




Universitetet  
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

## MASTEROPPGAVE

Studieprogram/spesialisering: Automasjon og Signalbehandling	Vårsemesteret, 2015  Åpen
Forfatter: Marius Nedrelid	 (signatur forfatter)
Fagansvarlig: Kjersti Engan Veileder(e): Kjersti Engan	
Tittel på masteroppgaven: Automatisk segmentering av LGE-CMR bilder ved bruk av graphcut  Engelsk tittel: Automatic segmentation of LGE-CMR images using GraphCuts	
Studiepoeng: 30	
Emneord: Digitalbildebehandling Segmentering GraphCuts	Sidetall: 50  + vedlegg/annet: zip fil med kildekode  Stavanger, 15.06.2015 dato/år



MASTER OPPGAVE

---

**Automatisk segmentering av  
LGE-CMR bilder ved bruk av  
Graphcut**

---

*Skrevet av*

Marius NEDRELID

15. juni 2015

---

## Sammendrag

Etter at en person har hatt hjerteinfarkt, blir det tatt Late Gadolinium Enhanced Cardiac Magnetic Resonance (LGE-CMR) bilder av pasientens hjerte. I disse bildene vil friskt muskelvev tilhørende hjertet fremstå som mørkt, mens arrvev vil fremstå som lysere. Dette gjør at konturene av indre og ytre hjertevegg (*endokard* og *epikard*) må finnes manuelt eller semi-automatisk av eksperter. Dette er tidkrevende arbeid og kan bære preg av forskjeller avhengig av personen som utfører arbeidet. Det er derfor ønskelig at denne segmenteringen kan automatiseres. Dette vil gjøre det mulig å behandle større datamengder på kortere tid.

I denne oppgaven skal segmenteringsmetoden graphcut vurderes. Målet med oppgaven er å utvikle en helautomatisk segmenteringsalgoritme for endokard og epikard i LGE-CMR bilder. Det skal gåes nærmere på hvordan graphcut fungerer, påkrevd initialisering og hvilke resultat som kan oppnås.

For å nå målet blir det benyttet en forbehandling av bildene før selve segmenteringen utføres. Forbehandling består av estimering av initielle konturer til endokard og epikard med et felles estimert hjertesenter. Dette er gjort slik at graphcutsegmenteringen tar utgangspunkt i de initielle konturene, og optimaliserer dem mot det aktuelle bildet.

For å teste algoritmen er det benyttet LGE-CMR bildesett av 54 pasienter, hvor alle tidligere har hatt hjerteinfarkt av varierende grad. Arrdannelsen på hjertemuskulaturen varierer i størrelse og plassering fra pasient til pasient. For noen er det et lite arr, og hjertemuskulaturen kan sees tydelig i bildene. For andre er det store gjennomgående arr som medfører at konturen av hjertemuskulaturen i noen områder forsvinner helt. Dette gjør at det er utfordrende data å jobbe med.

Resultatene blir målt i Dice Index, varierende fra 0 til 1, hvor 1 er identisk segmentering sammenlignet med korrekt segmentering, 0 er ingen overlapp. Gjennomsnittlig Dice Index av de initielle konturene etter forbehandlingen er 0.4833, med høyeste verdi på 0.7042 og laveste på 0.1891. Etter graphcutsegmenteringen er alle verdiene forbedret. Konklusjonen på oppgaven er at graphcut kan fungere til automatisk segmentering av bildene, men denne løsningsen er ikke godt nok til å kunne gi en løsning alene.

---

## Forord

Med denne oppgaven markeres slutten på slutten på studie på Universitetet i Stavanger. Jeg ønsker å takke forelesere for all kunnskap og hjelp de har bidrat med over disse to årene. Samt mine medstudenter som har vært med å gjøre dette til to artige år med et godt miljø.

Jeg ønsker også å takke min veileder Kjersti Engan for god oppfølging og veiledning gjennom hele prosjektet.

Takk for meg!



# Innhold

<b>1</b>	<b>Innledning</b>	<b>7</b>
<b>2</b>	<b>Bakgrunnsteori og datamateriell</b>	<b>9</b>
2.1	Litt om hjertet og datamaterialet . . . . .	9
2.1.1	Hjerteinfarkt . . . . .	9
2.1.2	CMR-bilder . . . . .	9
2.1.3	Datamaterialet . . . . .	10
2.2	Gradienter . . . . .	10
2.3	Korrelasjon . . . . .	11
2.4	Segmentering . . . . .	11
2.4.1	Terskelmetoden . . . . .	12
2.4.2	GraphCut . . . . .	13
<b>3</b>	<b>Innledende eksperiment</b>	<b>16</b>
3.1	Graphcut uten forbehandling . . . . .	16
3.2	Strekking av histogrammet til LGE-CMR-bildene . . . . .	18
<b>4</b>	<b>implementasjon</b>	<b>19</b>
4.1	Tilpassing av bildesettet . . . . .	20
4.2	Forbehandling . . . . .	21
4.2.1	Korrelasjon mellom mal og LGE-CMR-bilde . . . . .	21
4.2.2	Konvertering av mal til binærbilde . . . . .	23
4.2.3	Sjekk og korrigering av konturer . . . . .	23
4.3	Graphcut . . . . .	28
4.3.1	Kostnadsfunksjoner . . . . .	28
<b>5</b>	<b>Eksperimenter og Resultater</b>	<b>31</b>
5.1	Evalueringsmetode for segmenteringen . . . . .	31
5.2	Kostnadsfunksjoner . . . . .	35
5.2.1	Glatthetsbetingelse . . . . .	35

5.2.2	Datakostnad basert på avstand . . . . .	35
5.2.3	Datakostnad med kombinert avstand og gradient . . . . .	37
5.3	Beste segmentering . . . . .	41
<b>6</b>	<b>Drøfting</b>	<b>45</b>
6.1	Forbehandling . . . . .	45
6.2	Kostnadsfunksjoner . . . . .	45
6.3	Plassering av kilde og avløp . . . . .	46
6.4	min kut/maks flyt mot normalisert kutt . . . . .	47
<b>7</b>	<b>Konklusjon</b>	<b>47</b>
	<b>Appendices</b>	<b>50</b>
<b>A</b>	<b>Kildekode</b>	<b>50</b>
<b>B</b>	<b>Resultater</b>	<b>50</b>
B.1	Bilder . . . . .	50
B.2	Dice Index - eksperiment . . . . .	50

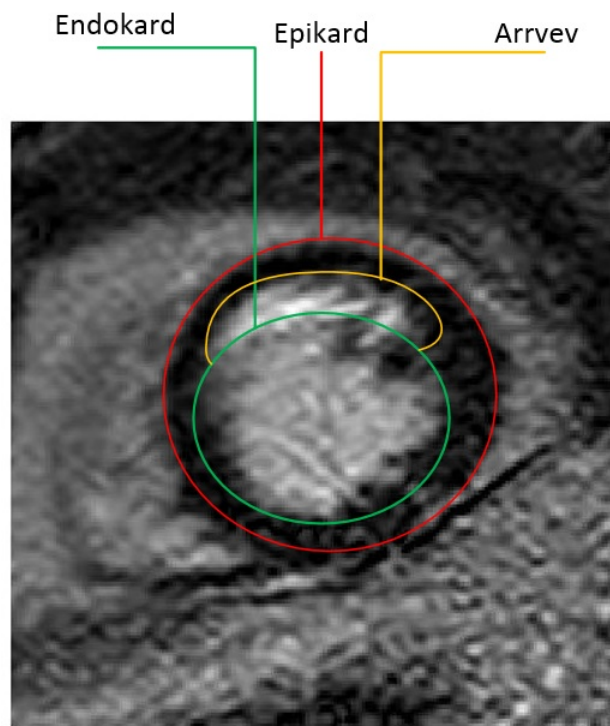
## Figurer

1	Plassering av endokard, epikard og arrvev i bildene . . . . .	7
2	Eksempel av LGE-CMR-bildesett . . . . .	8
3	Illustrasjon, gradient i x-retning . . . . .	11
4	Illustrasjon av korrelasjon mellom et bilde og en mal . . . . .	12
5	Illustrasjon av histogramsegmentering med terskel . . . . .	12
6	Eksempel på segmentering med terskelmetoden . . . . .	13
7	Eksempel på en segmentering med graphcut, figur er hentet fra [7] . . . . .	14
8	Resultat med GraphCut direkte på bildesett . . . . .	16
9	Resultat av GraphCut etter histogram utjamning . . . . .	18
10	Flytdiagram av prosessen . . . . .	19
11	LGE-CMR-bilder før (a) og etter sentrering og klipping (b) . . . . .	20
12	Illustrasjon av hvordan en beregnet mal . . . . .	21
13	Eksempel på korrelasjons resultat mellom bilde og mal . . . . .	22
14	Stegene i konverteringen fra mal til binærbilder . . . . .	23
15	Eksempel på feil og og korrekt plassering av mal fra høyeste korrelasjons verdi . . . . .	23
16	Klipping av bildestørrelse . . . . .	24
17	Maler før (a) og etter korrigering (b) . . . . .	26
18	Illustrasjon av tverrsnittet i et bildesett . . . . .	27
19	Beregning av avstand fra kontur til punkt . . . . .	29
20	Eksempel på Datakostnaden $D_c$ . . . . .	30
21	Initielle konturer funnet med korrelasjon . . . . .	33
22	Initielle konturer basert på den korrekte segmenteringen . . . . .	34
23	Resultat av kostnadsfunksjon basert på distanse for bildesett 2 . . . . .	36
24	Et utvalg av resultat fra bildesett 46 . . . . .	38
25	Eksperiment, funnet segmentering av bildesett 21 . . . . .	39
26	Illustrasjon av manglende konturer etter graphcut segmentering . . . . .	40
27	Beste segmentering funnet for bildesett 2 . . . . .	42

28	Beste segmentering funnet for bildesett 21 . . . . .	43
29	Beste segmentering funnet for bildesett 46 . . . . .	44
30	Illustrasjon av plassering av kilde og avløpsnoder i bildet . . . . .	46

# 1 Innledning

Etter et hjerteinfarkt dannes det arrvev på hjertemuskulaturen, som fører til redusert pumpeevne da arrvevet er stivere enn muskelvevet det har erstattet. Det tas derfor Late Gadolinium Enhanced Cardiac Magnetic Resonance (LGE-CMR) bilder av pasientens hjerte. I disse bildene vil en frisk hjertemuskel sees som en mørk sirkel (smultring formet). Kontrastvesken Gadolinium sprøytes inn for å fremheve arrvev på hjertemuskelene. I figur 1 er det illustrert hvor epikard og endokard er plassert, det er også vist hvordan arrvevet fremstår i et LGE-CMR bilde.



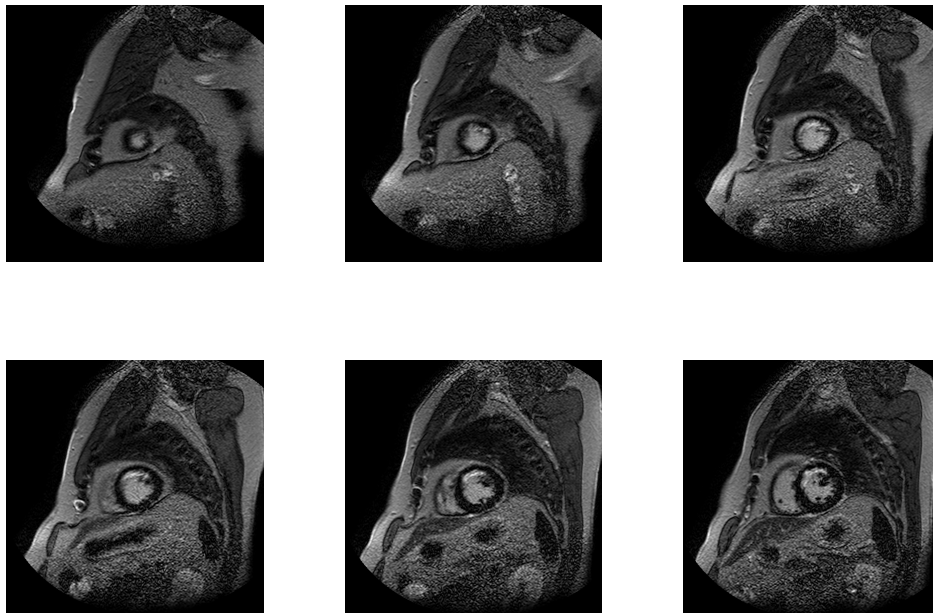
Figur 1: Plassering av endokard, epikard og arrvev i bildene

Siden arrvevet i bildet gjør at kontrasten mellom hjertemuskulaturen, blodet i hjertet og det myke vevet rundt blir dårligere eller forsvinner helt. Dette gjør at automatisk segmentering av LGE-CMR bilder blir vanskelig. Så konturene av hjertet må i dag finnes manuelt eller semiautomatisk med manuell input av eksperter[1]. Dette kan være tidkrevende arbeid og er derfor ønskelig at prosessen kan automatiseres.

Det har tidligere vært gjort forsøk for å automatisere denne prosessen, men de fleste metodene benytter en form for manuell input.

I denne oppgaven skal det sees på om graphcut segmentering, basert på min-kutt/maks-flyt algoritmer, kan benyttes til å automatisk segmentere LGE-CMR bilder. Algoritmen som er utviklet består av tre steg for hvert bilde.

- 1 → Estimering av initielle konturer tilhørende endokard og epikard, basert på kunnskaper om størrelse og form av hjertemusklatur.
- 2 → Kontroll og korrigering av de initielle konturene for hvert bilde i settet.
- 3 → Segmentering av endokard etterfulgt av epikard ved bruk av graphcut.



Figur 2: Eksempel av LGE-CMR-bildesett

## 2 Bakgrunnsteori og datamateriell

### 2.1 Litt om hjertet og datamaterialet

Hjertet er den muskelen som har som oppgave å pumpe blodet rundt i kroppen, slik at celler og vev får oksygenet og næringsstoffene de trenger for å fungere. Blodet transporterer også vekk avfallstoffer. Så det er viktig at hjertets pumpeevne opprettholdes, et hjerteinfarkt kan medføre at denne svekkes.

#### 2.1.1 Hjerteinfarkt

Et hjerteinfarkt forårsakes vanligvis av en blokkering i en av kranspulsårene i hjertet, som blokkerer åren delvis eller helt[2]. Dette medfører en akutt oksygenmangel til den delen av hjertemuskulaturen som får blod gjennom den blokkerte åren. Hvis hjertemuskelen ikke får blod og dermed oksygen begynner den å ta skade, hjerteceller dør og vil senere erstattes av arrvev hvis man ikke får rask behandling. Hvis arret blir for stort kan hjertes pumpeevne bli så svekket at blodtrykket ikke kan opprettholdes. Grunnen til dette er at arrvevet som dannes på hjertet ikke er like fleksibelt og dermed ikke kan trekkes sammen for å pumpe blodet ut av hjertet.

#### 2.1.2 CMR-bilder

Etter et hjerteinfarkt blir det tatt Cardiac Magnetic Resonance (*CMR*)-bilder av pasienten. I disse vil hjertet kunne sees som veldig mørkt, men for å fremheve arret på muskulaturen sprøytes det inn en kontrastveske kallet *Gadolinium*. Det ventes en gitt tid før det tas en ny *CMR*-bilde sekvens som kalles Late Gadolinium Enhanced *CMR* (*LGE-CMR*). Formålet med vesken som sprøytes inn, er at den trenger inn muskulaturen, men blir skyt fort ut igjen fra de delene med god blodsirkulasjon. De delene som består av arr har mindre blodgjennomstrømning, så det tar lenger tid før kontrastvesken blir skylt ut. Kontrastvesken vil i den nye *LGE-CMR*-bildeserien stå frem som lysere og indikere hvor det er dannet arr. Dette kan gjøre at kontrastene mellom hjertemuskulaturen, blodet i hjertet og det myke vevet rundt svekkes eller forsvinner helt. Dette gjør det vanskeligere å segmentere *endokard* og *epikard*. Segmenteringen må derfor gjøres manuelt eller

semi-automatisk av eksperter, noe som er tidkrevende arbeid.

### 2.1.3 Datamaterialet

Dataene som benyttes i denne oppgaven stammer fra Stavanger Universitets Sjukehus. Dataene er LGE-CMR bildesett av 54 forskjellige pasienter, som tidligere har hatt hjerteinfarkt før bildene er tatt. Av pasientene har 20 av disse så store arr at de senere har fått operert inn *ICD*<sup>1</sup>. Hvert av bildesettene består av mellom 6 og 11 bilder, totalt er det 433 LGE-CMR bilder. Bildene er horisontale tverrsnitt av overkroppen, tatt lagvis med en tykkelse på 10mm og med pikselstørrelse  $0.8 \times 0.8 \text{mm}^2$ [3]. Et eksempel på et slikt bildesett er vist i figur 2.

## 2.2 Gradienter

En gradient er i digital bildebehandling definert som lokale endringen i intensitet mellom pikslene i  $x$  og  $y$  retning. Endringen defineres av den partiellderiverte til bildet  $I$ .

$$\nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

Gradienten i  $x$  og  $y$  retning kan da skrives på følgende måte, hvor  $\Delta x$  og  $\Delta y$  er hvor stort sprang gradienten skal beregnes over.

$$\frac{\partial I(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{I(x + \Delta x, y) - I(x, y)}{\Delta x} \quad \frac{\partial I(x, y)}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{I(x, y + \Delta y) - I(x, y)}{\Delta y}$$

Dette uttrykket kan videre skrives over til  $G_x$  og  $G_y$  som er gradientene i  $x$  og  $y$  retning med sprang  $\Delta = 2$ .

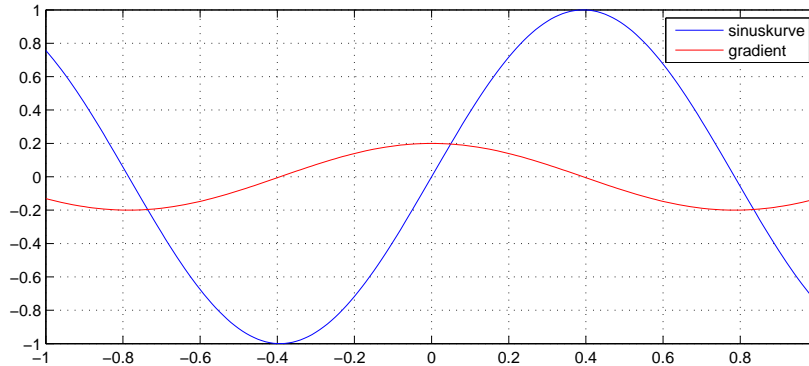
$$G_x(x, y) = \frac{I(x + 1, y) - I(x - 1, y)}{2} \quad G_y(x, y) = \frac{I(x, y + 1) - I(x, y - 1)}{2}$$

Formålet med en slik operator er å indikere hvilke og hvor det skjer intensitetendringer i et bilde. En slik operator er utsatt for støy i bildet om spranget  $\Delta$  er lite, med et større sprang er ikke operatoren like utsatt for støy, men endringene kommer da ikke like tydelig frem. I figur 3 er det vist en illustrasjon av en sinusbølge i  $x$ -retning med tilhørende gradient<sup>2</sup>.

<sup>1</sup>Implantable Cardioverter Defibrillator

<sup>2</sup>Beregnet med  $G_x$  og sprang  $\Delta x = 2$





Figur 3: Illustrasjon, gradient i x-retning

## 2.3 Korrelasjon

I digital bildebehandling brukes ofte korrelasjon som en metode for å lete etter likheter mellom et bilde og et mønster (*mal*). Dette er en metode som fungerer bra for både farge- og gråtone-bilder. La  $I(x,y)$  være et bilde og  $m(x,y)$  være mønsteret som skal finnes i bildet [4], så vil følgende gi hvor stor likhet det er mellom bilde og mal i i alle punkt.

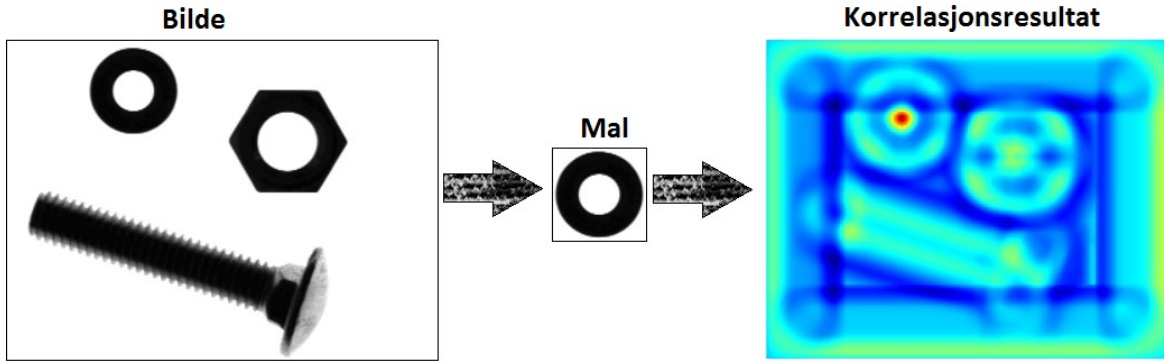
$$\rho_{bm}(x, y, p, q) = E[I(x, y)m(x + p, y + q)] = \sum_{p=0}^X \sum_{q=0}^Y I(x, y)m(x + p, y + q)$$

I figur 4 er korrelasjonen mellom et test bilde og en mal. I bildet letes det etter en sirkel som illustrert av malen. Resultatet av denne korrelasjonen, (vist til høyre i figuren) er rødt høyeste korrelasjonsverdi, altså den plasseringen av malen i bildet som gir høyeste likhet.

## 2.4 Segmentering

Segmentering er i digital bildebehandling prosessen å dele opp et bilde i meningsfulle regioner. I de enkleste tilfellene skal bildet deles inn i to segmenter, forgrunn og bakgrunn, mens i mange tilfeller kan det være ønskelig å dele bildet inn i flere segmenter. Prosessen tilegner hver piksel i bildet en etikett<sup>3</sup> som forteller hvilke segment pikselen tilhører. Målet med segmenteringen er å gruppere de pikslene som har like egenskaper og tilhører

<sup>3</sup>Etiketten er som regel et tall fra 0 til  $n - 1$ , hvor  $n$  er antall segment

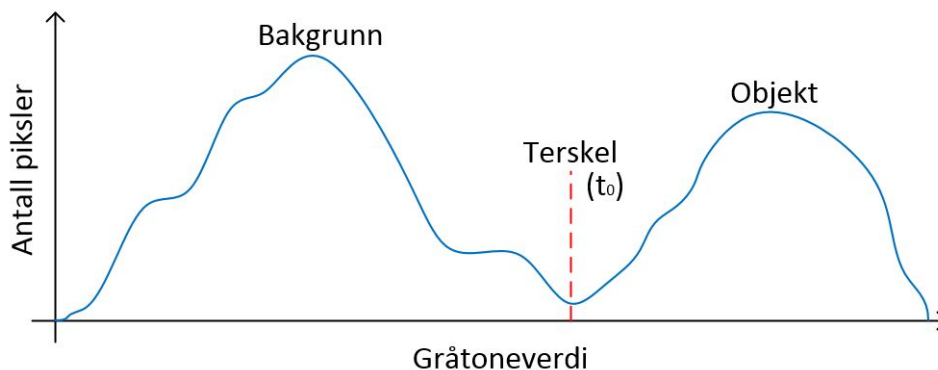


Figur 4: Illustrasjon av korrelasjon mellom et bilde og en mal

samme objekt slik at de havner i samme segment. Egenskapene kan f.eks. være intensitet eller fargenyanse. De forskjellige segmentene som bildet blir delt inn i skal samlet kunne gjengi hele det originale bildet [5].

#### 2.4.1 Terskelmetoden

Terskelmetoden er en av de enkleste segmenteringsmetodene innenfor digitalbildebehandling. Segmenteringen fungerer med at det settes en eller flere terskler i histogrammet til bildet, som setter skillene mellom segmentene. Gitt at histogrammet til et bilde har to distinkte topper, hvor den første er bakgrunnen og den andre er objektet. Terskelen settes da til laveste punkt mellom toppene, for å skille objektet fra bakgrunnen som illustrert i figur5.

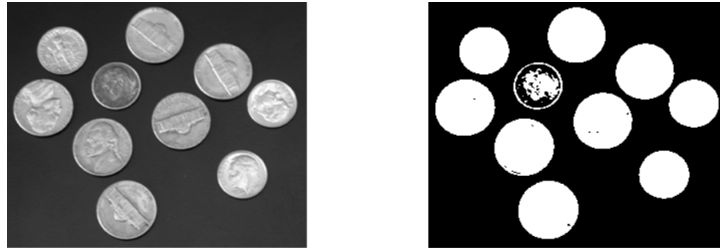


Figur 5: Illustrasjon av histogramsegmentering med terskel, figur er hentet fra [6]

Hvis piksel  $(x, y)$  har gråtoneverdi  $g$  høyere enn terskelen  $t_0$ , settes etiketten tilhørende pikselen til  $n = 1$ . Hvis  $g$  er lavere enn  $t_0$  settes etiketten til  $n = 0$ . Etikett  $n = 1$  indikerer at pikselen tilhører objektsegmentet og etikett  $n = 0$  indikerer at den tilhører bakgrunnsegmentet. Dette kan formuleres som:

$$n(x, y) = \begin{cases} 1 & \text{hvis } g \geq t_0 \\ 0 & \text{hvis } g < t_0 \end{cases} \quad (1)$$

En svakhet med denne segmenteringsmetoden er at deler av objekt kan gå tapt i segmenteringen. Dette kan skje hvis gråtoneverdiene inne i et av objektene ligger i samme område som bakgrunnen, som illustrert i figur 6.



Figur 6: Eksempel på segmentering med terskelmetoden på eksempelbilde fra matlab

### 2.4.2 GraphCut

Graphcut er et av de områdene som de siste årene har fått mye oppmerksomhet, til bruk i maskinsyn og bildebehandling. Ikke bare mot utvikling nye applikasjoner, men også i kombinasjon med andre algoritmer.

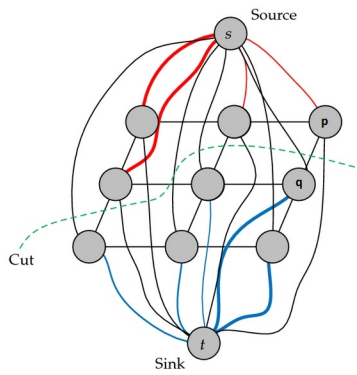
Gitt at en har en graf  $\mathcal{G}$  som består av et sett med noder<sup>4</sup>  $\mathcal{V}$  og koblingene  $\mathcal{E}$  mellom dem, en har da at  $\mathcal{G} = [\mathcal{V}, \mathcal{E}]$ . Settet med noder  $\mathcal{V}$  består av tre forskjellige nodetyper, to endepunkts noder kallet kilde  $s$ (source) og avløp  $t$ (sink) og de resterende nodene som ikke er endepunktnoder  $\mathcal{P}$  [7]. Koblingene kan deles inn i to forskjellige kategorier,  $t$ -kobling som er koblingene mellom en node i  $\mathcal{P}$  og en endepunktnode og  $n$ -kobling som er koblingene mellom to noder i  $\mathcal{P}$ .

<sup>4</sup>I digitalbildebehandling vil en node typisk korrespondere til en bestemt piksel i bildet

**Min kutt/Maks flyt**-algoritme

Alle koblinger mellom nodene er tilegnet en egen kostnad med verdi større enn 0, kostnaden kan variere avhengig om en ser på koblingen  $(p,q)$  eller omvendt  $(q,p)$  og kan skrives som  $w_{qp}$ . Hvis kostnaden mellom to noder er 0, indikerer det at der ikke er en kobling mellom nodene[8]. Formålet med graphcut er og finne det kuttet som deler nodene mellom kilde  $s$  og avløp  $t$  som har den minste kostnaden av alle mulig kutt. Når et kutt skal beregnes, er det kun den koblingen med lavest kostnad mellom de to nodene  $(q,p)$  eller  $(p,q)$  som blir medregnet i kostnaden for kuttet. Kostnaden for kuttet  $c(s,t)$  som deler nodene mellom kilde og avløp er formulert som:

$$c(s,t) = \sum_{q \in s, p \in t} w_{qp}$$



Figur 7: Eksempel på en segmentering med graphcut, figur er hentet fra [7]

**Normalisert kutt**-algoritme

Normalisert kutt er en algoritme som bygger på min kutt/maks flyt algoritmen. Algoritmen bygger videre på uttrykket for  $c(s,t)$ , og det normaliserte kuttet  $nc(s,t)$  kan skrives som:

$$nc(s,t) = \frac{c(s,t)}{c(s,\mathcal{V})} + \frac{c(s,t)}{c(t,\mathcal{V})}$$

Hvor  $nc(s,t)$  er den totale kostnaden til det normaliserte kuttet og  $c(s,t)$  er kostnaden til alle koblingene som er kuttet.  $c(s,\mathcal{V})$  er summen av alle koblinger i  $\mathcal{V}$  som berører  $s$ , og  $c(t,\mathcal{V})$  er summen av alle koblinger som berører  $t$ .

**Oppdatering av etiketter** Gitt at en har to etiketter  $\alpha$  og  $\beta$ , hvor  $\alpha$  er etiketten knyttet til kilden  $s$  og  $\beta$  knyttet til avløpet  $t$ . Det skal her ses på to forskjellige oppdateringsmetoder, som finner det kuttet som gir den laveste kostnaden,  $\alpha$ -**utviding** og  $\alpha, \beta$ -**bytting**.

$\alpha$ -utviding går ut på at en kan endre etikettene i  $\beta$  noder til  $\alpha$  noder. Gitt at  $\beta$  nodene er direkte koblet til en  $\alpha$  node via en  $t$ -*kobling* og endringen vil gi en lavere kostnad for kuttet. Nodene som har etiketten  $\alpha$  kan ikke endres til  $\beta$ . Denne prosessen blir gjentatt til det ikke er flere mulige endringer som vil gi en lavere kostnad for kuttet, altså metoden har konververt.

$\alpha, \beta$ -bytting går ut på at en forsøker alle mulige kombinasjoner mellom  $\alpha$  og  $\beta$  etiketten på de  $\alpha$  og  $\beta$  nodene som er sammenkoblet med en  $t$ -*kobling*. Dette gir muligheten for at både  $\alpha$  og  $\beta$  etikettene kan tilpasses for å finne det kuttet som gir minste kostnad. Denne prosessen gjentas til det ikke er noen endringer mellom de to etiketten som gir en lavere kostnad for kuttet.

### **Kostnadsfunksjon**

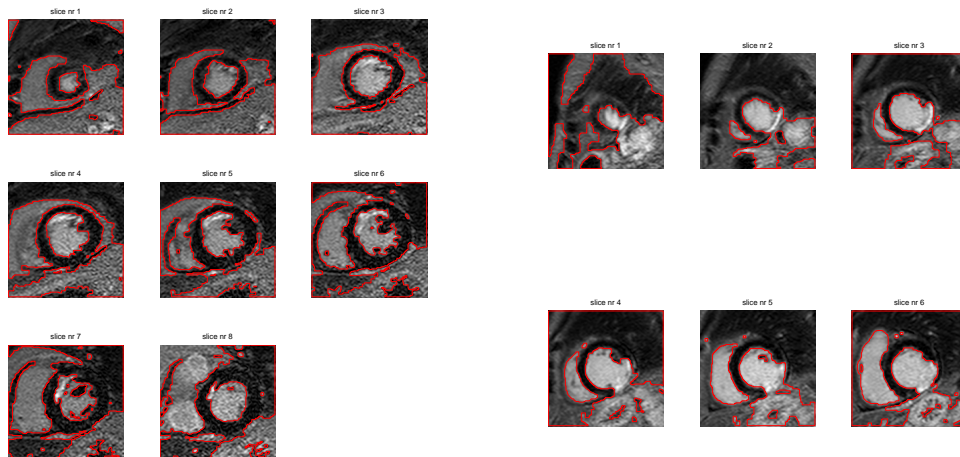
Utviklingen av en velegnet kostnadsfunksjon er av ytterste viktighet, da det er denne som bestemmer kostnaden satt til koblingene  $\mathcal{E}$  mellom nodene  $\mathcal{V}$ [8]. Kostnadsfunksjonen er ofte region og/eller gradientbaserte. Gradientene benyttes for å mer nøyaktig kunne tilnærme hvor kantene i bildet er og regionen bedrer den initielle modellen av hvor kuttet blir tatt.

### 3 Innledende eksperiment

I første del av prosjektet ble det gjort forskjellige eksperimenter på enklere testbilder med tilleggspakken for graphcut [14] i Matlab. Dette ble gjort for å bedre forståelsen av oppbygging, funksjonalitet og hvordan tilleggspakken best mulig kan benyttes. Det ble videre gjort forskjellige eksperimenter med tilleggspakken mot de forskjellige bildesettene. Det ble gjort eksperiment med bruk av graphcut direkte på bildesettene, altså uten noe form av forbehandling. Og med histogram operatører på bildesettene før graphcut, for å se hvordan dette innvirket på resultatene.

#### 3.1 Graphcut uten forbehandling

I figur 8 er segmenteringen funnet i to forskjellige bildesett hvor graphcut er benyttet direkte. I del (a) er endokard funnet, men arret på hjertet er tatt med i segmenteringen, og epikard har ikke hel kontur i alle bildene. Dette er feil som trolig kunne blitt korrigert med en etterbehandling av konturene. Største problemet med segmenteringen uten noen form for forbehandling er som vist i del (b), at kontrastene blir for små og konturene glir ut.



(a) Bildesett nr 2

(b) Bildesett nr 1

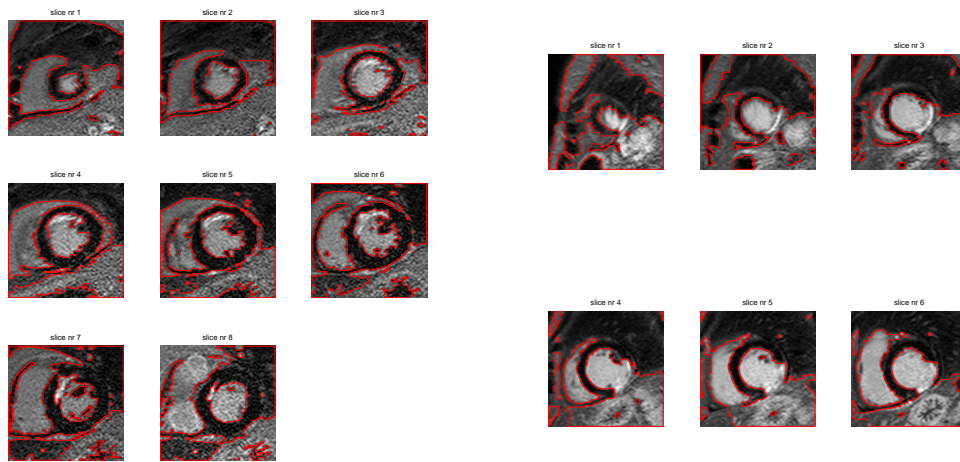
Figur 8: Resultat med GraphCut direkte på bildesett

Problemet med å ikke ha noe forbehandling er at det fungerer bare for noen få bildesett. Det er gjerne bildesett av pasienter med små arr slik at hjertemuskulaturen kommer tydelig frem i bildene. Så når arrene på hjertet blir store klare ikke graphcut direkte å skille mellom hjertemuskulaturen og det myke vevet rundt. Det som da skjer er at hjertekammeret, hjerteveggen og vevet rundt blir tolket som et og samme objekt i segmenteringen. Grunnen til dette er at pikselverdiene i området er like.

Av dette kommer det frem at det må utføres noe forbehandling på bildene, før graphcut kan benyttes på bildesettene. Dette for å bedre kunne sikre at både endokard og epikard får en mer korrekt kontur.

### 3.2 Strekking av histogrammet til LGE-CMR-bildene

Det ble eksperimentert med graphcut segmentering etter det var utført forbehandling på bildene. Formålet med denne forbehandlingen var øke kontrasten i bildene, håpet med dette var at skillet som indikerer hjerteveggene i bildet skulle bli mer utpreget. Til dette ble det benyttet histogramutjevning. Dette blir gjort med at i områder av histogrammet hvor det er høy tetthet av piksler, strekkes disse utover et større spekter i histogrammet. Resultatet er at kontrastene øker og det blir større skille mellom de gråtonene.



(a) Bildesett nr 2

(b) Bildesett nr 1

Figur 9: Resultat av GraphCut etter histogram utjamning

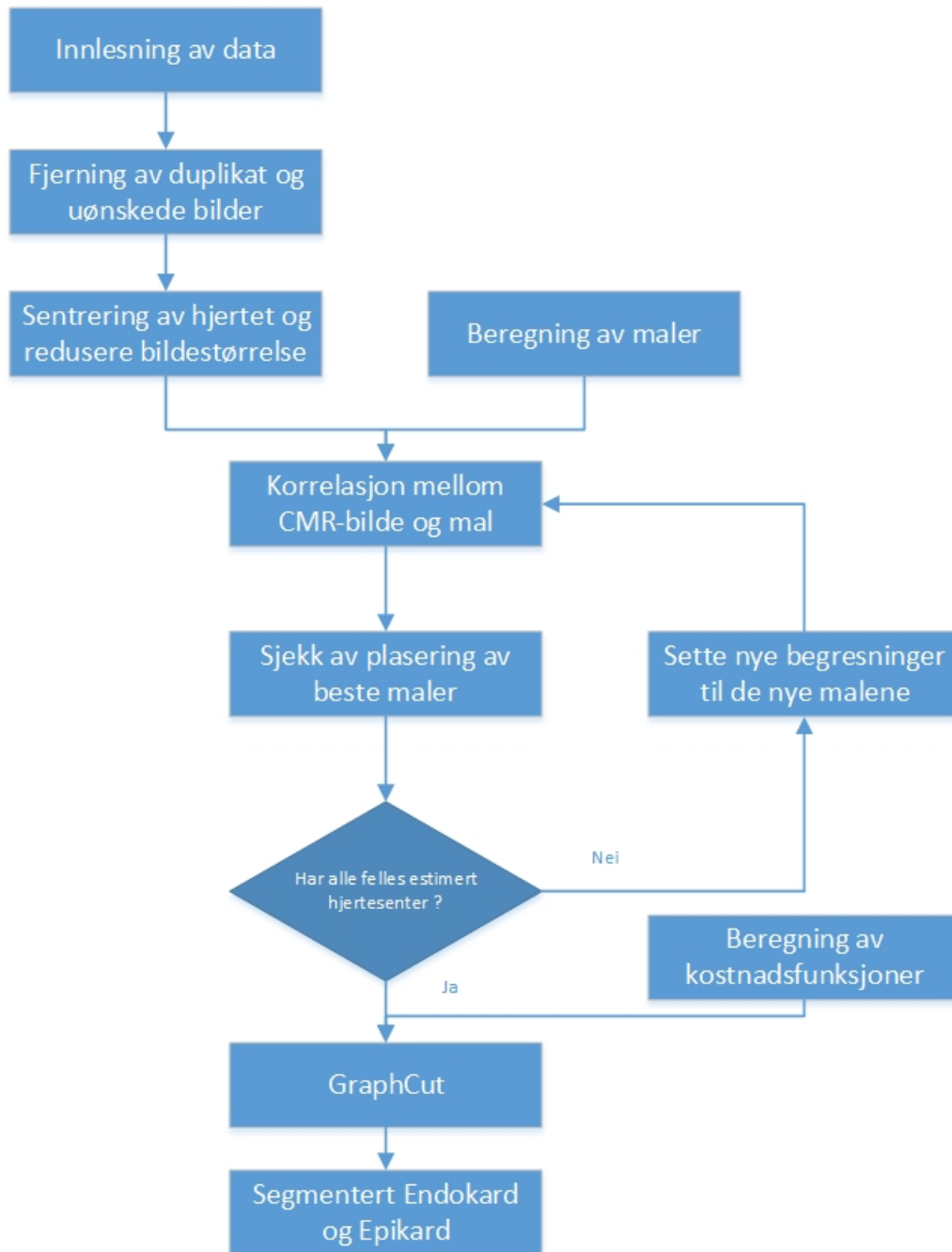
Resultatet av denne forbehandlingen er vist i figur 9. Som vist, så har ikke forbehandlingen gjort et stort nok utslag på segmenteringen funnet ved graphcut til at dette alene kan benyttes.

Konklusjonen basert på de innledende eksperimentene er at en bedre initialisering må beregnes. Den initialiseringen er i form av estimerte initielle konturer av endokard og epikard.



## 4 implementasjon

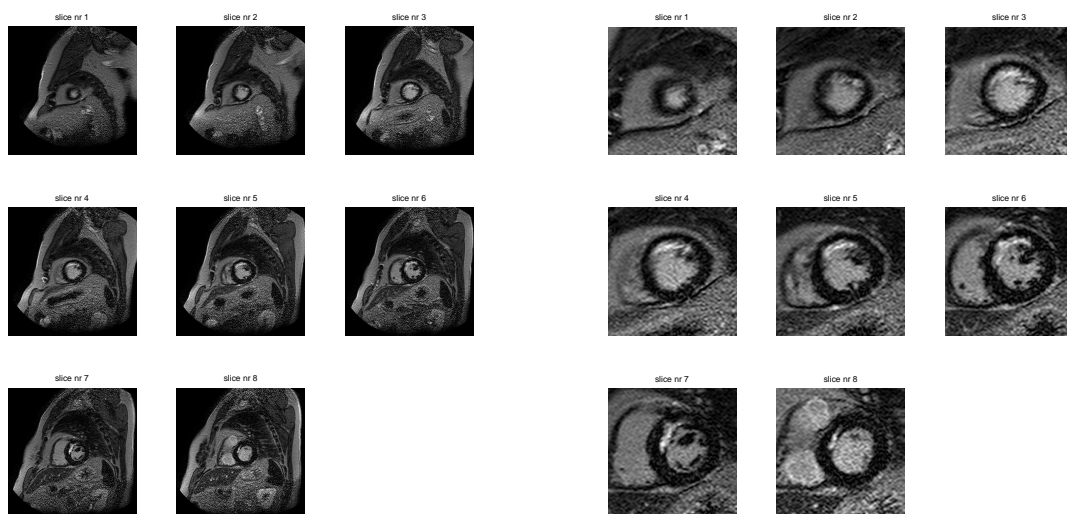
I dette kapitlet skal det sees nærmere på hvordan algoritmen er bygget opp. Fra innlesning og organisering av bildesettene, frem til de endelige konturene av endokard og epikard funnet ved graphcut segmentering.



Figur 10: Flytdiagram av prosessen

## 4.1 Tilpassing av bildesettet

Første steget i prosessen er at bildene sorteres i korrekt rekkefølge, slik at første bildet er bunnen av hjertet og videre oppover mot toppen av hjertemuskulaturen. Her fjernes også eventuelle duplikater av bildene, bilder som ikke inneholder både endokard og epikard fjernes også fra bildesettet. Dette blir gjort slik at bildesettene blir sendt ut som vist i figur 11 (a). Når dette er gjort blir alle bildene reduserte i størrelse med utgangspunkt i et estimert hjertesenter, resultatet av dette blir som vist i figur 11 (b).



(a) Original størrelse på LGE-CMR-bildene (b) Sentrert hjerte og redusert bildestørrelse

Figur 11: LGE-CMR-bilder før (a) og etter sentrering og klipping (b)

Etter dette skal alle bildesettene være sorterte og nummerert i stigende rekkefølge, fra bunnen og opp med hjertet sentrert i bildet[9]. Funksjonene for innlesning, organisering og sentrering er skrevet og tilsendt av Kjersti Engan. Videre normaliseres alle bilder slik at de har verdier mellom  $[0, 1]$ , dette gjøres slik at all videre behandling skal ha likest mulig utslag på de forskjellige bildene i settet.

## 4.2 Forbehandling

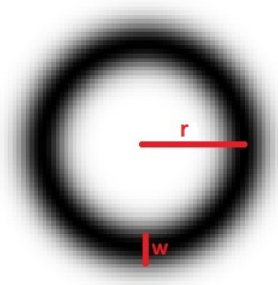
Fra de innledende eksperimentene kommer det frem at estimater av hjerteveggene må beregnes for alle bildene i bildesettet. Tanken er da at estimatene kan benyttes som initielle konturer i graphcut segmenteringen. Det er derfor ønskelig at de initielle konturene ligger i samme område som de faktiske hjerteveggene. Graphcut segmenteringen skal tilpasse hver av konturene mot endokard og epikard i hvert bilde.

Den grunnleggende fremgangsmåten er inspirert av arbeidet gjort i publikasjon [10] av Xènia Alba.

### 4.2.1 Korrelasjon mellom mal og LGE-CMR-bilde

#### Mal

I LGE-CMR bildene fremstår formen av hjertemuskelen som en smultring, av den grunn blir det beregnet sirkelformede maler. Malene er bygget på normalfordelt gaussisk funksjoner, med radius  $r$  og bølgebredde  $w$  som illustrert i figur 12.



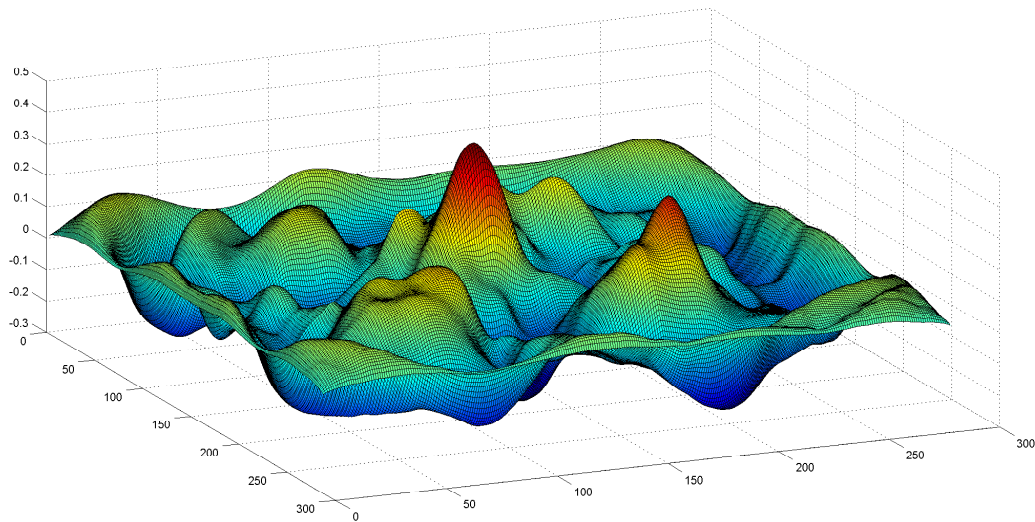
Figur 12: Illustrasjon av hvordan en beregnet mal med radius  $r$  og bølgebredde  $w$

Det genereres totalt 100 slike maler, med kombinasjoner av 10 forskjellige radier og 10 forskjellige bølgebredder. Størrelsen på malene er beregnet med kunnskap om at hjertet er mindre i nedre del enn det er i øvre. I bilde en til tre brukes lineært fordelte radier i området  $r = [17, 50]$  og i bildene over benyttes  $r = [30, 60]$ . I alle bildene benyttes bølgebreddene  $w = [5, 21]$  da denne er mer lik mellom alle bildene i hvert sett<sup>5</sup>.

<sup>5</sup>Alle radier og bølgebreddeverdier er oppgitt i antall piksler

## Korrelasjon

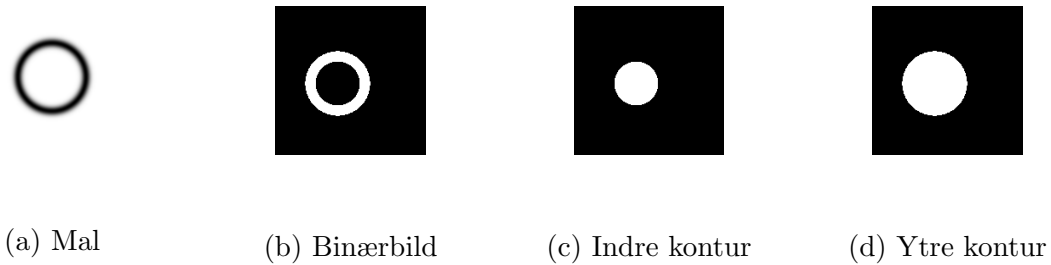
For hvert bilde berregnes korrelasjonene mellom bildet og alle malene. Blant disse resultatene så letes det etter beste korrelasjonen, det vil si høyeste verdi. De resterende korrelasjonene som har en lavere maks verdi forkastes. Grunnen til dette er at malen som gir den beste korrelasjonen, skal i utgangspunktet gi de beste initielle konturene til bildet. Fra den malen så lagres radiusen  $r$  og bølgebredde  $w$  for videre bruk, samt koordinatene  $[x_{max}, y_{max}]$  til punktet høyeste verdien ble funnet.



Figur 13: Eksempel på korrelasjons resultat mellom bilde og mal

### 4.2.2 Konvertering av mal til binærbilde

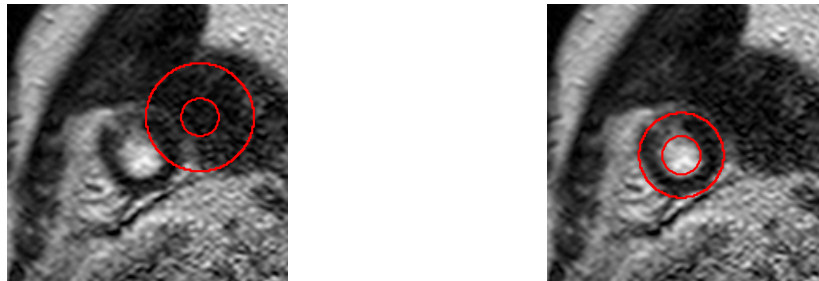
Malene som har gitt det beste korrelasjons resultatene for hvert av bildene, konverteres til et binærbilde i samme størrelse som det tilhørende LGE-CMR-bildet. I binærbildet plasseres malen med senterpunkt i  $[x_{max}, y_{max}]$ . Formålet med binærbildet er at det skal brukes som initialisering til graphcut metoden. Binærbildet blir delt i to nye, et av den indre- og et av den ytre-konturen (se (c) og (d) i figur 14). Hvor den indre skal benyttes mot endokard og den ytre skal benyttes mot epikard.



Figur 14: Stegene i konverteringen fra mal til binærbilder

### 4.2.3 Sjekk og korrigering av konturer

Siden det ikke er sikkert at den høyeste verdien funnet fra korrelasjonene stemmer overens med hjertet i bildet, må dette kontrolleres. I figur 15 er det vist et eksempel på feil og korrekte initiell konturer av endokard og epikard funnet med beste korrelasjon.



Figur 15: Eksempel på feil og korrekt plassering av mal fra høyeste korrelasjons verdi

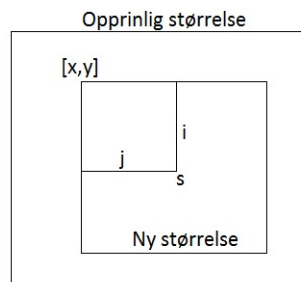
I korrigeringen blir det tatt forbehold om at hjertet skal være noenlunde sentrert i alle bildesettene og hjertesenteret skal være det samme i alle bildene for hvert sett.

### Sjekk av maler

Videre beregnes medianen av senterpunktet for alle konturene i hvert bildesett, dette blir gjort for å finne koordinatene  $[X_{med}, Y_{med}]$  til det estimert hjertesenter. Hver av konturene blir så sjekket mot dette senterpunktet, hvis avstanden er mindre enn  $\pm 16$  piksler i x og y retning anses konturen som rett. Når alle konturene er sjekket mot det estimerte hjertesenter, kontrolleres det hvor mange av malene som er ansett som rett. Hvis over 60% av malene er feil, så kan ikke det estimerte hjertesenteret antas som rett. Hvis ikke kan en anta at det estimerte hjertesenteret er korrekt og en kan forsette til korrigering av de konturene som er antatt feilplasserte

### Klipping av bildestørrelse

I tilfellene hvor en ikke kan anta at det estimerte hjertesenteret stemmer så klippes størrelsen på bildene i bildesettet ned. Dette blir gjort med utgangspunkt i det feilaktige hjertesenteret og de største malene som ble funnet, dette er illustrert i figuren under. Hvor  $S = [X_{med}, Y_{med}]$  og  $i, j = r_{maks} + w_{maks} + 10$ . Dette brukes til å beregne hvor øvre venstre hjørne  $[x, y]$  blir satt og bredde og høyde på bildet. En har da at koordinaten til hjørnet  $[x, y] = [S_x - j, S_y - i]$  og bredden  $\times$  høyden på det klippede bildet som blir  $(2 \times j) \times (2 \times i)$ . Etter bildene er blitt klippet ned i størrelse, starter prosessen på nytt med korrelasjon mellom de klippede bildene og malene.



Figur 16: Klipping av bildestørrelse

## Korrigerings

Når det kan antas at det estimerte hjertesenteret er korrekt, kan en forsette med korrigeringen av de feilplasserte konturene. Det første som blir gjort er å finne en øvre og nedre grense, disse blir definert i en regions vektor  $[X_{høy}, X_{lav}, Y_{høy}, Y_{lav}]$ .  $X_{høy}$  er høyeste og  $X_{lav}$  er laveste x-koordinat blant senterpunktene til de malene som er ansett som rett. Det samme gjelder y-koordinatene for å finne  $Y_{høy}$  og  $Y_{lav}$ . Selve korrigeringen blir gjort med en ny korrelasjon mellom LGE-CMR-bildet og et sett med nye maler. De nye malene blir begrenset av radiusen og bølgebredde i de nærmeste korrekte malene. Med dette finner en den malen som passer best med det estimerte hjertesenteret og bildet det tilhører. Rekkefølgen feilene korrigeres i er delt inn i tre steg, hvor hvert har sine sett med begrensninger for radius og bølgebredde for de nye malene. Stegene er forklart under:

Steg 1 Ingen korrekt kontur i bilde(ne) under.

↔ Første del som skal korrigeres er bunnen av hjertet, dette steget kan hoppes over om bilde 1 har en mal som er ansett som rett. Dette steget omfatter alle bildene fra bilde 1 og opp til første bildet som har en mal som er ansett som rett. Hvis bilde  $n$  er første bilde som har korrekt mal, så er den første malen som korrigeres i bilde  $n - 1$ . De nye malene blir begrenset av radius  $r(n)$  og bølgebredde  $w(n)$  i malen tilhørende bilde  $n$ . Slik at malene som det skal korreleres med bildet er mindre eller lik den begrensende malens parametere. En får da at malene får radier  $r = [17, r(n)]$  og bølgebredde  $w = [5, w(n)]$ . For hver korrigerings oppdateres  $n$  til nederste korrekte mal, når  $n = 1$  går funksjonen videre til steg 2.

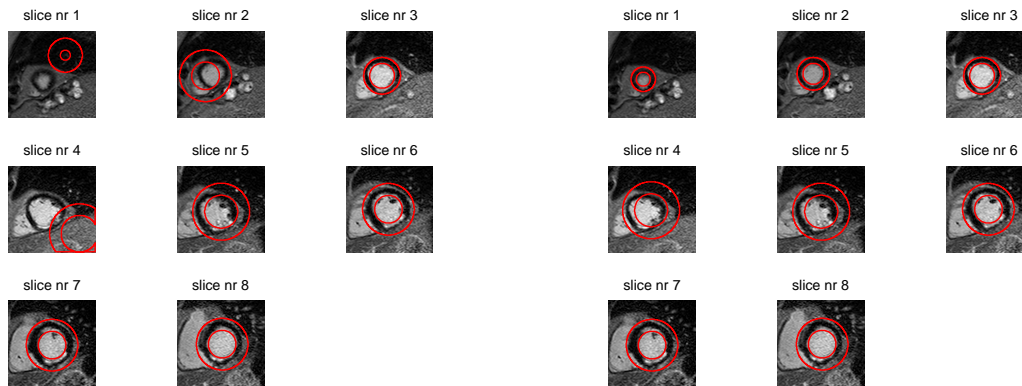
Steg 2 Ingen korrekt kontur i bilde(ne) over.

↔ Den neste delen som skal korrigeres er toppen av hjertet, dette steget kan hoppes over om det øverste bildet i settet har en mal som er ansett som rett. Dette steget omfatter alle bildene fra øverste bilde og nedover til de øverste bildet som har en mal som er ansett som rett. Hvis bilde  $n$  er øverste bilde som har en mal som er ansett som rett så er det bilde  $n + 1$  som korrigeres først. De nye malene blir begrenset av radius  $r(n)$  og bølgebredde  $w(n)$  i malen tilhørende bilde  $n$ , slik at malene som skal korreleres med bildet er større eller lik den begrensende malens parametere.

En får da at malene får radier  $r = [r(n), 60]$  og bølgebredder  $w = [w(n), 21]$ . For hver korrigerings oppdateres  $n$  til øverste korrekte mal, når  $n$  er lik bilde nummeret til øverste bildet i settet går funksjonen videre til steg 3.

Steg 3 Korrekte konturer i bilde over og under.

↔ Etter steg en og to gjenstår kun feil som har korrekte maler både i bilder over og under. I dette steget begynner funksjonen med å korrigere inn den nederste malen som er ansett som feil for og jobbe seg oppover igjennom settet til alle er korrekte. Hvis bilde  $n$  er bildet under med korrekt mal med parametre  $r(n)$  og  $w(n)$ . Bilde  $m$  er første bildet over med korrekt mal med parametre  $r(m)$  og  $w(m)$ . Så settes begrensningen for malene som skal korreleres med bildet til  $r = [r(n), r(m)]$  og  $w = [w(n), w(m)]$ .



(a) Maler funnet fra første korrelasjon

(b) Maler funnet etter korrigerings

Figur 17: Maler før (a) og etter korrigerings (b)





### 4.3 Graphcut

Segmentering blir gjort i tre steg for hvert bilde. Først finnes segmenteringen av endokard, basert på den beregnede indre konturen tilhørende bildet. Etter dette finnes epikard segmenteringen basert på den ytre konturen tilhørende bildet. Etter begge segmentene er funnet slås konturene sammen til en felles.

Til selve graphcut segmenteringen benyttes en utvidelsepakke for matlab[11][12][13][14]. For at utvidelsen skal kunne benyttes så må det defineres en datakostnad, glatthetsbetingelse og ønsket metode.

Metoden som er blir benyttet er  $\alpha, \beta$ -bytting (som forklart i kapittel 2.4.2), da denne tillater at de initielle konturene kan korrigeres både innover og utover. Noe som  $\alpha$ -utviding ikke har muligheten til, da den kun kan utvide den initielle konturen.

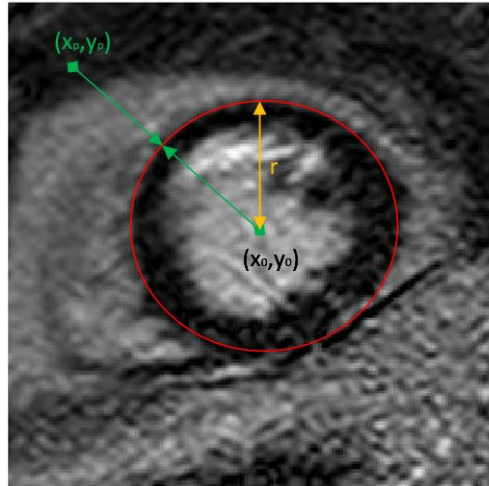
De initielle konturene er valgfritt om de skal benyttes, men basert på de innledende eksperimentene anbefales dette.

#### 4.3.1 Kostnadsfunksjoner

De to forskjellige kostnadsfunksjonene som må settes før graphcut segmenteringen er en *datakostnad* og en *glatthetsbetingelse*.

**Datakostnaden** blir beregnet som en vektet kombinasjon av avstand og gradient for alle punkter i bildet. Avstanden  $d$  blir beregnet for alle punkt i bildet, og er avstanden fra punktet til nærmeste punkt på den initielle konturen (se figur 19). For et gitt punkt  $(x_p, y_p)$  er avstanden beregnet på følgende måte:

$$d(x_p, y_p) = \left| \sqrt{(x_p - x_0)^2 + (y_p - y_0)^2} - r \right|$$



Figur 19: Beregning av avstand fra kontur til punkt

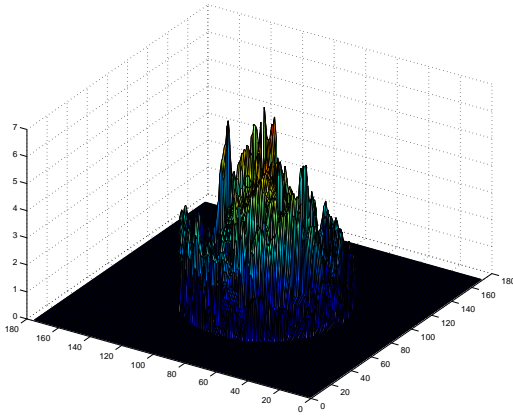
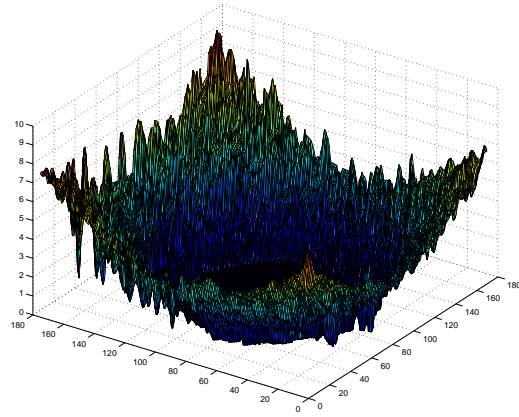
Gradienten som benyttes i datakostnaden behandles forskjellig avhengig av om det er endokard eller epikard som skal segmenteres. Det er også avhengig av hvor punktet  $(x_p, y_q)$  er i bildet, sett mot bredden  $X$  og høyden  $Y$  av bildet, som vist i tabellen under.

Endocard	Vertikal gradient	
Posisjon $(x_p, y_p)$	Positiv	Negativ
$y_p \leq \frac{Y}{2}$	+	-
$y_p > \frac{Y}{2}$	-	+
	Horisontal gradient	
$x_p \leq \frac{X}{2}$	+	-
$x_p > \frac{X}{2}$	-	+

Epicard	Vertikal gradient	
Posisjon $(x_p, y_p)$	Positiv	Negativ
$y_p \leq \frac{y}{2}$	-	+
$y_p > \frac{y}{2}$	+	-
	Horisontal gradient	
$x_p \leq \frac{x}{2}$	-	+
$x_p > \frac{x}{2}$	+	-

I tabellen er det fortegnet til gradienten i punktet  $(x_p, y_p)$ , om det er positivt eller negativt som avgjør om den skal legges til eller trekkes fra i datakostnaden.

Både avstanden og gradienten er vektet med to forskjellige faktorer. Hvor  $a$  er vekt faktoren knyttet til avstanden og  $b$  er vekt faktoren til gradienten. Kostnadene må så knyttes til alle koblingene mellom de forskjellige nodene. Av initialiseringen er det satt at alle nodene innenfor den initielle konturen tilhører  $\alpha$  etiketten og alle utenfor tilhører  $\beta$ . Siden  $\alpha$  er assosiert med kilden (objektet) i bildet. Av dette så settes alle kostnadene innenfor den initielle konturen til  $\beta$  og alle utenfor til  $\alpha$ . En ender da opp med en datakostnad  $Dc$  matrise, som får følgende format:  $Dc(X, Y, n)$ , hvor  $n$  er de forskjellige etikettene.

(a)  $Dc(X, Y, \beta)$ (b)  $Dc(X, Y, \alpha)$ Figur 20: Eksempel på Datakostnaden  $Dc$ 

**Glatthetsbetingelsen** som må settes er en  $n \times n$ -matrise. Elementene i matrisen forteller hvilke kostnad som skal assosieres mellom de forskjellige etikettene. Matrisen  $Sc$  settes normalt til en matrise med 1 i offdiagonalen[14], før den eventuelt manuelt tilpasses senere.

$$Sc = \begin{bmatrix} \beta \rightarrow \beta & \beta \rightarrow \alpha \\ \alpha \rightarrow \beta & \alpha \rightarrow \alpha \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

## 5 Eksperimenter og Resultater

### 5.1 Evalueringsmetode for segmenteringen

For å finne den mest korrekte segmenteringen av hjertemuskulaturen. Er det betraktet flere kombinasjoner av vekt faktorer i kostnadsfunksjonen og forskjellige glatthetsbetingelser. Til å måle hva som gir de beste resultatene brukes en funksjon for utregning av Dice Index ( $DI$ ). Funksjonen regner ut hvor mye som er felles for segmenteringen funnet ved graphcut og segmentet som er gitt som korrekt for hvert bilde i settet. I formelen under er  $A$  segmenteringen fra graphcut som skall sjekkes mot de korrekte segmentene  $B$ <sup>6</sup>.

$$DI = \frac{2|A \cap B|}{|A| + |B|}$$

Resultatet av  $DI$  er et tall mellom 0 og 1, hvor 0 vil si at de to segmentene ikke har noe felles og 1 vil si at de er identiske.

Det blir i tillegg for hvert bildesett, beregnet initielle konturer basert på den gitte korrekte segmenteringen. Dette blir gjort for å kunne sammenligne hvordan de forskjellige vektene i kostnadsfunksjonen virker på konturer plassert i det sanne hjertesenter. Disse konturene har en radius basert på gjennomsnittet av radiusen til epikard for hvert bilde. Bølgebredden er basert på gjennomsnittet av differansen mellom epikard og endokard konturene for hvert bilde.

Videre vil notasjonen  $bi$  angi de initielle konturene funnet med forbehandlingen, og  $ki$  angi de initielle konturene beregnet fra den korrekte segmenteringen. For målingene vil  $DI_{init}$  angi dice index av de initielle konturene  $bi$  og  $DI_{gc}$  angi dice index etter segmenteringen funnet ved graphcut. Det blir også testet mot de initielle konturene  $ki$ , som har benevning  $DI_{ki}$  og etter graphcut segmenteringen  $DI_{kgc}$ .

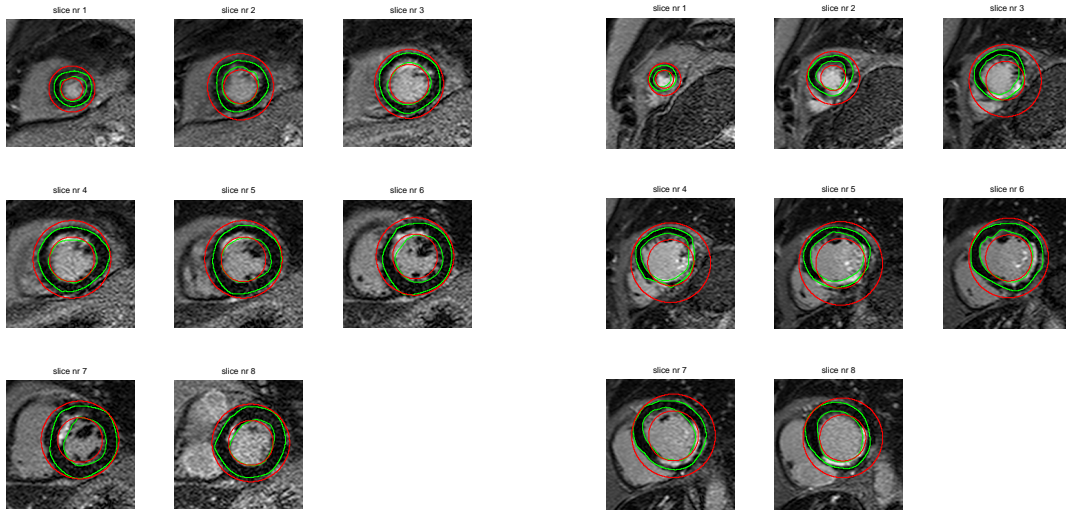
---

<sup>6</sup>Funksjonen for utregning av Dice Index er skrevet av Kjersti Engan

For alle eksperiment vil gjennomsnittlig dice index mellom alle bildesett oppgis, beregnet på gjennomsnittet i hvert av bildesettene. Det blir også sett nærmere på tre utvalgte bildesett. Det velges ut det bildesettet med høyest og lavest dice index av de initielle konturene  $DI_{init}$  og det har median verdien. Så videre vil det ses nærmere på bildesett 2, 21 og 46.  $DI_{init}$  for sett 2 er den høyeste verdien, sett 21 er medianene og sett 46 har den laveste.

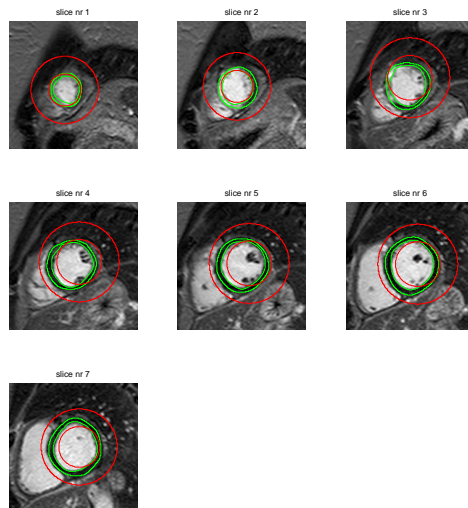
	Sett 2	Sett 21	sett 46	Gjennomsnitt
$DI_{init}$	0.7042	0.4884	0.1891	0.4833
$DI_{ki}$	0.8185	0.7989	0.7887	0.7865

I figur 21 er de initielle konturene som er funnet med forbehandlingen vist som *røde* og de gitte korrekte konturene er vist som *grønne*. Figuren viser disse for alle de tre bildesettene som skal betraktes videre. Det er dice indexen mellom disse som gjør at det er disse tre som blir sett på videre.



(a) Bildesett 2

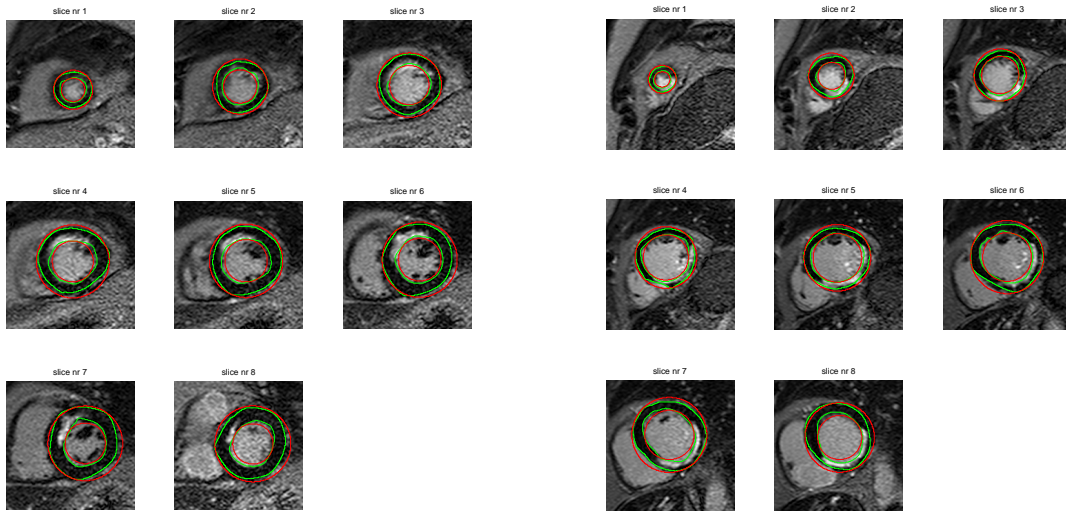
(b) Bildesett 21



(c) Bildesett 46

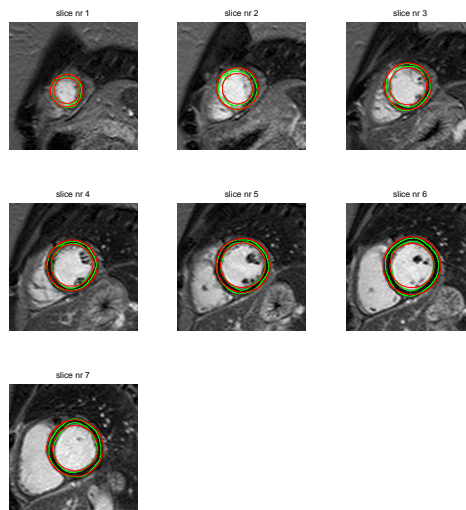
Figur 21: Initielle konturer funnet med korrelasjon

De initielle konturene som er beregnet av den korrekte segmenteringen er vist i figur 22. Som nevnt er dette ment som sammenligningsdata for resultatene med de forskjellige vektene.



(a) Bildesett 2

(b) Bildesett 21



(c) Bildesett 46

Figur 22: Initielle konturer basert på den korrekte segmenteringen



## 5.2 Kostnadsfunksjoner

### 5.2.1 Glatthetsbetingelse

Det er valgt ut tre forskjellige glatthetsbetingelser som brukes videre i eksperimentene.  $Sc_1$  er valgt utfra at det er den som brukes i eksempelet laget for å demonstrere bruken av graphcut pakken [14].  $Sc_2$  og  $Sc_3$  er satt for å se hvilke utslag endringer i glatthetsbetingelsen vil ha på segmenteringen.

$$Sc_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Sc_2 = \begin{bmatrix} 0.5 & 1 \\ 1 & 0.5 \end{bmatrix} \quad Sc_3 = \begin{bmatrix} 0.5 & 0.6 \\ 0.6 & 0.5 \end{bmatrix}$$

### 5.2.2 Datakostnad basert på avstand

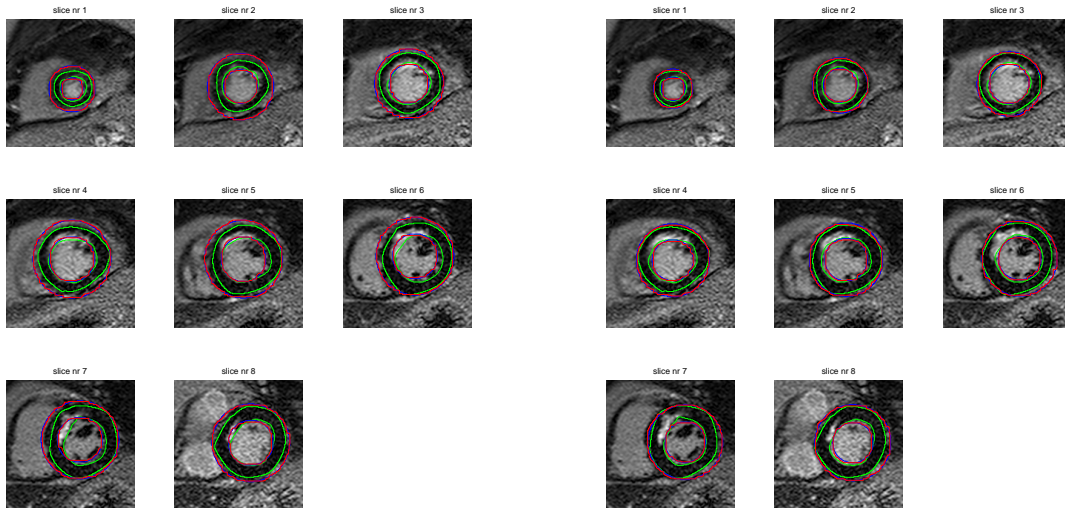
Det første eksperimentet ble gjort med en regionbasert datakostnad. Altså en datakostnad som er basert på **avstand**, dette gjort som et forsøke på å tvinge graphcut segmenteringen til å beholde en mest mulig sirkulær form. Kostnadsfunksjonen som ble brukt er som følger, med vekt  $a = 1$ .

$$Dc(x, y, n) = a \times d(x, y)$$

I tabellene under er  $DI$  av graphcut segmenteringen med denne kostnadsfunksjonen fremstilt.

$a = 1$ $Sc_1$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7068	0.0026	—	0.8185	0.8216	0.0031
Sett 21	0.4884	0.4998	0.0114	—	0.7989	0.8057	0.0068
Sett 46	0.1891	0.1990	0.0099	—	0.7887	0.7701	-0.0186
Gjennomsnitt	0.4833	0.4923	0.0090	—	0.7865	0.7824	-0.0041
$a = 1$ $Sc_2$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7043	0.0003	—	0.8185	0.8209	0.0024
Sett 21	0.4884	0.4956	0.0072	—	0.7989	0.8016	0.0027
Sett 46	0.1891	0.1945	0.0054	—	0.7887	0.8209	0.0322
Gjennomsnitt	0.4833	0.4886	0.0053	—	0.7865	0.7869	0.0004

$a = 1$ $Sc_3$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7031	-0.0011	—	0.8185	0.8197	0.0012
Sett 21	0.4884	0.4888	0.0004	—	0.7989	0.7991	0.0002
Sett 46	0.1891	0.1904	0.0013	—	0.7887	0.7933	0.0046
Gjennomsnitt	0.4833	0.4844	0.0011	—	0.7865	0.7871	-0.0006

(a) *bi*(b) *ki*

Figur 23: Resultat av kostnadsfunksjon basert på distanse for bildesett 2

Et problem med å benytte en ren avstandbasert kostnadsfunksjon, er at kostnaden assosiert med endringer av kuttet øker raskt når kuttet flyttes vekk fra den initielle konturen. Dette gjør at kuttet med laveste kostnad blir funnet i umiddelbar nærhet av de initielle konturene. Dette kommer frem i figur 23, hvor den røde konturen såvidt er endret fra den blå. I figuren er blå den initielle konturen, rød er segmenteringen funnet ved graphcut og grønn er den gitte korrekte segmenteringen. Dette er fargene som blir benyttet videre i alle figurer som illustrerer segmenteringen.

Dette tolkes til at datakostnaden eventuelt må kombineres med annen informasjon i bildene. Tanken er å ikke straffe avstand like mye, slik at en kuttet kan flyttes mer i for-

hold til de initielle konturene. Men det er fortsatt ønskelig at segmenteringen skal holde en tilnærmet sirkulær form.

### 5.2.3 Datakostnad med kombinert avstand og gradient

Etter eksperimentet med en datakostnad basert kun på avstand, er det eksperimentert med en datakostnad basert på **avstand** kombinert med **gradientene** i bildene. Dette ble gjort med formening om at graphcut segmenteringen mer fritt skulle kunne finne et bedre kutt. Slik at kuttet da kan flyttes i større grad vekk fra de initielle konturene, men innenfor gitte begrensninger. Begrensninger som er satt i form av avstand og tolkning av gradientene som forklart i kapittel 4.3.1. Det benyttes da en kostnadsfunksjon som er både region og gradientbasert. Det er utført eksperimenter med forskjellige vektorer av avstanden  $a$  og gradientene  $b$ . Datakostnaden er uttrykket som følger

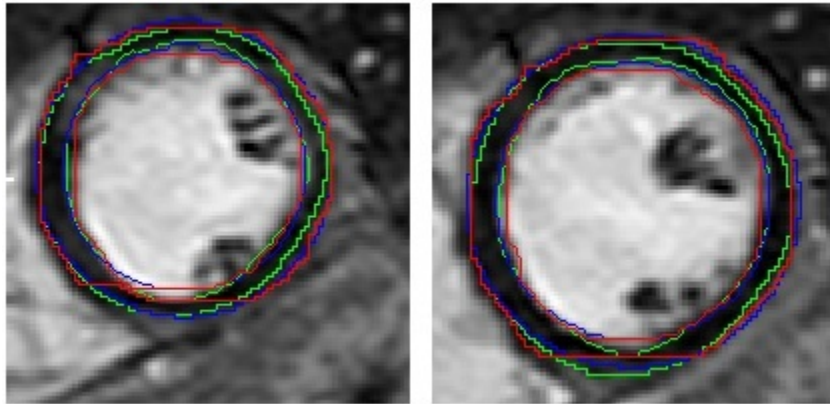
$$Dc(x, y, n) = a \times d(x, y) \pm b \times Gx(x, y) \pm b \times Gy(x, y)$$

Vektene er i første eksperiment satt til  $a = 0.5$  og  $b = 0.5$ . Grunnen til dette valget, er at avstanden ikke skal sette like stort preg som den gjorde i forrige eksperimentet. Slik at det kuttet som gir laveste kostnad skal kunne korrespondere bedre med faktisk endokard og epikard i bildene.

$a = 0.5$ $b = 0.5$ $Sc_1$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7127	0.0085	—	0.8185	0.8233	0.0048
Sett 21	0.4884	0.5120	0.0236	—	0.7989	0.8069	0.0080
Sett 46	0.1891	0.2096	0.0205	—	0.7887	0.7119	-0.0768
Gjennomsnitt	0.4833	0.4993	0.0160	—	0.7865	0.7673	-0.0192
$a = 0.5$ $b = 0.5$ $Sc_2$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7070	0.0028	—	0.8185	0.8243	0.0058
Sett 21	0.4884	0.5007	0.0123	—	0.7989	0.8057	0.0068
Sett 46	0.1891	0.1989	0.0098	—	0.7887	0.7614	-0.0273
Gjennomsnitt	0.4833	0.4930	0.0097	—	0.7865	0.7813	-0.0052

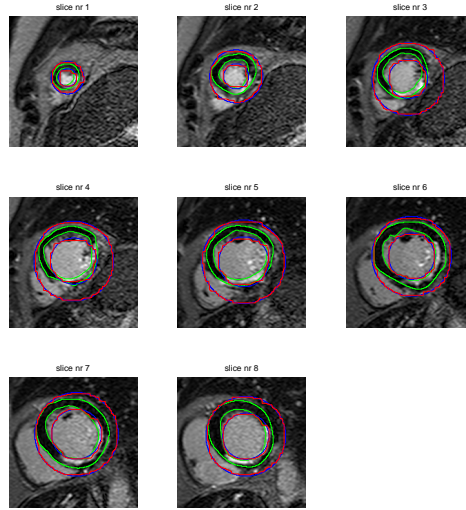
$a = 0.5$ $b = 0.5$ $Sc_3$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7046	0.0004	—	0.8185	0.8204	0.0019
Sett 21	0.4884	0.4922	0.0038	—	0.7989	0.8013	0.0024
Sett 46	0.1891	0.1910	0.0019	—	0.7887	0.7821	-0.0066
Gjennomsnitt	0.4833	0.4868	0.0035	—	0.7865	0.7867	-0.0002

I tabellene over er  $DI$  av graphcut segmenteringen med den nye kostnadsfunksjonen vist. Når disse verdiene sammenlignes med forrige eksperiment som var basert på avstand, er  $DI$  høyere for alle bildesett tilhørende  $bi$ , dette gjelder for alle  $Sc$ , men størst endring er med  $Sc_1$ . For  $ki$  er segmenteringen blitt dårligere, spesielt for bildesett 46. Grunnen til dette er vist i figur 24, i nedre del av konturene har graphcut segmenteringen funnet beste kutt av epikard i et område som berører den korrekte segmenteringen av endokard.



Figur 24: Et utvalg av resultat fra bildesett 46

Figur 25 viser de forskjellige konturene i bildesett 21 med  $Sc_1$ . Selv om dette hadde den største forbedringen, er det fremdeles store avvik fra funnet segmentering sammenlignet med de korrekte konturene. Grunnen til dette er de relativt store avviket med de initielle konturene.



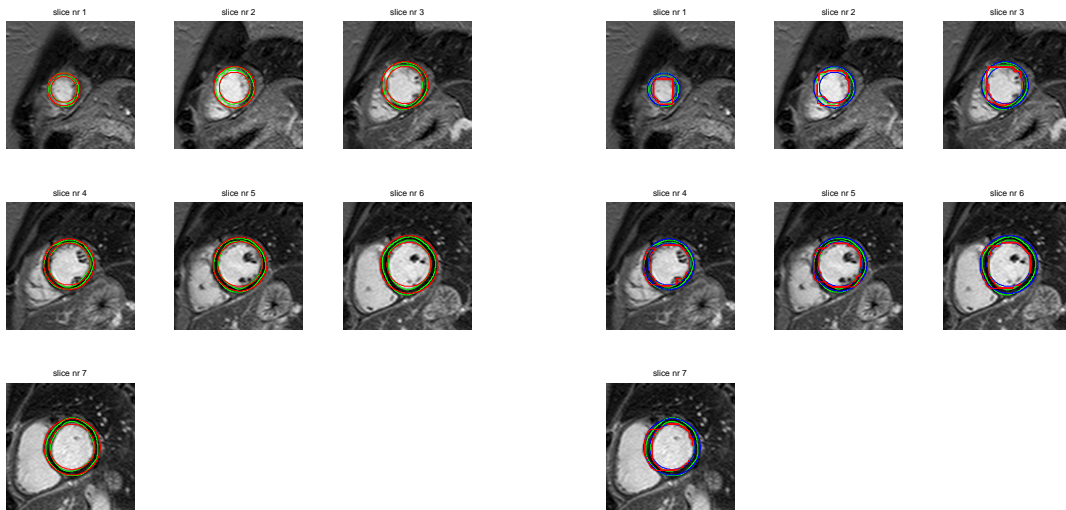
Figur 25: Resultat med kostnadsfunksjon basert på distanse og gradient for bildesett 21 med  $Sc_1$

På grunnlag av omfanget av korrigeringen som må til for noen av de initielle konturene i  $bi$ , endres vektene i kostnadsfunksjonen. Forholdet mellom vektene  $a$  og  $b$  endres for å sette større vekt på gradientene i bildene, fremfor avstanden fra de initielle konturene. Vektene endres til  $a = 0.1$  og  $b = 0.5$ . Resulterende  $DI$  av denne endringen er vist i tabellene under:

$a = 0.1$ $b = 0.5$ $Sc_1$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7365	0.0323	—	0.8185	0.7284	-0.0901
Sett 21	0.4884	0.5368	0.0484	—	0.7989	0.6970	-0.1019
Sett 46	0.1891	0.2772	0.0881	—	0.7887	0.3478	-0.4409
Gjennomsnitt	0.4833	0.5124	0.0291	—	0.7865	0.6104	-0.1761
$a = 0.1$ $b = 0.5$ $Sc_2$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7263	0.0221	—	0.8185	0.8191	0.0006
Sett 21	0.4884	0.5357	0.0473	—	0.7989	0.7905	-0.0084
Sett 46	0.1891	0.2345	0.0454	—	0.7887	0.0546	-0.2427
Gjennomsnitt	0.4833	0.5117	0.0284	—	0.7865	0.7109	-0.00756

$a = 0.1$ $b = 0.5$ $Sc_3$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.0784	0.0042	—	0.8185	0.8198	0.0013
Sett 21	0.4884	0.4990	0.0106	—	0.7989	0.8035	0.0046
Sett 46	0.1891	0.1974	0.0083	—	0.7887	0.7315	-0.0572
Gjennomsnitt	0.4833	0.4940	0.0107	—	0.7865	0.7735	-0.0130

Hvis resultatene med de nye vektene sammenlignes med de gamle, så forbedres  $DI$  etter graphcutsegmenteringen med de initielle konturene i  $bi$ . For segmenteringen funnet for de initielle konturene i  $ki$  er disse blitt endel dårligere. Dette spesielt for bildesett 46 med  $Sc_1$  og  $Sc_2$ , er  $DI$  blitt betraktelig redusert, grunnen til dette er tilsynelatende at bredden i de initielle konturene blir for små. Dette resulterer videre i at segmentet av endokard og epikard blir lagt i samme punkt. Som gjør at avstanden mellom segmentene blir null og konturen forsvinner helt i disse punktene. Resultatet av dette illustreres i figur 26 for bildesett 46.

(a)  $ki$  for bildesett 46

(b) Konturer etter segmentering

Figur 26: Illustrasjon av manglende konturer etter graphcut segmentering

Selv om  $DI$  for segmenteringen av konturene i  $ki$  er blitt endel dårligere, så er forbedringen funnet med dette settet av vektorer og  $Sc_1$  for konturene i  $bi$ . Valget faller da på å

fokusere på forbedring i segmenteringen av  $bi$ . Derfor vil videre eksperimentering ikke ta hensyn til  $DI$  for  $ki$ , dette vil kun bli betraktet hvis segmenteringen av  $bi$  ikke gir noe direkte svar på forbedringer.

### 5.3 Beste segmentering

Med grunnlag i eksperimentene som er utførte, og en kostnadsfunksjon basert på vekting mellom avstand og gradientene. Er den beste graphcutsegmenteringen som er funnet, en kombinasjon av vektene  $a = 0.1$  og  $b = 0.55$  og en moderat korrigert glatthetskostnad  $Sc_4$  som vist under.

$$Sc_4 = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}$$

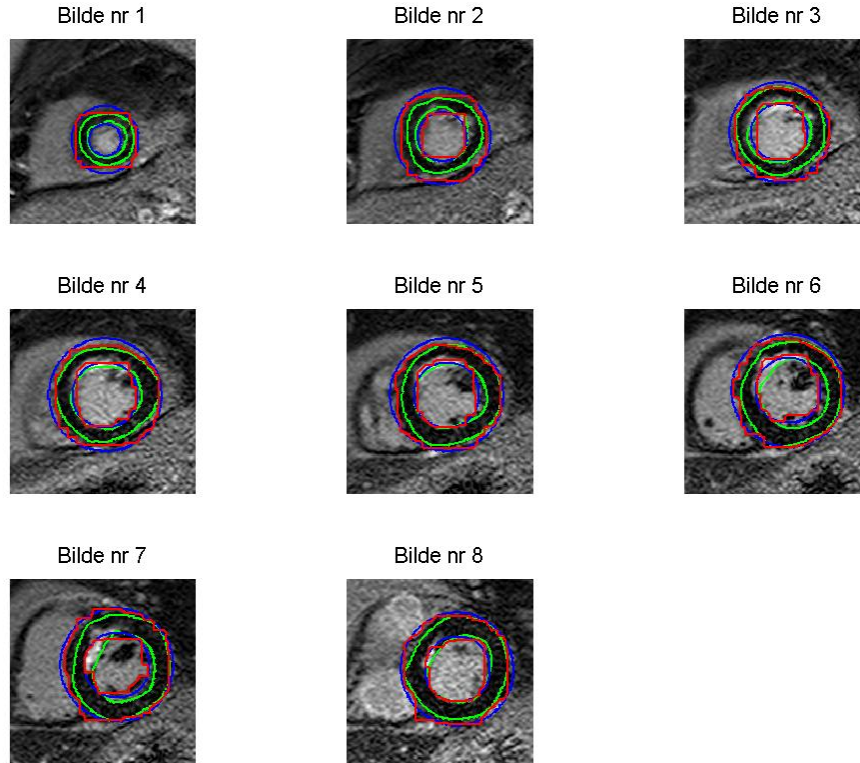
Når det her sies den beste segmenteringen funnet, så er det den beste segmenteringen funnet med graphcut for de initielle konturene  $bi$ . Grunnen til at konturene fra  $ki$  ikke er med i vurderingen er at fokuset er lagt mot bedring av konturene beregnet fra forbehandlingen. I vurderingen av beste segmentering er det også lagt mer vekt på bedringen for gjennomsnittet av alle bildette, fremfor bedringen av et av de tre enkeltsettene. I tabellen under er resultatene målt i  $DI$  fremstilt for den beste graphcut segmenteringen.

$a = 0.1$ $b = 0.55$ $Sc_4$	$DI_{init}$	$DI_{gc}$	Bedring	—	$DI_{ki}$	$DI_{kgc}$	Bedring
Sett 2	0.7042	0.7324	0.0282	—	0.8185	0.8195	0.0010
Sett 21	0.4884	0.5473	0.0589	—	0.7989	0.7644	-0.0345
Sett 46	0.1891	0.2596	0.0705	—	0.7887	0.4146	-0.3741
Gjennomsnitt	0.4833	0.5169	0.0336	—	0.7865	0.6528	-0.1337

Fra tabellen kommer det frem at bildsett 46 har den største forbedringen fra  $DI_{init} \rightarrow DI_{gc}$ . Denne forbedringen er på 0.0705, men i sluttresultatet så det er fremdeles endel å gå på før segmenteringen kan sies å være korrekt. Dette kan sees i figur 29

For bildesett 2 som har den høyeste sluttverdien for  $DI_{gc}$ , er graphcut segmenteringen nærme den korrekte segmentering. Det er fremdeles avvik som må kompenseres for, da

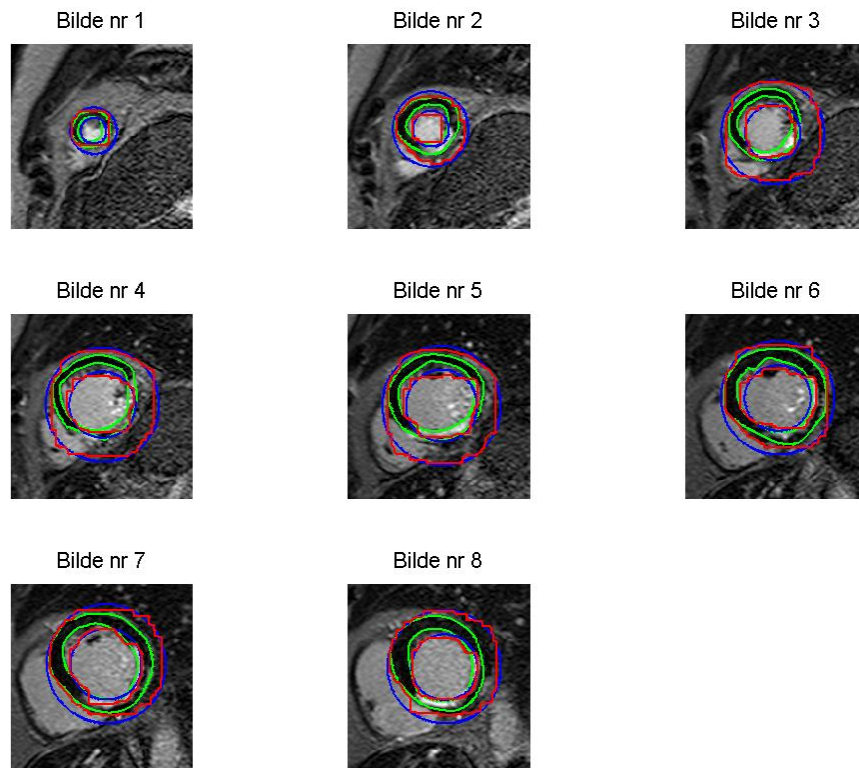
endringen fra de initielle konturene ikke er så mye endret, dette er vist i figur 27. Videre skal det sees nærmere på resultatene for bildesett 2, 21 og 46 hver for seg.



Figur 27: Beste graphcut segmentering funnet for bildesett 2

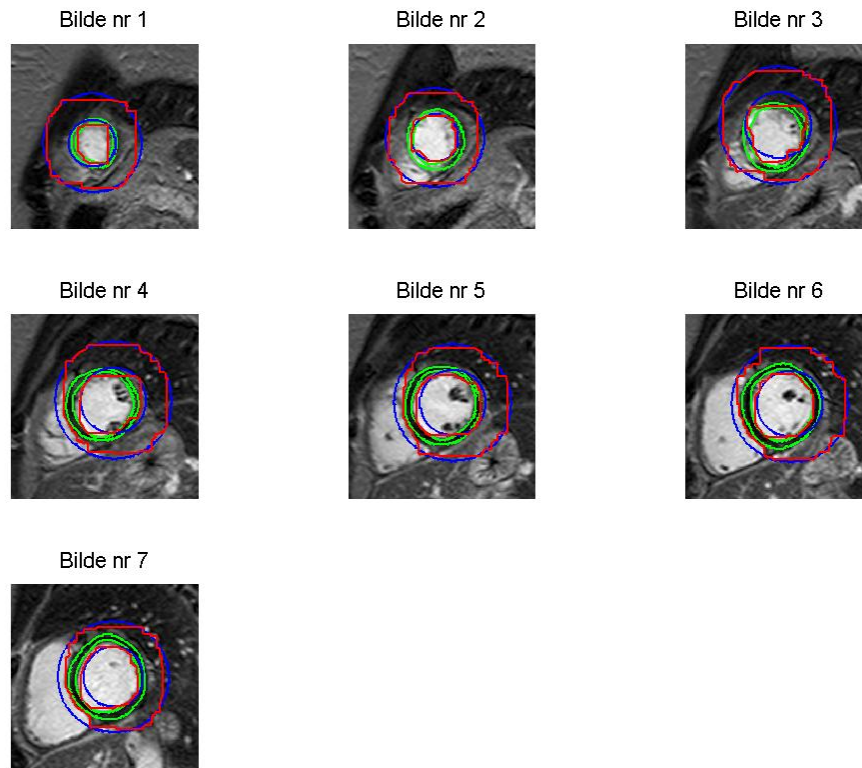
Resultatet av graphcut segmenteringen for bildesett 2 er vist i figur 27. Resultatet som blir funnet viser at det største avviket kommer i bilde nr 1. Her er konturene mellom endokard og epikard nesten de samme og ligger plassert hvor segmenteringen av epikard skal være. For resten av bildene i settet er det gjentakende at for mye av arret er tatt med i segmenteringen av endokard.





Figur 28: Beste graphcut segmentering funnet for bildesett 21

Resultatet av graphcut segmenteringen for bildesett 21 er vist i figur 28. Vi ser her at det er tilnærmet likt resultat for bildene som i bildesett 2. Men her er det noe større avvik grunnen dårligere initielle konturer.



Figur 29: Beste graphcut segmentering funnet for bildesett 46

Resultatet for bildesett 46 er vist i figur 29. For dette bildesettet er segmenteringen av endokard relativt bra i henhold til de korrekte konturene. Segmenteringen for epikard er generelt sett relativt dårlig for alle bildene. De plassene den initielle konturen av epikard er nært den korrekte konturen av epikard, har graphcut segmenteringen funnet et kutt som plasser godt mot epikard<sup>7</sup>. Men for resten av de initielle konturene som er relativt langt vekke fra den korrekte konturen er differansen stor.

I vedlegg B ligger resultatene fra graphcut segmenteringen, for alle bildesettene med de aktuelle vektene, glatthetsbetingelsen for både *bi* og *ki*.

<sup>7</sup>Dette er godt synlig i bilde nr 4 til 7, i nedre del på venstreside av bildene

## 6 Drøfting

Som det kommer frem i sluttresultatet, så er det rom for forbedring av sluttresultat. For å kunne oppnå dette er det flere muligheter, i dette kapitlet skal potensielle forbedringer i form av endringer i utført arbeid vurderes.

### 6.1 Forbehandling

De initielle konturene som er beregnet med forbehandlingen treffer ikke like godt for alle bildesettene. Her er det hovedsakelig de bildene av pasienter med store arr, gjerne av de som er diagnostisert med *ICD* som treffer relativt dårlig. Siden graphcut segmenteringen ikke kan kompensere for dette avviket, er det store muligheter at mer nøyaktige initielle konturer ville gitt bedre segmentering. Men for at dette skal kunne oppnås må en annen fremgangsmåte vurderes for å kunne finne bedre initielle konturer for pasientene med store arr.

### 6.2 Kostnadsfunksjoner

Fra teorien kommer det at kostnadsfunksjon normalt er regions og/eller gradientbaserte. Kostnadsfunksjonen som er benyttet i denne oppgaven er basert på en regionbestemmelse<sup>8</sup> kombinert med gradientene i bildene. Da utviklingen av en godt tilpasset kostnadsfunksjon er av ytterste viktighet for graphcut segmenteringen, er det stor sannsynlighet i at utviklingen av en bedre tilpasset kostnadsfunksjon kan gi bedre resultater. Potensielle punkt for endringer kan være i form av hvordan gradienten tolkes og benyttes mot eventuelt andre regionbestemmelser.

Endringer på tolkingen av gradienten kan eventuelt gjøres med å bruke en felles gradient fremfor gradientene i x- og y-retning hver for seg. Det er da viktig at retningene til gradientene ivaretas, siden fortegnet forsvinner og ikke kan benyttes slik som det blir

---

<sup>8</sup>Det blir benyttet avstand som regionbestemmelse

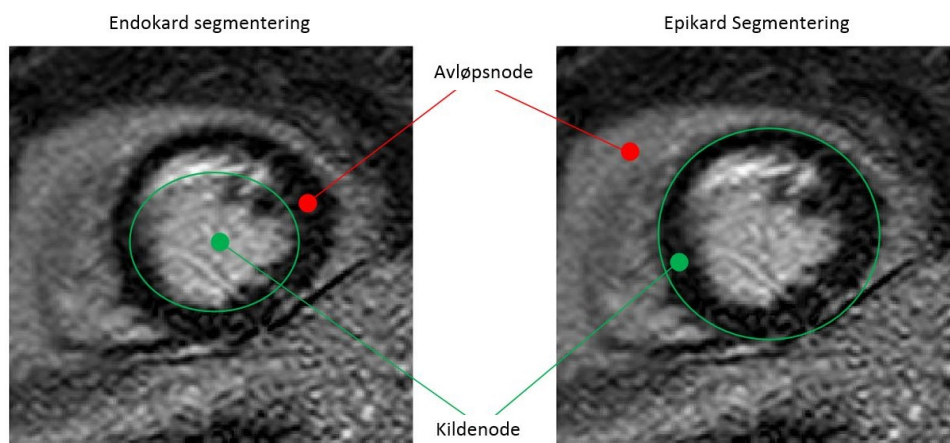
gjort nå. Gradienten kan beregnes på følgende måte:

$$|Gm| = \sqrt{Gx^2 + Gy^2}$$

Andre regionbestemmelser som eventuelt kan benyttes er en kantbegrensing. Slik at en forhindrer at segmentene blir dratt ut i kantene slik de ble i de innledende eksperimentene.

### 6.3 Plassering av kilde og avløp

Graphcut segmenteringen deler bildet inn i forskjellige segmenter kallet kilde og avløp. En mulighet for en mer robust løsning, er at det kan plasseres en eller flere noder som en med 100% sikkerhet kan sies at tilhører enten kilden eller avløpet. For segmenteringen av endokard må da kilde noden plasseres i hulrommet i hjertet, ikke på arret eller i eventuelle forstyrrelser i bildet. Avløpet må plasseres i den mørke konturen mellom endokard og epikard slik at den ligger i den mørke konturen i LGE-CMR bildene. Det samme må gjøres for epikard segmenteringen, men da må kilden plasseres i den mørke delen av konturen av hjertet og avløpet i en av de lysere delene av det myke vevet rundt. Figur 30 viser et eksempel på hvordan disse noder potensielt kan plasseres. Hvis en av disse noder plasseres feil vil segmenteringen få et dårlig utgangspunkt og segmenteringen vil bli feil. For at dette skal kunne benyttes må det også benyttes en annen implementasjon for graphcut segmentering som støttet dette.



Figur 30: Illustrasjon av hvordan eventuelle kilde og avløpsnoder kan plasseres i bildet

## 6.4 min kut/maks flyt mot normalisert kutt

Et siste forslag er at det kan vurderes å bruke en implementering av normalisert kutt algoritmen, fremfor min kutt/maks flyt algoritmen. Hvis dette skal gjøres må forbehandlingen og andre initielle betingelser endres, da normalisert kutt algoritmen benytter andre initielle tilstander.

## 7 Konklusjon

Fra resultatene kan det tas den konklusjonen om at segmenteringen som er funnet i denne oppgaven ikke er gode nokk til å kunne benyttes til automatisk segmentering av LGE-CMR bilder av pasienter som har hatt hjerteinfarkt. Grunnen til dette er at avviket fra funnet segmentering og korrekt segmentering er for stort.

Graphcut som segmenteringsmetode kan trolig benyttes til segmentering av LGE-CMR bilder, men da må forbehandlingen og/eller kostnadsfunksjonen forbedres. Det er da ønskelig at alle de initielle konturene har en  $DI_{init} \geq 0.7$ . Dette kombinert med en forbedret kostnadsfunksjon så vil det trolig kunne klare en  $DI_{gc} \geq 0.9$ , som skal være godt til bruk som en automatisk segmenteringsmetode.

---

## Referanser

- [1] K. Engan, V. Naranjo, T. Eftestoel, S. Oern, and L. Woie, “Automatic segmentation of the epicardium in late gadolinium enhanced cardiac mr images,” 2013.
- [2] K. Naess. Hjerteinfarkt. Landsforeningen for hjerte- og lungesyke. [Online]. Available: <http://www.lhl.no/hjertesykdom/hjerteinfarkt/>
- [3] K. Engan, V. Naranjo, T. Eftestoel, S. Oern, and L. Woie, “Segmentation of lg enhanced cardiac mri,” *Proceedings of the International Conference on Bioimaging*, 2015.
- [4] U. Qidwai and C. Chen, *Digital Image Processing*. CRC Press, 2009, no. ISBN: 978-1-4200-7950-0.
- [5] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd ed. PWS Publishing, 1999, no. ISBN: 0-534-95393-X.
- [6] M. Petrou and C. Petrou, *Image Processing - The Fundamentals*, 1st ed. WILEY, 2010, no. ISBN: 978-0-470-74586-1.
- [7] Y. Boykov and O. Veksler, “Graph cut in vision and graph: Theories and applications,” *Handbook of Mathematical Models in Computer Vision*, 2006.
- [8] O. Lezoray and L. Grady, *Image Processing and Analysis with Graphs, Theory and Practice*. CRC Press, 2012, no. ISBN: 978-1-4398-5507-2.
- [9] K. Engan, V. Naranjo, T. Eftestoel, L. Woie, A. Schuchter, and S. Oern, “Automatic detection of heart center in late gadolinium enhanced mri,” *Mediterranean Conference on Medical and Biological Engineering and Computing*, vol. XIII, pp. 165–168, 2013.
- [10] X. Alba, R. M. F. i Ventura, K. Lekadir, C. Tobon-Gomez, C. Hoogendoorn, and A. F. Frangi, “Automatic cardiac lv segmentation in mri using modified graph cuts with smoothness and interslice constraints,” *NA*, 2013.

- [11] Y. Boykov, O. Veksler, and R. Zabih, *Efficient Approximate Energy Minimization via Graph Cuts*, November 2001, iEEE transaction on PAMI, vol. 20, no. 12, p. 1222-1239.
- [12] V. Kolmogorov and R. Zabih, *What Energy Functions can be Minimized via Graph Cuts?*, February 2004, iEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 26, no. 2, pp. 147-159.
- [13] Y. Boykov and V. Kolmogorov., *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision.*, September 2004, in IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 26, no. 9, pp. 1124-1137.
- [14] S. Bagon, *Matlab Wrapper for Graph Cut.*, [www.wisdom.weizmann.ac.il/~bagon](http://www.wisdom.weizmann.ac.il/~bagon), Desember 2006.

# Appendices

## A Kildekode

- Hovedskript.m - Starter forbehandling og graphcut segmentering. Denne kaller alle funksjoner videre som kreves.

## B Resultater

### B.1 Bilder

Resultatbilder av beste segmentering kan fåes av Kjersti Engang ved UiS etter behov.

### B.2 Dice Index - eksperiment