

AARHUS TECH
SRP

21. december 2017

Monte Carlo Lokalisering

Forfatter

Jacob Emil Ulvedal Rosborg

Vejleder

Mikkel Stouby Petersen

Jørn Sanggaard

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Indhold

1. Indledning	3
1.1. Opgaveformulering	3
1.2. Afgrænsning	3
2. Stokastiske Variable	5
2.1. Udfaldsrum og delmængder	5
2.2. Sandsynlighedsfunktion	6
2.3. Sandsynlighedsfelt	6
2.4. Stokastiske variable	7
3. Monte Carlo Metoden	8
3.1. Plat eller Krone	8
3.2. Numerisk integration	9
4. Monte Carlo Lokalisering	12
4.1. Opbygning	12
4.2. implementering	13
5. Diskussion	14
6. Konklusion og perspektivering	15
A. Test	17

1. Indledning

Nogle problemer er så komplicerede at de næsten er umulige at analysere, modellere og løse algebraisk, f.eks den kaotiske og tilfældige proces der finder sted når man spalter uran-235 (neutron diffusion). Dette problem stod forskerne overfor i 1940 under udviklingen af atombomben i Manhattan projektet. Her blev Monte Carlo Simulering anvendt til at simulere neutroners vandring og dette blev brugt til at vurdere de optimale fysiske forhold for den kæde reaktion der skulle få bomben til at sprænge [2].

Begrebet Monte Carlo Metoden dækker egentlig over en række metoder der kan bruges til at analyse problemer der ikke fremstår løsbare, så som neutron diffusion. For eksempel findes der metoder så som Monte Carlo Simulering, Lokalisering og ud over disse er der yderligere metoder inden for Finans og Medicin. Denne opgave vil fokusere på numerisk integration og lokalisering. Jeg vil yderligere komme ind på Stokastiske Variabler og deres egenskaber, samt anvendelse og implementering af Monte Carlo Lokalisering hvilket er en form for partikelfilter i en autonom robot.

Monte Carlo Metoden blev først rigtigt anvendt da computere blev så udviklede at de kunne udføre disse simuleringer, også kaldet eksperimenter, for os. Da disse simuleringer kræver et forholdsvis stort antal gentagelser, et antal der både ville have krævet arbejdskraft og tid af umådelige proportioner. Alt dette for at opnå et resultat der ikke altid ville være brugbart. Dette resultat vil nemlig altid være en approximering hvorimod en algebraisk tilgang ville udlede værdien og derved være eksakt. Til gengæld er man med metoden i stand til at tackle problemer der så komplekse i sin natur at det ikke er praktisk muligt at udlede disse problemer algebraisk [3]

1.1. Opgaveformulering

- (I) Redegør for, hvad Monte Carlo-algoritmer er, og giv eksempler både praktiske og teoretiske anvendelser. For eksempel i forbindelse med numerisk integration.
- (II) Forklar centrale egenskaber ved stokastiske variable i det omfang det er nødvendigt for at forstå algoritmernes virkemåde.
- (III) Vis, hvordan Monte Carlo-algoritmen kan anvendes til lokalisering af robotter. Kom herunder ind på, hvordan algoritmen kan implementeres.
- (IV) Diskuter Monte Carlo-metodens muligheder og begrænsninger i forbindelse med anvendelse i en konkret autonom robot.

1.2. Afgrænsning

Denne opgave henvender sig til studerende på 3. årgang på en gymnasial uddannelse. For at læse opgaven kræves der ikke en dybdegående forståelse for hverken statistik eller

algoritmer. Opgaven vil forklare de begreber der er nødvendige for at forstå Monte Carlo Metoden, men kun på et redegørende niveau.

2. Stokastiske Variable

En væsentlig del af Monte Carlo Metodens teori og praksis bygger på Stokastiske Variable. For at forstå Stokastiske Variable kræves der en forståelse for nogle simple begreber indenfor statistik. Disse begreber er udfaldsrum, delmængde, sandsynlighedsfunktion samt sandsynlighedsfelt. De nævnte begreber er også essentielle inden for statistik generelt.

2.1. Udfaldsrum og delmængder

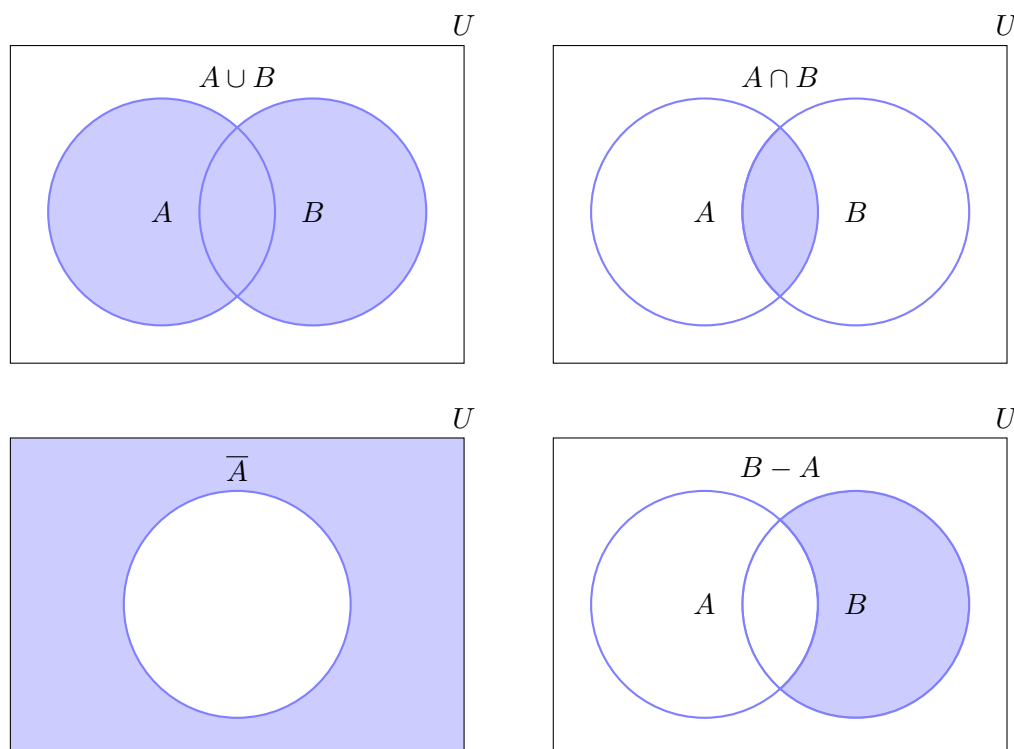
Der findes to typer af udfaldsrum, henholdsvis et diskret og kontinuert udfaldsrum. Forskellen på disse to er at der i et diskret udfaldsrum kun er et givent antal mulige udfald. Et eksempel på et diskret udfaldsrum kunne være en sekssidet terning. Denne sekssidet terning vil have et udfaldsrum U der indeholder elementerne $1, 2, 3, 4, 5, 6, 7$ dette kan skrives som $U = \{1, 2, 3, 4, 5, 6\}$. Altså vi har tale om et udfaldsrum med længden seks, hvilket skrives $|U| = 6$ eller som n . Hvorimod et kontinuert udfaldsrum kunne være højden på elever i en klasse, udfaldsrummet kunne altså være defineret som $U = [150cm; 210cm]$. Udfaldsrummet er kontinuert da elevers højde kan variere med et infinitesimal, altså elev a kunne være en uendeligt lille mængde højere end elev b . I sådan et udfaldsrum findes der ikke nogen længde af udfaldsrummet, altså $|U| \notin \mathbb{R}$. Derfor kaldes det også et endeligt og et ikke-endeligt udfaldsrum [5].

En delmængde er en mængde af udfaldsrummet, det beskrives ofte med A eller B . Hvis vi tager udgangspunkt i den sekssidet terning igen, kunne en delmængde af dets udfaldsrum være $A = \{1, 3\}$. Altså en delmængde indeholder altså en eller flere elementer fra udfaldsrummet.

Ydermere vil et bestemt element i et udfaldsrum eller delmængde denoteres u_i eller a_i , altså for henholdsvis A og U . Det vil sige hvis vi har en mængde Q vil et bestemt element i Q denoteres som q_i . Desuden findes der en række operatorer der beskriver delmængders relationer til hinanden og udfaldsrummet.

1. $A \cup B$: udfaldet ligger i enten A eller B, evt. i både A og B
2. $A \cap B$: udfaldet ligger i både A og B
3. $A \setminus B$: udfaldet ligger i A og ikke B
4. A^c : udfaldet ligger ikke i A (dette kan også denoteres \bar{A})

Betegnelserne er de samme, som vi bruger i mængdelæren, og vi taler derfor om foreningsmængden $A \cup B$, fællesmængden $A \cap B$, mængdedifferensen $A \setminus B$ samt komplementmængden A^c [5]. Nedenstående figurer også kaldet venn-diagrammer, illustreres overstående mængder og deres operatorer.



2.2. Sandsynlighedsfunktion

Sandsynlighedsfunktion denoteres P og beskriver sandsynligheden for et element i et udfaldsrummet U . For eksempel kunne $P(\{1, 2\}) = 0,8$ og $P(3) = 0,2$ hvor udfaldsrummet $U = \{1, 2, 3\}$. Dette betyder at sandsynligheden for 1 eller 2 er 80% hvorimod sandsynligheden for 3 er 20%. Desuden kan vi bruge additions loven til at udlede den totale sandsynlighed for udfaldsrummet [5].

$$P(1) + P(2) + P(3) = 1 \quad (2.1)$$

$$P(\{1, 2\}) + P(3) = 1 \quad (2.2)$$

Desuden kan $\{1, 2\}$ også beskrives som en delmængde af U på formen $A = \{1, 2\}$, altså bliver det desuden gældende at $P(A) = P(1) + P(2)$

2.3. Sandsynlighedsfelt

Et sandsynlighedsfelt findes på to former, det endelige - også kaldet det diskrete sandsynlighedsfelt - samt det kontinuerte sandsynlighedsfelt også kaldet et ikke-endeligt sandsynlighedsfelt. Et sandsynlighedsfelt kan denoteres (U, P) og er bestående af et udfaldsrum U og en sandsynlighedsfunktion P . Hvis der er tale om et symmetrisk sandsynlighedsfelt, vil følgende være gældende for P .

$$P(U) = 1 \quad (2.3)$$

$$P(u_1) + P(u_i) + \dots + P(u_n) = 1 \quad (2.4)$$

$$P(u_1) = P(u_i) = \dots = P(u_n) = \frac{1}{|U|} \quad (2.5)$$

De to udsagn udtrykker derved at sandsynligheden for $P(U_i)$ er i intervallet $[0; 1]$ samt at den samlede sandsynlighed for feltet er 1, hvor $1 = 100\%$. I et ikke-symmetrisk sandsynlighedsfelt vil de to første regler af de overstående tre gælde, men sandsynligheden for de enkelte elementer i udfaldsrummet er ikke nødvendigvis lig, altså kan $P(u_1) \neq P(u_2)$ [5].

2.4. Stokastiske variabler

Stokastiske variabler findes af to typer der begge denoteres med X , lad desuden x_i være et konkret udfald, disse to typer er henholdsvis diskret og kontinuert. En stokastiske variabel er en variable som kan tage alle værdier i et givent udfaldsrum, med sandsynligheden P , altså $P(X = u_i) = P(u_i)$. Forskellen mellem diskret og kontinuert stokastiske variabler er deres udfaldsrum og deres tilhørende regneregler. Et eksempel på en stokastiske variabel kunne være Europæisk Roulette, dette spil har et symmetrisk sandsynlighedsfelt med udfaldsrummet $U = \{0, 1, \dots, 36\}$ der har indeholder $37 = |U|$ udfald. Det vil altså sige at den kugle man smider ned i spillet kan beskrives som en stokastiske variabel X og den har en lige stor sandsynlighed for at tage et udfald i udfaldsrummet, da der er tale om et symmetrisk sandsynlighedsfelt. Altså $P(u_1) = P(u_i) = \dots = P(u_n)$, det betyder at $P(X = 1) = P(X = 4)$. Ydermere er roulettens udfaldsrum opdelt i tre farver grøn, rød og sort, disse kan betegnes som delmængderne henholdsvis G , R og S . Vi kan således beskrive sandsynligheden for at den stokastiske variabel X vil ligge i en delmængde således $P(X \in S) = P(X \in R) = \frac{18}{37}$ og $P(X \in G) = \frac{1}{37}$. Delmængden G kunne også have været defineret som $G = (R \cup S)^c$, altså de udfald som ikke falder ind under hverken R eller S , eller $G = 0$ [1]

Lad desuden $\mathbb{E}[X]$ være den forventede værdi af X , den forventede værdi skal forstås næsten som et gennemsnit. Det overstående eksempel, her ville den forventede værdi X være $\mathbb{E}[X] = 18$, dette kan udregnes på følgende måde:

$$\mathbb{E}[X] = \sum_{i=1}^n \frac{u_i}{n} \quad (2.6)$$

Dette er kun tilfældet da det er et symmetrisk sandsynlighedsfelt, hvis sandsynligheden for de forskellige udfald ikke er lig, altså $P(u_1) \neq P(u_2)$. Her ville man bruge følgende formel for den forventede værdi af X .

$$\mathbb{E}[X] = \sum_{i=1}^n u_i \times P(u_i) \quad (2.7)$$

Stokastiske Variabler på en computer er ofte repræsenteret som et pseudo-genereret tilfældigt tal, hvilket er et tilfældigt tal udvalgt af computeren. Disse tilfældigt genererede tal bliver brugt til at udføre eksperimenter og simulering af Monte Carlo Metoden [3]. Dette vil komme til udtryk i det følgende afsnit, hvori det vil blive vist hvordan Monte Carlo Simulering gør brug af Stokastiske Variabler.

3. Monte Carlo Metoden

I dette afsnit vil opgaven komme ind på...

Monte Carlo Metoden har sit navn efter Stanislaw Ulams onkel, som ofte spillede på Casino Monte Carlo i Monaco. Metoden referer herved til de sandsynligheder og de tilfældige udkom der kan finde sted i sådanne spil. Ulam var en af de forskere der arbejdede på Monte Carlo Metoden under Manhattan Projektet, som blev iværksat i 1942 [6]. Metoden blev udviklet for at kunne simulere neutroners diffusion. Det var nødvendigt at udvikle og benytte denne metode, da problemet var for komplekst til at kunne blive afløst algebragisk eller blive løst med datidige metoder. Metoden blev først anvendt på gamle analoge computere, hvilket begrænsede kompleksiteten af simulationen [2].

Ydermere er Monte Carlo Metoden også en algoritme, det betyder at det er en process som er udført i nogle logiske trin, trin som beskriver udførelse af metoden. En algoritme kunne f.eks. være trinnene involveret i at sortere et sæt kort. Lad kortspillet være et sæt kort og et element værende ét enkelt spillekort, kortspillet kan således sorteres på følgende måde.

1. Vælg et vilkårligt element e_i fra sættet.
2. Placer elementet bagest i sættet.
3. Sammenlign nu e_i med elementet lige før e_{i-1} hvis $e_i > e_{i-1}$ bytter de plads. Dette gentages indtil $e_i < e_{i-1}$ eller $i = 0$.
4. Gentag denne process indtil kortspillet er sorteret

Denne sorterings algoritme er også kaldet bubblesortering og dennes navn kommer af måden de elementer i sættet bobler til toppen. [4]

3.1. Plat eller Krone

Et eksempel på udførelse af Monte Carlo Simulering kunne være spillet Plat eller Krone, spillets regler er som følgende.

1. Hver spiller satser på en side af mønten.
2. Mønten kastes op i luften.
3. Vinderen er den spiller hvis sats er den side af mønten der vender op af.

Man kan også beskrive Plat og Krone med et symmetrisk sandsynlighedsfelt (U, P) , hvor udfaldsrummet U ville være $U = \{Plat, Krone\}$ og sandsynlighedsfunktionen P hvor følgende er sandt da det er et symmetrisk sandsynlighedsfelt.

$$P(Plat) = P(Krone) = \frac{1}{2} \quad (3.1)$$

Vi kan desuden beskrive mønten med en stokastiske variable X . Den forventede værdi af X kan ligedes beskrives som værende $\mathbb{E}[X] = \frac{1+2}{2} = 1,5$ hvis $Plat = 1$ og $Krone = 2$ altså den gennemsnitlige værdi af udfaldsrummet.

Vi kan finde frem til en approximering af den forventede værdi af X ved brug af Monte Carlo Simulering. I den sammenhæng lad $\mathbb{E}[X]$ værende den forventede værdi af X . Vi kan derved opskrive følgende udsagn hvor x_i er en konkret simulering af X .

$$\mathbb{E}[X] = \sum_{i=1}^n \frac{x_i}{n} \quad (3.2)$$

Vi kan udfører en simulering der simulere 1000 eksperimenter, hvor X er uafhængig.

```
import random

random.seed(42)

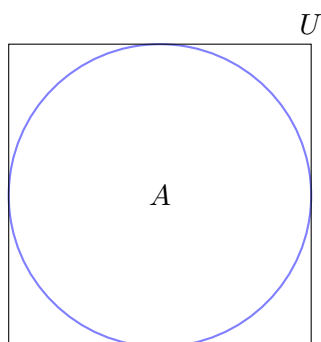
print(sum([random.randint(1,2) for _ in range(1000)]) / 1000.000)
```

Dette vil give os $\mathbb{E}[X] = 1,519$, hvis vi udførte dette eksperiment 100 gange ville vi finde at vores resultat varierer. Dette skyldes at X er tilfældigt bestemt for hver simulering og er uafhængig af forgående simuleringer, altså hvad der ville betragtes som normal opførelse for en mønt [1].

3.2. Numerisk integration

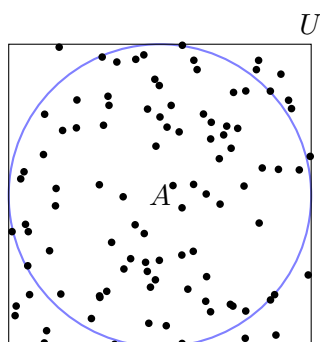
Det overstående eksempel med Plat eller Krone virker måske lidt dumt da vi med nemhed kan udlede det algebraisk, altså tage summen af udfaldsrummet og dele med antallet af mulige udfald. Men et andet tilfælde hvor man kan anvende Monte Carlo Metoden er numerisk integration. I nogle tilfælde er det let at udlede integralet algebraisk, men der findes også tilfælde hvor det er umuligt. Disse funktioners integraler kan udledes ved hjælp af numeriske metoder, her i blandt Monte Carlo Metoden også kaldet Hit-or-Miss. [3]

Lad os betragte enheds cirkel der befnder sig inden for en kvadrat af med dimensionerne 2×2 , denne figurs integral kan let udledes men er valgt for nemheds skyld.



Vi kan starte med at bestemme udfaldsrummet af den overstående figur, dette er U og er alle de punkter der findes i kvadratets plan. Det integral vi ønsker at finde er cirklen, lad

cirklen være delmængde A . Desuden er sandsynligheden P for alle udfald i udfaldsrummet U lige, altså for to given udfald i U er $P(U_1) = P(U_2)$. Altså dette betyder at der er tale om et symmetrisk sandsynlighedsfelt, med et udfaldsrum der er bestående af punkter der findes på kvadratets plan. Den stokastiske variable X vil ligeledes repræsentere et muligt punkt på kvadratets plan. Dette felt er i teorien et kontinuert sandsynlighedsfelt, hvilket også ville gøre vores stokastiske variable til en kontinuert variable. Hvis vi har n stokastiske variable X og n' betegner de stokastiske variable der ligger inden for cirkelens areal, altså således at hvis $X = (0,5; 0,5)$ vil det være et udfald i delmængden A . Dette betyder også alle elementer i delmængden A vil opfylde $a_i x^2 + a_i y^2 \leq 1^2$, hvor a_i er et konkret tilfælde og $a_i x$ er dets x koordinat samt $a_i y$ er dets y koordinat. Nedenstående figur viser et eksempel på n antal konkrete stokastiske variable x_i i udfaldsrummet U hvor n' er de konkrete x_i som befinder sig i A .



Vi kan herved udregne en approksimering af sandsynligheden for $P(A)$.

$$P(A) = \frac{n'}{n} \quad (3.3)$$

Hvis vi ønsker af finde arealet af cirklen skal vi gange med arealet af vores udfaldsrum, da dette var en kvadrat med dimensionerne 2×2 får vi følgende udtryk.

$$Areal(A) = \frac{n'}{n} \times 4 \quad (3.4)$$

En simulering af dette kan udtrykkes i følgende Python kode, hvor vi har 100000 konkrete stokastiske variable

```
import random

random.seed(42)

kvadrattet = [(random.uniform(-1,1),
                 random.uniform(-1,1)) for _ in range(100000)]
cirklen = [p for p in kvadrattet
            if p[0]*p[0] + p[1]*p[1] <= 1]

nk = len(kvadrattet) # lad dette være n
nc = len(cirklen)    # lad dette være n'
a = 2 * 2            # lad dette være arealet af kvadrattet

print("areal = {:.d}/{:.d}*{:.d}={:.f}".format(nc, nk, a, nc/nk*a))
```

Vi vil finde at arealet af cirklen ville være 3,140280 og derved vil $\pi = 3,140280$, da fejlen ved udregningen af et integral på denne måde er $\sqrt{\frac{1}{n}}$ ville vi finde at vi skulle øge antallet af punkter med en faktor 100 for at reducere fejlen med en faktor 10 [3].

4. Monte Carlo Lokalisering

Dette afsnit vil komme ind på hvordan en autonom robot kan implementere en Monte Carlo Algoritme for således at kunne lokalisere sig selv på et kendt kort. Denne Metoder er også kaldet Monte Carlo Lokalisering, og kræver

Hvis vi tager udgangspunkt i en autonom robot, der har fire hjul og mindst to elektriske motorer med enkoder. Elektriske motorer med enkoder afgiver et elektrisk signal n gange per total rotation af motorens akse, dette kunne være implementeret ved brug af en hall sensor der måler på en magnet eller en optisk sensor der måler på et optisk gitter som der følger akslen. Således at robotten er i stand til at udregne hvor langt den har bevæget sig i en given retning. Denne længde kan udregnes, lad i denne udregning d være diameteren af robottens hjul samt n antallet af impulser per rotation samt n_t værende et antal impulser målt efter en given tid. Længden L som robotten har bevæget sig ved antallet målte impulser n_t kan således udregnes på følgende måde:

$$L = d \times \pi \times \frac{n_t}{n} \quad (4.1)$$

Et eksempel på overstående formel, kunne være at robotten havde et sæt hjul med diameteren $5cm$ altså $d = 5cm$ samt en hallsensor der giver $20impulser$ per totale rotation af akslen, når robotten har bevæget sig $4512impulser$ kan vi udregne den længde robotten har bevæget sig:

$$L = 5cm \times \pi \times \frac{4512impulser}{20impulser} = 3543,71cm = 35,4371m \quad (4.2)$$

Vi kan desuden udregne vinklen robotten bevæger sig i, ved at...

Desuden har robotten en sensor der kan måle farven på formen (r, g, b) er hvad der befinder sig under

Med udgangspunkt i overstående robot der implementer et Dead-reckoning algoritme, kan vi lokalisere robotten på et kendt kort ved brug af Monte Carlo Lokalisering.

Man kan anvende Monte Carlo Metoden til at lokalisere, f.eks. en autonom robot, på et kendt kort. Dette kan lade sig gøre ved brug af Monte Carlo Lokalisering, dette er også en form for partikelfilter. Grunden til at Monte Carlo lokalisering er et partikelfilter er grundet i måden hvorpå metoden lokalisere. Lad os sige vi har et sandsynlighedsfelt hvor udfaldsrummet U er alle de mulige positioner. Lad X være en stokastisk variabel i sandsynlighedsfeltet, altså så X er en tilfældig mulig position for robotten på kortet. Vi kan nu lave et eksperiment, hvor X bestemmes til et punkt i sandsynlighedsfeltet, vi kan nu udregne

Monte Carlo Lokalisering består af et sæt af trin som gentages hver gang den autonome robot bevæger sig.

Lad os sige at vi har et kvadratisk område med forskellig farvet fliser, som set på nedenstående figur.

4.1. Opbygning

1. Først bestemmes et sæt af stokastisk variable X , som er punkter på formen (x, y) i udfaldsrummet U med højst koncentration omkring de punkter med lavest afvigelse fra målingen m
2. Robotten laver en måling m på formen (r, g, b) på dens nuværende position
3. Da laves nu en vægtede liste af fejlen mellem m og $rgb(X)$ og punkterne X
4. n antal punkter fjernes

4.2. implementering

5. Diskussion

Muligheder:?

Begrænsning: Praktisk robot på kvadratisk område i forhold til at anvende Monte Carlo Metoden i et åbent rum eller luften?

6. Konklusion og perspektivering

Bibliografi

- [1] Kasper K. Berthelsen. „Note om Monte Carlo metoden“. I: (2014).
- [2] Atomic Heritage Foundation. *Computing and the Manhattan Project*. 2014. URL: <https://www.atomicheritage.org/history/computing-and-manhattan-project>.
- [3] Daniel Kjær. „Sandsynlighedsbaserede metoder [Monte Carlo-metoden]“. I: (2013).
- [4] toptal. *Dubble Sort*.
- [5] Nitschky Schmidt Vestergaard. *Statistik C*. Systime, 2008.
- [6] Povl Lebeck Ølgaard. *Manhattanprojektet*. 2017. URL: <http://denstoredanske.dk/index.php?sideId=121576>.

A. Test