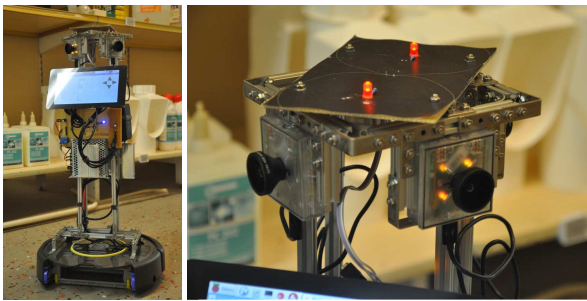


# Kartläggning av punktmoln i detaljhandel

Vi har konstruerat en liten robot på hjul som kan köra runt i matbutiker och kartlägga positionen av prislappar. Roboten använder sig av kameror för att triangulera fram en 3D position för prislapparna som den har sett från flera vinklar. Roboten skapar automatiskt en karta över sin omgivning genom att optimera sin egen och omgivningens position för att minimera skillnaden mellan den matematiskt förväntade bilden och den bild som kamerorna faktiskt har sett.



Figur 1. Roboten som byggdes i projektet

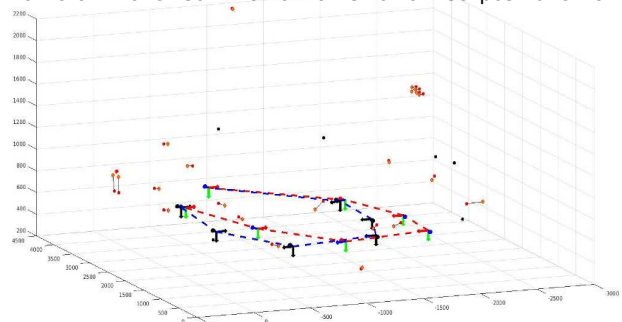
Roboten som är baserad på en dammsugarrobot är utrustad med 4 kameror. Kamerorna används för att hitta digitala prislappar som sitter på hyllkanterna i affären. Roboten är också utrustad med sensorer på hjulen som mäter hur långt den har kört och i vilken riktning. På toppen av roboten sitter två lysdioder. Dessa lysdioder blinkar i olika frekvens vilket gör det möjligt för en takmonterad kamera att se vad som är fram och bak på roboten. Robotens hjärna är en Raspberry Pi 3. Det är en liten dator som kan liknas vid den PC du antagligen har på bordet där hemma. Skillnaden är att denna datorn enbart är 85x56 millimeter stor! Operativsystem är Linux. Det sitter också en 7" touchskärm på roboten som man kan använda för att interagera med den. Detta behövs dock inte så ofta eftersom den går att styra via trådlöst nätverk, WiFi, från en annan dator. Roboten har batterier som klarar många timmars drift.

På hyllkanterna i butiken används digitala prislappar från Pricer AB. Dessa kan du se i många stora matbutiker. Dessa prislappar har en lysdiod som kan blinka om man skickar en signal till dem via trådlös kommunikation. Det är denna lysdiod som vi har använt för att detektera

**"Med en karta som denna kan du som konsument använda en app som talar om hur du hittar till den vara du letar efter"**

prislapparna i bilderna från kamerorna. En 3D karta som visar linjer för robotens uppskattade färdspår samt punkter för prislapparnas uppskattade position syns i bilden nedan. Varje orange prick är den position som roboten har gissat för en prislapp baserat på vad den har sett i kamerorna. De orangea prickarna är förbundna med varsin röd prick som visar den faktiska positionen av en prislapp, denna position är manuellt uppmätt. Som synes är felet enbart ett fåtal centimeter i denna kartläggning. Har du någon gång gått i en mataffär och letat efter en produkt som du inte hittar? Med en karta som denna kan du som konsument använda en app på telefonen som talar om hur du lättast hittar till den vara du letar efter. Genom att kartlägga prislapparna i en mataffär kan de som planerar butikens utseende få grafisk återkoppling. En upplösning på decimeternivå är då önskvärd och det har vi visat att man kan uppnå med ett sådant här system. Eftersom hårdvaran för digitala prislappar och trådlös kommunikation redan finns monterat i de flesta större mataffärer behöver ett system som detta inte bli dyrt.

Delarna på roboten behöver inte vara precisionsdelar utan vanligt förekommande hårdvara räcker. I examensarbetet har inkluderats att undersöka om kartläggningen kan göras utan annan fast monterad hårdvara. Ett exempel på det är kameror i taket som har använts för att se positionen av



Figur 2. En karta i 3D skapas av robotens vy



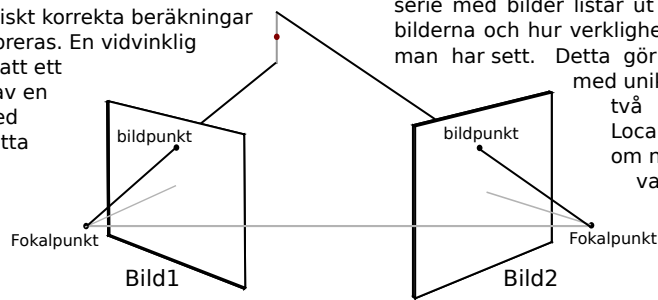
Figur 3. Man kan kalibrera en kamera så att raka linjer i verkligheten blir raka på bilden

roboten. Denna positionsmätning användes för att förbättra resultaten i kartläggningen. Med de algoritmer som utvecklades, och det 10m<sup>2</sup> rum som testen utfördes i, visade det sig dock att nästan lika bra resultat gick att uppnå genom att bara använda den position som var uträknad från hjulets rotation, så kallad dödräkning. Det utvecklade systemet med takkamerorna kan däremot mycket väl visa sig användbart i en större miljö. Med en längre färdsträcka ökar osäkerheten i dödräkning tydligt.

För att kunna utföra geometriska korrekta beräkningar i en bild måste kameran kalibreras. En vidvinklig kamera förvränger bilden så att ett fyrkantigt objekt får formen av en typisk tunna, det vill säga med utsvängda konvexa kanter, titta t.ex. på hörnan mellan vägg och tak i bakgrunden på den vänstra delen av Figur 3. I den högra delen visas bilden efter att den har blivit kalibrerad.

Kalibreringen sker genom att man tar foto av t.ex. ett schackbräde. Med datorprogrammet som vi har gjort hittas automatiskt hörnorna i schackbrädet och programmet skapar sen en matematisk transform som får schackbrädet att bli fyrkantigt. Denna transform kan man sen tillämpa på alla bilder för att göra samma kompensering. Efter denna kalibrering kan man räkna på bilderna som om de kommer ifrån en nålhålskamera utan lins, precis som den kamera Erasmus Reinhold i Nürnberg använde redan år 1558 för att iaktta en solförmörkelse. Bilder från denna typen av kameror, som inte har någon lins, är väldigt lätta att räkna på med enkel geometri. Därför är det mycket vanligt att man i bildanalys gör denna kalibrering. Kalibreringen är unik för varje kamera eftersom den dels beror på linsen som används men även på den exakta monteringen. Om linsen sitter lite snett så blir bilden förvrängd på ett annat sätt.

Genom att dra en linje från speglingen av kamerans fokuspunkt bakom bilden och genom prislappens position i bildplanet får man en oändligt lång linje som prislappen bör ligga nära. Har man två bilder från olika vinklar med samma prislapp kan man beräkna prislappens position



Figur 4. Med enkel geometri kan en punkt placeras i 3D om den syns i två bilder från olika håll

som den punkt där dessa två linjer korsas. I praktiken har man alltid störningar och brus i bilderna, därav kommer de två linjerna inte att korsa varandra. Man kan då använda sig av olika metoder för att räkna ut den mest sannolika positionen mellan dessa två linjer. Mycket forskning går ut på att hitta den bästa algoritmen för att gissa punktens position. Kanske den enklaste metoden är att gissa på punkten mittemellan linjerna. Detta har dock visat sig vara en ganska dålig metod och en bättre uppskattning kan hittas genom att använda sig av sannolikhetslära.

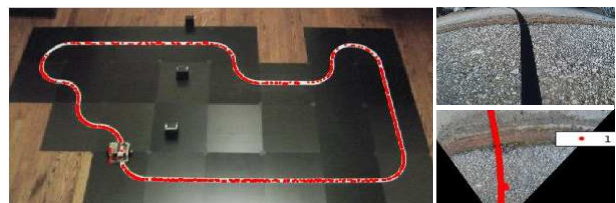
## Vad är SfM / SLAM?

Structure from Motion  
Simultaneous Localization And Mapping

SfM och SLAM är termer som man ofta hör när man pratar om robotar, kartor och datorseende. De två termerna är snarlika när man kommer till bakomliggande matematik men det man vill uppnå är olika. I "Structure from Motion" vill man göra en tredimensionell modell av verkligheten som en kamera har sett. Detta innebär att man utifrån en serie med bilder listar ut hur kameran har rört sig mellan bilderna och hur verkligheten ser ut för att skapa de bilder man har sett. Detta gör man genom att leta efter särdrag med unika utseenden som man kan hitta i två eller fler bilder. "Simultaneous Localization And Mapping" pratar man om när man skall göra en karta utifrån vad en robot ser. Om roboten inte från början vet var den är så ritar man en karta samtidigt som man gissar sin position i kartan, därav namnet. SfM baseras normal på bilder från en eller flera kameror medan SLAM kan använda sig av en mängd olika sensorer beroende på vad applikationen kräver.

## Linjeföljning med kamera

Många robotar, speciellt i industriella miljöer, navigerar genom att följa en vit eller svart linje på golvet. I detta projektet utvecklades en ny metod för att hitta dessa linjer med hjälp av en kamera. Metoden utvecklades som en del av en analys för hur roboten skall navigera när den kör runt i butikerna. En metod som föreslogs var just linjeföljning. I Figur 5 kan man se resultatet av att tillämpa algoritmen på



Figur 5. Automatisk detektering av linjer i bilder

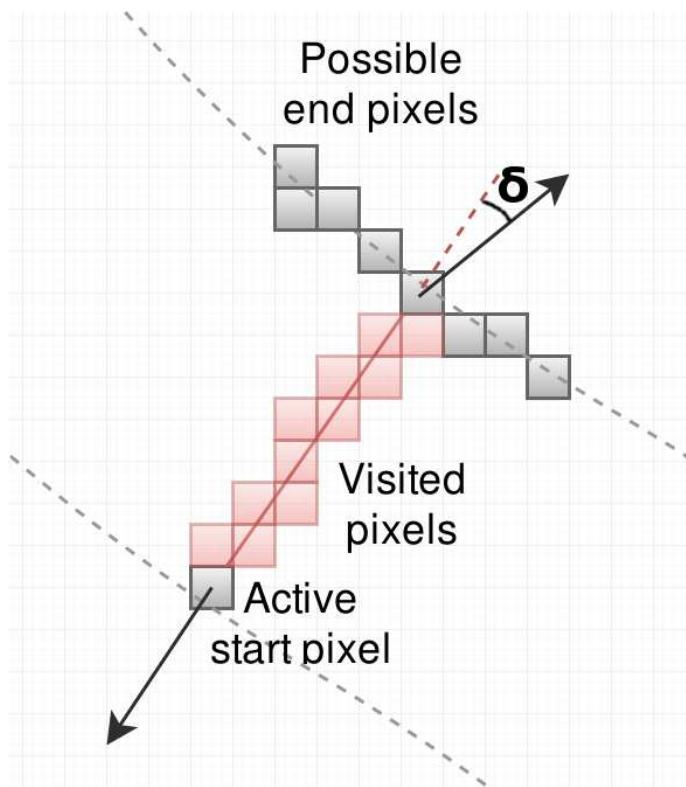
två olika bilder med linjer. I den vänstra delen av bilden är en vit linje som har markerats med röda prickar på de punkter som algoritmen tror att en linje finns. Till höger syns överst en originalbild från kameran på en robot. Under den är en modifierad bild där algoritmen först gjort en perspektivtransformation för att få en bild som om den sågs ovanifrån. Även här markerar de röda prickarna var algoritmen tror att linjen är. Kanten på en linje hittar den genom att titta på skillnaden mellan två pixlar, om skillnaden är stor så är det möjligt att det är en kant på en linje. För att bekräfta om det är en linje eller ej följer den längst med gradienten av varje kantpixel, det vill säga vinkelrätt mot linjens kant, för att se om den kan hitta en annan kant. Om den andra kanten har en motåtriktad gradient gissar den att det är en linje. Dessutom undersöker den om den eventuella linjen har en konstant bredd. Om linjen är taggig och ojämn sorteras den bort. Resultatet som har uppnåtts står sig väl och överträffar många andra publicerade metoder.

Metoden för linjeföljningen har sammanfattats i en akademisk artikel som har lämnats in för publikation:

*Daniel Falk: Cognitive vision for line following using stroke width vectorization, 2016.*

Hela examensarbetet finns presenterat i detalj i en rapport:

*Daniel Falk: Minimum Hardware SfM/SLAM for Sparse Data Point Mapping of Retail Stores (2016:E43), ISSN 1404-6342*



Figur 6. Algoritmen börjar på en pixel längst kanten av linjen och följer sedan gradienten tills andra sidan nås