

Semi-automatische Video Segmentatie

Jonas Buntinx

promotor :
Prof. dr. Philippe BEKAERT

Semi-automatische video segmentatie met behulp van keyframe-based tracking

Thesis voorgedragen tot het behalen van de graad van
master in de informatica/ICT/kennistechnologie

Jonas Buntinx

Promotor: Prof. dr. Philippe Bekaert

Begeleider: Cedric Vanaken

2006 - 2007

Samenvatting

Computer visie is het subdomein binnen informatica dat afbeeldingen interpreteert zodat de computer de scène begrijpt.

Meestal is niet alle informatie binnen de foto's (of video sequentie) even belangrijk. Vaak zijn slechts delen van de foto nodig. Daarom wordt segmentatie gebruikt om de interessante delen uit de foto te halen. Segmentatie in computer visie verdeelt een beeld in verschillende objecten. Hierdoor kunnen er individuele operaties op aparte objecten uitgevoerd worden. Bij video segmentatie is dit het segmenteren van de beelden uit een video sequentie. Omdat het apart segmenteren van elk beeld uit de video een dure operatie is, zijn er speciale technieken bedacht voor het segmenteren van video.

In deze thesis wordt een overzicht gegeven van verschillende beeld en video segmentatie technieken. Eén video segmentatie techniek wordt in detail uitgelegd, segmentatie met behulp van keyframe gebaseerde tracking. Tracking is de techniek om objecten in een scène te volgen. Keyframe gebaseerd wil zeggen dat met behulp van een beperkt aantal frames de overige berekend worden. Deze techniek is een semi-automatische video segmentatie techniek, omdat de gebruiker instaat voor de initialisatie en de mogelijkheid heeft om te corrigeren.

Segmentatie met behulp van keyframe gebaseerde tracking maakt gebruik van een energie functie minimalisatie voor het volgen van objecten doorheen de video. Een energie functie berekend de kost van de tracking, en deze kost dient dan geminimaliseerd te worden. De verschillende termen van de energie functie worden in deze thesis uitgelegd en er wordt een algoritme gegeven voor het minimaliseren van de functie: Powell's dogleg methode, die gebruik maakt van een trust region.

Inhoudsopgave

1	Introductie	4
2	Beeld segmentatie	9
2.1	Regiogebaseerde Technieken	10
2.1.1	Partitionele clustering methode: K-Means	12
2.1.2	Divisieve hiërarchische clustering methode: Histogram clustering .	12
2.1.3	Agglomeratieve hiërarchische clustering methode: Region-growing methode	13
2.1.4	Model-gebaseerde methode	14
2.2	Randgebaseerde Segmentatie	15
2.2.1	Lokale Filtering	15
2.2.2	Active Contours	16
2.2.3	Intelligent Scissors	17
2.3	Graaf methode	17
2.3.1	Graph Cut Segmentatie	17
2.4	Multi-scale methode	18
2.5	Conclusie	19
3	Video segmentatie	20
3.1	Background subtraction	21
3.1.1	Uitbreiding van background subtraction: Moving average	21
3.1.2	Uitbreiding van background subtraction: Clustering	22
3.2	Regiogebaseerde methode	23
3.3	Video Cutout	23
3.4	Randgebaseerde methode: Rotoscoping	24
3.4.1	Lineaire interpolatie	25
3.4.2	Motion capturing	26
3.4.3	Kalman tracking	26
3.4.4	Keyframe-gebaseerde tracking	27
3.5	Conclusie	28
4	De energie functie voor Keyframe gebaseerde tracking	29
4.1	Termen van de energie functie	30
4.1.1	Lengte term	30
4.1.2	Curvatie term	31
4.1.3	Snelheids term	31
4.1.4	Image term	31

4.1.5	Edge term	32
4.2	Conclusie	33
5	Optimalisatie	34
5.1	Niet lineaire kleinste kwadraten probleem	35
5.2	Optimalisatie	35
5.3	Gradient descent	36
5.3.1	Problemen	37
5.4	Newton methode	37
5.5	Gauss-Newton methode	38
5.5.1	De inverse van de Hessiaan	39
5.6	Levenberg methode	39
5.7	Levenberg-Marquardt methode	40
5.8	Trust-region methode	41
5.8.1	Voorbeeld van trust-region	44
5.8.2	Trust-region subprobleem	48
5.8.3	Approximatie methode	50
5.9	Conclusie	52
6	Implementatie	53
6.1	Keyframes	53
6.1.1	Eerste keyframe	53
6.1.2	Tweede keyframe	54
6.2	Object structuur	55
6.2.1	Joints	55
6.2.2	Spline	55
6.3	Powell's dogleg optimalistie methode	55
7	Resultaten	59
8	Uitbreidingen	67
9	Conclusie	69

Hoofdstuk 1

Introductie

Sinds de oudheid heeft de mens steeds met beelden gewerkt, van rotswandtekeningen over schilderijen, foto's tot film en video. Wij begrijpen zonder te verpinken wat er op staat en meestal ook wat er mee bedoeld is. Het Chinees spreekwoord “Eén beeld zegt meer dan duizend woorden” is geen fabel. Onze hersenen zijn in staat alles wat in een beeld staat te onderscheiden en te herkennen.

Het grootste probleem bij beeldherkenning is te weten te komen, welke pixels behoren tot een onderdeel (object) en welke niet. In figuur 1.1 bijvoorbeeld is het voor personen makkelijk de dalmatiër te herkennen uit wat lijkt een verzameling van zwarte vlekjes. Hoewel het menselijk brein de objecten kan herkennen is het toch moeilijk te vertellen of een bepaalde pixel tot de dalmatiër behoort of niet.



Figuur 1.1: Een persoon kan makkelijk de vorm van de dalmatiër herkennen uit de figuur. Echter het is onmogelijk op een gestructureerde manier te bepalen, welke punten bij de hond horen en welke bij de achtergrond. [15]

Door de zwart-wit foto zijn veel eigenschappen van het beeld verloren gegaan en toch kunnen onze hersenen de hond herkennen, omdat we ofwel reeds weten wat we zoeken

ofwel omdat we zoeken met bepaalde modellen (we weten namelijk dat we, door de vraagstelling, moeten zoeken naar een object met zwarte en witte delen, vb. konijn, koe, zebra, dalmatiër, e.d.). Onze hersenen kunnen hier een onderscheid tussen de verschillende vlekjes op basis van eerder verworven kennis.



Figuur 1.2: Omdat er meer informatie is in het beeld is het eenvoudig om de dalmatiër te onderscheiden.

Wanneer we van een andere representatie van het beeld (figuur 1.2) vertrekken, zullen we vlugger de hond herkennen omdat we een onderscheid kunnen tussen de achtergrond en de hond. De hersenen hebben dankzij de kleur in het beeld meer informatie om een onderscheid te kunnen maken tussen de verschillende objecten. In figuur 1.2 maken onze hersenen een onderscheid tussen de objecten op basis van kleur.

De techniek die wordt gebruikt voor het onderscheiden van verschillende objecten wordt segmentatie genoemd.

Wij gebruiken de term segmentatie voor een brede waaier van technieken. Hoewel deze technieken kunnen verschillen, sterven ze hetzelfde doel na, namelijk het verkrijgen een compacte informatie van wat in het beeld nuttig of interessant is. Naarmate welke informatie men wil verkrijgen kunnen bepaalde technieken betere resultaten opleveren dan andere.

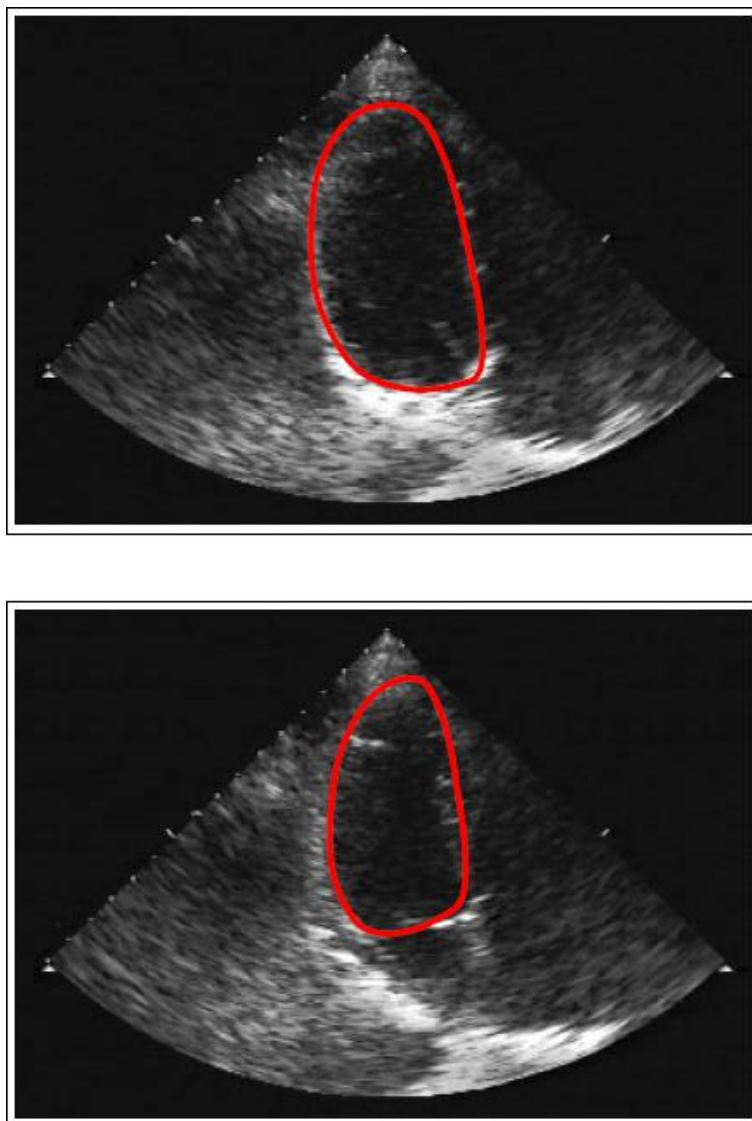
In computer visie is segmentatie het proces dat een digitaal beeld onderverdeelt in verschillende betekenisvolle objecten. De verdeling kan gebeuren op basis van verschillende eigenschappen. Zo kan er bijvoorbeeld gekozen worden om de objecten van beelden te verdelen op basis van kleur, intensiteit of locatie. Bij digitale segmentatie wordt er gekeken naar de gelijkenis en de verschillen tussen de eigenschappen van de pixels. Op deze manier kan beslist worden welke pixels samen horen. De pixels, die samen voldoen aan eenzelfde eigenschap, behoren tot eenzelfde object.

Doordat het gehele beeld verdeeld is in aparte objecten, kan elk object van het beeld apart aangesproken worden en kunnen er aparte operaties op uitgevoerd worden. Hierdoor wordt bijvoorbeeld het analyseren van beelden veel eenvoudiger. Segmentatie kan ook toegepast worden op videobeelden. Een extra eigenschap die video heeft tegenover een stilstaand beeld, is dat een video sequentie informatie bevat over de beweging van objecten, terwijl een beeld enkel statische informatie bevat. Bij video segmentatie wordt deze extra temporele eigenschap gebruikt voor in elk apart beeld van de video objecten te segmenteren. Deze segmentatie is consistent doorheen het verloop van de video.

Segmentatie van digitale beelden en video wordt in vele verschillende toepassingen gebruikt. Hier worden er enkele besproken:

Medische analyse:

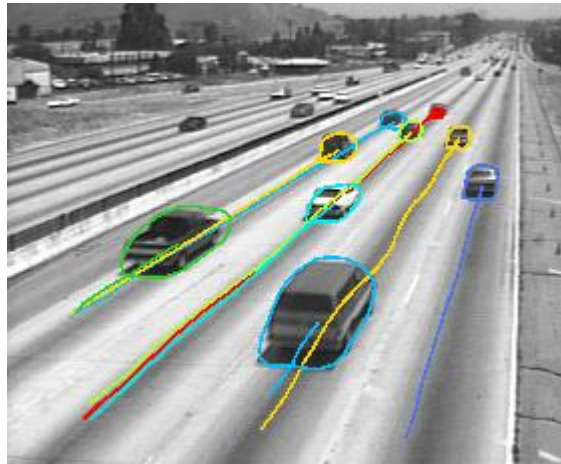
In de medische sector wordt veel gebruik gemaakt van speciale apparatuur, zoals ultrasound en x-ray machines, die beelden kunnen nemen van inwendige regio's van de mens. De verkregen beelden met zulke apparatuur zijn op eerste zicht niet erg duidelijk en bevatten veel ruis. Het is dan moeilijk voor de arts om specifieke zaken zoals tumors of abnormale hartkloppingen te herkennen (figuur 1.3). Door gebruik te maken van segmentatie wordt het medische beeld onderverdeeld in verschillende objecten. Hierdoor worden de randen tussen verschillende objecten duidelijker.



Figuur 1.3: Segmentatie van het kloppen van het hart. [4]

Verkeerscontrole:

Bij verkeerscontrole kan het handig zijn om de voertuigen die zich in het verkeer bevinden beter te analyseren. Door in een video sequentie van het verkeer bepaalde voertuigen te segmenteren kan de computer deze voertuigen makkelijker volgen doorheen de video. Immers het herkennen en identificeren van voertuigen is makkelijker voor de computer wanneer het voertuig gescheiden is van alle andere objecten uit de video. Hierdoor kan de computer gegevens zoals snelheid en rijgedrag van dit individueel voertuig registreren. Bijvoorbeeld in figuur 1.4 worden voertuigen worden gesegmenteerd uit een video. De lijnen achter de voertuigen laat hun verplaatsing zien.



Figuur 1.4: Gesegmenteerde voertuigen uit een video. [44]

Entertainment industrie:

De entertainment industrie maakt veel gebruik van segmentatie methodes. Door objecten te segmenteren uit een beeld kunnen er operaties zoals veranderingen in contrast, kleur en tint op een object toegepast worden zonder de rest van het beeld te wijzigen. Ook kunnen gesegmenteerde objecten verwijderd worden uit een beeld of video en vervangen worden door computer animatie. Eerder verwijderde objecten kunnen ook daarna in een ander beeld of video worden toegevoegd. Hoewel bij het bewerken van beelden nog veel handmatig wordt gewerkt, kunnen segmentatie technieken een animator veel tijd besparen. Voorbeelden van het gebruik van segmentatie voor video bewerking zien we in figuur 1.5¹ en figuur 1.6². In figuur 1.5 krijgen de lichten van de auto een hogere intensiteit. Ook is de achtergrond verwijderd en vervangen door computer animatie. In figuur 1.6 werd de acteur verwijderd uit de video en vervangen door een computer animatie met behulp van segmentatie.

¹<http://www.bmw.com/>

²<http://www.starwars.com/>



Figuur 1.5: Voorbeeld van het toepassen van individuele filters.



Figuur 1.6: Voorbeeld van het vervangen van objecten.

In het volgende hoofdstuk wordt een overzicht gegeven van verschillende beeld segmentatie technieken. In het derde hoofdstuk wordt besproken hoe beeld segmentatie methodes kunnen uitgebreid worden naar video. Er worden ook specifieke segmentatie methodes besproken, die ontwikkelt zijn voor video segmentatie.

Een belangrijke groep van video segmentatie technieken maakt gebruik van tracking technieken, ook wel rotoscoping genoemd, om de objecten doorheen de video te volgen en vervolgens te segmenteren. In deze thesis wordt een segmentatie techniek die gebruik maakt van tracking methodes in detail besproken en geïmplementeerd. In het vierde hoofdstuk worden de details van de tracking methode: keyframe gebaseerde tracking [2] uitgelegd. In hoofdstuk vijf wordt een overzicht gegeven van de optimalisatie technieken die kunnen gebruikt worden voor het oplossen van keyframe gebaseerde tracking. In het volgende hoofdstuk worden de details gegeven voor de implementatie van deze techniek en worden er uitbreidingen voorgesteld in het hoofdstuk daarna. De resultaten van de implementatie worden besproken in het volgende hoofdstuk. In het laatste hoofdstuk word de conclusie gegeven.

Hoofdstuk 2

Beeld segmentatie

In computer visie slaat segmentatie terug op het proces dat een digitaal beeld verdeelt in verschillende onderdelen, regio's genoemd. Een regio is een wel bepaalde verzameling van pixels uit een digitaal beeld. Iedere pixel, of groep van pixels, van een beeld wordt tijdens de segmentatie geanalyseerd en toegewezen aan een bepaalde regio. Een regio is dus een deelverzameling van de pixels, in een beeld, die voldoen aan eenzelfde eigenschap. Toepassen van segmentatie op een beeld versimpelt en/of verandert de representatie van het beeld. Door deze nieuwe representatie krijgt het beeld meer betekenis en is het eenvoudiger om het beeld te analyseren.

Het resultaat van beeld segmentatie is een verzameling van regio's die samen terug het volledige beeld vormen. Naast elkaar liggende regio's zijn verschillend ten gevolge van de gebruikte segmentatie eigenschap. Alle pixels in een regio voldoen aan dezelfde eigenschap. De eigenschappen kunnen simpel zijn, bijvoorbeeld de kleur, intensiteit, textuur of locatie van de pixel, of ze kunnen meer complex zijn. Complexe eigenschappen hebben meestal meer te maken met de betekenis achter de pixels, zoals behorend tot de voorgrond of de achtergrond van het beeld.

Er bestaan vele algoritmen en technieken om een beeld te segmenteren. Een algemene oplossing voor beeld segmentatie bestaat er echter niet. De bestaande technieken zijn meestal afhankelijk van het uiteindelijk doel, dat moet bekomen worden met behulp van de segmentatie.

Er zijn twee basis aanpakken voor het segmenteren van beelden: regiogebaseerde en randgebaseerde.

Bij regiogebaseerde technieken wordt er gekeken naar de uniformiteit in de gewenste eigenschap van pixels binnen een regio voor het verdelen van een beeld. Technieken zoals K-means clustering [26], Region-growing [6] en Split-and-Merge [16] zijn voorbeelden van regiogebaseerde technieken. Zij kijken in hoeverre pixels gegroepeerd kunnen worden onder eenzelfde eigenschap.

Randgebaseerde technieken kijken in hoeverre pixels verschillen van elkaar voor een bepaalde eigenschap. In deze technieken wordt er gezocht naar de discontinuïteit tussen

regio's, genaamd de randen. Technieken zoals snakes [21], active contours [4], intelligent scissors [29] en technieken gebaseerd op level-sets [36] [30] gaan actief zoeken naar de rand tussen de verschillende regio's en zijn verwant met technieken uit randdetectie. Een geavanceerde randtechniek beschouwt het beeld als een gewogen graaf, voorbeeld hiervan is Graph Cut [41].

Een uitbreiding voor technieken van regio en rand is werken met multi-scale. Bij multi-scale word een segmentatie meerdere keren uitgevoerd op het zelfde beeld maar op een andere resolutie. De resultaten van de segmentatie bekomen op lage resoluties worden herbruikt voor de segmentatie op hogere resolutie. Hierdoor wordt de segmentatie op hoge resolutie versneld.

Een ander onderscheid tussen segmentatie technieken kan gemaakt worden aan de hand van de gebruiker interactie. De meeste regiogebaseerde technieken vragen weinig interactie van de gebruiker, waardoor deze als automatische technieken worden omschreven. Randgebaseerde technieken zoals snakes [21] vragen meestal meer gebruikersinteractie in de vorm van initialisatie of correcties. Deze technieken worden dan semi-automatisch genoemd.

Verschillende technieken kunnen bij eenzelfde beeld heel andere resultaten geven. De keuze voor een segmentatie algoritme hangt sterk af van de applicatie.

2.1 Regiogebaseerde Technieken

Regiogebaseerde technieken zoeken naar uniformiteit binnen een regio gebaseerd op een bepaalde eigenschap zoals kleur, intensiviteit en textuur. Segmentatie technieken, die zoeken naar gelijkheid, maken gebruik van clustering methodes voor het verdelen van de pixels in verschillende regio's.

Clustering is een techniek uit statistische data analyse, die zoekt naar gelijkheden in een data-set. Clustering methodes worden gebruikt om een data-set te verdelen in verschillende sub-sets, clusters genaamd. De verdeling gebeurt zodat de data toebehorend aan eenzelfde cluster voldoet aan eenzelfde eigenschap. Data-clustering wordt gebruikt in verschillende toepassingen zoals machine learning, data-mining, bio-informatica en kan ook gebruikt worden voor het segmenteren van beelden.

Een methode voor de clustering van data kan hiërarchisch of partioneel zijn.

Bij partionele methodes wordt in het begin van het algoritme het aantal clusters vastgelegd, die doorheen heel het algoritme gebruikt worden. Het aantal clusters wordt nooit gewijzigd doorheen het verloop van het algoritme. Een voorbeeld van partionele clustering is K-Means [26]. Het aantal clusters bij de K-Means methode moet bepaald worden voor de start van het algoritme.

In hiërarchische clustering methodes beginnen de algoritmes met een bepaald aantal clusters. Het aantal clusters wordt doorheen de loop van het algoritme aangepast door telkens de informatie van de oude clusters te evalueren. Hiërarchische clustering methodes kunnen verder onderverdeeld worden in agglomeratieve of divisieve clustering methodes.

Agglomeratieve, of bottom-up, hiërarchische clustering methodes beginnen met elk element van de data te beschouwen als een aparte cluster. Doorheen het algoritme worden de aparte clusters samengevoegd in telkens grotere clusters. Region-growth [6] is een agglomeratieve hiërarchisch clustering methode waarbij telkens kleinere clusters worden samengevoegd tot grotere clusters.

divisieve, of top-down, hiërarchische clustering methodes beschouwen in het begin de volledige data-set als één cluster, die vervolgens wordt verdeeld in steeds kleinere clusters. Histogram clustering methodes [41] zijn een voorbeeld van divisieve clustering methodes waarbij één grote cluster telkens verdeeld wordt in kleinere clusters.

Er zijn twee belangrijke kwesties waarbij rekening gehouden moet worden bij clustering: cluster afstand (distance) en het aantal clusters.

Cluster afstand is een eigenschap, die een pixel binnen bepaalde grenzen moet hebben, om tot de cluster te kunnen behoren. Een simpel voorbeeld van cluster afstand is verschil in locatie tussen een pixel en de kern van de cluster. De cluster afstand kan ook gebaseerd zijn op andere eigenschappen zoals het verschil in intensiteit.

In de agglomeratieve hiërarchische clustering methodes wordt de cluster afstand gebruikt om te beslissen of overeenkomstige en gelijkaardige clusters kunnen worden samengevoegd. Bij divisieve hiërarchisch clustering methodes wordt cluster afstand gebruikt om te beslissen om een cluster te splitsen, in het geval deze onvoldoende samenhangend is.

Voor partitionele clustering methodes, waar het aantal clusters vast is, wordt de cluster afstand gebruikt om te oordelen bij welke cluster een element van de data-set, dat nog niet is toegevoegd, thuishoort. De definitie van de cluster afstand is afhankelijk van de data-set en de eigenschappen die aan de clusters zijn toegewezen.

Cluster afstand kan gezien worden als een waarde voor het splitsen of samenvoegen van clusters. Men dient een aanvaardbare instelling te vinden voor het criterium van gelijkenis, de bovenwaarde voor het splitsen en de onderwaarde voor het samenvoegen. Bij segmentatie worden deze waarden gebaseerd op eigenschappen zoals kleur, intensiteit, textuur of locatie van de pixels.

Bepalen hoeveel clusters er zijn is een moeilijke taak. Bij partitionele clustering methodes moet er in het begin van de methode bepaald worden hoeveel clusters er worden gebruikt. Dit aantal wordt nooit gewijzigd. De keuze van het aantal clusters heeft veel invloed op het uiteindelijke resultaat van het algoritme. Bij hiërarchische clustering methodes wijzigt het aantal clusters doorheen het algoritme. Hier is de vraag wanneer het algoritme moet stoppen

met het veranderen van het aantal clusters. Indien er geen drempel wordt vastgelegd voor het aantal clusters kan het resultaat zijn dat elke pixel een cluster op zijn eigen is of dat het volledig beeld als cluster wordt beschouwd. Het bepalen van het uiteindelijke aantal clusters is een cruciale taak.

2.1.1 Partitionele clustering methode: K-Means

Een bekende partitionele cluster methode is K-means clustering. Deze simpele clustering methode wordt veel gebruikt bij data analyse.

In het begin van het algoritme worden er K aantal clusters gekozen. Dit kan volledig willekeurig zijn of aan de hand van een heuristiek. Elke cluster wordt verondersteld een centrum te hebben. De cluster afstand kan berekend worden tussen pixels van het beeld en elk van de K centrum punten. Elke pixel van het beeld wordt toegewezen aan een cluster waarvoor de cluster afstand tussen de pixel en het centrum van de cluster wordt geminimaliseerd.

Als alle pixels zijn toegewezen aan een cluster wordt per cluster het nieuwe centrum punt berekend. Vervolgens worden de pixels opnieuw verdeeld over de clusters, door weer te kijken naar de cluster afstand. Dit wordt herhaald totdat er geen pixels meer zijn die van cluster veranderen.

De K-means methode verzekert dat het algoritme convergeert, echter de oplossing is niet altijd ideaal. Het kan zelfs voorkomen dat er geen enkele pixel wordt toegewezen aan een cluster. De kwaliteit van de oplossing hangt af van de keuze van het aantal en de initiële waarde van de centrum punten. De K-means methode is ook zeer gevoelig voor verstrooide data (ruis) en uitschieters.

2.1.2 Divisieve hiërarchische clustering methode: Histogram clustering

Anders dan bij de partitionele methodes, ligt het aantal clusters bij hiërarchische clustering methodes niet vast. Echter het gebruiken van hiërarchische methodes voor het segmenteren van beelden heeft veel problemen. Het grootste probleem is dat een beeld (de data-set) enorm veel pixels heeft om te evalueren. Het algoritme moet telkens de cluster afstand voor alle elementen uit de data-set berekenen naar alle andere elementen van de data-set. Hierdoor wordt het een zeer dure operatie om te beslissen welke clusters moeten worden samengevoegd of gesplitst. Eén van de vele oplossingen hiervoor is, in plaats van alle pixels, gebruik te maken van het centrum van een cluster, zoals bij de K-means methode. Een ander probleem is het beslissen wanneer het algoritme mag stoppen met het splitsen of samenvoegen van clusters.

Een techniek voor een divisieve hiërarchische clustering is het histogram clustering methode [8]. In deze techniek wordt een histogram berekend van alle pixels in het beeld. Een histogram is een grafische weergave van data in het frequentiedomein. Het toont aan hoe

vaak een bepaalde waarde voorkomt in de data, bijvoorbeeld hoe vaak er een pixel rood gekleurd is in het beeld. Uit de vorm, meer bepaald de pieken en dalen, van het histogram kan informatie gehaald worden voor het vinden van clusters. Bijvoorbeeld, indien de intensiteit histogram pieken geeft die overeenkomen met regio's met een hoge intensiteit. De histogram van het volledige beeld stelt de cluster voor die alle pixels van het beeld bevat.

Op het histogram kan nu de gepaste threshold techniek toegepast worden. Hierdoor wordt voor elke piek of dal uit het histogram een nieuw histogram gemaakt, die elk een nieuwe cluster voorstelt. Elke gescheiden histogram kan weer opnieuw worden gescheiden totdat er geen pieken of dalen meer gevonden worden. Door elk nieuwe histogram terug om te zetten naar pixels, worden de verschillende regio's van het beeld bekomen.

Een nadeel van het gebruiken van histogram clustering technieken voor segmentatie is dat het zeer moeilijk kan zijn om significante pieken of dalen te identificeren. Dit kan de kwaliteit van de oplossing sterk beïnvloeden.

2.1.3 Agglomeratieve hiërarchische clustering methode: Region-growing methode

De region-growing techniek [6] is een agglomeratieve hiërarchische clustering methode voor het vinden van clusters. Bij de start van het algoritme wordt er één of meerdere pixels gekozen, deze worden de seed pixels genoemd. Deze seed pixels kunnen gezien worden als clusters met maar één element. Vervolgens worden de aanliggende pixels geëvalueerd en toegevoegd aan de clusters indien ze voldoende gelijkaardig zijn aan de seed pixel. Indien een pixel onvoldoende overeenkomt met de cluster wordt deze pixel een nieuwe seed pixel en wordt er een nieuwe cluster gestart. Het algoritme stopt indien er geen pixels meer zijn die kunnen toegevoegd worden aan de clusters. De region-growing techniek wordt veel gebruik bij medische beelden voor het zoeken van tumors.

De initiële keuze van de seed pixels is afhankelijk van de segmentatie die moet bekomen worden. Een foute keuze kan leiden tot een slechte kwaliteit van de oplossing. Dit probleem is te vergelijken met de keuze van K clusters in het K-means algoritme. Een veel gebruikte techniek voor het vinden van de seed pixel is gebruik te maken van het histogram van het beeld. De hoogste piek of laagste dal wordt dan gekozen als de seed pixel.

Een ander probleem van de region-growing methode is dat een bepaalde cluster dominant kan worden tijdens het verloop van het algoritme. De volgorde van het evalueren van pixels heeft hier veel invloed op. Bij het evalueren van een pixel wordt er beslist of deze kan toegevoegd worden aan de cluster. Het kan zijn dat er besloten wordt dat deze pixel juist voldoende overeenkomt met de eigenschappen van de clusters. Het kan echter zijn dat er een cluster bestaat waarvoor de pixel heel sterk voldoet aan de eigenschappen van de cluster. Maar omdat deze cluster later werd verwerkt heeft het niet de kans gekregen om de pixel toe te voegen. Een oplossing hiervoor is voorkomen dat een cluster dominant wordt, door er voor te zorgen dat elke cluster parallel met elke andere cluster groeit.

Region-growing kan gezien worden als een speciaal geval van region-merging [19]. In Region-merging worden volledige clusters samengesmolten tot één grotere cluster. Bij region growing worden enkel pixels toegevoegd aan clusters. Region-merging kan gebruikt worden wanneer twee aanliggende clusters weinig van elkaar verschillen. Door deze te laten samensmelten worden overbodige clusters verwijderd. Voor het beslissen om twee clusters samen te voegen wordt er gebruik gemaakt van een statistische gelijkentest. Een veel gebruikte test is de Fisher'afstand [32]. Neem bijvoorbeeld twee clusters C_1 en C_2 met n_1 en n_2 de grootte van de clusters, μ_1 en μ_2 de gemiddelde en σ_1 en σ_2 de variantie. Om te beslissen om de twee clusters samen te voegen wordt dan de Fisher afstand berekend:

$$\frac{(n_1 + n_2)(\mu_1 - \mu_2)}{n_1\sigma_1^2 + n_2\sigma_2^2} = \frac{n\sigma^2}{n_1\sigma_1^2 + n_2\sigma_2^2} - 1 \quad (2.1)$$

Met $n = n_1 + n_2$ en σ de variantie van de gemengde cluster. Als deze waarde kleiner is dan een bepaalde drempel worden de cluster samengesmolten.

Een uitbreiding van region-merging techniek is region competition methode [48] [47]. Bij region-competition worden bijkomende criteria gebruikt voor het samensmelten van clusters. Bij vorige methodes werd er enkel gekeken naar de bepalende eigenschappen van de clusters. Region-competition kijkt naar de regio die de cluster moet voorstellen. Eigenschappen zoals vorm van de regio en scherpte van de rand van de regio spelen mee in de beslissing om regio's samen te voegen. Een alternatief voor regio-growing technieken is split-and-merge [16]. Bij de split-and-merge methode worden er niet enkel regio's samengevoegd maar ook gesplitst in de loop van het algoritme. Dit is dan meer een combinatie van agglomeratieve en divisieve clustering methodes.

2.1.4 Model-gebaseerde methode

Bij region-merging hebben we Fisher afstand vermeld, om te beslissen wanneer twee clusters samengevoegd kunnen worden. Dit is een voorbeeld van een clustering methode waar de clusters worden voorgesteld als modellen voor het segmenteren van een beeld.

Bij model-gebaseerde methodes wordt gebruik gemaakt van bepaalde modellen als clusters. De clustering bestaat erin om een optimale koppeling te vinden tussen het model en de data. Elke cluster kan mathematisch voorgesteld worden als een parametrische distributie. Parametrische distributie slaat terug op de verdeling van de parameters van een functie in tegenstelling tot de verdeling van de eigenlijke waarden bekomen met een functie. Deze distributie kan continu, zoals bijvoorbeeld een gaussiaan, of discreet zijn, zoals bijvoorbeeld Poisson. De volledige data-set wordt dan gemodelleerd als een verzameling, een mixture genoemd, van al deze distributies. Deze modellen van de clusters worden dan gebruikt voor het toevoegen van elementen aan de cluster of om te beslissen over het splitsen of samenvoegen van clusters.

Een veel gebruikte clustering methode maakt gebruik van het mixture of gaussians model [12]. Elke cluster wordt beschouwd als een gaussiaanse distributie gecentreerd rond een

seed punt. De seed punt is het gemiddelde van de gaussiaanse distributie. In elke iteratie wordt gekeken per cluster welke punten er kunnen worden toegevoegd aan de gaussiaan. Voor elke cluster wordt na elke iteratie het nieuwe gemiddelde berekend. Deze methode is gelijkaardig aan de K-means methode.

In de praktijk is het onmogelijk om rechtstreeks de parameters uit de clusters te halen, die nodig zijn voor het modelleren van de cluster. Deze parameters kunnen enkel bekomen worden uit de geobserveerde data en door gebruik te maken van de informatie of het beeld. Een populaire statistische aanpak maakt gebruik van Bayes's Rule [39] [40] en de kansverdelingfunctie $p(I(x, y), \theta_m)$. Deze kansverdeling stelt de kans voor dat bepaalde data $I(x, y)$ zal worden geobserveerd, gegeven dat de parameters van cluster m voldoen aan θ_m .

2.2 Randgebaseerde Segmentatie

Randgebaseerde segmentatie zoekt naar discontinuïteit van intensiteit in een beeld. Bij regiogebaseerde technieken gaat men zoeken naar regio's van pixels die samen horen. Pixels die gelijkaardig zijn worden samen gezet en pixels die anders zijn worden uitgesloten van de regio. Regio's worden gescheiden door hun ongelijkheid. Bij randgebaseerde technieken gaat men juist zoeken waar deze ongelijkheid van pixel plaats vindt, anders gezegd het lokaliseren van de randen van de regio's. Dit soort methodes zijn gelijkaardig aan randdetectie technieken. Een vereiste bij randgebaseerde segmentatie methodes is dat elke regio in het beeld voldoende uniform is, zodat de transitie tussen regio's kan gebaseerd worden op discontinuïteit. Indien dit niet het geval is, is het beter regiogebaseerde technieken te gebruiken.

2.2.1 Lokale Filtering

Een basis methode voor het vinden van randen is lokale filtering, bekend uit randdetectie. Het onderliggend idee van randdetectie is het berekenen van een lokale afgeleide operator, de gradiënt. Door deze operatie toe te passen op de pixels van het beeld, worden de locaties gevonden waar er een sterke discontinuïteit van intensiteit is.

De gradiënt vector van pixel in een beeld wijst in de richting van de grootste overgang of discontinuïteit in intensiteit. De gradiënt wordt bekomen door de partiële afgeleiden te berekenen op elke locatie in het beeld. De magnitude, of te wel de grootte van de gradiënt, bepaalt hoe sterk de discontinuïteit in intensiteit is op zijn locatie. Op basis van de richting van gradiënt en de waarde van de magnitude kan beslist worden, waar de randen van de regio's zich bevinden. Eenmaal de randen gevonden kan het beeld worden gesegmenteerd in verschillende regio's. Er zijn twee bekende methodes voor het vinden van de gradiënt in het beeld: de methode van Sobel [17] en de methode van Canny [7].

De Sobel methode berekent de gradiënt met behulp van een convolutie filter. De waarde van de gradiënt in een pixel wordt berekend als een gewogen gemiddelde van de waarde van zijn burens. Het voordeel om gebruik te maken van een convolutie filter is dat het naast

het geven van de afgeleide ook zorgt voor een gladde overgang, waardoor ruis verwijderd wordt. De methode van Sobel wordt veel gebruikt omwille van zijn eenvoudigheid.

De methode van Canny voor randdetectie bestaat uit verschillende stappen. Het is gebaseerd op het extremum van de eerste afgeleide van een gaussiaanse operator toegepast op een beeld. De verschillende stappen worden hier kort besproken. Eerst wordt de ruis verwijderd met behulp van de operator. Vervolgens gaat het op zoek naar de locaties met een hoge gradiënt. In de non-maximum suppressie stap word er gekeken welke gradiënten bekomen uit de vorige stap een lokale maximum zijn. Na de non-maximum suppressie stap worden de randen bepaald met behulp van twee drempelwaardes. Canny randdetectie wordt gezien als een optimale methode, omdat het een lage kans heeft op foutieve resultaten.

Randdetectie met behulp van de gradiënt zoals Sobel en Canny werken enkel goed wanneer er een sterke overgang is van intensiteit en er relatief weinig ruis is. Omwille van de gevoeligheid voor ruis is het soms vereist om een extra ruis reductie algoritme, als preprocessor stap, uit te voeren. Bij de methode van Canny is de ruis reductie al toegevoegd in het algoritme. Echter dit kan tot gevolg hebben dat de randen in het beeld wazig worden. Een groot voordeel bij lokale filtering technieken is dat de computatie kost heel laag is.

2.2.2 Active Contours

Een andere aanpak voor Randgebaseerde segmentatie is het gebruik maken van Active contours [4] of Snakes [21]. Deze technieken maken gebruik van energie minimaliserende curve. De Snakes methode begint met het initialiseren van een curve die ruw de contouren van de regio volgen, die men wilt segmenteren. Het initialiseren van deze curve kan gedaan worden door de gebruiker zelf of kan bekomen worden door gebruik te maken van randdetectie technieken.

Deze initiële curve kan geëvalueerd worden, in de mate dat deze overeenkomt met de eigenlijke contouren van de regio. De overeenkomst wordt geëvalueerd als een energie functie. De energie functie wordt berekend met de waarde van de curve en geeft een waarde terug, die vertelt hoe sterk de curve overeenkomt met de eigenlijke contouren. De energie functie is zo gedefinieerd dat, indien de curve perfect overeenkomt met de contouren, het de waarde nul terug geeft. De energie functie kan gezien worden als een functie die de curve afstraft, en dus een hoge waarde geeft op verschillende punten zoals intensiteit, vorm van de curve, scherpheid op hoeken van de regio, en waardes van omliggende gradiënten, indien deze niet overeenkomen met de contouren van de regio.

De Snake methode gaat na de initialisatie van de curve deze iteratief aanpassen, om zodanig de energie functie te minimaliseren. In elke stap wordt de waarde van de energie functie kleiner tot het zijn minimum bereikt heeft. De aangepaste curve die de energie functie minimaliseert is dan de beste benadering van de eigenlijke contouren. Er bestaan verschillende methodes voor het minimaliseren van een energie functie. Een bekend algoritme voor energie minimalisatie is deze van Levenberg-Marquardt [22] [27].

2.2.3 Intelligent Scissors

Een andere randgebaseerde segmentatie methode is Intelligent Scissors [29]. In deze methode moet de gebruiker, in het begin van het algoritme, een seed punt aanduiden. Deze kan eender welke pixel in het beeld zijn. Vervolgens begint de gebruiker zijn muis te bewegen doorheen het beeld. Het algoritme berekent dan het pad van het seed punt naar de positie van de muis. Dit pad probeert zo goed mogelijk de contouren, die tussen het seed punt en de muis positie liggen, te volgen. Dit pad wordt de live-wire genoemd. Indien de gebruiker ruw de contouren van de regio volgt, zal de live-wire zich op de contouren leggen van de regio. Dit is vergelijkbaar met Snakes, met het verschil dat de curve zich interactief vormt.

De live-wire, die zich legt op de contouren, wordt berekend door het beeld om te vormen tot een graaf zoals bij de normalized cut methode [37] en de Graph Cut methode [41] (zie volgende sectie) bij regiogebaseerde segmentatie. Het verschil is hier dat er actief gezocht wordt naar de randen van regio's. Elke pixel in het beeld is een knoop met een boog naar elke van zijn 8 burens. Elke boog heeft een waarde die gerelateerd is met de gradiënt van de boog in het beeld. Het pad wordt dan gevonden bij het vinden van het minimale pad van het seed punt naar de positie van de muis. Het minimale pad is geneigd om de contouren van het beeld te volgen in plaats van ze te kruisen, omdat dit grotere waarden zal creëren. Voor het vinden van het minimale pad worden technieken zoals Dijkstra [13] gebruikt.

2.3 Graaf methode

Bij Intelligent Scissors wordt het beeld voorgesteld als een gewogen ongerichte graaf. Deze graaf wordt gebruikt voor een kortste pad te berekenen. Men kan ook de graaf representatie van het beeld gebruiken om actief te gaan zoeken naar de randen tussen regio's. Elke pixel een knoop in de graaf. Vanuit elke pixel wordt een boog gevormd naar al zijn burens. Het gewicht van een boog meet de continuïteit tussen twee pixels. Bogen tussen gelijkaardige pixels krijgen een hoog gewicht. Hierdoor hangen gelijkaardige pixels sterker aan elkaar dan andere.

De segmentatie wordt bekomen door de graaf te verdelen in partities. Dit gebeurt door bogen uit de graaf te verwijderen aan de hand van hun gewicht. Dit proces wordt de cut genoemd. De normalized cut methode [37] zoekt de optimale verdeling van de graaf door de bogen, die verwijderd werden tijdens de cut, te minimaliseren. Methodes zoals de normalized cut methode, die gebruik maken van een graaf representatie van de data, worden ook wel graph partitioning methodes genoemd. Een specifieke methode hiervan, ontwikkeld voor beeldsegmentatie, is de Graph Cut methode.

2.3.1 Graph Cut Segmentatie

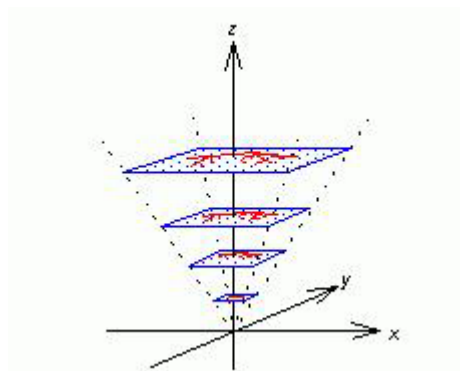
Graph Cut Segmentatie is een andere regiogebaseerde segmentatie methode die, zoals de normalized cut methode [37], het beeld voorstelt als een gewogen graaf. Het verschil met normalized cut is dat deze specifiek ontwikkeld is voor beeld segmentatie. Hierdoor kan deze methode veel beter een ideale cut verwezenlijken, op basis van de eigenschappen van het beeld.

Net zoals bij normalized cut worden de pixels voorgesteld als knopen en is elke pixel verbonden met zijn 8 burens via een boog, deze bogen worden rand-bogen genoemd. Het gewicht van een boog stelt de gelijkheid tussen de pixels voor. Daarenboven is elke pixel verbonden met een source en een sink knoop. Deze stellen beide een andere regio voor. In dit geval zijn er enkel 2 regio's, de voorgrond en de achtergrond. Het gewicht van de bogen tussen de pixels en de source of sink knoop hangt af hoe goed de pixel voldoet aan de eigenschappen van die regio, deze bogen worden regio-bogen genoemd.

De minimale cut moet nu rekening houden met zowel de rand-bogen als met de regio-bogen. Door gebruik te maken van optimalisatie technieken kan de ideale balans gevonden worden tussen rand- en regio-bogen voor het vinden van de minimale cut. De ideale cut wordt gevonden met behulp van een energie functie. Deze energie functie geeft aan hoeveel energie er nodig is voor een bepaalde cut. De energie functie is gebaseerd op de waarde van de bogen. Voor de minimale cut geeft de energie functie de minimale waarde.

2.4 Multi-scale methode

De space-scale is een raamwerk voor het voorstellen van een beeld op verschillende schalen. Een schaal in de space-scale beschrijft in hoeverre details herkenbaar zijn in een beeld. Voor een digitaal beeld zijn de schalen van de space-scale verschillende resoluties van hetzelfde beeld. De space-scale kan voorgesteld worden als een diagram zoals in figuur 2.1. De verschillende vlakken stellen hetzelfde beeld voor op verschillende resoluties. De resolutie stijgt naarmate de space-scale groter wordt.



Figuur 2.1: Een space-scale diagram. [31]

Multi-scale segmentatie maakt gebruik van meerdere instanties van een beeld op verschillende schalen van het space-scale raamwerk. Dit gebeurt meestal beginnend met de instantie met een lage resolutie (weinig detail zichtbaar) gevolgd door instanties met steeds hogere resoluties (meer detail zichtbaar). De informatie van vorige instanties wordt telkens gebruikt in de volgende. Er moet voor gezorgd worden dat de segmentatie regel op elk van de instanties dezelfde betekenis heeft. De segmentatie bekomen op een beeld met weinig detail is sneller dan een beeld met veel detail, maar is natuurlijk minder correct. Maar door deze ruwe schatting te gebruiken bij een beeld met steeds meer detail is een groot deel van

het werk al geleverd. Hierdoor moet de segmentatie op een beeld met hoge detaillering, dat veel tijd vraagt, minder werk doen. Ook kan, na een bepaald moment, blijken dat de segmentatie niet veel meer verandert, indien men de detaillering verhoogt. Hierdoor kan men het algoritme eerder stoppen op een lagere detaillering.

2.5 Conclusie

In dit hoofdstuk werd beeld segmentatie besproken, met enkele van de belangrijkste methoden. Een van deze technieken is randgebaseerde segmentatie, waar actief gezocht worden naar randen. Een voorbeeld is actieve contours, waar een ruwe schatting van de rand wordt gegeven en door middel van iteraties deze rand wordt verfijnd. De methode van Snakes is hier binnen heel populair, omdat deze een energie functie gebruikt die geminimaliseerd kan worden om de ideale contour te vinden.

Hoofdstuk 3

Video segmentatie

Video is een sequentie van opeenvolgende stilstaande beelden. Voor het segmenteren van videobeelden kan er op elk apart beeld van de video-sequentie segmentatie toegepast worden. Dit is echter een zeer duur proces, omdat typisch een video 25 beelden per seconde heeft. Ook moet er voor gezorgd worden dat de segmentatie consistent is over de volledige sequentie, hier houden individuele beeld segmentatie technieken geen rekening mee. Voor deze reden wordt voor het segmenteren van een video sequentie gebruik gemaakt van de temporele eigenschap van video.

In beeld segmentatie zijn regio's te onderscheiden door kenmerken zoals bijvoorbeeld kleur, intensiteit en locatie. Dankzij de temporele eigenschap kunnen regio's in video's onderscheiden worden, omdat deze regio's zich bewegen door de scène in het tijdsverloop van de video. Pixels die samen door de video bewegen horen meestal tot eenzelfde object, en dus tot eenzelfde regio. Bij video segmentatie wordt er gekeken naar de verschillen in pixels tussen de beelden van de sequentie voor het vinden van de verschillende regio's. Verschillen tussen beelden in de sequentie ontstaan immers doordat pixels zich door de scène verplaatsen.

Indien er enkel gekeken wordt naar het verschil in pixels tussen opeenvolgende beelden, kunnen er verschillende problemen ontstaan. Objecten bewegen zich meestal niet-uniform door een scène en op een moment kan een object zelfs stilstaan. Hierdoor kunnen er tussen opeenvolgende beelden verschillen zijn. Het algoritme kan dan de pixels behorend tot het bewegend object fout identificeren. Bijvoorbeeld een persoon kan bij de regio van de achtergrond gevoegd worden. Een ander probleem is dat in vele video sequenties er weinig verschil is tussen opeenvolgende beelden. Dit komt bijvoorbeeld voor wanneer objecten een trage beweging maken door de scène. Hierdoor worden er geen bewegingen gedetecteerd en is er geen temporele informatie beschikbaar voor de segmentatie.

Vervolgens worden vier methodes besproken voor het segmenteren van video.

- Background subtraction [28] maakt gebruik van het verschil tussen de frames van de video en een gegenereerde achtergrond.
- Een tweede methode breidt de regiogebaseerde technieken uit beeld segmentatie uit naar video.

- Bij Video Cutout [23] wordt de video voorgesteld als een gewogen ongericht graaf, vergelijkbaar met Graph Cut methode [37] uit beeld segmentatie.
- Een laatste video segmentatie methode bouwt verder op randgebaseerde technieken, dit soort technieken zijn beter bekend als rotoscoping technieken.

3.1 Background subtraction

Een bekende techniek voor het vinden van bewegende objecten is background subtraction [28]. In deze methode wordt er een referentie frame gebruikt voor het vinden van veranderlijke pixels doorheen de tijd. Een referentie frame wordt zo gekozen dat het verschil tussen het referentie frame en de huidige frame veel duidelijker is, dan het verschil tussen twee opvolgende frames. Een referentie frame bestaat uit regio's met pixels die weinig tot geen beweging hebben en die samen terug het volledige beeld vormen. De regio's kunnen gezien worden als behorende tot de achtergrond van de scène. Een referentie frame kan daarom simpelweg de achtergrond genoemd worden. Door het huidige beeld te vergelijken met de achtergrond kunnen bewegende objecten gesegmenteerd worden.

Voor de keuze van een referentie frame zijn er drie algemene gevallen. Indien de video werd opgenomen in een gecontroleerde scène, bijvoorbeeld een filmstudio, kan er een beeld genomen worden waaruit alle bewegende objecten verwijderd zijn. Dit beeld kan dan gebruikt worden als achtergrond voor alle overige frames. Een variatie op deze methode is gebruik te maken van de Blue Screen techniek [33]. Bij de blue screen techniek worden de objecten opgenomen voor een groot blauw of groen scherm. In de segmentatie worden dan pixels geclassificeerd als behorend tot voorgrond of achtergrond op basis van hun kleur. Namelijk alle pixels die niet groen of blauw zijn behoren tot een voorgrond object.

Indien de gebruiker weinig controle heeft over de scène, moet er uit de video sequentie een beeld gekozen worden die de ideale referentie frame benadert. De referentie frame kan ook een nieuw beeld zijn dat is samengesteld met beelden van de video sequentie. Voor de referentie frame kan bijvoorbeeld het gemiddelde of de mediaan van de pixels genomen worden van een verzameling van beelden. Een derde situatie is wanneer de camera niet statisch is en dus zelf van positie verandert tijdens het tijdsverloop van de video. In zo'n geval moet de verplaatsing van de camera berekend worden aan de hand van de bewegingen van objecten die in de scène stil staan. Door deze verplaatsing te berekenen kunnen de frames van de video gewijzigd worden, zodat het lijkt dat de camera stil stond.

3.1.1 Uitbreiding van background subtraction: Moving average

In het geval dat de gebruiker weinig controle heeft over de scène kan het voorkomen dat de achtergrond in verloop van tijd langzaam verandert. Bijvoorbeeld in een scène die buiten wordt opgenomen verandert de belichting en schaduwen doorheen de video sequentie. Om deze langzame veranderingen op te nemen in de achtergrond worden de waarden van de pixels uit de referentie frame geschat aan de hand van een moving average. De waarde van een bepaalde achtergrond pixel wordt dan genomen als een gewogen gemiddelde van de vorige waardes. Waardes van pixels uit een ver verleden krijgen dan typisch een gewicht

nul en het gewicht stijgt langzaam in de richting van de huidige waarde. Verschillende technieken bestaan om gewichten aan frames toe te kennen zoals Mixture of Gaussian [43] en Kernel Density Estimators [31].

In het ideale geval worden enkel pixels in het gemiddelde opgenomen die inderdaad tot de achtergrond van de scène horen. Hiervoor moet er een filter voorzien worden zodat de pixels die behoren tot de objecten niet worden opgenomen. Een kenmerk van de achtergrond is dat deze traag verandert in tegenstelling tot de bewegende objecten. Door het kijken naar de frequentie van de verandering over de tijd van pixel kan er beslist worden of deze wordt opgenomen in het gemiddelde. Pixels met een frequentie hoger dan een bepaalde waarde worden genegeerd, en pixels met een lagere frequentie worden opgenomen.

In sommige gevallen kan het zijn dat pixels met een zeer hoge frequentie moeten worden genegeerd. Dit komt bijvoorbeeld voor in een scène met bomen en veel wind. In veel gevallen is de gebruiker niet geïnteresseerd in deze bladeren als een voorgrond object en wilt deze dus bij de achtergrond plaatsen. Maar omdat de bladeren continu bewegen worden deze opgenomen in het algoritme als aparte objecten. Om deze snelle objecten te negeren wordt er nog een extra drempelwaarde opgelegd. De extra drempelwaarde zorgt ervoor dat pixels met een te hoge frequentie niet worden genegeerd bij het zoeken naar verschil tussen het huidige en het referentie frame.

3.1.2 Uitbreiding van background subtraction: Clustering

Het resultaat van het verschil tussen de referentie frame en het huidige beeld is een zwart beeld met daarin witte pixels die de locatie aangeven van de pixels die verschillend zijn in de twee beelden. Deze pixels zijn soms verspreid over de oppervlakte van het object en zijn meestal niet te onderscheiden van willekeurige ruis. Dit kan het gevolg zijn van verschillende eigenschappen van de video. Een reden kan zijn dat niet alle pixels een (voldoende) verplaatsing hebben ondergaan. Een andere oorzaak van het probleem kan zijn dat het algoritme de verplaatsing van bepaalde pixels niet heeft opgemerkt omdat de kleur van het object te weinig verschilt met die van de achtergrond.

Technieken uit beeld segmentatie zoals K-Means clustering kunnen gebruikt worden om het resultaat van background subtraction te verbeteren. Regiogebaseerde Clustering methodes kunnen dan toegepast worden, op de pixels uit het verschil, om te zien welke bij elkaar horen. De cluster met de pixels gevonden met background subtraction wordt de initiële cluster genoemd. Clustering technieken toepassen op het verschil van twee beelden gebeurt snel omdat er relatief weinig pixels moeten worden geëvalueerd.

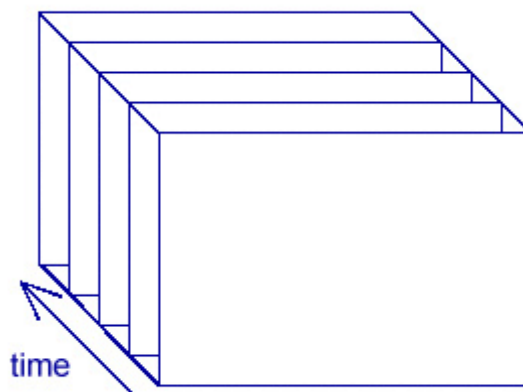
Voor segmentatie moeten alle pixels gevonden worden die behoren tot het object. Maar, zoals aangewezen in de vorige paragraaf, bevat de initiële cluster enkel de pixels die werden opgemerkt in het background subtraction algoritme. Met behulp van aangepaste technieken uit beeld segmentatie, zoals de K-Means techniek en region-growing techniek, worden de initiële clusters uitgebreid door in het huidige beeld te zoeken naar de overige pixels van het object. In dit geval is background subtraction een pre-processor stap.

3.2 Regiobaseerde methode

Regiobaseerde Clusterings methodes kunnen dus gebruikt worden in combinatie met background subtraction technieken. Een andere aanpak is om clusterings technieken uit beeld segmentatie uit te breiden naar video. In het Video Tooning systeem [33] worden clustering methodes gebruikt om de video te verdelen in 3D regio's.

Het basis idee is dat de video sequentie wordt voorgesteld als een 3D spatio-temporeel volume. Dit gebeurt door de frames uit de video sequentie achter elkaar te plaatsen in een 3D ruimte met de tijd als de derde dimensie. In figuur 3.1. word een visualisatie geven van video in een 3D spatio-temporeel volume.

Bij Video Tooning wordt op elk tijdstip apart een 2D clustering toegepast. Vervolgens wordt de gebruiker gevraagd om deze regio's handmatig samen te voegen in het 3D volume tot betekenisvolle regio's. Hierdoor kregen we de temporele consistentie, die vereist is bij video segmentatie. Een nadeel van deze methode is, dat deze veel gebruikers interactie nodig heeft.



Figuur 3.1: Een spatio-temporeel volume.

Een andere uitbreiding van clustering methodes naar video is het Video Paintbox systeem [9]. In deze methode wordt er, zoals bij Video Tooning, clustering methode individueel toegepast op elke 2D frame van de video. Vervolgens moeten de regio's temporeel gekoppeld worden. In plaats van dit handmatig te doen wordt er een heuristisch koppelingsalgoritme toegepast op basis van kleur, intensiteit, gebied, overlapping en vorm om de 2D regio's te associëren overheen de aanliggende frames van de video sequentie.

3.3 Video Cutout

Bij de Video Cutout methode [23] worden segmentatie methodes die het beeld voorstellen als een graaf, zoals de Graph Cut methode, uitgebreid voor gebruik in video segmentatie.

Ook hier wordt de video voorgesteld als een 3D spatio-temporeel volume. De bogen van de graaf worden dan ook getrokken tussen knopen op verschillende tijdstippen.

Het algoritme gaat als volgt te werk. In het begin van het algoritme moet de gebruiker twee keyframes kiezen uit de video sequentie. Op deze twee keyframes moet dan een beeld segmentatie methode toegepast worden die de voorgrond van de achtergrond scheidt. Keyframes worden typisch gekozen om de tien frames. Tussen deze keyframes wordt nu een 3D graaf geconstrueerd. In plaats van een graaf te bouwen op elke individuele pixel, wordt het gebouwd op basis van regio's [5]. Deze regio's worden bekomen met behulp van een presegmentatie.

Zoals in beeld segmentatie, waar het beeld voorgesteld is als een graaf, moet er nu gezocht worden naar de minimale cut. Hiervoor moet, net zoals bij Graph Cuts, een energie functie opgesteld worden. In de energie functie wordt het knippen van bogen, die regio's verbinden met overeenkomstige gesegmenteerde regio's uit de keyframes, afgestraft. De minimale cuts bevinden zich op de rand tussen twee regio's in dezelfde frame en tussen de regio's en de achtergrond die bepaald werd in de keyframes.

3.4 Randgebaseerde methode: Rotoscoping

Bij randgebaseerde beeld segmentatie methodes wordt er actief gezocht naar de randen van de verschillende regio's. Net zoals regiogebaseerde methodes kunnen randgebaseerde methodes op elke individuele frame van de video sequentie toegepast worden. Dit is zeer kostelijk en de temporele consistentie gaat verloren.

Echter in veel video sequenties gebeuren er weinig veranderingen van frame op frame. Typisch heeft een beeld 25 beelden per seconde. Met de individuele methode zou voor elk van deze 25 beelden segmentatie worden gedaan. Echter de meeste objecten in een video ondergaan weinig verandering tijdens een seconde. In principe kunnen de contouren van de eerste frame, die bekomen werden met een randgebaseerde methode, opnieuw gebruikt worden in de overige frames. Deze worden vervolgens aangepast volgens de kleine wijzigingen die het object heeft ondergaan.

De Snakes methode is een ideale kandidaat voor deze manier van werken. Bij Snakes wordt een initiële curve geleverd door de gebruiker. Het algoritme gaat dan deze initiële curve iteratief aanpassen aan de hand van een energie functie. Het algoritme stopt wanneer de aangepaste curve overeenkomt met de contouren van de regio. Voor video kan nu deze aangepaste curve overgezet worden op de volgende frame. Deze overgezette curve is dan de initiële curve voor deze frame. Het algoritme begint opnieuw totdat de curve op de contouren van de regio ligt. Dit wordt herhaald voor alle frames van de video sequentie. Omdat er meestal weinig verschil is tussen verschillende frames zijn er weinig aanpassingen nodig tussen opeenvolgende frames.

De aangepaste Snakes methode voor video kan gezien worden als een techniek die de bewegingen van de curve volgt doorheen de video. De curve volgt de contouren van het bewegend object. Deze methode van het volgen van contouren van objecten wordt roto-scoping genoemd. Rotoscoping is een verzamelnaam voor technieken die, met behulp van de gebruiker, gebruikt worden voor het volgen van vormen en curven doorheen een sequentie van beelden. Historisch is roto-scoping een manueel proces. Voor elke frame van de video wordt handmatig de contouren van de objecten overgetekend door de gebruiker. Door gebruik te maken van de temporele eigenschap van video kan voor roto-scoping een groot deel van het werk door de computer worden verricht. Rotoscoping met behulp van de computer wordt ook wel interpolated roto-scoping genoemd.

Bij de aangepaste Snakes methode is het nadeel dat de informatie, die de gebruiker bij de eerste frame geeft, verloren gaat. De volgende frames kunnen zich alleen maar baseren op het resultaat van de vorige. Dit kan tot gevolg hebben dat kleine fouten in loop van tijd zeer groot worden.

Een andere opmerking is dat aangepaste beeld segmentatie methodes, zoals Snakes methode, alleen maar voordeel heeft tegenover de individuele methode, indien de veranderingen tussen frames klein zijn. Er worden vervolgens enkele andere roto-scoping technieken besproken, die geen rechtstreekse aanpassing zijn van beeld segmentatie technieken.

3.4.1 Lineaire interpolatie

De simpelste methode voor roto-scoping is de gebruiker een curve te laten tekenen in een aantal niet opeenvolgende frames. In deze keyframes volgen de curves exact de contouren van de regio's. Voor de curve van de overige tussenliggende frames wordt lineaire interpolatie toegepast op de curve van de keyframes [18] [25]. Indien de tussenliggende curves te ver afwijken van de vorm die wordt gevolgd, kan de gebruiker deze curves nog aanpassen. De interpolatie wordt opnieuw gestart zodat deze nieuwe informatie van de aanpassing zich verspreidt over de vorige en volgende tussenliggende frames.

Lineaire interpolatie van de curves gebeurt door vanuit elk punt van de curve in een keyframe een rechte te trekken naar het overeenkomstige punt in de volgende keyframe. Het overeenkomstige punt van de curve in de tussenliggende frames ligt op deze rechte. Hoe ver een tussenliggende frame van een keyframe ligt, bepaalt de locatie van het tussenliggende punt op de rechte.

Simpele lineaire interpolatie kan enkel beperkt gebruikt worden. Enkel in specifieke gevallen van roto-scoping geeft lineaire interpolatie een aanvaardbaar resultaat. Doordat de interpolatie gebruik maakt van een rechte, geven enkel gevallen waarin objecten een simpele beweging maken volgens een rechte, goede resultaten. Echter voor complexe bewegingen zijn er enorme veel aanpassingen nodig van de gebruiker, waardoor de techniek onpraktisch wordt.

3.4.2 Motion capturing

Bij lineaire interpolatie heeft het algoritme enkel informatie van de verplaatsing die een object heeft ondergaan tussen twee keyframes. Het algoritme heeft geen informatie van welk pad het object heeft gevolgd tussen de twee keyframes, om van de ene locatie naar de andere te gaan. Bij de lineaire interpolatie wordt er aangenomen dat het pad een rechte lijn is. Dit komt enkel in zeer specifieke gevallen voor. Het pad waarlangs een object zicht beweegt tijdens zijn verplaatsing is meestal zeer complex. Om de complexe bewegingen, die een object maakt, te kunnen volgen heeft het algoritme meer informatie nodig.

Een bekende techniek om meer informatie te krijgen over een beweging is motion capturing [1]. In motion capturing technieken wordt tijdens het opnemen van een video sequentie de bewegingen van het object in de ruimte geregistreerd met behulp van speciale hardware. Op het te volgen object worden speciale markeringsen geplaatst die makkelijk voor een computer te volgen zijn. Markeringsen kunnen akoestisch, magnetisch, reflecterend of een combinatie van verschillende technieken zijn. Motion capturing geeft perfect weer hoe een object zich beweegt in de ruimte. Deze informatie, verkregen tijdens de opname, wordt dan gebruikt om de locatie van objecten te vinden in de video.

Motion capturing kan enkel en alleen gebruikt worden indien er zeer specifieke technieken werden gebruikt tijdens het opnemen van de video. Motion capturing is alleen mogelijk wanneer de gebruiker controle heeft over hoe de video werd opgenomen. Indien er met videobeelden wordt gewerkt waarvan de objecten niet werden gemarkeerd en geregistreerd tijdens de opname kan deze techniek niet gebruikt worden.

3.4.3 Kalman tracking

De informatie over de beweging van objecten met motion capturing technieken wordt verkregen tijdens de opname van de video. Hierdoor werkt motion capturing alleen maar met een zeer specifieke soort van video sequentie. Om informatie te vinden in alle soorten van video sequenties, wordt de informatie van de beweging uit de videobeelden zelf gehaald. Dit slaat neer op het zoeken naar de positie van de pixels van de contouren in de opeenvolgende frames van de video sequentie.

Een veel besproken techniek, voor het vinden van pixels in een beeld doorheen een video, is Kalman tracking [45]. De filosofie achter Kalman tracking is dezelfde als die van de Kalman filter. Een Kalman filter is een rekenmethode die gebruikt wordt om willekeurige ruis uit de gegevens te halen. De Kalman filter vindt nieuwe waarden voor gegevens, door deze te vervangen door gegevens met minder ruis.

In Kalman tracking worden de contouren van een vorige frame bekeken voor het vinden van de huidige contouren. Er wordt verondersteld dat de observatie van contouren uit de vorige frame veel last heeft van gaussianse ruis, zodat deze zogezegd moeten vervangen worden door een betere waarde. De betere waarden zijn hier dan de gewenste posities van de huidige contouren. Met de ruis veronderstelling kan, uit de gegevens van de vorige

frames, de conditionele dichtheid van de contouren gevonden worden. De dichtheid wordt dan gebruikt voor het schatten van de nieuwe positie van de contouren.

Een meer algemene vorm van Kalman tracking is Condensation (Conditional Density Propagation) tracking [20]. In Condensation tracking moet het systeem en het observatie proces, in tegenstelling met Kalman, niet lineair zijn.

3.4.4 Keyframe-gebaseerde tracking

Technieken, zoals de aangepaste Snakes techniek, voor het tracken van contouren worden feed forward technieken genoemd. Feed forward betekent dat, beginnend van een frame, de contouren enkel vooruit in de tijd worden gevolgd. Een probleem dat hieruit volgt is dat, wanneer er zich fouten voordoen, het heel moeilijk te achterhalen is, op welke frame het precies is fout gelopen. Een ander probleem is dat wanneer een gebruiker een fout probeert aan te passen deze correcties zich enkel propageren naar later frames, nooit naar vorige.

Een betere methode is gebruik te maken van keyframe-gebaseerde technieken, zoals bij lineaire interpolatie. Hier wordt er gezocht naar de optimale waarde in de frames tussen twee keyframes. Dit wordt zowel vooruit als achteruit in de tijd gedaan. Bij lineaire interpolatie wordt er echter geen rekening gehouden met de bewegingen van de contouren doorheen de video.

Keyframe-gebaseerde tracking [2] combineert de energie minimalisatie aanpak van de Snakes methode uit beeld segmentatie met de keyframe-gebaseerde aanpak van lineaire interpolatie. Bij keyframe-gebaseerde tracking worden er bij het begin van het algoritme twee keyframes gekozen. In deze keyframes wordt, zoals bij de Snakes methode, een initiële curve aangemaakt door de gebruiker. Deze curves volgen zo goed mogelijk de contouren van het te volgen object. Eenmaal de curves van de keyframes bekomen, gaat het algoritme de curves van alle tussenliggende frames berekenen, met behulp van een energie functie. Deze energie functie is zo gemaakt dat de curves, van vorige en volgende frames, rekening met elkaar moeten houden.

Deze methode kan gezien worden als de Snakes methode die tegelijkertijd op verschillende beelden wordt toegepast. Het verschil is dat de aanpassingen die een curve maakt in een frame, gevolgen heeft voor alle andere frames. Het doel van het algoritme is nu om voor elke frame de ideale curve te vinden, zodat de totale energie functie over alle frames wordt geminimaliseerd. Door deze methode kan de gebruiker een curve in eender welke frame aanpassen, indien deze afwijkt van de contouren van het object. Deze aanpassing propageert zich dan naar alle overige frames, wanneer de gebruik de minimalisatie herstart.

Hoe de energie functie van Keyframe-gebaseerde tracking methode er precies uitziet wordt besproken in het volgende hoofdstuk.

3.5 Conclusie

In dit hoofdstuk werd gezien hoe beeld segmentatie methodes konden uitgebreid worden voor gebruik in video segmentatie. Ook werden er enkele specifieke video segmentatie methodes besproken. De belangrijkste techniek besproken in dit hoofdstuk was segmentatie met behulp van keyframe gebasserde tracking. Deze methode wordt in de volgende hoofdstuk in meer detail besproken.

Hoofdstuk 4

De energie functie voor Keyframe gebaseerde tracking

In dit hoofdstuk wordt de energie functie van keyframe gebaseerde tracking [2] in detail besproken.

Vele segmentatie technieken maken gebruik van een energie functie. Energie functies worden zo opgesteld, dat een hoge waarde hebben (wanneer een foute beslissing wordt gemaakt bij het aanpassen van de parameters van het object) wordt veroordeeld. Een energie functie heeft een lage waarde, wanneer er een goede beslissing wordt genomen. Op deze manier zijn energie functies een soort van gids voor het aanpassen van de parameters van een object.

Om te oordelen of een aanpassing een goede beslissing is, maken energie functies een vergelijking tussen de huidige waarde en de voorgestelde nieuwe waarde van parameters.

Bijvoorbeeld als het een vereiste is dat het object zijn vorm moet behouden. De twee instanties van het object worden dan vergeleken met elkaar. Indien de voorgestelde aanpassingen er voor zorgen dat het object sterk van vorm verandert, zal de energie functie een hoge waarde terug geven. De uiteindelijke aanvaarde aanpassingen zijn diegene, die de vorm van het object zo weinig mogelijk veranderen.

Het zoeken naar de ideale aangepaste waarde van de parameters wordt minimalisatie of optimalisatie van de energie functie genoemd.

Een energie functie bestaat meestal uit meerdere termen. Het kan voorkomen dat de eigenschappen van de twee termen van de energie functie elkaar tegen spreken. In zo'n geval moet er een compromis gemaakt worden tussen de twee eigenschappen. Er kunnen ook gewichten aan de verschillende termen toegewezen worden. Hierdoor kan er aangegeven worden dat sommige eigenschappen belangrijker zijn dan andere.

Een voorbeeld van het gebruik van energie functies uit beeld segmentatie is de Snakes methode [21]. In deze methode geeft de energie functie, voor veranderingen aan para-

maeers, die er voor zorgen dat de curve dichter bij de contouren van het object liggen, een lage waarde.

Bij keyframe gebaseerde tracking [2] moeten de curves in alle tussenliggende frames voldoen aan dezelfde eigenschappen van de keyframes. Daarnaast moeten ze ook de bewegingen volgen van het object. Bij segmentatie moet de curve op de contouren van het object liggen. In het begin moet er een initiële curve aangemaakt worden in elke tussenliggende frame. Deze kan simpelweg een kopie zijn van de curve uit één van de keyframes. De parameters die de energie functie gaat beoordelen is de verzameling van alle curves in de tussenliggende frames. Het is belangrijk te zien dat aanpassingen niet gebeuren op één curve doorheen de tijd van frame op frame. Een instantie van de parameters bevat alle parameters van alle curves in alle tussenliggende frames.

4.1 Termen van de energie functie

De energie functie van keyframe gebaseerde tracking bestaat uit twee soorten van termen, shape termen en image termen. Bij keyframe gebaseerde tracking worden curves onderverdeeld in verschillende curves. Dit wordt gedaan omdat het makkelijker is om kleine korte curves te beoordelen dan één lange curve.

De shape termen straffen de curves af die te snel willen bewegen of te veel van vorm willen veranderen. De drie verschillende shape termen zijn de lengte term, de curvatie term en de snelheids term.

De image termen straffen de curves af die niet de beweging van het object volgen of niet op de contouren van het object liggen. De image termen zijn de image term en de edge term.

De verschillende termen zijn complementair. De shape termen kunnen geen bewegingen volgen van objecten, de image termen wel. Maar in situaties waar image termen geen resultaat kunnen leveren, zoals bij occlusie, zorgen de shape termen ervoor dat er toch nog een oplossing kan gevonden worden.

De uiteindelijke energie functie is een lineaire combinatie van vijf gewogen energie termen. Elke term heeft zijn eigen gewicht: w_L voor de lengte term, w_C voor de curvatie term, w_V voor de snelheids term, w_I voor de image term en w_G voor de edge term. De waardes voor de gewichten bij keyframe gebaseerde tracking zijn $w_L = 500$, $w_C = 40000$, $w_V = 0.1$, $w_I = 1$ en $w_G = 1000$ [2]. In de energie termen wordt er gekeken naar sample points van de curve. Elke curve wordt geschreven als $c_t(s)$. Met t die aangeeft in welke frame de curve zich bevindt en s die het sample point aangeeft.

4.1.1 Lengte term

De lengte term E_L bestraft de verandering van lengte in de curve. Er wordt gekeken naar de afstand tussen langs elkaar liggend sample points, en vergeleken met de overeenkomstige

afstand in de overige frames. De lengte term is de sommatie van de verschillen tussen de afstanden van alle samples points over alle frames.

$$E_L = \sum_{i,t} (\|c_t(s_{i+1}) - c_t(s_i)\|^2 - \|c_{t+1}(s_{i+1}) - c_{t+1}(s_i)\|^2)^2 \quad (4.1)$$

4.1.2 Curvatie term

De curvatie term E_C bestraft de verandering in de kromming in de curve overheen alle frames. Voor het meten van E_C wordt er gebruik gemaakt van de verandering in de tweede afgeleide van de lengte, als benadering van de effectieve verandering in kromming. De som van verschillen tussen de tweede afgeleiden, overheen alle frames, vormt E_C .

$$E_C = \sum_{i,t} \|(c_t(s_i) - 2c_t(s_{i+1}) + c_t(s_{i+2})) - (c_{t+1}(s_i) - 2c_{t+1}(s_{i+1}) + c_{t+1}(s_{i+2}))\|^2 \quad (4.2)$$

4.1.3 Snelheids term

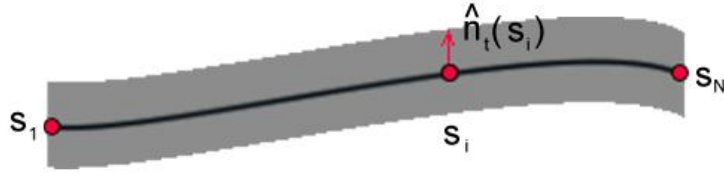
De twee vorige termen kijken enkel naar de verandering in de vorm van de curve. Het kan echter zijn dat de curve ook van positie verandert ten opzichte van de keyframes. Om dit in rekening te brengen wordt er gekeken naar de positie van overeenkomstige sample points in de verschillende frames en wordt de afgelegde verplaatsing berekend. Er wordt gekeken hoe snel een sample point van positie veranderd. Snelle veranderingen in positie van de sample points worden in de snelheids (velocity) term E_V afgestraft.

$$E_V = \sum_{i,t} \|c_t(s_i) - c_{t+1}(s_i)\|^2 \quad (4.3)$$

4.1.4 Image term

Voor het evalueren van de verandering in de beeldinformatie wordt er gekeken naar de pixels van de frame waarin de curve zich bevindt. Bij video segmentatie wordt algemeen aangenomen dat de verandering overheen de frames traag gebeurt. Er moet dus gekeken worden in de direct omgeving van de curve.

De theorie waarop de image term wordt gebaseerd is de bekende techniek van Lucas-Kanada [24] voor het tracken van beelden. In de techniek worden kleine vensters met image data vergeleken tussen twee frames. De techniek optimaliseert de locatie van het venster in de tweede frame om het verschil tussen de frames te minimaliseren. Dit idee wordt toegepast voor de image term. De interessante pixels voor de energie functie bevinden zich in een dunne band omheen de curve die wordt gevolgd. Indien de curve zich bevindt op de contouren van een bewegend object, worden enkel de pixels aan de zijde van het object bekeken. In de image term van de energie functie wordt nu gekeken in welke mate de pixels in de dunne baan rond de curve verschillen tussen twee instanties.



Figuur 4.1: De dunne band omheen een curve voor de image termen.

In E_I worden veranderingen afgestraft die de curve doet afwijken van het object waartoe zij behoren in de keyframes.

$$E_I = \sum_{i,k,t} \|I_t(c_t(s_i) + k \cdot n_t(s_i)) - I_{t+1}(c_{t+1}(s_i) + k \cdot n_{t+1}(s_i))\|^2 \quad (4.4)$$

De sample points die gebruikt worden in de image term E_I liggen in de richting van de normaal van de curve in frame t . Deze worden vergeleken met de sample points van frame $t + 1$. De eenheid normaal $\hat{n}_t(s_i)$ van de curve op punt s_i wordt gecreëerd door het roteren van de tangent vector $\frac{\delta c_t(s_i)}{\delta s_i}$ en deze te normaliseren door zijn lengte. $I_t(p)$ is de RGB kleur vector van het beeld in frame I op punt p . Let wel, RGB kleurwaarde is een vector met 3 elementen (R, G, B) in tegenstelling tot een punt dat een vector is van 2 elementen (x, y) . De variabele k varieert over de breedte van de band rond de curve. Voor tweezijdige tracking varieert k van $-n$ tot n , voor eenzijdige tracking varieert het van 0 tot n .

4.1.5 Edge term

In vele gevallen liggen de curves op de rand tussen voorgrond en achtergrond. Er wordt dan gebruik gemaakt van een eenzijdige band rond de curve voor het tracken van pixels. Een probleem, dat hier kan voorkomen, is dat de curve de neiging heeft om te bewegen naar de binnenkant van het voorgrond object tijdens de optimalisatie. Om dit te vermijden wordt er gebruik gemaakt van een edge term.

De edge term E_G maakt gebruik van de image gradiënt van de frame. Wiskundig toont een gradiënt de grootte en richting aan van een verandering in een functie. Bij een image betekent dit dat een gradiënt op een punt in de richting wijst waar zich de grootste overgang van kleurwaarde bevindt, namelijk bij de overgang van achtergrond en voorgrond. De grootte en de snelheid waarmee de verandering gebeurt, van een gradiënt, noemen we de magnitude van de gradiënt. Deze hebben de vorm van een vector. Het zijn de gradiënt magnitudes die gebruikt worden in de verdere berekeningen.

De meest gebruikte methode voor het vinden van de gradiënt image is de Sobel operator. De gradiënt van een image wordt bekomen door gebruik te maken van een convolutie filter. De nieuwe waarde van een pixel, de gradiënt, wordt berekend door het gemiddelde van zijn gewogen omliggende pixels. In feite is een gradiënt image de afgeleide van locale kleurwaardes. De gewichten voor de convolutie filter worden zodanig genomen dat deze de

afgeleide van die gezochte pixel, de gradiënt, benaderd. Er moeten hiervoor twee convolutie filters worden voorzien, een afgeleide in de horizontale richting (G_x) en een afgeleide in de verticale richting (G_y). De uiteindelijke gradient G wordt bekomen door het samenvoegen van G_x en G_y .

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad \text{en} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (4.5)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (4.6)$$

De edge term meet de magnitude van de image gradiënt van punten op de curve. Omdat er gezocht wordt naar het minimum van de energie term moet de gradiënt afgetrokken worden van de maximale mogelijke gradiënt, K . De term die dan geminimaliseerd wordt is $G'(p) = K - G(p)$. Het is gewenst om curves die in de keyframe een grote gradiënt hebben te benadrukken door ze een groter gewicht te geven in de optimalisatie. Het gewicht van een curve wordt gegeven door de gradiënt $G'(p)$ te normaliseren door het minimum, M , van de gradiënt op dat punt in de twee keyframes. De uiteindelijke term E_G is de som van al de genormaliseerde gradiënten.

$$E_G = \sum_{i,t} \left(\frac{G'(c_t(s_i))}{M_i} \right) \quad (4.7)$$

Met $M_i = \min(G'(c_{t_a}(s_i)), G'(c_{t_b}(s_i)))$.

4.2 Conclusie

De energie functies zijn de kern van de keyframe gebaseerde tracking. De energie functie voor deze techniek bestaat uit verschillende termen: de lengte term, de curvatie term, de snelheids term, de image term en de edge term. De drie eerst termen zorgen dat de vorm van de curve behouden wordt. De twee laatste termen zorgen ervoor dat de curve de beweging van het object waartoe het behoort blijft volgen doorheen de video. In het volgende hoofdstuk worden technieken gegeven voor het minimaliseren van de energie functie.

Hoofdstuk 5

Optimalisatie

Segmentatie methodes zoals key-framegebaseerde tracking [2] maken gebruik van energie functies. Een energie functie vergelijkt de huidige waarde van een object met een nieuwe voorgestelde waarde. Via deze weg geven energie functies aan welke de optimale nieuwe waarden zijn voor de parameters van het object. Dit zijn de parameters die de energie functie zo klein mogelijk houden, anders gezegd, die de energie functie optimaliseren.

Uit de opbouw van de energie functie van key-frame gebaseerde tracking valt op, dat elke term ongeveer is opgebouwd als een niet lineaire kleinste kwadraten probleem. Immers er worden telkens twee waarde vergeleken. De energie term is laag wanneer het kwadratisch verschil tussen deze twee waardes klein is. Bij de eenvoudigste vorm van een kleinste kwadraten probleem wordt er gezocht naar een ideale rechte door een puntenwolk, die het verschil tussen de punten en de rechte minimaliseren. Dit probleem is te vergelijken met het probleem van het optimaliseren van de energie functie.

Voor niet lineaire kleinste kwadraten problemen zijn er veel optimalisatie methodes bekend. Door de energie functie als een niet lineaire kleinste kwadraten probleem te beschouwen, kunnen deze methodes gebruikt worden om de energie functie te optimaliseren. De energie functie E kan herschreven worden als $E = \sum_k w_k \|f_k(x)\|^2$, met $f_k(x)$ een energie term met zijn behorende gewicht w_k , en met x de vector van onbekende waardes in de formule. De veranderlijke waarde van k stelt de verschillende energie termen voor uit het vorige hoofdstuk. Door de energie functie op deze manier te herschrijven, krijgt het de algemene vorm van een niet lineaire kleinste kwadraten probleem.

In het verdere verloop van dit hoofdstuk wordt beschreven hoe een niet lineaire kleinste kwadraten probleem kan worden geoptimaliseerd. Deze technieken kunnen dan toegepast worden op de herschreven energie functie. De optimalisatie geeft op het einde van zijn algoritme de ideale nieuwe waarde van het object. Met andere woorden het zoekt naar de ideale aangepaste waarde van de parameters van het object, waardoor de energie functie wordt geminimaliseerd.

5.1 Niet lineaire kleinste kwadraten probleem

Kleinste kwadraten worden gebruikt voor het vinden van een waarde, om toe te kennen aan onbekende parameters in een model gebaseerd op geobserveerde waardes. Een simpel voorbeeld hiervan is het zoeken van een rechte doorheen een puntenwolk. De geobserveerde waardes zijn dan de gekende punten van de wolk en de gezochte waardes zijn de parameters van de rechte. Er wordt hier gebruik gemaakt van het kwadratisch verschil tussen de geobserveerde waarden en de gezochte waarden (in de vorm van de functie f met onbekende parameters). Kleinste kwadraten problemen worden opgelost door het vinden van de onbekende parameters die het kwadratisch verschil minimaliseren.

Het voorbeeld dat werd aangehaald is een lineaire kleinste kwadraten probleem. De energie functie is echter niet lineair in zijn parameters. Hierdoor is de energie functie een niet lineaire kleinste kwadraten probleem. Niet lineaire kleinste kwadraten problemen zijn een uitbreiding van lineaire kleinste kwadraten problemen, doordat zij gebruikt kunnen worden in een veel ruimere en meer algemene klasse van functies. Anders dan lineaire kleinste kwadraten, hebben niet lineaire kleinste kwadraten problemen weinig beperkingen op welke manier de parameters gebruikt kunnen worden in de functie. De manier waarop de onbekende parameters geschat worden blijft conceptueel hetzelfde.

$$S = \sum_{i=1}^n (y_i - f(x_i))^2 \quad (5.1)$$

Een model voldoet aan de algemene vorm van een niet lineaire kleinste kwadraten probleem (formule 5.1) indien het de volgende eigenschappen heeft.

- De functie f is niet lineair ten opzichte van de onbekende parameters.
- Het kleinste kwadraten worden gebruikt voor het vinden van deze onbekende parameters.

Uit de herschreven energie functie blijkt dat deze voldoet aan de algemene vorm van een niet lineaire kleinste kwadraten probleem.

De functie is de sommatie van energie termen en de onbekende parameters zijn de sample points van de keyframes en alle tussenliggende frames. Voor de tussenliggende frames moeten er startwaardes voorzien worden voor de onbekende parameters, de sample points. Let wel dat in de energie functie er niet aan curve fitting gedaan wordt, zoals in het lineaire voorbeeld. In de energie functie wordt gezocht naar de onbekende sample points van de tussenliggende curve die de verschillende energie termen, zoals afstand tussen twee opeenvolgende punten, minimaliseert.

5.2 Optimalisatie

Een veel gebruikte techniek voor het minimaliseren, oftewel optimalisatie, van een niet lineaire kleinste kwadraten probleem, is het Levenberg-Maquardt [22] [27] algoritme. In

Levenberg-Marquardt beschouwen we het probleem als het oplossen van niet lineaire vergelijkingen:

$$r_i(x) = 0, i = 1, \dots, m \quad (5.2)$$

Met $r_i(x)$ als nonlineaire functies gedefinieerd van $\mathbb{R}^n$ naar $\mathbb{R}$ en $x = (x_1, x_2, \dots, x_n)$ als vector. Er wordt verondersteld dat $m = n$. De $r_i(x)$ zijn kleinste kwadraten problemen, dit betekent dat de functies kwadratische verschillen berekenen, deze worden in het algoritme het residu genoemd. Om deze vergelijkingen op te lossen worden de kwadratische verschillen geminimaliseerd.

$$\min \|F(x)\| \quad (5.3)$$

Waarin $F(x)$ een vector functie is met component i van $F(x)$ gelijk is aan $r_i(x)$. $F(x) = (r_1(x), r_2(x), \dots, r_m(x))$ is gedefinieerd van $\mathbb{R}^n \rightarrow \mathbb{R}^m$.

Het Levenberg-Marquardt algoritme is een iteratieve procedure. Dit betekent dat, beginnend van een initiële waarde, de parameters telkens vervangen worden door verbeterde waarden totdat er geen verbetering meer mogelijk is en dus de optimale waarde is gevonden. De manier waarop de parameters vernieuwd worden wordt de update regel genoemd.

Om $F(x)$ te kunnen minimaliseren moet er een initiële waarde gekozen worden voor de vector x . In vele gevallen voldoet het om een standaard vector te kiezen zoals $x = (1, 1, \dots, 1)$. In andere gevallen geeft het algoritme pas een goed resultaat indien de initiële vector al dicht in de buurt ligt van de uiteindelijke oplossing.

5.3 Gradient descent

Het Levenberg-Marquardt algoritme kan gezien worden als een combinatie van gradiënt descent en Gauss-Newton iteratie. Gradiënt descent is de simpelste en meest intuïtieve methode voor het vinden van het minimum in een functie. Bij het iteratief zoeken van een lokaal minimum van een functie kijkt gradiënt descent in de richting van het negatieve van de gradiënt van de functie op het huidige punt. Indien er gekeken wordt in de richting van de gradiënt, niet de negatieve, vindt men een lokaal maximum. Gradiënt descent vernieuwt de parameters door het toevoegen van het negatieve, van de aangepaste gradiënt, in elke stap van de iteratie tot het minimum gevonden is. Men beschouwt het minimum gevonden, indien er geen (voldoende) verbetering meer gevonden wordt.

Neem functie $f(x)$ die gedefinieerd is in punt a . $f(x)$ neemt het snelste af in de richting van de negatieve gradient van f op punt x_i , $\nabla f(x_i)$. De updateregels van gradiënt descent is:

$$x_{i+1} = x_i - \lambda \nabla f(x_i) \quad (5.4)$$

De grootte van de stappen, λ , mag aangepast worden in elke iteratie. De iteratie stopt wanneer blijkt dat verdere iteratie nauwelijks voor verbetering zorgt van de parameter. Met andere woorden wanneer $\lambda \nabla f(x_i) \approx 0$.

5.3.1 Problemen

Gradiënt descent heeft last van verschillende convergentie problemen. Voor het verkrijgen van een snelle convergentie tijdens het zoeken van het minimum zou het gewenst zijn dat gebieden met weinig verandering in waarde snel worden afgehandeld en dat op gebieden met veel verandering elke verandering bekeken wordt. Dit slaat neer dat er grote stappen genomen worden in de richting van de gradiënt op locaties waar er zich weinig verandering afspeelt, met andere woorden waar de gradiënt klein is, en kleine stappen genomen worden indien er veel veranderingen zijn, waar de gradiënt groot is. Bij gradiënt descent gebeurt juist het omgekeerde.

Een ander probleem bij gradiënt descent, toont zichzelf wanneer er zich lange smalle valleien voordoen in de functie. In zo'n gevallen heeft gradiënt descent veel iteraties nodig om het minimum te bereiken.

5.4 Newton methode

Door het gebruiken van curvatie informatie kan de convergentie versneld worden. Informatie over de curvatie kan gevonden worden in de tweede afgeleide van een functie. De Newton methode is een optimalisatie techniek die gebruik maakt van de tweede afgeleide voor oplossen van $\nabla f(x_i) = 0$. Voor de Newton methode wordt de gradiënt van f uitgebreid met behulp van een Taylor reeks rond het huidige punt x_0 . Een Taylor reeks is representatie van een functie door een oneindige som van de waarde van afgeleiden rond een punt.

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n \quad (5.5)$$

Voor de gradiënt van f , $\nabla f(x)$, geeft dit:

$$\nabla f(x) = \nabla f(x_0) + (x - x_0) \nabla^2 f(x_0) + \text{hogere orde termen van } (x - x_0) \quad (5.6)$$

Voor de Newton methode wordt er verondersteld dat f kwadratisch is rond x_0 . Hierdoor worden de hogere termen genegeerd. Voor het oplossen van het onbepikt minimum kan in de Taylor reeks $\nabla f(x)$ gelijk aan 0 worden gezet. Onbepikt (unconstrained) betekent dat het minimum elke waarde mag zijn. Hieruit wordt de updateregels van de Newton methode gevonden voor het vinden van een minimum:

$$x_{i+1} = x_i - (\nabla^2 f(x_i))^{-1} \nabla f(x_i) \quad (5.7)$$

Waar x_0 vervangen wordt door x_i en x door x_{i+1} , zoals bij gradient descent stopt de iteratie indien $\nabla f(x_i) \approx 0$.

5.5 Gauss-Newton methode

Bij de Newton methode wordt veronderstelt dat de functie f kwadratisch is. Hierdoor moet de tweede afgeleide, die gebruikt wordt in de Taylor reeks, niet exact berekend worden. Er kan gebruik gemaakt worden van een benadering met behulp van de Jacobiaan matrix.

De Jacobiaan matrix is de matrix van alle eerste afgeleide van een vector functie. Veronderstel $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ met $F(x) = (r_1(x), r_2(x), \dots, r_m(x))$ en $x = (x_1, x_2, \dots, x_n)$. De eerste afgeleide van de functie F kan geschreven worden met de hulp van de Jacobiaan matrix J van r ten opzichte van x als $J(x) = \frac{\delta r_j}{\delta x_i}$, $1 \leq j \leq m$, $1 \leq i \leq n$.

$$\nabla r(x) = \begin{bmatrix} \frac{\delta r_1}{\delta x_1} & \frac{\delta r_2}{\delta x_1} & \dots & \frac{\delta r_m}{\delta x_1} \\ \frac{\delta r_1}{\delta x_2} & \frac{\delta r_2}{\delta x_2} & \dots & \frac{\delta r_m}{\delta x_2} \\ \dots & \dots & \dots & \dots \\ \frac{\delta r_1}{\delta x_n} & \frac{\delta r_2}{\delta x_n} & \dots & \frac{\delta r_m}{\delta x_n} \end{bmatrix} \quad (5.8)$$

De eerste afgeleide van $F(x)$, de gradiënt, wordt dan gegeven door:

$$\nabla F(x) = \sum_{j=1}^m F_j(x) \nabla F_j(x) = J(x)^T F(x) \quad (5.9)$$

Voor de tweede afgeleide moet de Hessiaan matrix berekend worden. De Hessiaan matrix H is de matrix met alle tweede afgeleiden van een vector functie. Dit geeft $H(x) = \nabla^2 r_j(x)$, $1 \leq j \leq m$, met $(\nabla^2 r(x))_{ik} = \frac{\delta^2 r}{(\delta x_i \delta x_k)}$, $1 \leq i \leq n$, $1 \leq k \leq n$.

$$\nabla^2 r(x) = \begin{bmatrix} \nabla^2 r_1(x) \\ \nabla^2 r_2(x) \\ \dots \\ \nabla^2 r_m(x) \end{bmatrix} \quad (5.10)$$

$$\nabla^2 r_i(x) = \begin{bmatrix} \frac{\delta^2 r_i}{(\delta x_1 \delta x_1)} & \frac{\delta^2 r_i}{(\delta x_1 \delta x_2)} & \dots & \frac{\delta^2 r_i}{(\delta x_1 \delta x_n)} \\ \frac{\delta^2 r_i}{(\delta x_2 \delta x_1)} & \frac{\delta^2 r_i}{(\delta x_2 \delta x_2)} & \dots & \frac{\delta^2 r_i}{(\delta x_2 \delta x_n)} \\ \dots & \dots & \dots & \dots \\ \frac{\delta^2 r_i}{(\delta x_n \delta x_1)} & \frac{\delta^2 r_i}{(\delta x_n \delta x_2)} & \dots & \frac{\delta^2 r_i}{(\delta x_n \delta x_n)} \end{bmatrix} \quad (5.11)$$

De tweede afgeleide van $F(x)$ wordt dan gegeven door:

$$\nabla^2 F(x) = J(x)^T F(x) + \sum_{j=1}^m F_j(x) \nabla^2 F_j(x) = J(x)^T J(x) + H(x) F(x) \quad (5.12)$$

Door de eigenschappen van kleinste kwadraten problemen kan de Hessiaan ($\nabla^2 F(x)$) vrij makkelijk bekomen worden, gegeven de Jacobiaan, indien de $\nabla^2 r_j(x)$ klein zijn of indien de residu ($r_j(x)$) op zichzelf klein zijn. De Hessiaan in zo'n geval wordt dan:

$$\nabla^2 F(x) = J(x)^T J(x) \quad (5.13)$$

De Gauss-Newton methode is een aangepaste Newton methode die gebruik maakt van deze benadering. De updateregels van Gauss-Newton is dan:

$$x_{i+1} = x_i - (J(x_i)^T J(x_i))^{-1} J(x_i)^T F(x_i) \quad (5.14)$$

Het grote voordeel van de Gauss-Newton methode is dat deze een snelle convergentie heeft bij het zoeken van het minimum. Echter de snelheid van de convergentie hangt sterk af van de startwaarde van het algoritme.

5.5.1 De inverse van de Hessiaan

Het kan voorkomen dat de Hessiaan dicht aanleunt bij de vorm van een niet-inverteerbare matrix. Dit heeft als gevolg dat de inverse van de Hessiaan numeriek onstabiel is. Er zijn vele opties om dit te verhelpen. Een veel gebruikte oplossing is het toevoegen van een correctie matrix B_i aan de Hessiaan zodat de som $H(x_i) + B_i$ positief-definiet wordt.

Een positief-definiëte matrix heeft de eigenschap dat hij altijd een inverse heeft. Een $n \times n$ matrix M is positief definit, als deze voldoet aan één van vijf gedefinieerde eigenschappen van een positief-definiëte matrix. De eigenschappen zijn equivalent. Dit betekent dat als de matrix aan één eigenschap voldoet, hij aan ze allemaal voldoet. Er worden hier twee gegeven:

- $\forall$ niet-nul vectoren $z \in C^n$ geldt, $z^* M z > 0$ (z^* is de conjugate transpose)
- Alle eigenwaarde van M zijn positief.

Een methode om de Hessiaan positief-definiëte te maken is door eerst de Hessiaan te diagonaliseren. De correctie matrix B_i wordt dan zo gekozen dat $H(x_i) + B_i$ dezelfde eigenvectoren heeft als de oude H , maar elke negatieve eigenwaarde wordt vervangen door een waarde groter dan 0. Het is vereist voor de optimalisatie methodes dat de Hessiaan positief-definiëte is.

5.6 Levenberg methode

De optimalisatie methode, voorgesteld door Levenberg [22], is een techniek die gradiënt descent en de Gauss-Newton methode combineert. Hierdoor bevat Levenberg de voordelen van beide technieken, een snelle convergentie van de Gauss-Newton methode en de robuustheid van gradiënt descent, dat toelaat dat de initiële waarde van de iteratie zich ver van de

uiteindelijk waarde mogen bevinden. De updateregel van Levenberg is een combinatie van de twee vorige technieken:

$$x_{i+1} = x_i - (H + \lambda I)^{-1} \nabla f(x_i) \quad (5.15)$$

Met $H = J(x_i)^T J(x_i)$ de Hessiaan matrix, $\nabla f(x_i)$ de gradiënt die bekomen wordt door $J(x_i)^T F(x_i)$ en I de eenheids matrix. De λ wordt in de updateregel van Levenberg de dampingfactor genoemd. De updateregel gaat als volgt te werk. Indien de reductie, het verschil tussen $f(x_i)$ en $f(x_{i+1})$, afneemt na dat er een update is gedaan, betekent dat de veronderstelling, uit de Gauss-Newton methode, dat f een kwadratische functie is, correct blijkt. Het algoritme vermindert de waarde van λ (meestal met een factor 10), zodat de invloed van de gradiënt descent methode afneemt, waardoor het algoritme dichter bij de Gauss-Newton methode komt te liggen. Wanneer de reductie toeneemt is het gewenst om de meer gradiënt te volgen. De dampingfactor λ wordt dan verhoogd met dezelfde factor, zodat de gradiënt descent methode meer invloed krijgt ten opzichte van de Gauss-Newton methode.

Het Levenberg algoritme gaat als volgt:

1. Bereken de parameter vector volgens de update-regel.
2. Evalueer de reductie tussen de oude en de nieuwe parameter vector.
3. Indien de reductie vermeerderd is als gevolg van de update:
 - Reset de parameter vector naar zijn vorige waarde.
 - Verminder λ met een factor 10.
4. Indien de reductie verminderd is als gevolg van de update:
 - Behoud de waarde van de nieuwe parameter vector
 - Vermeerder λ met een factor 10.

Het algoritme stopt indien blijkt dat verdere iteraties geen bijdragen leveren voor het zoeken naar het minimum. Anders gezegd, wanneer de waarde die men toevoegt aan de oude parameter vector of de reductie onbeduidend klein is.

5.7 Levenberg-Marquardt methode

Het Levenberg algoritme heeft als nadeel dat wanneer de waarde van λ groot is, de Hessiaan matrix geen invloed heeft op de nieuwe parameter vector. Hierdoor gaat de informatie van de tweede afgeleide, de curvatie, verloren. Om dit tegen te gaan wordt de identiteitsmatrix vervangen door de diagonaal matrix van de Hessiaan in de updateregel. Deze aanpassing werd voorgesteld door Marquardt [27] en geeft de Levenberg-Marquardt updateregel:

$$x_{i+1} = x_i - (H + \lambda \text{diag}[H])^{-1} \nabla f(x_i). \quad (5.16)$$

Door gebruik te maken van Levenberg-Marquardt, bij het vernieuwen van de parameters, wordt de curvatie van f altijd in rekening gebracht. Dit zorgt er voor dat er grote stappen genomen worden, wanneer de gradiënt nauwelijks verandert, en kleine stappen indien deze vele veranderingen ondergaat.

Eerder werd er opgelegd dat de Hessiaan positief-definiet moet zijn. In het Levenberg-Marquardt algoritme wordt de Hessiaan H vervangen door $H + \lambda I$, met λ de dampingfactor die de stapgrootte (step-size) manipuleert. Indien de waarde van λ groot genoeg is wordt de Hessiaan positief-definiet. Echter indien λ te groot is wordt de informatie van H genegeerd. In zo'n geval slaat Levenberg terug op zijn gradiënt descent gedeelte, wat zorgt voor trage convergentie.

5.8 Trust-region methode

In de Levenberg-Marquardt methode wordt er beslist in welke richting de parameters worden verplaatst aan de hand van de gradiënt. Hoe ver de parameters zich verplaatsen tijdens de iteraties wordt beslist door de grootte van de stap α , de step-size genoemd. De step-size wordt in het algoritme in elke iteratie, i , rechtstreeks gemanipuleerd. Neem d_i als de richting voor de parameter verplaatsing, bijvoorbeeld de gradiënt, en α_i als de step-size. De volgende stap wordt dan gegeven door $x_{i+1} = x_i - \alpha_i d_i$. Een sub-probleem van deze methode is het bepalen van de step-size α_i . De step-size moet zo gekozen worden dat $f(x_i - \alpha_i d_i)$ wordt geminimaliseerd, en dus het minimum vindt, voor elke mogelijke step-size.

$$\min f(x_i - \alpha_i d_i) \forall \alpha_i > 0 \quad (5.17)$$

Het Levenberg-Marquardt algoritme bereikt dit door in elke iteratie de step-size te manipuleren door middel van de dampingfactor. Bij elke iteratie wordt aan de hand van de reductie beslist hoe de step-size zal worden aangepast voor gebruik in de volgende iteratie. Echter het rechtstreeks manipuleren van de step-size met behulp van een dampingfactor kan fatale gevolgen hebben, omdat de dampingfactor niet gebaseerd is op informatie van de functie.

In een variant van het Levenberg-Marquardt algoritme, de Trust-region methode [11] [14] [10] [35] [34], wordt de step-size niet rechtstreeks gemanipuleerd, maar wordt er een eindig gebied beschouwd waarin zich de volgende stap bevindt. Dit kan gezien worden als het opleggen van een maximale waarde voor de step-size. Alle mogelijke locaties van x_{i+1} bevinden zich dan in een gebied, Δ , rond de parameter x_i . Dit gebied wordt de trust-region genoemd. In plaats van de step-size past het algoritme voor elke iteratie de grootte van de trust-region, Δ_i , aan. Deze methode wordt het trust-region algoritme genoemd.

In het trust-region algoritme wordt er een model, m_i , geconstrueerd die de functie f benaderd in een eindig gebied rond x_i . Dit gebied waar m_i een goede benadering is van f is de trust-region, Δ_i . Het model m_i wordt bekomen door een Taylor expansie te doen van

f rond x_i . Zoals bij de Newton methode wordt er verondersteld dat f kwadratisch is rond het punt x_i . Opgelet, hier wordt Taylor toegepast op de functie f zelf, niet op zijn gradiënt zoals bij de Newton methode.

$$m_i = f(x_i) + \nabla f(x_i)(x_{i+1} - x_i) + \frac{1}{2}H(x_{i+1} - x_i)^2 \quad (5.18)$$

Met H de Hessiaan matrix ($\nabla^2 f(x_i)$), of een benadering, beschouwt p het verschil tussen de huidige waarde en de volgende waarde, $p = (x_{i+1} - x_i)$. Met andere woorden, p is de verplaatsing die de parameter ondergaat tijdens de iteratie $x_{i+1} = x_i + p$, de step (step-size en richting van de verplaatsing).

$$m_i = f(x_i) + \nabla f(x_i)p + \frac{1}{2}Hp^2 \quad (5.19)$$

Een subprobleem is nu het vinden van welke step p er moet genomen worden, tijdens de iteratie $x^{i+1} = x_i + p$, zodat het minimum van m_i gevonden wordt.

$$\min_{\|p\| \leq \Delta} (f(x_i) + \nabla f(x_i)p + \frac{1}{2}Hp^2) \quad (5.20)$$

Het trust-region algoritme is een sequentie van iteraties waarin telkens het minimum gezocht wordt van het model m_i dat een kwadratische benadering is van f in een eindig gebied rond het punt x_i . In elke iteratie wordt de grootte van het gebied, Δ , voor de volgende iteratie beslist. De heuristiek voor het vernieuwen van de grootte van de trust-region hangt af van de verhouding tussen de waarde van f en de voorspelde verandering. Het verschil tussen $m_i(0)$ en $m_i(x_i)$ is de verandering die het resultaat, de reductie, van de benaderingsfunctie ondergaat tussen de iteraties. Deze waarde wordt gebruikt als een voorspelling hoe de eigenlijk functie verandert. De eigenlijke reductie wordt gegeven door $f(x_i) - f(x_i + p)$.

$$q_i = (f(x_i) - f(x_i + p)) / (m_i(0) - m_i(x_i)) \quad (5.21)$$

Indien er een goede overeenkomst is tussen de voorspelde waarde en de eigenlijk waarde ($q_i \approx 1$), dan wordt Δ verhoogd. Indien de overeenkomst slecht is, wordt Δ behouden. Als de overeenkomst kleiner is dan een drempel waarde, wordt de stap genegeerd en de oude waarde van de parameter vector behouden. In zo'n geval wordt Δ verlaagd.

Het uiteindelijk algoritme van trust-region is zeer gelijkaardig als dat van de line-search methode. In plaats van de step-size in elke iteratie te manipuleren wordt de grootte van de trust-region aangepast. Het trust-region algoritme ziet er als volgt uit.

1. Vind de step p door het oplossen van het subprobleem.
2. Bereken de parameter vector volgens de updateregels.

3. Evalueer de overeenkomst q_i .
4. Indien $q_i \geq 0.9$:
 - Behoud de waarde van de nieuwe parameter vector.
 - Verdubbel Δ
5. Indien $0.01 \leq q_i \leq 0.9$:
 - Behoud de waarde van de nieuwe parameter vector.
 - Δ blijft onveranderd.
6. Indien $q_i \leq 0.01$:
 - Zet de parameter vector terug naar zijn vorige waarde.
 - Halveer Δ

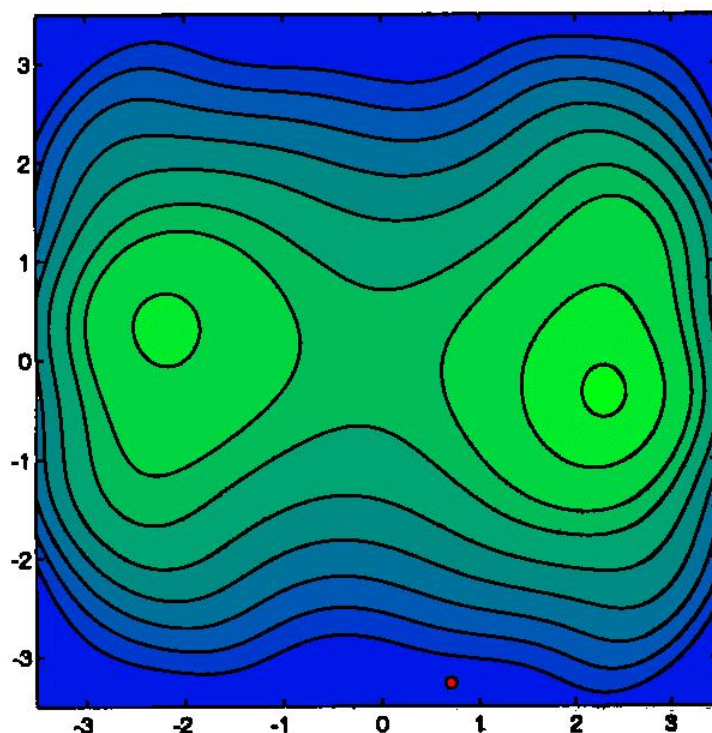
Zoals bij het vorige algoritme stopt trust-region indien blijkt dat verdere iteraties geen bijdragen leveren voor het zoeken naar het minimum.

5.8.1 Voorbeeld van trust-region

Nu volgt er een voorbeeld dat uitlegt hoe trust-region in principe werkt. Dit voorbeeld komt uit "Thesis on Continuous optimization" van Frank Vanden Berghen [3]. Gegeven volgende functie:

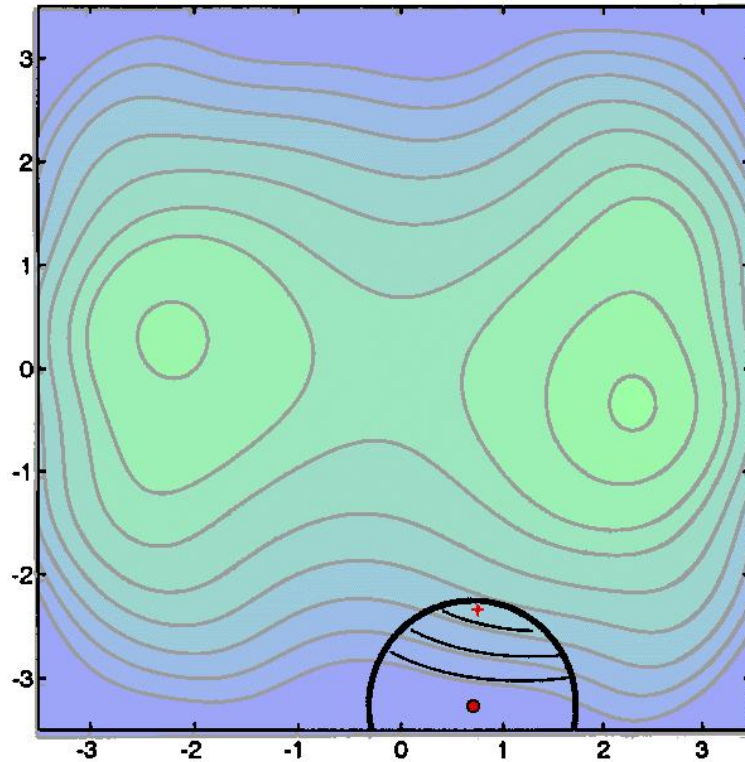
$$f(x_1, x_2) = -10x_1^2 + 10x_2^2 + 4\sin(x_1x_2) - 2x_1 + x_1^4 \quad (5.22)$$

Het huidige punt x_k wordt in de figuren voorgesteld als een rood punt. De figuur 5.1 geeft $f(x_1, x_2)$ voor verschillende waarden van het punt x_k .



Figuur 5.1: De begin situatie. De huidige waarde van x wordt aangegeven door een rood punt.

Vervolgens wordt de benadering $Q(x)$ van $f(x)$ rond het punt x_k berekend. $Q(x)$ wordt voorgesteld door de dikke zwarte lijnen in de figuur. Er wordt een trust-region Δ_k beschouwd rond x_k voorgesteld als een zwarte cirkel. Nu wordt het minimum van $Q(x)$ gezocht in de trust-region. Het minimum x_{k+1} is het rode kruisje in figuur 5.2.



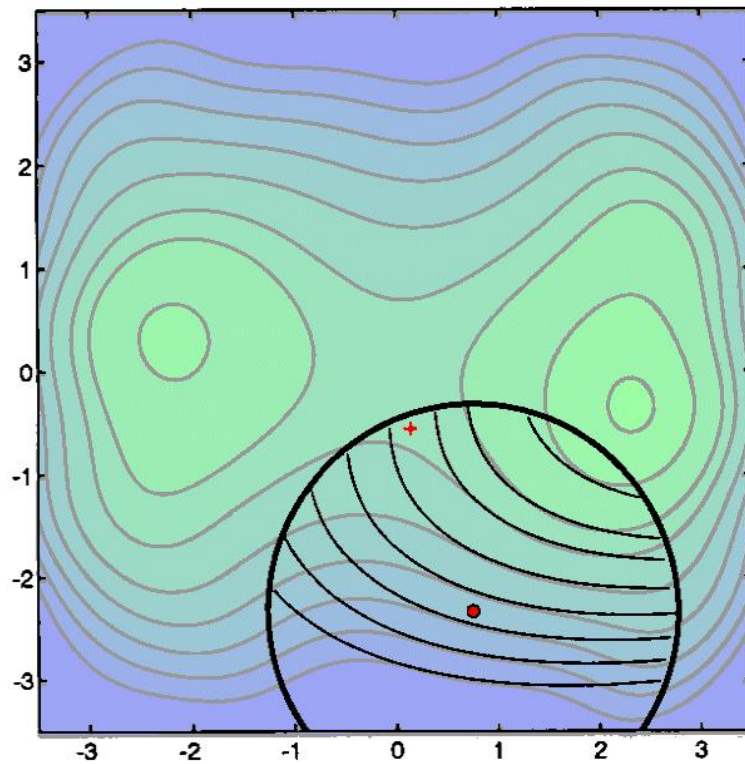
Figuur 5.2: In de trust region (cirkel) wordt gezocht naar het minimum. Binnen de trust region wordt de functie benaderd. Het gevonden minimum wordt aangegeven door het kruisje.

Er geldt dat $f(x_{k+1}) < f(x_k)$. $Q(x)$ is, dus een goede benadering van $f(x)$. Het huidige punt verplaatst zich naar x_{k+1} . Sinds $Q(x)$ een goede benadering is, wordt de radius van de trust-region vergroot. Vanuit het nieuwe huidige punt wordt de nieuwe benadering $Q(x)$ berekend. De volgende iteratie begint (figuur 5.3).

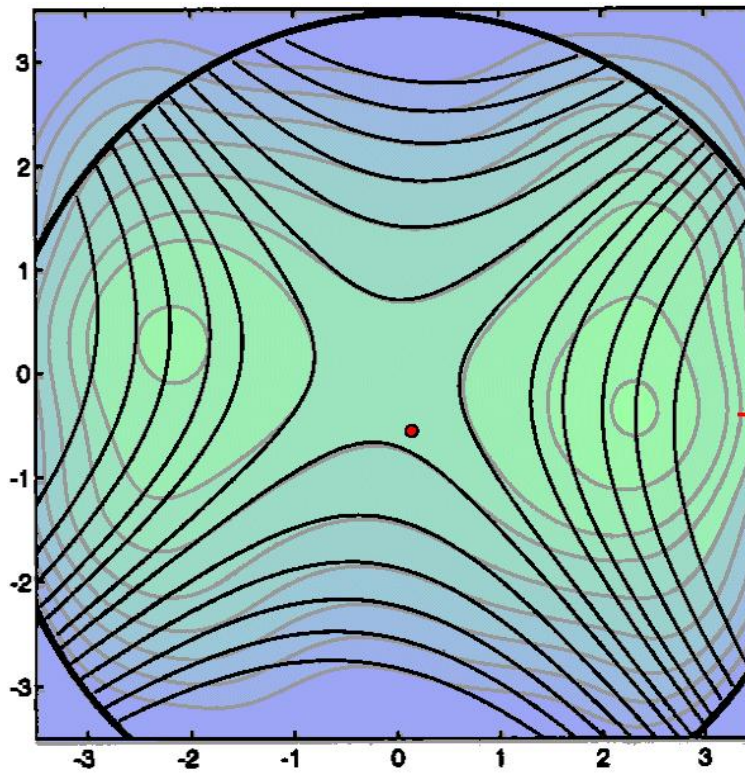
Er geldt dat $f(x_{k+1}) < f(x_k)$. Het huidige punt wordt verplaatst. De nieuwe benaderingen worden berekend. De radius wordt vergroot.

Deze iteratie (figuur 5.4) faalt omdat $f(x_{k+1}) > f(x_k)$. $Q(x)$ geen correcte benadering is van $f(x)$. De radius van de trust-region wordt verkleind. Het huidige punt blijft ongewijzigd.

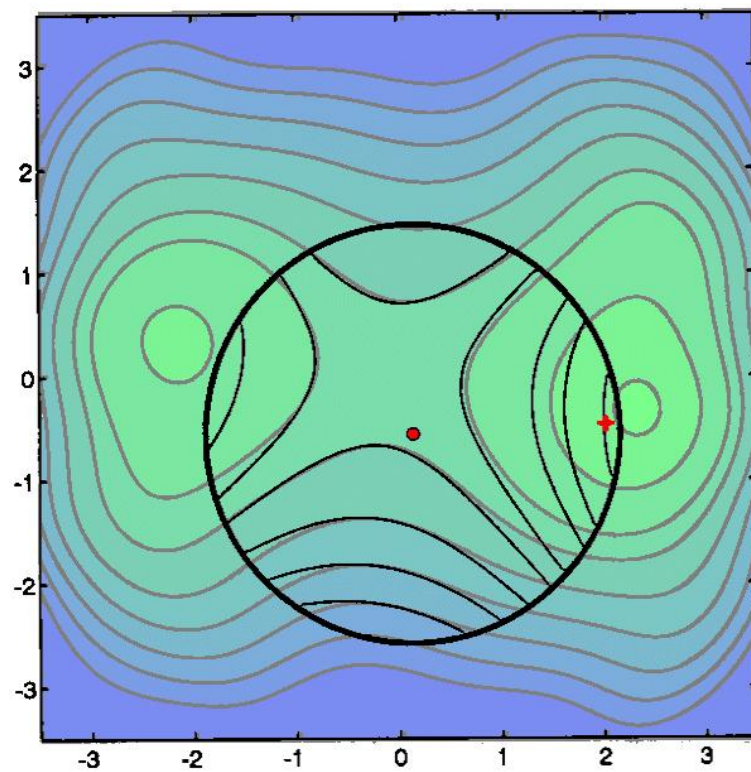
Dit is een gedeeltelijk succes (figuur 5.5). $f(x_{k+1}) < f(x_k)$ geldt, maar de reductie is kleiner als verwacht. De radius van de trust-region wordt nu behouden. De huidige positie van x_k wordt gewijzigd. De twee volgende figuren tonen dat de twee volgende iteraties (figuur 5.6, figuur 5.7) weinig veranderen aan de positie van x_k . Het algoritme stopt wanneer het verschil tussen x_k en x_{k+1} kleiner is dan een bepaalde waarde.



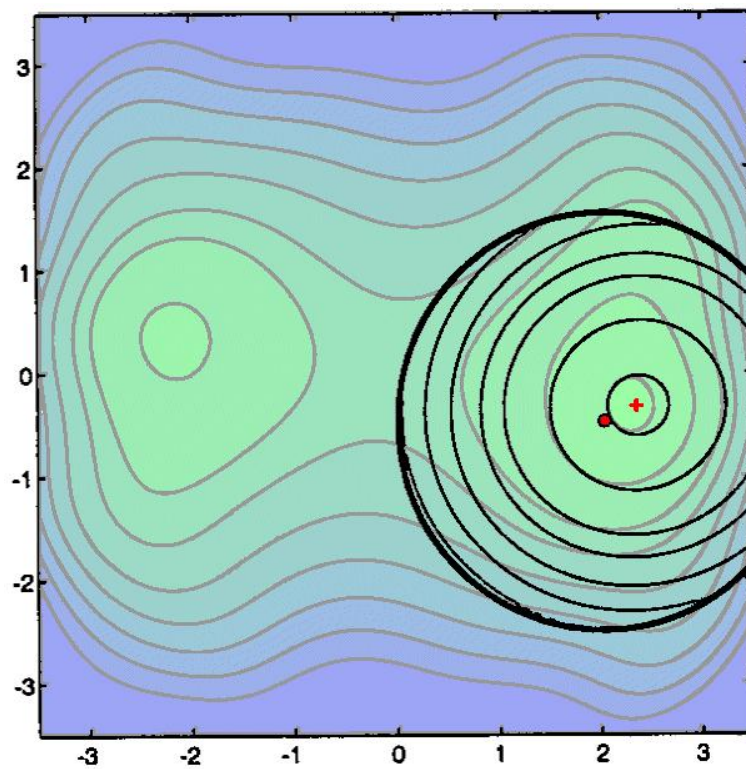
Figuur 5.3: Er wordt opnieuw gezocht naar het minimum met een grotere trust region.



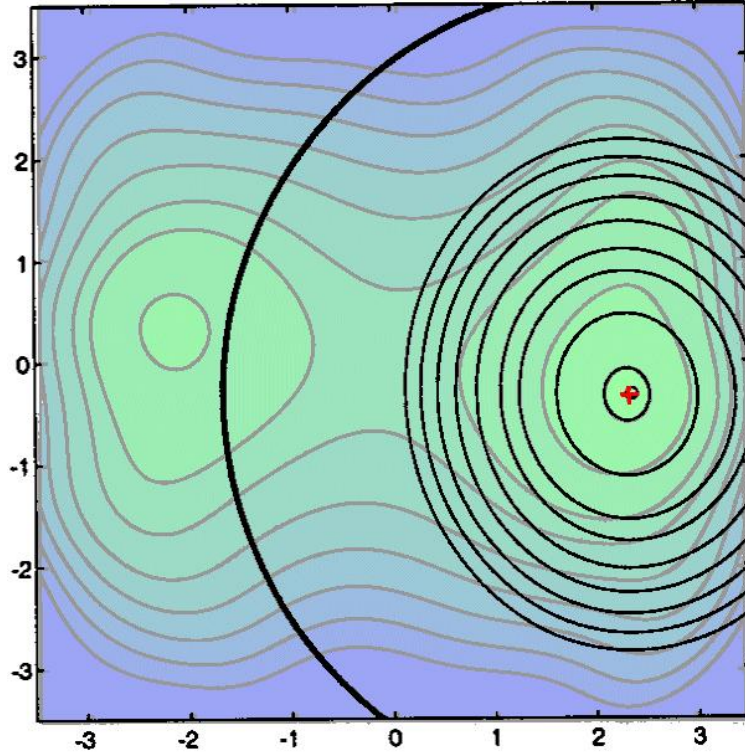
Figuur 5.4: Het nieuwe punt wordt aanvaard en wordt het nieuwe huidige punt. Er wordt verder gezocht met een vergrote trust region.



Figuur 5.5: Omdat de functie niet goed benaderd wordt binnen de trust region wordt de radius verkleind. Er wordt een nieuw minimum gevonden.



Figuur 5.6: In deze iteratie wordt de positie van het minimum nauwelijks veranderd.



Figuur 5.7: Zelfs wanneer de radius vergroot wordt van de trust region wordt het minimum amper gewijzigd.

5.8.2 Trust-region subprobleem

Het subprobleem is het meest essentiële onderdeel van het trust-region algoritme. In elke iteratie van het trust-region algoritme is het vereist om een oplossing, p , te vinden voor het subprobleem:

$$\min_{p \leq \Delta} (f(x_i) + \nabla f(x_i)p + \frac{1}{2}Hp^2) \quad (5.23)$$

Omdat $f(x_i)$ geen invloed heeft, wordt het achterwege gelaten voor het zoeken van het minimum van p . Voor het verder verloop wordt er gebruik gemaakt van de benadering voorgesteld in de Gauss-Newton methode voor de gradient en de Hessiaan. Meer bepaald, $\nabla f(x_i) = J(x_i)^T F(x_i)$ en de Hessiaan $H = J(x_i)^T J(x_i)$. Het subprobleem wordt herschreven als:

$$\min_p (J(x_i)^T F(x_i)p + \frac{1}{2}J(x_i)^T J(x_i)p^2) \quad (5.24)$$

$$p \leq \Delta_i \quad (5.25)$$

Het oplossen van dit subprobleem is veel besproken onderwerp. Een bekend lemma [46] voor de oplossing is als volgt:

Lemma 5.8.1

Een vector $p_i \in \mathbb{R}^n$ is een oplossing voor het probleem:

$$\min_p (J_i^T F_i p + \frac{1}{2} J_i^T J_i p^2) \quad (5.26)$$

$$p \leq \Delta_i \quad (5.27)$$

Met $\Delta > 0$, als en alleen als er een $\lambda_i \geq 0$ bestaat zodat:

$$(J_i^T J_i + \lambda_i I) p_i = -J_i^T F_i \quad (5.28)$$

$$\lambda(\Delta_i - p_i) = 0 \quad (5.29)$$

en dat $(J_i^T J_i + \lambda_i I)$ positief-definiet is.

De vereiste $\lambda(\Delta - p_i) = 0$ betekent dat wanneer $p_i < \Delta_i$ dan is λ gelijk aan 0, maar niet omgekeerd. De λ wordt de Lagrange multiplier genoemd. De grote vraag is nu welke waarde λ moet aan nemen zodat p_i een oplossing is voor het subprobleem. Beschouw volgend lemma [46]:

Lemma 5.8.2

Laat vector p_i een oplossing zijn voor (5.26) (5.27), dan bestaat er een unieke $\lambda_i \geq 0$ die voldoet aan (5.28) (5.29).

Als $\|J_i^T F_i\| \leq \Delta_i$, dan:

$$p_i = -J(x_i)^T F(x_i) \quad (5.30)$$

Anders $\lambda_i > 0$ en voldoet:

$$(J_i^T J_i + \lambda_i I)^{-1} J_i^T F_i = \Delta_i \quad (5.31)$$

De Newton methode kan toegepast worden op deze niet lineaire vergelijking om de waarde van λ_i te vinden. Hier wordt de Newton methode niet gebruik als optimalisatie techniek, hoewel gelijkaardig, maar als een numeriek methode voor het vinden van nulpunten, beter bekend als de Newton-Raphson methode.

$$q(\lambda) = \frac{1}{((J_i^T J_i + \lambda_i I)^{-1} J_i^T F_i)} - \frac{1}{(\Delta_i)} = 0 \quad (5.32)$$

De reden waarom de functie $q(\lambda)$ gebruikt wordt en niet de meer simpele vergelijking:

$$((J_i^T J_i + \lambda_i I)^{-1} J_i^T F_i) - (\Delta_i) = 0 \quad (5.33)$$

is omdat de functie $q(\lambda)$ zeer dicht in de buurt komt van een lineaire functie. Hierdoor convergeert de Newton-Raphson methode sneller. Door de Newton-Raphson methode toe te passen op $q(\lambda)$ kan de waarde van λ_i gevonden worden met behulp van het volgende iteratief proces:

$$\lambda_i^0 = 0 \quad (5.34)$$

$$\lambda_i^{k+1} = \lambda_i^k - \frac{(s_i^k)^T ((J_i^T J_i + \lambda_i^k I))^{-1} s_i^k}{\|s_i^k\|^3} \left(\frac{1}{s_i^k} - \frac{1}{\Delta_i} \right) \quad (5.35)$$

Met $s_i^k = ((J_i^T J_i + \lambda_i^k I)^{-1} J_i^T F_i)$.

Doordat $q(\lambda)$ concaaf is, kan er gezien worden dat de sequentie kwadratische convergeert. Uit de concaviteit van $q(\lambda)$ volgt:

$$q(\lambda_i^{k+1}) \geq q(\lambda_i^k) + \nabla q(\lambda_i^{k+1})(\lambda_i^{k+1} - \lambda_i^k) = q(\lambda_i^k)(1 - \nabla q(\lambda_i^{k+1})/\nabla q(\lambda_i^k)) \quad (5.36)$$

Hieruit geldt dat:

$$q(\lambda_i^{k+1}) \geq \frac{1}{2} q(\lambda_i^k) \quad (5.37)$$

of geldt er dat:

$$\nabla q(\lambda_i^{k+1}) \leq \frac{1}{2} \nabla q(\lambda_i^k) \quad (5.38)$$

De ongelijkheid toont aan dat de Newton iteratie een polynomiaal tijd algoritme is.

5.8.3 Approximatie methode

In plaats van het zoeken naar de exacte waarde van de Lagrange multiplier λ bestaan er algoritmes die rechtstreeks een benadering van p berekenen voor het oplossen van het subprobleem. Er zijn algemeen drie methodes: de dog-leg methode [34], de 2-dimensionale search methode [38] en de truncated conjugate gradiënt methode [42]49.

Net zoals bij de Levenberg-Marquardt methode is de dog-leg methode van Powell een combinatie van de Gauss-Newton methode en de gradiënt descent methode. Gegeven $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Op de huidige iteratie x wordt de step met behulp van de Gauss-Newton methode berekend.

$$h_{gn} = -(J(x)^T J(x))^{-1} J(x)^T f(x) \quad (5.39)$$

De richting die genomen wordt met behulp van gradiënt descent is als volgt

$$h_{gd} = -J(x)^T f(x) \quad (5.40)$$

Dit is een richting, niet een step. De grootte van de stap (α) wordt berekend met behulp van een lineaire model.

$$\alpha = -\frac{h_{gd}^T J(x)^T f(x)}{\|J(x)h_{gd}\|^2} = \frac{\|g\|^2}{\|J(x)g\|^2} \quad (5.41)$$

Het punt dat bekomen wordt door deze step (gradiënt descent) wordt het Cauchy punt genoemd. Voor λ wordt in de dog-leg methode de minimale waarde genomen. Er zijn nu twee kandidaten voor de step op huidige iteratie x . Powell stelde volgende strategie voor om de step te kiezen in de trust-region met radius Δ .

Indien de Gauss-Newton step h_{gn} binnen de trust-region ligt, wordt deze gebruikt als de step in de huidige iteratie. Als deze er buiten ligt wordt het Cauchy punt berekend. Er zijn weer twee mogelijkheden. Ligt het Cauchy punt buiten de trust-region, dan wordt de Cauchy step, afgekapt op de rand van de trust-region, als de benaderde oplossing genomen. Indien het Cauchy punt binnen de trust-region ligt, wordt er een lijn getrokken tussen de locatie gegeven door de Gauss-Newton step h_{gn} (die buiten de trust-region ligt) en het Cauchy punt. Het punt waar deze lijn de rand van de trust-region kruist wordt gebruikt als de step in de huidige iteratie.

In de 2-dimensional search methode wordt de objectieve functie geminimaliseerd in de deelruimte die gespannen is tussen de richting uit de gradiënt descent methode en de richting uit de Gauss-Newton methode binnen de trust-region. In deze deelruimte word dan gezocht naar het minimum met behulp van een simpele 2D zoek methode. De 2-dimensional search methode kan gezien worden als een ongedefinieerde dog-leg methode. Immers het minimum gevonden met de dogleg methode bevind zich ook in deze deelruimte. Echter het minimum bij de dogleg methode word gedefinieerd door de oplossingen van de gradiënt descent methode en de Gauss-Newton methode.

De truncated conjugate gradiënt methode bestaat in het toepassen van standaard conjugate gradiënt op de minimalisatie van het subprobleem. De conjugate gradiënt methode is een iteratieve methode voor het zoeken van een lokaal minimum van een lineaire functie. Deze methode kan toegepast worden voor het oplossen van het sub-probleem (5.24) van de trust-region methode. De truncated conjugate gradiënt methode is de aangepaste methode voor het oplossen van het sub-probleem. Zolang de waarde uit de iteraties van de methode er voor zorgen dat de oplossing binnen de trust-region blijft, is de truncated conjugate gradiënt methode identiek aan de standaard conjugate gradient methode. Indien waarde uit de iteratie buiten de trust-region loopt of indien er een negatieve gradiënt wordt berekend, wordt de stap afgekapt op de rand van de trust-region.

5.9 Conclusie

Keyframe gebaseerde tracking maakt gebruik van energie minimalisatie. In dit hoofdstuk werd een overzicht gegeven van verschillende technieken voor minimaliseren/optimaliseren van de energie functie. De belangrijkste techniek is de trust region variante van Levenberg-Marquardt. Voor de implementatie van segmentatie met behulp van keyframe gebaseerde tracking wordt de approximatie methode gebruikt van Powell, de dogleg methode.

Hoofdstuk 6

Implementatie

In dit hoofdstuk worden enkel praktische onderwerpen besproken die worden gebruikt tijdens de implementatie van de semi-automatische video segmentatie. De techniek die werd geïmplementeerd is de Keyframe gebaseerde [2] semi-automatische video segmentatie, die gebruik maakt van energie minimalisatie. Het semi-automatische deel betekent dat de keyframes bij de start van de optimalisatie gegeven worden door de gebruiker en dat de gebruiker fouten handmatig kan corrigeren. De optimalisatie die gebruikt wordt voor het oplossen van de energie term, is Powell's dogleg methode [34].

Eerst wordt er besproken hoe een gebruiker keyframes kan aanmaken in een bepaald frame met behulp van Bézier-curves en welke eigenschappen en beperkingen deze curves hebben. Vervolgens wordt er besproken welke object structuur wordt gebruikt voor deze curves. Als laatste wordt Powell's dogleg methode besproken, die wordt gebruikt als optimalisatie van de energie functies, besproken in hoofdstuk 4.

6.1 Keyframes

De keyframes worden gebruikt als startwaarde voor de optimalisatie. In de keyframes kan de gebruiker de contouren overtekenen van het te volgen object. In de implementatie wordt er gebruik gemaakt van Bézier-curves met C^0 -continuïteit. Hierdoor kan de gebruiker bijna alle mogelijk curves tekenen en kunnen er scherpe hoeken gemaakt worden tussen de curves.

6.1.1 Eerste keyframe

Vooreerst moet uit de sequentie van frames een bepaalde frame gekozen worden, die het begin van de tracking aanduidt. In dit frame worden de contouren van het object, waarvan we de beweging willen volgen, getraceerd door de gebruiker. Dit gebeurt door het tekenen van een verzameling van curves over het object. De gebruiker is niet beperkt tot enkel het traceren van contouren, curves kunnen overal in de frame geplaatst worden. De verzameling van curves vormt de eerste keyframe. De curves worden bepaald door de controle punten van een Bézier-curve. Het eerste keyframe is in feite een aaneenschakeling van verschillende Bézier-curves.

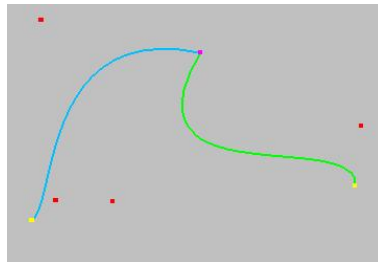
Een Bézier-curve beschrijft een kromme aan de hand van twee of meer controle punten. Het aantal controle punten geeft de graad van de Bézier-curve. Een Bézier-curve van graad n wordt bepaald door $n + 1$ controle punten $P_0, \dots, P_n$ in $\mathbb{R}^3$. Voor het tracen van een keyframe worden er 3de-graad, ofwel cubische Bézier-curves gebruikt.

$$B(t) = P_0(1 - t)^3 + 3P_1t(1 - t)^2 + 3P_2t^2(1 - t) + P_3t^3, t \in [0, 1] \quad (6.1)$$

Er wordt gebruik gemaakt van vier controle punten P_0, P_1, P_2, P_3 . De curve begint bij P_0 en eindigt bij P_3 . De twee overige controle punten bevinden zich niet op de curve, zij geven informatie over de buiging van de curve. $B(t)$ geeft de punten weer die op de curve liggen tussen P_0 en P_3 . De parameter t varieert van 0 tot 1.

Continuïteit

Een keyframe bestaat uit een aaneenschakeling van verschillende Bézier-curves. Elke Bézier-curve in het keyframe is onafhankelijk van elke andere Bézier-curve. Hiervoor moet er bij de overgang van de aaneengeschaalde Bézier-curves een continuïteitsregel worden opgelegd. In het keyframe wordt enkel continuïteit van positie, ofwel C^0 -continuïteit, behouden. Hierdoor kan de gebruiker scherpe hoeken maken. Gegeven twee Bézier-curves van dezelfde graad: $P(t)$, gedefinieerd door $P_0, P_1, \dots, P_n$, en $Q(t)$, gedefinieerd door $Q_0, Q_1, \dots, Q_n$. C^0 -continuïteit wordt bereikt door het zetten van $P(1) = Q(0)$.



Figuur 6.1: Twee Bézier-curve met C^0 -continuïteit

Constraints

De gebruiker kan ook een hard constraint opleggen in de vorm van joints op de eindpunten van verschillende Bézier-curves. Deze zorgen ervoor dat tijdens de optimalisatie de aaneengekoppelde Bézier-curves niet van elkaar wegdrijven.

6.1.2 Tweede keyframe

Voor het tweede keyframe kiest de gebruiker een latere keyframe. De gebruiker heeft de mogelijkheid om de Bézier-curves van het eerste keyframe te kopiëren naar de tweede keyframe. De gekopieerde curves kunnen dan aangepast worden, zodat deze overeenkomen met de eigenschappen van het tweede keyframe. De aanpassingen kunnen operaties zijn zoals rotatie, verplaatsen, scalaren van alle of individuele Bézier-curves en het verplaatsen van individuele controle punten. De gebruiker kan ook het tweede keyframe zelf opbouwen.

Er moet dan voor gezorgd worden dat het tweede keyframe evenveel Bézier-curves bevat als het eerste. Er moet ook rekening gehouden worden met de rangorde van de Bézier-curves. Elke Bézier-curve heeft een rangnummer. De Bézier-curves worden volgens rangnummer vergeleken tijdens de optimalisatie.

6.2 Object structuur

Een Bézier-curve bestaat uit controle punten. Deze worden echter niet gebruikt tijdens de optimalisatie. Elke Bézier-curve wordt geschreven als $c_t(s)$. Met t die aangeeft in welke frame de curve zich bevindt en s als spatiële parameter voor de punten die zich bevinden op de curve gaande van 1 tot N . Het optimalisatie algoritme maakt gebruik van de sample points bekomen met de functie $c_t(s)$.

6.2.1 Joints

Bij het bekomen van de sample points uit $c_t(s)$ moet er gelet worden op de joints tussen verschillende curves. De punten die verschillende curves aan elkaar schakelen moeten verwijzen naar dezelfde variabele. Indien de gebruiker een punt wijzigt, wordt deze vastgelegd. Deze worden niet gewijzigd tijdens de optimalisaties. Dit is het geval bij keyframes.

De energie functies, die worden gebruikt tijdens de optimalisaties, worden geformaliseerd met gebruik van $c_t(s)$. Ze zijn telkens gedefinieerd als zijnde voor één enkele curve, hoewel deze bestaat uit meerdere curves aaneengeschaakeld door joints.

6.2.2 Spline

De keyframes worden tijdens de optimalisatie niet veranderd. De wijzigingen vinden plaats in de tussenliggende frames. Een keyframe bestaat uit een verzameling van curves, een spline genaamd. Voor elke frame tussen de twee keyframes moet een spline van dezelfde grootte worden aangemaakt. Deze start-spline bevat de punten die worden gebruikt als startwaarde waarmee de optimalisatie zijn algoritme begint. De start-spline wordt bekomen door het kopiëren van één van de keyframes of een spline die gelijkaardig is, zoals bijvoorbeeld een interpolatie van de twee keyframes.

Een interpolatie aanpak of het kopiëren van één van de keyframes kan leiden tot foutieve resultaten, indien er weinig verschil is tussen de tussenliggende frames. Het optimalisatie algoritme kan de start-spline van de tussenliggende frames als ideale oplossing beschouwen. Om dit te voorkomen wordt de interpolatie methode aangepast. De aangepaste methode zorgt ervoor dat er kleine verschillen zijn tussen de initiële curves van de tussenliggende frames.

6.3 Powell's dogleg optimalistie methode

Keyframe gebaseerde segmentatie maakt gebruik van een energie functie voor het vinden van de ideale lengte, curvatie, positie van curves in alle tussenliggende frames zodat ze de

beweging volgen van het object waartoe zij behoren doorheen de video. Deze curves worden gevonden door de energie functie te optimaliseren. De optimalisatie methode, gebruikt in de implementatie, is Powell's dogleg methode.

Powell's dogleg methode is een combinatie van de gradiënt descent en de Gauss-Newton methode. De dogleg methode kiest bij elke iteratie één van deze twee methodes aan de hand van de radius van de trust region.

Gegeven $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Op de huidige iteratie x wordt de step met behulp van de Gauss-Newton methode berekend.

$$h_{gn} = -(J(x)^T J(x))^{-1} J(x)^T f(x) \quad (6.2)$$

De gradiënt descent wordt gegeven door:

$$h_{gd} = -g = -J(x)^T f(x) \quad (6.3)$$

De grootte van de step bij gradiënt descent wordt gegeven door:

$$\alpha = -\frac{h_{gd}^T J(x)^T f(x)}{\|J(x)h_{gd}\|^2} = \frac{\|g\|^2}{\|J(x)g\|^2} \quad (6.4)$$

Er zijn nu twee keuzes voor de step vanuit het huidige punt x : $a = \alpha h_{gd}$ indien gradiënt descent en $b = h_{gn}$ indien Gauss-Newton. Bij de dogleg methode wordt de step h_{dl} berekend door de lengte van de steps $\|h_{gn}\|$ en $\|\alpha h_{gd}\|$ te vergelijken met de radius Δ van de trust region. Deze berekening voor h_{dl} verloopt als volgt

$$\begin{aligned} &\text{if } \|h_{gn}\| \leq \Delta \\ &\quad h_{dl} = h_{gn} \\ &\text{else if } \|\alpha h_{gd}\| \geq \Delta \\ &\quad h_{dl} = (\Delta / \|h_{gd}\|) h_{gd} \\ &\text{else} \\ &\quad h_{dl} = \alpha h_{gd} + \beta (h_{gn} - \alpha h_{gd}) \\ &\quad \text{met } \beta \text{ zo gekozen dat } \|h_{dl}\| = \Delta \end{aligned}$$

Wanneer $\|h_{gn}\| > \Delta$ en $\|\alpha h_{gd}\| < \Delta$ moet er een β berekend worden zodat $\|h_{dl}\| = \Delta$. Deze β wordt gevonden met volgend algoritme, met $c = a^T(b - a)$:

$$\begin{aligned} &\text{if } c \leq 0 \\ &\quad \beta = (-c + \sqrt{c^2 + \|b - a\|^2 (\Delta^2 - \|a\|^2)}) / \|b - a\|^2 \\ &\text{else} \\ &\quad \beta = (\Delta^2 - \|a\|^2) / (c + \sqrt{c^2 + \|b - a\|^2 (\Delta^2 - \|a\|^2)}) \end{aligned}$$

Om te controleren hoe goed een iteratie is van de dogleg methode, wordt er gekeken naar de winst verhouding (ρ). De winst verhouding vergelijkt het verschil tussen de waarde van de functie f met de huidige parameters en de voorgestelde parameters met het verschil tussen de waarden van de benadering L van de functie binnen trust region:

$$\rho = (f(x) - f(x + h_{dl})) / (L(0) - L(h_{dl})) \quad (6.5)$$

$L(0)$ is de benadering van de functie binnen de trust-region rond de huidige waarde, er is hier geen step waarde bij de parameters opgeteld. Bij $L(h_{dl})$ is de stepwaarde bij de huidige waarde van de parameters opgeteld, dit geeft de benadering van de functie binnen de trust region rond de voorgestelde parameters. $L(0) - L(h_{dl})$ hangt af van de keuze van h_{dl} . Dit geeft volgend algoritme voor de waarde van $L(0) - L(h_{dl})$:

$$\begin{aligned} &\text{if } h_{dl} = h_{gn} \\ &\quad L(0) - L(h_{dl}) = f(x) \\ &\text{else if } h_{dl} = \frac{-\Delta}{\|g\|} g \\ &\quad L(0) - L(h_{dl}) = \frac{\Delta(2\|\alpha g\| - \Delta)}{2\alpha} \\ &\text{else} \\ &\quad L(0) - L(h_{dl}) = \frac{1}{2}(1 - \beta)^2 \|g\|^2 + \beta(2 - \beta)F(x) \end{aligned}$$

De waarde van ρ wordt gebruikt om de radius Δ van de trust region te controleren. Een grote waarde van ρ geeft aan dat de benadering van de functie binnen de trust region goed is. Hierdoor kan de radius van de trust region vergroot worden en kunnen er grotere stappen genomen worden. Deze steps zullen dan ook dichter aanleunen bij de Gauss-Newton step. Indien de ρ klein is betekent dit dat de benadering niet zo goed is. Dan wordt de radius van de trust region verkleind en worden kleinere stappen genomen, die aanleunen bij de gradient descent step. Indien ρ kleiner is dan nul wordt de step niet opgenomen en worden de parameters niet aangepast en wordt ook de radius van de trust region verkleint.

Het volledige algoritme van de dogleg methode gaat nu als volgt:

```

k = 0; x = x0; Δ = Δ0
found = (||f(x)||∞ ≤ ε3) or (||g||∞ ≤ ε1)
while (not found) and (k < kmax)
    k = k + 1
    Bereken α
    Bereken hgd
    Bereken hgn
    Bereken hdl met de waarden van hgd en hgn
    if ||hdl|| ≤ ε2(||x|| + ε2)
        found = true
    else
        xnew = x + hdl
        Bereken ρ

```

```

if  $\rho > 0$ 
   $x = x_{new}$ 
   $found = (\|f(x)\|_{\infty} \leq \epsilon_3) \text{ or } (\|g\|_{\infty} \leq \epsilon_1)$ 
if  $\rho > 0.75$ 
   $\Delta = \max \Delta, 3 * \|h_d\|$ 
elseif  $\rho < 0.25$ 
   $\Delta = \Delta/2$ 
   $found = (\Delta \leq \epsilon_2(\|x\| + \epsilon_2))$ 

```

Hoofdstuk 7

Resultaten

In dit hoofdstuk worden enkele resultaten getoond van de implementatie van keyframe gebaseerde semi-automatische segmentatie. De techniek die gebruikt wordt in de implementatie voor de optimalisatie van de energie functie, is Powell's dog-leg methode [34]. Er worden vijf verschillende video sequentie getest met de implementatie. Elk beeld uit de video bestaat uit een rode achtergrond, met daarin een blauwe balk die een contour van een object voorstelt. In deze resultaten wordt getoond hoe goed de keyframe gebaseerde techniek de curves aanpast, zodat deze de beweging van de contour volgt doorheen de video.

In de vier eerste sequenties word er gewerkt met een kunstmatig beeld. In de sequenties beweegt of vervormt de balk langzaam doorheen de video. Er wordt telkens in twee keyframes, handmatig door de gebruiker, de curves getekend over de blauwe balk. Een curve bestaat uit 20 punten, deze worden alle 20 individueel gewijzigd doorheen de optimalisatie. De implementatie gaat dan de curve berekenen voor de tussenliggende frame. Telkens werd de curve van de eerste keyframe gekopieerd naar de tussenliggende frame. De curve gevonden door de implementatie wordt nu beoordeeld op twee vlakken: volgt het de beweging van de blauwe balk en behoudt het zijn algemene vorm.

In de laatste sequentie word gewerkt met een opgenomen beeld. In deze sequentie wordt een rood boek langzaam verplaatst doorheen het beeld. Voor een duidelijk onderscheid te maken tussen de voorgrond en de achtergrond is de video opgenomen voor een groen scherm.

In de eerste sequentie beweegt een rechte blauwe balk langzaam doorheen de video zonder van vorm te veranderen. De optimalisatie geeft voor de tussenliggende frame een goed resultaat. De algemene vorm van de curve blijft behouden met een paar kleine afwijkingen. Ook volgt de curve goed de beweging van de balk doorheen de sequentie. Doordat hier met een rechte balk wordt gewerkt heeft de curvatie term weinig invloed bij de optimalisatie.

In de tweede sequentie wordt de rechte balk vervangen door een kromme balk. Deze beweegt zich ook langzaam door de video zonder van vorm te veranderen. De optimalisatie geeft ook voor deze sequentie een goed resultaat, maar minder goed dan die bij de rechte balk. De punten liggen meer verstrooid overheen de kromme balk. Dit heeft te maken dat nu de curvatie term probeert de curvatie van de kromme balk te behouden. De optimalisatie

moet nu rekening houden met een extra factor, waardoor er meer compromissen moeten gemaakt worden bij het aanpassen van de punten.

In de derde sequentie verandert de kromme balk langzaam van vorm in de video sequentie. In deze sequentie moet de optimalisatie telkens de keuze maken tussen het volgen van de beweging, in deze sequentie is dat de vervorming, en het behouden van de vorm van de curve. In de tussenliggende frame valt het op dat de curve goed de beweging volgt, maar dat bepaalde punten toch buiten de blauwe kromme liggen. Dit komt omwille van de termen die de vorm van de curve willen behouden, in het bijzonder de curvatie term. Indien we de gewichten van de vorm termen verlagen hebben deze termen minder invloed op het resultaat. Echter hierdoor gaan de punten in het resultaat verstrooid geraken. In het ergste geval is er dan helemaal geen curve meer te herkennen in het resultaat.

In de vierde sequentie verandert de kromme balk van vorm en beweegt hij doorheen de video sequentie. Het resultaat van deze sequentie is vergelijkbaar met die van de derde sequentie. Echter voor deze sequentie zijn de gewichten van de curvatie termen verlaagd. Hierdoor gaat de curve van de tussenliggende frame zich meer vormen naar de kromme balk.

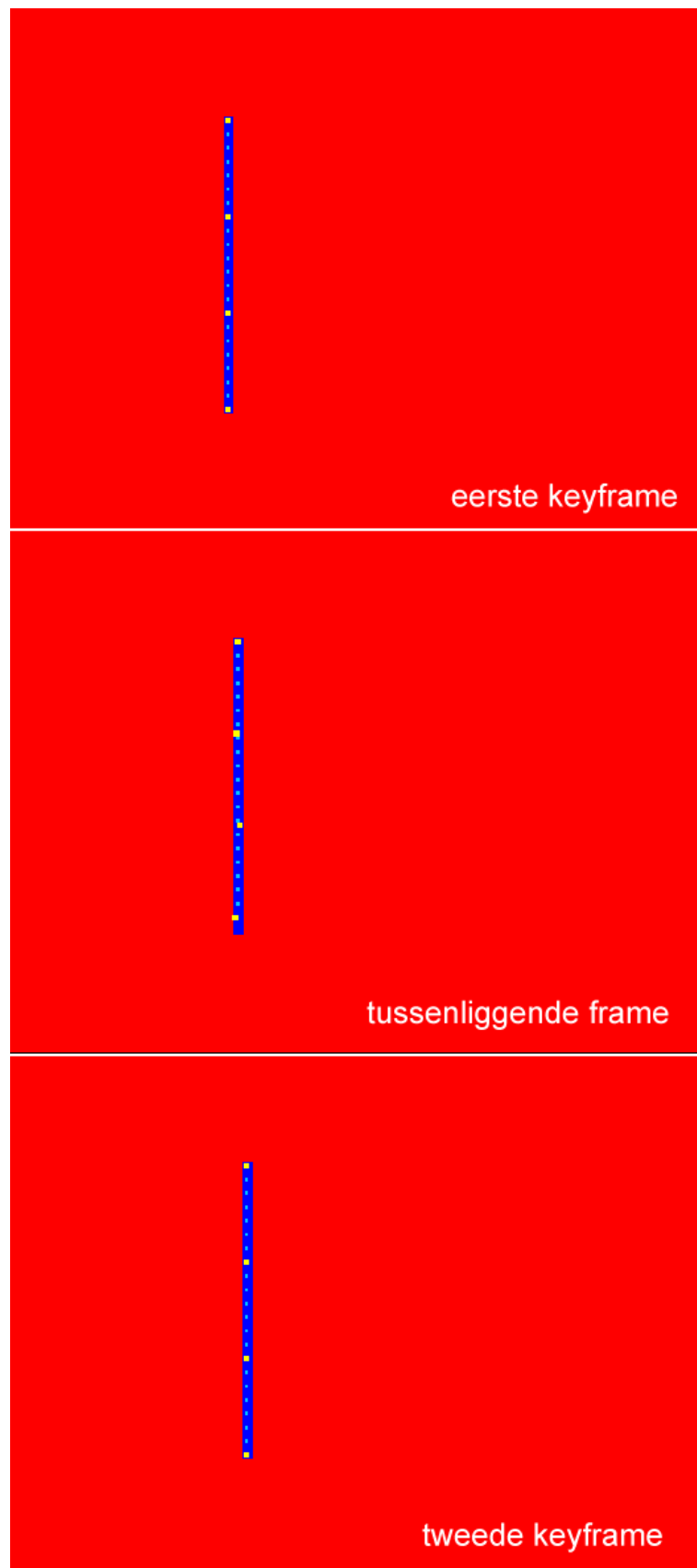
De convergentie voor de ene curve uit de vier sequenties ligt rond de 30 iteraties. De iteraties slaan terug op de veranderingen in de radius van de trust-region. De convergentie voor meerdere curves is in het algemeen traag doordat er veel punten moeten berekend worden voor de optimalisatie. Meerdere punten zorgen ervoor dat de berekeningen per iteratie toenemen. Een ander factor voor de snelheid van het algoritme is de hoeveelheid tussenliggende frames. Hoe meer tussenliggende frames hoe meer punten er moeten berekend worden tijdens de optimalisatie.

Een vereiste voor het goed verlopen van de keyframe-based semi-automatische segmentatie is dat er een scherp onderscheid moet zijn tussen de bewegende objecten en de achtergrond. Indien deze niet duidelijk is, kan het voorkomen dat de optimalisatie pixels uit de achtergrond gaat zien als behorende tot het object. Hierdoor kunnen de aanpassingen ervoor zorgen dat de curve wegdrijft van het te volgen object. Dit komt veel voor bij video beelden opgenomen met een slechte camera. Om video sequenties met bijvoorbeeld personen te segmenteren is er een scherp beeld nodig en goed onderscheid tussen objecten en achtergrond.

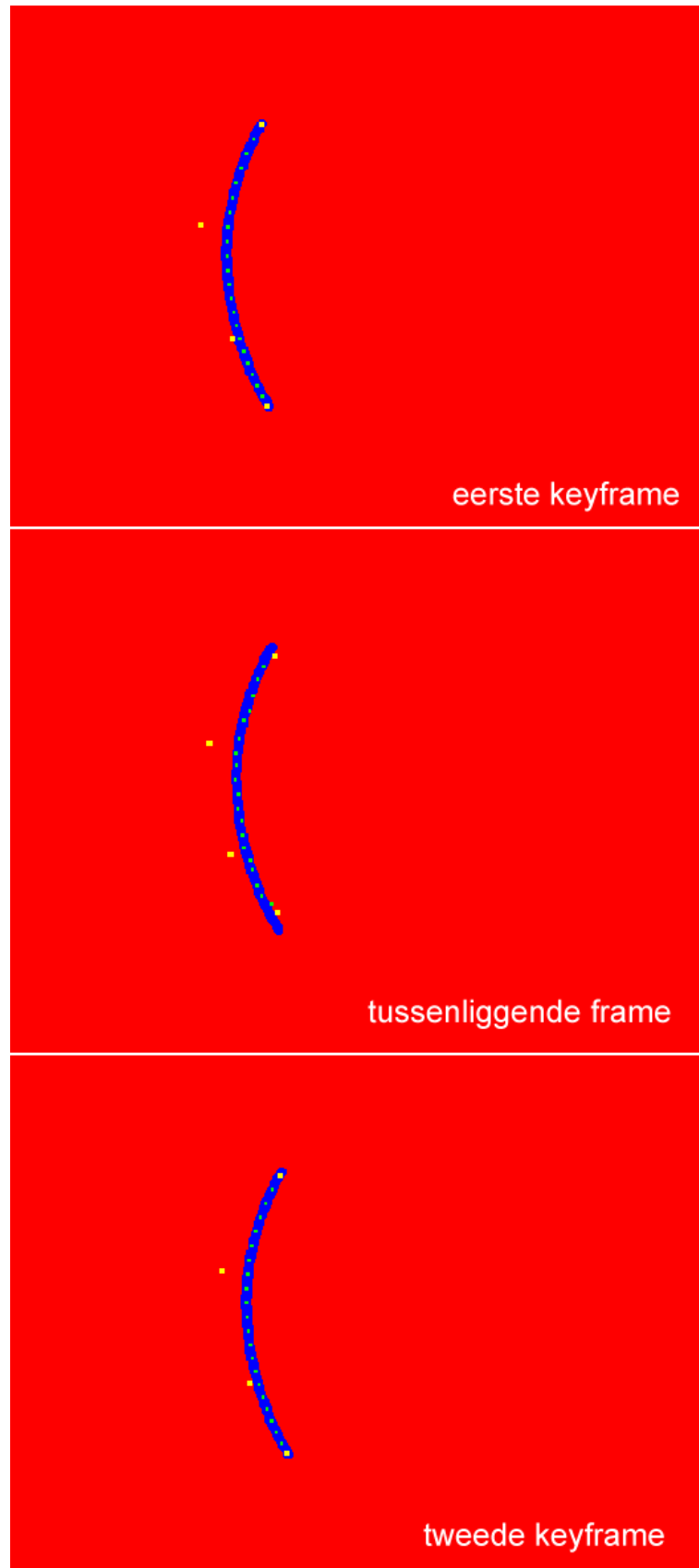
In de laatste sequentie word met een opgenomen video gewerkt. In de video word de curve getekend op de rechtse rand van het boek. De image termen werken met behulp van een dunne band rond de curve. In deze sequentie wordt er in plaats van een band te gebruiken van 5 pixels dik aan beide kanten, een band gebruikt van 15 pixels dik aan beide kanten. Hierdoor testen de image termen een groter gebied voor gelijkheid. Dit word gedaan om de wazigheid van de rand tegen te gaan.

In het resultaat kan gezien worden dat de curve in de tussenliggende frame de beweging van het boek voor het algemeen goed volgt. Op bepaalde delen, zoals de rechts onder, drijft

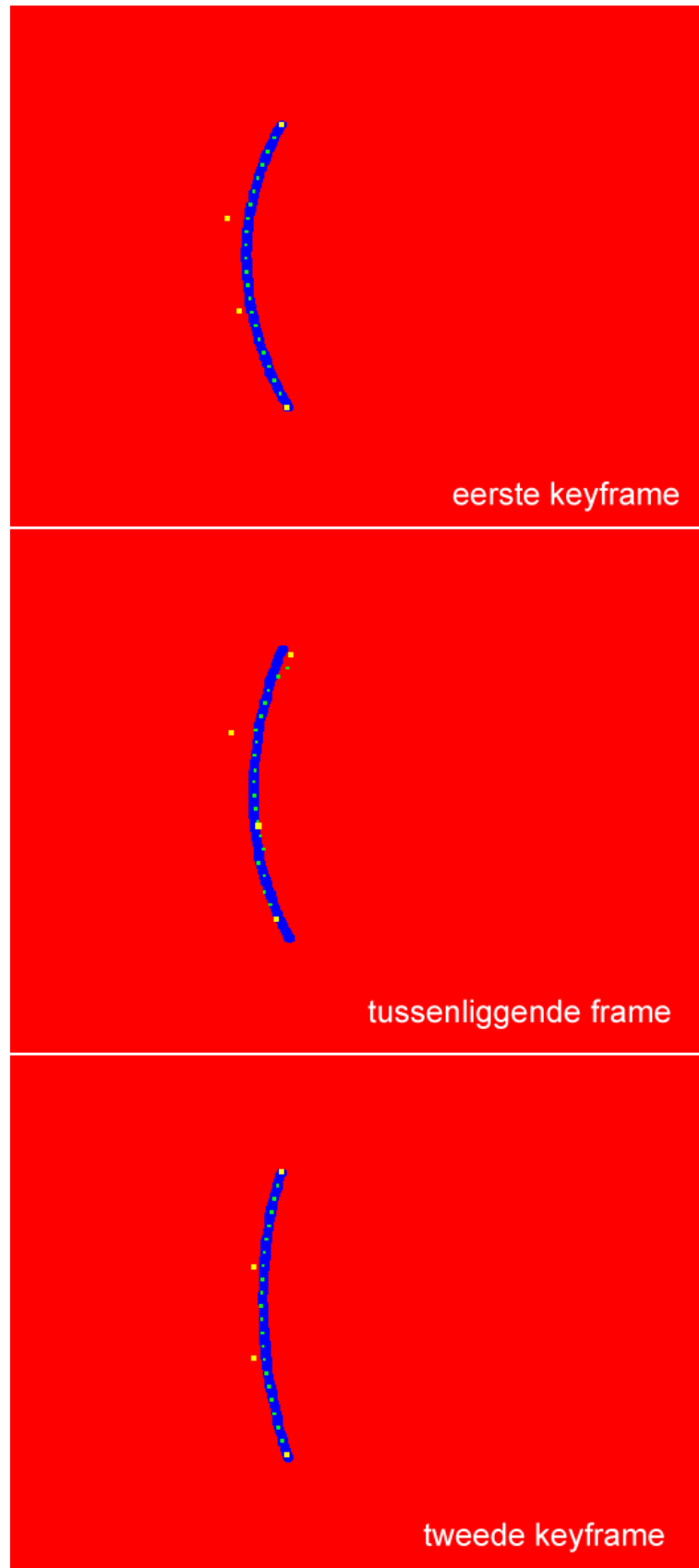
de curve wel naar de binnenkant van het boek. Op deze locatie is de rand waziger dan op andere locaties. Een nadeel van het gebruiken van een dikkere band is dat er meer pixels per iteratie berekend moeten worden. Dit zorgt voor een tragere convergentie.



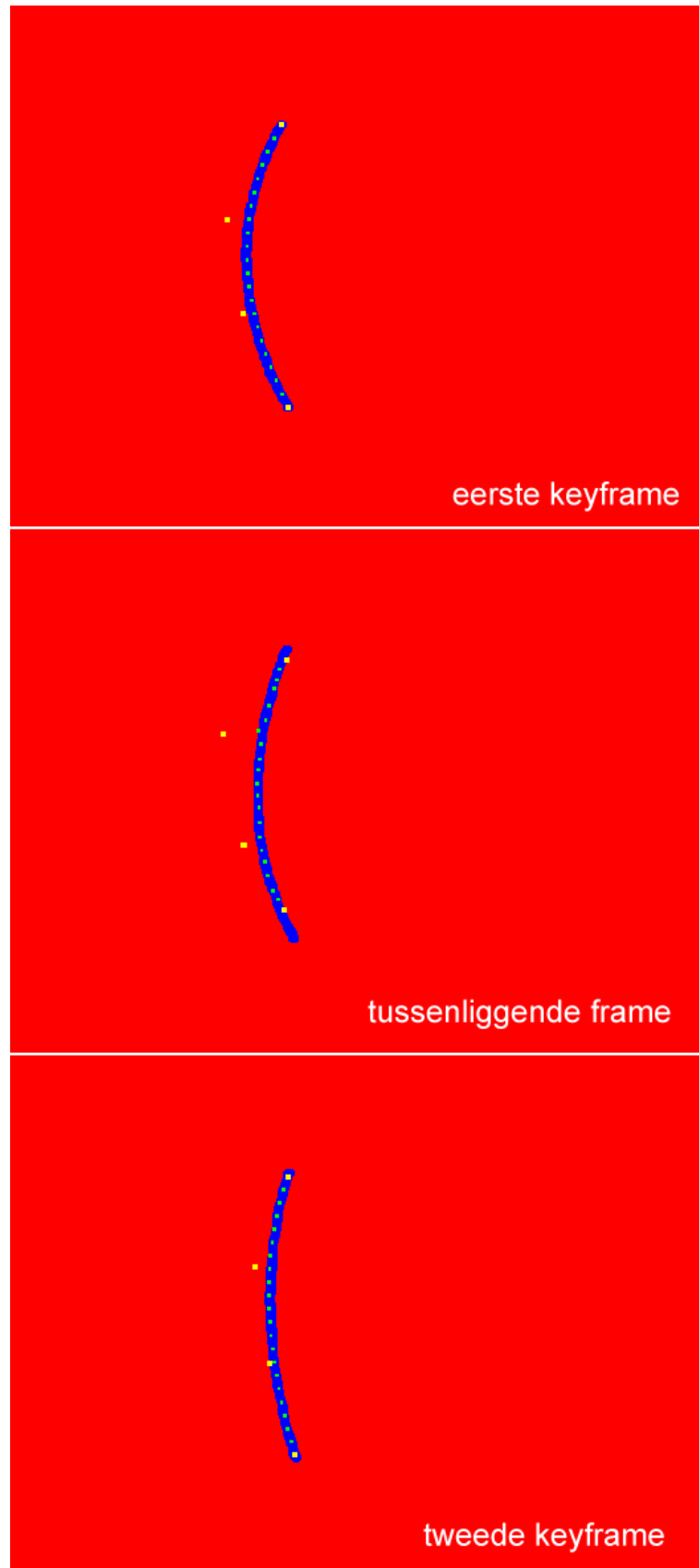
Figuur 7.1: Eerste sequentie: Een blauwe recht balk verplaatst zich langzaam door de scène.



Figuur 7.2: Tweede sequentie: Een blauwe kromme balk verplaatst zich langzaam door de scène.



Figuur 7.3: Derde sequentie: Een blauwe kromme balk vervormd zich langzaam door de scène.



Figuur 7.4: Vierde sequentie: Een blauwe kromme balk vervormd en verplaatst zich langzaam door de scène.



Figuur 7.5: Vijfde sequentie: In een opgenomen video beweegt een rood boek zich langzaam door de scène.

Hoofdstuk 8

Uitbreidingen

In dit hoofdstuk worden enkele problemen van de implementatie gegeven. Er worden uitbreidingen van de implementatie voorgesteld, die deze problemen kunnen oplossen.

Elke curve wordt voorgesteld door enkele sample points. Hoe meer sample points er gebruikt worden, hoe correcter de curve benaderd wordt. De optimalisatie gaat voor elke van deze sample points de verplaatsing berekenen. Al deze sample points moeten bijgehouden worden door het proces. Indien dit veel punten zijn kunnen die veel geheugen vragen.

In de implementatie worden de curves aangemaakt met behulp van Bézier curves. Deze Bézier curves zijn gedefinieerd aan de hand van vier controle punten. De punten van een curve worden gevonden door het toepassen van een wiskundige formule. De verplaatsing van sample points verplaatst ook de positie van de controle punten. Hierdoor kunnen de parameters van de controle punten worden aangepast in de optimalisatie, aan de hand van de verplaatsing van de sample points. Op deze manier moet het algoritme enkel de vier controle punten onthouden, onafhankelijk van het aantal sample points. Een tweede voordeel hiervan is dat de verstrooien van de punten geen gevolgen heeft op de gladde vorm van de curve.

Een bepalende factor voor de correctheid van het resultaat, zijn de initiële waarden van de curves in de tussenliggende frames, die gebruikt worden bij de start van de optimalisatie. Het beste resultaat en de snelste convergentie wordt bekomen, indien de startwaarden dicht bij de uiteindelijke waarden liggen. In de implementatie wordt een simpele kopieer methode gebruikt. De curves van het eerste of het tweede keyframe worden gebruikt als startwaarden. Dit is echter niet ideaal omdat tussenliggende curves sterk kunnen verschillen van die van de keyframes.

Een betere methode, voor het bepalen van de startwaarde is, om gebruik te maken van lineaire interpolatie. De tussenliggende curves worden dan genomen als een interpolatie tussen de twee keyframes. Bijvoorbeeld, in het geval dat er maar één tussenliggende frame is en de curves, in de keyframes, zijn verschillend ten opzichte van hun locatie. In de kopieer methode ligt de tussenliggende curve dan op één van de locaties van de keyframes. Bij de interpolatie methode ligt de tussenliggende curve in het midden tussen de twee locatie van de keyframes.

In de implementatie moet de gebruiker de curves van de keyframes volledig manueel aanbrengen met behulp van Bézier curves. Dit kan een vermoeiend en langdurig proces zijn. Om het aanmaken van keyframes door de gebruiker gemakkelijker te maken kunnen er technieken zoals Snakes [21] of Intelligent Scissors [29] gebruikt worden. Ook andere technieken uit de beeld segmentatie en rand detectie kunnen de gebruiker helpen voor het aanmaken van curves in de keyframes.

Hoofdstuk 9

Conclusie

In deze thesis werden verschillende beeld en video segmentatie methodes besproken, met in het detail de keyframe gebaseerd tracking methode. Voor segmentatie, met behulp van keyframe gebaseerd tracking, werden ideeën gehaald uit zowel beeld als video segmentatie. Keyframe gebaseerde tracking neemt de ideeën van de beeld segmentatie techniek Snakes en combineert die met het gebruiken van keyframes uit de video segmentatie.

De voorgestelde segmentatie techniek van Snakes maakt gebruik van curves aangemaakt door de gebruiker. Deze curve werden aangepast zodat deze overeenkomen met de contouren van een object in het beeld. Keyframe gebaseerde tracking breidt Snakes uit. In keyframe gebaseerde tracking worden de curves voor elke frame tussen twee keyframes in de video tegelijk aangepast zodat deze op de contouren van het object liggen in alle frames. Dit zorgt ervoor dat de segmentatie consistent is doorheen de video.

Het voordeel van deze techniek is dat de gebruiker veel controle heeft over de segmentatie. De gebruiker kan in het begin van de segmentatie aangeven welke contouren worden gesegmenteerd. Hierdoor wordt er veel werk bespaard tijdens de segmentatie, omdat men specifiek kan kiezen welk object gesegmenteerd dient te worden.

Een tweede voordeel is dat het resultaat van de segmentatie mooie curves zijn (Let wel, dit is enkel indien men de controle punten aanpast tijdens de optimalisatie). Dit ontbreekt in vele andere video segmentatie technieken, waar de segmentatie pixel per pixel gebeurt, zoals in background subtraction. Bij zulke technieken zijn de randen van de gesegmenteerde gebieden erg verstrooid. Met behulp van keyframe gebaseerde tracking worden de regio's gesegmenteerd met curves, waardoor zij gladde randen hebben.

Zoals de naam doet blijken is keyframe gebaseerde tracking een tracking techniek. Het doel van keyframe gebaseerd tracking is dus dat het de beweging volgt van objecten doorheen de video. Het enige wat er voor zorgt dat de techniek de contouren zoekt van object, is wanneer de curves in de keyframe op de contouren liggen van het object. Omdat de gebruiker de keyframes bepaalt, kan het zijn, dat deze niet perfect de contouren volgen waardoor de segmentatie niet perfect is.

Hierdoor is segmentatie met behulp keyframe gebaseerd tracking geen perfecte techniek. Dit kan natuurlijk vermeden worden door de curves van de keyframes door een beeld segmentatie techniek te laten bepalen. Maar hierdoor verliest het zijn voordeel van een meer gecontroleerde segmentatie.

Nog een probleem is dat de resultaten van segmentatie met behulp keyframe gebaseerd tracking, curves zijn. Deze curves geven enkel de locatie van de randen in de tussenliggende frames. Dit kan gezien worden uit de resultaten uit hoofdstuk 7. Er is hier nog niet aan segmentatie gedaan. De informatie over de locatie van de randen, die resulteren uit het algoritme, moeten vervolgens gebruikt worden om het beeld te verdelen in verschillende regio's. De eigenlijk segmentatie word dan bekomen met behulp van bijkomende algoritmes. Het is dan voor de segmentatie wel vereist dat de curves een regio in het beeld omsluiten. Hierdoor word er weer een extra beperking gelegd op de vrijheid van de gebruiker.

Voor deze redenen is segmentatie met behulp keyframe gebaseerd tracking een techniek, die het best gebruikt wordt wanneer men enkel een minder correcte segmentatie wilt van de video. Meestal wordt enkel de beweging van het object gebruikt die men wou segmenteren. Wanneer men een meer correcte segmentatie wilt, kan men beter technieken zoals Video Cutout gebruiken.

Bibliografie

- [1] Motion capturing. http://en.wikipedia.org/wiki/Motion_capture.
- [2] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. *Proceedings of ACM SIGGRAPH*, pages 584–591, 2004.
- [3] F. Vanden Berghen. Condor: a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions. Master’s thesis, Universit Libre de Bruxelles, 2004.
- [4] A. Blake and M. Isard. *Active Contours*. Springer, 1998.
- [5] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *In Proceedings of ICCV 2001*, pages 105–112, 2001.
- [6] C. Brice and C. Fennema. Computer method in image analysis, ch. scene analysis using regions. *Los Angeles: IEEE Computer Society*, 1977.
- [7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 8, page 769, 1986.
- [8] T. Chan, S. Esedoglu, and K. Ni. Histogram based segmentation using wasserstein distances. *Department of Mathematics, University of Michigan and Department of Mathematics, UCLA*.
- [9] J. P. Collomosse, D. Rowntree, and P. M. Hall. Stroke surfaces: A spatio-temporal framework for temporally coherent non-photorealistic animations. *University of Bath, Technical Report CSBU 2003-01*, 2003.
- [10] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Trust region methods. *Monograph*, 1999.
- [11] G. Dahlquist and A. Bjorck. *Numerical Methods Second Ed*. Prentice Hall, 1995.
- [12] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, vol. 39, pages 1–38, 1977.
- [13] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–270, 1959.

- [14] R. Fletcher. *Practical Methods of Optimization*. Wiley; 2 Sub edition, 1987.
- [15] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice hall, 2003.
- [16] K. Fu, R. Gonzales, and C. Lee. *Robotics: Control, sensing, vision, and intelligence*. New York: McGraw-Hill, 1987.
- [17] R. Gonzales and R. Woods. *Digital Image Processing*. Addison-Wesley Publishing, 1993.
- [18] J. Hall, D. Greenhill, and G. Jones. Segmenting film sequences using active surfaces. *International Conference on Image Processing (ICIP)*, pages 751–754, 1997.
- [19] T. H. Hong and A. Rosenfeld. Compact region extraction using weighted pixel linking in a pyramid. *IEEE Trans. Anal. and Mach. Intell. PAMI-6, No. 2. March*, pages 222–229, 1984.
- [20] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision* 29(1), pages 5–28, 1998.
- [21] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.
- [22] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.* 2, pages 164–168, 1944.
- [23] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *Microsoft Research Asia*.
- [24] B. D. Lucas and T. Kanade. An interactive image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, 1981.
- [25] H. Luo and A. Eleftheriadis. Spatial temporal active contour interpolation for semi-automatic video object generation. *International Conference on Image Processing (ICIP)*, pages 944–948, 1999.
- [26] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [27] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.* 11, pages 431–441, 1963.
- [28] A. M. McIvor. Background subtraction techniques. *California State University East Bay*.
- [29] E. N. Mortensen and W. A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, pages 349–384, 1998.
- [30] S. J. Osher and R. P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer-Verslag, 2002.

- [31] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.* 33, pages 1065–1076, 1962.
- [32] T-Y. Philips, A. Rosenfeld, , and A. C. Sher. $O(\log n)$ bimodality analysis. *Pattern Recognition* 22, pages 741–746, 1989.
- [33] T. Porter and T. Duff. Compositing digital images. *Proceedings of SIGGRAPH '84*, 1984.
- [34] M. J. D. Powell. A hybrid method for nonlinear equations. *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–144, 1970.
- [35] M. J. D. Powell. A new algorithm for unconstrained optimization. *Nonlinear Programming*, pages 31–66, 1970.
- [36] J. A. Sethian. Level set methods and fast marching methods. *Cambridge University Press*, 1999.
- [37] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Anal. and Mach. Intell.* Vol. 22 No. 8, 2000.
- [38] G. A. Shultz, R. B. Schnabel, and R. H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence. *SIAM J. Numer. Anal.* 22, pages 47–67, 1985.
- [39] J. Silverman and D. Cooper. Bayesian clustering for unsupervised estimation of surface and texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 57, pages 373–387, 1993.
- [40] W. Snyder and H. Qi. Machine vision. *Cambridge University Press*, 2004.
- [41] A. Srivastava and A. Bouarfa. Image segmentation by a parallel, non-parametric histogram based clustering algorithm. *Pattern Recognition*, pages 961–973, 1990.
- [42] T. Steihaug. The conjugated gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.* 20, pages 626–637, 1983.
- [43] D. Titterton, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley and Sons, 1985.
- [44] J. Weber. Vision Based Traffic Surveillance. California Institute of Technology.
- [45] G. Welch and G. Bishop. An introduction to the kalman filter. *University of North Caroline*, 2006.
- [46] Y. Yuan. A review of trust region algorithms for optimization. *Chinese Academy of Sciences*.
- [47] S. Zhu and A. Yuille. Region competition and its analysis: a unified theory for image segmentation. *Tech. Rep. 7, Robotics Lab, Harvard University*, 1995.

- [48] S. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 9, pages 884–200, 1996.

Auteursrechterlijke overeenkomst

Opdat de Universiteit Hasselt uw eindverhandeling wereldwijd kan reproduceren, vertalen en distribueren is uw akkoord voor deze overeenkomst noodzakelijk. Gelieve de tijd te nemen om deze overeenkomst door te nemen, de gevraagde informatie in te vullen (en de overeenkomst te ondertekenen en af te geven).

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

Semi-automatische Video Segmentatie

Richting: **Master in de informatica**

Jaar: **2007**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Ik ga akkoord,

Jonas Buntinx

Datum: **21.08.2007**