



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Kybernetikk/Signalbehandling	Vårsemesteret, 2011 Åpen / Konfidensiell
Forfatter: Magnus Kopperud (signatur forfatter)
Fagansvarlig: Kjersti Engan Veileder(e): Kjersti Engan, Trygve Eftestøl	
Tittel på masteroppgaven: Automatisk segmentering av MRI-avbildninger av hjertet hos pasienter som har hatt hjerteinfarkt. Engelsk tittel: Automatic segmentation of MR images of the heart, who's patients had a myocardial infarction.	
Studiepoeng: 30	
Emneord: Automatisk segmentering Watershed Sirkulær tilnærming	Sidetall: 48 + vedlegg/annet: 0 Stavanger, dato/år

Sammendrag

I denne oppgaven presenteres en helautomatisk segmenteringsmetode til bruk på MRI-avbildninger av hjertet. Metoden som er utviklet benytter seg av morfologisk watershed til å definere segmenter for frisk muskulatur, samt arrvev hos pasienter som har hatt hjerteinfarkt. Den store variasjonen i gråtoneverdier MRI-bilder av denne typen har, gjør dette til et vanskelig problem. Systemets ytelse blir målt ved hjelp av et datasett bestående av ulike MRI-avbildninger. På disse avbildningene har en kardiolog annotert områdene som skal segmenteres.

Innhold

1	Introduksjon	1
2	Element og metode	2
2.1	Teori	3
2.1.1	Erosjon	3
2.1.2	Dilasjon	4
2.1.3	Terskling	6
2.1.4	Morfologisk Rekonstruksjon	7
2.1.5	Morfologisk gradient	9
2.1.6	Watershed	10
2.1.7	Analyse av form	13
2.2	Tidligere arbeid	16
2.2.1	Segment, programvare for segmentering av MRI- avbildninger	16
2.3	Metodeutvikling	17
2.3.1	Parametersetting	18
2.3.2	Markørdeteksjon	18
2.3.3	Morfologisk gradient	22
2.3.4	Watershed	23
2.3.5	Sirkulær tilnærming	24
2.3.6	Treningsdata og testdata	28
2.3.7	Resultatberegning	28
3	Test av algoritmen	32
3.0.8	Resultat Watershed	32
3.0.9	Resultat Sirkulær Tilnærming	35
4	Konklusjon	41
5	Videre arbeid	41
A	Appendix	44
A.1	Alternativ løsning på problemet	44
A.2	Funksjon for generering av enkeltresultater	48

1 Introduksjon

Hjerteinfarkt er et stadig voksende problem i dagens samfunn. Et hjerteinfarkt forårsakes ved at blodstrømmen i koronararteriene som er blodårene som tilfører hjertemuskulaturen oksygen og næring, blir blokkert [7]. En slik blokkering kan skyldes åreforkalkning, eller fettavleiringer forårsaket av høyt kolesterol. Det som da skjer er at deler av hjertemuskulaturen dør. Alvorlighetsgraden på infarkt er bestemt av hvor stor del av hjertemuskulaturen som dør. Heldigvis har behandlingsmetodene blitt gode, og de aller fleste som får hjerteinfarkt overlever. Det døde muskelvevet blir erstattet med bindevev, men dette vevet har ikke evnen til aktivt å trekke seg sammen. Resultatet av dette er derfor at hjertet blir svekket. Hvor mye det blir svekket avhenger av skadeomfanget på hjertemuskulaturen.

De pasientene med størst hjerteskaader har økt risiko for å få hjertestans på et senere punkt i livet [8]. De vil trenge en annen type oppfølging enn de med mindre skade. Det kan derfor være nyttig å dele pasientene som har hatt hjerteinfarkt inn i risikogrupper etter skadeomfang.

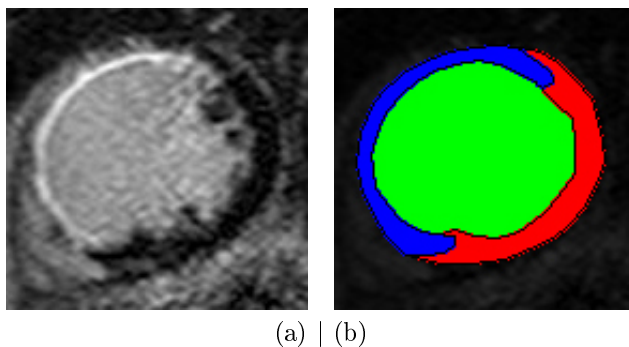
Dette foregår i dag ved at kardiologene bruker Magnetic Resonance (MR) avbildninger av hjertet, og markerer den skadde hjertemuskulaturen. Dette er både tid og ressurskrevende arbeid. Det skal i denne oppgaven bli undersøkt om det er mulig å gjøre denne markeringen automatisk ved hjelp av morfologisk watershed.

Denne rapporten er delt inn i tre hoveddeler. En teoridel hvor den nødvendige teorien som ligger til grunn for løsningen gjennomgås. En del som beskriver algoritmen som er blitt utviklet til å løse problemstillingen. Og en test av denne algoritmen.

2 Element og metode

Dette kapitlet inneholder teorien som ligger til grunn for den endelige løsningen på problemstillingen. Kapitlet inneholder også en beskrivelse av algoritmen som er blitt utviklet.

En MR-avbildning av hjertet kan se ut som følger:



Figur 1: MR-avbildning, (a) viser et snitt igjennom hjertet, (b) markerer områdene som skal segmenteres.

Figur 1(a) viser et snitt igjennom hjertemuskulaturen etter en MR-avbildning, figur 1(b) viser de områdene vi er interessert i. Det lyseste partiet i (a), som er markert blått i (b) er arrvevet, og det mørke partiet i (a) som er markert rødt i (b) er frisk muskulatur. Det er disse to områdene som vi ønsker å skille fra hverandre og det omliggende vevet.

Universitetet i Stavanger (UiS), Institutt for data- og elektroteknikk (IDE), har et pågående samarbeid med Stavanger Universitets Sykehus (SUS) som omhandler analyse av slike MR bilder av pasienter som har hatt infarkt. UiS har derfor tilgjengelig MR bilder fra et sett med pasienter som har hatt infarkt der en ekspert kardiolog manuelt har markert de interessante segmentene. I denne oppgaven anser vi kardiologens markeringer som fasit.

Problemstillingen vil bli forsøkt løst ved hjelp av morfologisk bildebehandling. Dette kapitlet vil introdusere de viktigste morfologiske operasjonene som blir benyttet i arbeidet med segmentering av MRI-avbildningene.

2.1 Teori

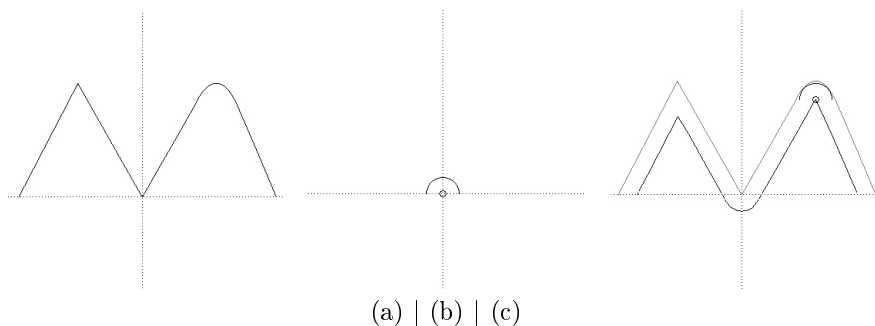
Morfologiske operasjoner brukes til å trekke ut vitale komponenter av et bilde [6]. I dette kapitlet skal de viktigste morfologiske verktøyene som blir benyttet til å løse problemstillingen bli introdusert.

2.1.1 Erosjon

Gråtoneerosjon brukes til å undertrykke lyse gråtoneelementer. Dersom et gråtonebilde har mange mindre topper som skaper støy, kan man ved hjelp av erosjon gjøre disse punktene mindre.

Erosjonen av et signal f om et strukturelement g er definert som [1]:

$$(f \ominus g)(x) = \max\{y : g_x + y \leq f\} \quad (1)$$

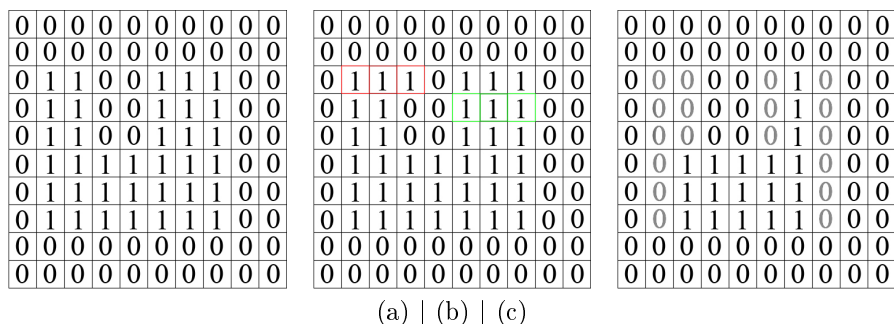


Figur 2: Erosjon av et signal. (a) er signalet, (b) er strukturelementet, (c) viser resultatet.

Figur 2 illustrerer likning 1. Figur 2(a) er signalet f , og figur 2(b) er strukturelementet g . Måten gråtoneerosjon fungerer i et gitt punkt x er som følger: Strukturen g forskyves punktvis til dens origo ligger i punktet x , etter dette forskyves g oppover så langt det er mulig uten at strukturen overstiger verdien i signalet f . Den nye strukturen $(f \ominus g)(x)$ blir liggende på innsiden av det originale signalet f , som vist i figur 2(c). Det er tydelig at strukturelementet passer bedre inn i den avrundede toppen av signalet, noe som fører til at $(f \ominus g)(x)$ får ulike verdier på de to toppene.

Binær erosjon

Nært beslektet med gråtoneerosjon, finnes også en metode for erosjon av binære bilder [4]. Denne metoden benytter et strukturelement til å redusere et binært objekt.



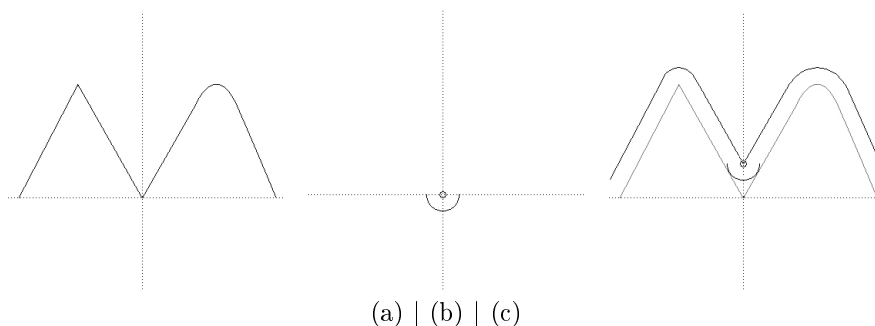
Figur 3: Viser binær erosjon, (a) er objektet før erosjon, (b) illustrerer strukturelementets tilpasning, (c) er resultatet

Figur 3 viser et eksempel på binær erosjon. I figur 3(a) er det et binært objekt. (b) viser hvordan binær erosjon fungerer. I dette tilfellet er strukturelementet en vektor bestående av tre enere, $[1 \ 1 \ 1]$. Strukturelementets origo er i midten. Når hele strukturelementet passer innenfor objektet, settes en ener i strukturelementets origo i en ny bildematrix. Et punkt hvor dette er tilfellet i eksempelet over, er det punktet som er markert med grønn ramme i (b). De punktene hvor strukturelementet ikke passer inn, settes til null i den nye bildematrixen. Et punkt hvor dette er tilfellet, er markert rødt i (b). Resultatet av erosjonen kan ses i figur 3(c). Hele elementet er blitt mindre, i tillegg til at den smaleste “antenna” er blitt borte. De grå tallene i (c) viser piksler som blitt satt til svart.

2.1.2 Dilasjon

Dilasjon kan anses som det motsatte av erosjon ref.2.1.1. I stedet for å se hvor høyt strukturen kan skyves opp, og fremdeles være mindre enn signalet. Vil strukturen nå roteres 180° , og målet blir å se hvor lite strukturen kan skyves opp og fremdeles ligge over signalet $[1]$. Dilasjon er definert som:

$$(f \oplus g)(x) = \min\{y : -\check{g}_x + y \geq f\}$$

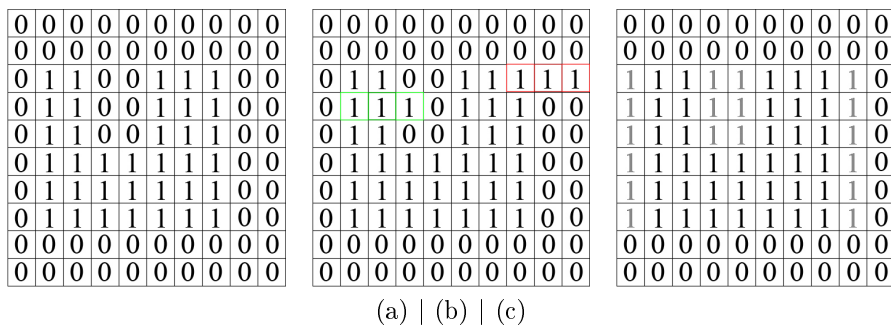


Figur 4: Dilasjon av et signal. (a) er signalet, (b) er strukturelementet, (c) viser resultatet.

På samme måte som i kapittel 2.1.1 er figur 4(a) signalet f , (b) er det roterte strukturelementet \check{g} . (c) illustrerer hvordan den diliderte funksjonen $(f \oplus g)(x)$ blir liggende over signalet f . Resultatet blir i dette tilfellet et signal med høyere avrundede topper.

Binær dilasjon

Gråtonedilasjon har også en nært beslektet metode for bruk på binære bilder [4]. Dilasjon på et binært bilde vil medføre en utvidelse av bildeelementet.



Figur 5: Viser binær dilasjon, (a) er objektet før dilasjon, (b) illustrerer strukturelementets tilpasning, (c) er resultatet

Figur 5 viser et eksempel på binær dilasjon. (a) viser bildeelementet, og (b) illustrerer hvordan dilasjon av et binært bilde fungerer. Strukturelementet er en vektor bestående av tre enere, $[1\ 1\ 1]$. Når strukturelementets origo treffer innenfor bildeobjektet, vil det settes enere tilsvarende hele strukturelementet i en ny bildematrix. Et eksempel hvor dette er tilfellet er vist i figur 5(b) markert med grønn ramme. Når strukturelementets origo ikke treffer, skjer det heller ingen ting. Et eksempel på dette er illustrert i (b) med en rød ramme.

Resultatet i dette tilfellet kan ses i (c), mellomrommet mellom de to “antennene” er nå blitt fylt igjen. I tillegg har hele objektet fått en ekstra kolonne på hver side. De grå tallene i (c) viser piksler som blitt satt til hvit.

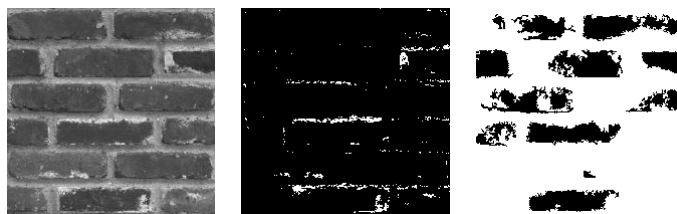
2.1.3 Terskling

Terskling er en metode som gjør et gråtonebilde om til et binært bilde. Og er definert som [4]:

For en bildematrise $f(x, y)$

$$g(x, y) = \begin{cases} 1 & \text{dersom } f(x, y) > T \\ 0 & \text{dersom } f(x, y) \leq T \end{cases} \quad (2)$$

En ny bildematrise $g(x, y)$, av like dimensjoner som $f(x, y)$ blir opprettet. Funksjonen går over alle pikslene i $f(x, y)$, dersom pikselverdien er over en bestemt gråtoneverdi T , vil denne pikselen settes til hvit i den nye bildematriksen $g(x, y)$. Dersom pikselverdien er mindre eller lik den bestemte gråtoneverdien settes pikselen til svart. Gråtoneverdien T vil ofte være vanskelig å bestemme. Settes den for høyt kan man risikere å miste viktig informasjon, mens dersom den settes for lavt kan det komme med bildeelementer som forstyrrer bildet.



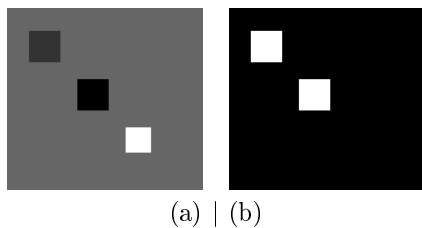
(a) |(b) |(c)

Figur 6: Terskling av (a) med for høy terskelverdi (b), og for lav teskelverdi (c).

Figur 6 illustrerer resultatet som oppnås med for høy eller for lav terskelverdi. Dersom målet her var å skille murklossene og sementen, er det tydelig at terskelen er satt for høyt i (b), og for lavt i (c).

Regional minima

En annen form for terskling som også vil bli benyttet er regional minima. Kort forklart segmenterer denne funksjonen ut sammenhengende mørke partier, som er omsluttet av lysere pikselverdier.



Figur 7: Illustrerer regional minimum, (a) er bildet, og (b) er resultatet etter funksjonen.

Figur 7 illustrerer regional minimum. I (a) er det tre bokser med ulike gråtone, etter kommandoen `imregionalmin()` er det kun de to boksene med lavere gråtoneverdi enn bakgrunnen som står igjen.

2.1.4 Morfologisk Rekonstruksjon

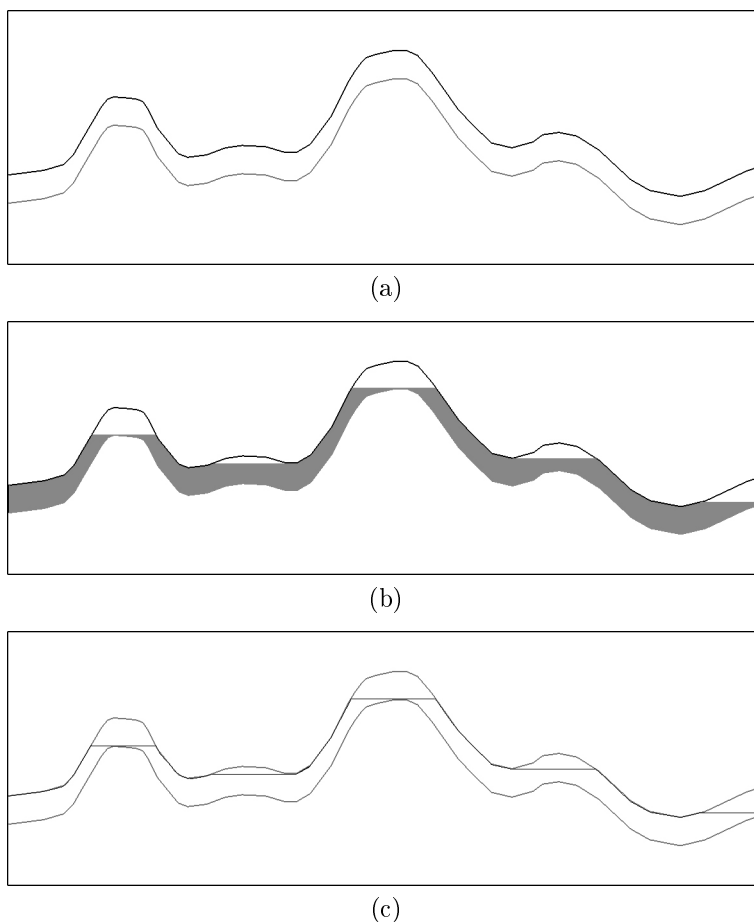
Morfologisk rekonstruksjon vil bli benyttet på gråtonebilder for å flate ut “toppene” og gi et bedre utgangspunkt for terskling. Morfologisk rekonstruksjon er en transformasjon som brukes for å oppnå en flatere bildestruktur. Transformasjonen benytter seg av to bilder og et strukturelement, et originalbilde (fra nå av markør), og dens tilhørende transformasjon (fra nå av kalt maske). Strukturelementet som blir brukt beskriver sammenhengen mellom markøren og masken. Stegene for rekonstruksjon er beskrevet i [1]. Der G er en maske og F er en markør vil rekonstruksjonen av G fra F fra nå av $R_G(F)$, være definert av følgende:

1. Initialiserer h_1 til å være markørbildet, F .
2. Skaper et strukturelement, B .
3. Repeterer følgende interaksjon:

$$h_{k+1} = (h_k \oplus B) \cap G$$

frem til $h_{k+1} = h_k$.

Markøren F må være et element av masken G for at denne prosessen skal være mulig.



Figur 8: Endimensjonal illustrasjon av morfologisk rekonstruksjon av et gråtonebilde. Svart kurve i (a) er markøren, grå kurve i (a) illustrerer masken, (c) viser resultatet.

Figur 8 viser hvordan rekonstruksjon av et gråtonebilde fungerer i en dimensjon. Den svarte kurven i (a) er markøren, og den grå kurven er masken. I dette enkle eksempelet er masken generert ved å trekke en konstant fra markøren. Måten morfologisk rekonstruksjon virker på er at masken blir utvidet helt til den treffer markøren (figur 8(b)). Den flatede strukturen oppstår i maskens toppunkt, grunnen er at masken kun utvides i y -retning i de toppunktene som er lavere en markørens bunnpunkt. Den endelige rekonstruksjonen er illustrert i figur 8(c), de to mindre toppene er nå blitt borte, mens de to største har fått større flate.

2.1.5 Morfologisk gradient

Gradienter blir brukt i bildebehandling til å finne kanter, der en kant anses som en brå overgang mellom lyse og mørke pikselverdier. En mye brukt metode for deteksjon av kanter er sobel kantdeteksjon. Sobel kantdeteksjon benytter seg av et 3×3 naboskap til å fremskaffe gradienten. Matematisk er sobel kantdeteksjon definert som[1]:

$$\nabla f = [g_x^2 + g_y^2]^{\frac{1}{2}} \quad (3)$$

$$= \{[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)^2] + [(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)^2]^{\frac{1}{2}} \quad (4)$$

Der:

	z_1	z_2	z_3			
	z_4	z_5	z_6			
	z_7	z_8	z_9			

$$g_x = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \quad g_y = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \quad (5)$$

Ved å gi senterpikselen dobbelt verdi vil gradienten få en noe mykere overgang, dette er med på å redusere støy.

Måten denne formen for kantdeteksjon fungerer på i praksis er som følger:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	3	4	4	4	4
0	0	0	0	1	3	4	4	4	4
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1

Figur 9: Matrisene viser konvolusjonen mellom matrisen A og g_x

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	3	3	0	0	0	0
0	0	0	0	4	4	0	0	0	0
0	0	0	0	4	4	0	0	0	0
0	0	0	0	4	4	0	0	0	0
0	0	0	0	4	4	0	0	0	0
0	0	0	0	4	4	0	0	0	0

Figur 10: Matrisene viser konvolusjonen mellom matrisen A og g_y

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1.43	1.4	4	4	4	4
0	0	0	0	3.14	2.4	4	4	4	4
0	0	0	0	4	4	0	0	0	0
0	0	0	0	4	4	0	0	0	0
0	0	0	0	4	4	0	0	0	0
0	0	0	0	4	4	0	0	0	0

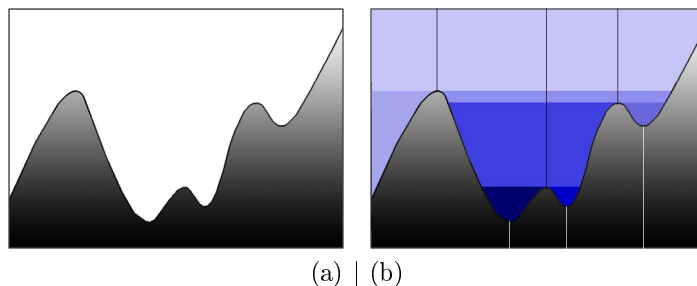
Figur 11: Matrisen $\nabla f = \sqrt{A(g_x)^2 + A(g_y)^2}$

Resultatet blir en tydelig linje som viser kanten til objektet i bildematrixen A . Hjørnet til strukturen blir også synlig med en jevnere overgang.

2.1.6 Watershed

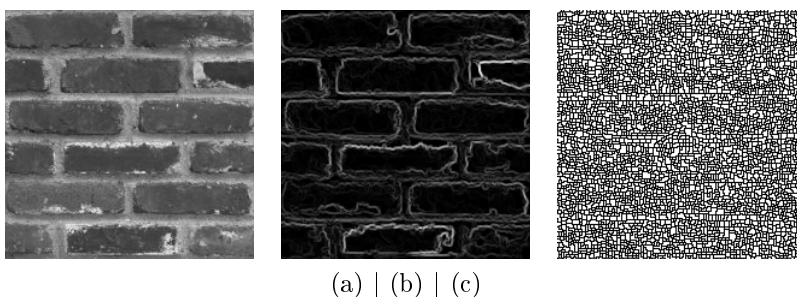
Watershed er en segmenteringsmetode for digitale bilder. Den enkleste måten å illustrere metoden på, er ved hjelp av en landstruktur. Dersom vann tilføres fra bunnen av en landstruktur, vil først de dypeste dalførene bli oversvømt [4].

På samme måte kan et gråtonebilde betraktes i en dimensjon, de mørkeste partiene av denne strukturen blir dalbunnene, mens de lyseste partiene utgjør toppene.



Figur 12: Endimensjonal illustrasjon av watershed, (a) er før, og (b) er etter “oversvømmelse”.

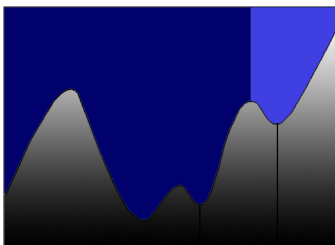
Etterhvert som strukturen blir oversvømt og vannet går fra et “basseng” over i et annet, blir det satt en watershed for å skille mellom “bassengene”. For hver nye topp blir det satt en ny skillelinje. Hver watershed vil nå representere et lokalt toppunkt i gråtonebildet. Noe som er viktig dersom denne metoden skal brukes er derfor å finne frem til en bildestruktur hvor områdene vi ønsker å skille ut er dalbunnene. En mye brukt metode for å oppnå dette er ved hjelp av morfologiske gradienter som beskrevet i kapittel 2.1.5.



Figur 13: Mur (a), gradient (b), oversegmentering (c).

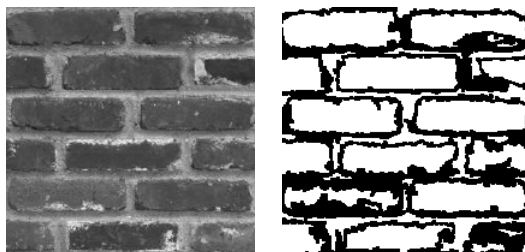
En av de største utfordringene med gradientbasert watershedsegmentering er at gradienten ikke er helt “ren”, i figur 13(b) er det tydelig at gradienten inneholder mye støy. Dette fører til oversegmentering og et ubrukelig watershedresultat (figur 13(c)).

En mye brukt metode for å begrense de gyldige områdene er markørkontrollert watershedsegmentering [1, 4]. Dersom tankegangen med landstrukturen videreføres, vil markører bety at man bare stikker hull i utvalgte bunnpunkter. Idet strukturen oversvømmes er det kun når vann fra to ulike bassenger møtes at en watershed blir satt.



Figur 14: Endimensjonal illustrasjon av markørkontrollert watershed.

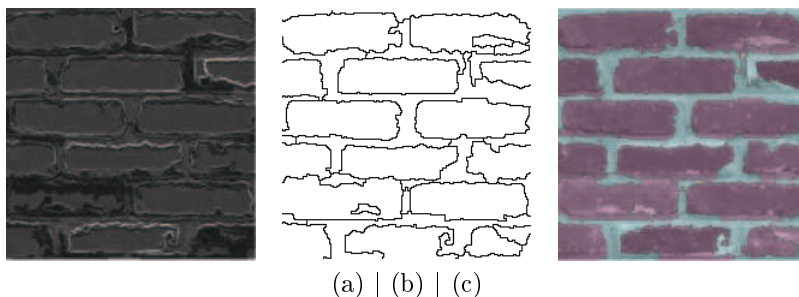
Det vil bli benyttet et sett av interne og eksterne markører for hvert av bildene. De interne markørene ligger på innsiden av objektene som skal segmenteres mens en ekstern markør vil bli plassert på utsiden. Ulike markører kan fremskaffes på ulike måter [4], lineær og ulineær filtrering er utbredt. I denne rapporten vil det bli lagt vekt på morfologisk bildebehandling for å bestemme markørene.



(a) | (b)

Figur 15: Mur (a), markørbilde (b).

I figur 15(b) vises det hvordan et markørbilde funnet ved hjelp av terskling (se 2.1.3) og erosjon (se 2.1.1) kan se ut. De indre markørene ligger på innsiden av hver enkelt murstein, og de eksterne markørene markerer områder hvor det ligger sement. Denne markøren kan nå benyttes til å modifisere gradienten fra figur 13, slik at watershedalgoritmen har større mulighet for et vellykket resultat.



Figur 16: Gradient (a), resultat watershed (b), resultat lagt over mur for illustrasjon (c).

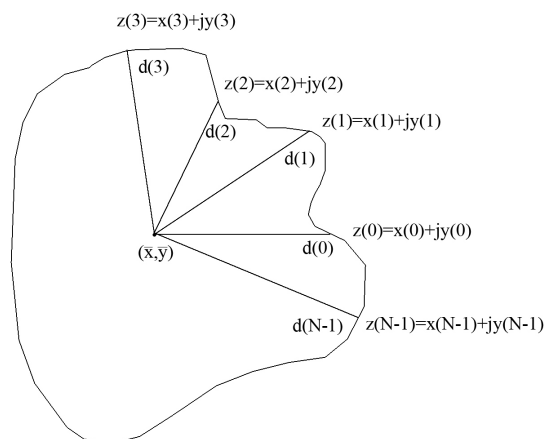
Figur 16 viser hvordan markørkontrollert watershed løser problemet med oversegmentering. Figur 16(a) viser markørbildet fra figur 15(b), lagt over gradienten fra figur 13(b). Gradienten blir nå modifisert ved å sette de overlappende områdene svart. Resultatet når watershedalgoritmen kjøres på den modifiserte gradienten blir som vist i figur 16(b). I figur 16(c) er resultatet lagt over originalbildet fra figur 15(a), og fargelagt for å gjøre resultatet mer synlig.

2.1.7 Analyse av form

Hjertet er et organ som er blitt forsket mye på. Det er derfor kjent at hjertet har glatte kanter, og er sirkulært eller elliptisk formet [10]. Resultatet av watershedalgoritmen er derimot ofte ujevne konturer. Det er derfor nyttig å foreta en analyse av formen som watershedalgoritmen har kommet frem til.

En naturlig måte å representere formen til et objekt på, er ved hjelp av (x,y) koordinater langs kanten til objektet. Dersom objektet har N punkter, kan konturen til objektet beskrives som:

$$\{x(n), y(n)\}, \quad n = 0, 1, 2, 3, 4, \dots, N - 1 \quad (6)$$



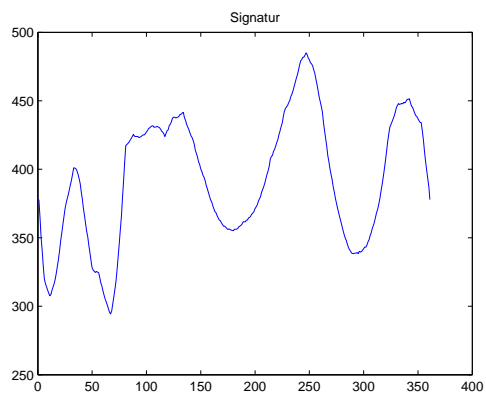
Figur 17: Figuren illustrerer hvordan analyse av formen til et objekt fungerer.

For å finne punktene langs kanten tas det utgangspunkt i objektets senter. Punktet for kanten finnes ved vinkel $\theta(0)$, $\theta(1)$, ..., $\theta(N-1)$. Resultatet vil være en rekke med imaginære verdier som nøyaktig beskriver objektets form.

Signatur

Formen representert ved hjelp av koordinater kan være vanskelig å forholde seg til. En signatur løser dette ved å beskrive formen som en rekke distanser fra et referansepunkt, til de aktuelle koordinatene. Denne måten å representere den aktuelle formen på kan ofte være enklere å forstå. Et naturlig referansepunkt å benytte er senterpunktet (\bar{x}, \bar{y}) fra figur 17. Formelen for å omgjøre de imaginære verdiene fra kapittel 2.1.7 til distanser fra objektets senter ser ut som følger:

$$d(n) = \sqrt{[x(n) - \bar{x}]^2 + [y(n) - \bar{y}]^2} \quad (7)$$



Figur 18: Figuren viser signaturen til objektet i figur 17.

Figur 18 viser objektet i figur 17 representert som distanser fra senteret (\bar{x}, \bar{y}) , og ut til kanten av objektet for alle n . Denne signaturen vil variere med formen til objektet. Som figur 17 viser, tilhører signaturen i figur 18 en ujevn struktur. Dette resulterer i en ujevn signatur med flere topper og bunner. Strukturen til en sirkulær form vil gi en mye jevnere signatur, der radiusen fra sentrum av objektet og ut til kanten varierer lite. Det er kjent at hjertet har en relativt sirkulær form, det er derfor ønskelig med en så jevn og flat signatur som mulig.

2.2 Tidligere arbeid

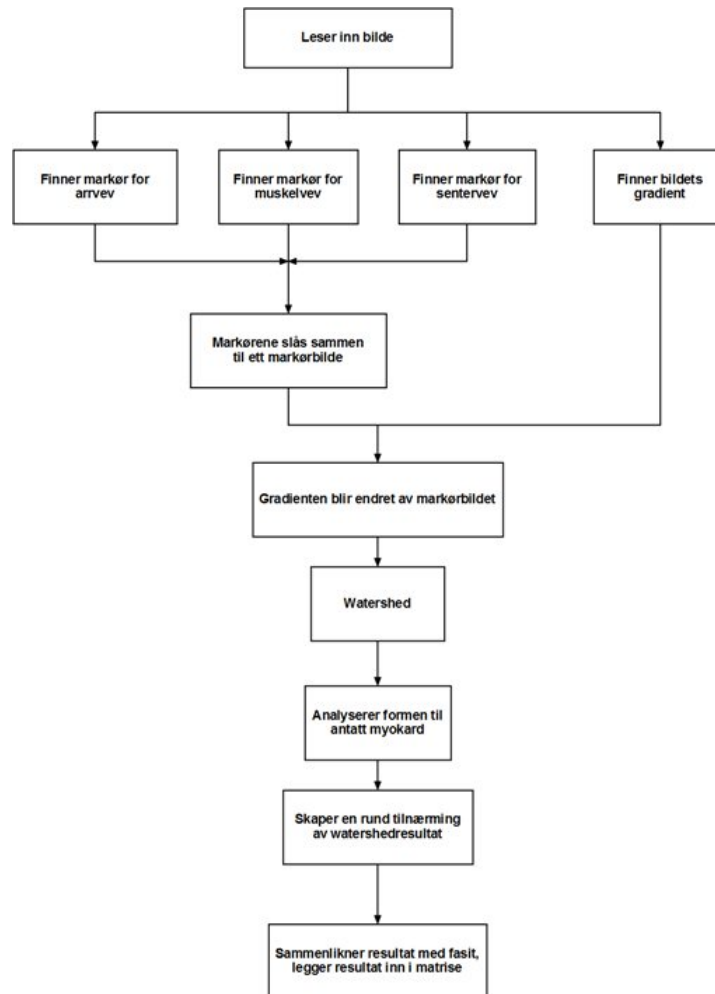
Innledningsvis ble det gjort litteratursøk for å finne frem til artikler som omhandler problematikken rundt segmentering av MRI-avbildninger av hjertet. I løpet av denne perioden ble det ikke funnet noen artikler som tok for seg bruk av watershed for å løse denne typen problemer. I en artikkel[5] ble det foreslått en forbedret watershedalgoritme som benytter seg av sannsynlighetsberegninger til å bestemme markørene, for på denne måten å forbedre resultatet. Metoden det her er snakk om ble ikke testet på MRI-avbildninger av hjertet. Men den ga gode resultater på segmentering av hjerne, og på segmentering av bruk i kneet.

2.2.1 Segment, programvare for segmentering av MRI-avbildninger

Segment er en kraftig programvare for analyse av medisinske bilder, hovedsakelig utviklet av Einar Heiberg[3, 2]. Denne programvaren kan løse veldig mange problemer rundt medisinsk bildebehandling. Et av områdene som denne programvaren omfatter er automatisk segmentering av hjertet. Spesielt med tanke på pasienter som har hatt hjerteinfarkt, ble det derfor innledningsvis kjørt en mindre test. Formålet med testen var å undersøke om denne programvaren har løst segmenteringsproblematikken rundt MRI-avbildninger av hjertet. Testen ble utført ved at tre ulike bildesett ble lastet inn i programvaren for segmentering. Programvaren greide ikke å plukke ut de riktige områdene automatisk. Det var behov for manuell markering av indre og ytre myokard før programvaren ga tilfredsstillende resultat. Det ble på bakgrunn av denne testen besluttet at arbeidet med å finne en helautomatisk algoritme for segmentering av MRI-avbildninger av hjertet skal fortsette.

2.3 Metodeutvikling

I dette kapitlet vil implementeringen av teorien fra kapittel 2.1, samt bruksområdene bli gjennomgått. En forenklet fremstilling av algoritmen som er utviklet kan ses her:



Figur 19: Enkelt blokkdiagram som beskriver algoritmen.

Figur 19 viser et blokkdiagram, der fremgangsmåten i den endelige algoritmen vises for et enkelt bilde.

Det første som skjer i algoritmen er deteksjon av markører (se kapittel 2.3.2), markørene skal som beskrevet i kapittel 2.1.6 begrense antallet områder i bildet som er av interesse. Markørene for arr, frisk muskulatur og hjertets senter blir funnet hver for seg, før de settes i sammen til et markørbilde (for eksempel,

se figur 23). Parallelt med deteksjonen av markører blir også den morfologiske gradienten utvunnet (se kapittel 2.1.5). Watershedalgoritmen blir kjørt på den morfologiske gradienten, men før dette skjer vil gradienten bli modifisert ved hjelp av markørbildet (se kapittel 2.3.3).

Etter at watershedalgoritmen beskrevet i kapittel 2.1.6 er kjørt på den modifiserte gradienten, foreligger det forhåpentligvis et segmentert resultat som inneholder de markerte områdene i figur 1 (b). Disse resultatene har ofte ujevne kanter. For å jevne ut kantene er det blitt utviklet en algoritme (se kapittel 2.3.5) som benytter seg av formanalyse, som beskrevet i kapittel 2.1.7. Nå foreligger det et sett med to ulike segmenter for hvert interesseområde. Disse settene vil til slutt bli sammenliknet med kardiologenes fasit, og lagt i hver sin resultatmatrise (se kapittel 2.3.7). Når flere sett med MRI-avbildninger er gjennomgått vil det foreligge et godt sammenlikningsgrunnlag mellom kun watershed, og den avrundede utgaven.

2.3.1 Parametersetting

For å bestemme de ulike parameterene i kapittelet som følger ble det kjørt en test med flere ulike parametersett. Det parametersettet som totalt sett ga det beste resultatet er det parametersettet som ble benyttet i det videre arbeidet.

2.3.2 Markørdeteksjon

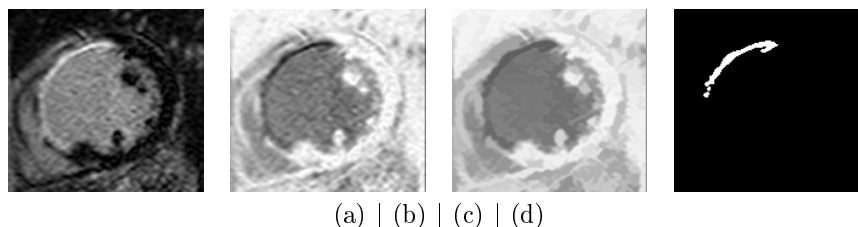
Markørene er veldig viktige for å få til gode segmenteringer. MRI-avbildninger av hjertet varierer mye i gråtoneverdi og skarphet. På grunn av dette har det vært veldig vanskelig å komme frem til en robust metode. Funksjonene som finner de tre ulike markørene er alle basert på morfologisk rekonstruksjon, beskrevet i 2.1.4.

I Arrmarkør

Arret lar seg identifisere som de lyseste pikslene vist i figur 20(a), metoden som ble utviklet for å plukke ut dette området følger disse stegene:

1. Inverterer bildet, funksjonen som benytter morfologisk rekonstruksjon skaffer best resultater på mørke pikselområder. Det er derfor nyttig å invertere bildet slik at området vi er interessert i blir det mørkeste.
2. Lavpassfiltrering for å flate ut støyelementene. Det benyttes her et medianfilter med vindusstørrelse 2×2 . Dette filteret er effektivt på "salt and pepper" støy, og brukes for å jevne ut overflaten i hjertets senter. Det er viktig at ikke betydningsfull informasjon blir borte, og det benyttes derfor et lite vindu.
3. Flater ut bildet ved hjelp av morfologisk rekonstruksjon. Først benyttes matlabfunksjonen `strel('disk', 3)`, denne skaper et rundt strukturelement med radius 3. Etter dette blir en rekke rekonstruksjoner med masker basert på erosjon og dilasjon gjennomført. Resultatet kan ses i figur 20 (b).

4. Figur 20(c) viser resultatet av matlabfunksjonen “`imhmin()`”, denne funksjonen får en fastsatt gråtone, og alle gråtoner som er mørkere en denne fastsatte størrelsen vil bli undertrykt [11]. For å gjøre funksjonen fleksibel ovenfor ulike bilder, blir grensen for gråtonen satt til 1% av den mørkeste tonen i bildet. Dette gjør at grensen som blir satt varierer fra bilde til bilde, i stedet for en fastsatt verdi som kunne medført problemer.
5. Segmentering ved hjelp av terskling, se kapittel 2.1.3. Her blir grensen for terskling satt som laveste gråtoneverdi +10%, grensen blir satt ganske lavt for å unngå støy. Dersom resultatet blir flere markører, sjekker algoritmen den gjennomsnittlige gråtoneverdien for hver av dem. Den markøren med høyest gjennomsnittlig gråtoneverdi i originalbildet, antas å være arrvevet. Dersom det ligger flere markører innenfor 85% av denne verdien antas også disse å være arrvev.



Figur 20: Viser veien fra originalbildet frem til arrmarkøren.

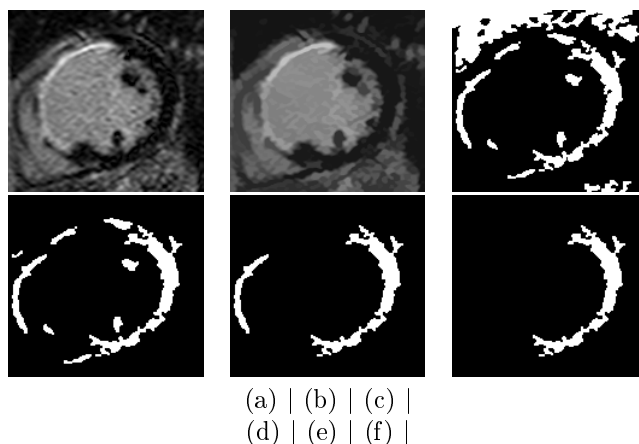
Figur 20 viser veien fra originalbildet til endelig arrsegment.

II Markørfremstilling frisk muskulatur

Den friske muskulaturen er den mørke halvsirkelen som kan ses i figur 21(a). En av de store utfordringene med deteksjonen av friskt muskelvev er at den kan ha samme gråtoneverdi som store deler av bakgrunnsvevet. Det er derfor i dette tilfellet nødvendig med flere steg for å oppnå tilstrekkelig godt resultat.

1. Første steg i fremstillingen av markøren til den friske muskulaturen er en sekvens av morfologisk rekonstruksjon, resultatet av dette steget kan ses i figur 21(b).
2. For å bestemme markøren brukes terskling av typen lokalt minimum (kapittel 2.1.3), denne sørger for å få med alle de mørke områdene i bildet. Som vist i figur 21(c) er det kommet med mange områder som ikke er av interesse. Det er derfor nødvendig med filtrering for å finne det elementet som faktisk tilsvarer den friske muskulaturen.
3. Første steg er en enkel funksjon som fjerner alle elementer som er i kontakt med bildets kant. Det antas at hjertet vil ligge innenfor bilderammene, og et hvert segment som går ut over rammen vil derfor være uten nyttig informasjon for segmenteringen.

4. Neste steg ser på størrelsen på elementene som er igjen, dersom noen av elementene er mindre enn en fastsatt størrelse anses de for å være støy. En risiko som må vurderes i denne funksjonen er det faktum at enkelte reelle muskelelementer kan være små, det er derfor en risiko at denne funksjonen fjerner viktig informasjon. Resultatet av denne filtreringen kan observeres i figur 21(e).
5. Det siste steget sammenlikner den foreløpige markøren med originalbildet (figur 21(a)). Her beregnes den gjennomsnittlige gråtoneverdien for hvert av elementene. Det elementet med mørkest gråtoneverdi antas å være frisk muskulatur. Dersom det er flere elementer som ligger innenfor 90% av den mørkeste gråtoneverdien, vil også disse elementene antas å være friskt muskelvev.



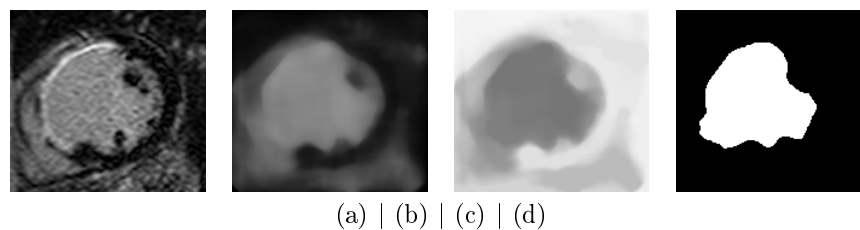
Figur 21: Viser fremgangen for fremstilling av friskt myokard.

Figuren viser stegene som finner frisk myokard.

III Markørfremstilling Senter

Markøren for senteret av hjertemuskulaturen er den enkleste å finne.

1. Ved hjelp av morfologisk rekonstruksjon, med et stort sirkulært element blir "sirkelen" som hjertet består av stående igjen.
2. Denne inverteres, og gråtonene jevnes ut.
3. Regionalt minimum benyttes for å plukke ut kjernen.

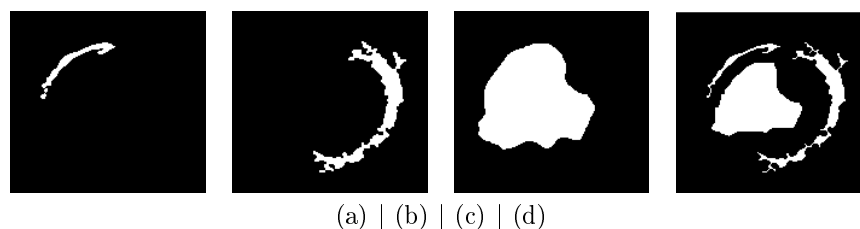


Figur 22: Viser fremgangen for deteksjon av sentermarkør.

Figuren illustrerer stegene som gjennomgås for å finne markøren for hjertets senter.

IV Markørbilde

Til slutt slås markørene for arr (figur 23(a)), friskt myokard (figur 23(b)), og sentervev (figur 23(c)) sammen til et markørbilde (figur 23(d)). Dette markørbildet vil bli brukt i neste steg for å endre på gradienten slik at den endelige segmenteringen blir så god som mulig. For at markørene skal fungere som tiltenkt er det veldig viktig at de ikke overlapper hverandre. Dersom for eksempel arr og senter gjør det, vil de antas å være samme segment. Før de slås sammen trekkes derfor arrmarkøren og markøren for friskt myokard, fra senteret. I tillegg til dette blir det benyttet erosjon for å unngå overlapping. En utfordring med å benytte erosjon til dette, er å bestemme størrelsen på strukturelementet som skal benyttes. Dersom strukturelementet er for stort kan det medføre at markørene blir delt opp i flere markører, eller i verste fall blir helt borte. Både arrmarkøren og markøren for frisk myokard varierer i størrelse, markøren for sentervev vil på sin side være stabilt størst. Det blir derfor benyttet små strukturelementer på arrmarkøren og markøren for friskt myokard, og et grovere strukturelement på markøren for sentervev.



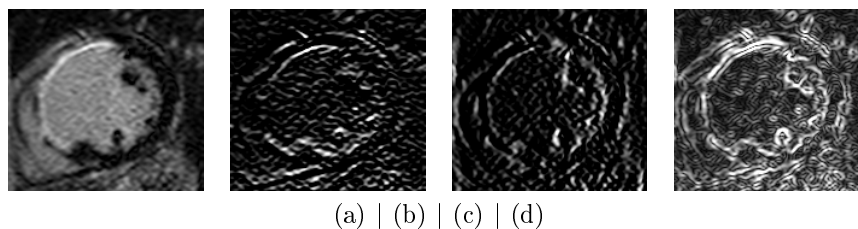
Figur 23: Markørbilde.

Figuren viser det endelige markørbildet som vil bli benyttet til å endre markøren.

2.3.3 Morfologisk gradient

Som beskrevet i kapittel 2.1.5 er den morfologiske gradienten et av de viktigste elementene for å oppnå et godt segmenteringsresultat. Det er gradienten som blir benyttet når watershedalgoritmen beskrevet i kapittel 2.1.6, bestemmer de ulike bunnpunktene som utgjør de ønskede muskulaturelementene.

Implementeringen av den morfologiske gradienten er hentet fra [9], og benytter seg av en filtreringsfunksjon som allerede eksisterer i matlabbiblioteket.

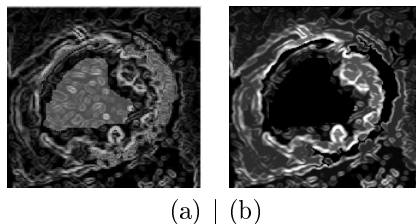


Figur 24: Illustrasjon av gangen fra originalbildet til ferdig gradient.

Figur 24 viser sobels gradient, (a) er utgangspunktet. (b) er resultatet av konvolusjon av bildematriksen (a) med g_x se likning 5 i kapittel 2.1.5, (c) er konvolusjonen med g_y se likning 5 i kapittel 2.1.5. (d) er den euklidske distansen mellom (b) og (c) se likning 3 i kapittel 2.1.5.

Ny gradient

Nå er både gradientbilde og markørbilde på plass. Disse skal kombineres til en ny gradient, som forhåpentligvis vil gi et godt segmenteringsresultat. Markørbildet legges over gradienten, og de felles områdene settes til svart i den nye gradientmatriksen. Til dette formålet benyttes matlabfunksjonen “imimposemin()”. Denne funksjonen benytter seg av et intensitetsbilde, i dette tilfellet gradienten i figur 24(d). Og et binært bilde, i dette tilfellet markøren i figur 23(d). Funksjonen oppretter en ny bildematrikse hvor kun områdene i det binære bildet blir regionale minimumer [11].

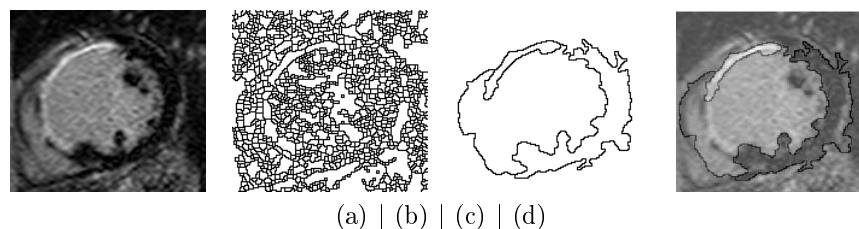


Figur 25: Fremstilling av modifisert gradient. (a) er gradienten fra figur 24 (d) pålagt markøren fra figur 23 (d), figur 25 (b) viser den endelige modifiserte gradienten.

Figur 25(a) viser gradient pålagt markør, figur 25(b) viser resultatet etter at funksjonen “imimposemin” er utført. Det er nå helt tydelig at interesseområdene fra markørbildet er de mørkeste delene av bildet, mens omliggende vev har fått en lysere gråtone.

2.3.4 Watershed

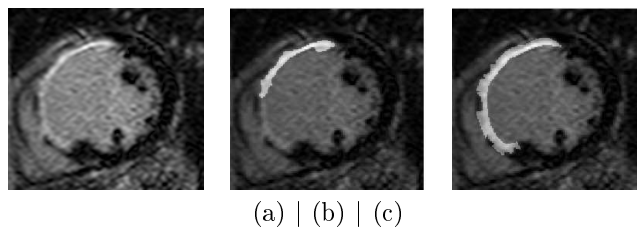
Watershedalgoritmen som er beskrevet i kapittel 2.1.6 er allerede implementert i matlab.



Figur 26: Fremstilling av watershedresultat, (a) er originalbildet, (b) illustrerer oversegenteringsproblematikk, mens (c) og (d) viser det endelige watershedresultatet.

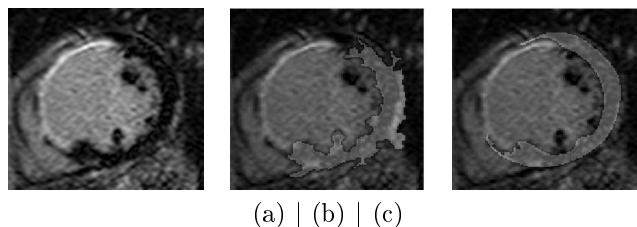
Figur 26 er en god illustrasjon på oversegenteringsproblematikken, og hvordan denne kan løses ved hjelp av markører. 26(b) er segmenteringsresultatet når watershedalgoritmen er påført gradienten i figur 24(d). Det er tydelig at det ikke er et resultat som kan benyttes til å løse problemstillingen. Resultatet som kan ses i figur 26(c) og (d) blir generert når watershedalgoritmen blir brukt på den redigerte gradienten fra figur 25(b). Det er helt tydelig at dette er et langt bedre resultat.

Slik resultatet foreligger nå er det en bildematrise med fire ulike segmenter som alle har verdi en. Det må nå avgjøres hvilket segment som er arrvev, hvilket som er friskt myokard, og hvilket som tilhører bakgrunnsvev. Dette blir gjort ved å sammenlikne hvert enkelt segment med sine tilhørende piksler i originalbildet. Segmentet med lavest snittverdi antas å være friskt myokard, segmentet med høyest snittverdi antas å være arr, og resten klassifiseres som bakgrunnsvev. Dersom det er flere segmenter som ligger innenfor 95% av hverandre i verdi, antas begge å tilhøre samme klasse. Dette muliggjør flere arrsegment, eller flere segmenter som inneholder friskt myokard. Resultatet etter denne sorteringen blir i dette tilfellet seende ut som følger.



Figur 27: Fremstilling av watershedsegmentene, (a) er originalbildet, (b) viser watershedalgoritmens antatte arrvev, og (c) er kardiologens annotering.

Som det går fram av figur 27, er arrsegmentet (b) mindre en kardiologens annotering (c). Dersom originalbildet (a) betraktes, kan man se at den lyseste delen av arret har kommet med, mens den svakere delen ikke er blitt detektert.

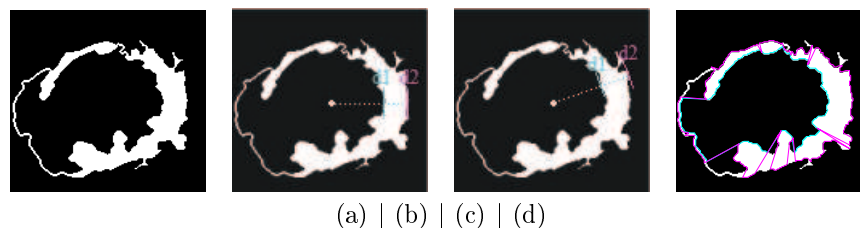


Figur 28: Fremstilling av watershedsegmentene, (a) er originalbildet, (b) viser watershedalgoritmens antatte friske myokard, og (c) er kardiologens annotering.

I figur 28 treffer watershedalgoritmen bedre. Estimaten (b) og kardiologens annotering (c) er ganske like. Men det er tydelig at estimaten er veldig ujevnt og ruglete i forhold til fasiten.

2.3.5 Sirkulær tilnærming

Som vist i figur 26(c) og figur 28(b) er det foreløpige resultatet veldig ujevnt og ruglete i kantene. For å forbedre dette benyttes kunnskapen om at hjertet er relativt rundt og har noenlunde glatte kanter. Ved å benytte signaturer som beskrevet i kapittel 2.1.7 kan man få et mål på hvor irregulære kantene er og et middel til å rette de ut med. Det er blitt utviklet en algoritme gjør dette. Algoritmen fungerer som følger:



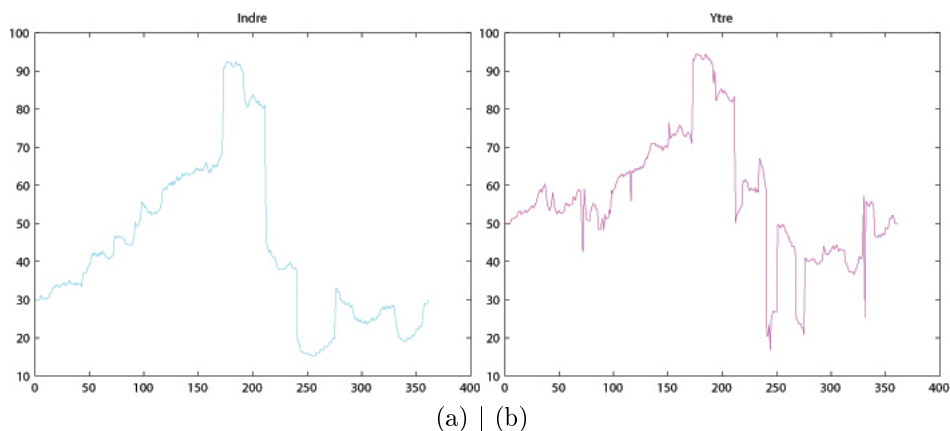
Figur 29: Figuren illustrerer sirkulær tilnærming. (a) viser antatt myokard, (b) viser fremgangen for analyse av indre og ytre myokard, (c) viser det samme som (b), men for en ny vinkel, og (d) viser det endelige resultatet.

Figuren illustrerer hvordan signaturen til henholdsvis indre og ytre myokard blir funnet.

Med utgangspunkt i watershedsegmentene som utgjør antatt myokard, finner algoritmen først senteret av hjertet.

1. For vinkel $\theta = [0, 1, 2, \dots, 360]$.
2. For første vinkel $\theta(0)$, økes radiusen helt til verdien på pikselet ikke lenger er svart (figur 29(b), d1), indre radius for vinkel $\theta(0)$ blir satt.
3. Fortsetter å øke radiusen for vinkel $\theta(0)$, helt til pikselverdien er svart igjen (figur 29(b), d2), ytre radius for vinkel $\theta(0)$ blir satt.
4. Steg 2 og 3 gjentas for alle θ .

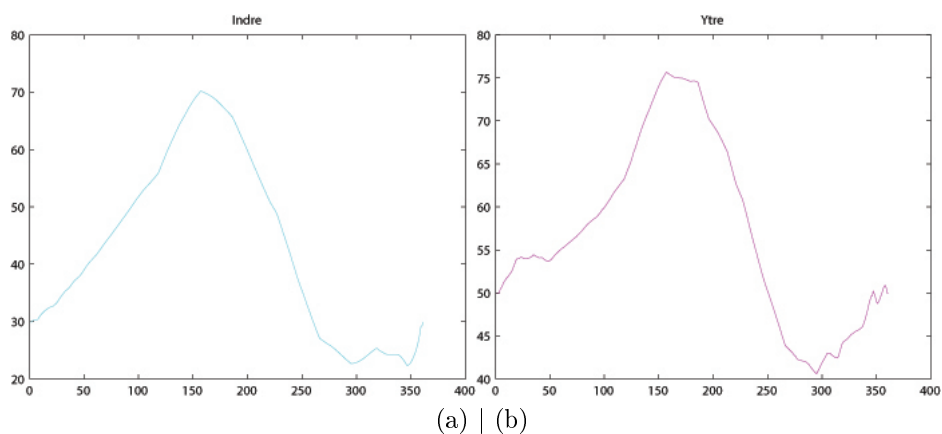
Når disse stegene er fullført blir resultatet som vist i figur 29 (d). Resultatet er to kurver som ganske nøyaktig beskriver formen til antatt myokard. For å kunne si noe om formen til kurvene, trekkes signaturen (les kapittel 2.1.7) ut.



Figur 30: Signaturen til indre (a) og ytre (b) myokard

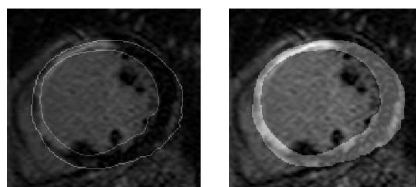
Figuren viser signaturen til indre myokard (a), og ytre myokard (b), dette er de samme kurvene som også kan ses i figur 29(d). Men her er absoluttverdiene for hver vinkel beregnet med utgangspunkt i hjertets senter (se likning 7 i kapittel 2.1.7). Disse signalene er en tilnærmet gjenskapning av den antatte myokardiale muskulaturen. De ujevne kantene fra figur 32(d), lar seg nå identifisere som brå overganger i de to kurvene. I realiteten er det kjent at avgrensningene for indre og ytre myokard er relativt glatte kanter, det er derfor naturlig å benytte et filter for å jevne ut de to signalene.

Det ble her valgt å benytte matlabs ferdig implementerte smoothingfilter. Dette filteret benytter gjennomsnittlig verdi over et bestemt antall datapunkter til å glatte ut et signal. Dette filteret tar godt vare på signalets form, samtidig som brå overganger blir borte.



Figur 31: Glattet fremstilling av sirkulær tilnærming

Figur 31 viser signalene fra figur 30 etter en smoothingoperasjon. Resultatet er en langt glattere signatur som følger formen til watershedsegmentene.



(a) | (b)

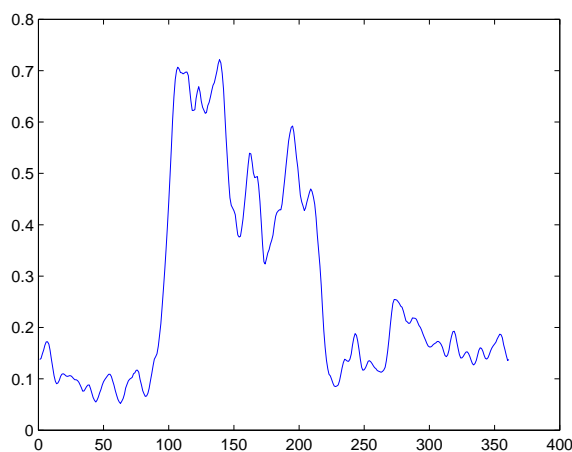
Figur 32: Resultat sirkulær tilnærming

Det er tydelig at dette er en langt bedre tilnærming til de reelle formene til hjertet, enn det som ble oppnådd i figur 26(d).

Neste steg i algoritmen er å fylle området mellom ytre og indre linje i figur 32(a) med hvitt. Resultatet blir en hvit “smultring” som vist i figur 32(b), denne “smultringen” inneholder både arrvev og friskt myokard. For å klassifisere hva som er hva, og dele ringen opp i segmenter, benyttes følgende fremgangsmetode.

- For vinkel $\theta = [0, 1, 2, \dots, 360]$, finner algoritmen den gjennomsnittlige gråtonen i originalbildet, mellom indre og ytre markørlinje.
- Resultatet av dette blir en graf som inneholder gjennomsnittlig gråtone for “smultringen” i 360 punkter.

Det vil nå være tydelig forskjell mellom delen av smultringen som er arr, og delen som er friskt myokard. Dersom disse gjennomsnittlige verdiene bli plottet, vil resultatet bli seende slik ut.

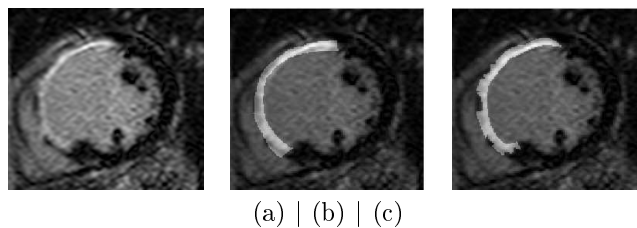


(a)

Figur 33: Graf som beskriver gjennomsnittlige gråtoneverdier

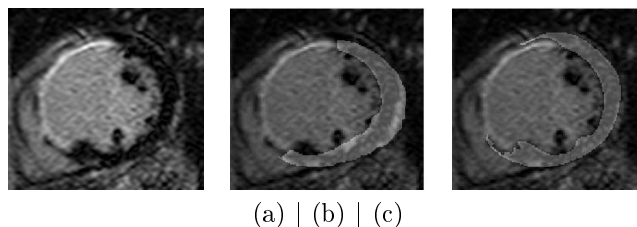
Arrsegmentet skiller seg her tydelig ut mellom punkt 100 til 210 (omtrent). Funksjonen kutter nå opp “smultringen” i disse overgangene, og resultatet er segmentene som beskriver arr og myokard. Punktene hvor kuttene skal gå er beregnet som overgangen hvor grafen er $6x$ høyere enn bunnivået, grunnen til at det gjøres på denne måten er at det ikke alltid er noe arrvev. Dersom det er tilfellet vil hele grafen være relativt lav, og punktene som etterspørres vil ikke eksistere. Noe som igjen fører til at “smultringen” ikke blir delt opp i flere segmenter.

Etter at dette steget er gjennomført blir resultatet av sirkulær tilnærming seende ut som følger.



Figur 34: Fremstilling av resultatene etter sirkulær tilnærming, (a) er originalbildet, (b) viser antatt arrvev, og (c) er kardiologens annotering.

Figur 35 viser antatt arrvev (b) etter sirkulær tilnærming. Dersom dette resultatet sammenliknes med watershedresultatet i figur 27, er det tydelig at likheten mellom antatt arrvev og fasit nå er større.



Figur 35: Fremstilling av resultatene etter sirkulær tilnærming, (a) er originalbildet, (b) viser antatt friskt myokard, og (c) er kardiologens annotering.

Dersom figur 35(b) sammenliknes med figur 28(b), er det her tydelig at resultatet etter sirkulær tilnærming er mye glattere. På grunn av dette ligger også det nye segmentet nærmere kardiologens annotering.

2.3.6 Treningsdata og testdata

Datasettet som var tilgjengelig ble delt opp i to kategorier. Et treningsdatasett som er blitt benyttet for å tilpasse de ulike parameterene som benyttes til watershedalgoritmen, og et testsett. For å sikre at testresultatene er så realistiske som mulig, har ikke testsettet blitt benyttet til noen form for trening av algoritmen. Måten denne inndelingen ble utført på var ganske enkel, de første seks settene ble brukt til trening, mens de resterende seksten settene ble benyttet som testsett.

2.3.7 Resultatberegning

Fastsettelse av resultatene til algoritmen som er beskrevet er problematisk. Grunnen er at fasiten som foreligger er en kardiologs subjektive tolkning av bildesettene. Dersom testbildene ble sendt til ti ulike kardiologer, ville det mest

sannsynlig bli returnert 10 ulike markeringer. Det som må finnes ut i den forbindelse er om resultatene som algoritmen kommer frem til er gode nok. Løsningen blir å sammenlikne algoritmens resultater direkte med kardiologens markeringer, for så å se nærmere på et utvalg av resultatene.

Fasiten er et binært bilde som inneholder henholdsvis arrvev, frisk muskulatur, og bakgrunnsvev. Segmentene som algoritmen kommer frem til er binære bilder av samme størrelse. Målingen som beskriver hvor bra algoritmen treffer beregnes på følgende måte:

1. De tre segmentene algoritmen kommer frem til multipliseres punkt for punkt med de tre fasitbildene. Dette fører til at pikslene hvor begge matriser har verdien en vil bli stående tilbake.
2. Pikslene i den nye bildematriksen telles opp og legges inn i en "forvirringsmatrise". Denne matrisen er en 3×3 -matrise som inneholder all informasjon om hvordan segmentene treffer.

	<i>Sa</i>	<i>Sm</i>	<i>Sb</i>
<i>Aa</i>	50	20	30
<i>Am</i>	25	55	20
<i>Ab</i>	25	25	50

Figur 36: Forvirringsmatrise

Figur 36 viser hvordan en forvirringsmatrise ser ut og fungerer. De tre punktene langs diagonalen i matrisen er punkt hvor sant og antatt vev er like. Med andre ord er det ønskelig at tallene langs diagonalen er høye i forhold til de øvrige verdiene.

Forvirringsmatrisene som tilhører bildeeksempelet som er blitt fulgt ser ut som følger.

	<i>Sa</i>	<i>Sm</i>	<i>Sb</i>
<i>Aa</i>	364	19	527
<i>Am</i>	0	1937	385
<i>Ab</i>	92	997	19264

(a)

	<i>Sa</i>	<i>Sm</i>	<i>Sb</i>
<i>Aa</i>	790	11	246
<i>Am</i>	52	2001	454
<i>Ab</i>	245	804	20173

(b)

Figur 37: Forvirringsmatriser som viser treff og bompunkter for, watershedalgoritmen (a), og sirkulær tilnærming (b).

Matrisene i figur 37 er resultat av den direkte sammenlikningen mellom estimat og fasit. Matrise (a) er resultatet til watershedalgoritmen, og matrise (b) er resultatet etter sirkulær tilnærming.

Forvirringsmatrisene som inneholder resultatene til watershedalgoritmen og sirkulær tilnærming i figur 37 leses på følgende måte:

- Begge matrisene inneholder ni punkter, hvert av de ni punktene inneholder treffpunkter mellom Sann(S) (arr, myokard, bakgrunn) og Antatt(A)(arr, myokard, Bakgrunn).
- Punkt (1, 1) i matrisene viser summen av felles punkter mellom arrsegmentet som anses som fasit (Sann arr), og arrsegmentet som algoritmen har funnet frem til (Antatt arr).
- Punkt (1, 2) viser summen av felles punkter mellom segmentet som tilsvare friskt myokard i fasit (Sann myokard), og arrsegmentet fra algoritmen (Antatt arr).
- Punkt (1, 3) viser summen av felles punkter mellom bakgrunnsegmentet i fasit (Sann bakgrunn), og arrsegmentet fra algoritmen (Antatt arr).

Matrisene holder full oversikt over alle treffpunkter mellom de ulike segmentene, og blir brukt til å finne algoritmens ytelse. Ytelsen måles gjennom å beregne algoritmens treffsikkerhet, samt hvor ofte den bommer. Disse to tallene beregnes på følgende måte:

$$Treff_{arr} = \frac{(Aa, Sa)}{(Aa, Sa) + (Am, Sa) + (Ab, Sa)} \quad (8)$$

$$Bom_{arr} = \frac{(Aa, Sm) + (Aa, Ab)}{(Aa, Sa) + (Aa, Sm) + (Aa, Sb)} \quad (9)$$

$$Ytelse_{arr} = Treff_{arr} - Bom_{arr} \quad (10)$$

Likning 8 og 9 viser beregningen av treffprosent arr, og bomprosent arr. Beregningsmetoden er den eksakt samme for myokard, og bakgrunn. Et tilfelle som kan skape problemer med denne måten å beregne treff og bomprosent på, er når (Sa, Aa) eller $(Sm, Aa) + (Sb, Aa)$ ikke inneholder noen piksler. Dette vil gi en ugyldig brøk og matlab vil returnere NaN (Not-a-Number) som svar, på den måten blir beregningen verdiløs. Dette er løst på følgende måte:

For $Treff_{arr}$ når (Sa, Aa) er lik null:

Dersom $(Sa, Am) + (Sa, Ab)$ er lik null, betyr dette at det ikke er noe arrvev i fasiten. Det er med andre ord antatt at det ikke er noe arr i et tilfelle hvor det i realiteten ikke er noe arr. Treffprosent settes til 1.

Dersom $(Sa, Am) + (Sa, Ab)$ inneholder en verdi betyr dette at det eksisterer arrvev. Algoritmen har på sin side antatt at det ikke eksisterer noe arrvev.. Treffprosenten settes derfor til 0.

For Bom_{arr} når $(Sm, Aa) + (Sb, Aa)$ er lik null

Dersom $(Sm, Aa) + (Sb, Aa)$ er lik null, betyr dette at ingen piksler er feilklassifisert. Bomprosenten settes da lik 0.

Likning 8 og 9 beregner treffprosent og bomprosent for arr. Et identisk likningssett beregner resultatene for myokard og bakgrunnsvev også. Resultatene fra disse beregningene legges i hver sin matrise. Totalt for dette systemet blir det derfor seks matriser som inneholder alle resultatene.

Sett \ Snitt	1	2	3	4
1	0,4	0,5	0,5	0,4
2	0,3	0,3	0,4	0,3
3	0,4	0,3	0,4	0,4
4	0	0	1	1

Tabell 1: Viser hvordan en resultatmatrise kan se ut.

Tabell 1 viser et eksempel på hvordan en resultatmatrise kan se ut. Hver rad representerer et bildesett tilhørende en pasient. Mens hver kolonne representerer et bestemt utsnitt av hjertet. Disse matrisene blir fort veldig store. For å minske tallmengden vil gjennomsnittet av hver matrise bli beregnet.

Eksempel:

$$\begin{array}{cccc}
 0,4 & 0,5 & 0,5 & 0,4 \\
 0,3 & 0,3 & 0,4 & 0,3 \\
 0,4 & 0,3 & 0,4 & 0,4 \\
 0 & 0 & 1 & 1
 \end{array}
 \longrightarrow
 \begin{array}{c}
 0,45 \\
 0,325 \\
 0,375 \\
 0,5
 \end{array}
 \longrightarrow 0,4125$$

Først beregnes gjennomsnittet i matrisen for hver rad slik at resultatet blir en 4×1 -matrise, i neste steg beregnes gjennomsnittet for kolonnen som er igjen. Resultatet av dette blir et tall som representerer hele matrisen. Grunnen til at gjennomsnittet av matrisen beregnes på denne måten er at ikke alle bildesettene har samme antall utsnitt. Snittet av hver rad må derfor beregnes individuelt slik at det gjennomsnittlige tallet blir riktig.

I tillegg til resultatmatrisene vil det bli opprettet en tredje matrise for hvert av settene som vist i likning 10. I et totalt sett godt resultat vil treffprosenten være høy, og bomprosenten lav. Ved å trekke bomprosenten fra treffprosenten slik likning 10 viser, vil resultatet bli en matrise hvor algoritmens totale ytelse for hvert enkelt bilde kommer til syne. I de tilfellene hvor algoritmen gir høy treffprosent og lav bomprosent, vil ytelsesforholdet være høyt. I tilfellene hvor treffprosenten er lav og bomprosenten er høy, vil ytelsesforholdet være lavt. Denne tredje matrisen vil bli benyttet til å finne ulike bildeeksempler som illustrerer systemets ytelser.

3 Test av algoritmen

Til å teste algoritmen ble det benyttet 16 sett av 8-11 bilder pr sett. Dette gir totalt 144 ulike bilder som algoritmen blir testet på. Resultatene blir tolket i to omganger, først vil ytelsen til watershedalgoritmen bli vurdert. Og etterpå vil resultatene av den sirkulære tilnærmingen bli vurdert.

3.0.8 Resultat Watershed

I dette kapittelet vil resultatene til watershedalgoritmen bli presentert.

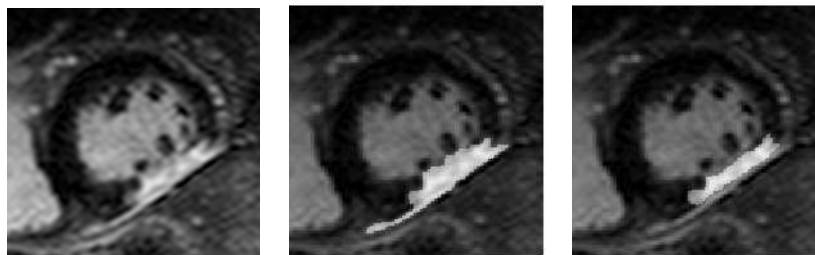
	Arrveg	Myokard	Bakgrunn
Treff i %	51,78	46,33	82,23
Bom i %	61,11	35,5	7,74

Tabell 2: Resultat Watershed

Tabell 2 viser resultatene etter watershedalgoritmen. Resultatene for hvert enkelt punkt er beregnet som vist i kapittel 2.3.7. Det er tydelig at algoritmen har støtt på problemer i dette testsettet. Bakgrunnsdeteksjonen er den klart beste, men det er enkelt forklart med at disse segmentene inneholder klart flest piksler. Unøyaktigheter og feilklassifiseringer blir ikke like synlige. Det beste resultatet som er av betydning i dette tilfellet er deteksjon av friskt myokard. Den har noe lavere treffprosent en arrdeteksjonen, men også mye lavere bomprosent. Myokarddeteksjon fungerer altså totalt sett bedre en hva arrdeteksjonen gjør.

For å gi et bedre bilde på hvordan systemet yter vil det bli presentert et utvalg bilder. Disse bildene vil ta for seg tilfeller hvor algoritmen har fungert bra, og tilfeller hvor algoritmen har fungert dårlig.

Arr:



(a) | (b) | (c)

Figur 38: Original (a), estimert arr (b), kardiologens fasit (c).

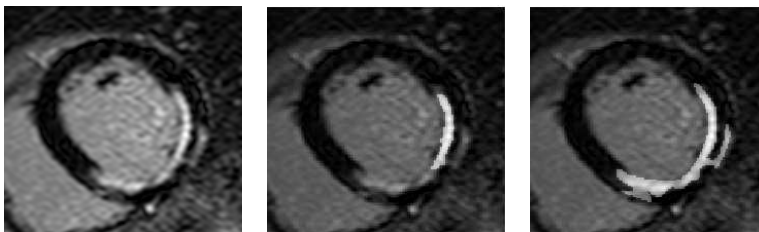
I figur 39 vises en brukbar arrdeteksjon. Figur 39(b) viser det estimerte arrsegmentet som watershedalgoritmen har returnert. Dersom dette bildet sam-

menliknes med fasiten som kan ses i figur 39(c), er det tydelig at deteksjonen treffer bra. I dette tilfellet er 99% av fasitsegmentet innenfor det estimerte segmentet. Det er mulig å se at det estimerte segmentet er større en kardiologens fasit, feilklassifiseringen er på 49%.

	Arrvev	Myokard	Bakgrunn
Treff i %	85,11	87,65	82,45
Bom i %	74,12	34,46	1,96

Tabell 3: Resultat Watershed, pasient fra figur 38

Tabell 3 viser det totale watershedresultatet fra alle utsnittene til pasienten fra figur 38. Hos denne pasienten er deteksjon av friskt myokard bra. Også treffprosenten på arrvev er høy, men dette er på bekostning av høy bomprosent.



(a) | (b) | (c)

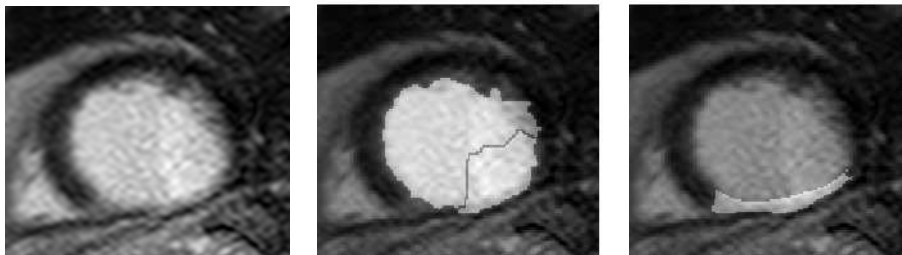
Figur 39: Original (a), estimert arr (b), kardiologens fasit (c).

Eksempelet i figur 39 viser et nytt arrsegment. I dette tilfellet kan det se ut til at markøren som skal beskrive arrsegmentet er blitt for liten, noe som fører til at det endelige arrestimatet (b) blir mindre enn kardiologens fasit (c). I dette tilfellet har algoritmen oppnådd en treffprosent på 32%, men til gjengjeld blir bare 5% feilklassifisert. Dette er med andre ord det motsatte av tilfellet som ble vist i figur 40.

	Arrvev	Myokard	Bakgrunn
Treff i %	40,9	81,04	86,34
Bom i %	69,27	24,65	4,85

Tabell 4: Resultat Watershed, pasient fra figur 39

Tabell 4 viser resultatet av alle utsnittene til pasienten i figur 39. Også i dette tilfellet er deteksjon av friskt myokard brukbart. Arrdeteksjonen over hele settet er for dårlig.



(a) | (b) | (c)

Figur 40: Original (a), estimert arr (b), kardiologens fasit (c).

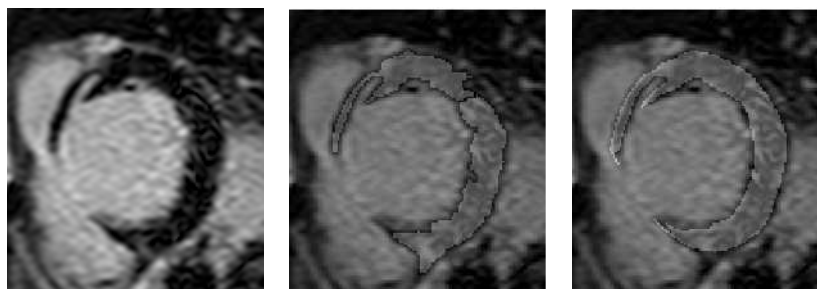
Figur 40 viser et tilfelle hvor algoritmen faller igjennom. I 40(a) er det tydelig at senteret av hjertet i dette tilfellet inneholder de lyseste gråtonene. Dette fører som vist i 40(b) til at hjertets senter blir antatt å være arret. Figur 40(c) viser kardiologens markering, her kan det ses at det kun er en smal stripe nederst som er det faktiske arret. Treffprosenten i dette tilfellet er på 70%, dette hjelper lite når feilklassifiseringsprosenten blir på 93%.

	Arrveg	Myokard	Bakgrunn
Treff i %	19,84	29,4	84,25
Bom i %	66,06	46,05	8,82

Tabell 5: Resultat Watershed, pasient fra figur 40

Tabell 5 viser resultatet fra alle utsnittene til pasienten i figur 40. I dette tilfellet er ingen av resultatene brukbare.

Myokardeksempler:



(a) | (b) | (c)

Figur 41: Original (a), estimert myokard (b), kardiologens fasit (c).

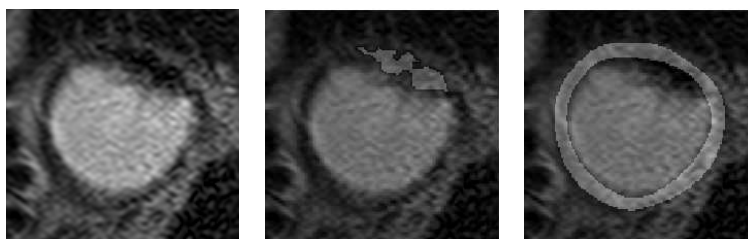
Figur 41 viser et tilfelle hvor myokard skal klassifiseres. (b) viser det estimerte myokardet, mens (c) viser kardiologens annotering. Det er tydelig at

watershedalgoritmen i dette tilfellet treffer veldig bra. Treffprosenten er på 94%, mens det kun er 7% feilklassifisering.

	Arrvev	Myokard	Bakgrunn
Treff i %	55,9	28,25	75,17
Bom i %	75,19	34,61	9,76

Tabell 6: Resultat Watershed, pasient fra figur 41

Resultatet når man ser på alle snittene til pasienten fra figur 41 som vises i tabell 6 er svake. Til tross for en god deteksjon av friskt myokard, trekkes snittet ned av andre dårlige resultater.



(a) | (b) | (c)

Figur 42: Original (a), estimert myokard (b), kardiologens fasit (c).

Figur 42 viser et problematisk tilfelle. Som det går frem av 42(a) er dette et bilde med dårlig kontrast mellom myokard og omliggende vev. Som følge av dette blir den morfologiske gradienten veldig svak, noe som igjen fører til at watershedalgoritmen har et vanskelig grunnlag å jobbe ut i fra. Resultatet av algoritmen kan ses i 42(b), og er at kun et lite mørkt parti blir estimert til å være friskt myokard. Treffprosenten er på 9%, mot 60% feilklassifisering.

	Arrvev	Myokard	Bakgrunn
Treff i %	76.45	28,52	74,85
Bom i %	96.52	38,3	9,95

Tabell 7: Resultat Watershed, pasient fra figur 39

Tabell 7 viser resultatet av alle snittene til pasienten fra figur 42.

3.0.9 Resultat Sirkulær Tilnærming

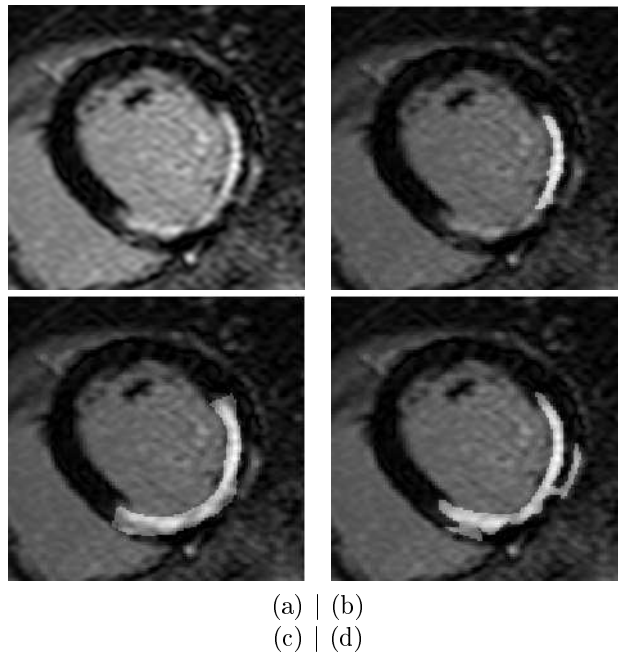
I dette avsnittet skal vi se på resultatene til algoritmen som utfører sirkulær tilnærming.

	Arrvev	Myokard	Bakgrunn
Treff i %	20,83	19,11	86,34
Bom i %	31,77	65,12	11,34

Tabell 8: Resultat Sirkulær Tilnærming

Resultatet som kan leses ut i fra tabell 8 tilsier at resultatet her ble dårligere enn det ble for watershed alene. Den eneste forbedringen til algoritmen for sirkulær tilnærming er feilkalssifisering arr. Grunnen til de svake resultatene er at funksjonaliteten ikke ble bygd robust nok. Den sirkulære tilnærmingen baserer seg direkte på watershedresultatene. Dette fører til at når watershedalgoritmen kommer frem til feilaktige segmenter, fungerer heller ikke avrundingsalgoritmen som den skal. Men som figurene under viser fungerer algoritmen tidvis veldig bra.

Arr:



Figur 43: Original (a), watershed estimert arr (b), sirkulært estimert arr (c), kardiologens fasit (d)

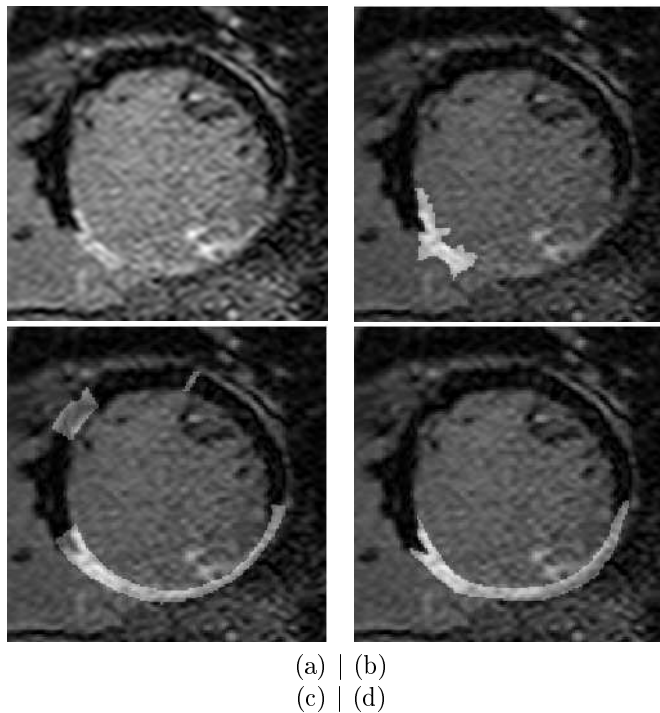
I figur 43 er det tydelig at estimatet (c), og fasiten (d) samsvarer ganske bra. Dette er det samme bildet som ble vist i figur 40. Dersom watershedestimatet (b) og estimatet etter sirkulær tilnærming (c) sammenliknes, er det tydelig at

estimatet etter sirkulær tilnærming inneholder langt flere treffpunkter. Treffprosenten er nå 76%, med en feilklassifiseringsprosent på 29%.

	Arrvev	Myokard	Bakgrunn
Watershed Treff i %	40,9	81,04	86,34
Watershed Bom i %	69,27	24,65	4,85
Sirkulær Treff i %	11,87	23,58	89,60
Sirkulær Bom i %	40,08	56,13	10,96

Tabell 9: Resultat Watershed og Sirkulær tilnærming, pasient fra figur 43

Tabell 9 viser resultatutviklingen fra watershed til sirkulær tilnærming. Tabellen tilhører pasienten fra figur 43. Det går her tydelig frem at selv om sirkulær tilnærming forbedret resultatet i bildeeksempelet, er ikke dette tilfellet for hele serien av bilder.



Figur 44: Original, watershed estimert arr, sirkulært estimert arr, kardiologens fasit

Figur 44 viser ytterligere et eksempel der et i utgangspunktet dårlig watershedresultat blir brukbart etter sirkulær tilnærming. I dette tilfellet går treffprosenten opp fra 33% etter watershed, til 77% etter sirkulær tilnærming.

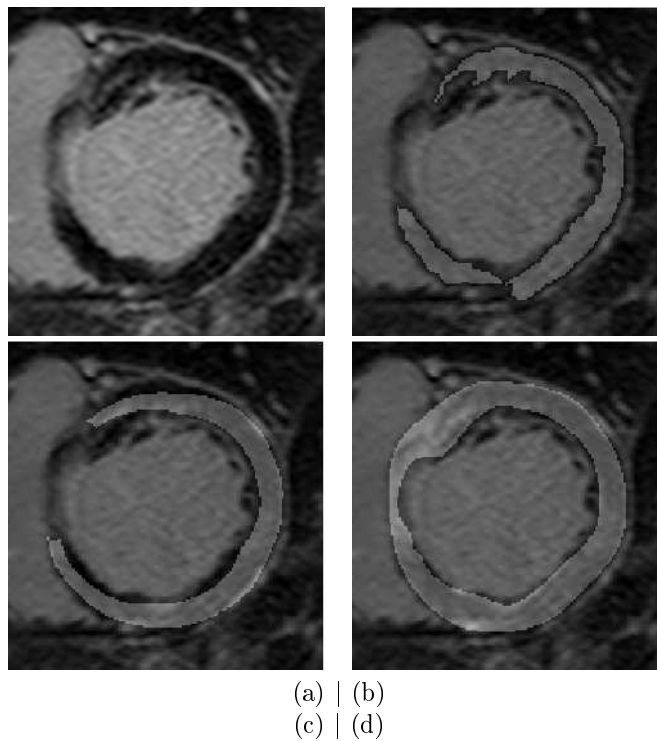
Andelen feilklassifiseringer går fra 43% etter watershed, til 40% etter sirkulær tilnærming. Totalt sett er forbedringen betydelig.

	Arrvev	Myokard	Bakgrunn
Watershed Treff i %	49,31	59,23	84,25
Watershed Bom i %	65,63	18,37	4,74
Sirkulær Treff i %	21,01	24,07	85,44
Sirkulær Bom i %	34,68	53,74	7,57

Tabell 10: Resultat Watershed og Sirkulær tilnærming, pasient fra figur 44

Tabell 10 viser at det totale resultatet av hjerteutsnittene til pasienten fra figur 44, blir dårligere etter sirkulær tilnærming.

Myokard:



Figur 45: Original, watershed myo, sirkulær tilnærming myo, kardiologens fasit

Figur 45 viser at sirkulær tilnærming også kan hjelpe til med å forbedre deteksjon av friskt myokard. Dette eksempelet illustrerer også hvordan et bilde

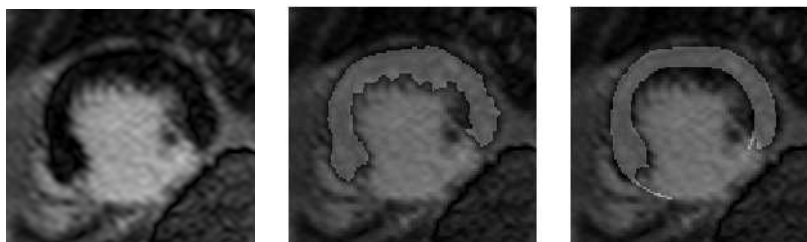
som skal segmenteres kan være vanskelig å tolke. Dersom man ser på figur 45(a). En naturlig antagelse vil være at den markant mørkere delen av “smultringen” er friskt myokard, mens overgangen hvor pikslene får verdier tilnærmet likt sentervevets, er arrvev. Dette er en antagelse algoritmen også tar, noe som kommer frem av figur 45(c). Men som det kommer frem av fasit i figur 45(d) er ikke dette tilfellet, her er alt blitt markert som friskt myokard. I dette tilfellet får algoritmen for sirkulær tilnærming en treffprosent på 59%, med en tilhørende feilklassifisering på 6%.

	Arrvev	Myokard	Bakgrunn
Watershed Treff i %	67,43	44,69	86,37
Watershed Bom i %	53,73	40,61	7,09
Sirkulær Treff i %	42,86	17,51	85,61
Sirkulær Bom i %	42,86	35,45	10,75

Tabell 11: Resultat Watershed og Sirkulær tilnærming, pasient fra figur 45

Tabell 11 viser resultatet av watershed og sirkulær tilnærming til hjertesnittene som tilhører pasienten i figur 45. Også her blir resultatet dårligere etter sirkulær tilnærming en det var med watershed. Resultatet for sirkulær treff og bom på arrsegmentet er i dette tilfellet likt. Grunnen til dette er at algoritmen har truffet 100%, og bommet 100% på like mange utsnitt.

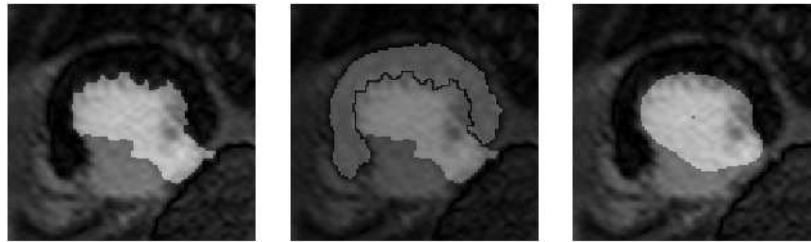
Som tidligere nevnt er sirkulær tilnærming en etterbehandling av watershedresultatene. Stort sett når watershedresultatene er blitt svake, har dette ført til svake resultater etter sirkulær tilnærming også. Det vil nå bli gjennomgått et eksempel hvor dette er tilfellet, med forklaring på hva som går galt.



(a) | (b) | (c)

Figur 46: Original, watershed myo, kardiologens fasit

I figur 46 er det vist et eksempel der estimert myokard er veldig nær fasttitsvaret. Man skulle da tro at etter en sirkulær tilnærming skulle bli sittende igjen med et nært perfekt resultat, men dette er ikke tilfellet.



(a) | (b) | (c)

Figur 47: arr, arr+myo, estimert myo

Figur 47 (a) viser antatt arrvev etter watershedalgoritmen. Sentervevet antas å være arr, noe som fører til følgende feil i algoritmen for sirkulær tilnærming:

Som beskrevet i kapittel 2.3.5 forventer algoritmen et svart senter, med rundtliggende arr og myokard. Det første steget i algoritmen tilsier at radiusen skal økes så lenge pikselverdien er svart. Når verdien til senterpikselet i dette tilfellet er hvit, hopper algoritmen umiddelbart ut av denne løkken og fortsetter til neste steg. Dette steget sier at radiusen skal økes så lenge pikselverdien ikke er svart. Dette fører til at løkken avsluttes så fort kanten til det feilaktige arrsegmentet nås. Som figur 47(c) viser blir resultatet en avrundet utgave av det feilklassifiserte arrsegmentet.

I de eksemplene som ble vist i dette kapittelet var ikke metoden for sirkulær tilnærming i stand til å forbedre resultatene for et helt sett, kun i enkelttilfeller. Grunnen til dette er ulike tilfeller som det ikke ble tatt høyde for under utviklingen av algoritmen. Dersom metoden hadde blitt gjort mer robust enn den er nå ville nok resultatene blitt mye bedre.

4 Konklusjon

Konklusjonen på dette prosjektet er at watershed ikke er en anbefalt metode for segmentering av MRI-avbildninger av hjertet. Det er flere grunner til dette.

- Watershedalgoritmen er avhengig av mange elementer for å virke på denne typen problemstillinger.
- Det er for stor variasjon i bildematerialet til at en gradientbasert metode kan anbefales, det er kun på gode bilder at resultatet ble brukbart.
- Bildematerialets kultur tilsier at det også er veldig utfordrende å finne gode markører for de ulike elementene.

Til tross for at sirkulær tilnærming ikke var i stand til å forbedre resultatene fra watershedalgoritmen, er oppfatningen at dette av like vel er en metode som det er verdt å se nærmere på.

5 Videre arbeid

Watershed

- Forbedre den morfologiske gradienten. Et av problemene som gjør at resultatene er variable er det faktum at gradienten ikke har markert enkelte kanter godt nok. Det er mulig resultatet hadde blitt bedre med en mer avansert gradienttype.
- Kontrastforsterkende forarbeid. Dersom man kan finne frem til en robust måte å forsterke kanter og kontraster på uten og ødelegge viktig bildeinformasjon, vil dette gjøre det enklere å finne en god gradient samt gode markører.
- Forbedre markørdeteksjon. Et annet av problemene som oppsto i denne prosessen var rundt deteksjon av gode markører, spesielt arrmarkøren kan være veldig utfordrende å finne. Det ble i denne oppgaven benyttet enkelte fastsatte parametere for markørdeteksjon, dette har vist seg å ikke fungere. Å finne frem til en god algoritme for markørdeteksjon, spesielt for arr og myokard er en enorm utfordring. Som det er blitt vist i kapittel 3, er det stor variasjon i skarphet og gråtone i MRI-avbildninger av hjertet. Deteksjonen av markører må fungere i alle tilfeller dersom et godt resultat skal kunne oppnås.

Sirkulær tilnærming

- Gjøre mer robust. Til tross for dårlige resultater under test kan dette være en lovende metode for avrunding. En test som ser om senterpikselet har verdien en eller null er uproblematisk å legge til. Det kan tenkes at ved å fjerne arrelementet dersom senterpikselet er en del av dette, ville gjort resultatet langt bedre.

- Utvide virkemåte. Slik tilnærmingen fungerer pr dags dato er det kun en glatting som blir foretatt, denne glattingen gjør at elementet blir mer sirkulært. Som bildematerialet har vist, kan det tenkes at hjertet i enkelte tilfeller har en noe mer elliptisk form. Dersom man kan få på plass en algoritme som kan gjøre en antagelse rundt eksakt form, og eventuell vinkel, kan det tenkes at dette vil gi et bedre resultat.

Alternativ løsning

Etter hvert som arbeidet med å fremstille algoritmene for watershed og sirkulær tilnærming ble gjennomført, har det også dukket opp et par idéer til en alternativ løsning av problemet. Løsningsmetodene det her er snakk om innebærer ikke bruk av watershed, og var derfor ikke av interesse for problemstillingen i denne oppgaven. Den alternative løsningsalgoritmen vil derfor bli beskrevet grovt her. En mer utfyllende beskrivelse, samt eksempler på virkemåte er vedlagt i Appendix.

- Finner hjertets senter
- Benytter analyse av form rett på gråtonebildet, enten en flatet utgave eller original.
- Mørke partier med stabil radius fra hjertets senter over flere vevsprøver antas å være friskt myokard.
- Lyse partier med stabil radius fra hjertets senter over flere vevsprøver antas å være arrvev.

Dette er kun en grov skisse av idéen, det er mulig at det er behov for forbehandling av bildene, eller etterbehandling. Men dette må det tas stilling til på et senere tidspunkt.

Referanser

- [1] Edward R. Dougherty and Roberto A. Lotufo. *Hands-on Morphological Image Processing (SPIE Tutorial Texts in Optical Engineering Vol. TT59)*. SPIE Publications, July 2003.
- [2] H. Engblom M. Götberg G. K. Olivecrona D. Erlinge E. Heiberg, M. Ugander and H. Arheden. Automated quantification of myocardial infarction from mr images by accounting for partial volume effects: animal, phantom, and human study. *Radiology 246(2) pp. 581-8*, 2008.
- [3] J. Engvall E. Hedstrom M. Ugander E. Heiberg, H. Engblom and H. Arheden. Semi-automatic quantification of myocardial infarction from delayed contrast enhanced magnetic resonance imaging. *Scand Cardiovasc J 39(5) pp. 267-75*, 2005.
- [4] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB, 2nd ed.* Gatesmark Publishing, 2nd edition.
- [5] A.U.J.; Alcaniz M.; Kikinis R.; Warfield S.K.; Grau, V.; Mewes. Medical imaging, iee transactions on. *Brigham & Women's Hosp. & Harvard Med. Sch., Boston, MA, USA, 23 Issue:4:447 – 458*, 2004.
- [6] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT3/node3.html.
- [7] <http://nhi.no/forside/animasjoner/hjerte-og-kar/hjertefarkt 31689.html>.
- [8] <http://nhi.no/sykdommer/hjerte-kar/koronarsykdom/hjertefarkt-oversikt 1515.html>.
- [9] <http://www.mathworks.com/products/demos/image/watershed/ipexwatershed.html>.
- [10] Rangaraj M. Rangayyan. *Biomedical Image Analysis*. CRC Press; 1 edition (December 30, 2004), 2005.
- [11] version 7.5.0 (R2007b). Mathworks inc., 2007.

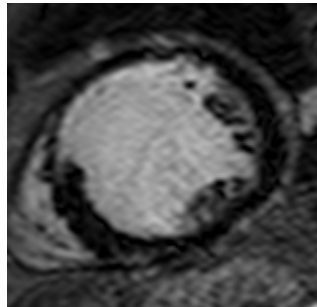
A Appendix

A.1 Alternativ løsning på problemet

Etterhvert som erfaringer er blitt gjort og forståelsen for problemstillingen har fått tid til å modne seg litt. Er det blitt gjort noen tanker rundt en alternativ løsning som kanskje kan gi et bedre resultat. Mye av problematikken rundt bruk av watershed, er at metoden er avhengig av mange elementer. Disse elementene må fungere bra individuelt, dersom et brukbart resultat skal oppnås. Både gradient og de tre ulike markørene må fungere bra for at et bra resultat skal oppnås.

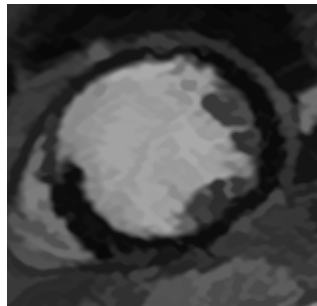
Den alternative løsningsmetoden ser derfor bort fra watershed som en del av løsningen, og baserer seg heller på analyse av form for å bestemme de ulike segmentene. I dette avsnittet vil metoden bli gjennomgått. Det er viktig å huske på at denne metoden foreløpig kun er på idestadiet.

Metoden baserer seg på enkeltelementer fra denne masteroppgaven som har vist seg å fungere bra. Det vil bli tatt utgangspunkt i et tilfeldig valgt bilde for å illustrere hvordan algoritmen skal fungere.



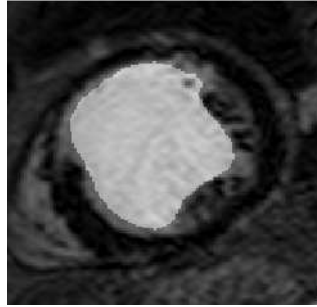
Figur 48: Bilde

Første steg i algoritmen er å benytte morfologisk rekonstruksjon til å flate ut bildet som er vist i figur 48.



Figur 49: Flatet bilde ved hjelp av morfologisk rekonstruksjon

Etter at bildet er flatet ut, bestemmes et segment som tilsvarer den indre muskulaturen.



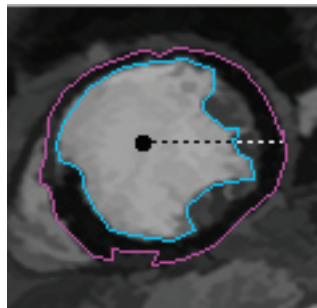
Figur 50: Senter

Dette sentersegmentet blir bukt til to ting:

1. Bestemmer midlere gråtoneverdi i senteret av hjertet, denne gråtoneverdien benyttes til å sette grenser for kantdeteksjon. Dette gjøres ved å se på alle gråtoneverdiene i det flatede bildet som tilsvarer segmentet funnet i figur 50, og tar gjennomsnittet av disse.
2. Sentersegmentet ligger på innsiden av arr og myokard. Dette segmentets senterpunkt ligger sannsynligvis i nærheten av hjertets senterpunkt. Det tas derfor utgangspunkt i dette segmentets senterpunkt når analysen av hjertet skal bli utført.

Etter dette steget er det to mulige veier som kan gi et resultat. Disse to mulighetene bør det eventuelt bli foretatt noen tester for å se hvilken som presterer best. Begge mulighetene baserer seg på fremgangsmåten beskrevet i kapittel 2.1.7.

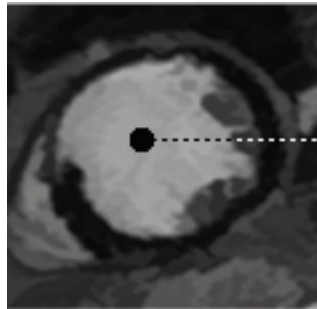
Den første muligheten er å ta utgangspunkt i den midlere gråtoneverdien som ble funnet i det forrige steget.



Figur 51: Senter

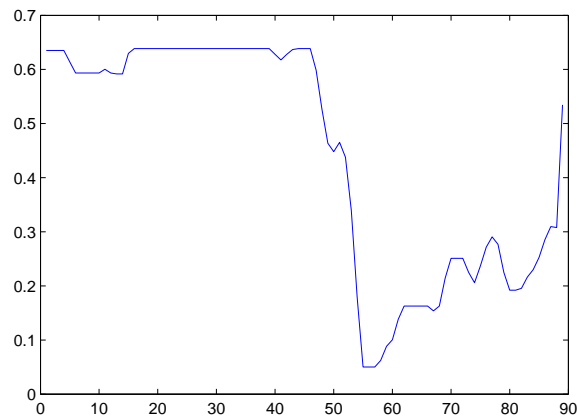
- Så lenge pikselverdien ligger innenfor en bestemt prosentandel av gjennomsnittlig gråtoneverdi, økes radiusen.
- Når gråtoneverdien blir lavere/høyere en tillatt verdi settes grense for indre myokard.
- Fortsetter å øke radiusen til det skjer en ny betydelig endring i gråtoneverdien, grense for ytre myokard settes.

Det andre alternativet er å kun benytte senterpunktet på denne måten:



Figur 52: Senter

- Radiusen øker frem til kanten på bildet nås, resultatet vil være en graf som inneholder alle gråtonene i snittet.



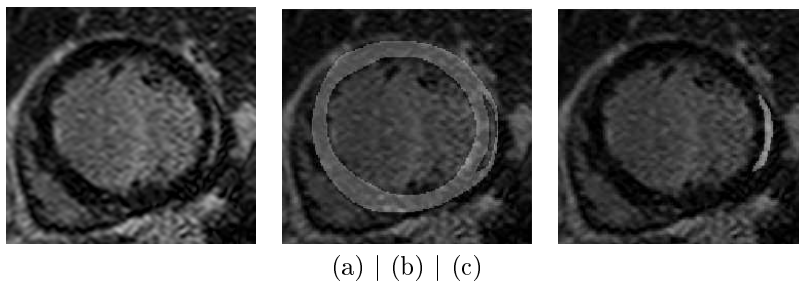
Figur 53: Utsnitt

Ut i fra dette utsnittet ligger myokardet som de laveste punktene (ca 55-70). Siden hjertet er relativt rundt og glatt burde ikke punktene hvor den myokar-

diale muskulaturen antas å være, endre seg vesentlig. I dette tilfellet burde myokardet ligge stabilt rundt punkt 45-80 fra senteret. Det kan derfor tenkes at ved å sammenlikne et utsnitt med tidligere prøver, kan enkelte utsnitt rettes på. De fleste arrstrukturene vil få en høyere verdi en hjertesenterets overflate, og identifiseres ved at kurven får en økning. Denne metoden er veldig enkel, men vil nettopp på grunn av det fungere i de aller fleste tilfeller. Utfordringen med metoden er å tolke de ulike kurvene, å ta en beslutning med tanke på hva som er arr og hva som er friskt myokard.

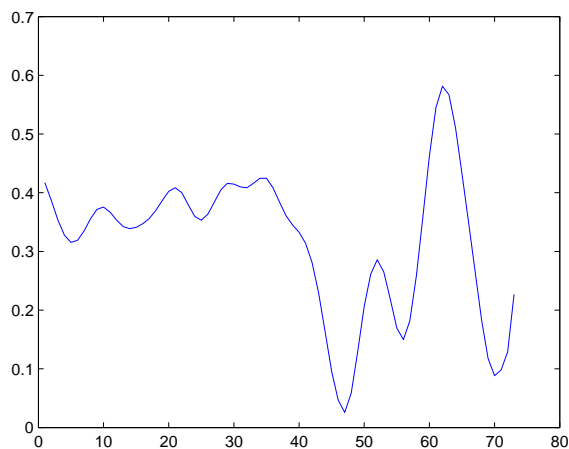
Om de to metodene fungerer som tiltenkt, vil resultatet nå være to vektorer som beskriver myokardets og arrets form nøyaktig. En glatting kan eventuelt vurderes for å jevne ut eventuelle avvik.

Av nysgjerrighet ble det kjørt en veldig liten test på et tilfeldig bilde for å teste det enkleste konseptet.



Figur 54: Fasit myo, fasit arr

Figur 54 viser et vanskelig tilfelle der det samme området inneholder både frisk muskulatur, og arrvev. Ved å benytte funksjoner som allerede er utviklet til å finne senteret til hjertet, ble et senterpunkt automatisk plukket ut. Ved å manuelt plukke ut utsnittet fra dette senterpunktet ut til høyre kant i bildet. Ble dette resultatet etter en smoothing.



Figur 55: Graf utsnitt

Det er her synlig begge områdene for frisk muskulatur (bunnpunkt i 47 og 57), i tillegg til arrvevet (toppunkt i 52). Det kan tenkes at man kan utvikle en algoritme som greier å ta høyde for disse dataene, og at man derfor greier å løse problemet.

A.2 Funksjon for generering av enkeltresultater

Det er blitt programmert en funksjon for å gjøre det enkelt å gjenskape enkeltresultater fra rapporten. Denne funksjonen heter `vis_resultat` (Pasientnummer, Utsnittnummer). Denne funksjonen genererer alle resultater for et enkelt bilde, og skriver disse ut på en oversiktlig måte. En beskrivelse av hvordan funksjonen brukes, og hvordan resultatene skal tolkes er beskrevet i filen `000_Read_me_000` på den medfølgende CDen.

Pasientnummer	Utsnittnummer	Som vist i figur nummer
17	7	Eksempelbilde brukt i kapittel 2
3	5	38
5	10	39, 43
1	5	40
6	9	41
14	4	42
13	8	44
10	7	45
15	4	46, 47

Tabell 12: Oversikt over bilder brukt i rapporten