen klarar också av sig globala lokaliseringsproblemet då man har en karta (Thrun m. fl., 2005). Istället för att anta bara ett förtroende om robotens position i kartan så uppehåller den en sannolikhetsfördelning över hela kartan. Som exempel om roboten börjar navigera så är sannolikhetsfördelningen enhetlig och då den rör på sig och om den observerar till exempel en dörr så ökar sannolikheten av robotens position för varje plats det finns en liknande dörr i kartan. När den rör på sig vidare så kommer den troligtvis att observera mera dörrar och på basis av rörelseinformationen och observerade landmärken kan den vid något skede vara ganska säker av sin position.

I lokal lokalisering där roboten har en vetskap om sin position och riktning i början av navigering behöver lokaliseringsalgoritmen hålla koll på rörelseinformationen som roboten utför (Dellaert m. fl., 1999; Thrun m. fl., 2005). Då roboten rör på sig så minskar sannolikheten av robotens position på basis om hur lokaliseringsalgoritmen tar i beaktande felmarginal i rörelsen. Då osäkerhet av sin egen position är för stor så använder den observerade landmärken för att öka på förtroendet av sin position.

## 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). Kartor kan byggas i två eller tre dimensioner (Thrun m. fl., 2005). Fördelar med att bygga en karta i 3D och använda denna för lokalisering är att det uppkommer mindre missvisningar då roboten lokaliserar sig själv. Som exempel i 2D-kartor, om en robot navigerar i en korridor och kommer till slutet av korridoren kan det vara svårt för roboten att anta i vilken ända den är om korridoren är symmetrisk. Med 3D så finns det mera data att använda för att lokalisera men detta betyder att man behöver mera beräkningskapacitet. 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 använt 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 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 då man lokaliserar roboten.

Denna metod att konstruera kartor och lokalisera roboten har ändå problem då det kommer till loopstängning och uppdatering av kartan (Se, D. G. Lowe m. fl., 2005). Med att använda samma bildbearbetningsmetoder och med att kartlägga delar av områden och senare bygga en stor global karta fick forskarna loopstängning och kart-uppdateringen löst.

Globala kartan uppbyggdes så att de identifiera identiska landmärken från dessa små kartor och kunde från dessa foga ihop kartor som var bredvid varandra till en stor karta. Indirekta metoder fungerar dåligt i då det finns få egenskaper att extrahera ur bilder, såsom i strukturlösa miljön (Lu m. fl., 2018).

### 3.3.2 Direkta metoder

Direkta metoder fungerar bättre i strukturlösa miljön än indirekta metoder (Engel m. fl., 2014). Som i indirekta metoder där man provar 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). Kartan kan sedan användas för att

## 4 Sammafattning

Samtidig lokalisering och kartläggning i dynamiska miljön är ett problem som måste lösas för att drönare skulle kunna navigera autonomiskt. Oftast är robotens omgivning oändlig och då man estimerar positionen av roboten måste man approximera. Då man approximerar förtroende av robotens position är sällan fullständigt medveten om sin position.

För att verkligen ha autonoma drönaren som skulle använda sig av visionsbaserad navigering måste man kunna kombinera effektiva bildbearbetningsalgoritmer, lokaliseringsalgoritmer och kartläggningsalgoritmer så att dessa algoritmer skulle fungera samtidigt, i realtid samt ge bra uppskattningar av drönarens läge.

En drönare har begränsningar då det kommer till beräkningsbehov på grund av att kommersiella drönare byggs mindre än förr och har begränsad batterikapacitet. Drönare har inte tillgång till samma rörelseinformation som robotar på stadig grund som oftast har inbyggda tröghets- och hastighetsmätare i sig. Därför är det viktigt att det finns forskning inom beräkning av rörelseinformationen från kamerabilder.

Denna avhandling var en ytlig översikt om SLAM problemet och vad är basis för de SLAM-lösningar som finns idag. De problem som finns idag är att konstruera pålitliga algoritmer i visionsbaserad navigering som kan behandla olika scenario, till exempel att estimerar djuphet för att bygga 3D-kartor, positionen av roboten, kartläggning och uppdatering av kartan, dataförening, loopstängning samt ruttplanering och undvika hinder under navigeringen som avhandlingen har inte haft fokus på.

# 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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1109/IROS.2011.6095058).
- Kehoe, J., Watkins, A., Causey, R. och Lind, R. (aug. 2006). "State Estimation Using Optical Flow from Parallax-Weighted Feature Tracking". I: DOI: [10.2514/6.2006-6721](https://doi.org/10.2514/6.2006-6721). URL: [https://mae.ufl.edu/ricklind/rick\\_pro/of/GNC06\\_OF.pdf](https://mae.ufl.edu/ricklind/rick_pro/of/GNC06_OF.pdf).

- 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1109/MCOM.2017.1600587CM).
- Rogers, B. och Graham, M. (febr. 1979). "Motion parallax as an independent cue for depth perception". I: *Perception* 8, s. 125–34. DOI: [10.1068/p080125](https://doi.org/10.1068/p080125).
- 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](https://doi.org/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](https://doi.org/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/TR0.2004.839228](https://doi.org/10.1109/TR0.2004.839228).
- 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](https://doi.org/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.