Universitetet i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Informasjonsteknologi - Automatisering og signalbehandling	Vårsemesteret, 2017 Åpen
Forfatter: Øyvind Johnsen Bækkemoen	…… (signatur forfatter)
Fagansvarlig: Ivar Austvoll Veileder(e): Ivar Austvoll	
Tittel på masteroppgaven: Analyse av egenskapsdetektorene SURF, SUSAN og BRISK Engelsk tittel: Analysis of feature detectors SURF, SUSAN and BRISK	
Studiepoeng: 30	
Emneord: Egenskapsdetektorer Bildebehandling SURF, SUSAN og BRISK Robusthet Geometrisk og fotometrisk invarians MATLAB	Sidetall:73 +vedlegg/annet:10 Stavanger, 15.6.2017 dato/år

Analyse av egenskapsdetektorene SURF, SUSAN og BRISK

Masteroppgave vår 2017

Øyvind Johnsen Bækkemoen

15. juni 2017

Sammendrag

Egenskapsdeteksjon er å finne fremtredende punkter i et bilde. Det som definerer hvor god en egenskapsdetektor er, avhenger av hvor robust en detektor er ved forskjellige geometriske- og fotometriske variasjoner. Disse variasjonene kan være lys, rotasjon, støy og skala.

I denne oppgaven blir tre egenskapsdetektorer beskrevet teoretisk og testet mot forskjellige former for variasjoner. De tre utvalgte detektorene er SURF, SUSAN og BRISK. Hvor den første skal lokalisere områder som er lysere eller mørkere enn bakgrunnen. Mens de to siste detektorene lokalisere hjørner. Det blir benyttet to bilder for eksperimentene, et med enkle geometriske figurer og et bilde tatt i sentrum av Stavanger. Bakgrunnen for oppgaven er å teste detektorene opp mot hverandre og se hvem som er best. Eksperimenter hvor egenskapsdetektorer blir testet direkte opp mot hverandre har blitt gjort i mindre grad per dags dato.

SUSAN-detektoren bruker lengst tid av de tre detektorene, men er den detektoren som har lavest antall feildeteksjoner ved normale og lyssterke bilder ved bra terskelverdier. BRISK-detektoren er den raskeste, og gjør det bra ved store deler av eksperimentene foruten feildeteksjonene som går igjen i de fleste eksperimentene. SURF gjør det også bra, avhengig av hva som skal godkjennes om egenskapspunkt og ikke. Fra eksperimentene kommer det frem at detektorene har sine positive og negative sider.

Forord

Med denne masteroppgaven fullfører jeg min 2-årig masterutdannelse i automatisering og signalbehandling, ved Universitetet i Stavanger, våren 2017.

Jeg vil takke veilederen min, Ivar Austvoll, for god veiledning og tilbakemelding under arbeidet med denne masteroppgaven.

Jeg vil også takke kollektivet mitt for god råd og for å holde motivasjonen min oppe.

1 Innhold

Forkortelser

PMR	Portion of Error Registration
CMR	Correctly Matched Registration
SUSAN	Smallest Univalue Segment Assimilating Nucleus
FAST	Features from Accelerated Segment Test
AST	Accelerated Segment Test
AGAST	Adaptive and Generic Corner Detection Base on the Accelerated Segment Test
SIFT	Scale-Invariant Features Transfrom
SURF	Speeded Up Robust Features
BRISK	Binary Robust Invariant Scaleable Keypoints
2D og 3D	Todimensjonal og tredimensjonal
NSM	Non-Maximum Suppression

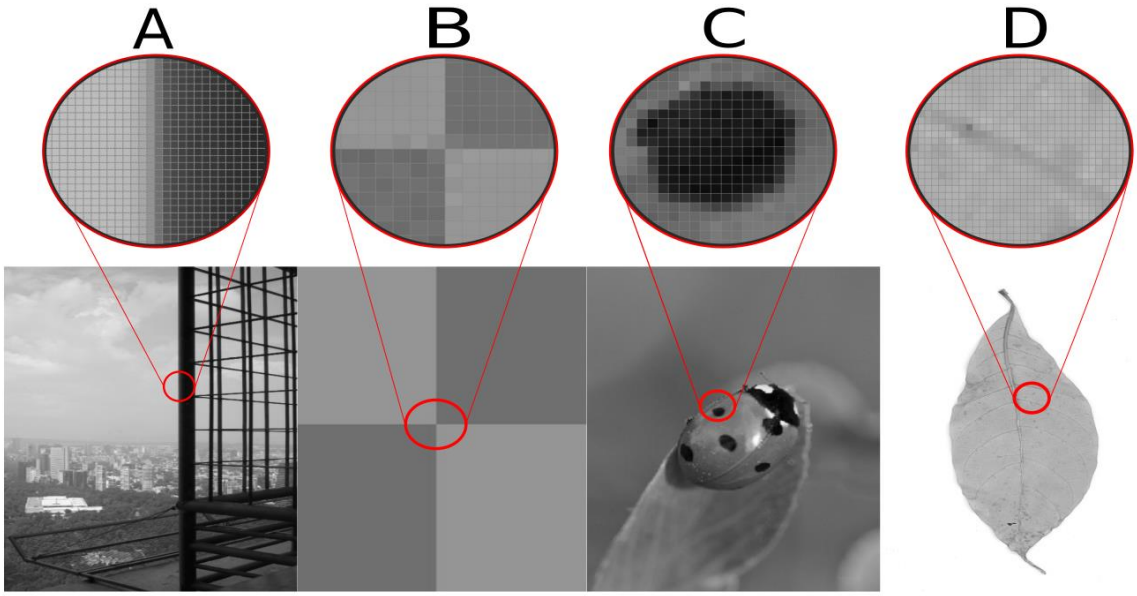
1 Innledning

Egenskapsdeteksjon er å finne fremtredende punkter i et bilde. For å løse denne oppgaven har flere egenskapsdetektorer blitt introdusert i nyere tid. Kanter og hjørner er to av disse fremtredende punktene som kan bli funnet. I denne oppgaven er det valgt ut tre detektorer som blir beskrevet teoretisk og gjort eksperimenter med. Eksperimentene innebærer blant annet å teste robustheten til de tre utvalgte detektorene og visuelt godkjenne egenskapspunktene funnet. Den visuelle godkjennelsen er for å se om egenskapene funnet faktisk er den type egenskap som detektoren skal finne, at et hjørne faktisk er et hjørne. Bakgrunnen for å gjennomføre en analyse av denne typen er fordi det har blitt gjort i mindre grad til dags dato.

1.1 Egenskapspunkter og -detektorer

Et egenskapspunkt kan defineres som et fremtredende punkt som er interessant å finne i et bilde. Et eksempel kan være når et menneske ser på et bilde av et ansikt. Øynene vil i starten fokusere på helheten av bildet. Ved å se på bildet over lenger tid vil det dukke opp flere detaljer, som for eksempel et lite arr, et hårstrå eller andre detaljer som ikke blir oppdaget med det første. Det er dette som er egenskapsdetektering. I maskinsyn brukes egenskapsdetektering ofte som en av de første leddene i bildebehandling. Forskjellige anvendelser hvor egenskapsdetektering blir brukt i maskinsyn kan være bildejustering (panorama stifting), 3D rekonstruksjon, objektgjenkjenning eller robotstyring.

I maskinsyn blir de fremtredende punktene kjent som egenskapspunkter delt inn i to hovedkategorier. Disse to kategoriene er globale og lokale egenskapspunkt [1]. Ved å bruke et globalt egenskapspunkt for å beskrive et bilde, blir bare et egenskapspunkt brukt. Ved å bruke lokale egenskapspunkter, blir bildet beskrevet ved å dele opp bildet i flere deler og finne egenskapspunkter i de forskjellige delene. Det blir lagt vekt på at egenskapspunktene er mest mulig geometrisk og fotometrisk invariant. Geometrisk invariant vil si at egenskapspunktene ikke blir påvirket av translasjon, rotasjon eller skalering. Fotometrisk invariant er at punktene ikke blir påvirket av intensitet forandringer. Disse punktene kan være kanter, hjørner, områder som har annen egenskap enn bakgrunnen [2] og rygg (*ridges*) [3].



Figur 1.1 Viser kant (a), hjørner (b), objekt som er mørkere enn bakgrunnen (c) og rygg (d)

Figur 1.1 viser eksempler på de fire mulige egenskapene. Her viser *a* en kant, de defineres av punkter som har en høy gradient størrelsesorden. Disse blir festet sammen for å skildre en kant. Hjørner (her vist med *b*) defineres av område hvor bildegradienten har et høyt nivå av krumning eller hvor kanten forandrer retning raskt. Områder som varierer i egenskaper sammenlignet med området rundt (kan for eksempel være farge eller lysstyrke) kan bli brukt som egenskapspunkt (her vist med *c*). Siste viser en rygg (*d*), som ikke må forveksles med kant. Ved kant ser man etter grenser mellom høy og lav gråverdi. For å finne rygg letes det etter tynne linjer som er lysere eller mørkere enn sin piksel naboer.

Å finne egenskapspunktene kan deles inn i to steg. Det er selve detektor delen som finner de bestemte egenskapene. For å beskrive egenskapene funnet og området rundt, får hvert punkt en egenskapsvektor. Tildeling av disse vektorene er det deskriptoren som gjør. Ved hjelpe av denne beskrivelsen er det mulig å finne igjen samme egenskapspunkt i to bilder. Hvor god en detektor er bestemmes ut ifra flere kriterier [4]. De tre viktigste er repeterbarhet, tidsbruk og invarians mot endring i skala.

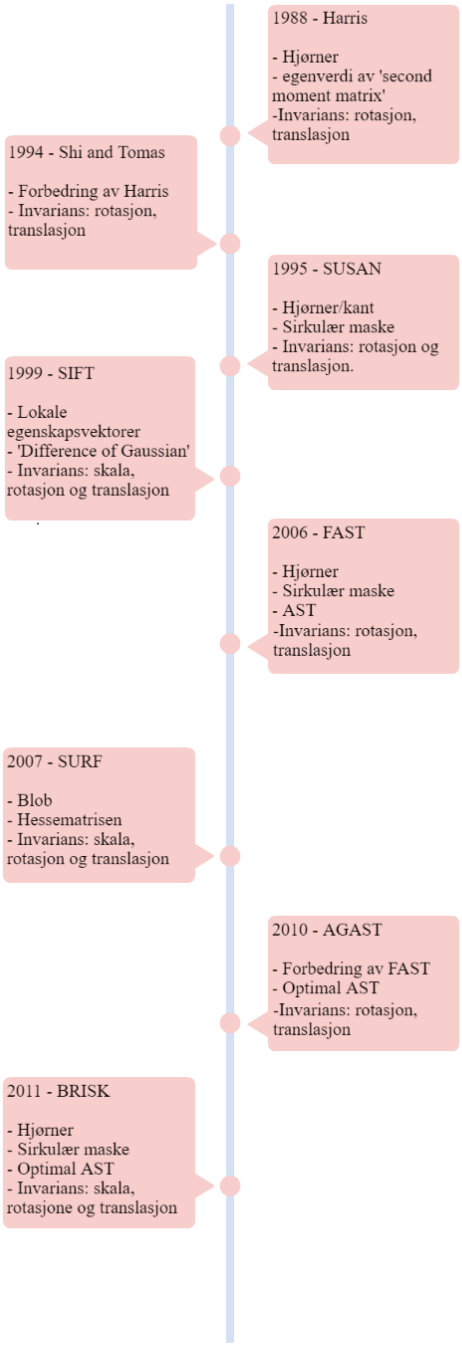
- Repeterbarhet er hvor god en detektor er til å finne igjen det samme egenskapspunktet ved forandring i bildet, enten det er rotasjon eller lysendring.

- Tidsbruk er hvor lang tid detektoren bruker til å finne egenskapspunktene. Dette er spesielt viktig i video eller sanntid applikasjoner.
- Invariant mot endring i skala er hvor god en detektor er hvis skalaen på bildet blir forandret.

Noen av de andre kravene er kvantitet, nøyaktighet og informasjon i mønsteret. Kvantitet er antall egenskapspunkter funnet. Hvor mange egenskaper som bør bli funnet avhenger av hva egenskapsdetektoren skal bli brukt til og hva slags bilde det er. Et minimum av egenskaper bør bli funnet uansett størrelse på bilde. Nøyaktighet er hvor godt posisjonen til egenskapspunktene funnet representerer den virkelige posisjonen til egenskapspunktet i bildet. Det siste kravet er for intensitetsmønstrene benyttet av detektorene og området beskrevet av dem. Her skal området rundt ha mye variasjon slik at egenskapene kan bli beskrevet.

Kravene for detektorene blir strengere med tiden. For å kunne konkurrere med allerede eksisterende detektorer, eller teste detektorene mot hverandre for en spesifikk anvendelse blir nye kriterier laget. Ved evaluering av detektorer og deskriptorer for deteksjon av bevegelse i luftvideo ble for eksempel laget to nye kriterier, PMR (*Portion of Error Registration*) og CMR (*Correctly Matched Registration*) [5].

1.2 Noen publiserte egenskapsdetektorer



Figur 1.2 Tidslinje over noen av de kjente egenskapsdetektorer

En av de mest kjent og første publiserte metodene er Harris-detektoren [5]. Laget av Chris Harris og Mike Stephens, publisert i 1988. Detektoren bruker egenverdiene til autokorrelasjonsmatrisen (*second moment matrix*). Metoden er en forbedring av Moravec [6] detektoren ved å gjør den mer repeterbar ved små bilde variasjoner og nærme kanter. Flere metoder har blitt utviklet for å forbedre Harris detektoren, de mest kjente er Shi-Thomas [7] og Harris affine [8].

En annen kjent metode er SUSAN-detektoren (*Smallest Univalue Segment Assimilating Nucleus*) [9]. Presentert i 1995 av S.M Smith og J.M Brady. Denne metoden benytter seg av en sirkulær maske som sammenligner intensiteten til senterpikselet med nabopiskelene. Metoden er kjent for å være robust mot støy ettersom metoden ikke benytter seg av noen form av den deriverte.

I nyere applikasjoner har beregningshastigheten blitt mer og mer viktig, spesielt for bruk i sanntid applikasjoner. Siden 1995 har andre metoder benyttet seg av sirkulær maske men med sin egen vri. I motsetning til SUSAN som tester alle pikslene innenfor sirkelen, blir intensiteten til senterpikselen i FAST (*Features From Accelerated Segment Test*) [10] sammenlignet med pikslene på grensen til masken. Ved hjelp av AST (*Accelereated Segment Test*) blir det bestemt om punktet er et egenskapspunkt. AGAST (*Adaptive and Generic Accelerated Segment Test*) [11] bygger videre på FAST ved å bruke de samme maskene, i tillegg blir optimal AST benyttet istedenfor ordinær AST som FAST bruker. Forskjellen mellom ordinær og optimal er at ordinær bruker et treningssett for å lage det binære beslutningstreet, mens optimal AST bruker en adaptiv og generisk algoritme for å lage det binære beslutningstreet. Det er også mulig å bruke flere trær i AGAST. Dette gjør at AGAST ofte kan være mer nøyaktig og fleksibel enn hva FAST er.

Dessverre er ingen metodene nevnt over invariant mot skala forandringer. Tony Lindeberg viste at det gaussiske filteret og den deriverte er et passende glattfilter for skala analyse [12]. Siden den gang har flere metoder benyttet denne metoden for å oppnå invarians mot skalaendring. Den mest kjente av de skala invariante detektorene er SIFT *(Scale-Invariant Features Transfrom*) [13] som ble introdusert av David G. Lowe i 1994. Metoden finner egenskapspunkter ved hjelp av gaussisk differanse og lokale ekstremalpunkter, dessverre kan SIFT være tung beregningsmessig. SURF (*Speeded Up Robust Features*) [14] bygger på de samme prinsippene som SIFT. Metoden benytter integral bilder og tilnærmede boksfilter for å

finne egenskapspunkter, og ved hjelp av det oppnår metoden en raskere beregningshastighet enn SIFT. BRISK (*Binary Robust Invariant Scalable Keypoints*) [15] er en annen detektor som er invariant mot skala endring ved å bruke gaussisk filter. Metoden er mindre ressurskrevende ettersom den bygger videre på AGSAT.

1.3 Tidligere arbeid innen evaluering av detektorer

Å evaluere hvor god en egenskapsdetektor er vanlig å gjøre ved hjelp av en av fire metoder. Disse metodene er visuell inspeksjon, nøyaktighet i plassering, teoretisk analyse, absolutte sannhet og bruk detektorene på en bestem oppgave [16].

- Visuell inspeksjon er en subjektiv metode hvor egenskapene blir godkjent ved å se på det med menneskeøyne. Det kan være å studere egenskapene på pikselnivå eller studere bildet på en større skala. Metoden ble blant annet brukt ved å finne beste detektor for deteksjon av rygger og daler [17].
- Nøyaktighet i plassering er en metode som bestemmer hvor god en detektor er ved å se på om punkter i 2D-scenen samsvarer med de virkelige 3D-punktene. Ved bruk av denne metoden må punktene i 3D-scenen være kjent. Er spesielt viktig i kamerakalibrering.
- Teoretisk analyse metoder er nokså begrenset ettersom de bare kan brukes på veldig spesifikke egenskaper. Metodene undersøker oppførselen til detektoren for teoretiske funksjonsmodeller.
- Absolutte sannhet er en metode som bruker antall falsk positive og ikke detekterte egenskaper til å bestemme hvor god en detektor er. Denne metoden ble blant annet brukt for å evaluere beste detektor og deskriptor til å finne bevegelse i video tatt fra luften [18].
- Bestemt oppgave er en metode som ikke ser på det å finne egenskapspunkter som det endelige steget, men heller som en av de første prosessene i bildebehandling. Ved hjelp algoritmer for å gjenkjenne spesifikke objekter i bildet, kan egenskapsdetektorene bli testet. Objekter som skal bli funnet kan være for eksempel være biler [19].

I nyere tid har det blitt vanlig å teste nyutviklede metoder opp mot kjente metoder eller metoder som er laget for å takle det samme problemet (det kan være invarians, beregningshastighet eller ressursbruk). I flere av publikasjonene nevnt i forrige delkapittel er det inkludert grafer eller

tabeller for å vise resultatet av disse testene. Figur 1.3 viser et eksempel på en sammenligning av SIFT, SURF og BRISK, som ble gjort ved publiseringen av BRISK [15].

	SIFT	SURF	BRISK
Detection threshold	4.4	45700	67
Number of points	1851	1557	1051
Detection time [ms]	1611	107.9	17.20
Description time [ms]	9784	559.1	22.08
Total time [ms]	11395	667.0	39.28
Time per point (ms)	**6.156**	**0.4284**	**0.03737**

Figur 1.3 Viser antall egenskaper funnet, tid brukt på å finne egenskapene og til å beskrive egenskapene for de tre detektorene. Hentet fra [15]

1.4 Valg av detektorer

For denne oppgaven har det blitt valgt ut tre kjente egenskapsdetektorer. Detektorene valgt anvendt i denne oppgaven er BRISK, SUSAN og SURF. Disse detektorene har forskjellige styrker og svakheter, slik som endring i skala, støy og intensitet.

- BRISK er en metode som består av detektor og deskriptor. Den er kjent for å være rask og beregningsmessig effektiv. Metoden detekterer hjørner.
- SUSAN er den eldste av de valgte metodene. Metoden består av en detektor som kan detektere kanter eller hjørner. Metoden var den første til å benytte sirkulær maske, som senere har blitt benyttet av andre metoder.
- SURF er en av de mest kjente detektorene når det kommer til invarians mot skala endring. Metoden detekterer sammenhengende piksler som er mørkere eller lysere enn området rundt (bakgrunnen).

1.5 Rapportens struktur

Kapittel 2 – Teori

I dette kapittelet blir teorien bak de tre valgte detektorene anvendt i rapporten beskrevet: BRISK, SURF og SUSAN. Hvor teorien bak SUSAN danner grunnlaget for implementeringen i MATLAB. I tillegg til teorien bak detektorene, blir metoder avvendt av detektorene beskrevet.

Kapittel 3 – Eksperiment

I dette kapittelet blir programmene brukt beskrevet samt utførelsen av eksperimentene. Robustheten til detektorene blir testet ved å finne egenskapspunkter i bilder påvirket av støy, lydendringer og endringer i skala.

Kapittel 4 – Resultat og diskusjon

I dette kapittelet blir resultatet av de forskjellige eksperimentene lagt frem ved hjelp av bilder og tabeller. Tidsbruken de forskjellige egenskapsdetektorene bruker på å finne egenskapspunkter i de forskjellige eksperimentene vil også bli tatt med. Diskusjonen vil også være inkludert i dette kapittelet.

Kapittel 5 – Konklusjon

I dette kapittelet blir det lagt frem en konklusjon basert på resultatene fra eksperimentene og diskusjonen.

Appendiks A Forkortelser og ordforklaring

Inneholder forkortelser og ordforklaring på ord brukt i rapporten

Appendiks B MATLAB kode

Inneholder informasjon om kildekoden brukt i rapporten.

Appendiks C Bilder

Inneholder bilder brukt under eksperiment delen av prosjektet.

Appendiks D Tabeller

Inneholder komplette tabeller av resultatet fra eksperimentene.

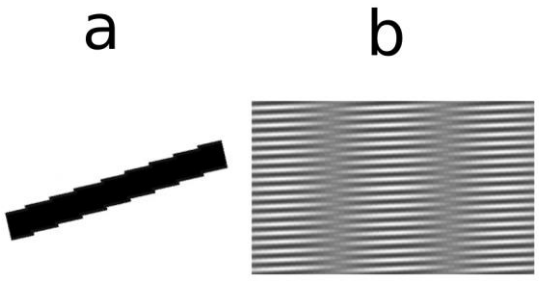
2 Teoretisk beskrivelse av aliasing, metoder brukt av detektorene og valgte detektorer

I dette kapittelet blir teorien for SURF, SUSAN og BRISK presentert. I tillegg til teorien bak de valgte detektorene, blir teorien for Aliasing, *Non-Maximum Suppression* (NMS) og integralbilde beskrevet. NMS er en metode som setter pikselverdien til 0 hvis pikselen ikke er et maksimum, metoden blir benyttet i SURF og BRISK. Integralbildet blir brukt i SURF for å øke beregningshastigheten. Aliasing er en bildetransformasjon som kan påvirke antall egenskaper lokalisert av egenskapsdetektorene i et bilde.

2.1 Aliasing

Ved digital sampling av signal, enten det er lyd, digitale bilder eller video, kan det oppstå aliasing. Det kan skje når et bilde med høy oppløsning blir vist frem ved en lavere oppløsning. I bildebehandling kan dette forkomme hvis bildedata blir prosessert, eksempler på dette kan være når digitale bilder lagres på en datamaskin eller ved å gjøre et bilde om til gråskala [20]. Dette kalles for spatial aliasing.

Ved spatial aliasing kan bildet bli påvirket på forskjellige måter, det kan være taggete kanter eller moarémønster [21]. Ved taggete kanter kan glatte kanter i originalbildet bli mer hakkete og røffe i det nye bildet. Moarémønster er en optisk effekt som oppstår når to jevne og fine mønstre blir lagt over hverandre og danner et tredje mønster. Figur 2.1 viser disse to påvirkningene.



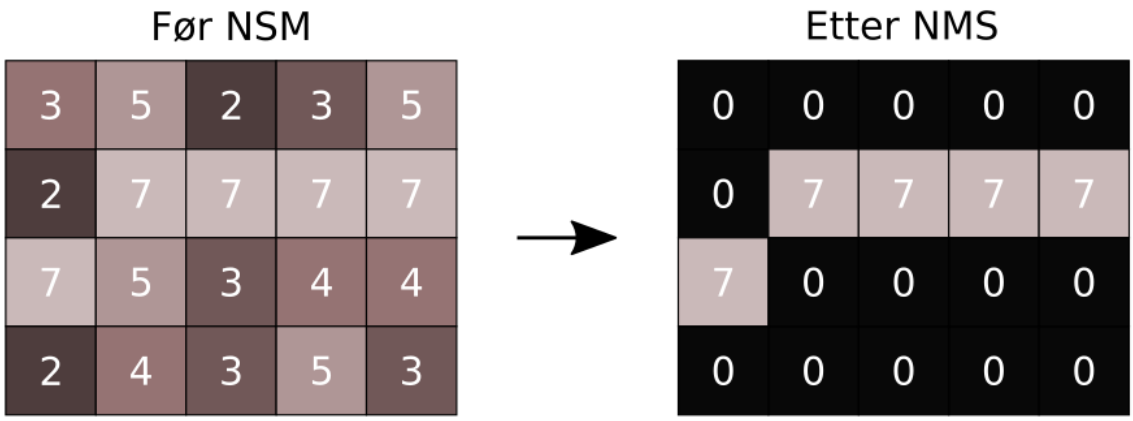
Figur 2.1 viser aliaseffekten taggete kanter (a) og moarémønster (b)

For å fjerne eller minimere de optiske forstyrrelsene forårsaket av aliasing blir det benyttet anti-aliasing. En av de enkleste formene for anti-aliasing er å sende bildet gjennom et lavpassfilter, her blir høye frekvenser i bilde fjernet. Figur 2.2 viser resultatet av figur 2.1 a sendt gjennom et lavpassfilter.

Figur 2.2 viser resultatet av anti-aliasing på figur a 2.1

2.2 Non-Maximum Suppression

Non-Maximum Suppression (NMS)) er en metode som blir benyttet i flere bildebehandlings algoritmer, blant annet i to av de tre valgte egenskapsdetektorene. Algoritmen går ut på å lete etter den største lokale maksima. I 2D blir området rundt pikselen beskrevet ved $(2n + 1) \times (2n + 1)$ [22]. Ved funn av lokal maksima får resterende av pikslene i området satt sin skalarverdi til null. Dette gjør at det er mulig å blant annet tynne ut kanter og fjerne støy. Figur 2.3 viser et eksempel på hvordan et område på $5x4$ piksler ser ut før og etter NMS.



Figur 2.3 viser et enkelt eksempel av et område på 4x5 piksler før og etter NMS

2.3 Integralbilder

Innenfor maskinsyn ble metoden gjort populær av Lewis samt Paul Viola og Michael Jones med metoden deres Viola-Jones for objektgjenkjenning [23]. Hvor sistnevnte brukte navnet integralbilde.

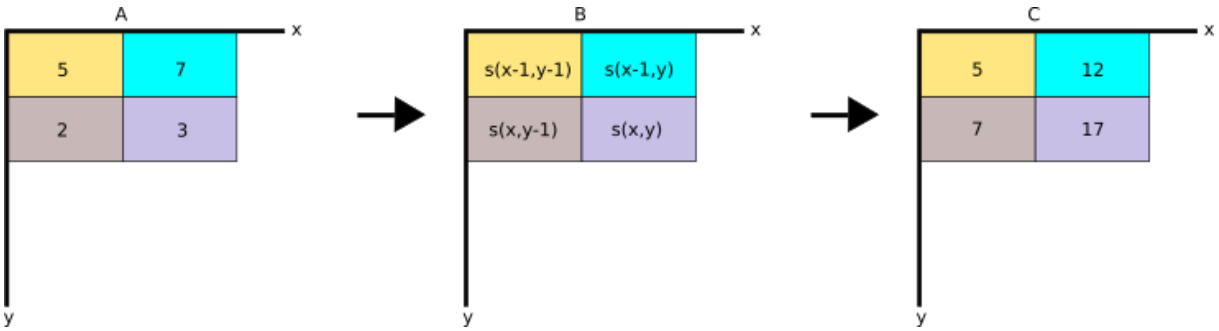
Et integralbilde representerer summen av alle pikslene i et rektangulært område dannet av origo og et gitt punkt $x = (x, y)^T$ i bildet I. Integralbilder blir hovedsakelig brukt til å finne gjennomsnittintensiteten i et bilde. Fordelen med integralbilder er at de kan bli konstruert ved å bare gå over bildet en gang.. Formel 2.3.1 viser beregningen av integralbildet $I_{\sum(x)}$.

$$I_{\sum(x)} = \sum_{i=0}^{i \leq x} \sum_{i=j}^{j \leq y} I(i,j) \tag{2.3.1}$$

Integralbildet blir beregnet ved addere den aktuelle pikselverdien med pikselverdiene som er direkte over og under. Formel 2.3.2 viser utregning av et integralbilde. Her er s integralbildene og i er verdien pikselen har før integralbildet blir regnet ut.

$$s[x,y] = i[x,y] + s[x-1,y] + s[x,y-1] - s[x-1,y-1] \tag{2.3.2}$$

Figur 2.4 viser et eksempel på hvordan et ønsket integralbilde blir beregnet ved hjelp av formelen over. Her er A originalbildet og C er det endelige integralbildet etter at integralbildene har blitt regnet ut.



Figur 2.4 Utregning av integralbilde. Bilde *a* viser før integralbildene blir regnet ut, bilde *b* viser formelen for å regne ut integralbilder og bilde *c* viser resultatet.

2.4 SURF

SURF (*Speeded-Up Robust Features*) ble utviklet av Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool i 2007. Som navnet tilsier var motivasjonen å lage en robust og rask måte å detektere egenskapspunkter på. Metoden leter etter områder som er lysere eller mørkere enn resten av området rundt. SURF-detektoren benytter seg av Hessematrisen for å finne egenskapspunktene. Deskriptoren beregner Haar-wavelet responser for å lage en beskrivelse av egenskapspunktet.

2.4.1 Detektoren

SURF-detektoren bruker en enkel form av Hessematrisen. Denne matrisen er en kvadratisk matrise som består av andreordens partiellderiverte til skalarfunksjonen. Detektorer som benytter seg av Hessematrisen er raskere og mer repeterbare enn detektorer som baserer seg på Harris [14]. Formel 2.4.1 viser en generell $2x2$ Hessematrise.

$$\mathcal{H} = \begin{bmatrix} \frac{\delta^2 f}{\delta^2 x^2} & \frac{\delta^2 f}{\delta x \delta y} \\ \frac{\delta^2 f}{\delta y \delta x} & \frac{\delta^2 f}{\delta^2 y^2} \end{bmatrix} \tag{2.4.1}$$

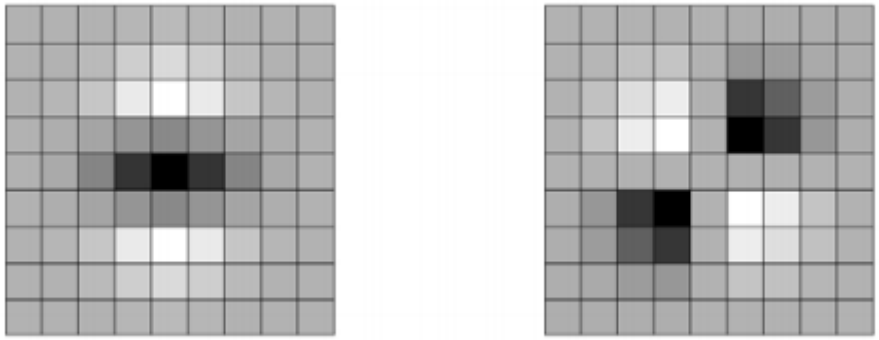
Ved å konvolvere bildet med den gaussiske andre ordens partiell deriverte er det mulig å lokalisere egenskapspunkter og passende skala. Dette gjøres ved å se på størrelsen til determinanten til den nye konvolusjons matrisen. Egenskapspunktet er lokalisert der determinanten er størst. Formel 2.4.2 viser konvolusjonen mellom den gaussiske andre ordens partiell deriverte og punktet x i bildet. Hvor $L_{xx}(\boldsymbol{x}, \sigma)$ er resultatet av konvolusjonen, $\frac{\partial^2}{\partial x_2} g(\sigma)$ er gaussisk utjevningsfilter (med standardavvik σ) som er derivert to ganger og $I(\boldsymbol{x})$ er intensiteten i punktet $\boldsymbol{x} = (x, y)$ i bildet.

$$L_{xx}(\boldsymbol{x}, \sigma) = I(\boldsymbol{x}) * \frac{\partial^2}{\delta x^2} g(\sigma) \tag{2.4.2}$$

Tilsvarende blir det for $L_{xy}(\boldsymbol{x}, \sigma)$ og $L_{yy}(\boldsymbol{x}, \sigma)$. Dermed blir konvolusjons matrisen seende ut som formel 2.4.3. Her er $\mathcal{H}(\boldsymbol{x}, \sigma)$ konvolusjons matrisen og elementene er som beskrevet over.

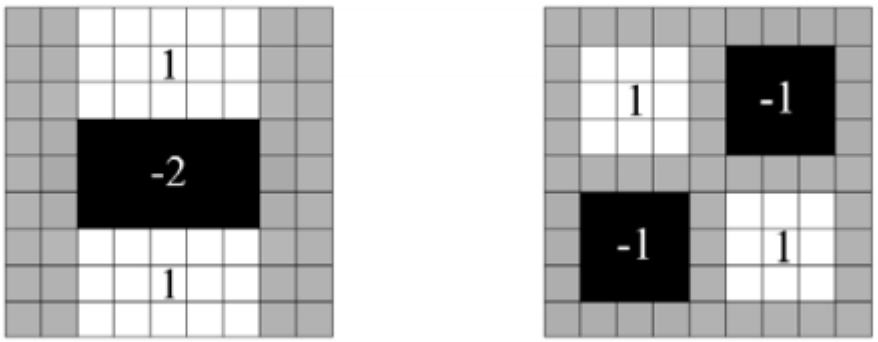
$$\mathcal{H}(\boldsymbol{x},\sigma) = \begin{bmatrix} L_{xx}(\boldsymbol{x},\sigma) & L_{xy}(\boldsymbol{x},\sigma) \\ L_{xy}(\boldsymbol{x},\sigma) & L_{yy}(\boldsymbol{x},\sigma) \end{bmatrix} \tag{2.4.3}$$

Figur 2.5 viser hvordan den gaussiske andre orden partiell deriverte masken ser ut for $L_{xy}(\boldsymbol{x},\sigma)$ og $L_{yy}(\boldsymbol{x},\sigma)$ (masken har her blitt klippet til og diskretisert).



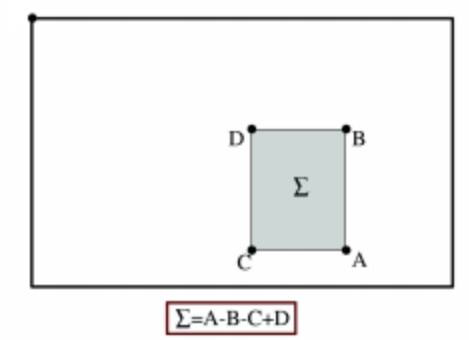
Figur 2.5 Til venstre er den gaussiske partiell deriverte i y-retning, den til høyre er den deriverte i xy-retning. Bildet er hentet fra [14]

Figur 2.6 viser hvordan en gaussiske partiell deriverte maske med $\sigma = 1.2$ blir tilnærmet ved hjelp av boksfilter. Denne tilnærmingen representerer den minste masken benyttet av detektoren. Ytelsen er tilnærmet lik eller bedre som om det hadde blitt brukt en gaussisk maske som klippes til og diskretiseres.



Figur 2.6 Til venstre 9x9 boksfilter for yy-retning, til høyre viser for xy-retning. Bildet er hentet fra [14]

For å kunne beregne boksfiltrene så raskt som mulig benytter SURF seg av integralbilder (*summed area table*). Ved hjelp av integralbildet kan den gjennomsnittlige intensiteten bli funnet ved hjelp av tre addisjoner uansett størrelse på rektangelet. Dette gjør at beregningstiden ikke blir påvirket av størrelsen på rektangelet. Dette er viktig ettersom SURF-metoden benytter seg av store filtre. Figur 2.7 viser hvordan den total intensiteten blir utregnet ved hjelp av tre addisjoner.



Figur 2.7 Utregning av totale intensiteten ved bruk av tre addisjoner. Bildet er hentet fra [14]

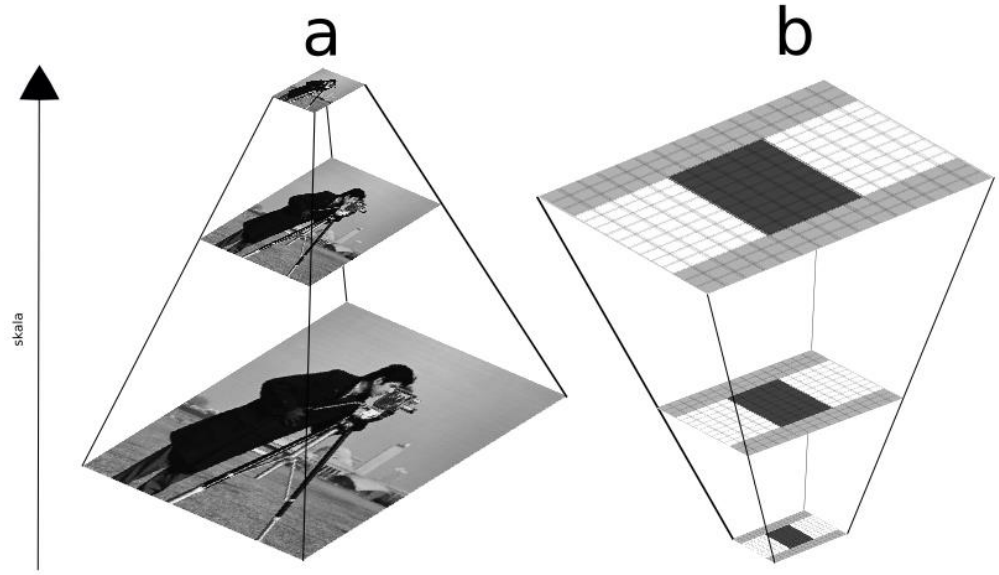
Tilnærmingene blir kalt D_{xx}, D_{xy} og D_{yy}. Det blir også lagt på en vekting på de rektangulære områdene. Det blir brukt en så enkel som mulig vekting for at det ikke skal påvirke ytelsen av detektoren. Grunnen til vektingen er å bevare energien mellom gaussiske kjernen og den tilnærmede kjernen. Likningen for vektingen w er her vist ved formel 2.4.4.

$$w = \frac{\left|L_{xy}(1.2)\right|_F \left|D_{yy}(9)\right|_F}{\left|L_{xx}(1.2)\right|_F \left|D_{xy}(9)\right|_F} = 0.912 \ldots \simeq 0.9 \qquad (2.4.4)$$

Hvor $|\sim|_F$ Frobenius norm, σ er satt til 1.2 og et område på $9x9$ er brukt. Ut ifra formelen over vil vektingen w kunne forandre seg avhengig av skala. SURF-metoden setter vektingen til konstant 0.9. Det gjøres ettersom det ikke vil ha særlig påvirkning på det endelige resultatet. Dette gjør at formelen for å regne ut determinanten til den estimerte Hessematrisen blir som vist ved formel 2.4.5.

$$\mathrm{Det}(\mathcal{H}_{tilnærmet}) = D_{xx} \cdot D_{yy} - \left(0.9 \cdot D_{xy}\right)^2 \qquad (2.4.5)$$

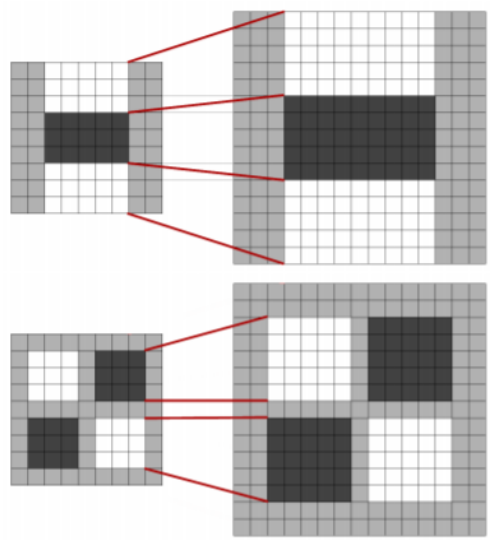
For at detektoren skal være skala invariant benytter ikke SURF seg av en gaussisk bildepyramide som mange andre egenskapsdetektorer gjør. Ved å bruke boksfilter og integralbilder slipper SURF å nedskalere bildet gjentatte ganger. I stedet blir filter størrelsen oppskalert for hver iterasjon. Dermed beholder bildet sin originale oppløsning gjennom hele prosessen. Ettersom det ikke er noen form for nedskalering av bildet trengs det heller ikke noen form for kantutjevning (anti-aliasing). Figur 2.8 viser forskjellen mellom de to metodene.



Figur 2.8 Til venstre viser nedskalering av et bilde for hver iterasjon (a), bildet til høyre viser oppskalering av et filter for hver iterasjon (b).

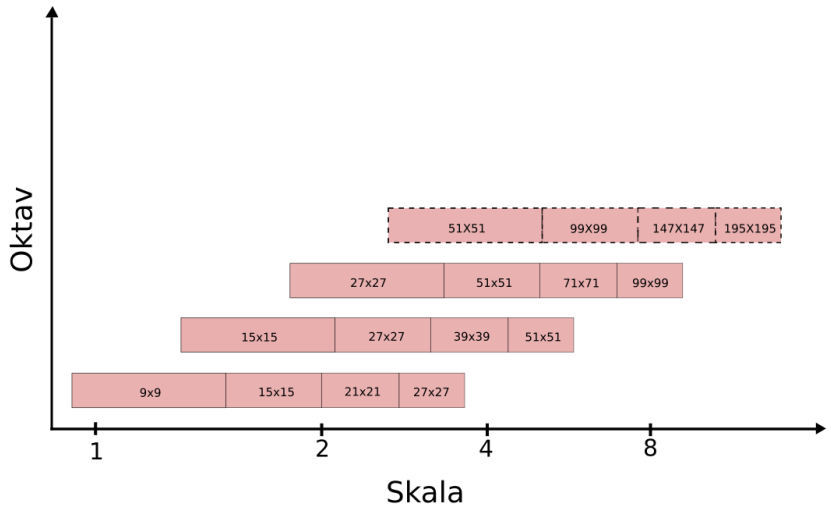
Utgangsverdien av 9x9 filteret blir brukt som den første skalaen, også referert som skala $s = 1.2$. De neste skalaene blir laget ved å gradvis øke filter størrelsen over bildet. Bakgrunnen for å benytte denne metoden er at det er beregningsmessig effektivt.

I en gaussisk bildepyramide blir skalaen delt inn i oktaver. Hvor mellom hver oktav bildet blir nedskalert med en faktor på 2. I SURF vil en oktav tilsvare en serie filterrespons kart oppnådd ved å konvolvere inngangsbildet med et filter av økende størrelse. Figur 2.9 viser hvordan et 9x9 filter blir skalert opp til å bli et 15x15 filter.



Figur 2.9 Viser oppskalering av 9x9 filter til 15x15 filter. Bildet hentet fra [14]

For å forsikre at filteret forblir odde og at et senter piksel er tilstede når filteret blir oppskalert, må filteret minimum bli økt med to piksler (en piksel på hver side). Dette gjør at filteret øker med en størrelse på 6 piksler i første oktav. Den første oktaven vil bestå av initial filteret 9x9 som søker etter egenskapspunktet i den laveste skalaen. De neste størrelsene i den første oktaven blir 15x15, 21x21 og 27x27. For hver ny oktav økes filter størrelsen med en faktor på 2 (fra 6 i første oktav til 12 for andre oktav. De neste blir 24 og 48). Oktavene overlapper for å dekke alle skalaene sømløst. Figur 2.10 viser hvordan filter størrelsen øker i hver oktav og hvordan de overlapper.



Figur 2.10: Viser hvordan filter størrelsen øker innenfor oktav og neste oktav. Siste oktaven (stiplet linje) blir brukt hvis bildet er større en tredje oktav.

Hvis bildestørrelsen er større enn siste oktaven kan en fjerde oktav med filterstørrelse 51x51 ,99x99, 147x147 og 195x195 bli brukt. Flere oktaver kan bli undersøkt, men antall egenskapspunkter oppdaget minker fort for hver oktav.

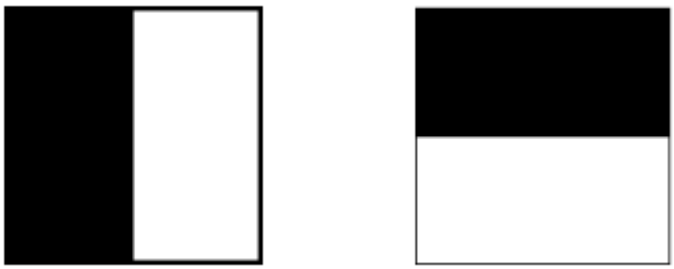
For å lokalisere egenskapspunktene blir en rask variant av NMS utviklet av Neubeck og Van Gool brukt. Deretter blir maksimalverdien av Hessematrisens determinant interpolert med skala og '*image space*' med en metode utviklet av Brown.

2.4.2 Deskriptoren

SURF-deskriptoren kan bli delt opp i to deler. Hvor den første delen sørger for at deskriptoren blir invariant for rotasjon. Den andre delen er beskrivelsen av selve egenskapspunktet. Begge delene benytter seg av Haar wavelet filterrespons.

For å gjør deskriptoren invariant for rotasjon blir Haar wavelet filterrespons beregnet i x- og y-retning innenfor et sirkulært område med $6 \cdot s$. Hvor s er skalaen hvor egenskapspunktet ble detektert. Ved å sette størrelsen på sidelengden av wavelet'en til $4 \cdot s$ kan det benyttes integralbilder for å utføre beregningen. Dette gjør at det trengs bare seks operasjoner for å

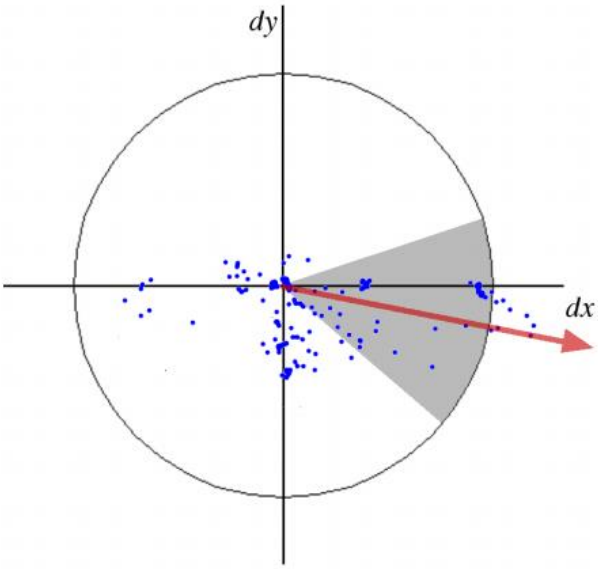
beregne filterresponsen i x- og y-retning uansett skala. Figur 2.11 viser konstruksjonene av Haar wavelet filteret i x- og y-retning.



Figur 2.11 Haar wavelet filter i x-retning (venstre) og y-retning (høyre). Her har de svarte områdene vekting -1 og de hvite +1. Bildet hentet fra [14]

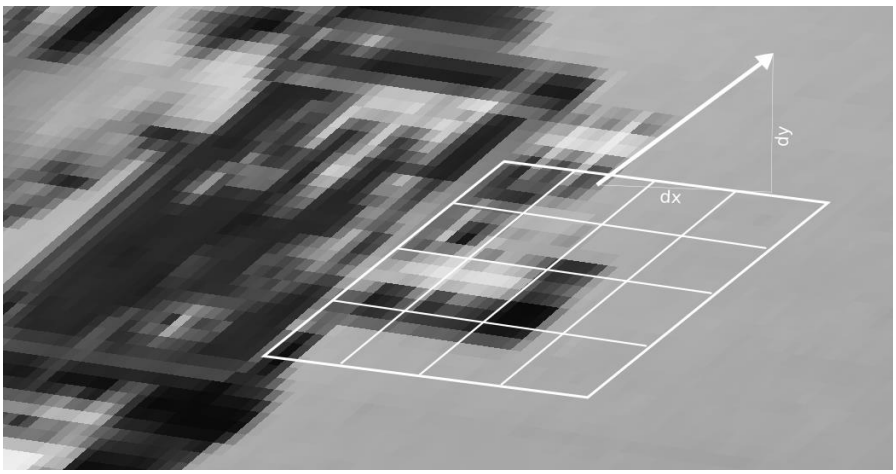
Etter filterresponsen har blitt regnet ut og vektet med et gaussisk filter sentrerte rundt egenskapspunktet satt til $\sigma = 2 \cdot s$, blir responsen representert ved hjelp av punkter langs den vertikale og horisontale aksen. Den mest dominerende orienteringen blir estimert ved å beregne den totale summen av de horisontale- og vertikale responsene innenfor et roterende vindu av størrelse $\frac{\pi}{3}$. De summerte responsene blir gjort til en vektor, hvor den lengste vektoren definerer den beste orienteringen av egenskapspunktet. Figur 2.12 viser responsen representert med de tilhørende punktene, samt vinduet og den lengste vektoren.

U-SURF er en metode patentert i USA som er en variant av SURF som ikke er invariant for rotasjon. Noe som resulterer i raskere beregning av egenskapspunktet.



Figur 2.12 De blå punktene representerer responsen, grå området er vinduet med størrelse $\frac{\pi}{3}$ og den røde pilen er den lengste vektoren. Bildet hentet fra [14]

For å beskrive egenskapspunktet blir et kvadratisk område med størrelse $20 \cdot s$ konstruert rundt egenskapspunktet med samme orientering funnet tidligere. For å bevare viktig informasjon blir området splittet opp i mindre underregioner av størrelse $4x4$. For hver underregion blir Haar wavlet filterresponsen beregnet (Haar responsen i horisontal retning betegnet d_x og i vertikal d_y). Ved å vekte Haar wavlet filterresponsen i underregionene med et gaussisk filter sentrert rundt egenskapspunktet lik $\sigma = 3.3 \cdot s$, kan robustheten mot geometrisk deformasjoner og lokaliseringsfeil bli økt.



Figur 2.13 Viser hvordan deskriptoren blir bygget. Her vises en underregion på 4x4. For hver underregion blir Haar wavlet filterresponsen beregnet.

Etterpå blir Haar wavlet responsen d_x og d_y i hver underregion summert og satt inn i en egenskapsvektor. I tillegg blir summen til absoluttverdien av d_x og d_y beregnet for å inkludere informasjon om polariteten til intensiteten. Dermed vil hver underregion ha en firedimensjonal egenskapsvektor $\boldsymbol{v} = [\sum d_x, \sum d_y, \sum|d_x|, \sum|d_y|]$. Dette fører til at egenskapspunktet vil kunne bli beskrevet ved hjelp av en egenskapsvektor med en lengde på 64. For å oppnå invarians mot kontrastforandringer blir vektoren normalisert.

2.5 BRISK

BRISK (*Binary Robust Invariant Scalable Keypoints*) ble utviklet av Stefan Leutenegger, Margarita Chli og Roland Y. Siegwart i 2011. Motivasjonen bak BRISK var å lage en metode av høy kvalitet som var rask og lite ressurskrevende. Dette er spesielt viktig i sanntid applikasjoner. Den består av en detektor som bygger videre på den populære FAST-detektoren (*Features From Accelerated Segment test*) laget av Edward Rosten og Tom Drummond i 2005. Og en binær deskriptor blir laget ved hjelp av en test som sammenligner intensiteten.

2.5.1 Detektoren

For at detektoren skal være invariant til endring av skala, starter BRISK med å skalere bildet i en gaussisk pyramide (se figur 2.14). Den gaussiske pyramiden fungere på den måten at originalbildet blir nedsamplet for hvert nivå [24].

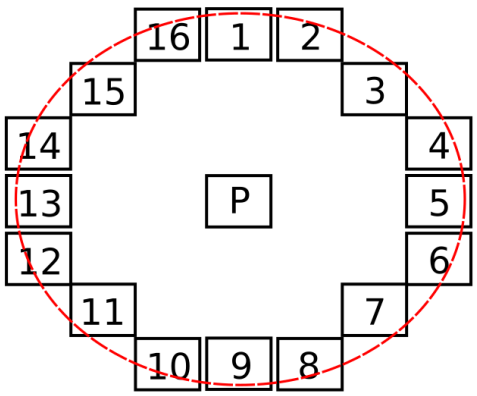
Figur 2.14 Gaussisk pyramide. Nederste bildet er original skala, de øvrige bildene er de forskjellige nivåene som er nedsamplet med 2.

Metoden bruker ordet oktaver for hvert av nivåene. Pyramiden vil innehold n oktaver c_i som er nedsamplet med 2. I tillegg til oktavene benytter metoden seg også av intra-oktaver. Disse kommer mellom hver oktav, og er nedsamplet med 1.5. Så ved n oktaver c_i finnes det også n midt-oktaver d_i. Hvis t er skala blir formlene $t(c_i) = 2^i$ og $t(d_i) = 1.5 \cdot 2^i$. For BRISK er n

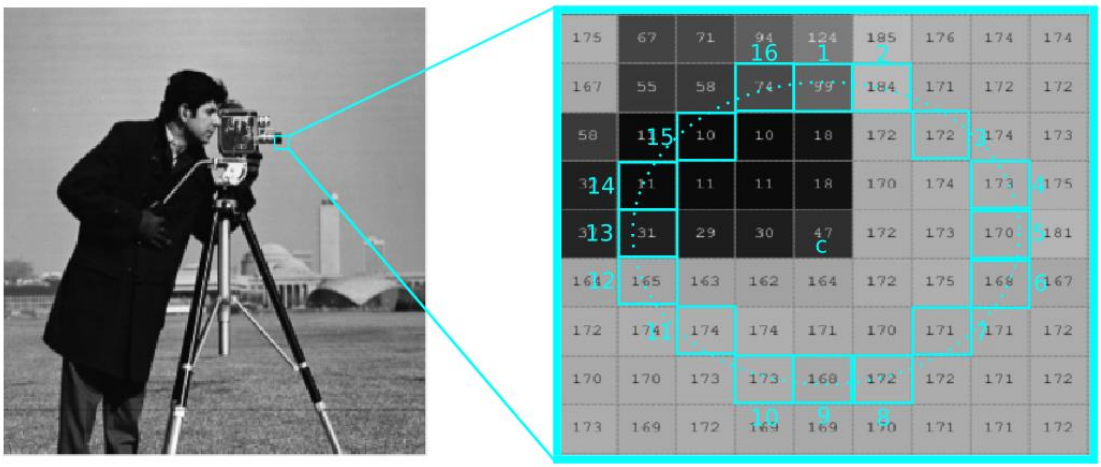
satt til $n = 4$. Figur 2.15 nedenfor viser hvordan et bilde vil se ut når det har blitt nedsamplet med 2 i tre omganger.

Figur 2.15 Bildet til venstre er det originale, de andre er nedsamplet versjoner (henholdsvis 2,4 og 8 ganger nedsamplet i forhold til originale). Bildene er skalert opp for at de skal være like store.

Etter nedsamplingen blir en FAST-detektor lagt på hver av oktavene og intra-oktavene. Det finnes flere forskjellige masker, i BRISK blir det benyttet to masker. FAST 9-16 masken fungerer på den måten at den går over bildet og tester hver piksel. For at pikselen som blir testet skal bli godkjent som egenskapspunkt, må 9 av 16 sammenhengende nabopiksler være mørkere eller lysere enn pikselen som blir testet. Figur 2.16 og 2.17 viser hvordan en slik maske ser ut.



Figur 2.16 Fast 9-16 maske. Her er P pikselen som blir testet.



Figur 2.17 Viser 9-16 masken på et ekte bilde, hvor c er senterpikselen av masken.

Den andre masken som blir brukt er FAST 5-8. Fungerer på samme måten som 9-16. Men som navnet tilsier blir mindre nabopiksler testet. Denne masken blir brukt på oktav c_0. Dette gjøres for at det skal være mulig å ha en virtuell intra-oktav d_{-1} under c_0.

For å teste intensiteten av pikselen mot sine naboer, blir et beslutningstre med tre valg benyttet. Pikselverdien kan være mellom 0 og 255. Formel 2.5.1 viser hvordan beslutningstreet ser ut.

Tilstandene i beslutningstreet er definert som:

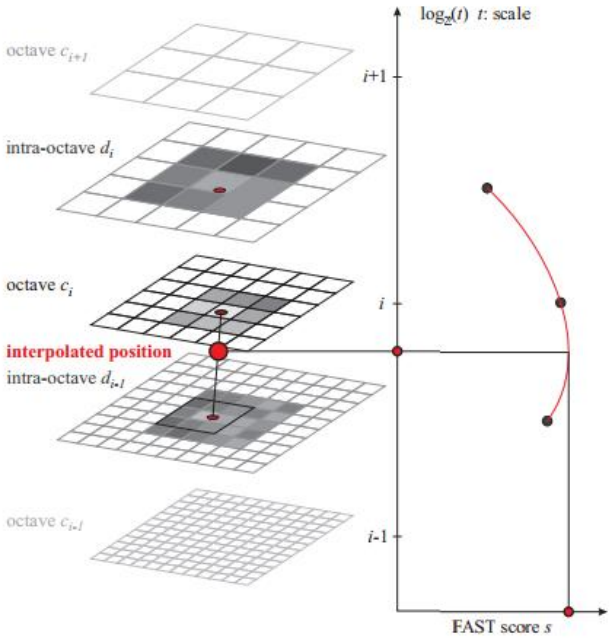
- d (darker) - mørkere enn test piksel (intensitet verdi nærmere 0)
- s (similar) – piksel intensitet er lik for de to pikslene
- b (brighter)– lysere enn test piksel (intensitet verdi nærmere 255)

$$S_{p\to x} = \begin{cases} d, I_{p\to x} \leq I_p - t \\ s, I_p - t < I_{p\to x} < I_p + t \\ b, I_p + t \leq I_{p\to x} \end{cases} \quad (2.5.1)$$

Pikselverdien her er I (hvor I_p er pikselen i midten av masken og $I_{p\to x}$ er piksel 1 til 16 testet mot I_p). Tilstanden til hver piksel i forhold til test piksel blir betegnet med $S_{p\to x}$. Terskling her betegnet med t, bestemmer verdien for et at piksel skal bli satt som enten mørkere eller lysere

enn testpikselen. En høy terskelverdi vil gi skarpe hjørner, men samtidig øker kravet for at det skal bli definert som hjørne. En lav terskelverdi vil resultere i det motsatte.

Etter egenskapspunktene er funnet i hver oktav, blir NMS benyttet for å få maksima pikselen alene. Det betyr at piksler som ikke er et maksimum får skalarverdien satt til null (svart), som gjør at bare de lokale maksimaene står igjen. Slik at de kan bli tildelt en FAST verdi *s* ('*salience*'). Verdien beskriver hvor fremtredende egenskapspunktet er. For å finne ut hvilken oktav som inneholder det mest fremtredende egenskapspunktet, blir oktaven testet opp mot oktaven over og under. Ved å bruke en maske på 3x3 piksler blir det samme området testet i de tre oktavlag.



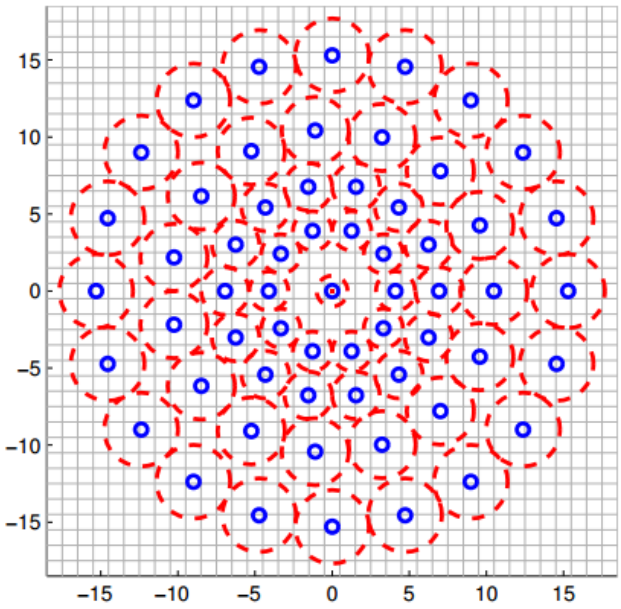
Figur 2.18 Til venstre viser oktaver og intra-oktaver. Til høyre viser hvordan s verdien blir forandret ved test av oktavene. Bildet hentet fra [15]

Ved å benytte en todimensjonal kvadratisk funksjon er det mulig å forenkle kompleksiteten av forbedringsprosessen. Den kvadratiske funksjonen blir lagt over tre oktaver og resulterer i tre forbedrete 'salience' maksima verdier. I det neste steget blir de forbedrete verdiene brukt til å passe en endimensjonal parabel som resulterer i den endelige FAST verdi estimatet og estimerte skalaen. Til slutt re-interpoleres koordinatene til oktavene som ligger nærmeste den bestemte skalaen.

2.5.2 Deskriptoren

BRISK-deskriptoren består av en binær deskriptor som sammenligner pikselverdien mellom egenskapspunktet og nabopikslene. Og resulterer i en binær streng. Det er en enkel men effektiv metode.

Detektoren har allerede funnet plassering og beste skala av egenskapspunktene. Deskriptoren starter med å sette et sirkulært samplingsmønster over hvert egenskapspunkt p_i . Samplingsmønsteret består av N antall punkter som blir plassert rundt egenskapspunktet. Figur 2.19 viser hvordan et slikt mønster vil se ut. Her er punktet i midten egenskapspunktet.



Figur 2.19 Viser sampling mønsteret med N=60 punkter og skala t=1. Bildet hentet fra [15]

For å jevne ut kantene benytter BRISK et gaussisk glattefilter som anti-aliasing filter. Det gaussiske glattefilteret blir satt over hver av de N punktene med standardavvik σ_i. Størrelsen av standardavviket vil være proporsjonal med avstanden mellom punktene. De røde stiplede linjene i figur 2.19 viser hvordan standardavviket er plassert i forhold til punktene.

Etter kantutjevningen blir det parvis søkt etter punkter som kan hjelpe med å beskrive egenskapspunktet på best mulig måte. Formelen $N \cdot \frac{(N-1)}{2}$ blir brukt for å finne disse parene. Parene blir brukt til å estimere den lokale gradienten, her vist ved formel 2.5.2. Den estimerte

lokale gradienten er her $\boldsymbol{g}(p_i, p_j)$. Intensiteten til punktene som danner samplings-paret er her definert som er $I(p_j, \sigma_j)$ og $I(p_i, \sigma_i)$.

$$\boldsymbol{g}(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \tag{2.5.2}$$

Når den lokale gradient har blitt estimert, blir parene plassert inn i en av to grupper basert avstanden mellom dem. Gruppene er kort avstand S og lang avstand L. Kravet for at de forskjellige parene skal havne i hvilken gruppe blir vist med formel 2.5.3 og 2.5.4. Her er $\delta_{max} = 9.95 \cdot t$ og $\delta_{min} = 13.67 \cdot t$ (t er her skalaen funnet av detektoren).

$$S = \|p_j - p_i\| < \delta_{max} \tag{2.5.3}$$

$$L = \|p_j - p_i\| > \delta_{min} \tag{2.5.4}$$

De to gruppene blir brukt til forskjellige ting. De lange avstandene blir brukt til å estimere retningen til egenskapspunktet. Mens de korte avstandene blir brukt til selve beskrivelsen av egenskapspunktet. Retningen blir estimert ved hjelp av formel 2.5.5.

$$\boldsymbol{g} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \frac{1}{L} \cdot \sum g(p_i, p_j) \tag{2.5.5}$$

For å lage BRISK-deskriptoren blir samplingsmønsteret rotert rundt egenskapspunktet med vinkelen $\alpha = 2 \cdot \tan^{-1}(g_y, g_x)$. Gruppen med de korte avstandene blir brukt til å lage bit-vektoren $\boldsymbol{d}_k$. Sammenligningen av intensiteten i de roterte parene (p_i^α, p_j^α) gjøres ved hjelp av valgtreet i formel 2.5.6.

$$b = \begin{cases} 1, I(p_j^\alpha, \sigma_j) > I(p_j^\alpha, \sigma_j) \\ 0, ellers \end{cases} \tag{2.5.6}$$

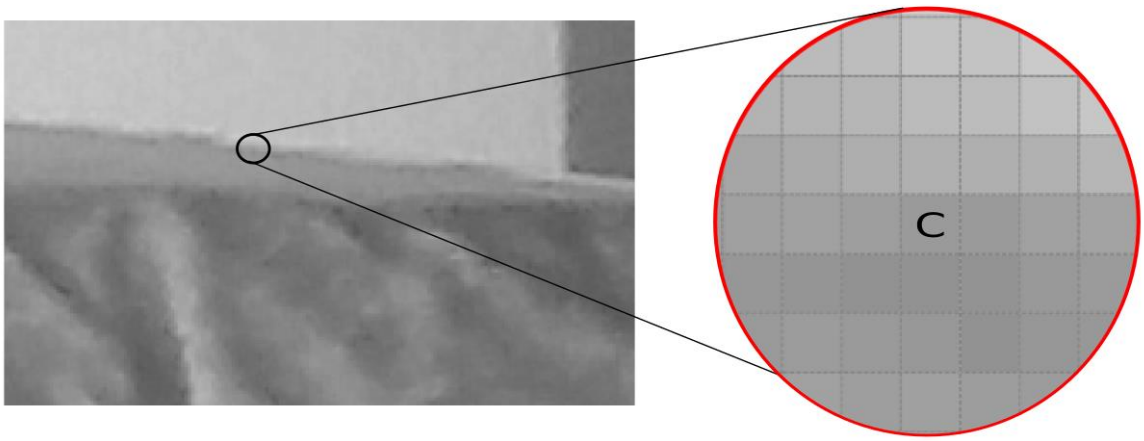
Ved hjelp av bit-vektoren $\boldsymbol{d}_k$ blir det konstruert en bit-streng som inneholder informasjon om intensitet og rotasjon til egenskapspunktene.

2.6 SUSAN

SUSAN (*Small Univalue Segment Assimilating Nucleus*) ble utviklet av S.M Smith og H.M Brady i 1995. Metoden kan brukes til å detektere endimensjonale (kanter) og todimensjonale (blant annet hjørner og kryss) egenskapspunkter, hvor grunnlaget for deteksjon er den samme for begge to. SUSAN er kjent for å være robust mot støy, ettersom den ikke benytter seg av noen form for den deriverte.

2.6.1 Detektoren

For å detektere egenskapspunkter blir en maske med bestemt radius forskjøvet over bildet. Masken sammenligner intensiteten av nukleonet (senterpikselen) mot nabopikslene innenfor det sirkulære område. Den mest anvendte masken har radius satt til $r = 3.4$ (dette gjør at masken vil bestå av 37 piksler). Figur 2.20 viser en enkel skisse av masken.



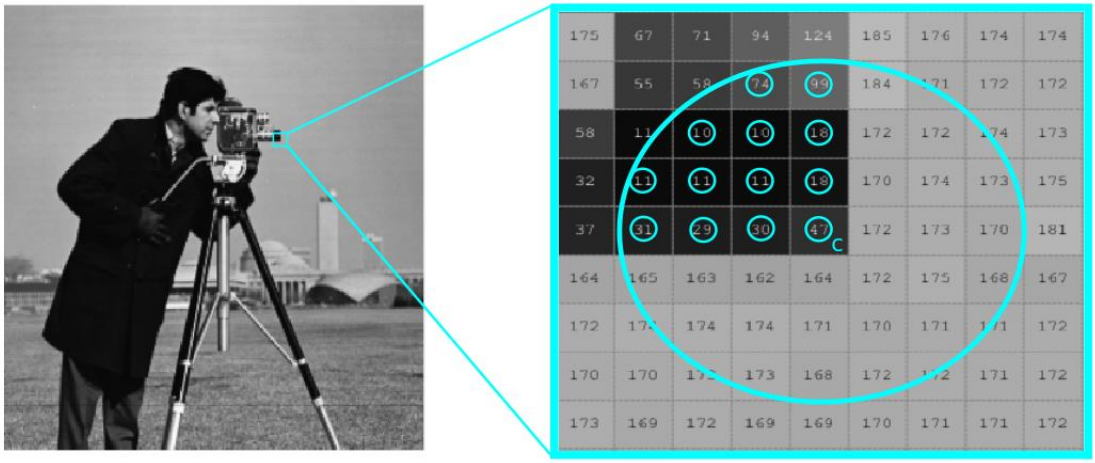
Figur 2.20 Enkel skisse av SUSAN-masken med radius 3.4 plassert i et tilfeldig punkt i bildet. Hvor C er nukleonet og røde streken rundt er yttergrensen.

For å avgjøre om nabopikselen har mørkere eller samme intensitet benytter metoden seg av et enkelt valgtre. Valgtreet er her vist ved formel 2.6.1. En tersklingsverdi t blir brukt for å bestemme om pikselen er mørkere eller lysere enn nukleonet. Hvis differansen mellom pikselen som blir testet og nukleonet er mindre eller lik valgt terskelverdi, blir pikselen godkjent som en

del av bildeobjektet. Alle pikslene innenfor masken som blir godkjent blir kalt USAN ('*Univalue Segment Assimilating Nucleus*').

$$c(\boldsymbol{r}, \boldsymbol{r}_0) = \begin{cases} 1, hvis\ |I(\boldsymbol{r}) - I(\boldsymbol{r}_0)| \leq t \\ 0, hvis\ |I(\boldsymbol{r}) - I(\boldsymbol{r}_0)| > t \end{cases} \tag{2.6.1}$$

Funksjonen $c(\boldsymbol{r}, \boldsymbol{r}_0)$ forteller om punktet som blir testet er godkjent som USAN eller ikke. Her er $\boldsymbol{r}_0$ er posisjonen til nukleonet og $\boldsymbol{r}$ posisjonen til nabopikselen som blir testet. Intensiteten til punktene blir betegnet med henholdsvis $I(\boldsymbol{r})$ og $I(\boldsymbol{r}_0)$ for nukleonet.



Figur 2.21 Viser SUSAN-masken på et ekte bilde med terskelverdien t satt til 47. Piksler godkjent som en del av USAN-området er markert med sirkler, c er senterpiksel (nukleonet).

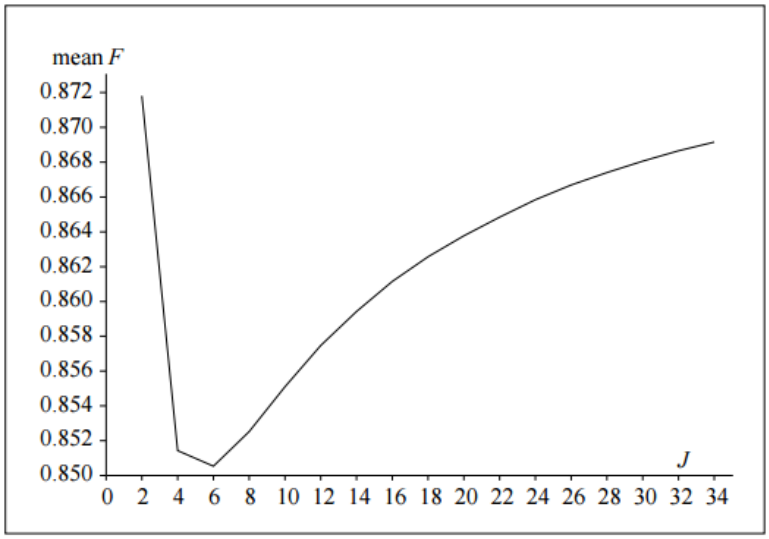
Terskelverdien bestemmer kravet for at pikselen skal blir godkjent. Jo høyere terskelverdien er, desto større må differansen være for at pikselen skal bli godkjent som USAN. Dette fører til mer nøyaktige og skarpere egenskaper blir detektert. Terskelverdien er også med på å avgjøre hvor mye støy som vil bli ignorert av detektoren.

$$F(d, t, s) = \frac{\sqrt{var(R_s)} + \sqrt{var(R_N)}}{< R_s > - < R_s >} \tag{2.6.2}$$

Det er ønskelig å oppfylle kriteriet om å ha et minimum med antall falsk positive og falsk negative egenskaper funnet. Dette kriteriet er her gitt med formel 2.6.2. Her er F proporsjonal med antall falsk positive og falsk negative, s er standardavviket til bildestøyen. R_s representerer kantresponsen ved ingen kant til stede, mens R_n er kantresponsen når masken er plassert over en kant med styrke d.

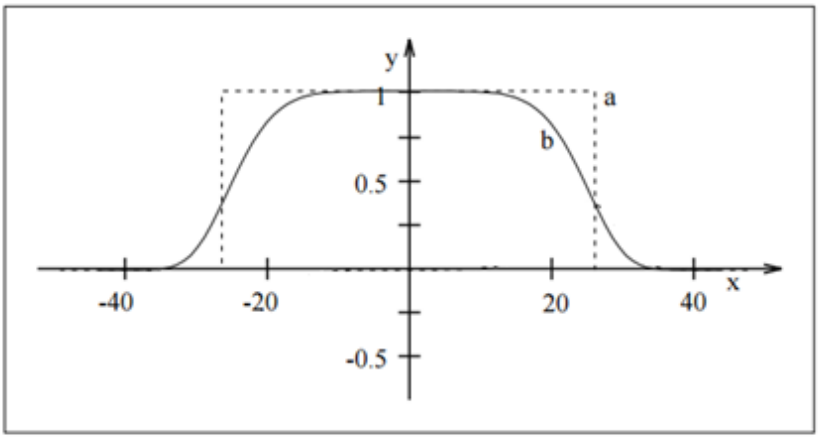
$$c(\boldsymbol{r}, \boldsymbol{r}_0) = e^{-\left(\frac{I(\boldsymbol{r})-I(\boldsymbol{r}_0)}{t}\right)^J} \tag{2.6.3}$$

For å oppfylle kriteriet over blir formel 2.6.1 byttet ut med formel 2.6.3. Ved å gjøre denne forandringen blir kravet oppfylt samt det gir mulighet for at intensiteten kan variere noe uten at det vil påvirke $c(\boldsymbol{r}, \boldsymbol{r}_0)$ i særlig stor grad (selv om intensiteten er nærme terskelverdien). Verdien J blir funnet ved å evaluere forskjellige J verdier opp mot formel 2.6.3. Utledning og evaluering kan bli funnet i artikkel [9]. Figur 2.22 viser grafisk forholdet mellom F og J.



Figur 2.22 Viser forholdet mellom F og J. Hentet fra [9]

Ved figur 2.23 blir det vist at det vil være lavest antall falsk negative og falsk positive når $J = 6$. Dermed blir denne verdien av J brukt i SUSAN-metoden for å gi en best mulig optimalisering. Opphøyningen i sjette gir i tillegg en god balanse om tersklingsverdien og den opprinnelige funksjon (2.6.1).



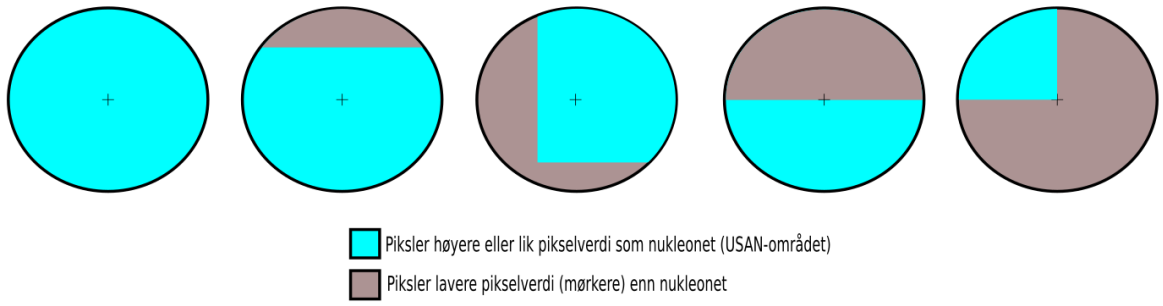
Figur 2.23 Viser den formel 3.1 (a) sammenlignet med 3.2 (b). Her er t satt til ± 27.Hvor y-aksen er funksjonsverdien (ingen benevning) og x-aksen (gråskala verdi) er forskjell i intensitet til pikslene. Bildet hentet fra [9]

Figur 2.23 viser forskjellen mellom formel 2.6.1 (vist ved graf a) og 2.6.3 (vist ved graf b). For graf a må intensitetsforskjellen være i intervallet $-27 \leq I \leq 27$ (samme som terskelverdien) for at pikselen skal bli godkjent som USAN. Grafen for b viser en mer avrundet form, som gjør at et punkt kan bli godkjent som USAN selv om intensitetsforskjellen kan avvike noe fra terskelverdien. Formel 2.6.3 er mer stabil enn sin motpart.

Hvor stort USAN-området i hver maske er avhenger hvor mange godkjente USAN piksler masken inneholder. For å beregne USAN-området blir alle pikslene som har blitt godkjente bli plusset sammen. Formel 2.6.4 viser utregningen USAN-området i masken.

$$n(\boldsymbol{r}_0) = \sum_{\boldsymbol{r}} c(\boldsymbol{r}, \boldsymbol{r}_0) \tag{2.6.4}$$

Her tilsvarer n total antall piksler i masken som er godkjente som USAN. Senere blir denne verdien sammenlignet med en fiksert geometrisk terskel g. Hva denne terskelen er avhenger om det er kanter eller hjørner som skal detekteres.



Figur 2.24 SUSAN-masken med fem forskjellige USAN-områder

Figur 2.24 viser hvordan USAN-området innenfor en maske forandrer seg etter som hvor på bilde masken befinner seg. Som tidligere representerer krysset nukleonet (senterpiksel) og sirkelen rundt er yttergrensen til masken. Her vil maske a være den som har det største USAN-området (også det maksimale en maske kan ha, alle pikslene innenfor masken har lik intensitet som nukleonet). Maskene b, c og d vil ha et mindre USAN-område enn maske a ettersom det er flere piksler som har ulik intensitet. Maske e vil være den masken med minst USAN-område. Denne vil bli kategorisert som et hjørne.

2.6.2 Kantdeteksjon

For å detektere kanter blir formel 2.6.5 brukt til å finne kantresponsen. Desto mindre USAN-området er jo sterkere vil kantresponsen være.

$$R(\boldsymbol{r}_0) = \begin{cases} g - n(\boldsymbol{r}_0), & hvis\ n(\boldsymbol{r}_0) < g \\ 0, & ellers \end{cases} \tag{2.6.5}$$

Her er g den geometriske terskelen som er satt til $\frac{3n_{max}}{4}$ ved deteksjon av kanter. Dette for å gi en så optimal støyreduksjon som mulig. Denne verdien er beregnet ved å analysere forventningsverdien til kantresponsen når det kun er støy til stede. n_{max} er antall piksler det maksimalt er i masken.

Når kantresponsen har blitt beregnet blir kanten sin retning funnet. I stedet for å regne ut bilde deriverte som mange andre metoder gjør, har det blitt laget en metode som finner retningen til kanten ved hjelp av USAN-området. Metoden går ut på å assosiere retningen til en kant med et billedpunkt hvor kantresponsen ikke er lik null. Retningen blir funnet ved å analysere USAN-

området. Analyseringen kan resultere i en av to tilfeller. Disse tilfellene kalles for inter-piksel og intra-piksel.

Intra-piksel tilfellet blir brukt hvis USAN-området (i piksler) er mindre enn diameteren til masken (i piksler). Hvis USAN-området er større enn diameteren, blir senterpunktet av tyngden til USAN-området funnet ved hjelp av inter-piksel. Unntaket er hvis senterpunktet til USAN-områdets tyngde ligger mindre enn en piksel unna nukleonet (senterpunktet til masken). Da vil intra-piksel kunne gi et bedre resultat.

Med inter-piksel vil vektoren av senterpunktet av tyngden $\boldsymbol{r}$ og nukleonet stå normalt på den lokale kantretningen. Tyngdepunktet beregnes i det tilfellet med formel 2.6.6.

$$\bar{\boldsymbol{r}}(\boldsymbol{r}_0) = \frac{\sum_{\boldsymbol{r}} \boldsymbol{r}\, c(\boldsymbol{r}, \boldsymbol{r}_0)}{\sum_{\boldsymbol{r}} c(\boldsymbol{r}, \boldsymbol{r}_0)} \tag{2.6.6}$$

Ved intra-piksel er USAN formet som en tynn linje i den retningen kanten går. Kantretningen kan da bli funnet ved å beregne den lengste symmetriaksen. Dette blir gjort ved å forta summasjonene i formel 2.6.7, 2.6.8 og 2.6.9

$$\overline{(x - x_0)^2}(\boldsymbol{r}_0) = \sum_{\boldsymbol{r}} (x - x_0)^2 c(\boldsymbol{r}, \boldsymbol{r}_0) \tag{2.6.7}$$

$$\overline{(y - y_0)^2}(\boldsymbol{r}_0) = \sum_{\boldsymbol{r}} (y - y_0)^2 c(\boldsymbol{r}, \boldsymbol{r}_0) \tag{2.6.8}$$

$$\overline{(x - x_0)(y - y_0)}(\boldsymbol{r}_0) = \sum_{\boldsymbol{r}} (x - x_0)(y - y_0) c(\boldsymbol{r}, \boldsymbol{r}_0) \tag{2.6.9}$$

Forholdet mellom $\overline{(x - x_0)^2}$ og $\overline{(y - y_0)^2}$ blir brukt til å bestemme orienteringen til kanten. Fortegnet til $\overline{(x - x_0)(y - y_0)}$ blir brukt til å bestemme om en diagonal kanten har en positiv eller negativ gradient.

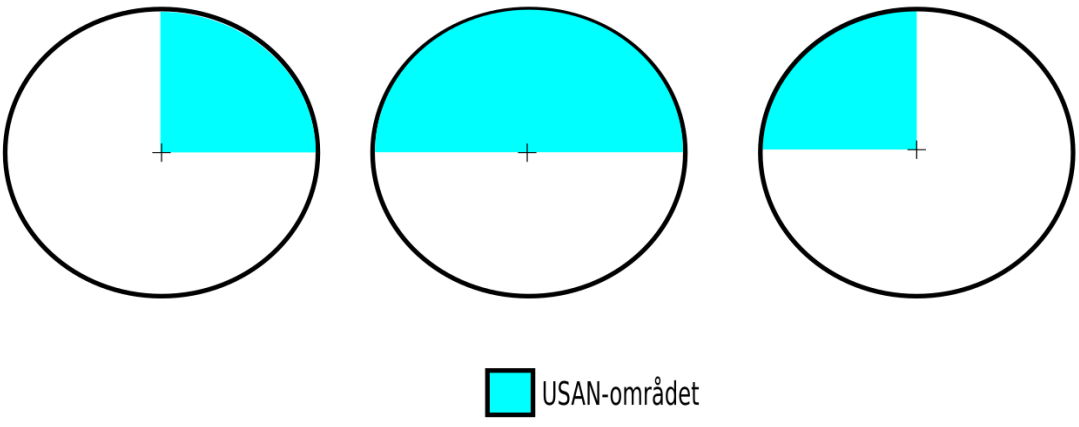
Når kantresponsen og kantretningen har blitt funnet, benyttes NMS på ikke maksima punkter som står vinkelrett på selve kanten. Dette for at de ikke skal blir definert som kant. Det har i tillegg blitt implementert et regelverk som gjør de binære kantene/strekene tynnere. Prosessen gjør at de detekterte kantene følger kravet om antall nabo piksler som skal være koblet sammen.

Dette gjør at resterende støy blir fjernet, i tillegg kan kantpiksler fjernet ved en feiltagelse ved NSM bli restaurert.

2.6.3 Hjørnedeteksjon

Deteksjon av hjørner deler flere av de samme stegene som deteksjon av kanter. Den samme sirkulære masken blir benyttet (figur 2.20). Pikslene innenfor masken blir sammenlignet med senterpunktet (nukleonet) ved hjelp formel 2.6.3. Akkurat som ved kantdeteksjon blir summen av sammenligningen beregnet med formel 2.6.4. Hovedforskjellen mellom hjørne- og kantdeteksjon er ved utregning av kantresponsen. Hvis nukleonet ligger på et hjørne, så vil USAN-området være mindre enn halvparten av området masken dekker, og et lokalt minimum. Ved hjørnedeteksjon blir geometriske terskelen satt til $\frac{n_{max}}{2}$ (vist her ved formel 2.6.10). Terskelen kan trygt settes til halvparten av n_{max} ettersom en rett kant vil alltid dekke mer enn halvparten av USAN-området.

$$R(\boldsymbol{r}_0) = \begin{cases} \frac{n_{max}}{2} - n(\boldsymbol{r}_0), & hvis\ n(\boldsymbol{r}_0) < g \\ 0, & ellers \end{cases} \qquad (2.6.10)$$



Figur 2.25 Viser forskjellige masker som har et USAN-område mindre eller lik $\frac{n_{max}}{2}$ som resulterer i godkjent hjørne

For å redusere antall falsk positiv deteksjon (punkter som ikke er hjørner blir definert som hjørne) forårsaket av støy eller et uskarpt område kommer mellom to bilderegioner. Har det blitt utviklet to metoder.

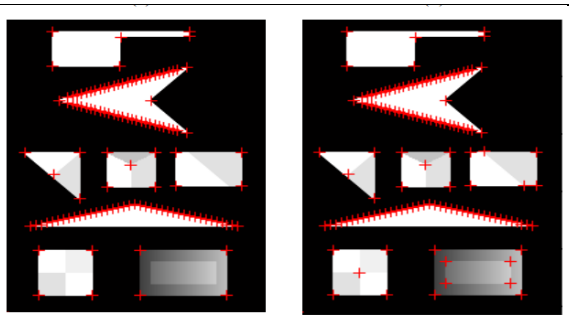
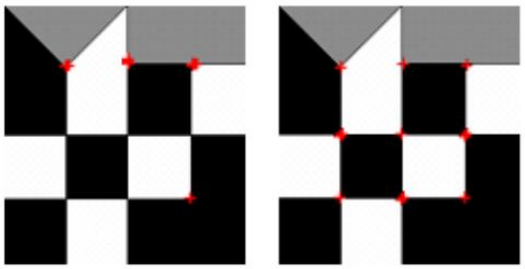
1. Første metoden går ut på å finne senter av tyngden til USAN-området. For så finne avstanden mellom dette punktet og nukleonet. Dersom det er lang avstanden mellom de to punktene vil det si at USAN-området tilsvarer et ekte hjørne. En kort avstand vil tilsvare en tynn linje som går gjennom nukleonet. Som kan bli definert som falsk negative, derfor sett bort ifra.

2. Andre metoden er en enkel regel som tvinger kontinuitet i USAN-områdene. For at et hjørne skal bli detektert må alle pikslene i masken, ligge på en rett linje og peke bort (utover) fra nukleonet mot senteret til tyngden må være en del av USAN. Dette gjør at USAN tvinges til å ha en form for homogenitet og reduserer antall falsk positive.

Siste steget er å søke etter lokale maksima fra den første kantresponsen i et område på 5x5 piksler for å finne hjørner.

2.6.4 Forbedringer av SUSAN-metoden

Siden SUSAN-metoden ble publisert i 1995, har det blitt utviklet flere metoder for å forbedre den originale metoden. Under vises en tabell med noen av modifiserte metodene.

Utvikler	Modifisering	Resultat
Yang Xingfang, Huang Yumei og Li Yan [25]	Ved hjelp av histogram, blir gjennomsnittet av intensiteten i nukleonet funnet. Og ved hjelp av det bestemme terskelen t automatisk (i den originale metoden må terskelen bestemmes manuelt)	 Bildet over viser SUSAN med $t = 15$ (venstre) og den modifiserte (høyre). Den modifiserte metoden resulterer i litt flere kanter funnet.
Hou Minglian g og Xing Shubin [26]	I stedet for å beregne USAN-området ved hjelp nabopikslene til nukleonet. Blir pikslene som er i kontakt med nukleonet benyttet.	 Hjørner detektert av originale SUSAN (venstre) og modifiserte (høyre). Detektere flere hjørner enn originale SUSAN, samtidig som den har lavere antall falsk positive.
Jie Zhao, Haichao Ma og Guozun Men [27]	Ved å ta i bruk 'lifting wavelet transform', kan den tilnærmede posisjonen til hjørnene bli funnet raskt ved å finne	Resultatet viste til raskere beregningshastighet og forbedret sanntid. Bilde sammenligning ble også forbedret.

	områder med høy frekvens. Dette øker beregningshastighete n	
Byung-Seung Jeon, Dong-Gi Woo, Young-Hak Mo og Myo-Teg Lim. [28]	Metoden ser på ekstremverdier innenfor masken for å avgjøre om det er et hjørne eller ikke.	 Modifiserte SUSAN (høyre) viser flere egenskaper funnet i forhold til originale SUSAN (venstre). Fra resultatet kommer det frem at den modifiserte metoden er mer nøyaktig og raskere beregningsmessig.

3 Testing av egenskapsdetektorene

I dette kapittelet blir funksjonene laget i MATLAB og eksperimentene beskrevet. I tillegg blir bildene brukt i eksperimentene presentert, samt hva som blir definert som et godkjent egenskapspunkt for de tre detektorene.

3.1 Verktøy

Alle eksperimentene blir utført i MATLAB® R2015a på en bærbar datamaskin med Intel® Core™ i7-6700HQ prosessor på 2.60 GHz, og 8 GB RAM.

3.1.1 MATLAB-funksjoner

I *Image Processing Toolbox*™ som er inkludert i MATLAB® finnes det allerede eksisterende funksjoner for BRISK og SURF, disse er benyttet i denne oppgaven. Funksjonene er:

detectBRISKFeatures:

Funksjon tar gråskala bilde som inngang og returnerer objekt med informasjon om de detekterte BRISK egenskapene som er funnet.

detectSURFFeatures:

Funksjon tar gråskala bilde som inngang og returnerer objekt med informasjon om de detekterte SURF egenskapene som er funnet.

3.1.2 Egen produsert kode

SUSAN.m:

Egen implementering av SUSAN hjørne-detektor. Laget ut ifra teorien beskrevet i delkapittel 2.6 med modifiseringer. I dette programmet blir USAN definert ut ifra formel 2.6.1. Modifiseringene er to geometriske terskelverdier og to intensitet terskelverdier, i stedet for en av hver. Dette har blitt gjort for at detektorene skal skille mellom kanter og hjørner. Plotter egenskapene funnet på inngangsbildet og viser brukt tid.

SURF.m:

Programmet bruker SURF funksjonen i MATLAB forklart ovenfor i delkapittel 3.1.1. Plotter egenskapene funnet på inngangsbildet og viser brukt tid.

BRISK.m:

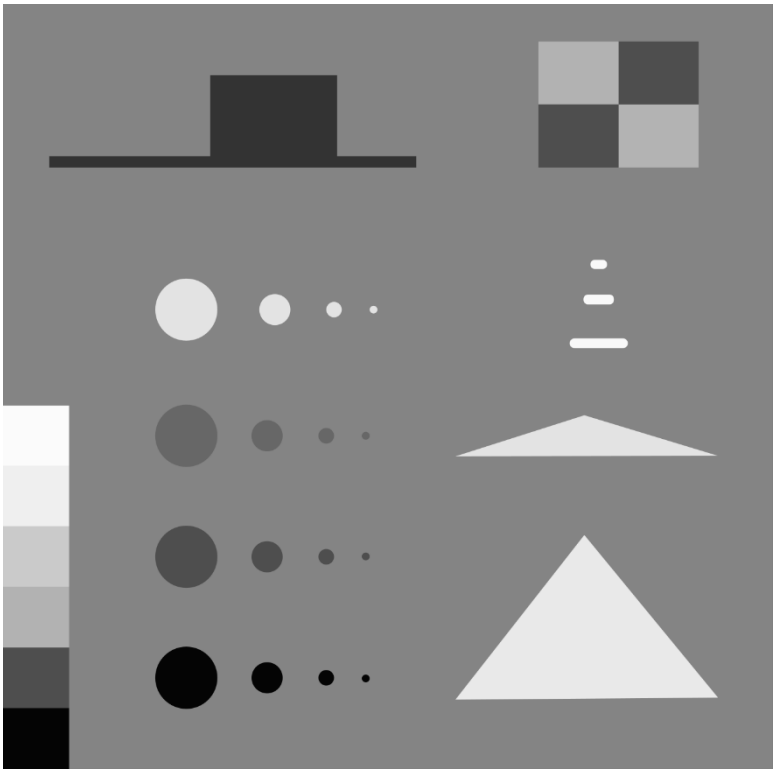
Programmet bruker BRISK funksjonen i MATLAB forklart ovenfor i delkapittel 3.1.1. Plotter egenskapene funnet på inngangsbildet og viser brukt tid.

testSUSAN.m

Dette programmet inneholder funksjoner for å kalle på SUSAN.m og plotte egenskapene funnet. Den leser i tillegg inn bilder og konverterer til gråskala. Filstien til bildene må forandres manuelt.

3.2 Bilder brukt for å analysere egenskapsdetektorene

I denne oppgaven blir det benyttet to bilder for å analysere hvor robuste egenskapsdetektorene er. Det første bildet har blitt laget i Inkscape™ som er et gratis vektor-tegneprogram med åpen kildekode. Bildet består av enkle 2D geometriske figurer for enkel visuell bekreftelse av egenskapspunkter. Det har blitt laget figurer som representerer egenskaper som de utvalgte egenskapsdetektorene skal lokalisere. Dette bildet vil bli brukt i de fire første eksperimentene. Figur 3.1 viser bilde 1.



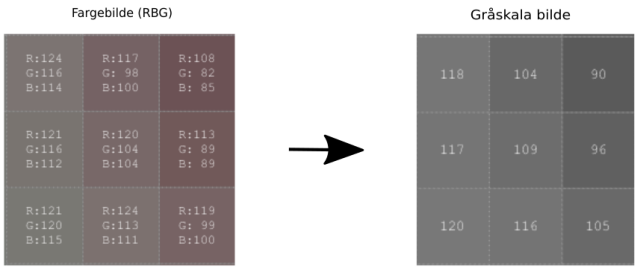
Figur.3.1 viser bilde en brukt for å analysere egenskapsdetektorene.

Det andre bildet som blir brukt for å analysere detektorene har blitt tatt av forfatteren. Bildet er tatt i Stavanger sentrum og inneholder muligheten for sub-sampling (bildet består av objekter som er nærme og langt unna). I denne oppgaven blir bildet benyttet for å teste hvor bra detektorene er til å lokalisere egenskapspunkter ved forskjellig skala. Kameraet brukt er av typen FujiFilm FINEPIX HS10, figur 3.2 viser bilde 2.

Figur.3.2 viser bilde to som er tatt i Stavanger sentrum. Bildet inneholder objekter som bør gjøre det mulig for alle tre detektorene å lokalisere ønskede egenskaper.

3.3 Konvertere fra fargebilde til gråskala bilde

For å kunne benytte valgte metoder må bildene konverteres fra fargebilde til gråskala bilde. Dette gjøres ved å bruke den innebygde MATLAB funksjonen rgb2gray. Her blir fargetone og metning fjernet mens lysstyrken beholdes. Figur 3.3 viser et område på $3x3$ piksler før og etter konverteringen til gråskala.



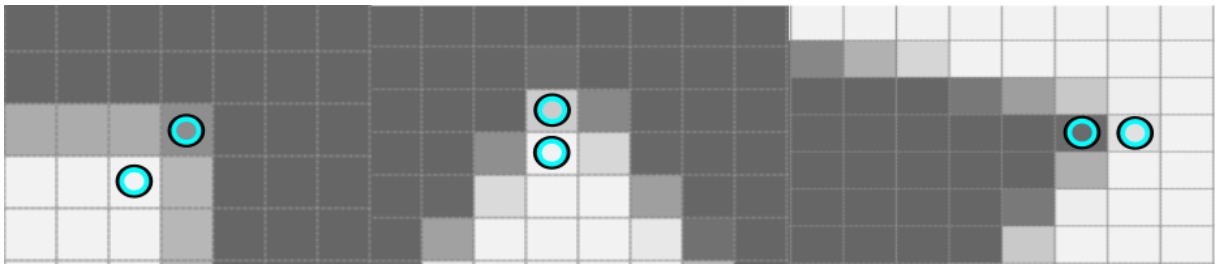
Figur 3.3 viser et område på 3x3 piksler før og etter konverteringen til gråskala. Området er hentet fra bilde 2 beskrevet i delkapittelet over.

3.4 Avgjøre hva som er et godkjent egenskapspunkt

For å kunne avgjøre hvor bra de utvalgte detektorene er på å lokalisere egenskapspunkter, blir det satt betingelser på hva som er et godkjent egenskapspunkt. Dette blir gjort for at den visuelle inspeksjonen skal være mest mulig konsekvent.

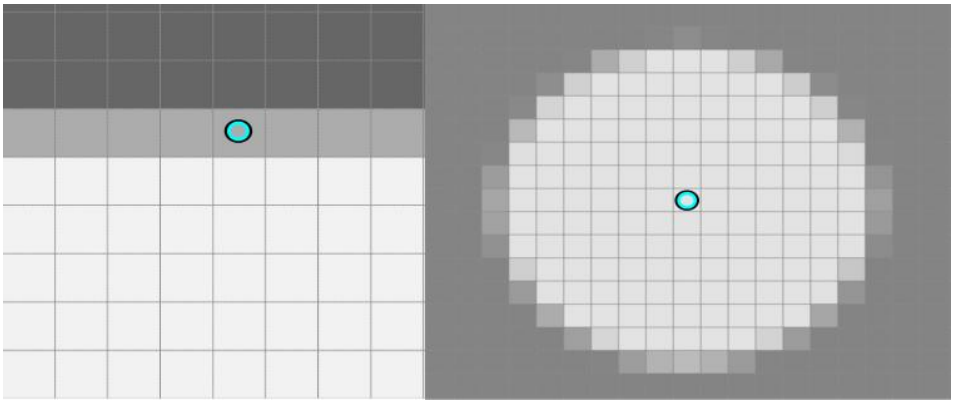
3.4.1 Godkjent egenskapspunkt for BRISK og SUSAN

Ettersom både SUSAN og BRISK skal lokalisere hjørner, vil de samme betingelsene gjelde for om et egenskapspunkt er godkjent eller ikke for de to detektorene. Et hjørne defineres som innsiden eller yttersiden av to linjer som møtes i vinkel. Figur 3.4 viser tre godkjente hjørner. Hvis flere punkter er i nærheten av hjørnet vil de bli sett på som et godkjent egenskapspunkt.



Figur 3.4 viser tre hjørner som defineres som godkjente hjørner. Hvert eksempel viser to godkjente hjørner på grunn av nyanse overgangen. De to første fra venstre er på innsiden, mens siste er både innside og ytterside.

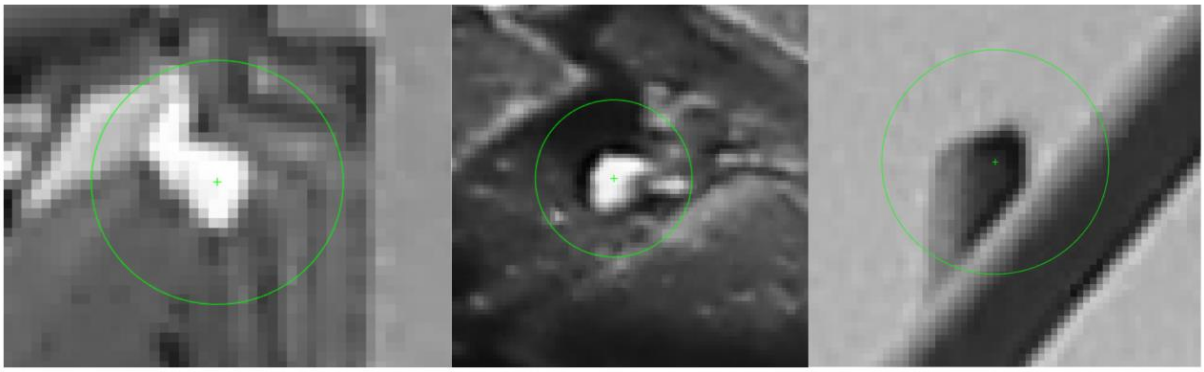
Detekterte punkter som ikke godkjennes som egenskapspunkter vises ved noen eksempler på figur 3.5.



Figur 3.5 viser to eksempler på ikke godkjente egenskapspunkter for SUSAN og BRISK. Fra venstre vises detektert punkt langs en kant, andre viser et punkt i senter av et objekt.

3.4.2 Godkjent egenskapspunkt for SURF

I motsetning til SUSAN og BRISK som skal lokalisere hjørner, skal SURF lokalisere områder som er lysere eller mørkere enn bakgrunnen. Dette gjør det vanskeligere å si om et punkt er godkjent eller ikke. Ettersom områdene ikke har noen fast struktur eller størrelse. Figur 3.6 viser eksempler på tre godkjente strukturerer som SURF-detektoren har lokalisert.



Figur 3.6 viser tre godkjente punkter detektert av SURF-detektoren.

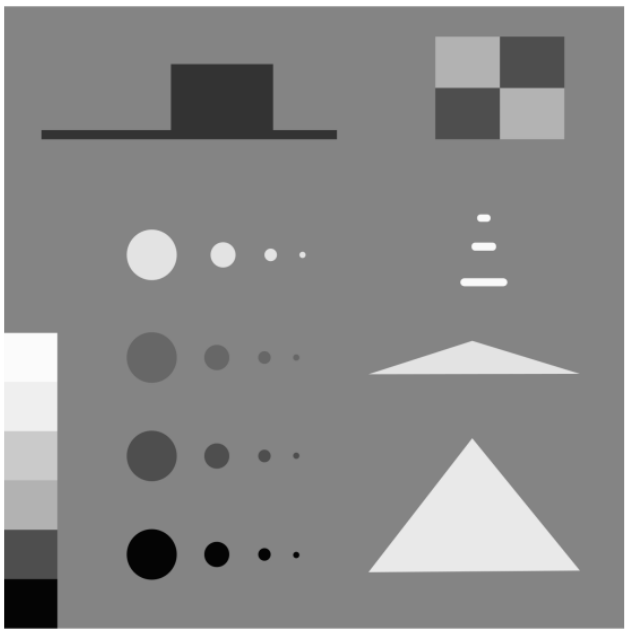
3.5 Analyse av egenskapsdetektorene

Det vil bli utført fem forskjellige eksperimenter for å teste robustheten til detektorene. De fem eksperimentene er:

- Bilde 1 uten geometriske eller fotometriske forstyrrelser
- Endring av intensitet i bilde 1
- Støy i bilde 1
- Rotere bilde 1
- Endring av skala i bilde 2

3.5.1 Bilde 1 uten geometriske eller fotometriske forstyrrelser

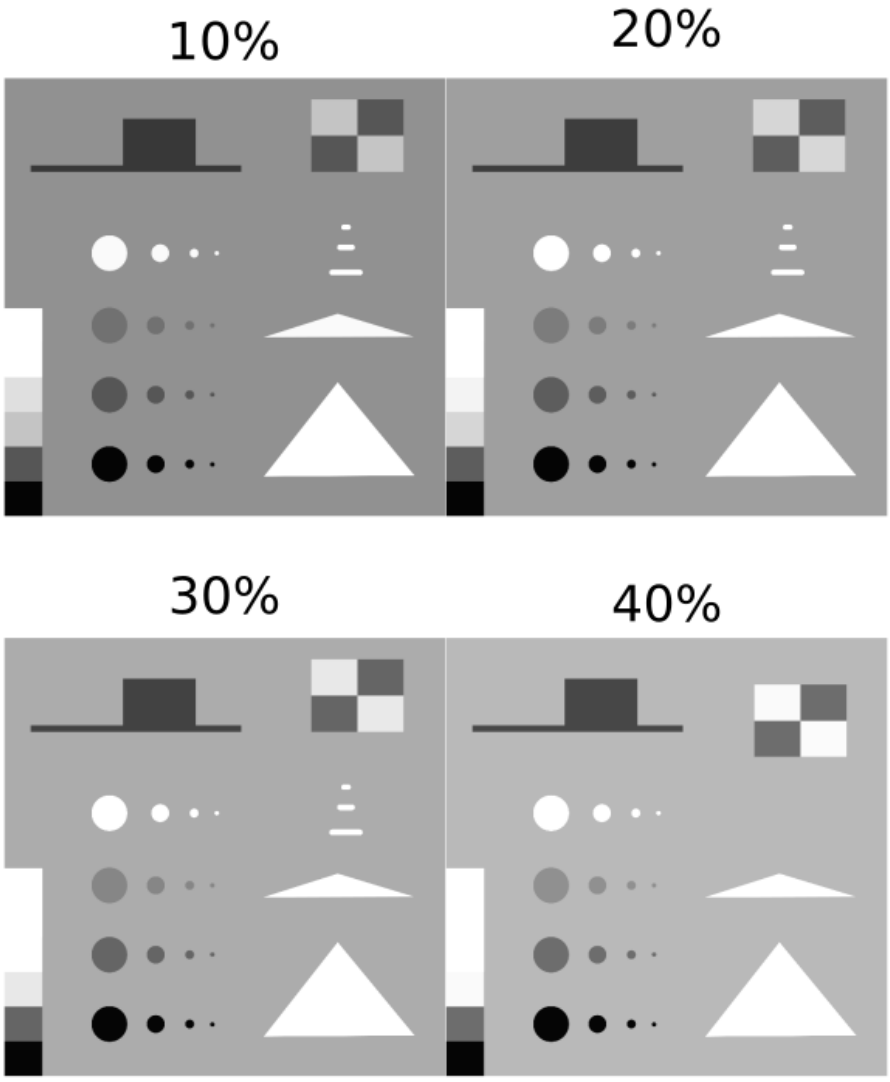
I dette eksperimentet vil detektorene bli testet på et bilde med enkle figurer som har tydelig definerte kanter. Grunnlaget for eksperimentet er å se at detektorene opptrer som de skal og at resultatet samstemmer med hva som er forventet. Bildet blir laget i Inkscape ved å lage trigonometriske figurer med kjent gråverdi mot en bakgrunn av kjent gråverdi. Bildet brukt i dette eksperimentet vises med figur 3.7.



Figur 3.7 viser bildet brukt i dette eksperimentet.

3.5.2 Endring av intensitet i bilde 1

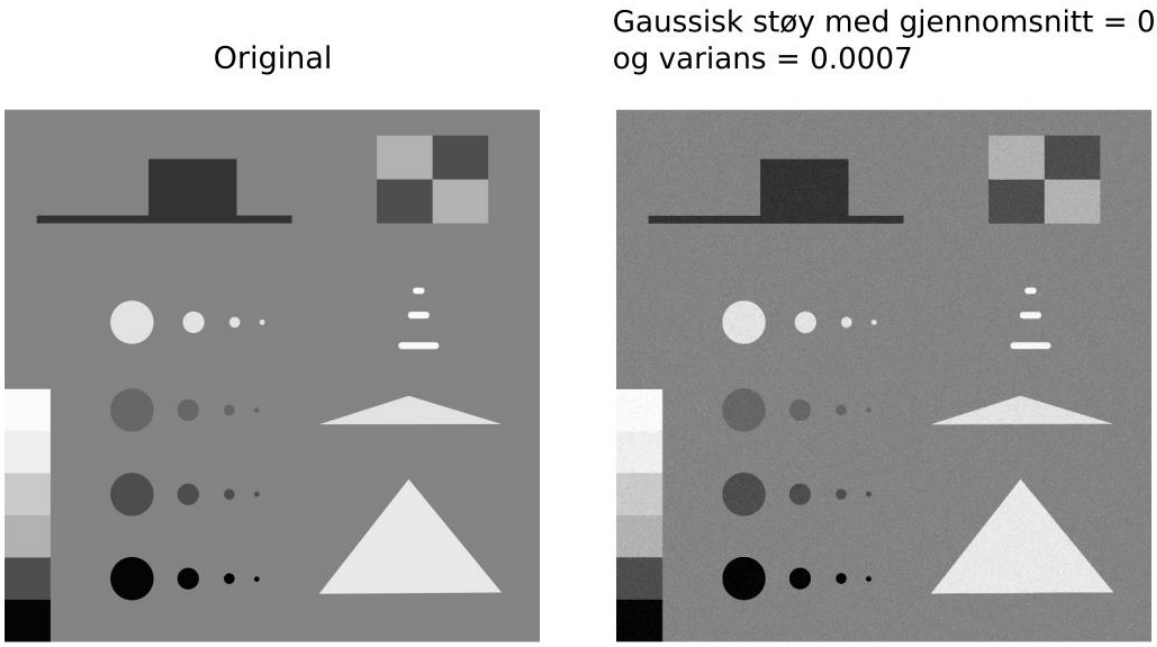
En av de fotometriske egenskapene detektorene blir testet med er intensitet forandringer. I dette forsøket vil bildet brukt i forrige forsøk bli benyttet, for så øke intensiteten i bildet gradvis. Punktene funnet i forrige eksperiment vil bli brukt som en fasit ved å sammenligne de med ny punkter oppdaget ved forandringen. Intensiteten vil øke med 5% fra 0%-40%. Figur 3.8 viser hvordan bildet ser ut når intensiteten blir forandret med 10% for hvert bilde.



Figur 3.8 Viser bildet med en intensitet forandring på 10%, 20%, 30% og 40%.

3.5.3 Støy i bilde 1

I dette eksperimentet vil det bli lagt til hvitt gaussisk støy bildet fra eksperiment i delkapittel 3.5.1. Hvitt gaussisk støy er statistisk støy som har en normal fordeling. Støyen blir økt gradvis med 0.0002 hver deteksjon. Figur 3.9 viser original bildet og bildet påvirket av gaussisk støy.

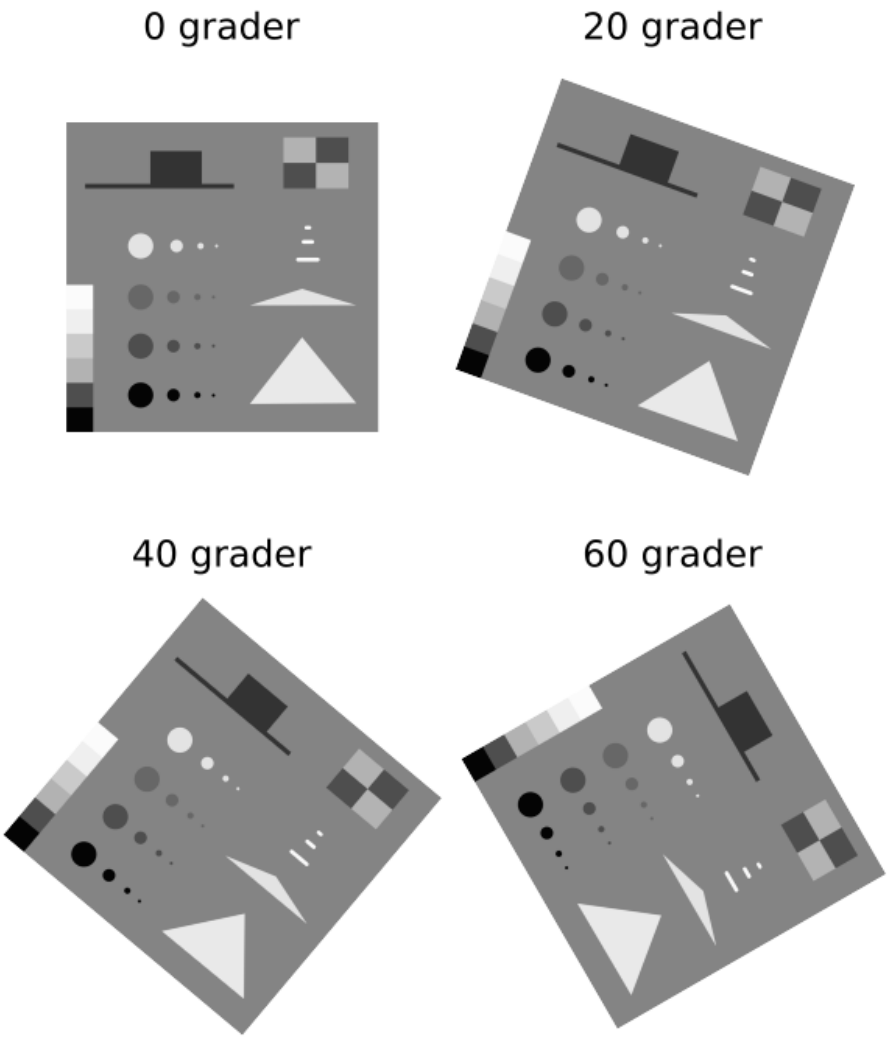


Figur 3.9 Viser original bildet (venstre) og bildet med gaussisk støy med gjennomsnitt satt til 0 og varians satt til 0.01 (høyre).

3.5.4 Rotere bilde 1

Rotasjon er en av de geometriske påvirkningene som et egenskapspunkt kan bli utsatt for. I dette eksperimentet blir bildet og resultatet fra eksperiment i delkapittel 3.5.1 brukt. Resultatet blir brukt som en fasit.

I dette eksperimentet blir hele bildet rotert med en fast vinkel som er mellom 0-90 grader. Med en økning på 10 grader mellom hver omgang egenskapspunktene blir funnet. Ettersom hele bildet blir rotert vil alle egenskaper i bildet også bli rotert med lik vinkel. Figur 3.10 viser bildet rotert 0, 20, 40 og 60 grader.



Figur 3.10 viser bildet rotert 0, 20, 40 og 60 grader.

3.5.5 Endring av skala i bilde 2

Endring i skala blir oppnådd ved å forandre fokallengden til kameraet [16], i stedet for å forandre fokallengden blir et område i bildet forstørret. Området som blir forstørret velges ut ifra hvor detektorene allerede har funnet godkjente egenskapspunkt eller egenskapspunkter, men også hvor det skulle være mulig å oppdage flere egenskapspunkter. Dette for å se om detektorene klarer å finne samme eller flere egenskapspunkt på forskjellig skala. Figur 3.11 viser bildet bilde før og etter forstørring.

Figur 3.11 viser bilde 2 før og etter en forstørrelse på 4 ganger.

4 Resultat og Diskusjon

Resultatet av eksperimentene vil bli lagt frem her samt diskusjonene rundt disse. SUSAN blir kjørt med terskelverdier som har blitt testet for å gi best mulig resultat. Viktigste er å finne terskelverdier som eliminerer deteksjon av kanter og fremhever hjørner. Det blir benyttet de samme terskelverdiene i de fire første eksperimentene, og for det siste eksperimentet blir en ny passende terskelverdi funnet.

- For de fire første eksperimente20ne blir disse terskelverdiene brukt: $geoTerskel = 14$, $geoTerskel = 21, interskel = 0.0005\ og\ interskel1 = 0.0007$
- For det siste eksperimentet blir disse terskelverdiene brukt: $geoTerskel = 10$, $geoTerskel2 = 14, interskel = 0.18\ og\ interskel1 = 0.0001$

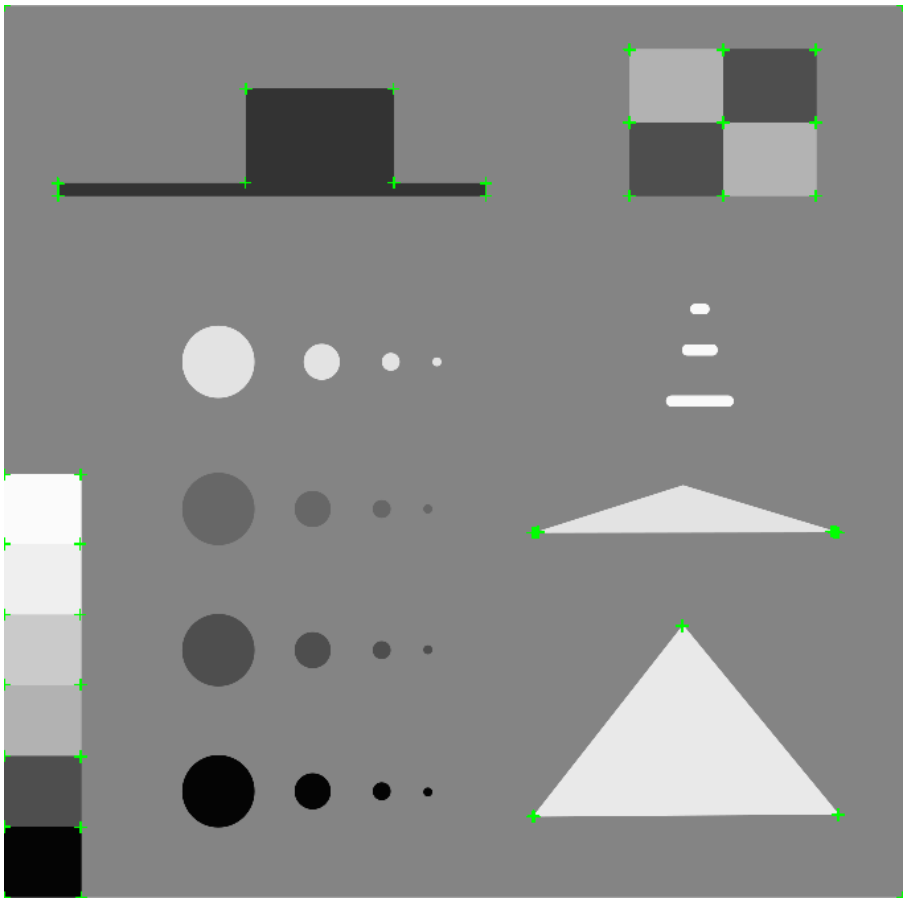
Ved bruk av andre terskelverdier for forbedring av resultat, blir disse nevnt i resultatet for det aktuelle eksperimentet.

For BRISK og SURF har det ikke blitt gjort noen form for modifikasjoner på koden som allerede er implementert i MATLAB.

4.1 Resultat av bilde 1 uten geometriske eller fotometriske forstyrrelser

Under vises resultatet når detektorene har blitt testet på bilde 1 uten noen form for forstyrrelser. De grønne kryssene representerer lokaliserte egenskapspunkter. Og de grønne sirklene representerer skalaen BRISK og SURF har detektert egenskapspunktene på.

SUSAN-detektoren

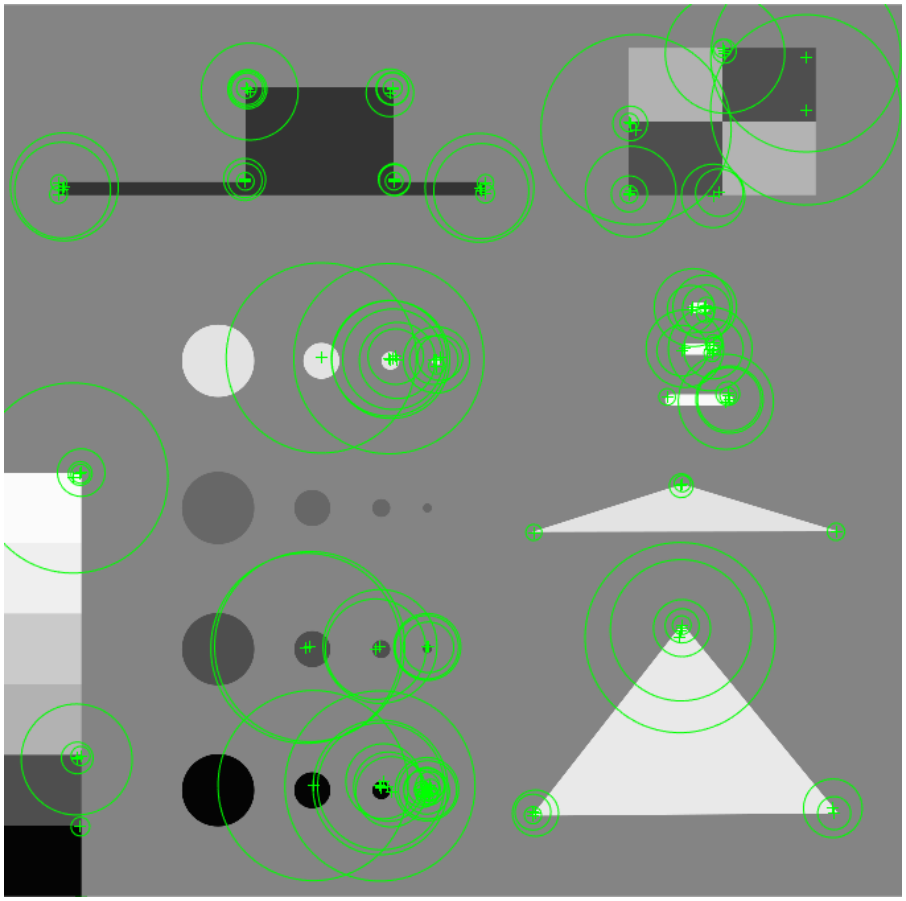


Figur 4.1 viser lokaliserte egenskapspunkter av SUSAN-detektoren.

Fra resultatet vises det at SUSAN-detektoren er veldig nøyaktig med de brukte terskelverdiene for geometrisk og intensitet beskrevet i starten av kapittelet. Figur 4.1 viser resultatet av SUSAN. Den har ikke feil detektert egenskapspunkter i noen av de geometriske figurene som ikke inneholder hjørner. Av alle hjørnene som er å finne, er det bare et hjørne som ikke blir lokalisert av SUSAN-detektoren. Hjørnet som ikke blir lokalisert er hjørnet med størst vinkel. Ved å forandre $geoTerskel1$ til 20 gjør det mulig å lokalisere det siste hjørnet, dessverre øker

antall feil deteksjoner samtidig. Resultatet viser også at detektoren er robust når det kommer til forandring i gråskalaverdi. Det vises ved at detektorene lokaliserer alle hjørnene i de seks kvadratene ned til venstre hjørne i bildet.

BRISK-detektoren



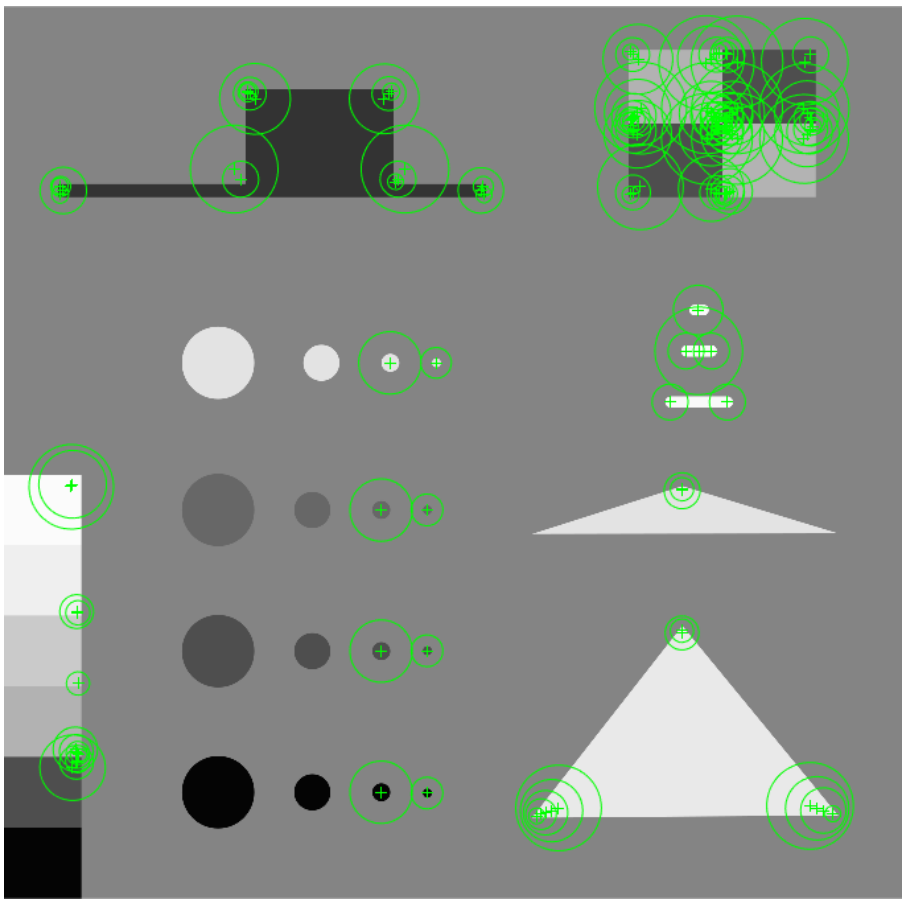
Figur 4.2 viser lokaliserte egenskapspunkter for BRISK- detektoren.

Resultatet for BRISK-detektoren blir her vist ved figur 4.2. Ut i fra resultatet kan det bli sett at BRISK-detektoren ikke er like nøyaktig som SUSAN. Det har blitt lokalisert flere egenskapspunkter i geometriske figurer som ikke inneholder hjørner. Totalt har 56 punkter som ikke er i nærheten av et hjørne, blitt detektert som et hjørne. Grunnen til dette kan være fordi BRISK-detektoren leter etter egenskapspunkter i forskjellig skala ved hjelp av 9-16 masken. De fleste punktene som har blitt lokalisert som egenskapspunkt og som faktisk er et hjørne, er funnet på lav skala. I forhold til de 56 punktene som ikke er hjørner, hvor de fleste er lokalisert på høy skala.

Positive med BRISK-detektoren er at den lokaliserer hjørner med flere forskjellige vinkler. Her har hjørner med vinkler mindre enn 90°, 90° og større enn 90° blitt detektert. Dessverre gjør den ikke like bra hvis hjørnet har tilnærmet lik intensitet som området rundt.

SURF-detektoren

SURF-detektorene skal lokalisere områder som er mørkere eller lysere enn bakgrunnen. Figur 4.3 viser resultatet av det første eksperimentet med SURF-detektoren.



Figur 4.3 viser lokaliserte egenskapspunkter for SURF-detektoren.

Resultatet viser at SURF-detektoren lokalisere de to minste sirklene av fire, uansett gråverdi på objektet i forhold til bakgrunnen. Avlange objekter blir også detektert, hvis lengden på objektet er en viss lengde, blir det detektert to egenskapspunkter på hver side av objektet. Det kommer også frem at flere av de lokaliserte egenskapspunktene kunne ha vært detektert som hjørner. Ved å se på et lite område i bildet kan helheten av objektet forsvinne. Derfor kan små deler av et objekt (f.eks. et hjørne) bli godkjent som et egenskapspunkt av SURF-detektoren.

Tidsbruk

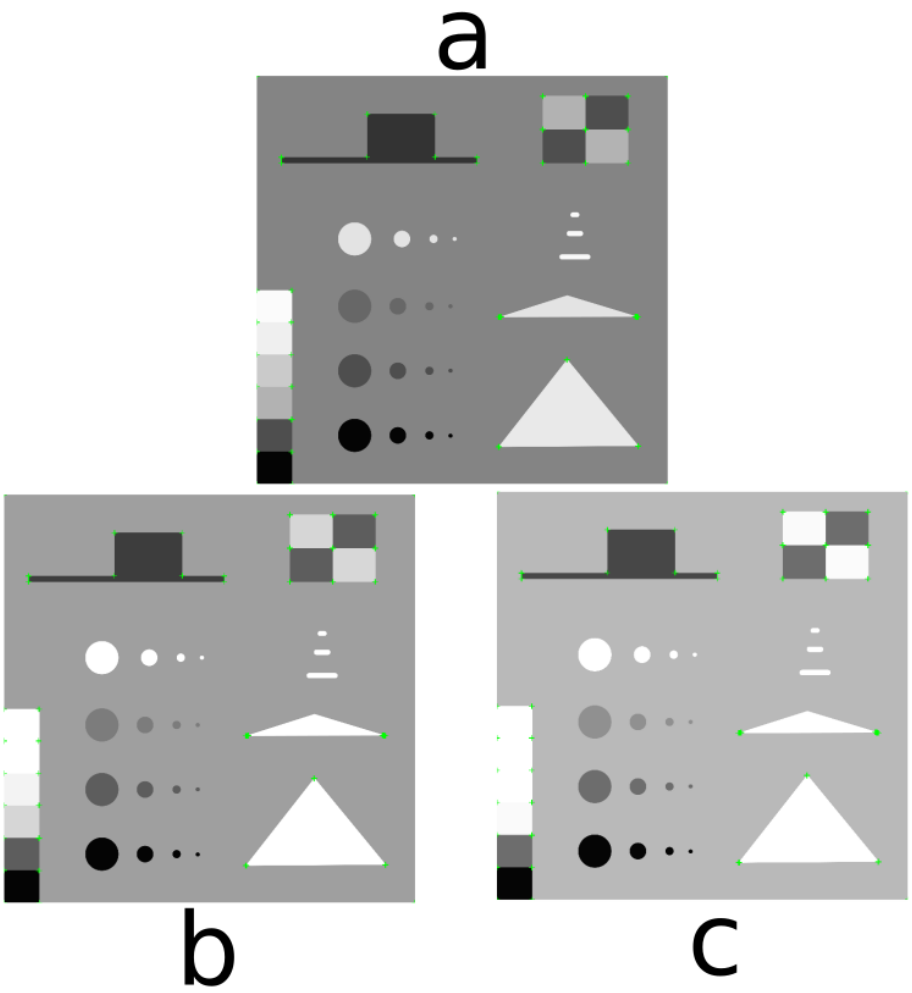
Tabell 1 viser at BRISK-detektoren var den raskeste av de tre, den var nesten tre ganger så rask som SURF-detektoren. Egen implementasjonen av SUSAN-detektoren var den tregeste av de tre, over 300 ganger så treg som BRISK-detektoren. Dette er et forventet resultat ettersom både BRISK og SURF er kjent for å være raske detektorer.

Detektor	Antall punkt	Tid[s]	Tid per egenskapspunkt [s]
SUSAN	156	42.422594	0.271939
BRISK	114	0.127784	0.001121
SURF	121	0.311386	0.002573

4.2 Resultat av intensitet endring i bilde 1

Under vises resultatet når detektorene har blitt testet på bilde 1 når intensiteten blir forandret. De grønne kryssene representerer lokaliserte egenskapspunkter. Og de grønne sirklene representerer skalaen BRISK og SURF har detektert egenskapspunktene på.

SUSAN-detektoren



Figur 4.4 viser lokaliserte punkt funnet av SUSAN-detektoren ved 0% (a), 20% (b) og 40% (c) intensitet forandring.

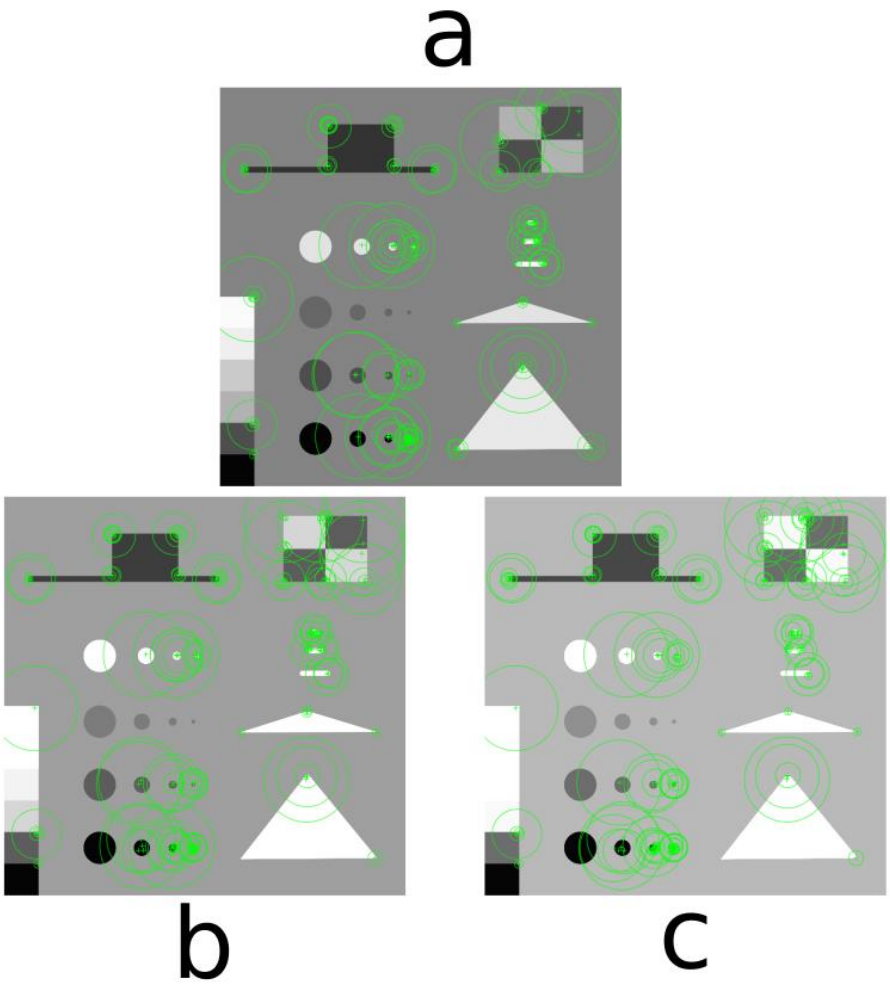
Figur 4.4 viser resultatet av SUSAN-detektoren ved 0%, 20% og 40% forandring i intensitet. Detektoren finne alle hjørnene som den gjorde fra det første eksperimentet og samme antall punkter rundt hjørnene. Ved de to øverste kvadratene nederst i venstre hjørne har det blitt oppdaget fire egenskapspunkter i et område med tilnærmet lik gråverdi. Ved studere de lokaliserte egenskapspunktene på pikselnivå kommer det frem at punktene funnet er riktig lokalisert. Dette skyldes at intensitet terskelen er såpass lav at detektoren kan finner hjørner

som det blotte øye ikke kan se. Figur 4.5 viser pikselverdien på området rundt et av de lokaliserte hjørnene.

1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.17	0.72	0.72	0.72	0.72	0.72
1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.17	0.72	0.72	0.72	0.72	0.72
1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.17	0.72	0.72	0.72	0.72	0.72
1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.17	0.72	0.72	0.72	0.72	0.72
1.35	1.35	1.35	1.35	1.35	1.35	1.35	1.35	1.15	0.72	0.72	0.72	0.72	0.72
1.31	1.31	1.31	1.31	1.31	1.31	1.31	1.31	1.13	0.72	0.72	0.72	0.72	0.72
1.31	1.31	1.31	1.31	1.31	1.31	1.31	1.31	1.13	0.72	0.72	0.72	0.72	0.72
1.31	1.31	1.31	1.31	1.31	1.31	1.31	1.31	1.13	0.72	0.72	0.72	0.72	0.72

Figur 4.5 viser hvordan et hjørne mellom to tilsynelatende like farger kan bli detektert av SUSAN-detektoren ved å bruke en lav intensitet terskelverdi.

BRISK-detektoren

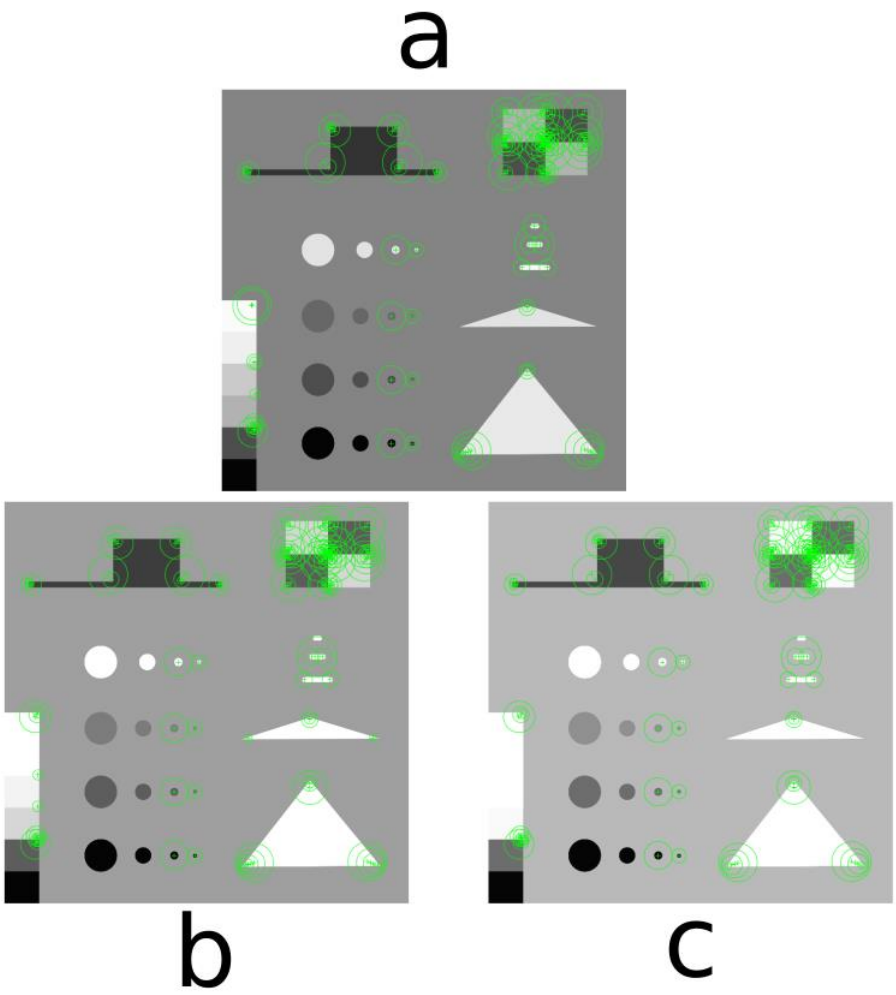


Figur 4.6 viser lokaliserte punkt funnet av BRISK-detektoren ved 0% (a), 20% (b) og 40% (c) intensitet forandring.

Figur 4.6 viser resultatet av BRISK-detektoren ved 0%, 20% og 40% intensitet forandring. Med eksperimentet kommer det tydeligere frem at BRISK-detektoren fortsatt lokaliserer punkter som ikke er hjørner. Denne negative egenskapen kommer frem i alle eksperimentene med intensitet forandring.

I original bildet blir det detektert 23 hjørner, hvor i bildet med 40% intensitet forandring blir også lokalisert 26 hjørner. Ved den høyeste intensitet forandring blir det lokalisert flere hjørner i kvadranten bestående av fire mindre kvadrater. Bare et hjørne ved den største trekanten blir ikke lokalisert i forhold til deteksjon i original bildet. Derfor kan det virke som BRISK-detektoren er bedre på bilder som er lyse.

SURF-detektoren



Figur 4.7 viser lokaliserte punkt funnet av SURF-detektoren ved 0% (a), 20% (b) og 40% (c) intensitet forandring.

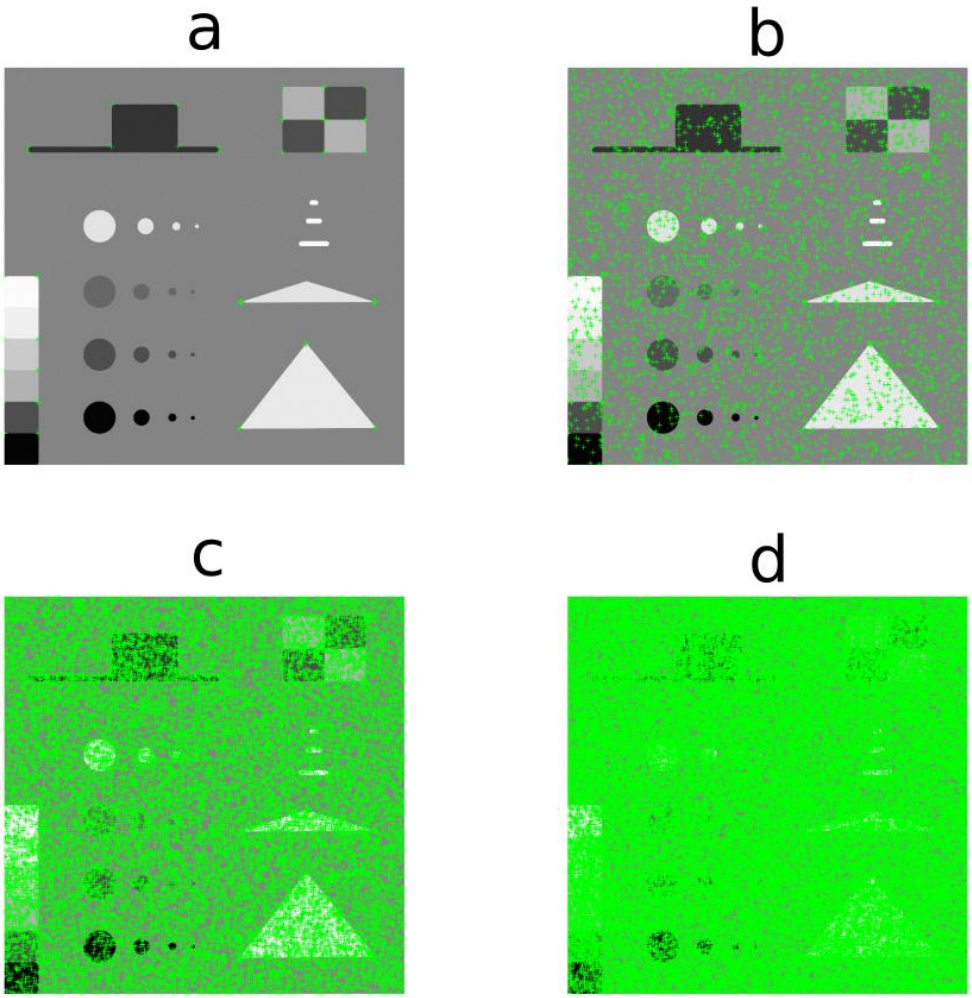
Resultatet for SURF-detektoren er konstant gjennom store deler av eksperimentet, vist her ved figur 4.7. Det blir lokalisert egenskapspunkter i de samme områdene fra 0% til 40%. Når det kommer til antall punkter lokalisert skiller bare et få antall når det kommer til lokalisering i original bildet og bildet med 40% lysere gråverdier.

4.3 Resultat av støy

Under vises resultatet når detektorene har blitt testet på bilde 1 når det blir lagt til gaussisk støy på bilde 1. De grønne kryssene representerer lokaliserte egenskapspunkter. Og de grønne sirklene representerer skalaen BRISK og SURF har detektert egenskapspunktene på.

SUSAN-detektoren

SUSAN-detektoren er kjent for å være robust mot støy ettersom den ikke beregner noen form for den deriverte. I dette eksperimentet kommer det frem at selv implementerte SUSAN-detektoren ikke er spesielt robust mot gaussisk støy. Figur 4.8 viser resultatet av eksperimentet.

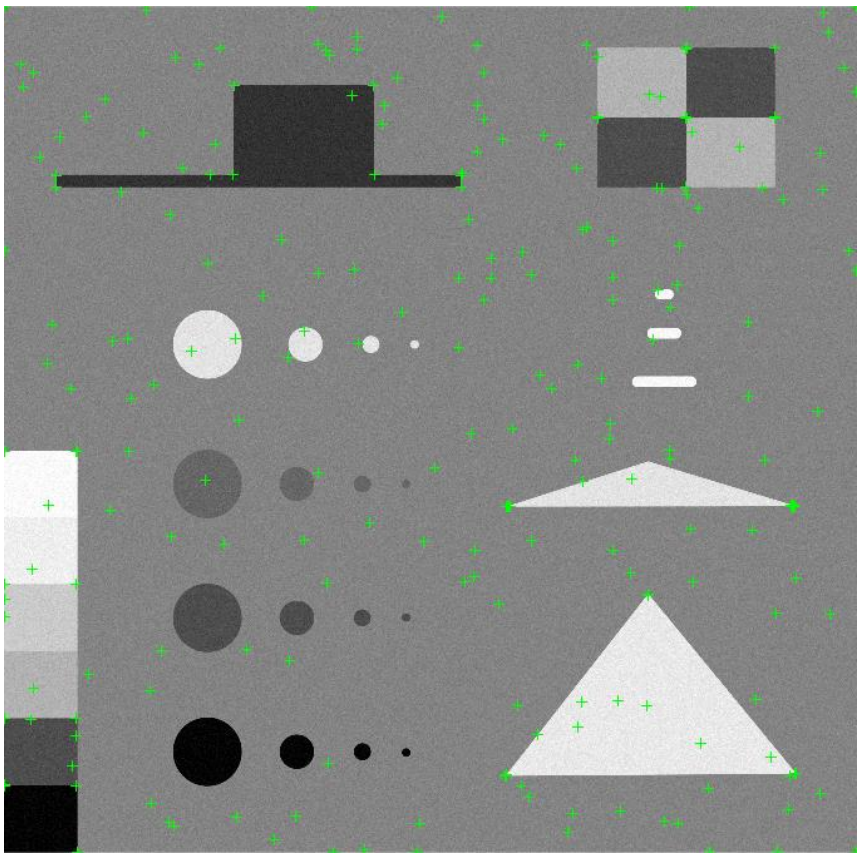


Figur 4.8 viser resultatet av SUSAN-detektoren ved varians 0,0001 (a), 0,0003 (b), 0,0005 (c) og 0,0007 (d)

Ved den laveste støymengden med gaussiske variansen satt til 0.0001, finner detektoren de samme hjørnene som i det første eksperimentet, unntatt to av dem. Ved øking av variansen øker også antall feildeteksjoner betraktelig ved å bruke samme terskelverdiene som i det første eksperimentet. Tabell 2 viser hvordan antall punkter stiger ved økende varians.

Støy [varians]	Antall punkter	Tid [s]
0.0001	95	37.031720
0.0003	3377	37.077239
0.0005	23417	39.274172
0.0007	56923	38.004665

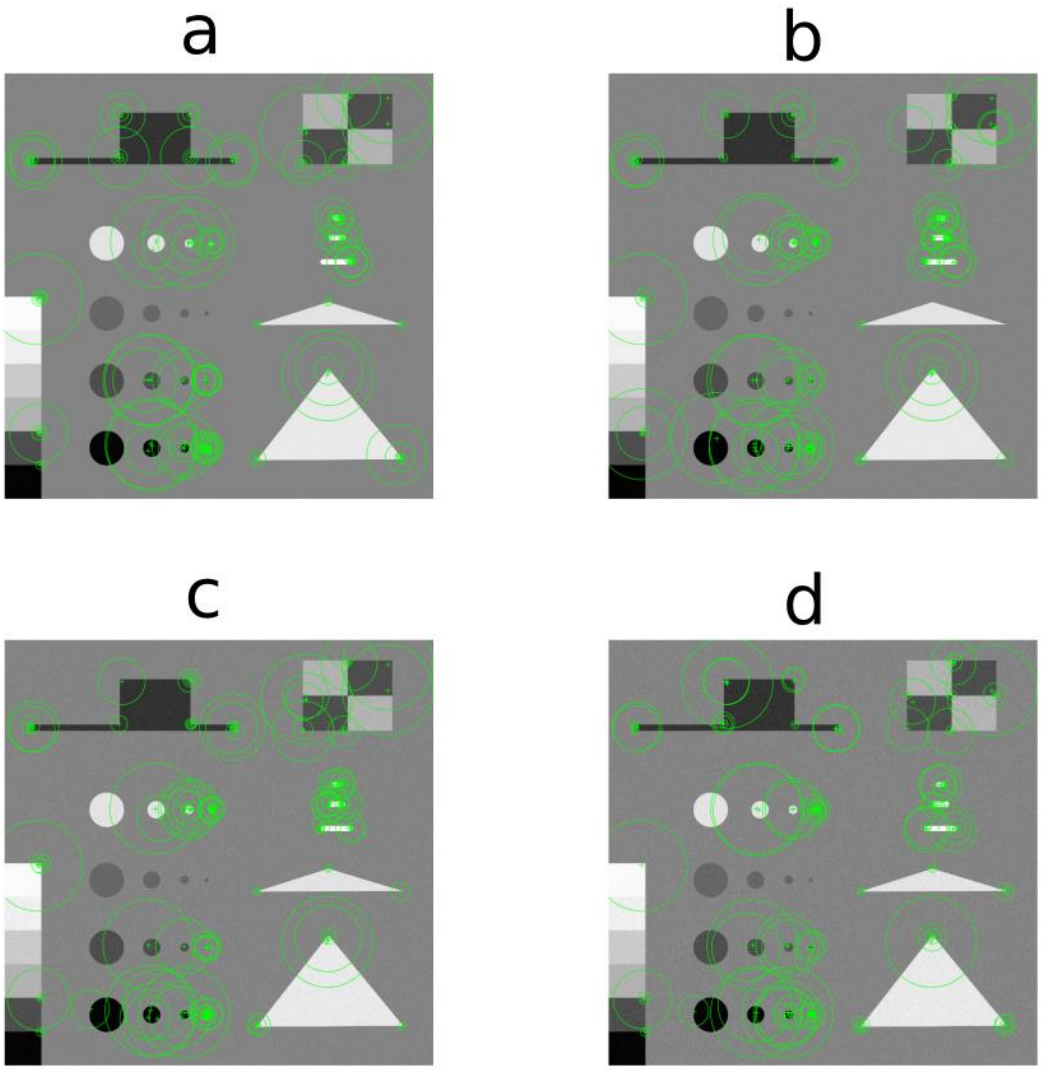
Ved å forandre på de geometriske og intensitet tersklene kan resultatet forbedres. Figur 4.9 viser resultatet med $varians = 0.0007$ og terskelverdiene satt til: $geoTerskel = 20, geoTerskel2 = 21, interskel = interskel1 = 0.099$.



Figur 4.9 viser resultatet av SUSAN-detektoren ved bruk av nye terskelverdier

Resultatet viser at det fortsatt er stort antall feil deteksjoner, men det har gått betraktelig ned. Fra å detektere 56923 punkter til å detektere 285 punkter med de nye terskelverdiene. Totalt er det sju hjørner som ikke blir detektert.

BRISK-detektoren

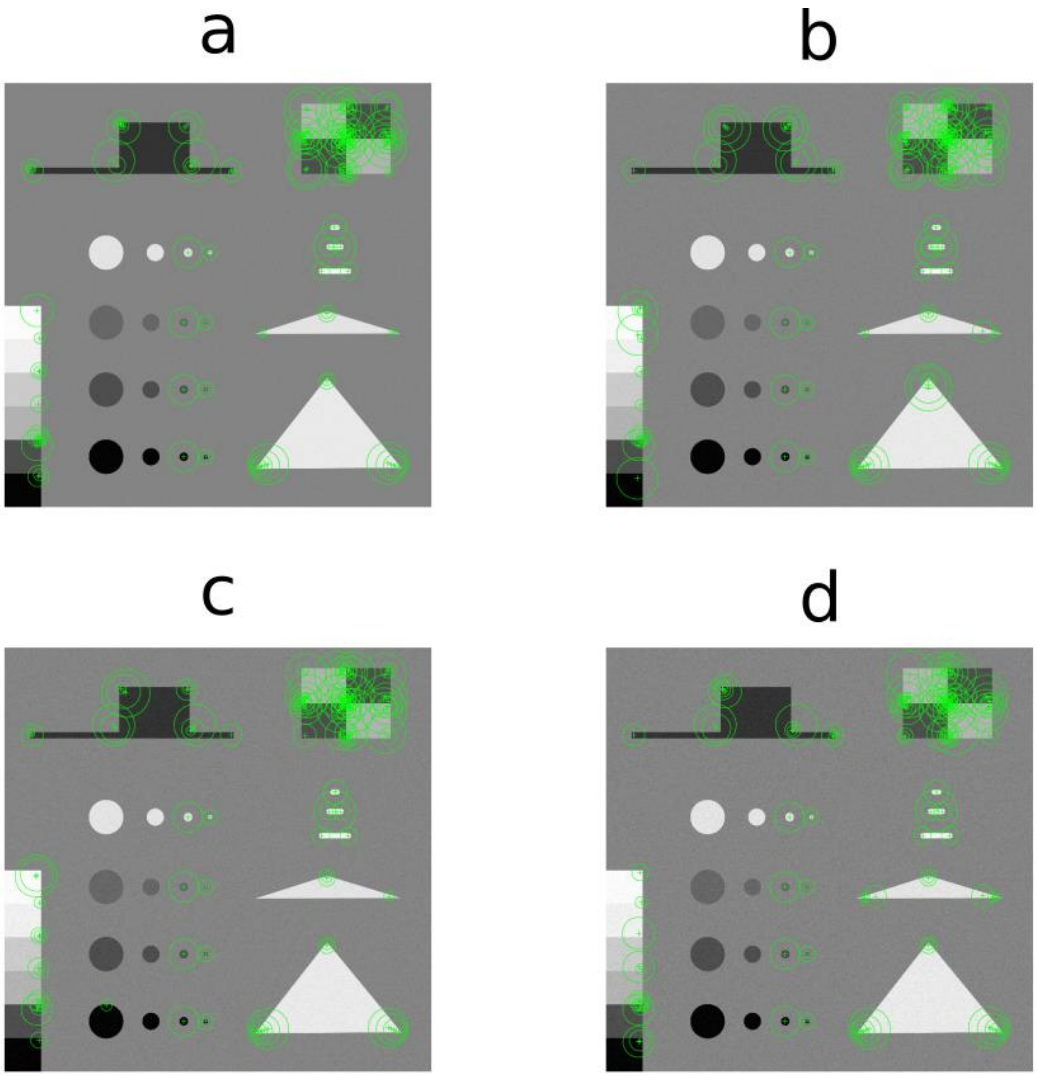


Figur 4.10 viser resultatet av BRISK-detektoren ved varians 0,0001 (a), 0,0003 (b), 0,0005 (c) og 0,0007 (d)

Figur 4.10 viser resultatet av BRISK-detektoren. Fra eksperiment 1 kommer det frem at BRISK-detektoren har en stor andel feildeteksjoner, og dette kommer frem i dette eksperimentet også. På det positive er BRISK-detektoren robust når det kommer til støy. Metoden lokaliserer stort sett de samme punktene bildene med forskjellig mengde støy. Tabell 3 viser antall punkt funnet av detektoren ved de forskjellige støymengdene.

Støy [varians]	Antall punkter	Tid [s]
0.0001	109	0.242345
0.0003	99	0.237085
0.0005	112	0.249322
0.0007	99	0.244022

SURF-detektoren



Figur 4.11 viser resultatet av SURF-detektoren ved varians 0,0001 (a), 0,0003 (b), 0,0005 (c) og 0,0007 (d)

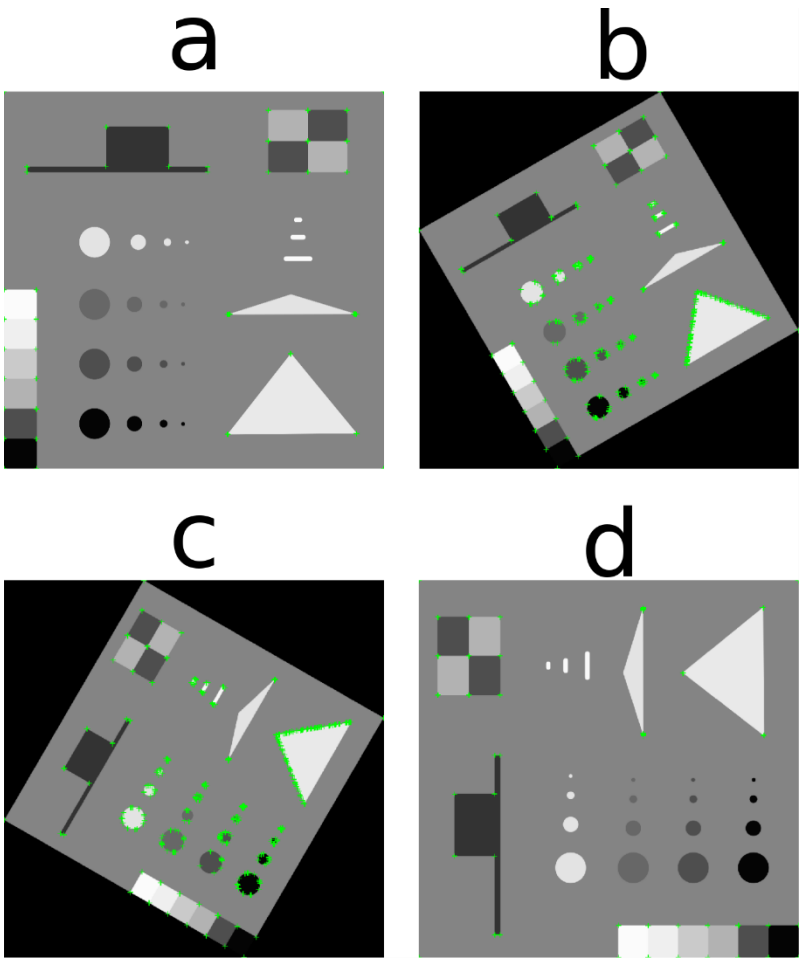
Figur 4.11 viser resultatet av SURF-detektoren ved forskjellig mengde støy. Detektoren lokaliserer de samme punktene i de sirkulære og avlange figurene som i eksperiment 1. Tap av flest punkter skjedde ved varians 0.0005 og 0.0007, men antallet tapte i forhold til antall funnet i eksperiment 1 er veldig lavt. Ved varians satt til 0.0003, ble det oppdaget flere punkter enn i eksperiment 1.

4.4 Resultat av rotasjon av bilde 1

Under vises resultatet når detektorene har blitt testet på bilde 1 når det blir lagt til gaussisk støy på bilde 1. De grønne kryssene representerer lokaliserte egenskapspunkter. Og de grønne sirklene representerer skalaen BRISK og SURF har detektert egenskapspunktene på.

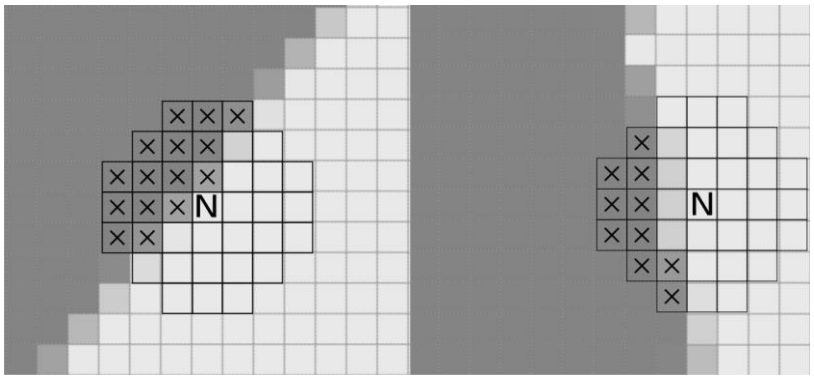
SUSAN-detektoren

SUSAN-detektoren finner de samme punktene ved 0º og 90º, og dermed blir alle hjørnene unntatt hjørnet med størst vinkel lokalisert. Det kan da forventes at detektoren vil oppnå samme resultat ved grader $90 \cdot n, \forall\, n \in \mathbb{Z}$. Fra 10º endring øker antall feildeteksjoner og stiger til det er oppnådd en rotasjon på 40º. Fra 50º og høyere synker antall feildeteksjoner, men antallet er fortsatt høyt. Figur 4.12 viser resultatet av SUSAN-detektoren ved rotasjonsvinkel 0 º, 30 º, 60 º og 90º.



Figur 4.12 viser resultatet av SUSAN-detektoren ved rotasjonsvinkel 0 º, 30 º, 60 º og 90º.

Grunnen til feildeteksjonene skyldes MATLAB-funksjonen benyttet for å legge SUSAN-masken over pikslene i bildet. Funksjonen beveger seg horisontalt i x-retning og tester 36 piksler i en fast $7x7$ maske. Figur 4.13 viser hvordan en piksel har blitt godkjent ved 50º og ikke ved 0º.



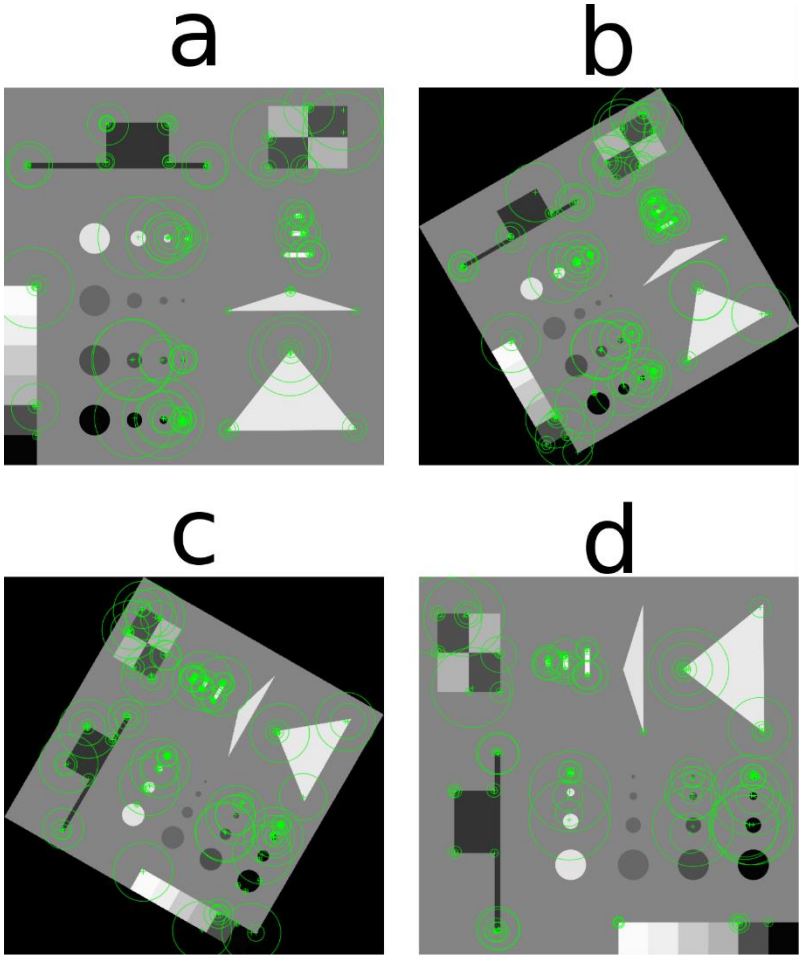
Figur 4.13 viser hvordan et punkt kan bli ikke godkjent (venstre) og godkjent (høyre) som egenskapspunkt avhengig av vinkel.

Ettersom alle figurene i bildet er laget med hensyn på eksperiment 1, ligger de horisontalt langs x-aksen. Alle hjørnene lokalisert i eksperiment 1 blir også lokalisert ved alle de forskjellige rotasjonene, ved 70º og 80º blir også hjørnet med størst vinkel lokalisert. Tabell 4 viser antall punkter funnet av SUSAN-detektoren ved rotasjonsvinkel 0º, 30º, 60º og 90º.

Vinkel [grader]	Antall punkter
0	156
30	525
60	534
90	156

BRISK-detektoren

Ved tidligere eksperimenter har BRISK-detektoren hatt et stort antall feildeteksjoner (piksler som ikke er en del av et hjørne blir detektert som et hjørne). Dette kommer frem i dette eksperimentet også. Antall hjørner detektert varierer fra 18 (rotasjonsvinkel på 20º og 80º) til 23 (rotasjonsvinkel på 0º, 50º og 70º). Figur 4.14 viser resultatet av BRISK-detektoren ved 0º, 30º,60º og 90º.



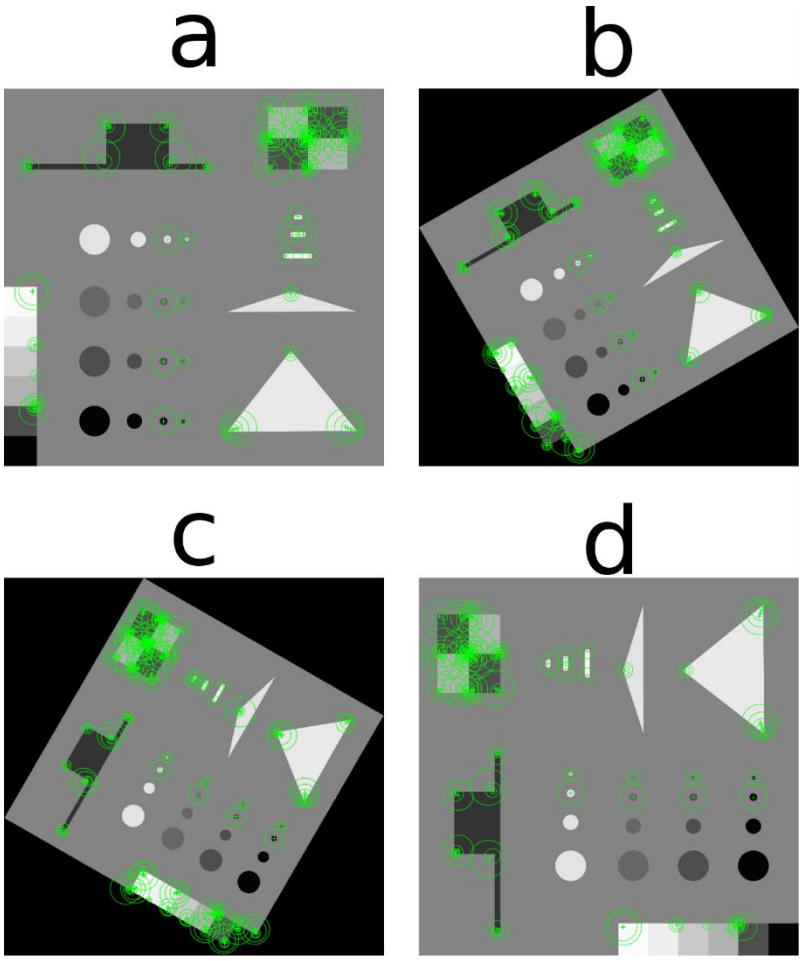
Figur 4.14 viser resultatet av BRISK-detektoren ved rotasjonsvinkel 0 º, 30 º, 60 º og 90º

På samme måte som SUSAN-detektoren, bruker BRISK-detektoren en sirkulær maske for å avgjøre om en piksel er hjørne eller ikke. Derfor ville det være naturlig at BRISK-detektoren ville ha et økt antall feildeteksjoner i dette eksperimentet som SUSAN-detektoren hadde. Men dette er ikke tilfelle, BRISK-detektoren har et mindre avvik av detekterte punkter i forhold til SUSAN. Tabell 5 viser antall punkter detektert ved rotasjonsvinkel 0º, 30º, 60º og 90º.

Vinkel [grader]	Antall punkter
0	114
30	154
60	150
90	110

SURF-detektoren

SURF-detektoren øker i antall punkter detektert i alle forsøkene i forhold til eksperiment 1, utenom med vinkel 90º hvor det blir detektert likt antall punkter. Ved vinkel 10º og 80º blir det detektert flest punkter, over 100 flere punkter i forhold til resten. Flertallet av disse punktene er å finne i venstre hjørne (ved de 6 kvadratene med forskjellig gråverdi). Resten av resultatene er rundt 170 punkter for de forskjellige vinkelrotasjonene. Figur 4.15 viser resultatet av SURF-detektoren ved 0º, 30º,60º og 90º.



Figur 4.15 viser resultatet av SURF-detektoren ved rotasjonsvinkel 0 º, 30 º, 60 º og 90º

SURF-detektoren er en av flere detektorer som er basert på Hessematrisen. Detektorer som er bruker Hessematrisen har en svakhet ved deteksjon av egenskaper rundt vinkler $n \cdot \frac{\pi}{4}$, hvor n er oddetall [15] (45º, 135º og 225º er eksempler på vinkler hvor en Hessematrise basert detektor kan ha svakheter). Dette kommer ikke frem i dette eksperimentet, SURF-detektoren finner

omtrent like mange punkter ved 40º som ved for eksempel 20º og 70º. Tabell 6 viser antall punkter funnet av SURF-detektoren ved 0º, 30º,60º og 90º.

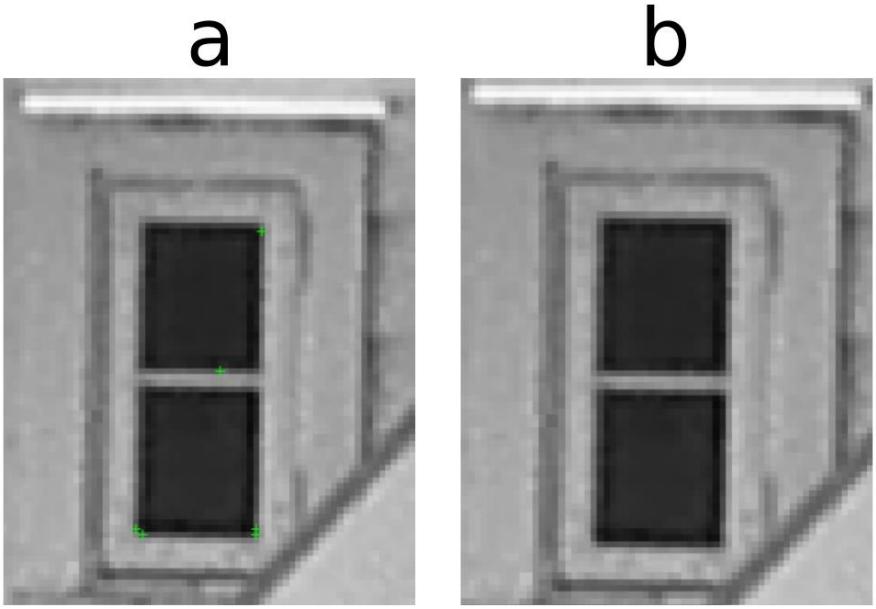
Vinkel [grader]	Antall punkter
0	121
30	164
60	166
90	121

4.5 Endring av skala i bilde 2

Under vises resultatet når detektorene har blitt testet på bilde 1 uten noen form for forstyrrelser. De grønne kryssene representerer lokaliserte egenskapspunkter. Og de grønne sirklene representerer skalaen BRISK og SURF har detektert egenskapspunktene på.

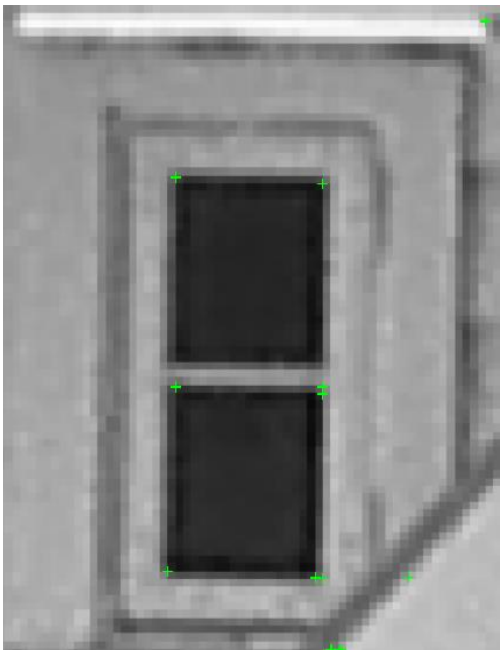
SUSAN-detektoren

Resultatet av eksperimentet for SUSAN-detektoren blir vist ved figur 4.16. Det kommer frem at SUSAN-detektoren ikke finner de samme punktene ved original skala og forstørret ved bruk av de samme terskelverdier. Ved den originale skalaen (bilde a) lokaliserer detektoren tre hjørner og en feildeteksjon. Ingen egenskapspunkter blir lokalisert i bildet som viser området forstørret.



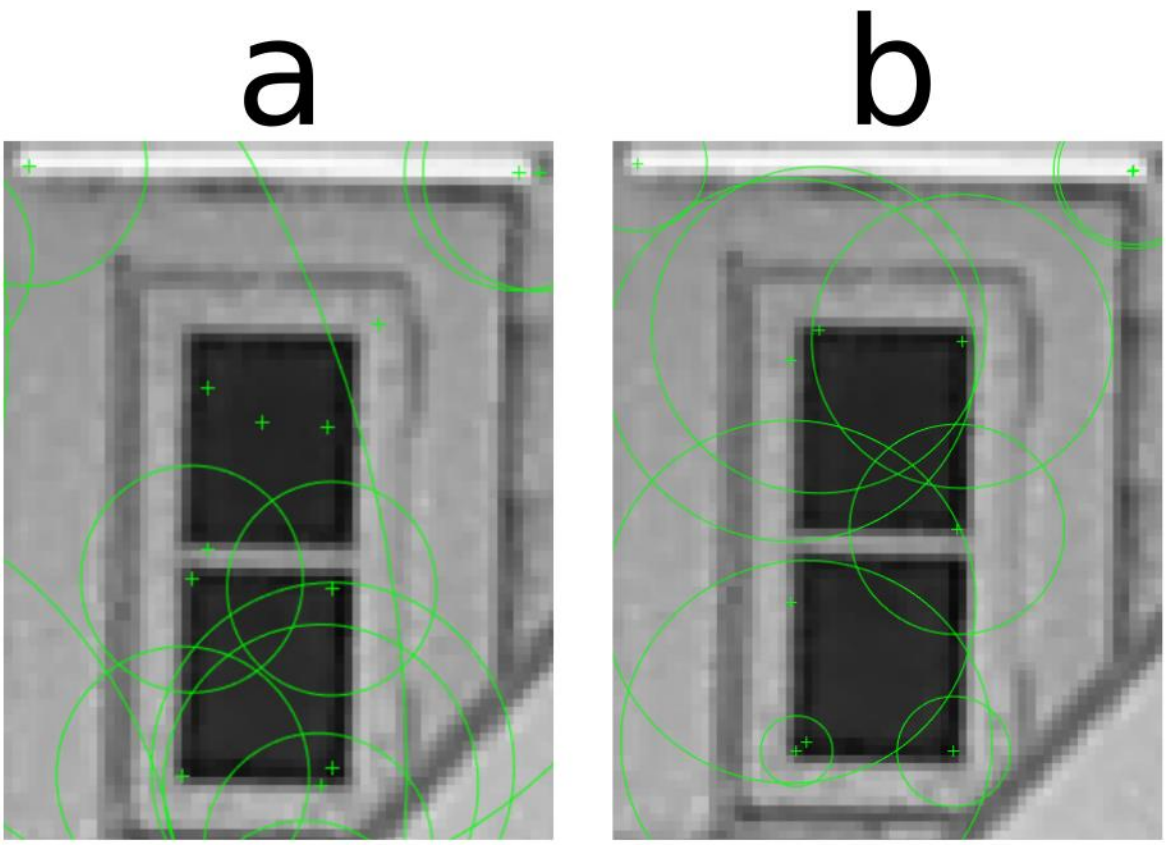
Figur 4.16 viser lokaliserte punkter ved original skala (a, forstørret 4 ganger) og ved ny skala (b)

Ved å forandre terskelverdiene til $geoTerksel = 5, geoTerksel2 = 15, interskel = 0.05$ og $interskel1 = 0.01$. Kan resultatet bli forbedret, her vist ved figur 4.17. Her blir sju hjørner lokalisert i motsetning til null ved bruk av de andre terskelverdiene. Ved bruk av disse terskelverdiene i original bildet, øker antall kanter detektert og antall feildeteksjoner (punkter som ikke er hjørne eller kant).



Figur 4.17 viser lokaliserte punkter ved bruk av nye terskelverdier.

BRISK-detektoren



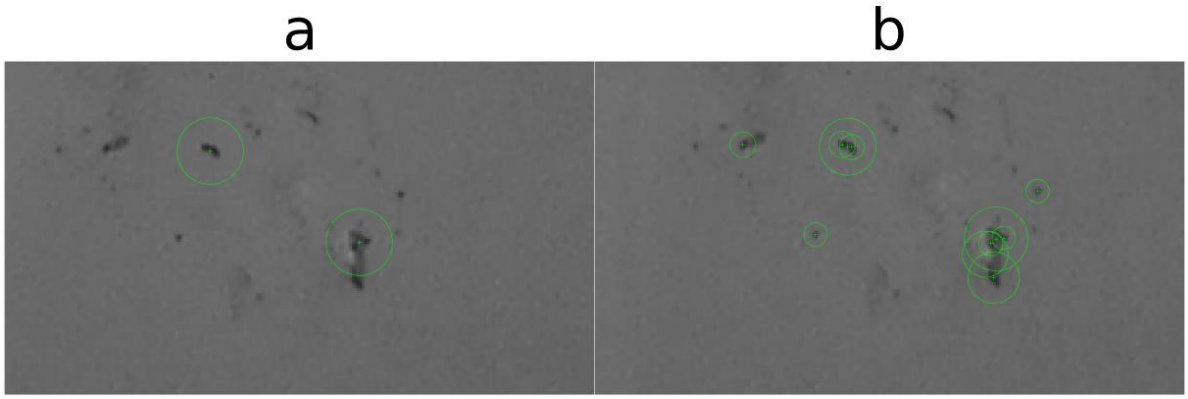
Figur 4.18 viser egenskapspunkt som BRISK-detektoren har lokalisert. Her viser bilde (a) lokaliserte egenskapspunkt i et område i original bildet (forstørret 4 ganger for å lettere se) og bilde (b) viser lokaliserte egenskapspunkt lokalisert ved å forstørre området i bildet 4 ganger.

Resultatet for BRISK-detektoren blir her vist ved figur 4.18. Detektoren lokaliserer fem hjørner på den originale skalaen (seks hjørner hvis yttersiden telles som et hjørne). Tre av de lokalisert punktene kan bli sett på som feildeteksjon. Ettersom de ligger for langt unna et hjørne. Hvor på bildet som er forstørret lokaliserer BRISK fem hjørner. Bare et av punktene lokalisert i nærheten av vinduet er feildeteksjon.

Hva som tolkes som et riktig lokalisert punkt kan diskuteres. Hvor nærme et punkt skal være et hjørne for at det skal bli godkjent, avhenger hva detektoren skal brukes til. Hvis antall punkter er mer viktig enn nøyaktighet, kan noen av de punktene som er tolket som feildeteksjoner bli godkjent.

SURF-detektoren

SURF-detektorene skal lokalisere områder som er mørkere eller lysere enn bakgrunnen. Figur 4.19 viser resultatet av det siste eksperimentet med SURF-detektoren.



Figur 4.19 viser lokaliserte egenskapspunkter for SURF-detektoren. Her viser bilde (a) lokaliserte egenskapspunkt i et område i original bildet (forstørret 4 ganger for å lettere se) og bilde (b) viser lokaliserte egenskapspunkt lokalisert ved å forstørre området i bildet 4 ganger.

Resultatet viser at SURF-detektoren lokalisere to områder som er mørkere enn omgivelsene på original skala. Mens ved fire ganger forstørrelse, blir det lokalisert totalt elleve slike punkter. Av disse elleve punktene er flere av samme objekt, bare lokalisert ved forskjellige skalaer (punkter har blitt lokalisert ved forskjellig filter størrelser).

Ettersom bildet er mindre, blir de mørker områdene enda mer fremtredende i bildet som er forstørret. Figur 4.20 viser hvordan de mørke områdene på søppelcontaineren blir mindre fremtredende i det originale bildet.

Figur 4.20 viser området valgt ut for å teste SURF-detektoren i original bildet.

5 Konklusjon

I denne oppgaven har de tre egenskapsdetektorer SURF, SUSAN og BRISK sin robusthet blitt testet ved å utføre fem ulike eksperimenter. Det har i tillegg blitt laget en egen versjon av SUSAN-detektoren i MATLAB for deteksjon av hjørner, som fungerer tilfredsstillende.

Det kommer frem i eksperimentene at BRISK-detektoren har en god del feildeteksjoner under alle de forskjellige eksperimentene. Men detektoren finner stort sett de samme hjørnene i eksperimentene. BRISK-detektoren er i tillegg den raskeste av de tre detektorene og gjør bra ved skala forandring i forhold til SUSAN.

SUSAN-detektoren er veldig nøyaktig ved riktig terskelverdier. Dette kommer godt frem i de to eksperimentet hvor detektoren finner alle hjørnene unntatt ett. Dessverre er ikke detektoren like robust mot støy eller rotasjon, hvor antall feildeteksjoner øker betraktelig. Den er i tillegg den tregeste av de tre detektorene.

Den siste av de tre detektorene skiller seg fra de to andre med at den finner områder lyser eller mørkere enn bakgrunnen istedenfor hjørner. Det kommer frem at SURF-detektoren lokaliserer de to minste sirklene i de fire første eksperimentene. Den godkjenner også områder som er del av et større objekt, ettersom den bruker forskjellig skalaer til å finne egenskapspunkter.

Hvilken detektor som er best, avhenger av hva detektoren skal brukes til. Ved sanntid applikasjoner er BRISK og SURF de tydeligste valgene. SUSAN-detektoren er nøyaktig så lenge de riktig terskelverdiene blir brukt, som er en tidkrevende prosess. Hver detektor har sine positive og negative svakheter, så den beste er den som har de fleste positive egenskapene ved valgt bruksområde.

6 Referanser

[1] K. Mikolajczyk and T. Tuytelaars, «Local Invariant Feature Detectors: A Survey,» *Computer Graphics and Vision*, vol. 3, nr. 3, 2008.

[2] T. Lindeberg, «Detection Salient Blob-Like Image Structures and Their Scales with a Scale-Space Primal Sketch: A Method for Focus-of-Attention,» *International Journal of Computer Vision*, vol. 11, nr. 3, pp. 283-318, 1993.

[3] P. Majer, «A Statictical Approach to Feature Detection and Scale Selection in Images,» Doktorgrad i økonomi, Universitetet i Göttingen, Göttingen, 2000.

[4] R. Szeliski, «Feature detection and matching,» i *Computer Vision: Algorithms and Applications*, London, Springer, 2010, pp. 221-228.

[5] C. Wang,S. Li,Y. Shen,Y.Song og H. Chang, «Evaluation of Feature Detectors and Descriptors for Motion Detection From Aerial Videos,» i *22nd International Conference on Pattern Recognition*, Stockholm, 2014.

[6] C. Harris and M. Stephens, «A combined corner and edge detector,» i *Proceedings of the Fourth Alvey Vision Conference*, Manchester, 1988.

[7] H. P. Moravec, «Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover,» The Robotics Institute of Carnegie Itiellon University, 1980.

[8] J. Shi and C. Tomasi, «Good Features to Track,» i *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, 1994.

[9] K. Mikolajczyk and C. Schmid, «Scale and Affine Invariant Interest Point Detectors,» *International Journal of Computer Vision*, vol. 60, nr. 1, pp. 63-86, 2004.

[10] S.M. Smith and J.M Brady, «SUSAN - A new Approach to Low Level Image Processing,» 1995.

[11] E. Rosten and T. Drummond, «Machine learning for high-speed corner detection,» i *9th European Conference on Computer Vision (ECCV)*, Graz, 2006.

[12] D. Burschka, G. D. Hager, G. Hirzinger, E. Mair and M. Suppa, «Adaptive and Generic Corner Detection Based on the Accelerated Segment Test,» i *Proceedings of the European Conference on Computer Vision*, 2010.

[13] T. Lindeberg, «Scale-space theory: A basic tool for analysing strctures at different scales,» *Journal of Applied Statistics*, vol. 21, nr. 2, pp. 224-270, 1994.

[14] D. G. Lowe, «Object Recognition from Local Scale-Invariant Features,» i *Proc. of the International Conference on Computer Vision* , Corfu, 1999.

[15] H. Bay, A. Ess, T. Tuytelaars and L.Vl Gool, «Speeded-up Robust Features (SURF),» *Computer Vision and Image Understanding (CVIU),* vol. 110, nr. 3, pp. 346-359, 2008.

[16] M. Chli, S. Leutenegger and R. Y. Siegwart, «BRISK: Binary Robust Invariant Scalable Keypoints,» i *IEEE International Conference on Computer Vision*, Barcelona, 2011.

[17] C. Schmid, R. Mohr og C. Bauckhage, «Evaluation of Interest Point Detectors,» *International Journal of Computer Vision,* vol. 37, nr. 2, pp. 151-172, 2000.

[18] A. M. López, F. Lumbreras, J. Serrat, og J. J. Villanueva, «Evaluation of Methods for Ridge and Valley Detection,» *IEEE Transactions on pattern analysis and machine intelligence,* vol. 21, nr. 4, pp. 327-335, 1999.

[19] T. Moranduzzo og F. Melgani, «Comparison of different feature detectors and descriptors for car classification in UAV images,» i *IEEE International Geoscience and Remote Sensing Symposium*, Melbourne, 2013.

[20] T. Pavlidis, Graphic and image processing, Rockville: Computer Science Press, 1982.

[21] D. P. Mitchell og A. N. Netravali, «Reconstruction Filters in Computer Graphics,» AT&T Bell Laboratories, New Jersey, 1974.

[22] A. Neubeck og L. V. Gool, «Efficient Non-Maximum Suppression,» i *International Conference on Pattern Recognition*, Washington, DC, 2006.

[23] P. Viola og M. Jones, «Rapid object detection using a boosted cascade of simple features,» i *Computer Vision and Pattern Recognition*, Kauai, 2001.

[24] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt and J. M. Ogden, «Pyramid methods in image processing,» RCA Engineer , 1984.

[25] Y. Xingfang, H.Yumei og L.Yan, «An improved SUSAN corner detection algorithm based on adaptive threshold,» i *2nd International Conference on Signal Processing System (ICSPS)*, Dalian, 2010.

[26] H. Mingliang og X. Shubin, «Study of Improving the Stability of SUSAN Corner Detection Algorithm,» i *International Conference on Computer Application and System Modeling* , Taiyuan, 2010.

[27] J. Zhao, H. Ma og G. Men , «A new corner detection algorithm with SUSAN fast hierarchical method,» i *International Asia Symposium on Intelligent Interaction and Affective Computing*, Wuhan, 2009.

[28] B.Jeon, D. Woo, Y. Mo og M. Lim, «An Improved Corner Point Detection Using Extreme Value,» i *ICROS-SICE International Joint Conference*, Fukuoka, 2009.

Appendiks A MATLAB kode

MATLAB kodene er plassert i zipfilen MATLAB. En oversikt over funksjonene er gitt her:

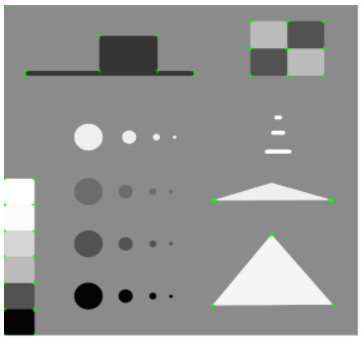
- **SURF.m**
- **BRISK.m**
- **SUSAN.m**
- **testSUSAN.m**

Appendiks B Resultatbilder

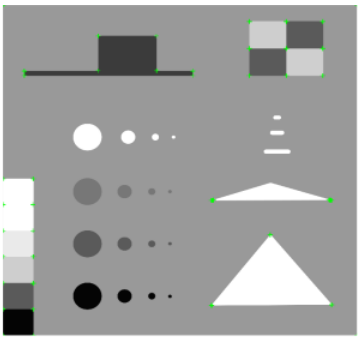
Bilder for eksperiment 2 og 4 er lagt ved. Eksperiment 1, 3 og 5 er ikke lagt ved i vedlegg ettersom de allerede er inkludert i hovedteksten.

B.1 Eksperiment 2 – Lysforandring

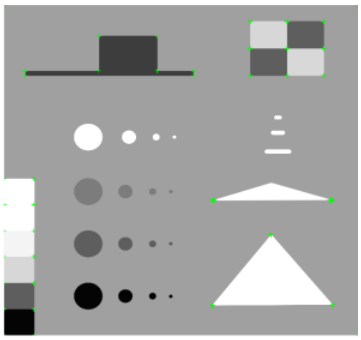
SUSAN



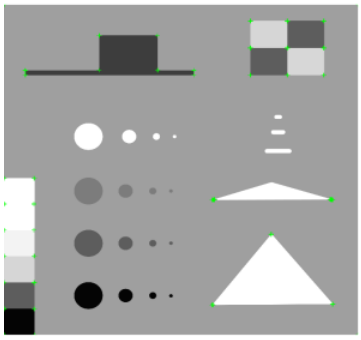
SUSAN 5%



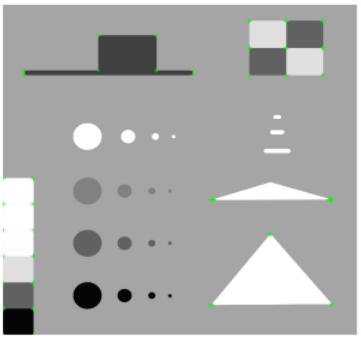
SUSAN 10%



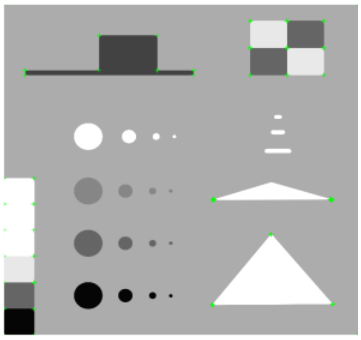
SUSAN 15%



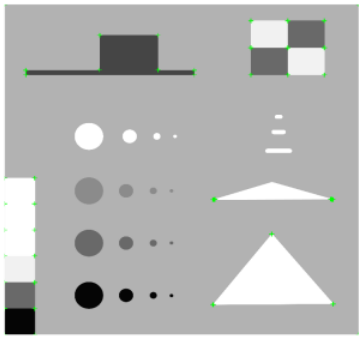
SUSAN20%



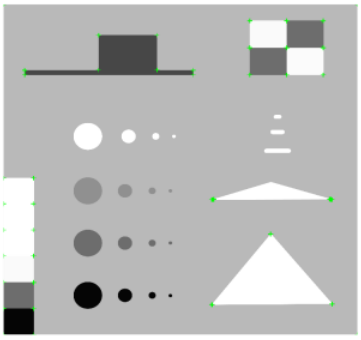
SUSAN 25%



SUSAN 30%

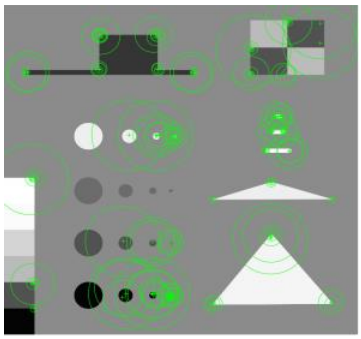


SUSAN 35%

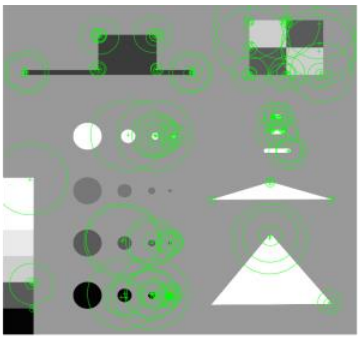


SUSAN 40%

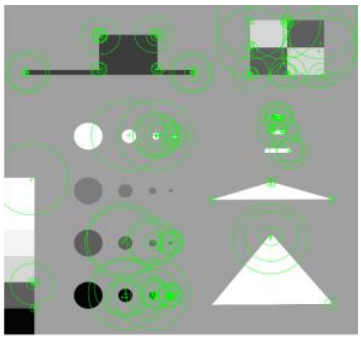
BRISK



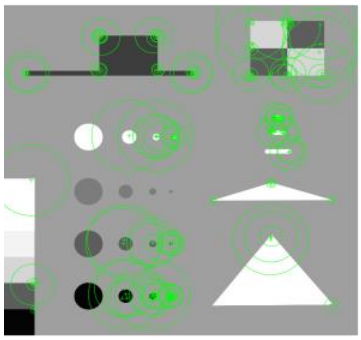
BRISK 5%



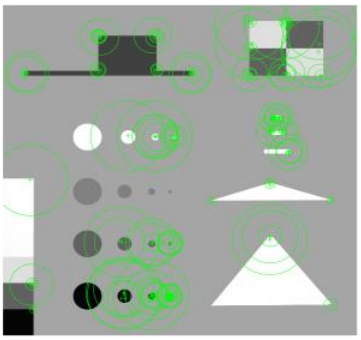
BRISK 10%



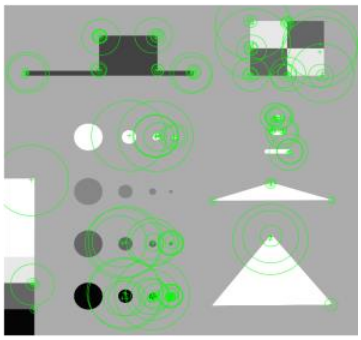
BRISK 15%



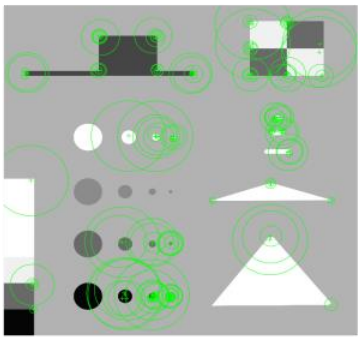
BRISK 20%



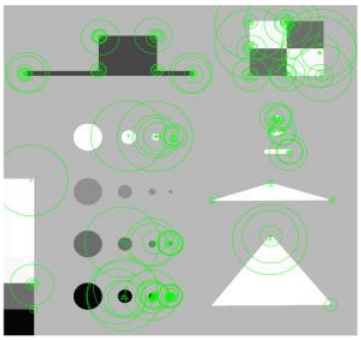
BRISK 25%



BRISK 30%

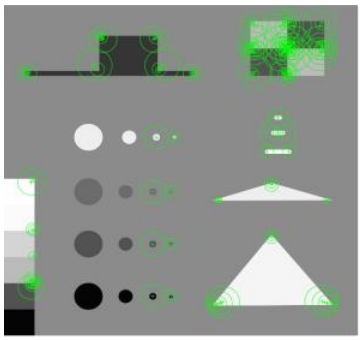


BRISK 35%

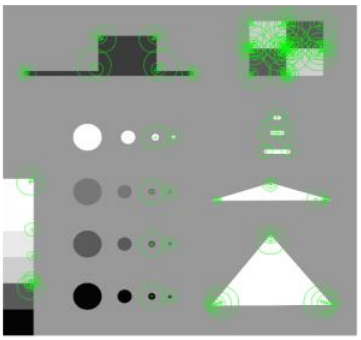


BRISK 40%

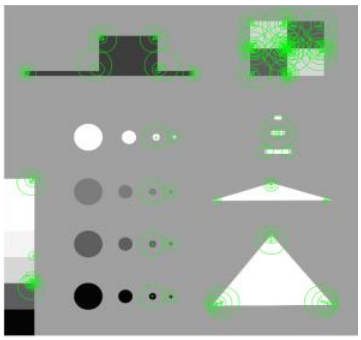
SURF



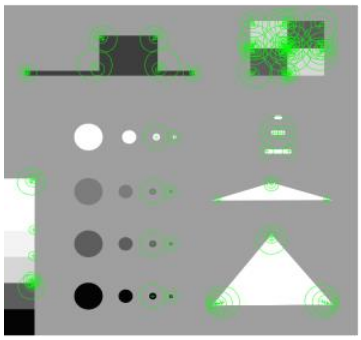
SURF 5%



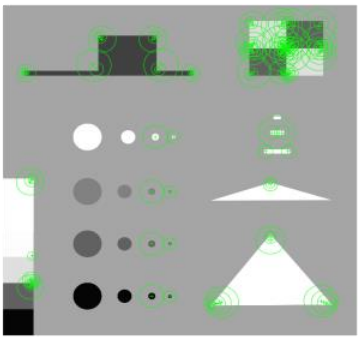
SURF 10%



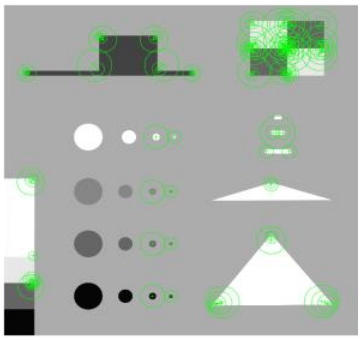
SURF 15%



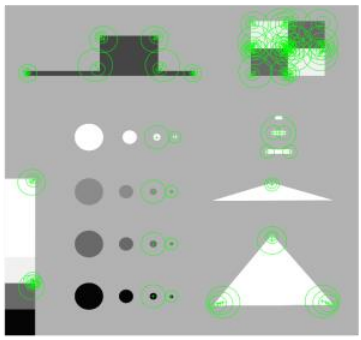
SURF 20%



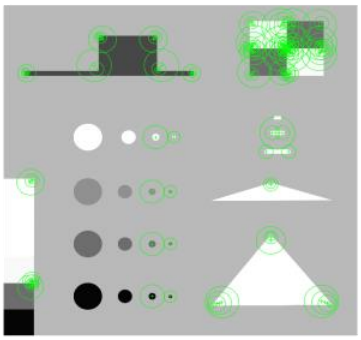
SURF 25%



SURF 30%



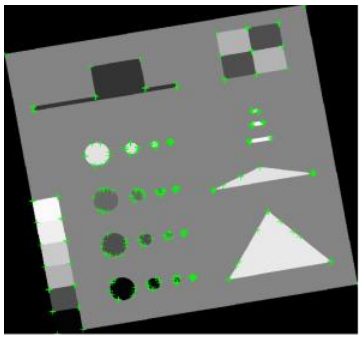
SURF 35%

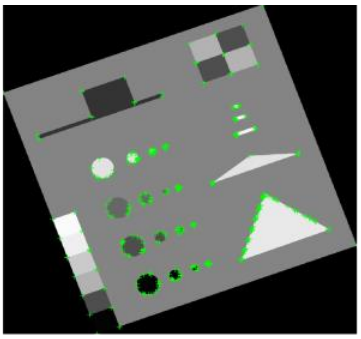


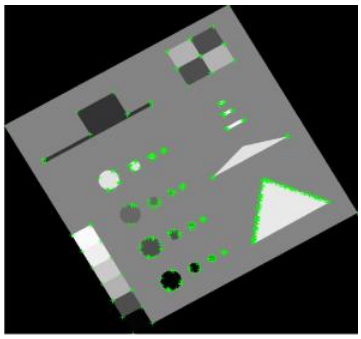
SURF 40%

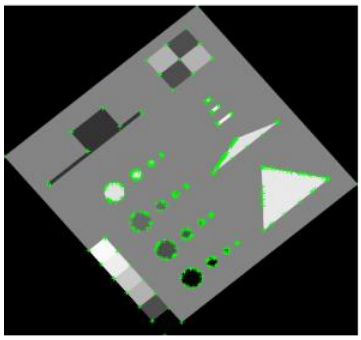
B.2 Eksperiment 4 - Rotasjon

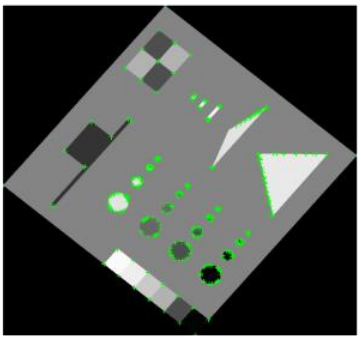
SUSAN

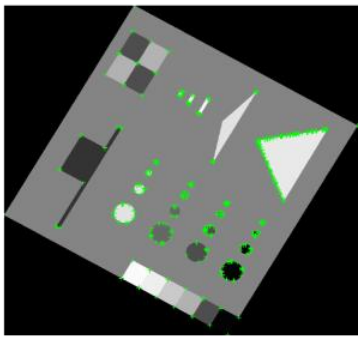

SUSAN 10 grader

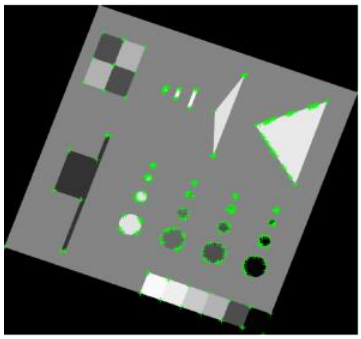

SUSAN 20 grader

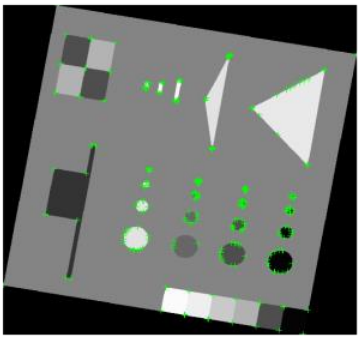

SUSAN 30 grader

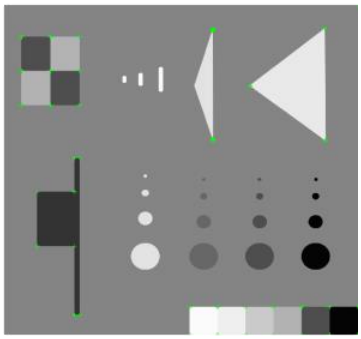

SUSAN 40 grader


SUSAN 50 grader

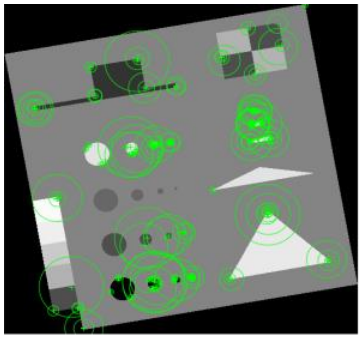

SUSAN 60 grader

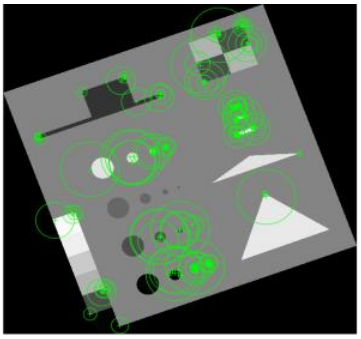

SUSAN 70 grader

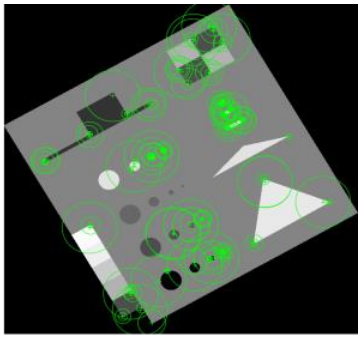

SUSAN 80 grader

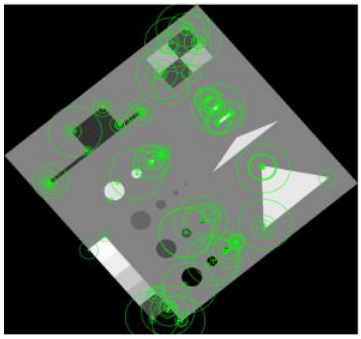

SUSAN 90 grader

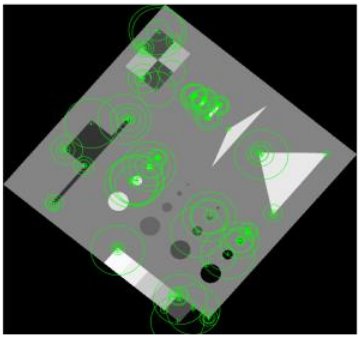
BRISK

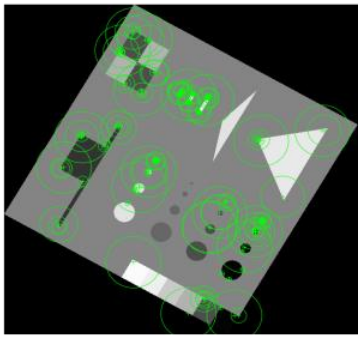

BRISK 10 grader

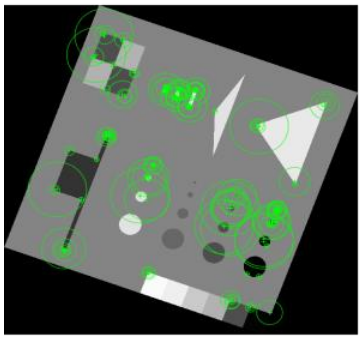

BRISK 20 grader

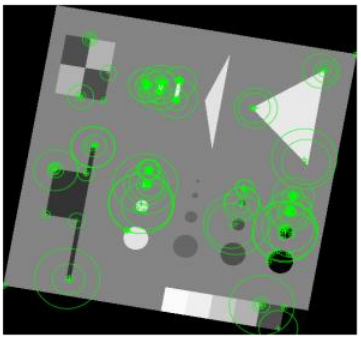

BRISK 30 grader

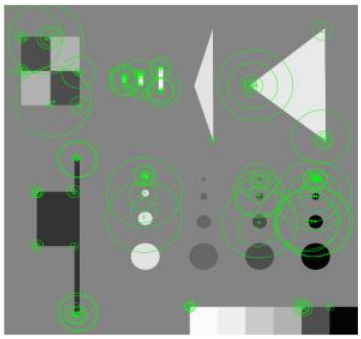

BRISK 40 grader


BRISK 50 grader

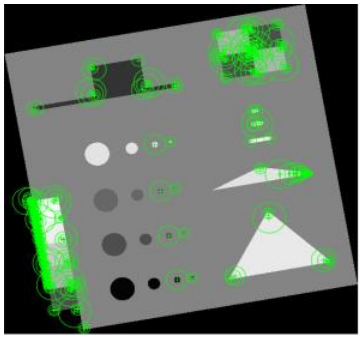

BRISK 60 grader


BRISK 70 grader

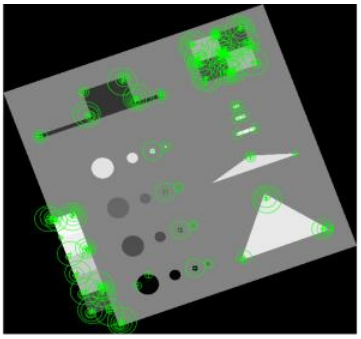

BRISK 80 grader


BRISK 80 grader

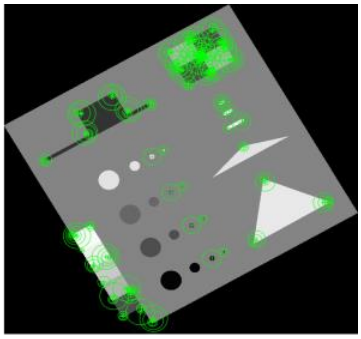
SURF



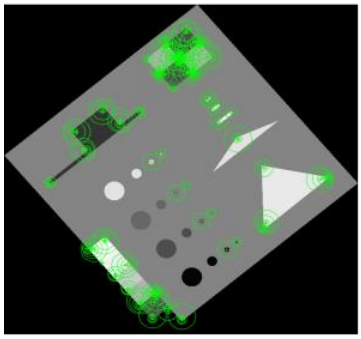
SURF 10 grader



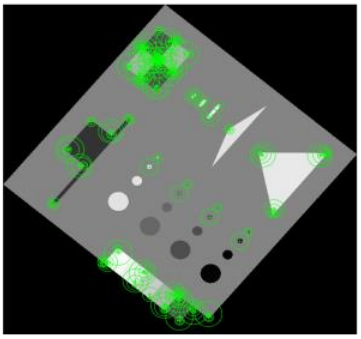
SURF 20 grader



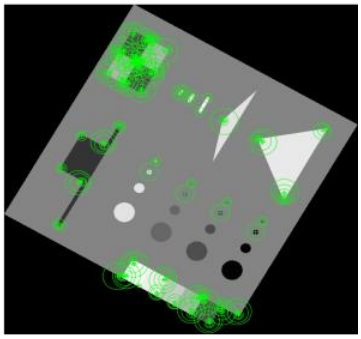
SURF 30 grader



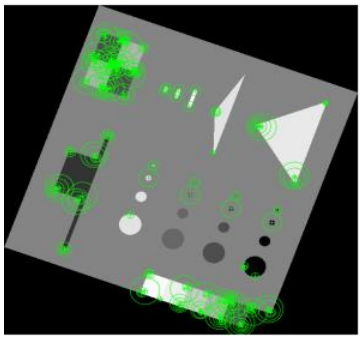
SURF 240 grader



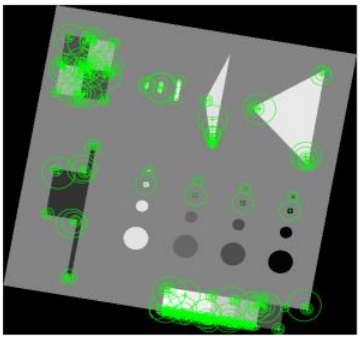
SURF 50 grader



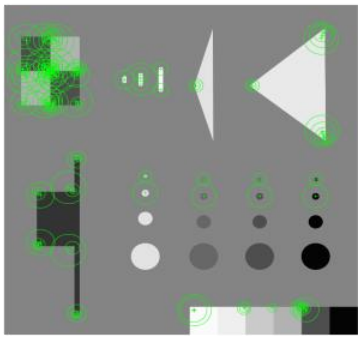
SURF 60 grader



SURF 70 grader



SURF 80 grader



SURF 80 grader

Appendiks C Tabeller

Tabell 7.1: SUSAN-detektoren ved lysendring (0-40%). Bilde: Bilde1

Lysendring [%]	Antall punkter	Tid [s]
0	156	42.422594
5	156	37.772770
10	156	35.794376
15	156	35.700833
20	156	35.883994
25	156	35.953140
30	156	35.526650
35	156	35.648046
40	156	36.061109

Tabell 7.2: BRISK-detektoren ved lysendring (0-40%). Bilde: Bilde1

Lysendring [%]	Antall punkter	Tid [s]
0	114	0.127784
5	115	0.212622
10	124	0.219465
15	123	0.218805
20	124	0.221538
25	126	0.218074
30	128	0.218903
35	128	0.217932
40	134	0.217815

Tabell 7.3: SURF-detektoren ved lysendring (0-40%). Bilde: Bilde1

Lysendring [%]	Antall punkter	Tid [s]
0	121	0.311386
5	124	0.310121
10	124	0.308694
15	121	0.300930
20	122	0.296733
25	118	0.303316
30	121	0.290688
35	116	0.306231
40	118	0.300447

Tabell 7.4: SUSAN-detektoren ved støyendring (0-0.0007). Bilde: Bilde1

Støy [varians]	Antall punkter	Tid [s]
0.0001	95	37.031720
0.0003	3377	37.077239
0.0005	23417	39.274172
0.0007	56923	38.004665

Tabell 7.5: BRISK-detektoren ved støyendring (0-0.0007). Bilde: Bilde1

Støy [varians]	Antall punkter	Tid [s]
0.0001	109	0.242345
0.0003	99	0.237085
0.0005	112	0.249322
0.0007	99	0.244022

Tabell 7.6: SURF-detektoren ved støyendring (0-0.0007). Bilde: Bilde1

Støy [varians]	Antall punkter	Tid [s]
0.0001	120	0.374735
0.0003	128	0.368954
0.0005	116	0.352339
0.0007	116	0.341600

Tabell 7.7: SUSAN-detektoren ved rotasjonsendring (10-90 grader). Bilde: Bilde1

Vinkel [grader]	Antall punkter	Tid [s]
10	391	49.911845
20	466	61.856766
30	525	69.572787
40	579	77.593237
50	560	74.035237
60	534	81.277120
70	472	68.001131
80	410	56.360000
90	156	37.335297

Tabell 7.8: BRISK-detektoren ved rotasjonsendring (10-90 grader). Bilde: Bilde1

Vinkel [grader]	Antall punkter	Tid [s]
10	151	0.28196
20	135	0.303382
30	154	0.330848
40	153	0.352702
50	133	0.342411
60	150	0.333041
70	137	0.308943
80	151	0.287208
90	110	0.240360

Tabell 7.9: SURF-detektoren ved rotasjonsendring (10-90 grader). Bilde: Bilde1

Vinkel [grader]	Antall punkter	Tid [s]
10	299	0.452702
20	165	0.489917
30	164	0.535299
40	169	0.573193
50	170	0.575710
60	166	0.548230
70	172	0.475130
80	296	0.441838
90	121	0.363352