

MASTER

Volgmethode voor versmeerde roentgenmarkerschaduwen

Bellussi, A.M.M.

Award date:
1997

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

VOLGMETHODE VOOR VERSMEERDE
RÖNTGENMARKERSCHADUWEN

door A.M.M. Bellussi



soort verslag:	afstudeerverslag
door:	A.M.M. Bellussi
afstudeerbegeleiders:	prof. dr. ir. T. Arts, UM prof. dr. ir. A. Hasman, TUE
vakgroepen:	Biofysica, UM MBS/Medische Elektrotechniek, TUE
periode:	mei 1996 - augustus 1997

Samenvatting

Aan de Universiteit Maastricht wordt onderzoek gedaan naar de lokale groei van het hart. Hiertoe worden bij honden metalen bolletjes in de hartwand geïmplanteerd. Over een periode van maanden worden de honden regelmatig doorgelicht met Röntgen, waarvan digitale beelden worden gemaakt. De metalen bolletjes worden afgebeeld als donkere schaduwen, die soms versmeerd zijn door bewegingsonscherpte. Met de huidige technieken zijn de versmeerde schaduwen niet goed identificeerbaar. In deze afstudeerstudie is een methode ontwikkeld waarmee (versmeerde) markerschaduwen geïdentificeerd en gevolgd kunnen worden. De volgmethode is gebaseerd op een hier ontwikkeld model van de markerbaan. In een optimalisatieproces convergeert het model naar de werkelijke markerbaan.

In beelden waarin kunstmatig versmeerde markerschaduwen zijn aangebracht, zijn de verschillen gemeten tussen gedefinieerde marker-centrumposities en door de volgmethode geschatte marker-centrumposities. De verschillen zijn maximaal 15% van de markerdiameter in pixels.

De ontwikkelde methode is in staat de versmeerde markerschaduwen in echte videobeelden automatisch te volgen. Hierbij is de invloed van het contrast van de markerschaduwen op het resultaat onderzocht.

Inhoudsopgave

1.	Inleiding en probleemstelling	5
2.	Modellering van het markervolgprobleem	8
2.1	Model van de markerbaan	8
2.2	Model van het smeergebied langs een baansegment	11
3.	De volgmethode	14
3.1	Vergelijken van model en werkelijkheid	14
3.2	Filtering van de Röntgenmarkerbeelden	16
3.3	Volgen van markers door optimalisatie van BIP's	20
3.3.1	Initiële schattingen en extrapolatie van de BIP's	22
3.3.2	Schatting van de markerposities in de eerste twee beelden	24
3.3.3	Verfijning van schattingen door interpolatie	26
4.	Volgvermogen in gesimuleerde smeerbeelden	30
4.1	Variatie van markerbaan en markersnelheid	30
4.2	Invloed van Gaussische ruis	33
5.	Volgvermogen in videobeelden	36
6.	Conclusies	41
	Appendix	42
A	Handleiding bij het markervolgprogramma	42
B	Source code	46
C	Compileren van de source code	47

D	Datastructuur Multi-Marker Tracking	49
	Lijst van begrippen, symbolen, subscripts en superscripts	52
	Begrippen	52
	Symbolen	55
	Subscripts	58
	Superscripts	59
	Literatuurlijst	60

1. Inleiding en probleemstelling

Bij de vakgroep Biofysica van de faculteit der Geneeskunde aan de Universiteit Maastricht vindt onderzoek plaats naar de lokale groei van de hartwand van honden. De groei wordt onderzocht over een periode van enkele maanden. Met fysiologische experimenten wordt de regionale beweging en deformatie van de hartwand gemeten. De huidige technieken zijn gebaseerd op twee-dimensionale echocardiografie, magnetic resonance imaging (MRI) en op het volgen van in de hartwand geïmplanteerde markers. Echocardiografie heeft een beperkte resolutie en identificeerbare punten kunnen niet gevolgd worden in drie dimensies. Met tagged MRI, een methode waarbij een raster in het myocard gemagnetiseerd wordt, kan beweging van het myocard afgeleid worden uit de verplaatsing van het rasterpatroon. De labels hebben een beperkte levensduur van maar enkele tienden van een seconde waardoor de tagged MRI methode ongeschikt is voor onderzoek naar de lokale groei van de hartwand. In ons onderzoek wordt regionale beweging en deformatie van het hart gemeten door het volgen van in de hartwand bevestigde markers in een serie beelden. Metalen bolletjes met een diameter van 1,5 mm worden tijdens een open hart operatie in de hartwand geïmplanteerd, waarna de borstkas gesloten wordt. De grootte van de bolletjes is zo gekozen dat ze kunnen worden afgebeeld en ze de dynamica van het hart niet verstoren. MRI als afbeeldingstechniek is niet geschikt omdat de markers te klein zijn. Het doorlichten van het hart met Röntgen is wel een geschikte afbeeldingstechniek. Een Röntgenbron wordt op 40 cm boven het hart geplaatst, 40 cm onder het hart zit een fluoricerende plaat die de invallende Röntgenstralen omzet in lichtpulsjes. De lichtpulsjes worden in een beeldversterker omgezet in elektrische signalen die naar een CCD-camera worden geleid. De beelden van de CCD-camera worden met een videorecorder op tape opgeslagen. De drie dimensionale posities van de markers worden gemeten door twee Röntgenopnamen te maken waarbij de afbeeldingsvlakken circa loodrecht op elkaar staan [1]. Bij de eerste toepassingen van de twee-vlak Röntgentechniek moest een persoon de markercoördinaten met de hand aangeven. Dit was een tijdrovend karwei. Een eerste aanzet tot het automatisch herkennen van de markers was de procedure ontwikkeld in [2]. Markers werden elektronisch herkend in iedere videolijn op basis van edge

detectie, grootte en locatie. Het volgprobleem (identificatie en bepalen van de correspondentie van de markers in opeenvolgende frames) werd later opgelost door toepassing van een adaptieve lineaire predictor op de markercoördinaten [3]. Deze predictor werd gebruikt in een automatisch systeem voor de detectie en het volgen van de markers [4]. Het volgsysteem werd verder verfijnd door toepassing van digitale beeldverbetering en segmentatietechnieken [5].

Het volgen van de markers lukt niet in beelden waarin de markers te weinig contrasteren met de achtergrond, of door occlusie bedekt raken. Het volgproces wordt dan onderbroken waardoor incomplete markersporen ontstaan. Incomplete sporen kunnen geëxtrapoleerd worden door de informatie te gebruiken uit de bekende gedeelten van de onderbroken sporen en de complete sporen [6]. Wanneer de marker te snel beweegt in relatie tot de sluitertijd van de camera, versmeert het markerbeeld. Vaak wordt de sluitertijd zo groot mogelijk gekozen om een contrastrijk beeld te krijgen. Hierdoor wordt de kans op smeer groter. Met de huidige technieken kunnen de gedeeltelijk versmeerde markers niet gevolgd worden.

Het doel van dit afstudeeronderzoek is het ontwikkelen van een methode waarmee ook de versmeerde markers in gedigitaliseerde Röntgenbeelden automatisch gevolgd kunnen worden. De problemen die opgelost moeten worden zijn het automatisch identificeren en het volgen van de (versmeerde) markers.

Er is een literatuurstudie uitgevoerd naar methoden voor het identificeren en volgen van bewegende objecten. De in de gevonden artikelen beschreven methoden zijn in te delen naar gebruik van correlatie [7], substractie [8], contourdetectie [9,10,11,12,13,14] en bewegingsmodellen [15,16,17,18]. De methoden die op bewegingsmodellen baseren veronderstellen voorkennis van de objectsnelheid of een constante translationele en rotationele dynamica van de bewegende objecten. De markersnelheid is op voorhand onbekend en een marker beweegt met variabele versnelling. De genoemde bewegingsmodellen zijn daarom niet toepasbaar voor het markervolgprobleem. Bij contourdetectie worden objecten gesegmenteerd op basis van contouranalyse. Actieve contourmodellen (snakes) kunnen bewegende contrastranden detecteren en volgen, mits de contouren niet te snel bewegen. De (versmeerde) markers zijn klein (enkele pixels in doorsnede) en vertonen soms grote snelheidsverschillen, waardoor de

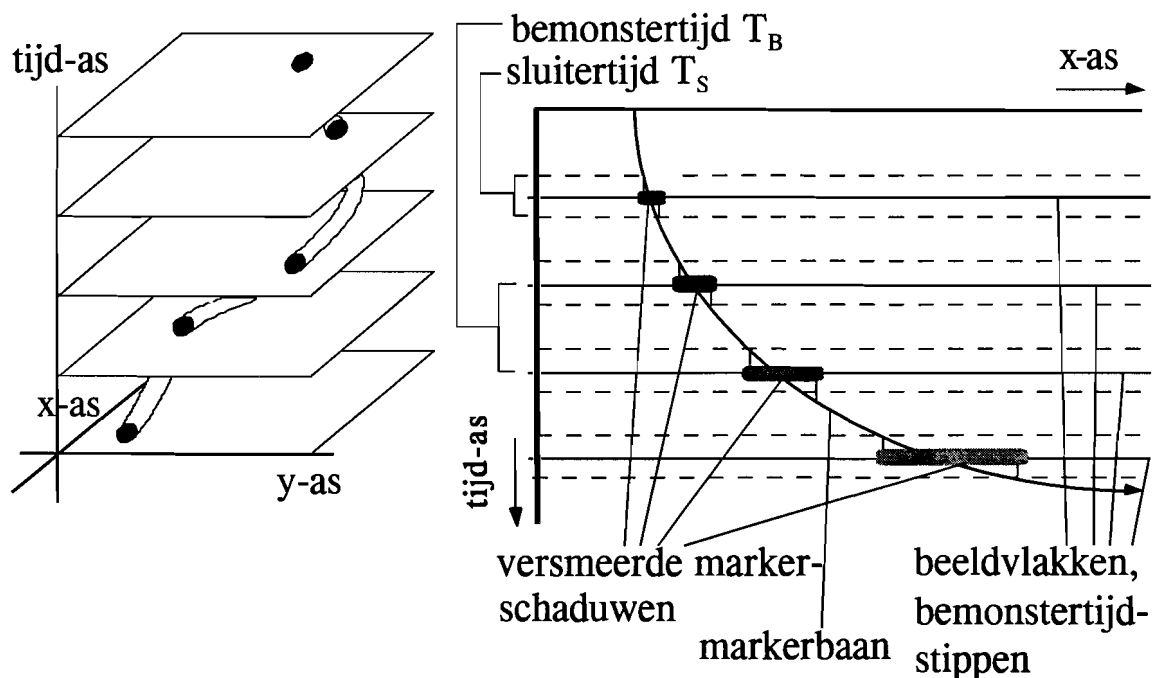
snakes bij het volgen kunnen ontsporen. De substractiemethode beschreven in [8] berekent het verschilbeeld tussen twee opeenvolgende beelden. Door bewegende achtergrondstructuren in de Röntgenmarkerbeelden kunnen de markers in het verschilbeeld niet of nauwelijks geïdentificeerd worden. Template matching wordt toegepast in de correlatiemethode [7]. Het beeld van een versmeerde marker in de beeldsequentie varieert in grootte en helderheid, al naar gelang de markersnelheid. Dit betekent dat voor de identificatie per beeld een aparte template nodig is. Hierdoor is ook deze methode niet geschikt. De conclusie van het literatuuronderzoek is dat geen enkel van de methoden zonder verdere aanpassing geschikt is voor het volgen van onze Röntgenmarkers.

Binnen de vakgroep is het idee ontstaan de (versmeerde) markers te volgen door gebruik te maken van een model van de markerbaan. De volgmethode die in dit verslag wordt ontwikkeld maakt gebruik van dat model. In hoofdstuk 2 wordt het model van de markerbaan en de smeer uitgewerkt. De volgmethode wordt beschreven in hoofdstuk 3. In hoofdstuk 4 wordt de volgmethode getest in gesimuleerde en in hoofdstuk 5 in echte Röntgenbeelden. Het laatste hoofdstuk bevat de algemene conclusies.

2 Modelling van het markervolprobleem

2.1 Model van de markerbaan

Een bewegende marker doorloopt een baan in de plaats-tijd ruimte. Deze ruimte wordt opgespannen door drie orthogonale richtingsvectoren die wijzen in respectievelijk de x, y en tijd richting, zie ook figuur 1 links. De positie van de marker in de plaats-tijd ruimte wordt bepaald door de volgende drie parameters:



Figuur 1. Links een markerbaan door de plaats-tijd-ruimte. Rechts een markerbaan geprojecteerd op het x-tijd-vlak. De versmeerde markerschaduwen ontstaan door beweging van de marker wanneer de camera sluitertijd open staat.

Het beeld van de marker wordt gedurende de tijd dat de camera sluitertijd open staat, T_s , geprojecteerd op een beeldvlak in de plaats-tijd-ruimte waarvoor geldt dat de tijd gelijk is aan het bemonstertijdstip $t_{B,i}$. Wanneer de marker voldoende snel beweegt of de sluitertijd voldoende groot is, ontstaat in dat vlak een markerspoor. De lengte van het spoor is evenredig met de markersnelheid. De helderheid van het spoor wordt bepaald door de markersnelheid en heeft in de spoor-dwarsdoorsnede de vorm van een vallei. Het hoogteverschil tussen de rand en het dal is omgekeerd evenredig met de markersnelheid. Wanneer de marker met constante snelheid beweegt zal de positie van het markermidpunt op het bemonstertijdstip samenvallen met het midden van de (versmeerde) markerschaduw. De markerschaduwen in ieder beeld kunnen worden geprojecteerd op één beeldvlak. Door de achtereenvolgende markerschaduwen kan in dat beeldvlak een baan gelegd worden: de geprojecteerde markerbaan.

Deze markerbaan wordt gemodelleerd. Voorlopig nemen we aan dat de modelmarkerbaan continue is. De modelmarkerbaan bestaat uit baansegmenten, één voor ieder geprojecteerd beeldvlak. Een baansegment in het geprojecteerde beeldvlak voor het bemonstertijdstip $t_{B,i}$ wordt gedefinieerd met behulp van drie baan-interpolatie-punten (in het vervolg aangeduid als BIP's), \mathbf{P}_{i-1} , \mathbf{P}_i en \mathbf{P}_{i+1} uit de geprojecteerde beeldvlakken voor t gelijk aan respectievelijk $t_{B,i-1}$, $t_{B,i}$ en $t_{B,i+1}$. Het coördinaatsysteem waarbinnen de BIP's zijn gedefinieerd bestaat uit een x -as en loodrecht daarop een y -as, waarvoor geldt $\{x \in \mathbb{R} \mid 0 \leq x \leq N-1\}$ en $\{y \in \mathbb{R} \mid 0 \leq y \leq M-1\}$ met N, M het aantal pixels in de x respectievelijk de y richting van het beeldvlak. De oorsprong valt samen met de linker bovenhoek van het beeld. De coördinaat $(N-1, M-1)$ valt samen met de linker benedenhoek van het beeldvlak.

De tijd-as wordt genormaliseerd tot τ volgens: $\tau = (t - t_{B,i})/T_B$

Tussen \mathbf{P}_{i-1} , \mathbf{P}_i en \mathbf{P}_{i+1} wordt een 2^{de} orde baan (parabool) geïnterpoleerd die voldoet aan de volgende voorwaarde:

- De baan raakt aan het midden van de verbindingslijn tussen \mathbf{P}_{i-1} en \mathbf{P}_i en het midden van de verbindingslijn tussen \mathbf{P}_i en \mathbf{P}_{i+1} . Voor het eerste en tweede raakpunt geldt:

$$t = t_{B,i} - \frac{1}{2} * T_B \quad \text{en} \quad \tau = -0.5$$

respectievelijk

$$t = t_{B,i} + \frac{1}{2} * T_B \quad \text{en} \quad \tau = 0.5$$

Een baanpunt op het baansegment voor $-0.5 \leq \tau \leq 0.5$ wordt gedefinieerd door formule 2.1.1.

$$B(\tau) = \frac{1}{8}(P_{i-1} + 6P_i + P_{i+1}) + \tau \frac{1}{2}(P_{i+1} - P_{i-1}) + \tau^2 \left(\frac{1}{2}(P_{i+1} + P_{i-1}) - P_i \right) \quad (2.1.1)$$

B baanpunt met coördinaten (B_x, B_y) op tijdstip $-0.5 \leq \tau \leq 0.5$

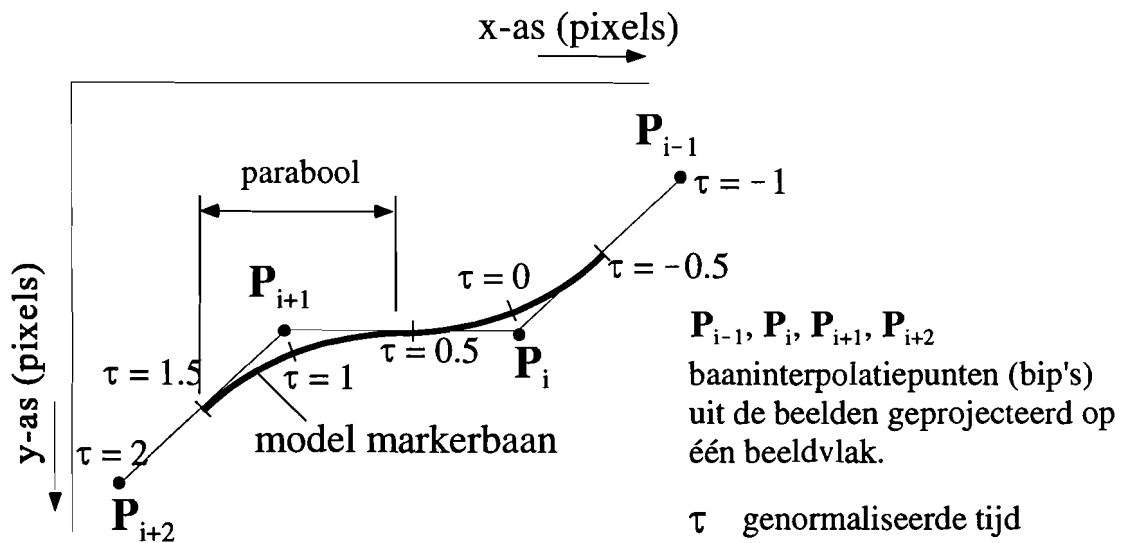
τ de genormaliseerde tijd

$\tau = -1$ voor P_{i-1}

$\tau = 0$ voor P_i

$\tau = +1$ voor P_{i+1}

De op bovenbeschreven wijze gedefinieerde baansegmenten vormen samen een continue baan. De modelmarkerbaan is weergegeven in figuur 2.1.2. De positie van de baansegmenten wordt bepaald door de positie van de BIP's. De BIP's dienen zo gekozen te worden (via een optimalisatieproces) dat de segmenten zo goed mogelijk samenvallen met de markerschaduw.



Figuur 2.1.2 De gemodelleerde markerbaan geprojecteerd op één beeldvlak.

2.2 Model van het smeergebied langs een baansegment

Gedurende de sluitertijd T_S laat een bewegende marker in het beeld i een spoor achter waarvan de lengte L afhankelijk is van de markersnelheid. De baanvergelijking van het stukje spoor wordt gegeven door formule 2.1.1 voor $-0.5*(T_S/T_B) \leq \tau \leq 0.5*(T_S/T_B)$. De lengte L wordt gegeven door formule 2.2.2.

$$L = \int_{-\frac{1}{2} \frac{T_S}{T_B}}^{\frac{1}{2} \frac{T_S}{T_B}} \sqrt{\mathbf{B}' \cdot \mathbf{B}'} d\tau \quad \text{met } \mathbf{B}' = \frac{\partial \mathbf{B}}{\partial \tau} \quad (2.2.2)$$

Ter vereenvoudiging van het rekenwerk wordt het baansegment in beeld i benaderd door het rechte lijnstuk: $\frac{1}{2}*(T_S/T_B)*(P_{i+1} - P_{i-1})$. De lengte s van het lijnstuk is dan gelijk aan:

$s = \frac{1}{2}*(T_S/T_B)*\|P_{i+1} - P_{i-1}\|_2$, , waarbij de uitdrukking tussen de $\|$ -tekens de euclidische lengte is.

De lengte L van het markerspoor kan dan benaderd worden door \hat{s} . In het vervolg van dit verslag wordt steeds gerekend met s (naar boven afgerond op hele pixels) als de lengte van het markerspoor in het beeld. Ook wordt aangenomen dat $T_S = T_B$. Het geprojecteerde markerspoor wordt daarmee continu.

Op het tijdsinterval $-0.5 \leq \tau \leq 0.5$ worden een aantal hulptijdstippen τ_k gedefinieerd:

$$\tau_k = -0.5*(1-\tau_h) + k*\tau_h, \quad \text{voor } k = [0, 1, 2, \dots, (1/\tau_h) - 1]$$

Het hulptijdstapinterval τ_h is de tijdsduur waarin de marker een afstand van 1 pixel aflegt over het spoor in het beeld. De gemiddelde snelheid van de marker over het spoor in het beeld is gelijk aan:

$$v_{\text{marker,gem}} = s / \Delta\tau = s, \quad \text{want } \Delta\tau = 0.5 - (-0.5) = 1$$

Het hulptijdstapinterval wordt dan gegeven door:

$$\tau_h = 1 / v_{\text{marker, gem}} = 1 / s$$

Het smeergebied voor het baansegment in beeld i wordt gedefinieerd door een aantal geïnterpoleerde posities, verder aangeduid als smeergebiedpunten, zie figuur 3. :

- op het baansegment:

Voor iedere τ_k worden de x - en y -coördinaten, I_x respectievelijk I_y , van de smeergebiedpunten $\mathbf{I}_{\text{baan}}(\tau_k)$ op de baan berekend volgens formule (2.1.1), waarbij $\tau = \tau_k$

- buiten het baansegment:

Voor iedere τ_k wordt een raakvector $\mathbf{r}(\tau_k)$ berekend volgens:

$$\mathbf{r}(\tau_k) = \partial \mathbf{B}(\tau_k) / \partial \tau \quad \text{met } \mathbf{B} \text{ uit formule (2.1.1)} \quad (2.2.3)$$

$\mathbf{r}(\tau_k)$ wordt genormeerd op lengte 1 pixel. Met behulp van $\mathbf{r}(\tau_k)$ wordt de normaalvector $\mathbf{n}(\tau_k)$ gedefinieerd.

$$n_x(\tau_k) = -r_y(\tau_k) \quad \text{en } n_y(\tau_k) = r_x(\tau_k)$$

Het aantal smeergebiedpunten buiten het baansegment op τ_k hangt af van de grootte van de marker loodrecht op de bewegingsrichting, uitgedrukt in de markerstraal r_{marker} in pixels. r_{marker} wordt afgerond naar beneden op een geheel getal. De smeergebiedpunten buiten het baansegment, $\mathbf{I}_{\text{buiten}, j}(\tau_k)$ worden gedefinieerd als:

$$\mathbf{I}_{\text{buiten}, j}(\tau_k) = \mathbf{I}_{\text{baan}}(\tau_k) + j \cdot \mathbf{n}(\tau_k), \quad \text{voor } j = [-r_{\text{marker}}, \dots, -2, -1, 1, 2, \dots, r_{\text{marker}}]$$

Voor $k = 0$ en $k = 1/\tau_h - 1$ worden nog een aantal extra smeergebiedpunten gedefinieerd in de bewegingsrichting van de marker. Op $\tau_k = -0.5 \cdot (1 - \tau_h)$ geldt:

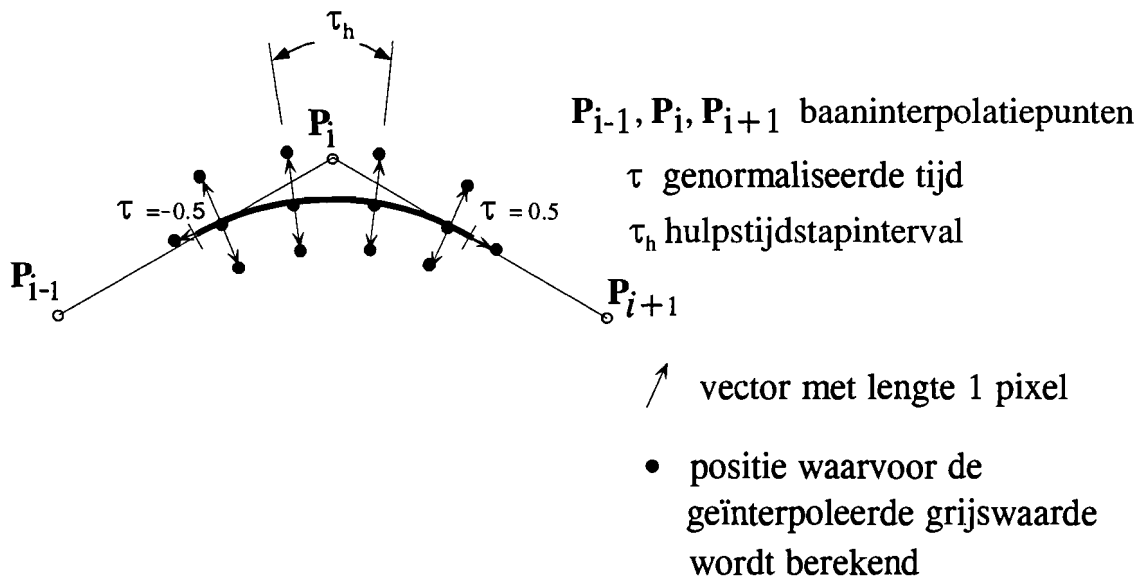
$$\mathbf{I}_{\text{achter}, j}(-0.5 \cdot (1 - \tau_h)) = \mathbf{I}_{\text{baan}}(-0.5 \cdot (1 - \tau_h)) + j \cdot \mathbf{r}(-0.5 \cdot (1 - \tau_h)), \quad \text{voor } j = [-r_{\text{marker}}, \dots, -2, -1]$$

Op $\tau_k = -0.5 \cdot (1 - \tau_h) + (1/\tau_h - 1) \cdot \tau_h$ geldt:

$$\mathbf{I}_{\text{voor},j}(-0.5*(1-\tau_h)) = \mathbf{I}_{\text{baan}}(-0.5*(1-\tau_h)) + j*\mathbf{r}(-0.5*(1-\tau_h)), \text{ voor } j = [1, 2, \dots, r_{\text{marker}}]$$

De smeergebiedpunten worden gebruikt bij de berekening van de grijswaardenmatrix van het smeergebied, $S_{q,r}$ in beeld i . In iedere kolom staan steeds de geïnterpoleerde grijswaarden uit het discrete beeld $G_{m,n}$ van $\mathbf{I}_{\text{baan}}(\tau_k)$ en $\mathbf{I}_{\text{buiten}}(\tau_k)$. De eerste en laatste kolommen bevatten de geïnterpoleerde grijswaarden voor $\mathbf{I}_{\text{achter},j}$ respectievelijk $\mathbf{I}_{\text{voor},j}$, opgevuld met nullen zodat de kolommen even groot zijn als de kolommen voor $\mathbf{I}_{\text{baan}}(\tau_k)$ en $\mathbf{I}_{\text{buiten}}(\tau_k)$. Het totaal aantal kolommen van $S_{q,r}$ is gelijk aan: $1/\tau_h - 1 + 2*r_{\text{marker}}$.

De interpolatiemethode die toegepast wordt is de bilineaire interpolatie. Deze methode gebruikt de grijswaarden van de vier dichtst bij zijnde pixels in $G_{m,n}$ van ieder smeergebiedpunt. De bilineaire interpolatiemethode geeft een nauwkeuriger resultaat dan nearest neighbour en is minder rekenintensief dan spline interpolatie.



Figuur 2.2.1 Het gemodelleerde smeergebied voor een marker met $3 < \text{diameter} < 4$ pixels

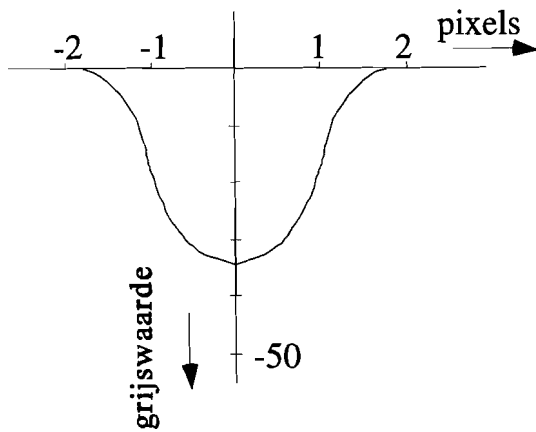
3 De volgmethode

3.1 Vergelijken van model en werkelijkheid

De gemodelleerde markerbaan dient gefit te worden op de werkelijke baan. Hiertoe moeten de posities van de BIP's juist gekozen worden. Het fitten van een model op de werkelijkheid kan in het algemeen met verschillende methoden. Een methode maakt gebruik van substractie. De grijswaarden van het model worden afgetrokken van de werkelijke grijswaarden. Bij substractie is het resultaat afhankelijk van de absolute grijswaarden in het beeld en van het model. De grijswaarden van de markers in het beeld verschillen onderling sterk door de invloed van de achtergrond. Hierdoor is substractie niet geschikt om de markers te lokaliseren. Een andere methode maakt gebruik van kruiscorrelatie tussen model en werkelijkheid. Hierbij wordt de *grijswaardenverdeling* van het markermodel $M_{q,r}$ vergeleken met de grijswaardenverdeling in het werkelijke beeld $S_{q,r}$. Het voordeel van het correleren van grijswaardenverdelingen is dat het resultaat van de kruiscorrelatie niet afhangt van de absolute grijswaarden van model en werkelijkheid.

$M_{q,r}$ wordt afgeleid uit de absorptiefunctie voor markers in Röntgenbeelden. De absorptiefunctie beschrijft de helderheid als functie van de plaats van het markerbeeld. Het blijkt dat de helderheidsverdeling dwars door het centrum van de marker volgens de absorptiefunctie de vorm heeft van een u-profiel. De werkelijke helderheidsverdeling van de marker in het beeld lijkt meer op de vorm van een goot. De afwijking wordt verklaard uit het feit dat het afbeeldingssysteem een vervorming introduceert. De vervorming wordt gemodelleerd door de absorptiefunctie te convolueren met een Gaussfunctie. Variatie van de spreiding van de Gauss-functie heeft als resultaat dat de vorm van het markerbeeld verandert. Bij een bepaalde spreiding is de vorm van de helderheidsverdeling volgens de geconvolueerde absorptiefunctie nagenoeg gelijk aan de werkelijke markerhelderheidsverdeling in het beeld. Figuur 3.1.1 laat zien de gemodelleerde helderheidsverdeling van de dwarsdoorsnede door het centrum van een

niet versmeerde marker.



Figuur 3.1.1. De gemodelleerde grijswaardenverdeling van de dwarsdoorsnede door het centrum van een niet versmeerde marker. Het grijswaardeniveau van de achtergrond is gelijk aan 0 gesteld. De marker heeft hier een diameter van 3.5 pixels.

Het correlatieproces wordt uitgedrukt in de volgende doelfunctie:

$$C(\mathbf{P}_i) = \frac{-1}{(Q_{\max} - 1) * (R_{\max} - 1)} \sum_{(q=0), (r=0)}^{(Q_{\max}-1), (R_{\max}-1)} S_{q,r}(\mathbf{P}_i) * M_{q,r}(\mathbf{P}_i) \quad (3.1.1)$$

- $C(\mathbf{P}_i)$ = correlatie waarde voor punt \mathbf{P}_i
 $S_{q,r}(\mathbf{P}_i)$ = grijswaardenmatrix van smeergebied rond BIP \mathbf{P}_i
 $M_{q,r}(\mathbf{P}_i)$ = modelgrijswaardenmatrix van smeergebied rond BIP \mathbf{P}_i
 q = rij-index, $q = [0, 1, 2, \dots, Q_{\max} - 1]$
 r = kolom-index, $r = [0, 1, 2, \dots, R_{\max} - 1]$
 Q_{\max} = aantal rijen, $Q_{\max} = 2 * r_{\text{marker}}$
 R_{\max} = aantal kolommen, $R_{\max} = 1/\tau_h - 1 + 2 * r_{\text{marker}}$

De waarde van de doelfunctie is afhankelijk van de positie van het BIP in het beeld. Bij die positie van het BIP waarvoor de waarde van de doelfunctie maximaal is geldt dat de gemodelleerde marker-baan het dichtst bij de werkelijke markerbaan ligt.

3.2 Filtering van de Röntgenmarkerbeelden

Röntgenbeelden van een hondhart met geïmplanteerde markers hebben een laag contrast. Door het contrast te vergroten kunnen meer markers gedetecteerd worden. Contrastvergroting kan op verschillende manieren worden bereikt. Een manier is de dosis van Röntgenstralen te verhogen. Hierbij ontstaat het gevaar van stralingsschade van het weefsel en van de onderzoeker. Een andere manier is de signaalruisverhouding in de Röntgenbeelden te vergroten door de beeldmatrix $G_{m,n}$ te convolueren met een banddoorlaat filtermasker. Dit heeft gevolgen voor de correlatiefunctie. Een banddoorlaatfilter kan in het spatiële domein worden voorgesteld door zijn impulsresponsie $h(x,y)$. Om de correlatiepiek zo hoog mogelijk te houden moet zowel het model als de werkelijkheid geconvolueerd worden met het bandfilter. In formulevorm:

$$\begin{aligned} \text{Correlatie} &= w \oplus h \oplus m \oplus h \\ &= m \oplus w \oplus h \oplus h \approx m \oplus w \oplus h, \quad \text{omdat geldt: } h \oplus h \approx h \end{aligned}$$

w = werkelijk markerbeeld, m = model markerbeeld,

h = impulsresponsie van banddoorlaatfilter,

\oplus = convolutie operator.

Uit bovenstaande afleiding blijkt dat het voldoende is om alleen het werkelijke beeld te bandfilteren en het gebied waarover telkens de correlatie bepaald wordt blijft beperkt.

Er wordt een afstembaar banddoorlaatfilter gebruikt. De afstemming vindt plaats door decompositie van het originele beeld in subbeelden. Ieder subbeeld representeert een frequentieband uit het gehele frequentiespectrum van het originele beeld. De subbeelden ontstaan uit de Laplace piramide, die op de volgende manier wordt opgebouwd. De beeldmatrix van het originele beeld wordt geconvolueerd met een laagdoorlaatfilter. Dit laagdoorlaatfilter zorgt ervoor dat alle frequenties die minder dan circa vier keer per golflengte bemonsterd worden, onderdrukt worden. Het resultaatbeeld na laagdoorlaatfiltering wordt daarna herbemonsterd. Er ontstaat een subbeeldmatrix die een kwart kleiner is dan de oorspronkelijke beeldmatrix.

Herhalen van de filter en herbemonsteringsoperaties levert een serie beeldmatrices op: de Gauss piramide [19]. De originele beeldmatrix, het laagste niveau in de piramide heet $G_{m,n}^{(0)}$. Het niveau van de piramide wordt aangeduid met het superscript tussen haakjes. De filter operator heet $B^{(0)}$ en hoort tot de binomiaal filters. Het superscript tussen haakjes geeft aan het niveau van het beeld waarop de operator wordt toegepast. Het eerste niveau van de Gauss piramide ontstaat door toepassen van een herbemonsteringsoperator $R^{(0)}$ op de gefilterde beeldmatrix. De $R^{(0)}$ operator selecteert iedere tweede pixel in iedere tweede rij:

$$G_{m,n}^{(1)} = (BG)_{2m,2n}^{(0)} \text{ ook te schrijven als: } G_{m,n}^{(1)} = (RB)^{(0)} G_{m,n}^{(0)}. \quad (3.2.1)$$

Dezelfde operaties worden uitgevoerd met de nieuwe beeldmatrix $G_{m,n}^{(1)}$ en alle volgende niveaus van de piramide.

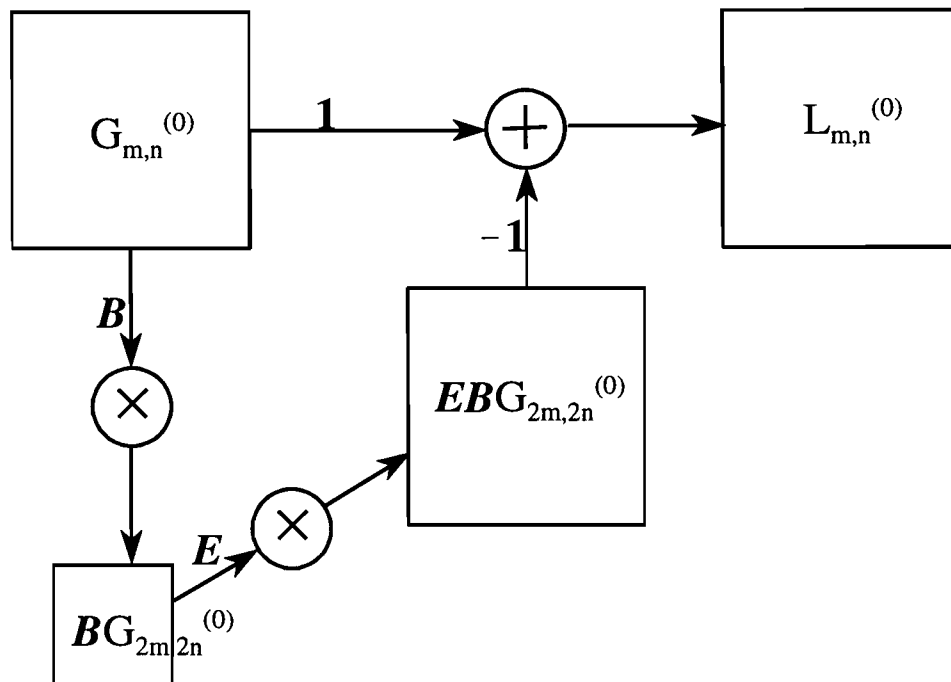
Figuur 3.2.1 laat zien hoe een bandgefilterd beeld ontstaat uit de Gausspiramide. De Laplace-matrix, gelijk aan het bandgefilterde beeld, van niveau (0) ontstaat als volgt:

$$L_{m,n}^{(0)} = G_{m,n}^{(0)} - E^{(1)}G_{m,n}^{(1)} \quad (3.2.2)$$

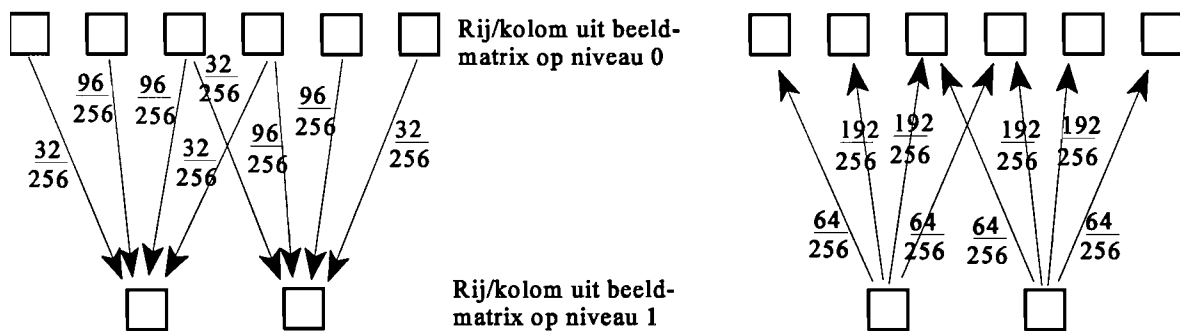
waarin E de expansie operator is toegepast op de beeldmatrix van niveau (1). E plaatst de beeldpunten van het subbeeld op de even pixelposities op de even rijen en interpoleert de ontbrekende andere punten.

In de praktijk wordt $L_{m,n}^{(0)}$ berekend zoals geschetst in figuur 3.2.2. Eerst worden alle kolommen van $G_{m,n}^{(0)}$ gewogen, zie ook het linker deel van figuur 3.2.2. Het resultaat wordt opgeslagen in een matrix met dimensie $\frac{1}{2}m, n$. Nu ondergaan de rijen van de resultaatmatrix dezelfde weging en het resultaat wordt opgeslagen in $G_{m,n}^{(1)}$. De weegfactoren bepalen de doorlaatband van het bandfilter. Hier is gekozen voor een binomiaal filter met $\sigma^2 = \frac{3}{4}$. B is gelijk aan $1/256[32 \ 96 \ 96 \ 32]$. De terugtransformatie van $G_{m,n}^{(1)}$ naar $G_{m,n}^{(0)}$ vindt ook plaats door eerst de kolommen van $G_{m,n}^{(1)}$ te wegen zoals geschetst in het rechter deel van figuur 3.2.2. Dan worden de rijen van het tussenresultaat gewogen met dezelfde weegfactoren waarna het resultaat wordt opgeslagen in een matrix met gelijke dimensies als $G_{m,n}^{(0)}$. Als laatste stap wordt het resultaat van de filtering/interpolatie afgetrokken van $G_{m,n}^{(0)}$, waarmee

$L_{m,n}^{(0)}$ ontstaat.



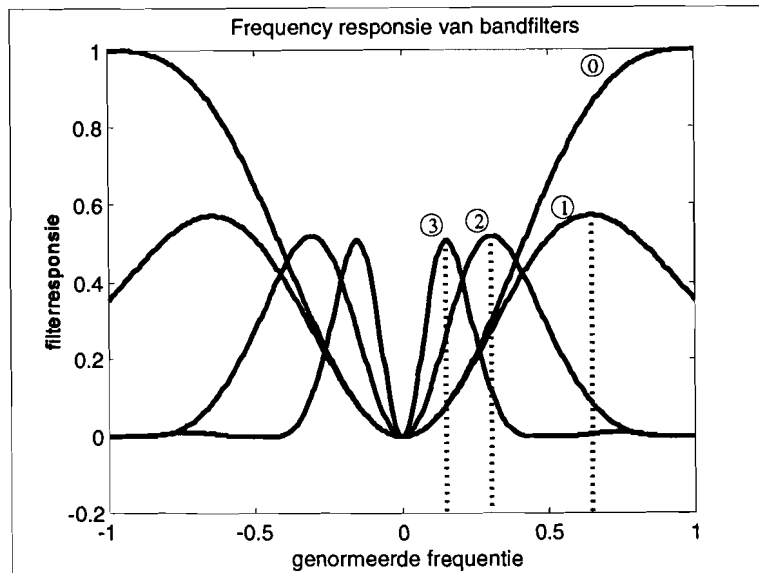
Figuur 3.2.1 Schematische weergave van de afleiding van een Laplacebeeld op niveau 0.



Figuur 3.2.2 Links de transformatie van een rij of kolom uit de beeldmatrix op niveau 0 naar een rij of kolom op niveau 1. De pixels uit niveau 0 ondergaan een laagdoorlaatbewerking. Rechts de transformatie (interpolatie) van een rij of kolom op niveau 1 naar niveau 0.

Het bepalen van de banddoorlaat van het Laplace filter gebeurt door het selecteren van een niveau uit de Laplace piramide. Per niveau wordt een frequentiegebied doorgelaten. Figuur 3.2.3 geeft voor het gekozen binomiaal filter de frequentieresponsie van een aantal niveaus uit de Laplace piramide. De centrumfrequentie van de doorlaatband neemt per oplopend niveau met een factor 2 af. Bovendien verzwakken de bandfilters van niveau 1,2 en 3 de grijswaar-

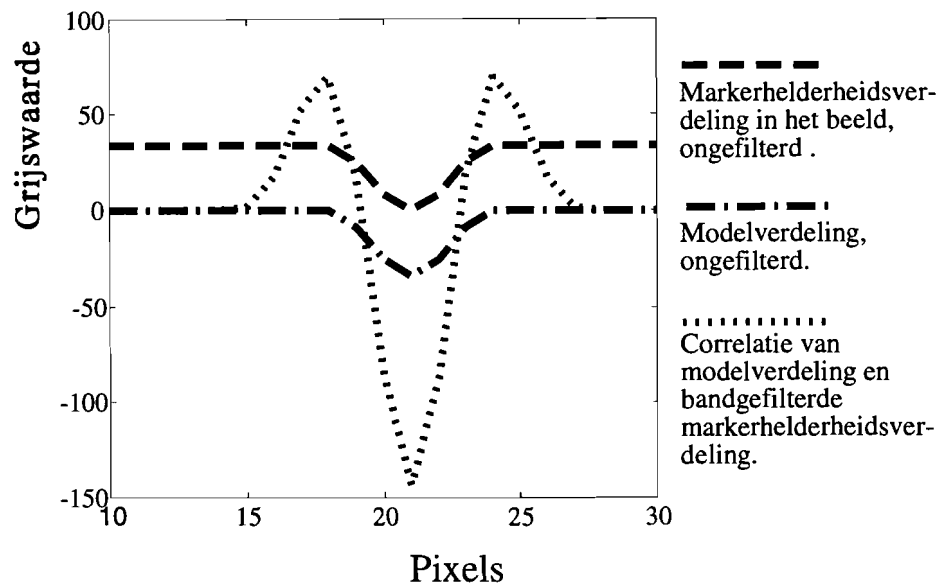
den met een factor van ongeveer 0.5 bij de centrumfrequenties. De markers hebben in de Röntgenbeelden een doorsnede van



Figuur 3.2.3 Frequentieresponsie van bandfilters afgeleid uit de Laplace piramide. 0 is de responsie van $L_{m,n}^{(0)}$, 1 van $L_{m,n}^{(1)}$, 2 van $L_{m,n}^{(2)}$ en 3 van $L_{m,n}^{(3)}$.

6-7 pixels. Per olopend niveau wordt de resolutie van het bandgefilterde beeld in pixels gehalveerd. Dit betekent dat de doorsnede van de markers in het Laplace-piramide-beeld op niveau 2 nog maar 1.5 - 1.75 pixels bedraagt. De markers zijn bij die doorsnede te klein om door het model uit hoofdstuk 2 beschreven te worden. Daarom wordt gekozen de markers te identificeren en te volgen in de Laplace beelden van niveau 1.

Het effect van de bandfiltering op de correlatiefunctie wordt weergegeven in figuur 3.2.4. De doorsnede van de marker in het Laplacebeeld op niveau 1 is $D_{\text{marker}} = 6$ pixels. Voor de eenvoud wordt de onversmeerde markerverdeling beschouwd in één dimensie.



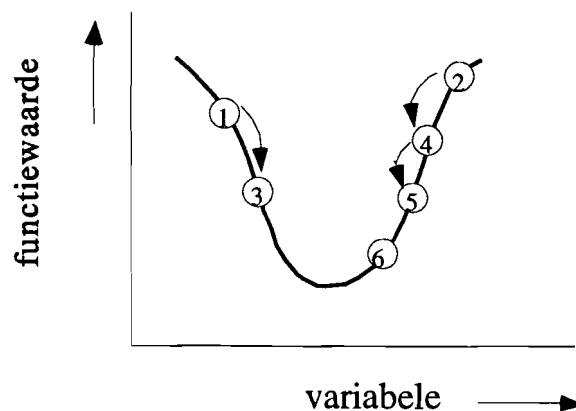
Figuur 3.2.4 Verloop van de correlatiefunctie bij filtering van de markerverdeling. De diameter van de marker $D_{\text{marker}} = 6$ pixels. De model- en markerverdeling zijn genomen in één dimensie. Het verloop van de correlatiefunctie is over een aantal pixels verschoven zodat het minimum samenvalt met de minima van marker- en modelverdeling.

3.3 Volgen van markers door optimalisatie van BIP's

Het identificeren en volgen van de markers (bepalen van de correspondentie) bestaat uit het optimaliseren van de posities van de BIP's. Het gaat hierbij om een optimalisatie in twee dimensies (de x- en y-coördinaten van de BIP's). De doelfunctie die geoptimaliseerd wordt is $C(\mathbf{P}_i)$, formule 3.1.1. Er bestaan verschillende optimalisatiemethoden; methoden die geen, de eerste, of de tweede afgeleide van de doelfunctie gebruiken, bijvoorbeeld de Downhill Simplex of Powells methode. De genoemde methoden optimaliseren de doelfunctie voor een set van zoekrichtingen. Het aantal zoekrichtingen is gelijk aan het aantal te optimaliseren parameters. In het algemeen worden de zoekrichtingen gekozen zo dat deze onderling loodrecht staan. Er is sprake van een conjugate set van zoekrichtingen wanneer bij het minimaliseren van de doelfunctie langs een zoekrichting, de minimale afgeleide van de doelfunctie langs de andere zoekrichting niet verandert. Het voordeel van een conjugate set is dat het minimum

van de doelfunctie in zo min mogelijk iteratiestappen, dus efficiënt, bereikt wordt.

De keus voor het optimalisatiealgoritme langs een zoekrichting is gevallen op de Golden Section methode [20]. Deze methode is eenvoudig en veilig. De enige eis die voor de te optimaliseren functie geldt is dat deze niet-singulier is. De Golden Section methode stelt geen eisen aan de differentieerbaarheid van de doelfunctie. De Golden Section methode heeft ook nadelen. De convergentie is lineair en hangt af van het initieel afbakeninterval. Bij een verkeerde keuze van het afbakeninterval kan de functiewaarde convergeren naar een lokaal in plaats van het globale minimum. Het principe achter de Golden Section methode is dat het afbakeninterval (hierbinnen ligt het minimum) op den duur bij iedere iteratiestap met een factor 0.618 verkleind wordt totdat het afbakeninterval voldoende klein is. Figuur 3.3.1 laat het successief afbakenen van het minimum zien. Het minimum wordt oorspronkelijk afgebakend door de punten 1,3,2. De doelfunctie wordt geëvalueerd bij 4 en vervangt 2; dan bij 5 waardoor 1 vervangen wordt; dan bij 6, waardoor 4 vervangen wordt. Na de getoonde stappen wordt het minimum afgebakend door de punten 3,6,5.



Figuur 3.3.1 Het successief afbakenen van het interval waarbinnen het minimum van een functie zit.

Nu volgt de beschrijving van twee optimalisatieprocessen. Het eerste berekent een initiële schatting in de buurt van de werkelijke posities van de BIP's in ieder beeld. Het tweede verfijnt de schattingen van de posities van de BIP's.

3.3.1 Initiële schattingen en extrapolatie van de BIP's

Het uitgangspunt bij het initieel schatten van de BIP's in beeld i is dat de posities van de BIP's in de twee voorafgaande beelden gegeven zijn. De posities van de BIP's in beeld i worden dan als volgt geschat; de hierna volgende beschrijving geldt voor alle BIP's in een beeld i .

Lineaire extrapolatie van de BIP's in de twee voorafgaande beelden, \mathbf{P}_{i-2} en \mathbf{P}_{i-1} , geeft \mathbf{P}_c in het beeld i . Zie ook het bovenste deel van figuur 3.3.1.1. De eerste zoekzichting wordt gekozen zo dat de werkelijke markerbaan dwars doorkruist wordt. Door \mathbf{P}_c wordt een cirkelbaan c gedefinieerd. Een punt \mathbf{P}_c op de cirkelbaan is gelijk aan:

$$\mathbf{P}_c(\phi) = \mathbf{P}_{i-1} + \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix} \cdot \begin{pmatrix} P_{i-1,x} - P_{i-2,x} \\ P_{i-1,y} - P_{i-2,y} \end{pmatrix} \quad (3.3.1.1)$$

waarbij $0 \leq \phi \leq 2\pi$.

Voordat de optimalisatie start moeten twee startposities aangegeven worden, \mathbf{P}_a en \mathbf{P}_b , die dicht in de buurt van het minimum van de doelfunctie 3.1.1 liggen. Anders bestaat de kans dat de doelfunctie naar een lokaal minimum convergeert. Voor deze startposities geldt:

$$\mathbf{P}_a = \mathbf{P}_c(\phi_0) \quad \text{en} \quad \mathbf{P}_b = \mathbf{P}_c(-\phi_0).$$

ϕ_0 wordt gekozen zo dat voldaan wordt aan:

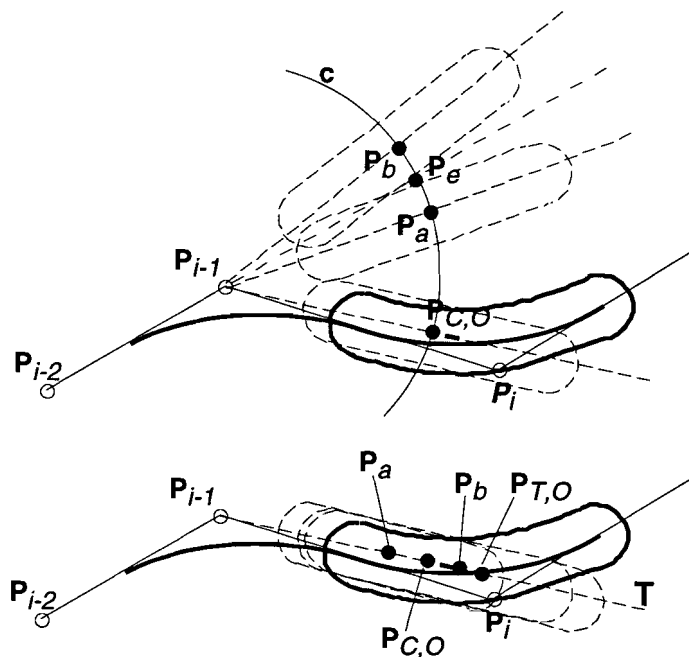
de afstand tussen \mathbf{P}_a en $\mathbf{P}_b = \frac{1}{2}D_{\text{marker}}$ indien de afstand tussen \mathbf{P}_{i-1} en $\mathbf{P}_{i-2} > \frac{1}{4}D_{\text{marker}}$

Wanneer de afstand tussen \mathbf{P}_{i-1} en $\mathbf{P}_{i-2} \leq \frac{1}{4}D_{\text{marker}}$ dan wordt $\phi_0 = \pi/4$ genomen,

De doelfunctie wordt geëvalueerd door in formule 3.1.1 $\mathbf{P}_i = \mathbf{P}_a$ en $\mathbf{P}_i = \mathbf{P}_b$ te nemen en vergeleken met een drempelwaarde C_{drempel} . De drempelwaarde bepaald of \mathbf{P}_a , \mathbf{P}_b , of beide voldoende dicht bij het minimum liggen. Uit het verloop van de correlatiefunctie in figuur 3.2.4 volgt de keuze van de drempel: $C_{\text{drempel}} = 0$. Zijn de waarden van de doelfunctie voor \mathbf{P}_a en $\mathbf{P}_b > C_{\text{drempel}}$, dan worden twee nieuwe punten $\mathbf{P}_{a, \text{nieuw}}$ en $\mathbf{P}_{b, \text{nieuw}}$ gekozen op de cirkelbaan \mathbf{P}_c waarvoor geldt: $\mathbf{P}_{a, \text{nieuw}} = \mathbf{P}_c(i \cdot \phi_0)$ en $\mathbf{P}_{b, \text{nieuw}} = \mathbf{P}_c(-i \cdot \phi_0)$ voor $i = [2, 3, 4, \dots]$, totdat voor zekere i de waarde van de doelfunctie voor $\mathbf{P}_{a, \text{nieuw}}$ of $\mathbf{P}_{b, \text{nieuw}} \leq C_{\text{drempel}}$. Veronderstel de

situatie $\mathbf{P}_{a, \text{nieuw}} = \mathbf{P}_c(i^* \phi_0)$ voor zekere i uit interval $[2, 3, 4, \dots]$, de waarde van de doelfunctie hierbij $\leq C_{\text{drempel}}$ en voor $\mathbf{P}_{b, \text{nieuw}} > C_{\text{drempel}}$. $\mathbf{P}_{b, \text{nieuw}}$ wordt nu gelijk genomen aan $\mathbf{P}_{b, \text{nieuw}} = \mathbf{P}_c((i-1) \cdot \phi_0)$.

$\phi_1 = i^* \phi_0$ en $\phi_2 = (i-1) \cdot \phi_0$ worden toegevoerd aan het Bracket algoritme uit Numerical Recipes. Hieruit volgen drie waarden voor ϕ , waarbij drie posities horen op de cirkelbaan c , die de minimale waarde van de doelfunctie insluiten. De drie waarden voor ϕ zijn de startposities voor de Golden section methode. Het resultaat van de Golden Section optimalisatie is de optimale waarde ϕ_{optimum} . Het optimale punt op de cirkel wordt daarmee $\mathbf{P}_{C,0} = \mathbf{P}_c(\phi_{\text{optimum}})$.



Figuur 3.3.1.1 Zoekrichtingen bij de initiële schatting. Boven: \mathbf{P}_a en \mathbf{P}_b , startposities voor de optimalisatie over de cirkel c , $\mathbf{P}_{C,0}$ het punt waarvoor de doelfunctie over de cirkel minimaal is. Onder: \mathbf{P}_a en \mathbf{P}_b , startposities voor de optimalisatie over lijn T , $\mathbf{P}_{T,0}$ het punt waarvoor de doelfunctie minimaal is over lijn T .

De tweede zoekrichting wordt gekozen in de bewegingsrichting van de marker, langs de lijn T , zie ook het onderste deel van figuur 3.3.1.1. Voor een willekeurig punt op de lijn T geldt:

$$\mathbf{P}_T(\lambda) = \mathbf{P}_{C,0} + \lambda \frac{(\mathbf{P}_{C,0} - \mathbf{P}_{i-1})}{\|\mathbf{P}_{C,0} - \mathbf{P}_{i-1}\|} \quad (3.3.1.2)$$

met $\lambda \in \mathbb{R}$ en de uitdrukking tussen $\|$ -tekens de lengte van $(\mathbf{P}_{C,0} - \mathbf{P}_{i-1})$.

Voor de punten P_a en P_b op de lijn T geldt: $P_a = P_T(-\frac{1}{4}D_{\text{marker}})$ en $P_b = P_T(\frac{1}{4}D_{\text{marker}})$. Ook hier wordt de doelfunctie berekend voor $P_i = P_a$ en $P_i = P_b$ en vergeleken met de drempelwaarde C_{drempel} . Zijn de waarden van de doelfunctie voor beide punten $> C_{\text{drempel}}$ dan worden twee nieuwe punten gekozen: $P_{a, \text{nieuw}} = P_T(-i * D_{\text{marker}})$ en $P_{b, \text{nieuw}} = P_T(i * D_m)$ met $i = [2, 3, 4, \dots]$ totdat voor één van beide punten de waarde van de doelfunctie $\leq C_{\text{drempel}}$, bijv. $P_{b, \text{nieuw}}$ voor zeker i . Dan wordt

$$P_{a, \text{nieuw}} = P_T((i-1) * D_{\text{marker}}).$$

Het Bracket algoritme berekent met $\lambda = -(i-1) * D_{\text{marker}}$ en $\lambda = i * D_{\text{marker}}$ drie waarden van λ die de minimale waarde van de doelfunctie insluiten. De Golden Section optimalisatie langs de lijn T waarbij de variabele λ wordt geoptimaliseerd geeft de optimale waarde λ_{optimum} , waaruit de positie van $P_{T,O} = P_T(\lambda_{\text{optimum}})$ volgt.

De optimalisatie langs over cirkel c en over lijn T wordt herhaald, waarbij nu de cirkel gedefinieerd wordt rond P_{i-1} en door $P_{T,O}$. Optimaliseren levert een nieuwe $P_{C,O, \text{nieuw}}$. De nieuwe lijn T loopt door P_{i-1} en $P_{C,O, \text{nieuw}}$. Optimaliseren over lijn T geeft $P_{T,O, \text{nieuw}}$. Is de afstand tussen $P_{T,O}$ en $P_{T,O, \text{nieuw}}$ groter dan een te kiezen tolerantie, dan word een nieuwe $P_{T,O, \text{nieuw}}$ berekend. Is de afstand tussen $P_{T,O}$ en $P_{T,O, \text{nieuw}}$ kleiner dan de tolerantie dan wordt de optimalisatie herhaald voor het BIP in de volgende beelden, zodat al extrapolierend een eerste schatting van de markerbaan verkregen wordt.

3.3.2 Schatting van de markerposities in de eerste twee beelden

De posities van de markers in de eerste twee beelden kunnen niet door het proces zoals beschreven in 3.3.1 geschat worden omdat van bekende posities van de BIP's uit de voorgaande beelden niet bekend zijn. Dit probleem is opgelost door de markers in de eerste twee beelden te identificeren met behulp van een neurale netwerk. Het neurale netwerk heeft als voordeel dat storende achtergrondobjecten, zoals de katheter in het Röntgenbeeld, sterk onderdrukt worden. Een neurale netwerk is een niet lineair filter waarmee markers exclusief gedetecteerd worden tegen een achtergrond van weefsel storingen en catheters. Als input van het neurale netwerk worden de grijswaarden van het Röntgenbeeld genomen. De output is een beeld waarin de pixels een waarde hebben tussen -1 en 1. Die waarde geeft een maat voor de

kans, dat een pixel behoort tot een markerpixel. Het hier toegepast neurale netwerk programma is ontwikkeld binnen de Universiteit Maastricht en bevat slechts één hidden layer.

Het neurale netwerk programma bestaat uit twee delen. Het eerste deel (LVCONV) berekent voor ieder input pixel een waarde tussen -1 en 1 en kent deze toe aan het overeenkomstige output pixel. De grijswaarden van het resultaatbeeld worden daarna nog geschaald tussen 0 en 255. Het aanbrengen van een drempel in het resultaatbeeld geeft een binair beeld waarin de kandidaat markerpixels grijswaarde 255 en de niet-markerpixels grijswaarde 0 toegekend krijgen. De drempelwaarde wordt bepaald met behulp van een beeldverwerkingsprogramma. Het resultaatbeeld wordt in het beeldverwerkingsprogramma ingelezen waarna een drempel wordt aangebracht. De drempelwaarde Thresh wordt met de hand gevarieerd totdat zoveel mogelijk werkelijke en zo min mogelijk vals positieve markerpixels overblijven.

Het binaire beeld wordt toegevoerd aan het tweede deel van het neurale netwerkprogramma (LVCLUST). Hierin worden eilandjes van kandidaat markerpixels geïsoleerd uit het binaire beeld. Twee parameters, het minimum aantal markerpixels en het maximum aantal markerpixels per eilandje, NPMI respectievelijk NPMA, bepalen de grootte van de eilandjes. De pixelwaarden van de eilandjes die groter zijn dan NPMA pixels of kleiner zijn dan NPMI pixels worden 0 gemaakt. LVCLUST berekent daarna voor ieder van de overgebleven eilandjes (de kandidaat markers) het zwaartepunt. De posities van de zwaartepunten zijn de centrumposities van de markers in het beeld. Ook het tweede beeld wordt verwerkt door LVCONV en LVCLUST. Ook hier levert dit een set op van kandidaat centrumposities van de markers.

Het correspondentieprobleem wordt als volgt opgelost. Voor iedere centrumpositie uit de set van het eerste beeld wordt de afstand berekend tot iedere centrumpositie van de set van het tweede beeld. Er is sprake van correspondentie tussen de centrumposities in het eerste beeld en het tweede beeld voor paren van centrumposities met de kleinste onderlinge afstand. De centrumposities van deze paren worden gedefinieerd als de BIP's in de eerste twee beelden. De twee beelden voor de opstart worden gekozen in een periode, dat het hart weinig beweegt. Dan is de detectie en het correspondentieprobleem relatief risicoloos.

3.3.3 Verfijning van schattingen door interpolatie

Het eerste optimalisatieproces beschreven in 3.3.1 is een baanextrapolatiemethode. Hierbij is het smeergebied steeds recht verondersteld door gebrek aan informatie (de positie van het BIP in het volgende beeld), zie figuur 3.3.1. Na het eerste optimalisatieproces is in ieder beeld een schatting van de positie van het werkelijk BIP bekend, $\mathbf{P}_{T,O}$. De schattingen vormen de startwaarden voor het tweede optimalisatieproces, dat tot verfijning leidt. Het proces verloopt als volgt.

In beeld i wordt de eerste zoekrichting van het verfijningsproces gedefinieerd loodrecht op de modelmarkerbaan door het punt $\mathbf{P}_{T,O}$ (de laatst berekende schatting uit het extrapolatieproces); de lijn N , zie ook het bovenste deel van figuur 3.3.3.1.

Een punt, \mathbf{P}_N , op lijn N wordt gedefinieerd door de vergelijking (3.3.3.1):

$$\mathbf{P}_N(\lambda) = \mathbf{P}_{T,O} + \lambda \frac{\mathbf{n}(0)}{\|\mathbf{n}(0)\|} \quad (3.3.3.1)$$

waarbij $\mathbf{n}(0)$ de normaalvector van de modelmarkerbaan op $\tau = 0$, $\lambda \in \mathbb{R}$ en de uitdrukking tussen $\|$ -tekens de lengte van $\mathbf{n}(0)$.

$\mathbf{n}(0)$ wordt bepaald uit formule 2.2.2 waarbij in $\mathbf{B}(\tau)$ (formule 2.1.1) moet worden ingevuld:

$$\tau = 0, \quad \mathbf{P}_{i-1} = \mathbf{P}_{i-1,T,O} \quad \mathbf{P}_i = \mathbf{P}_{T,O}, \quad \text{en} \quad \mathbf{P}_{i+1} = \mathbf{P}_{i+1,T,O}$$

$n_x(0) = -r_y(0)$ en $n_y(0) = r_x(0)$, met $n_x(0)$ en $n_y(0)$ de x - respectievelijk y - component van $\mathbf{n}(0)$

De startposities \mathbf{P}_a en \mathbf{P}_b worden gedefinieerd als $\mathbf{P}_N(\frac{1}{4}D_{\text{marker}})$ respectievelijk $\mathbf{P}_N(-\frac{1}{4}D_{\text{marker}})$.

De afstand tussen \mathbf{P}_a en $\mathbf{P}_b \leq \frac{1}{2}D_{\text{marker}}$ waardoor de kans dat het globale minimum wordt gemist gering is. Dezelfde doelfunctie wordt gebruikt als bij het extrapolatieproces. De

waarden van de doelfunctie voor $\mathbf{P}_i = \mathbf{P}_a$ en $\mathbf{P}_i = \mathbf{P}_b$ worden vergeleken met de drempelwaarde C_{drempel} . Zijn beide waarden $> C_{\text{drempel}}$ dan worden twee nieuwe punten gedefinieerd waarvoor

geldt:

$$\mathbf{P}_{a, \text{nieuw}} = \mathbf{P}_N(i * 1/4 D_{\text{marker}}) \text{ en } \mathbf{P}_{b, \text{nieuw}} = \mathbf{P}_N(-i * 1/4 D_{\text{marker}})$$

voor $i = [2, 3, 4, \dots]$, totdat voor zeker i de waarde van de doelfunctie voor $\mathbf{P}_i = \mathbf{P}_{a, \text{nieuw}} \leq C_{\text{drempel}}$ of $\mathbf{P}_i = \mathbf{P}_{b, \text{nieuw}} \leq C_{\text{drempel}}$. Het andere punt wordt dan gedefinieerd door $\mathbf{P}_N(\pm(i-1) * 1/4 D_{\text{marker}})$.

De twee waarden voor λ worden toegevoerd aan het Brent algoritme. Hieruit volgen drie startwaarden van λ voor de Golden Section optimalisatie. Het optimum λ_{optimum} definieert het punt $\mathbf{P}_{N,0} = \mathbf{P}_N(\lambda = \lambda_{\text{optimum}})$.

De tweede zoekrichting wordt genomen parallel aan de bewegingsrichting van de marker, zie het onderste deel van figuur 3.3.3.1. Voor \mathbf{P}_a en \mathbf{P}_b geldt:

$$\mathbf{P}_a = \mathbf{P}_{N,0} + \mathbf{V} \text{ en } \mathbf{P}_b = \mathbf{P}_{N,0} + \mathbf{W}$$

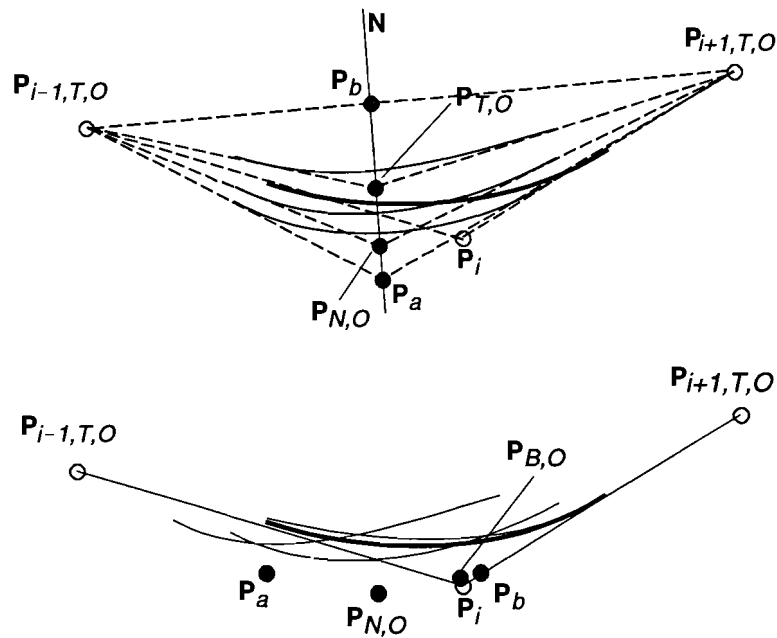
Voor \mathbf{V} en \mathbf{W} geldt:

$$\mathbf{V} = \mathbf{B}(-\tau_1) - \mathbf{B}(\tau_2) \text{ en } \mathbf{W} = \mathbf{B}(\tau_1) - \mathbf{B}(\tau_2) \quad \text{met } \mathbf{B} \text{ de formule 2.1.1, } \tau_1 = i * \tau_h, \tau_2 = 0, \\ i = [0, 1, 2, \dots]$$

Zodra voor zekere i geldt dat de waarde van de doelfunctie (3.1.1) voor bijvoorbeeld $\mathbf{P}_i = \mathbf{P}_a \leq C_{\text{drempel}}$, dan wordt \mathbf{P}_b gedefinieerd als

$$\mathbf{P}_b = \mathbf{P}_{N,0} + \mathbf{B}(\tau_1) - \mathbf{B}(\tau_2) \quad \text{voor } \tau_1 = -(i-1) * \tau_h \text{ en } \tau_2 = 0.$$

τ_1 en τ_2 worden toegevoerd aan het Bracket algoritme, dat drie startwaarden van τ berekent voor de Golden Section optimalisatie. Het optimum τ_{optimum} na de Golden Section optimalisatie definieert $\mathbf{P}_{B,0} = \mathbf{B}(\tau_{\text{optimum}})$.



Figuur 3.3.3.1. De zoekrichtingen bij de tweede optimalisatie. Boven: de eerste zoekrichting over lijn N loodrecht op de modelmarkerbaan voor $\tau = 0$ en $P_{T,O}$, P_a en P_b de startposities, $P_{N,O}$ het optimum na de Golden Section optimalisatie. Onder: zoekrichting parallel aan modelmarkerbaan rond $P_{N,O}$, P_a en P_b de startposities, $P_{B,O}$ het optimum na de Golden Section optimalisatie. Boven en onder: Vetgedrukte baan is de werkelijk markerbaan rond P_i .

De boven beschreven optimalisatieprocedure voor het vinden van een fijnere schatting voor het BIP in een beeld wordt een aantal malen herhaald totdat de afstand tussen de nieuwe $P_{B,O}$ en de vorige binnen een te kiezen tolerantie ligt. De keuze van de tolerantie bepaalt de nauwkeurigheid van de schatting. De positie van de laatst berekende $P_{B,O}$ wordt onthouden. Alle volgende beelden, behalve het laatste, ondergaan op dezelfde wijze de boven beschreven optimalisatieprocedure (zogenoemde RUN). Daarna volgt nog een RUN waarbij geldt voor ieder beeld:

$$P_{i-1,T,O} = P_{i-1,B,O} \quad \text{en} \quad P_{i+1,T,O} = P_{i+1,B,O}$$

Is de afstand tussen de $P_{B,O}$'s in overeenkomstige beelden uit de eerste en tweede RUN gelijk dan stopt het tweede optimalisatieproces. Is de afstand niet gelijk dan volgt er nog een RUN

waarbij geldt in ieder beeld:

het BIP in het vorige respectievelijk volgende beeld uit *huidige* RUN = geschatte BIP in het vorige respectievelijk volgende beeld uit de *vorige* RUN. De optimalisatie stopt wanneer in twee achtereenvolgende RUNS de *afstanden* tussen de $\mathbf{P}_{B,O}$'s in de overeenkomstige beelden *gelijk* blijven. Er is dan sprake van een stabiele situatie. De *laatst* berekende $\mathbf{P}_{B,O}$ in ieder beeld is de *uiteindelijke schatting* van het werkelijk BIP bij het interpolatieproces. De positie van de marker wordt gegeven overeenkomstig het model van de markerbaan waarbij \mathbf{P}_i 's gelijk zijn aan de $\mathbf{P}_{B,O}$'s.

4 Volgvermogen in gesimuleerde smeerbeelden

Het volgvermogen van de in hoofdstuk drie ontwikkelde volgmethode wordt onderzocht met behulp van gesimuleerde smeerbeelden. Onderzocht wordt het verschil tussen de posities van de werkelijke en geschatte BIP's bij variatie van de markerbaan, markersnelheid en toevoeging van Gaussische ruis.

4.1 Variatie van markerbaan en markersnelheid

In ieder beeld van de beeldsequentie wordt een BIP P_i gedefinieerd. In de projectie van alle beelden op één beeldvlak worden de posities van de BIP's ($P_{i,x}$, $P_{i,y}$) voor iedere markerbaan gedefinieerd door de volgende vergelijkingen:

$$\text{baan 1: } P_{i,x} = 1.12 * \cos (w*(i+3)*2\pi/NFRAMES) - 0.7$$

$$P_{i,y} = 1.12 * \sin (w*(i+3)*2\pi/NFRAMES) - 0.7$$

$$\text{baan 2: } P_{i,x} = 0.07 * \cos (w*(i+3)*2\pi/0.2*NFRAMES)$$

$$P_{i,y} = 0.07 * \sin (w*(i+3)*2\pi/0.2*NFRAMES)$$

$$\text{baan 3: } P_{i,x} = 0.07 * \cos (w*(i+3)*2\pi/1.5*NFRAMES)$$

$$P_{i,y} = 0.07 * \sin (w*(i+3)*2\pi/1.5*NFRAMES)$$

$$\text{baan 4: } P_{i,x} = 0.14 * \cos (w*(i+3)*2\pi/1.5*NFRAMES)$$

$$P_{i,y} = 0.14 * \sin (w*(i+3)*2\pi/1.5*NFRAMES)$$

$$\text{baan 5: } P_{i,x} = 0.5 * \cos (w*(i+3)*2\pi/8*NFRAMES) - 0.25$$

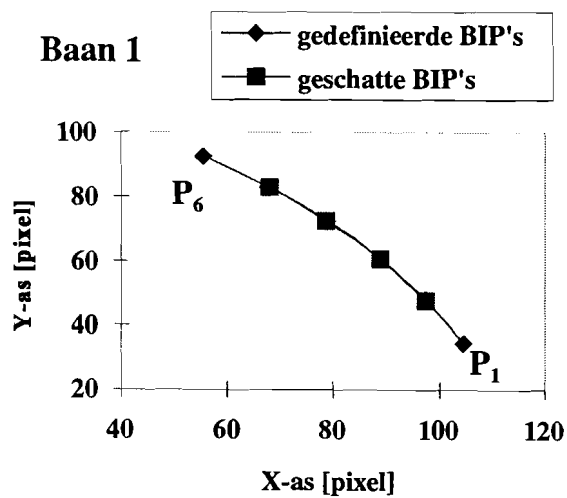
$$P_{i,y} = 0.5 * \sin (w*(i+3)*2\pi/8*NFRAMES) - 0.25$$

voor alle banen geldt: $i = [0, 1, 2, \dots, NFRAMES - 1]$, en $NFRAMES$ het aantal beelden in de sequentie, $w = 0.12$.

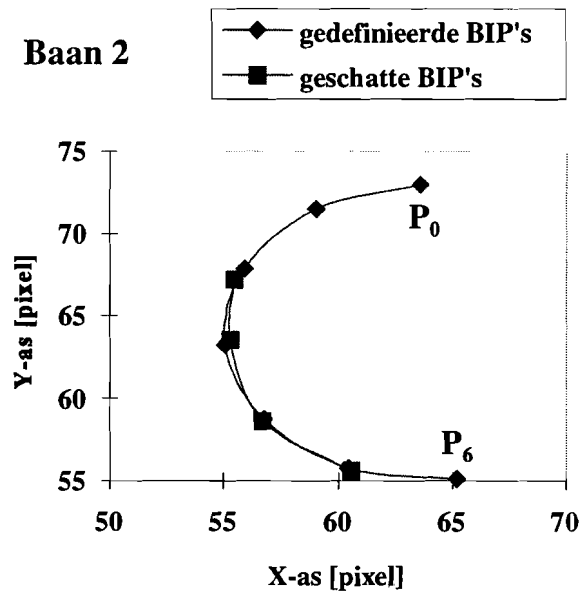
Bij iedere baan wordt de werkelijke markerbaan gedefinieerd zoals in paragraaf 2.1. De werkelijke smeergebieden worden gedefinieerd door voor ieder geïnterpoleerd smeergebiedpunt de drie dichtst bij zijnde pixels uit het beeld een grijswaarde toe te kennen gewogen naar de positie van het geïnterpoleerde smeergebiedpunt. De resterende pixels in het beeld krijgen grijswaarden 0.

De markerdiameter $D_{\text{marker}} = 3.5$. De beelden worden gefilterd met het bandfilter uit 3.2 en zijn 128×128 pixels groot. $\text{NFRAMES} = 7$.

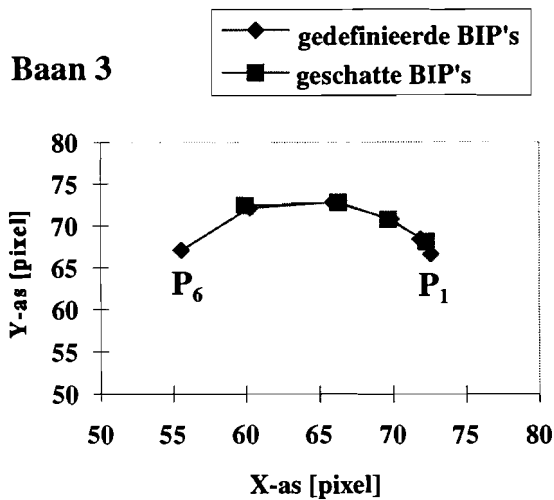
Voor de 5 genoemde banen worden de BIP's geschat door het volgalgoritme uit 3.3. In de figuren 12 t/m 16 zijn de gedefinieerde en geschatte BIP's weergegeven.



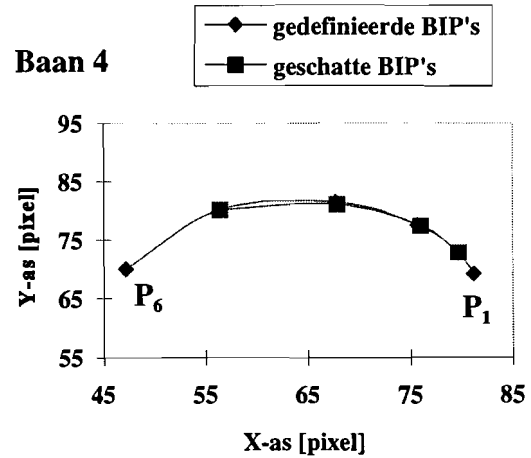
Figuur 4.1.1. Geschatte en gedefinieerde BIP's geprojecteerd op één beeldvlak.



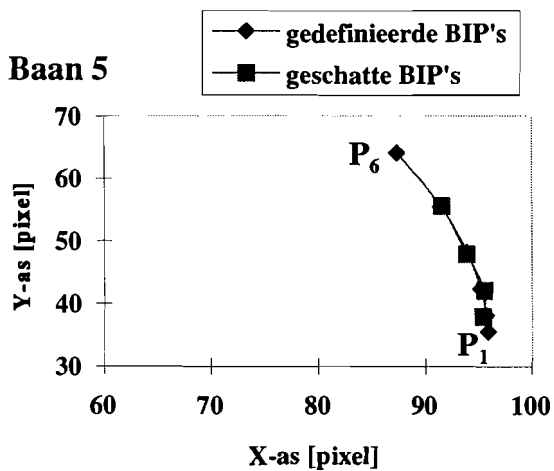
Figuur 4.1.2. Geschatte en gedefinieerde BIP's geprojecteerd op één beeldvlak.



Figuur 4.1.3. Geschatte en gedefinieerde BIP's geprojecteerd op één beeldvlak.



Figuur 4.1.4. Geschatte en gedefinieerde BIP's op één beeldvlak.



Figuur 4.1.5. Geschatte en gedefinieerde BIP's geprojecteerd op één beeldvlak.

In de figuren 4.1.1 t/m 4.1.5 zijn steeds de geschatte BIP's voor de beelden 3 t/m 6 afgebeeld. Het BIP in beeld 7 is niet geschat omdat daarvoor een schatting van het BIP in beeld 7 nodig is. De BIP's in beeld 1 en beeld 2 zijn als gegeven beschouwd en zijn daarom ook niet geschat. Het neurale netwerk algoritme voor de identificatie van de markers in de eerste twee beelden is niet toegepast omdat dat pas werd ontwikkeld.

Alle geschatte BIP's liggen op een afstand van < 0.5 pixels van de gedefinieerde BIP's. Uit de vergelijking van de schattingen tussen baan 1 en 2 blijkt dat de mate van kromming van de markerbaan weinig invloed heeft op het volgvermogen. De straal van baan 2 is gelijk aan de straal van baan 3. De markersnelheid is bij baan 2 lineair en bij baan 3 kwadratisch. Het blijkt dat de geschatte BIP's gemiddeld even ver van de gedefinieerde BIP's liggen bij lineaire en lineair versnelde beweging van de marker. Dat de kromming van de markerbaan geen invloed heeft op de nauwkeurigheid van de schattingen blijkt ook bij vergelijking van de resultaten van baan 3, 4 en 5, waarin telkens de kromming kleiner is.

4.2 Invloed van Gaussische ruis

De achtergrondstructuren in echte Röntgenbeelden zijn te beschouwen als ruis. Om de invloed van ruis op het volgvermogen te testen wordt aan de gesimuleerde smeerbeelden Gaussisch verdeelde ruis toegevoegd. De signaal ruisverhouding in de gesimuleerde verruiste smeerbeelden wordt gedefinieerd door formule 4.1.

$$SNR = \frac{S_A}{S_{noise}} \quad [dB] \quad (4.2.1)$$

SNR = gewenste signaalruisverhouding van het samengesteld beeld; gesimuleerd smeerbeeld + ruisbeeld.

S_A = Signaal Amplitude; gemiddelde som van grijswaarden van de markerpixels

S_{noise} = standaarddeviatie van het ruisbeeld.

Het ruisbeeld wordt gemaakt door iedere pixel in een beeld een grijswaarde toe te kennen afkomstig van een ruisgenerator met een Gaussische verdeling, waarvan de standaarddeviatie instelbaar is. De standaarddeviatie is gelijk aan de verhouding tussen S_A en de *gewenste* SNR. Het samengesteld beeld ontstaat door pixelsgewijs optellen van de grijswaarden van het ruisbeeld en het gesimuleerde smeerbeeld.

Voor de gesimuleerde smeerbeelden in de sequentie gelden dezelfde condities als in paragraaf 4.1 (D_{marker} , beeldgrootte en NFRAMES). In tabel 4.2.1 staan de afstanden tussen gedefinieerde en geschatte posities van de BIP's weergegeven voor baan 2, bij verschillende SNR. Ieder gesimuleerd smeerbeeld uit de beeldsequentie is vervangen door een samengesteld smeerbeeld. De samen-gestelde beelden worden gefilterd voordat het volgalgoritme start.

Tabel 4.2.1 Afstanden tussen gedefinieerde en geschatte BIP's bij verschillende SNR voor baan 2.

SNR van samen-gestelde beelden in [dB]	Afstand tussen geschatte en gedefinieerde BIP's in [pixels] na eerste / en na tweede optimalisatieproces. Baan 2.			
	Beeld 3	Beeld 4	Beeld 5	Beeld 6
30	0.54/0.30	0.57/0.41	0.39/0.12	0.54/0.10
20	0.54/0.30	0.57/0.41	0.39/0.13	0.56/0.09
18	0.54/0.30	0.57/0.41	0.39/0.39	0.61/0.07
15	0.56/0.29	0.61/0.45	0.38/0.38	0.63/0.13
12	0.55/0.31	0.57/0.44	0.40/0.32	0.64/0.06
9	0.55/0.51	0.68/1.12	0.85/1.03	0.68/0.47
6	0.56/0.48	0.85/1.03	1.30/0.98	1.02/0.78
3	0.63/1.51	3.07/2.99	7.25/1.23	18.13/ 19.87

Uit tabel 4.2.1 blijkt dat het interpolatieproces voor de meeste BIP's betere schattingen geeft. Wanneer de ruis toeneemt worden de schattingen naar verwachting slechter. Voor beelden met een SNR van 12 dB of hoger zijn de fouten (afstanden tussen de schattingen en gedefinieerde posities van de BIP's) < 0.5 pixel, ongeveer 15% van de markerdiameter. Wanneer de ruis varieert tussen 30 en 9 dB, verandert de fout in beeld 3 nauwelijks. Pas bij 3 dB SNR is de fout bijna een factor 3 groter dan bij 6 dB SNR. Ook in beeld 4 blijft de fout nagenoeg constant voor SNR tussen 30 en 12 dB. Dit is waarschijnlijk toe te schrijven aan het toegepaste bandfilter. Wordt de SNR te laag dan verliest het volgalgoritme het spoor zoals bij 3 dB

SNR in beeld 6. De fout in beeld 6 is voor alle SNR's kleiner dan de fouten in de andere beelden. Dit komt doordat bij het optimalisatieproces van beeld 6 steeds het gedefinieerde BIP uit het laatste beeld 7 wordt gebruikt in plaats van de schatting uit de vorige RUN. De baan in beeld 6 wordt daardoor beter geschat dan in de andere beelden.

De ontwikkelde volgmethode is in staat gesimuleerde smeerbeelden te volgen ook bij relatief slechte signaal ruisverhouding. Pas bij een zeer lage SNR divergeert het volgalgoritme van het markerspoor. De conclusie dat het ontwikkelde smeermodeel voldoet aan de werkelijkheid kan pas getrokken worden wanneer de volgmethode in echte videobeelden wordt getest, omdat de gesimuleerde smeer te zeer overeenkomstig het model is gedefinieerd.

5 Volgvermogen in videobeelden

De videobeelden waarop de volgmethode wordt getest zijn afkomstig uit de opstelling waarmee de Röntgenbeelden van het hondhart gemaakt worden. De videobeelden hebben het PAL formaat en bestaan uit twee halve beelden, geïnterlinieerd. De frame grabber digitaliseert deze beelden en kwantiseert het helderheidssignaal tot gehele getallen tussen 0 en 255. De digitale beelden hebben een afmeting van 768x576 pixels. Door de interlinieering worden de markerschaduwen in de digitale beelden extra versmeerd. De extra versmering wordt verwijderd door de gedigitaliseerde PAL beelden te spitsen in de twee halve beelden, die afmetingen hebben van 768x288 pixels. De halve beelden worden daarna herbemonsterd. De herbemonsterde beelden hebben afmeting 384x288 pixels. Van de herbemonsterde beelden worden een aantal randpixels verwijderd totdat beelden met afmeting 256x256 pixels overblijven. Deze vierkante beelden worden toegevoerd aan een programma waarin de volgmethode geïmplementeerd is. Het splitsen, herbemonsteren en vierkant maken van de beelden gebeurt met behulp van het beeldverwerkingspakket Khoros (draait onder het Operating System Linux).

In de vierkante beelden hebben de ongesmeerde markerschaduwen een diameter $D_{\text{marker}} = 6.5$ pixels. D_{marker} is bepaald door eerst de diameter in pixels te meten van de ring in de beelden.

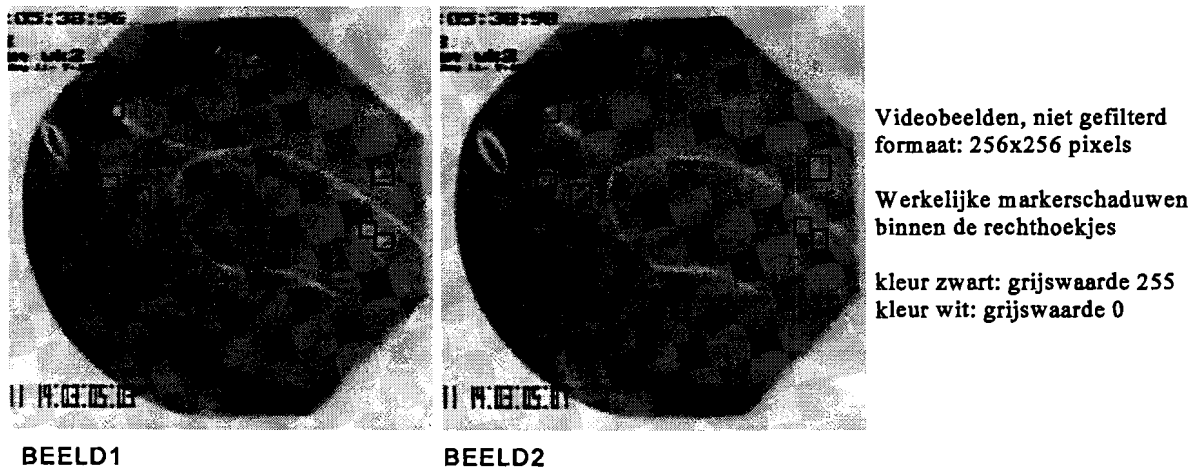
D_{marker} volgt dan uit de vergelijking

$(\text{diameter ring in pixels}) \cdot (\text{diameter marker in mm}) / (\text{diameter ring in mm})$.

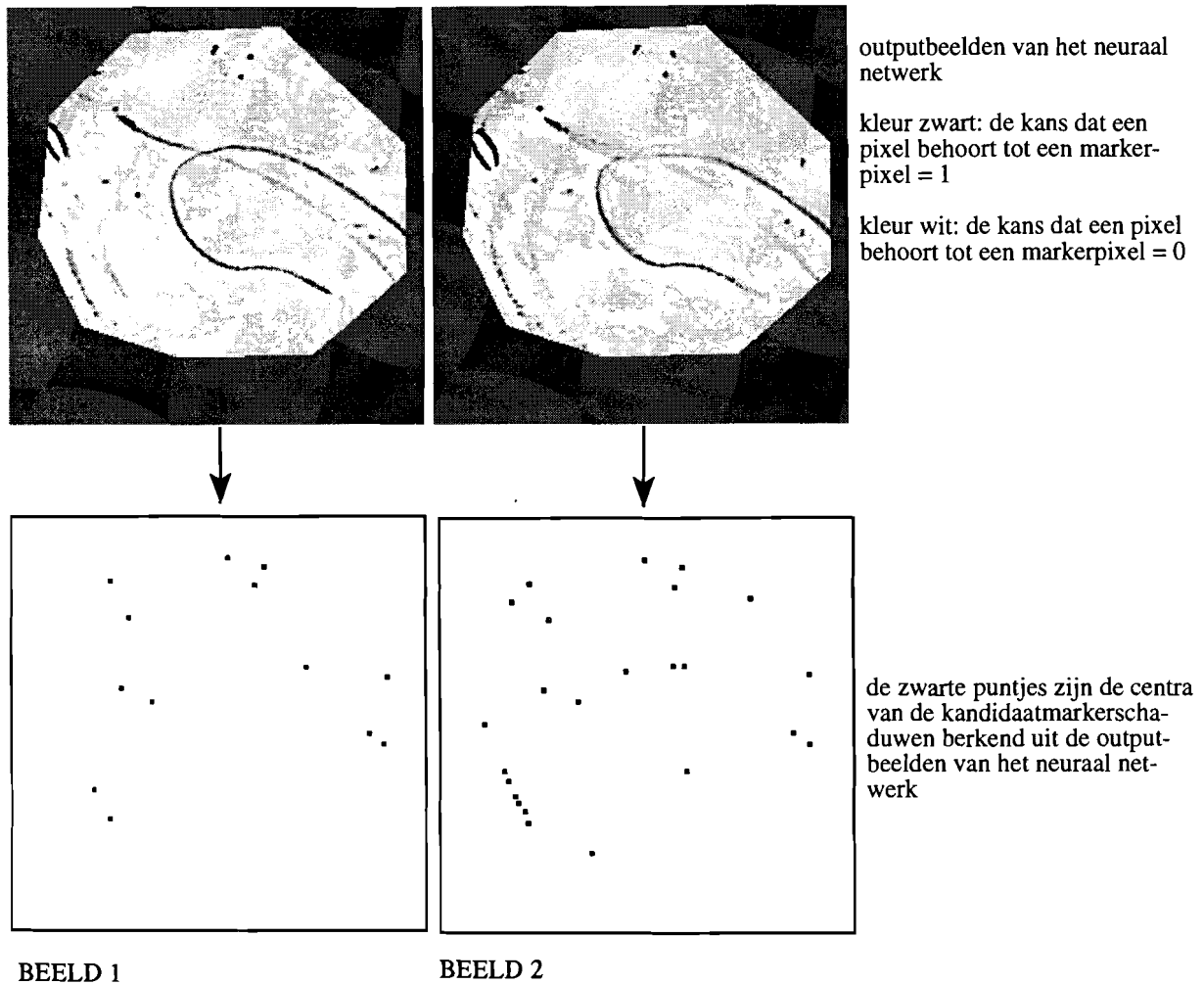
Na bandfiltering is de diameter van de ongesmeerde markerschaduwen in het beeld (128x128 pixels) $D_{\text{marker,f}} = 3.25$ pixels. Figuur 5.1 toont het eerste en tweede ongefilterde vierkante beeld.

De parameters voor het neurale netwerk worden bepaald zoals beschreven in paragraaf 3.3.2, in een beeld uit de rustige fase van de hartcyclus, waardoor de kans op smeer minimaal is. De parameterwaarden zijn $NPMA = 20$ pixels, $NPMI = 7$ pixels en $Thres = 40$ grijswaarde. Voor het eerste beeld uit een sequentie van 8 vindt LVCLUST met de gegeven parameterwaarden in

totaal 13 markerkandidaten, waaronder 9 werkelijk positieven en 4 vals positieven. In het tweede beeld vindt LVCLUST bij dezelfde parameterwaarden 24 markerkandidaten, waaronder de 9 werkelijk positieven.



Figuur 5.1 Videobeelden waarin de markers gedetecteerd en gevolgd worden.



Figuur 5.2 Boven: Het resultaat van filtering van de beelden uit figuur 5.1 met het neurale netwerk.

Onder: De centra van de kandidaat markerschaduwen berekend in de beelden na toepassing van het neurale netwerk.

De correspondentie tussen de kandidaten in het eerste en tweede beeld wordt berekend. De resulterende posities worden gebruikt als startposities voor het volgalgoritme. Het bovenste deel van figuur 5.2 toont de beelden gefilterd het neurale netwerk, het onderste deel geeft voor dezelfde beelden de centra van de kandidaatmarkerschaduwen.

Het volgalgoritme is gemodificeerd. Wanneer in het eerste optimalisatieproces het volgalgoritme in een beeld dreigt te divergeren van de werkelijke positie van het BIP dan wordt P_e als schatting voor het BIP genomen in plaats van het gedivergeerde punt $P_{T,O}$. De kans dat het volgalgoritme in het volgende beeld het spoor terug vindt neemt hierdoor toe. Divergeert het volgalgoritme op een marker in drie achtereenvolgende beelden, dan wordt de marker als

verloren of als vals positief beschouwd.

Het volgalgoritme heeft in ieder beeld uit de sequentie voor iedere markerkandidaat schattingen berekend van de posities van de BIP's. Deze schattingen zijn vertaald naar posities op de markerbanen voor $\tau = 0$ (het bemonstertijdstip) volgens formule 2.1. Het neurale netwerk is ook toegepast in de beelden 3 tot en met 8. De resultaten hiervan zijn gebruikt als referentie voor de schattingen berekend met de volgmethode.

Het volgalgoritme heeft 5 van de 9 werkelijke markers kunnen volgen. Het verschil tussen de posities berekend door het neurale netwerk en de geschatte posities van de met succes gevolgde markers varieert tussen 0.2 en 1.5 pixels. De markerschaduwen van twee paar markers in het eerste beeld liggen zo dicht bij elkaar dat het neurale netwerk maar twee van de vier markers als kandidaten aanmerkt. Dat geldt ook voor beeld 2 en 3. In beeld 4 en de volgende beelden zijn de 4 markers afzonderlijk zichtbaar. Het algoritme volgt slechts 2 van de 4 in de beelden 4 t/m 8. Twee andere markers waarbij het volgalgoritme ontspoord hebben in beeld 3 t/m 7 een laag contrast in vergelijking met de andere markers. In beeld 4 berekent het neurale netwerk voor een gesmeerde marker twee afzonderlijke posities. Een van de vier vals positieve kandidaten wordt herkend. De BIP's van de andere drie convergeren in alle beelden en de minimale correlatiewaarde is steeds van dezelfde grootte orde als de met succes gevolgde werkelijke markers. Samenvattend zijn van de 13 kandidaatmarkers 5 werkelijke markers met succes geïdentificeerd en gevolgd. 1 vals positieve is herkend. Bij 4 markers faalt het volgalgoritme, hetzij doordat het ontspoord hetzij doordat in het eerste beeld niet alle werkelijke markers worden herkend. 3 vals positieve kandidaten zijn door het volgalgoritme ten onrechte gevolgd.

Het ontwikkelde algoritme is in staat gesmeerde markerschaduwen in een reeks beelden te volgen onder de volgende beperkingen:

- ▶ De markerschaduwen moeten voldoende contrasteren met de achtergrond.
- ▶ Markersschaduwen wier posities in het eerste beeld samenvallen worden niet allemaal herkend en gevolgd.
- ▶ Indien vals positieve markers een correlatieminimum hebben van dezelfde orde grootte als de werkelijke markers kunnen ze niet op grond van de correlatiewaarde worden

herkend als niet-markers.

Met in acht neming van de genoemde beperkingen kan geconcludeerd worden dat het model van de markerbaan en de markersmeer overeenkomt met de werkelijkheid. Het verschil tussen de posities berekend door het neurale netwerk en de geschatte posities van de met succes gevolgde markers kan verklaard worden door de wijze waarop LVCLUST de centra van de eilandjes (markergebieden) berekent. LVCLUST gebruikt de binaire waarden van de eilandpixels. Hierdoor worden de randpixels even zwaar gewogen als de centripixels. Een nauwkeuriger berekening van het zwaartepunt wordt verkregen door een extra factor in de berekening mee te nemen, namelijk de grijswaarde van de pixels [24]. Een preciese aanduiding van de nauwkeurigheid waarmee het volgalgoritme de baanposities berekend is niet te geven omdat een nauwkeurige referentie ontbreekt.

6 Conclusies

De ontwikkelde methode voor het automatisch identificeren en volgen van door beweging versmeerde Röntgenmarkerschaduwen is toegepast in gesimuleerde en echte videobeelden. In de gesimuleerde smeerbeelden is het volgvermogen onderzocht voor markerbanen met verschillende kromming en markersnelheden. Zowel bij veranderende kromming als bij variërende markersnelheid liggen de gevonden modelmarkerbaansegmenten dicht (< 0.5 pixel, 15% van de markerdiameter) bij de gedefinieerde modelmarkerbaansegmenten. De methode is niet gevoelig voor de kromming van de markerbaan en de markersnelheid.

De ruisgevoeligheid op het volgvermogen van de methode is getest in gesimuleerde smeerbeelden waaraan Gaussische ruis is toegevoegd. De afstand tussen de gedefinieerde en gevonden markerbaansegmenten verandert nauwelijks (< 0.5 pixel) voor signaal-ruisverhoudingen van > 12 dB en neemt zichtbaar toe voor signaal-ruisverhoudingen < 12 dB. De methode ontspoot voor signaal-ruisverhouding van < 3 dB.

In de echte videobeelden blijkt dat de methode versmeerde markers kan volgen, met inachtneming van een aantal beperkingen met betrekking tot het contrast van de markers en het samenvallen van markerschaduwen in het eerste beeld. Bij te weinig contrast ontspoot de volgmethode. In het eerste beeld samenvallende markerschaduwen die in de volgende beelden afzonderlijk zichtbaar worden, kunnen niet afzonderlijk gevolgd worden.

Het ontwikkelde model van de markerbaan geeft de centrumposities van de (versmeerde) markers als functie van de tijd. Hierdoor is een nauwkeuriger bepaling van de markerpositie mogelijk in vergelijking met de huidige technieken, waarin de markerpositie alleen op de bemonstertijdstippen bekend is.

Appendix

A Handleiding bij het markervolprogramma

Het volgprogramma is ontwikkeld voor het volgen van door bewegingsonscherpte versmeerde Röntgenmarkers. Het programma berekent in een beeldsequentie van gedigitaliseerde Röntgenbeelden een reeks beeldcoördinaten, de centrumposities van de markers in ieder beeld en de BIP's (definiëren de baansegmenten van de markers).

Hieronder volgen stapsgewijs de handelingen die nodig zijn voor de juiste invoer van het volgprogramma. Uitgangspunt is een videoband met daarop de Röntgenbeelden.

Stap 1. Maak met een frame grabber een beeldsequentie in *.tif* formaat. Hiervoor kan bijvoorbeeld het Windows programma *Movie* worden gebruikt. *Movie* is ontwikkeld binnen de vakgroep Biofysica. De ingevangen beelden kunnen vanuit *Movie* direct in *.tif* formaat worden weggeschreven op de harde schijf of op een diskette. Ook bevat *Movie* een optie om de beelden te spitsen in halfbeelden. De halfbeelden kunnen binnen *Movie* niet vierkant gemaakt worden. Geef de beelden *DOS* namen. De (half)beelden zijn PAL beelden in formaat (384x288) 768x576 en zijn niet geschikt voor het volgprogramma. Ze moeten nog een *voorbewerking* ondergaan.

Stap 2. Laat een *user account* aanmaken (door de systeem beheerder) op een machine waar *LINUX OS versie 1.2.13* en *Khoros 1* draait. Log hierop in. Je zit dan automatisch in je home directory. Kopieer het bestand *.profile* van een user die reeds met *Khoros* werkt naar je home directory (voorbeeld: `cp ../michael/.profile ~`). Het '~'-teken geeft het pad naar je home directory aan. In de *.profile* staan alle omgevingsvariabelen gedefinieerd voor *Khoros*

evenals de padverwijzing naar de Khoros routines. Maak een sub directory waarin de .tif beelden uit stap 1 worden geplaatst (voorbeeld: `mkdir ~/marker`).

Kopieer de beelden van diskette naar de ~/marker directory (voorbeeld: `mcopy a:*.tif ~/marker`), of transporteer de beelden met FTP naar de LINUX machine (hiervoor moet de computer waarmee de beelden zijn gedigitaliseerd zijn aangesloten op het (lokale) netwerk en een FTP programma moet zijn geïnstalleerd). Kopieer ook het script bestand *splitimg* van diskette naar de ~/marker directory. Maak het script bestand executable (voorbeeld: `chmod a+x ~/marker/splitimg`). De .tif beelden worden gesplitst en getransformeerd in het juiste formaat door voor ieder beeld de script file te starten (voorbeeld: `splitimg seq10`). Let op! Laat de extensie .tif van ieder beeldbestand weg, splitimg gaat ervan uit dat de naam die op de commandoregel wordt meegegeven eindigt op .tif. Na afloop zijn er twee nieuwe bestanden aangemaakt met extensies a.byx en b.byx (voorbeeld: seq10a.byx en seq10b.byx). Dit zijn de gesplitste beelden die vierkante afmetingen in pixels hebben (256x256).

In detail voert het script bestand *splitimg* achtereenvolgens de hieronder genoemde Khoros routines uit:

- `tiff2viff -i <invoer bestand> -o <uitvoer bestand>`, zet de .tif beelden om in Khoros formaat beelden. Voorbeeld:
`tiff2viff -i seq0.tif -o seq2_0.xv`
- `vsplit -i1 <invoer bestand> -o1 <uitvoer halfbeeld 1> -o2 <uitvoer halfbeeld 2>`, splitst de beelden in twee halfbeelden met formaat 384x288. Dit programma is ontwikkeld binnen de vakgroep Biofysica en maakt geen deel uit van het pakket standaardfuncties van Khoros. Voorbeeld:
`vsplit -i1 seq2_0.xv -o1 seq2_0a.xv -o2 seq2_0b.xv`
- `vextract -i <invoer bestand> -o <uitvoer bestand> -x <integer> -y <integer> -w 256 -h 256`, extraheert een gedeelte uit een beeld, waarbij x,y de pixel coördinaten van de linker bovenhoek waar de extractie begint, w de breedte en h de hoogte van het te extraheren gedeelte. Tip: Bekijk de originele .tif beelden met een beeldverwerkingsprogramma en bepaal daarin de coördinaten van x,y. Halveer daarna die coördinaten en rond af op gehele getallen (vextract gebruikt beelden die de helft zo groot zijn als de originele

.tif beelden). Voorbeeld:

```
vextract -i seq2_0a.xv -o seq2_0aa.xv -x 60 -y 7 -w 256 -h 256
```

- `vconvert -i <invoer bestand> -o <uitvoer bestand> -t 'byte'`, converteert het dataformaat van de beelden naar bytes. Voorbeeld:

```
vconvert -i seq2_0aa.xv -o seq2_0aa.xvc -t 'byte'
```

- `viff2raw -i <invoer bestand> -o <uitvoer bestand>`, transformeert Khoros beelden naar byte beelden. Voorbeeld:

```
viff2raw -i seq2_0aa.xvc -o seq2_0a.by
```

Noot: Indien in stap 1 de beelden al zijn gesplitst binnen Movie wordt vsplit NIET uitgevoerd, de script file moet dan gemodificeerd worden.

Stap 3. Plaats de volgende bestanden in dezelfde directory als het volgprogramma:

<code>g0_0_0_</code>	bestand met de gewichten van het neuraal netwerk
<code>imagein.txt</code>	tekstbestand met bestandsnamen van de beelden uit de sequentie, op iedere regel één bestandsnaam
<code>netspec.run</code>	bestand waarin beschreven wordt welke structuur het neuraal netwerk heeft
<code>roi.by</code>	region of interest byte beeld waarmee het neuraal netwerk de randen van het beeld buiten beschouwing laat

Stap 4. Twee beelden met de kandidaatmarkers moeten gemaakt worden ten behoeve van de initialisatie van het volgprogramma. Dit gebeurt met behulp van een apart programma:

- Zet in het bestand `imagein.txt` de namen van de beelden uit de sequentie die gevolgd moet worden. Kies het *eerste* beeld uit de *rustige* fase van de hartcyclus, zodat de markerschaduwen niet of nauwelijks versmeerd zijn.
- Start `nnimage(.exe)`, dit programma maakt de beelden met de kandidaatmarkers.
- Geef aan het *aantal* beelden uit `imagein.txt`; parameter `NFrames`.
- `nnimage` maakt een aantal nieuwe beelden en vraagt onder welke *bestandsnamen* ze moeten worden weggeschreven; de naam van het *eerste* beeld met kandidaatmarkers, de naam van het *eerste* beeld met de centra van de eiland-

jes.

- nnimage vraagt om een drempelwaarde *Thresh*. Is dit de eerste keer dat nnimage draait, geef dan de default waarde 40 in, anders een waarde bepaald in stap vijf.
- nnimage vraagt om twee parameterwaarden in te geven; NPMA en NPMI, het maximale respectievelijk het minimale aantal pixels waaruit de kandidaat markereilandjes bestaan. Is dit de eerste keer dat nnimage draait geef dan NPMA=15 en NPMI=4.
- nnimage maakt een aantal nieuwe beelden en vraagt onder welke *bestandsnamen* ze moeten worden weggeschreven; de naam van het *tweede* beeld met kandidaatmarkers, de naam van het *tweede* beeld met de centra van de eilandjes.

Stap 5. Enkele invoerparameters voor het neurale netwerk moeten bepaald worden. Hiervoor worden de eerste twee beelden gedrempeld:

- Start een *beeldverwerkingsprogramma* op, bijvoorbeeld NIH Image. Lees het *eerste en tweede beeld* met de *kandidaatmarkers* (zoals bepaald in de vorige stap).
- Breng een *drempel* aan in het eerste beeld en varieer de drempel totdat zoveel mogelijk markers zichtbaar zijn.
- Leg in het tweede beeld dezelfde drempel als in beeld 1 aan en controleer of dezelfde markers zichtbaar zijn als in beeld 1. Varieer de drempel indien nodig om in beide beelden het maximale aantal markers zichtbaar te maken en zo min mogelijk achtergrond. Onthoud de drempelwaarde.

Nu moeten NPMA en NPMI nog bepaald worden. Tel in de gedrempelde beelden het aantal pixels van de grootste (NPMA) en kleinste (NPMI) markereilandjes en onthoud de waarden.

- *Doorloop* nogmaals *stap 4* en geef voor *Thresh* de zojuist gevonden drempelwaarde, en ook de waarden voor NPMA en NPMI.
- Lees nu het eerste beeld in met de centra van de kandidaatmarkers in NIH en controleer of de centra samenvallen met de centra van de smeergebieden uit het eerste beeld met kandidaatmarkers.

- Zijn er *teveel of te weinig* kandidaten, *wijzig* dan de *drempelwaarde*, *NPMA*, of *NPMI* en doe de twee hierboven beschreven substappen nogmaals.
- Onthoud de laatst ingestelde drempelwaarde, *Thresh*.

Stap 6. Start het volgprogramma *realtrak(.exe)* en geef het *aantal beelden uit de sequentie*, de *drempelwaarde Thresh*, *NPMA* en *NPMI* uit stap 5 aan. Let op: het aantal beelden moet overeenkomen met het aantal bestandsnamen in *imagein.txt*. Geef ook aan de bestandsnaam die de uitvoer data gaat bevatten (de beeldcoördinaten van de BIP's en de centra van de markers). Het volgproces verloopt daarna geheel automatisch.

B Source code

Hieronder de lijst met source code, header en andere noodzakelijke bestanden die nodig zijn voor het compileren van *realtrak(.exe)*:

<i>clust.c</i>	berekent de markereilandjes		
<i>conv.c</i>	berekent de convolutie van het beeld met het neuraal netwerk		
<i>imlib.c</i>	library met beeldroutines		
<i>matrxlib.c</i>	library met matrixroutines		
<i>network.c</i>	definieert het object met de gewichten		
<i>tr_set.c</i>	definieert de structuur van de invoer van het neuraal netwerk		
<i>realtrak.c</i>	markervolgprogramma		
<i>clust.h</i>	<i>conv.h</i>	<i>network.h</i>	<i>tr_set.h</i>
<i>imlib.h</i>	<i>matrxlib.h</i>	<i>realtrak.h</i>	<i>terminal.h</i>
<i>g0_0_0_</i>	<i>netspec.run</i>	<i>roi.by</i>	<i>imagein.txt</i>

Het programma *nnimage(.exe)* is gecompileerd met de volgende sources:

<i>nnimage.c</i>	berekent de eerste twee beelden met kandidaatmarkers		
<i>clust.c</i>	<i>conv.c</i>	<i>imlib.c</i>	<i>matrixlib.c</i>

<i>network.c</i>	<i>tr_set.c</i>		
<i>clust.h</i>	<i>conv.h</i>	<i>defs.h</i>	<i>imlib.h</i>
<i>matrixlib.h</i>	<i>network.h</i>	<i>realtrak.h</i>	<i>terminal.h</i>
<i>tr_set.h</i>			
<i>g0_0_0_</i>	<i>netspec.run</i>	<i>roi.by</i>	<i>imagein.txt</i>

Het programma waarmee het volgvermogen in gesimuleerde beelden is getest (noise(.exe)) is gecompileerd met de volgende sources:

<i>noise.c</i>	<i>noiseimg.c</i>	<i>noisemat.c</i>
<i>noise.h</i>	<i>noiseimg.h</i>	<i>noisemat.h</i>

C Compileren van de source code

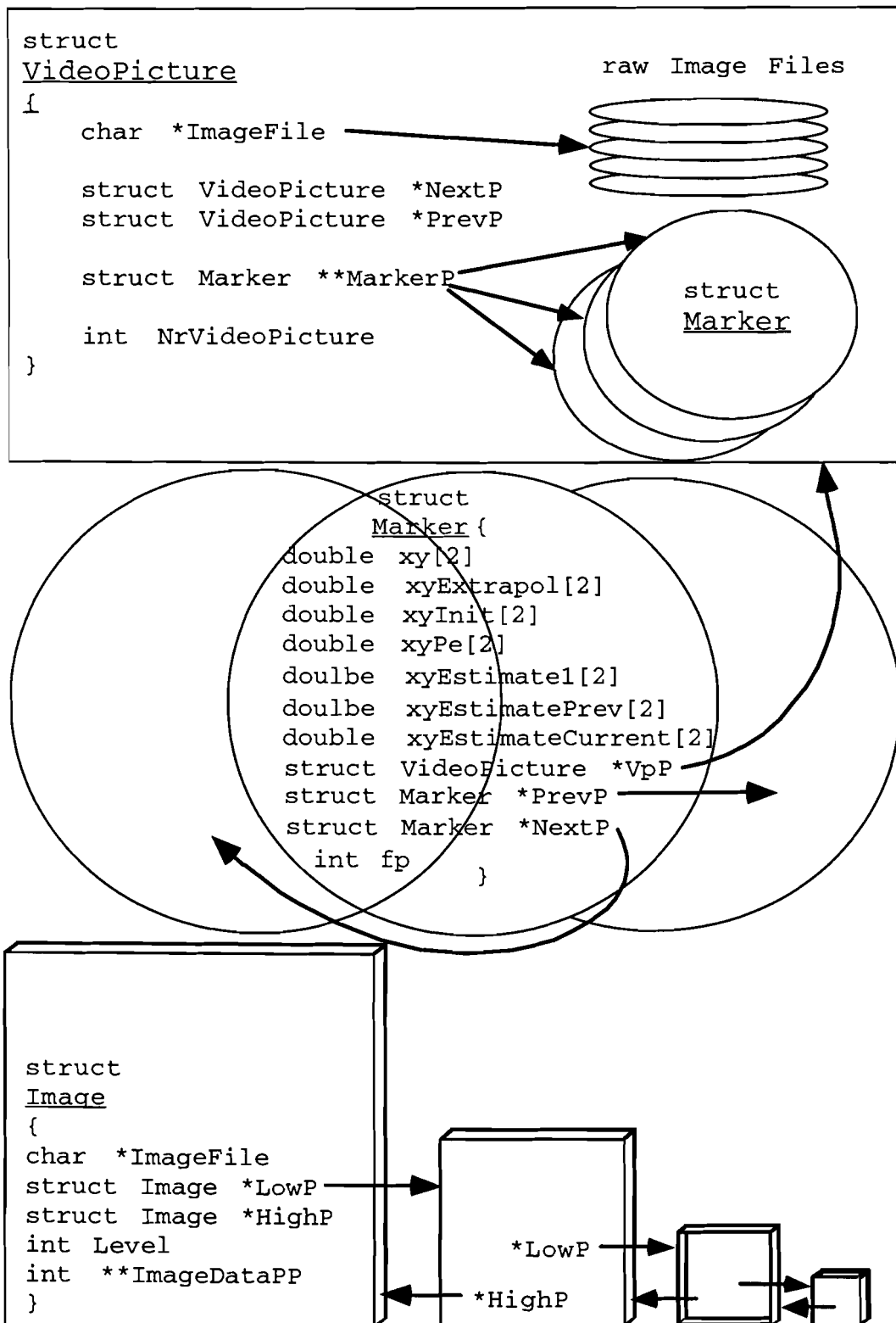
De sources zijn gecompileerd op een APPLE Power Macintosh 7200/90. De compiler is Codewarior 1.3. Een nieuw project wordt als volgt aangemaakt:

- Start Codewarior 1.3
- Klik in de menubalk achtereenvolgens op File, New Project...
- Selecteer de directory voor het project of maak een nieuwe directory aan
- Selecteer onder Project Stationary *~ANSI PPC C/C++.*
- Vul een naam in voor het project
- Klik op SAVE
- Klik in de menubalk achtereenvolgens op Project, Add Files...
- Ga naar de gewenste directory of de floppy en selecteer de SOURCE CODE FILES (eindigen meestal op .C) door ze aan het project toe te voegen (klik op Add)
- Klik op Done wanneer alle .C files aan het project zijn toegevoegd
- Klik in de menubalk achtereenvolgens op Edit, Preferences...
- Klik in de linker balk op het icoon C/C++ Language

- Selecteer bij Source Model *ANSI C*
- Klik in de linker balk op *C/C++ Warnings* icoon
- Selecteer alle items onder *Warnings Info* behalve de eerste (treat all warnings as errors)
- Klik in de linker balk op *PPC Project* icoon
- Vul in naast *Preferred Heap Size 4000*, *Minimum Heap Size 4000* en *Stack Size 256*
- Klik op *OK*
- Klik in het project window onder *Sources* op *replace me.ansi c*
- Klik in de menubalk achtereenvolgens op *Project, Remove Files*.

De source code bestanden kunnen gewijzigd worden door ze in het project-window dubbel aan te klikken. Het project wordt in zijn geheel gecompileerd door in de menubalk op het *make icoon* te klikken. Het *run icoon* start de gecompileerde code.

D Datastructuur Multi-Marker Tracking



```

struct VideoPicture
{
    struct Image *ImageFile    pointer naar een bestandsnaam op harde schijf
    struct VideoPicture *NextP pointer naar het volgende VideoPicture
    struct VideoPicture *PrevP pointer naar het vorige VideoPicture
    struct Marker **MarkerP    array van pointers naar de markers in het huidige beeld
    int NrVideoPicture         volgnummer van VideoPicture = Bemonstertijdstip
}

```

```

struct Marker
{
    struct Marker *NextP      pointer naar volgende Marker in huidig beeld
    struct Marker *PrevP     pointer naar vorige Marker in huidig beeld
    struct VideoPicture *VpP pointer naar VideoPicture
    double xy[2]             array met x, y coördinaat van BIP
    double xyExtrapol[2]     array met x,y coördinaat van BIP
    double xyEstimate1[2]    array met x,y coördinaat van BIP
    double xyEstimatePrev[2] array met x,y coördinaat van BIP
    double xyEstimateCurrent[2] array met x,y coördinaat van BIP
    double xyInit[2]        array met x,y coördinaat van BIP
    double xyPe[2]          array met x,y coördinaat van BIP
    int fp                  false positive indicatie van BIP, fp=0 : true positive;
                           fp=(1,2,3): Marker wordt in 1,2,3 beelden niet ge-
                           vonden
}

```

```

struct Image
{
    char *ImageFile          pointer naar een bestandsnaam met beelddata op harde
                           schijf
}

```

```
struct Image *LowP      pointer naar subbeeld, 1 niveau lager
struct Image *HighP    pointer naar subbeeld, 1 niveau hoger
int Level              niveau van beeld
int **ImageDataPP      dubbele pointer naar beelddata in geheugen (2D array
                       met grijswaarden)
}
```

Lijst van begrippen, symbolen, subscripts en superscripts

Begrippen

Begrip	Definitie/Omschrijving
Baansegment	Een deel van de geprojecteerde markerbaan rondom een bemonster-tijdstip.
Beeld	Een beeld is de 2-D(imensionale) weergave van de 3-D werkelijkheid. Wiskundig wordt een beeld gedefinieerd met een beeldfunc-tie g . Deze geeft weer de helderheid als functie van de plaats in het beeldvlak: $g(x, y)$
Beeldmatrix	Een beeldmatrix ontstaat door afbeelding van de beeldfunctie in \mathbb{R}^2 naar een matrix met een eindig aantal elementen: $g(x, y) \Rightarrow G_{m,n}$ $m, n \in [0, 1, 2, \dots]$ indices van de elementen van de beeldmatrix Het afbeeldingsproces heet digitalisatie. Op een discrete set punten (een 2-D grid van $m \times n$ punten) in het beeldvlak wordt $g(x, y)$ be-monsterd. Bovendien wordt de grootte van de helderheden ge-kwantiseerd tussen 0 en 255.
Beeldpunt	Een beeldpunt \mathbf{P} is een punt in het beeldvlak met componenten P_x

en P_y , de x - respectievelijk de y -coördinaat in het beeldvlak.

Beeldsequentie	Een beeldsequentie ontstaat door de werkelijkheid op achtereenvolgende tijdstippen te bemonsteren. Op ieder bemonstertijdstip ontstaat één beeld.
Beeldvlak	Een beeldvlak is een twee dimensionale ruimte in \mathbb{R}^2 . Hierin wordt een x - y assenstelsel gedefinieerd waarvoor geldt dat: $\{x \in \mathbb{R} \mid 0 \leq x \leq N_{\max} - 1\}$ en $\{y \in \mathbb{R} \mid 0 \leq y \leq M_{\max} - 1\}$, N_{\max}, M_{\max} is het aantal pixels in de x - respectievelijk de y -richting. De oorsprong van het x - y assenstelsel valt samen met de linker bovenhoek van het beeldvlak en $(N_{\max} - 1, M_{\max} - 1)$ valt samen met de rechter benedenhoek van het beeldvlak.
Bemonstertijd	De tijdsduur tussen twee achtereenvolgende bemonstertijdstippen: T_B . Bij videobeelden is $T_B = 20$ ms.
Bemonstertijdstip	Het tijdstip waarop het beeld bemonsterd wordt: $t_{B,i} = i * T_B$, met $i = [0, 1, 2, \dots]$
BIP	Baaninterpolatiepunt, punt in het beeldvlak waarlangs het baansegment van een marker wordt gedefinieerd
Grijswaarde	Helderheidswaarde, $\in \mathbb{R}$
Marker	De marker is een object in het beeld. De helderheidsfunctie van de marker is een deelfunctie van de beeldfunctie: $m(k, l) \in g(x, y) \mid k, l \in \mathbb{R}^2$ De grijswaardenmatrix van de marker is een deelmatrix van de beeldmatrix: $M_{q,r} \in G_{m,n} \mid q, r \in [0, 1, 2, \dots]$ indices van de markergrijswaarden-

	matrix
Markermidpunt	Het markermidpunt is het centrum van een marker.
NPML,NPMA	Het minimum respectievelijk maximum aantal pixels van de kandidaat markereilandjes.
NFRAMES	Het aantal beelden in een beeldsequentie.
Pixel	Een punt op de 2-D grid heet een pixel en representeert de discrete grijswaarde (uit het interval $[0, \dots, 255]$) op de corresponderende gridpositie volgens de normale notatie voor matrices. De eerste index, m , geeft de positie van de rij aan, de tweede, n , de positie van de kolom. Als het beeld wordt gerepresenteerd door een $M \times N$ matrix, dan loopt de index n van 0 tot $N_{\max} - 1$, de index m van 0 tot $M_{\max} - 1$. M_{\max} geeft het aantal rijen weer, N_{\max} het aantal kolommen.
SNR	Gewenste signaalruisverhouding van het samengesteld beeld; gesimuleerd smeerbeeld + ruisbeeld.
Sluittijd	De tijdsduur waarbij de (video)camera zijn sluitersluiter geopend heeft wordt gedefinieerd als de sluitertijd: T_s . De sluitertijd ligt symmetrisch rond het bemonstertijdstip.
Thresh	Drempelwaarde in een grijswaardenbeeld.

Symbolen

Symbool	Eenheid	Betekenis
\mathbf{B}	–	model van de geprojecteerde markerbaan (formule 2.1.1)
$\mathbf{B}^{(l)}$	–	binomiale filteroperator op $\mathbf{G}_{m,n}^{(l)}$
B_x	pixel	x-coördinaat van \mathbf{B}
B_y	pixel	y-coördinaat van \mathbf{B}
C	grijswaarde	doelfunctie (formule 3.1)
C_{th}	grijswaarde	drempelwaarde van C
$C_{optimum}$	grijswaarde	optimale waarde van C
c	–	cirkel
D_{marker}	pixel	markerstraal
$D_{marker,f}$	pixel	markerstraal in bandgefilterd beeld
$\mathbf{E}^{(l)}$	–	expansieoperator toegepast op $\mathbf{G}_{m,n}^{(l)}$
g	–	continue helderheidsfunctie van een beeld
$\mathbf{G}_{m,n}$	–	grijswaardenmatrix van het beeld
$\mathbf{G}_{m,n}^{(l)}$	–	grijswaardenmatrix van het beeld op niveau l , (formule 3.2.1)
\mathbf{I}_{baan}	–	smeergebiedpunt op de baan
$\mathbf{I}_{baan,x}$	pixel	x-coördinaat van \mathbf{I}_{baan}
$\mathbf{I}_{baan,y}$	pixel	y-coördinaat van \mathbf{I}_{baan}
\mathbf{I}_{buiten}	–	smeergebiedpunt buiten de baan
$\mathbf{I}_{buiten,x}$	pixel	x-coördinaat van \mathbf{I}_{buiten}
$\mathbf{I}_{buiten,y}$	pixel	y-coördinaat van \mathbf{I}_{buiten}
\mathbf{I}_{voor}	–	smeergebiedpunt aan voorkant van de baan
$\mathbf{I}_{voor,x}$	pixel	x-coördinaat van \mathbf{I}_{voor}
$\mathbf{I}_{voor,y}$	pixel	y-coördinaat van \mathbf{I}_{voor}
\mathbf{I}_{achter}	–	smeergebiedpunt aan achterkant van de baan
$\mathbf{I}_{achter,x}$	pixel	x-coördinaat van \mathbf{I}_{achter}

$I_{\text{achter},y}$	pixel	y-coördinaat van I_{achter}
i	–	index, $i = [0,1,2,\dots]$
L	pixel	lengte van het markerspoor in het beeld (formule 2.2.2)
$L_{m,n}^{(1)}$	–	Laplacebeeld op niveau 1, (formule 3.2.2)
M_{max}	–	aantal pixels in de x-richting van het beeldvlak
$M_{q,r}$	–	grijswaardenmatrix van de marker
N_{max}	–	aantal pixels in de y-richting van het beeldvlak
n	–	lijn
\mathbf{n}	–	normaalvector aan de gemodelleerde markerbaan
n_x	pixel	x-coördinaat van \mathbf{n}
n_y	pixel	y-coördinaat van \mathbf{n}
\mathbf{P}	–	beeldpunt in het beeldvlak
P_x	pixel	x-coördinaat van \mathbf{P}
P_y	pixel	y-coördinaat van \mathbf{P}
\mathbf{P}_a	–	beeldpunt op $\tau=0$
\mathbf{P}_b	–	beeldpunt op $\tau=0$
$\mathbf{P}_{B,0}$	–	optimale schatting van BIP over baan in verfijnings interpolatie
\mathbf{P}_C	–	beeldpunt op cirkel C (formule 3.3.1.1)
$\mathbf{P}_{C,0}$	–	optimale schatting van BIP over cirkel in initiële schatting
\mathbf{P}_e	–	beeldpunt na extrapolatie van \mathbf{P}_{i-1} en \mathbf{P}_{i-2}
\mathbf{P}_{i+2}	–	BIP in beeld $i+2$ op $\tau=2$
\mathbf{P}_{i+1}	–	BIP in beeld $i+1$ op $\tau=1$
\mathbf{P}_i	–	BIP in beeld i op $\tau=0$
\mathbf{P}_{i-1}	–	BIP in beeld $i-1$ op $\tau=-1$
\mathbf{P}_{i-2}	–	BIP in beeld $i-2$ op $\tau=-2$
\mathbf{P}_N	–	beeldpunt op lijn N (formule 3.3.3.1)
$\mathbf{P}_{N,0}$	–	optimale schatting van BIP over lijn n in verfijnings interpolatie
\mathbf{P}_T	–	beeldpunt op lijn T (formule 3.3.1.2)
$\mathbf{P}_{T,0}$	–	optimale schatting van BIP over lijn t in initiële schatting
$\underline{R}^{(1)}$	–	herbemonsteringsoperator toegepast op $G_{m,n}^{(1)}$

Lijst van begrippen, symbolen, subscripts en superscripts

\mathbf{r}	-	raakvector aan de gemodelleerde markerbaan (formule 2.2.3)
r_x	pixel	x-coördinaat van \mathbf{r}
r_y	pixel	y-coördinaat van \mathbf{r}
r_{marker}	pixel	markerstraal van markerschaduw in het beeld
\mathbb{R}	-	één dimensionale ruimte van alle reële getallen
\mathbb{R}^2	-	twee dimensionale ruimte van alle reële getallen
S_A	grijswaarde	gemiddelde som van de grijswaarden van de markerpixels in $G_{m,n}$
S_n	grijswaarde	standaarddeviatie van het ruisbeeld
$S_{q,r}$	-	geïnterpoleerde grijswaardenmatrix van het smeergebied model
s	pixel	lengte van recht lijnstuk; benadering voor L
T_B	sec	bemonstertijdsinterval
T_S	sec	sluittijdsinterval
t	sec	de tijd [$t=(i+\tau)*T_B$]
$t_{B,i}$	sec	bemonstertijdstip [$t_{B,i} = i*T_B$]
$v_{\text{marker,gem}}$	pixel	gemiddelde snelheid van de marker over het spoor in het beeld
\mathbf{V}	-	vector
V_x	pixel	x-coördinaat van \mathbf{V}
V_y	pixel	y-coördinaat van \mathbf{V}
\mathbf{W}	-	vector
W_x	pixel	x-coördinaat van \mathbf{W}
W_y	pixel	y-coördinaat van \mathbf{W}
x	pixel	beeldcoördinaat in het beeldvlak, $\{x \in \mathbb{R} \mid 0 \leq x \leq N-1\}$
y	pixel	beeldcoördinaat in het beeldvlak, $\{y \in \mathbb{R} \mid 0 \leq y \leq M-1\}$
λ	-	lopende variabele, $\{\lambda \in \mathbb{R}\}$
λ_{optimum}	-	optimale waarde van λ
τ	-	genormaliseerde tijd [$\tau = (t - t_{B,i})/T_B$]
τ_h	-	hulptijdstapinterval
τ_{optimum}	-	optimale waarde voor τ

ϕ	rad	hoek tussen $(\mathbf{P}_e - \mathbf{P}_{i-1})$ en $(\mathbf{P}_C - \mathbf{P}_{i-1})$
ϕ_0	rad	startwaarde van ϕ
ϕ_{optimum}	rad	optimale waarde van ϕ

Subscripts

Symbol	Betekenis
0	startwaarde
A	amplitude
a	$\tau=0$ in beeld i
b	$\tau=0$ in beeld i
baan	op de baan
e	geëxtrapolleerd
B	bemonster
C	cirkel in beeld i
gem	gemiddeld
i	index van beeldnummer in sequentie [0,1,2,...]
j	index van smeergebiedpunt $[-r_{\text{marker}}, \dots, -2, -1, 1, 2, \dots, r_{\text{marker}}]$
m	index van de rijen in een matrix [0,1,2,...]
n	index van kolommen in een matrix [0,1,2,...]
noise	ruis
N	lijn in beeld i
optimum	optimale waarde
q	index van de rijen in een matrix [0,1,2,...]
r	index van kolommen in een matrix [0,1,2,...]
S	sluiter
T	lijn in beeld i

Superscripts

Symbool	Betekenis
----------------	------------------

l	index van het niveau van het beeld uit de Gausspiramide $l = [0, 1, \dots]$
-----	---

Literatuurlijst

- [1] Davis, Peter L. and Gilbert L. Raff, Stanton A. Glantz
A METHOD TO IDENTIFY IMPLANTED RADIOPAQUE MARKERS DESPITE ROTATION OF THE HEART.
American Journal of Physiology, vol. 239, 1980, p. H573-H580.
- [2] Philips, Charles M. and John Prenis, William P. Santamore, Alfred A. Bove
RECOGNITION AND STORAGE OF METAL HEART MARKER POSITION FROM BIPLANE X-RAY IMAGES AT VIDEO RATES.
IEEE Transactions on Biomedical Engineering, vol. BME-30, no. 1, January 1983, p. 10-17.
- [3] Hannaford, Blake and Stanton A. Glantz
ADAPTIVE LINEAR PREDICTOR TRACKS IMPLANTED RADIOPAQUE MARKERS.
IEEE Transactions on Biomedical Engineering, Vol. BME-32 (1985), no. 2, februari, p. 117-125.
- [4] Macerata, Alberto and William J. Sanders, Edwin L. Alderman M.D., D. Craig Miller M.D.
AN AUTOMATED VIDEO IMAGE ANALYSIS FOR DETECTION AND TRACKING IMPLANTED MYOCARDIAL MARKERS.
IEEE Proceedings, Computers in Cardiology, 1988, p.547-550.

- [5] Niczyporuk, Marek A. and D.Craig Miller
AUTOMATED TRACKING AND DIGITIZATION OF MULTIPLE RADIO-
PAQUE MYOCARDIAL MARKERS.
Computers and Biomedical Research, 24 (1991), p. 129-142.
- [6] Muijtjens, Arno M.M. and Jef M.A. Roos, Theo Arts, Arie Hasman
EXTRAPOLATION OF INCOMPLETE MARKER TRACKS BY LOWER
RANK APPROXIMATION.
Int. J. Biomed. Comput., 33 (1993), p 219-239.
- [7] Zhong-ying, Zhang and Yuan Bao-zong
MULTI RESOLUTION TARGET DETECTION AND TRACKING THROUGH
A PARALLEL COARSE-TO-FINE SEARCH APPROACH.
In: Proceedings TENCON '93. 1993 IEEE Region 10 Conference on 'Computer,
Communication, Control and Power Engineering', cat. no. 93CH3286-2, Beijing,
China, 19-21 Oct. 1993. Ed. By Yuan Baozong, IEEE, New York, 1993, vol 2, p.
1198-1202.
- [8] Celenk, Mehmet and Hasnain Reza
MOVING OBJECT TRACKING USING LOCAL WINDOWS.
In: Third International Symposium on Intelligent Control 1988, Arlington, VA,
USA, 24-26 Aug 1988.
IEEE, IEEE Service Center, Piscataway, NJ, USA, cat n 882012, 1988, p. 180-
185.
- [9] Kass, Michael and Andrew Witkin, Demetri Terzopoulos
SNAKES: ACTIVE CONTOUR MODELS.
International Journal of Computer Vision, 1988, p. 321-331.

- [10] Hwang, Junghyn and Yoshiteru Ooi, Shinji Ozawa
A MOTION/SHAPE ESTIMATION OF MULTIPLE OBJECTS USING AN
ADVANCED CONTOUR MATCHING TECHNIQUE.
IEICE Transs. Inf. & Syst., vol. E77-D, No. 6 june 1994, p. 676-685.
- [11] Samy, Roger and Jean-François Bonnet
ROBUST AND INCREMENTAL ACTIVE CONTOUR MODELS FOR OB-
JECT TRACKING.
In: Intelligent Robots and Computer Vision XIII: 3D Vision, Product Inspection,
and Active Vision, Proceedings of SPIE- The International Society for Optical
Engineering, v. 2354, 1994, p. 50-56.
- [12] Ueda, Naonori and Kenji Mase, Yasuhito Suenaga
A CONTOUR TRACKING METHOD USING AN ELASTIC CONTOUR
MODEL AND AN ENERGY-MINIMIZATION APPROACH.
Systems and Computers in Japan, Vol. 24, No. 8, 1993, p. 59-69.
- [13] Lobregt, Steven and Max A. Viergever
A DISCRETE DYNAMIC CONTOUR MODEL.
IEEE Transactions on Medical Imaging, vol. 14, No. 1, March, 1995, p. 12-24.
- [14] Vieren, Christophe and François Cabestaing, Jack-Gérard Postaire
CATCHING MOVING OBJECTS WITH SNAKES FOR MOTION TRACKING.
Pattern Recognition Letters, v. 16 n 7 July, 1995, p. 679-685.
- [15] Reed, Irving S. And Robert M. Gagliardi, Larry B. Stotts

A RECURSIVE MOVING-TARGET-INDICATION ALGORITHM FOR OPTICAL IMAGE SEQUENCES.

IEEE Transactions on Aerospace and Electronic Systems, vol. 26, no. 3, May 1990, p. 434-439.

- [16] Wang, Gan and Rafael M. Inigo
DETECTION AND TRACKING OF SINGLE-PIXEL TARGETS BASED ON TRAJECTORY CONTINUITY.
Image and Vision Computing, Volume 11, No. 10, december 1993, p. 641-655.
- [17] Meyer, François G. And Patrick Bouthemy
REGION-BASED TRACKING USING AFFINE MOTION MODELS IN LONG IMAGE SEQUENCES.
CVGIP: Image Understanding v 60, n 2, Sept 1994, p. 119-140.
- [18] Yao, Yi-Sheng and Rama Chellappa
TRACKING A DYNAMIC SET OF FEATURE POINTS.
IEEE Transactions on Image processing, vol. 4, no. 10, October 1995, p. 1382-1395.
- [19] Jähne, Bernd.
DIGITAL IMAGE PROCESSING, CONCEPTS, ALGORITHMS, AND SCIENTIFIC APPLICATIONS, second edition.
Springer-Verlag, Berlin Heidelberg, 1993, paragraaf 8.2, p. 174-184.
- [20] Press, William H, and Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling

NUMERICAL RECIPIES.

Cambridge University Press, Cambridge, third edition 1988, hoofdstuk 10, p.

277-282.