

Motion Capturing

Koen VANDORMAEL

promotor :
Prof. dr. Philippe BEKAERT

Eindverhandeling voorgedragen tot het bekomen
van de graad Licentiaat in de informatica
afstudeervariant multimedia



Motion Capturing

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

Koen Vandormael

Promotor: Prof. Dr. Philippe Bekaert

Begeleider: Jan Fransens

Transnationale Universiteit Limburg
2005-2006

Abstract

Motion capturing is een techniek om bewegingen te digitaliseren en over te zetten naar een animatiemodel. Deze techniek kan toegepast worden in verscheidene vakgebieden, maar komt het meest tot uiting in films en games. De verschillende motion capture systemen worden besproken en met elkaar vergeleken.

Vervolgens wordt er dieper ingegaan op de werking van een optisch motion capture systeem. Eerst wordt het algemene werkingsproces uitgelegd. Daarna worden de stappen van dit proces in detail beschreven.

De eerste stap is camerakalibratie. Er wordt uitgelegd met welke parameters een camera kan gemodelleerd worden en hoe men deze kan vinden.

In de tweede stap worden de markers gedetecteerd in de camerabeelden. Dit kan onder meer gebeuren op basis van vorm, kleur en intensiteit.

De volgende stap is de markertracering. Hier worden de markers in 3D gereconstrueerd en hun individuele trajecten berekend.

Tot slot wordt er aan de hand van deze markertrajecten een skelet -en markermodel opgesteld en bekomen we een geanimeerd skelet. Dit wordt ook wel skeleton fitting genoemd.

De algoritmes die in deze stappen gebruikt worden, kunnen verbeterd worden indien er al van in het begin een skelet -en markermodel aanwezig is.

Tot slot worden de implementaties van de beschreven motion capture algoritmes besproken, evenals de resultaten die ermee behaald werden.

Woord vooraf

Allereerst zou ik enkele personen willen bedanken voor de hulp en steun tijdens het maken van deze thesis. Mijn promotor Prof. Dr. Philippe Bekaert en begeleider Jan Fransens voor de nodige ondersteuning en begeleiding. Bert De Decker, die me geholpen heeft met de cameraopnames. En uiteraard ook mijn familie en vrienden, die hielpen bij het nalezen. Ik ben ook dankbaar voor de kans om dit onderwerp te kunnen bespreken omdat het zoveel nuttige en veel voorkomende toepassingen heeft.

Inhoudsopgave

1	Inleiding	1
2	Overzicht	3
2.1	Definitie	3
2.2	Technieken	3
2.2.1	Magnetisch systeem	4
2.2.2	Mechanisch systeem	4
2.2.3	Akoestisch systeem	5
2.2.4	Inert systeem	6
2.2.5	Optisch systeem	6
2.3	Classificatie	7
2.3.1	Resultaatweergave	7
2.3.2	Plaats van de bron en de sensor	8
2.4	Toepassingen	8
3	Een optisch motion capture systeem	10
3.1	Het proces	10
4	Cameramodel	12
4.1	Pinhole-model	12
4.2	Cameraparameters	12
4.2.1	Extrinsieke cameraparameters	13
4.2.2	Intrinsieke cameraparameters	13
4.3	Projectiematrix	14
4.4	Vervorming	15
4.4.1	Radiale vervorming	15
4.4.2	Tangentiale vervorming	15
4.5	Camerakalibratie	16
4.6	Epipolaire geometrie	16
4.6.1	Fundamentele matrix	17

5	Markerdetectie	18
5.1	Achtergrond	18
5.1.1	Achtergrondsubtractie	18
5.1.2	Speciale achtergrond	19
5.2	Markers	19
5.2.1	Intensiteit	19
5.2.2	Kleur	19
5.2.3	Vorm	19
5.3	Exacte markerpositie bepalen	20
5.4	Maximum aantal markers	20
5.5	Temporele correlatie	20
5.5.1	Verwacht aantal markers	20
5.5.2	Markerposities in het vorige frame	20
6	3D-reconstructie	21
6.1	Correspondentieprobleem	21
6.1.1	Epipolaire lijnen	21
6.1.2	Herprojectie	22
6.1.3	Foutieve correlaties	22
6.1.4	Resultaat	22
6.2	Reconstructieprobleem	23
6.2.1	Twee camerastandpunten	23
6.2.2	Meerdere camerastandpunten	25
7	Markertracering	27
7.1	Marker datastructuur	27
7.1.1	2D-marker	27
7.1.2	3D-marker	27
7.2	Glijdende venster van vier frames	28
7.3	Globale traceringsalgoritme	29
8	Skeleton fitting	31
8.1	Skeletmodel	31
8.2	Lokale methode	33
8.2.1	Opdeling en associatie van de markers	33
8.2.2	Berekening van de posities van de gewrichten	34
8.2.3	Schalering en positionering van het skelet	36
8.3	Globale methode	38
8.3.1	Zonder initialisatie	38
8.3.2	Met initialisatie	38

9	Optimalisatie met het skelet -en markermodel	40
9.1	Het proces	40
9.2	Gymbeweging	40
9.3	3D-reconstructie	41
9.3.1	Zichtbaarheidstest	41
9.3.2	Occlusietest	42
9.4	Markertracering	44
9.5	Skeleton fitting	44
9.6	Resultaten	45
10	Implementatie	47
10.1	Gebruikte bibliotheken en programma's	47
10.2	Invoer	48
10.2.1	Camera's	48
10.2.2	Markers	49
10.3	Grafische gebruikersomgeving	51
10.4	Camerakalibratie	51
10.5	Markerdetectie	53
10.5.1	Achtergrond verwijderen	53
10.5.2	Kleur zoeken	54
10.5.3	Ruis verwijderen	54
10.5.4	Cirkels zoeken	54
10.6	3D-reconstructie	55
10.7	Markertracering	56
10.8	Skeleton fitting	56
10.8.1	Lokale methode	56
10.8.2	Globale methode	57
10.9	Resultaten	57
11	Conclusies	60

Hoofdstuk 1

Inleiding

Motion capturing is een techniek die gebruikt wordt om bewegingen te digitaliseren en over te zetten naar een animatiemodel. De meeste mensen kennen het uit de filmindustrie. Het bekendste voorbeeld is waarschijnlijk het populaire karakter Gollum uit de „The Lord of the Rings”-trilogie. Dit karakter werd tot leven gewekt dankzij de acteerprestaties van acteur Andy Serkis.

Het doel van deze thesis is om een overzicht te geven van de verschillende motion capture systemen en om één van deze systemen, namelijk het optische systeem, meer in detail te bespreken. Er werd eveneens een implementatie gemaakt van de algoritmes die gebruikt worden bij een optisch systeem.

In hoofdstuk 2 wordt uitgelegd hoe motion capturing precies in zijn werk gaat en wat de mogelijke toepassingen ervan zijn. Er wordt aandacht besteed aan de verschillende motion capture systemen en hoe deze kunnen geclassificeerd worden.

Hoofdstuk 3 beschrijft in stappen de algemene werking van een optisch motion capture systeem. Hoofdstukken 4 tot en met 8 gaan dieper in op deze stappen.

In hoofdstuk 4 wordt het cameramodel beschreven. Er wordt uitgelegd welke parameters er nodig zijn om een camera te kunnen modelleren en hoe deze parameters gevonden kunnen worden.

Hoofdstuk 5 bespreekt enkele beeldsegmentatietechnieken die kunnen gebruikt worden om markers in de camerabeelden te vinden.

In hoofdstuk 6 wordt uitgelegd op welke manieren de gevonden markers in 3D kunnen gereconstrueerd worden. Dit hoofdstuk wordt opgesplitst in het correspondentieprobleem en het reconstructieprobleem.

In hoofdstuk 7 wordt beschreven hoe men de individuele trajecten van de markers kan

bepalen.

Aan de hand van deze markertrajecten kan men een skeletmodel creëren en dit positioneren in elk frame. Dit wordt skeleton fitting genoemd en wordt besproken in hoofdstuk 8. Na de skeleton fitting verkrijgen we een skelet -en markermodel.

Hoofdstuk 9 bespreekt welke optimalisaties er kunnen doorgevoerd worden indien er reeds van in het begin een skelet -en marker model beschikbaar is.

Tot slot wordt in hoofdstuk 10 de implementatie besproken. Er zal hier een overzicht gegeven worden van de geïmplementeerde technieken. Verder worden ook de resultaten getoond.

Hoofdstuk 2

Overzicht

2.1 Definitie

Motion capturing is een techniek die gebruikt wordt om bewegingen digitaal op te nemen en daarna over te zetten naar een animatiemodel.

Hiervoor worden herkenningspunten op het lichaam van de acteur aangebracht. Deze herkenningspunten worden ook wel markers genoemd. Aan de hand van de bewegingen van de markers, wordt de beweging van de acteur gereconstrueerd [Mot].

2.2 Technieken

Zoals net al aangegeven werd, is het traceren van de markers een belangrijk onderdeel van het motion capture proces. Hier bestaan verschillende technieken voor, met elk hun specifieke eigenschappen. Naargelang de toepassing wordt er voor een bepaalde techniek gekozen. In sommige gevallen worden meerdere technieken tesamen gebruikt, zodat de nadelen van de technieken op zich wegvallen. Enkele criteria bij de keuze van een motion capture techniek zijn:

- **Nauwkeurigheid:** Voor films is er bijvoorbeeld een grote nauwkeurigheid nodig om de animaties zo realistisch mogelijk te laten overkomen. Bij real-time toepassingen zal de nauwkeurigheid meestal lager zijn.
- **Prijs:** Deze kan sterk variëren en hangt dikwijls af van de nauwkeurigheid.
- **Kalibratietijd:** Voor sommige systemen duurt het langer om ze klaar te zetten voor een opname.
- **Opnameruimte:** Dit kan binnen in een speciaal ingerichte kamer zijn, of in de open lucht. De grootte van de opnameruimte varieert ook per techniek.
- **Bemonsteringsfrequentie:** Dit is het aantal keer per seconde dat er een meting gebeurt.

- **Latentietijd:** Dit is de tijd tussen het uitvoeren van een beweging en de registratie ervan. De latentietijd wordt uitgedrukt in milliseconden.
- **Aantal vrijheidsgraden:** Sommige systemen registreren enkel de positie of de oriëntatie van de markers.

De volgende secties bespreken kort de meest gebruikte motion capture technieken en vergelijken hun voor- en nadelen.

2.2.1 Magnetisch systeem

Een transmitter creëert een magnetisch veld in de ruimte. Dit magnetische veld wordt opgemeten door sensoren die op het lichaam bevestigd zijn. De sensoren zijn met kabels bevestigd aan een computer waar de positie en de oriëntatie van de sensoren berekend worden. Uit deze gegevens worden dan de posities van de gewrichten afgeleid [Cap].

- De precisie van de sensoren vermindert naargelang de afstand tot de transmitter groter wordt.
- De opnameruimte blijft daardoor beperkt tot enkele meters.
- Wanneer de sensoren niet draadloos zijn, kan men gehinderd worden bij het uitvoeren van complexe bewegingen.
- Door het wegfilteren van ruis kan het aantal metingen per seconde al snel terugvallen. Deze ruis kan ontstaan wanneer er zich andere magnetische velden of metalen objecten in de omgeving bevinden.
- De aanpassing van de sensorposities en de kalibratie kunnen veel tijd in beslag nemen.
- De resultaten kunnen in real-time getoond worden.
- Er doen zich geen oclusies voor.

2.2.2 Mechanisch systeem

Er wordt hier een uitwendig skelet rond het lichaam bevestigd. Wanneer men dan beweegt, zal het mechanische skelet mee bewegen. Op de gewrichten worden potentiometers aangebracht, die de gevormde hoeken registreren [Mot].

- Deze techniek is zeer accuraat en de metingen gebeuren aan een hoge snelheid.
- De globale positie en oriëntatie van de acteur zijn niet bekend.
- Het uitwendige skelet kan hinderlijk zijn bij het uitvoeren van bewegingen.
- De opnameruimte kan groot zijn.



Figuur 2.1: Magnetisch motion capture systeem.

- De resultaten kunnen in real-time getoond worden.
- Er doen zich geen oclusies voor.

2.2.3 Akoestisch systeem

In de ruimte worden een drietal geluidsontvangers opgesteld. Op het lichaam worden transmitters aangebracht, die elk om beurt een ultrasoon klikgeluid uitzenden. Daarmee worden de afstanden tot de geluidsontvangers berekend. Aan de hand van deze afstanden kunnen de 3D-posities bepaald worden [Tra].

- Er wordt geen momentopname gemaakt van de lichaamspositie omdat men maar voor één transmitter tegelijk de positie kan bepalen.
- Geluiden van buitenaf kunnen voor ruis zorgen.
- Oclusies zorgen er voor dat het langer duurt vooraleer het klikgeluid wordt opgevangen. Daardoor wordt het bepalen van de positie verstoord.
- Kabels maken complexe bewegingen moeilijk uitvoerbaar.
- De opnameruimte wordt begrensd door het aantal transmitters en de snelheid van het geluid in lucht.
- Er kan een hoge nauwkeurigheid behaald worden bij de positiebepaling.
- Het aantal transmitters dat realistisch gebruikt kan worden, is te laag om een nauwkeurige reconstructie van een menselijke beweging te maken.

2.2.4 Inert systeem

Een inert systeem maakt gebruik van gyroscopen en accelometers om de posities en de hoeken van de gewrichten te bepalen. Een gyroscoop bevat een snel draaiend wiel. Fysische wetten zorgen ervoor dat dit wiel weerstand biedt tegen een verandering van oriëntatie. Deze weerstand wordt gemeten en daaruit worden de hoeken afgeleid. Een accelometer meet de versnelling van de sensor. Hieruit wordt de positie afgeleid [BB].

- Onnauwkeurige metingen kunnen accumuleren waardoor de positie verkeerd bepaald wordt.
- Daarom wordt deze techniek meestal gebruikt in combinatie met een andere techniek.
- De opnameruimte is groot.
- De sensoren halen een hoge bemonsteringsfrequentie.
- Er doen zich geen oclusies voor.



Figuur 2.2: Verschillende motion capture systemen: mechanisch (wit), optisch (blauw) en inert (geel).

2.2.5 Optisch systeem

Rond de opnameruimte worden een aantal camera's geplaatst en op het lichaam van de acteur worden markers aangebracht. Dit resulteert voor elke camera in een aantal 2D-punten

in de beelden. Door de overeenkomsten te vinden tussen de beelden van de verschillende camera's, kunnen de 3D-posities van de markers berekend worden.

Er kan hier nog een onderscheid gemaakt worden tussen de systemen met actieve en passieve markers.

Een passieve marker is meestal een bol of een halve bol waarop een reflecterende tape werd aangebracht. Rond de cameralens bevindt zich een ring van infrarode LED's¹, waarvan het licht terug naar de camera gereflecteerd wordt.

Een actieve marker is zelf een infrarode LED die het licht naar de camera stuurt.

De camera's zijn voorzien van een speciale infraroodfilter om de markers te kunnen detecteren [Mot].

- De opnameruimte kan vergroot worden door meer camera's te gebruiken.
- Het is zeer nauwkeurig en kan een hoge bemonsteringsfrequentie halen. Sommige camera's kunnen aan duizenden frames per seconde opnemen.
- Het kalibreren van de camera's kan veel tijd in beslag nemen.
- Er kunnen zich oclusies voordoen, maar wanneer er voldoende camera's gebruikt worden, is dit geen probleem.

2.3 Classificatie

De verschillende motion capture technieken kunnen ook geclassificeerd worden op andere manieren.

2.3.1 Resultaatweergave

Online systeem

Een online systeem kan gebruikt worden om een animatiemodel direct aan te sturen. De resultaten worden dus in real-time getoond.

Een magnetisch systeem is een voorbeeld van een online systeem [HFP⁺00].

Offline systeem

Bij een offline systeem worden de bewegingen van de acteur eerst opgenomen en daarna pas verwerkt tot een geanimeerd model. Offline systemen, zoals bijvoorbeeld een optisch systeem, hebben als voordeel dat de algoritmes die de data verwerken niet onder een tijdsdruk staan. Hierdoor kunnen er meestal nauwkeurigere resultaten behaald worden [HFP⁺00].

¹Light Emitting Diode

2.3.2 Plaats van de bron en de sensor

De beschreven motion capture technieken maken steeds gebruik van een bron, die bepaalde signalen uitzendt, en een sensor, die deze signalen opvangt. Naargelang de positie van de bron en de sensor kunnen de verschillende technieken opgedeeld worden [Mul94].

Inside-in systeem

Bij een inside-in systeem bevinden de bron en de sensor zich beide op het lichaam van de acteur. De sensoren kunnen klein zijn, waardoor ze geschikt zijn voor het traceren van kleine lichaamsdelen. De opnameruimte is ongelimiteerd omdat de bron en de sensor zich samen verplaatsen. Dit heeft echter als nadeel dat er geen globale 3D-informatie geregistreerd wordt.

Inside-out systeem

Bij een inside-out systeem bevinden enkel de sensoren zich op het lichaam van de acteur. Deze sensoren registreren de signalen van een externe bron. Een magnetisch systeem is hier een voorbeeld van. Globale 3D-informatie is wel beschikbaar bij een inside-out systeem, maar de opnameruimte is meestal beperkt.

Outside-in systeem

Bij een outside-in systeem bevinden enkel de bronnen zich op het lichaam van de acteur. Een voorbeeld hiervan is een optisch systeem waarbij de markers en de camera's respectievelijk de bronnen en de sensoren voorstellen. Ook hier is de globale 3D-informatie beschikbaar. Een nadeel van dit systeem is de mogelijkheid op het voorkomen van oclusies.

2.4 Toepassingen

Motion capturing heeft toepassingen in verschillende vakgebieden:

- **Animatie:** Motion capturing kan het animatieproces aanzienlijk versnellen. Een acteur kan op één dag een heleboel animaties opnemen, die nadien wel nog opgeschoond moeten worden. Deze animaties op de normale manier maken, zou maanden in beslag kunnen nemen.
Motion capturing neemt ook al de natuurlijke bewegingen van de acteur over. Deze zouden door de traditionele animator over het hoofd gezien kunnen worden.
Een derde voordeel is dat bewegingen, die fysisch moeilijk te modelleren zijn, accuraat opgenomen kunnen worden [Mot].
- **Film:** In de filmindustrie worden tegenwoordig veel computergegenereerde beelden gebruikt. De bekendste voorbeelden zijn „King Kong” en Gollum uit de „The Lord of the Rings”-trilogie.

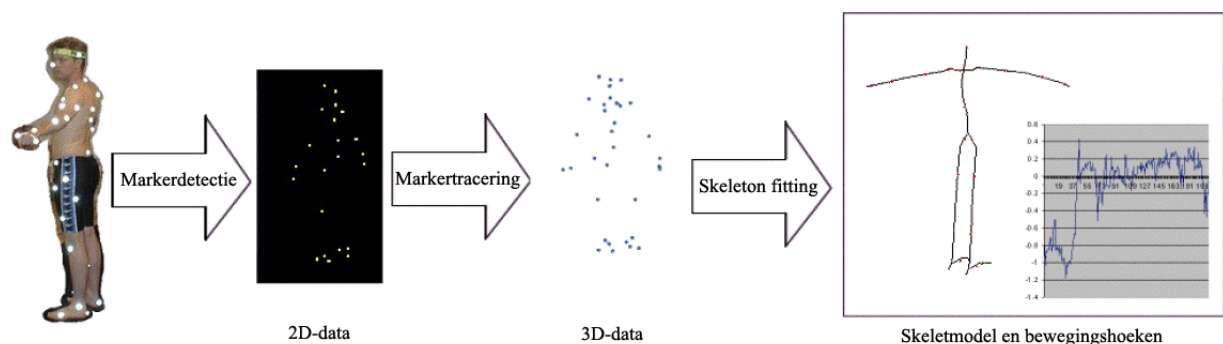
- **Games:** Motion capturing wordt vaak toegepast bij het maken van sportgames om de animaties zo realistisch mogelijk te maken.
- **Virtuele realiteit:** De bewegingen van de gebruiker kunnen in real-time getoond worden in de virtuele wereld.
- **Sport:** Hier kan motion capturing gebruikt worden om de techniek van een sporter te bestuderen en eventueel aan te passen.
- **Geneeskunde:** Er kan onderzoek gedaan worden naar ergonomie of naar de oorzaken van bepaalde fysieke problemen.

Hoofdstuk 3

Een optisch motion capture systeem

In dit hoofdstuk wordt het algemene motion capture proces van een offline optisch systeem uitgelegd. De verdere hoofdstukken gaan dieper in op de verschillende stappen van dit proces.

3.1 Het proces



Figuur 3.1: De verschillende stappen van het motion capture proces.

De stappen van het proces worden weergegeven in figuur 3.1 [HFP⁺01].

1. **Invoer:** Rond de opnameruimte staan een aantal camera's opgesteld. Er zijn minstens twee camera's nodig, maar er worden betere resultaten gehaald wanneer er een tiental camera's gebruikt wordt.
Op het lichaam van de acteur zijn markers aangebracht. Het is belangrijk dat de markers op dezelfde plaats blijven zitten. Daarom kan de acteur best strakke kledij dragen.
2. **Camerakalibratie:** Allereerst moeten de camera's gekalibreerd worden. Dit levert ons informatie over de positie, de oriëntatie en de interne eigenschappen van de

camera's. Met deze informatie wordt elke camera gemodelleerd zodat we alle camera's op een uniforme manier kunnen gebruiken. Het cameramodel en de camerakalibratie worden besproken in hoofdstuk 4.

3. **Markerdetectie:** De camera's nemen synchroon de bewegingen van de acteur op, waardoor er voor elk camerastandpunt een film ontstaat. Uit de beelden van deze films moeten de markers gedetecteerd worden. De 2D-posities van de markers in de beelden zijn de gegevens die verder gebruikt zullen worden in het proces. Markerdetectie wordt besproken in hoofdstuk 5.
4. **Markertracering:** Deze stap zet de 2D-posities van de verschillende camera's om naar 3D-posities. Dit wordt ook wel 3D-reconstructie genoemd en wordt besproken in hoofdstuk 6. Het is eveneens belangrijk dat de trajecten van de markers kunnen gevolgd worden in 3D. Het zoeken naar deze trajecten noemt men markertracering. Markertracering wordt besproken in hoofdstuk 7.
3D-reconstructie wordt beschouwd als een onderdeel van markertracering omdat bij het traceringsalgoritme wordt bijgehouden welke 2D-posities gebruikt werden bij de 3D-reconstructie van een marker.
5. **Skeleton fitting:** De 3D-trajecten van alle markers zijn nu bekend. Met behulp van skeleton fitting kan men aan de hand van deze trajecten een passend skeletmodel creëren en dit in elk frame positioneren zodat het zo goed mogelijk overeenkomt met de posities van de markers. Dit geanimeerde skelet is tevens het resultaat van het motion capture proces. Skeleton fitting wordt besproken in hoofdstuk 8.

Hoofdstuk 4

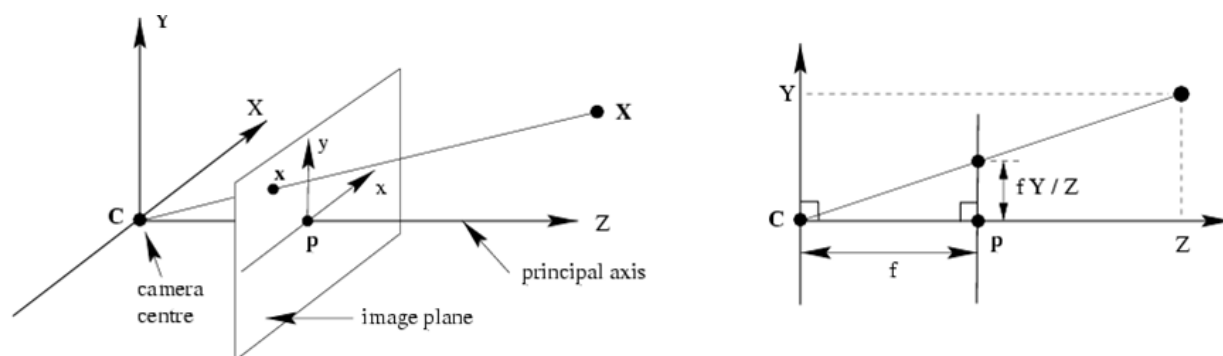
Ceramodel

In dit hoofdstuk wordt het gebruikte cameramodel kort beschreven en de eigenschappen ervan besproken.

Er wordt uitgelegd welke parameters nodig zijn om de werking van een camera te modelleren en hoe deze bekomen kunnen worden. Een gedetailleerde beschrijving kan gevonden worden in [Pol00, Fus06, Deb96, FR94].

4.1 Pinhole-model

Dit cameramodel wordt vaak gebruikt omdat de meeste camera's er goed mee gemodelleerd kunnen worden. De opstelling van een camera wordt bepaald door het kiezen van een cameracentrum en een beeldvlak zoals men kan zien in figuur 4.1. De perspectieve projectie van een 3D-punt op het beeldvlak wordt dan bekomen door de intersectie te berekenen van de lijn tussen het cameracentrum en het 3D-punt met het beeldvlak.



Figuur 4.1: Pinhole-ceramodel

4.2 Cameraparameters

We maken een onderscheid tussen de extrinsieke en de intrinsieke parameters.

4.2.1 Extrinsieke cameraparameters

De extrinsieke cameraparameters beschrijven de relatieve positie en oriëntatie van de camera ten opzichte van een vast wereldcoördinatenstelsel. Aan de hand van deze parameters weten we ook waar de verschillende camera's zich ten opzichte van elkaar bevinden. Deze parameters zijn onafhankelijk van de gebruikte camerahardware.

Rotatie

De rotatie wordt meestal¹ voorgesteld met een 3×3 -matrix.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Translatie

De translatie wordt voorgesteld met een 3×1 -vector.

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

4.2.2 Intrinsieke cameraparameters

De intrinsieke cameraparameters beschrijven de omzetting van de coördinaten in het beeldvlak naar de eigenlijke pixelcoördinaten zoals men kan zien in figuur 4.2. Deze parameters hangen af van de gebruikte camerahardware en hoe ze geconfigureerd wordt. De camera-kalibratiematrix K wordt dan:

$$K = \begin{bmatrix} \frac{f}{p_x} & (\tan \alpha) \frac{f}{p_y} & c_x \\ 0 & \frac{f}{p_y} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

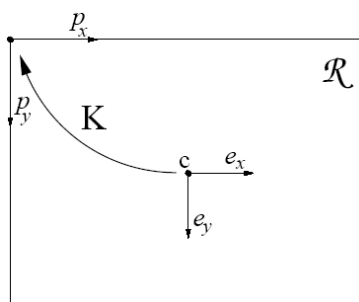
Focale lengte

De focale lengte f is de afstand tussen het cameracentrum en het beeldvlak in millimeters uitgedrukt. Zie figuur 4.1.

Pixelafmetingen

De pixelafmetingen (p_x, p_y) zijn de breedte en de hoogte van een pixel in millimeters uitgedrukt. Deze zijn vooral van belang wanneer de pixels niet vierkant zijn. Zie figuur 4.3.

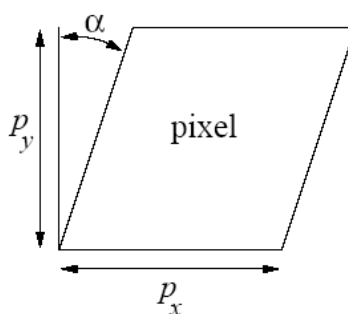
¹In OpenCV wordt een rotatievector gebruikt. Zie 10.2.1.



Figuur 4.2: De kalibratiematrix K zet de afbeeldingscoördinaten in R om naar pixelcoördinaten.

Pixelhellingshoek

De hellingshoek α van een pixel is van belang wanneer de pixels niet rechthoekig zijn. Deze hoek wordt weergegeven in figuur 4.3. Bij de meeste camera's zal deze hellingshoek bijna nul zijn.



Figuur 4.3: De pixelafmetingen en de hellingshoek

Beeldcentrum

Het beeldcentrum (c_x, c_y) stelt het optische centrum voor uitgedrukt in pixelcoördinaten. Zie figuur 4.2.

4.3 Projectiematrix

De uiteindelijke 3×4 -cameraprojectiematrix P die een 3D-punt in de wereld omzet naar pixelcoördinaten in een afbeelding ziet er als volgt uit:

$$P = K[R|t]$$

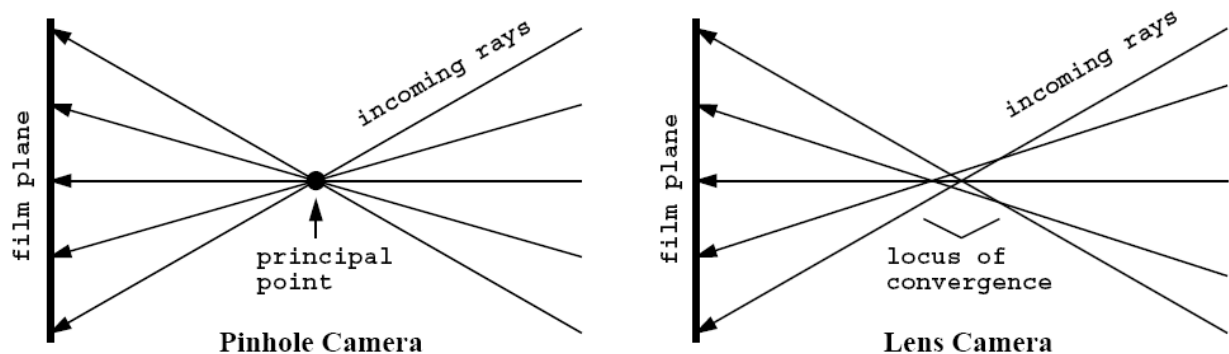
zodat:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{p_x} & (\tan \alpha) \frac{f}{p_y} & c_x \\ 0 & \frac{f}{p_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

4.4 Vervorming

Een echte camera werkt niet exact hetzelfde als voorgesteld wordt in het pinhole-model. Bij een echte camera gaan immers niet alle projecties exact door hetzelfde punt zoals men kan zien in figuur 4.4. Dit zorgt er voor dat het beeld vervormd wordt. Rechte lijnen worden dan in het beeld weergegeven als krommen. Dit kan ongedaan gemaakt worden indien er enkele parameters berekend worden die de vervorming beschrijven. Deze parameters zijn eveneens intrinsieke parameters want ze hangen niet af van de scène. Een uitvoerige beschrijving van beeldvervorming en hoe men deze berekent, kan gevonden worden in [Deb96].

Er wordt een onderscheid gemaakt tussen radiale en tangentiale vervorming. Het is echter de radiale vervorming die het meest en het duidelijkst voorkomt, vooral bij goedkope camera's en webcams.



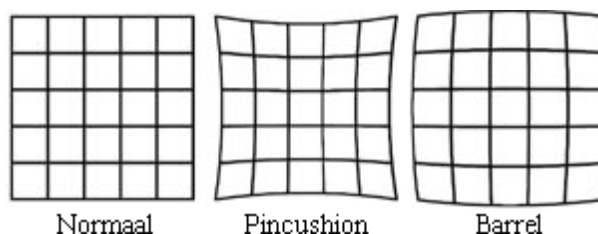
Figuur 4.4: De stralen gaan bij een echte camera niet exact door één punt. Hierdoor wordt het beeld vervormd.

4.4.1 Radiale vervorming

De radiale vervorming is zichtbaar als een verschuiving van de beeldpunten naar het vervormingscentrum toe (Pincushion) of ervan weg (Barrel). In het vervormingscentrum is de verschuiving nul. Radiale vervorming wordt gemodelleerd met twee of meer coëfficiënten.

4.4.2 Tangentiale vervorming

De tangentiale vervorming kan voor de meeste toepassingen genegeerd worden, omdat ze zo miniem is. Indien deze vervorming toch gemodelleerd wordt, worden er meestal twee



Figuur 4.5: Radiale vervorming

coëfficiënten voor gebruikt.

4.5 Camerakalibratie

Het kalibreren van een camera is eigenlijk het berekenen van de hierboven beschreven intrinsieke en extrinsieke cameraparameters. Dit kan op verschillende manieren gebeuren en hangt af van de reeds beschikbare cameragegevens, de omgeving waarin het gebeurt en de toepassing.

Hier wordt er gebruik gemaakt van een schaakbordpatroon als kalibratieobject (figuur 10.4). De hoekpunten van de zwarte en witte vlakken kunnen gebruikt worden als referentiepunten die op één lijn horen te liggen. Aan de hand van een aantal verschillende afbeeldingen van dit schaakbordpatroon, kunnen de intrinsieke -en vervormingsparameters berekend worden.

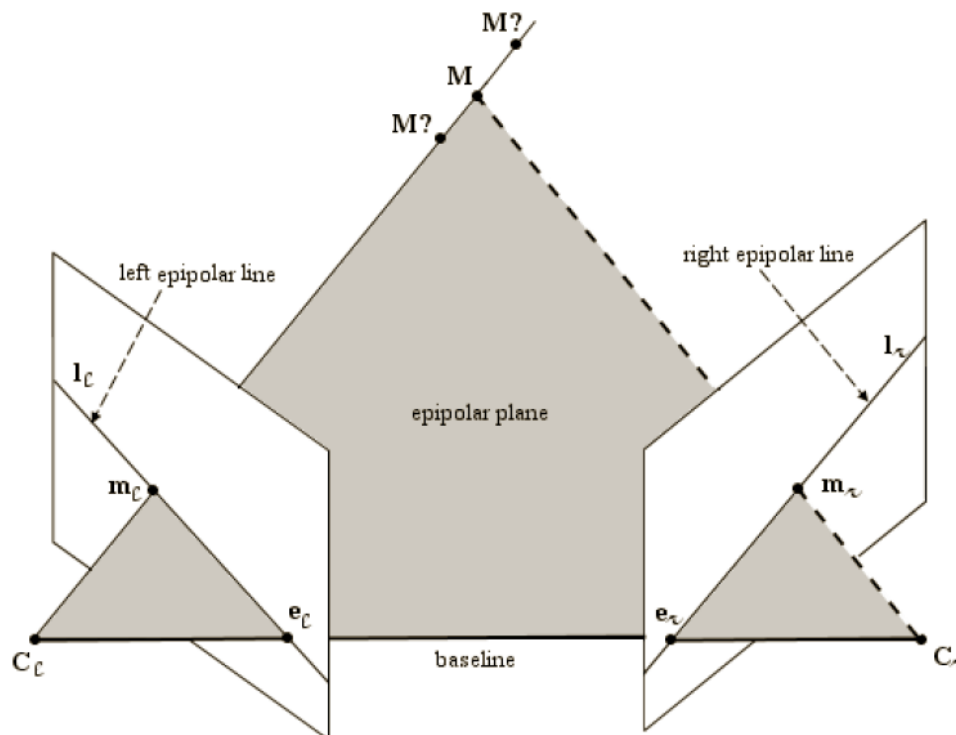
De extrinsieke parameters kunnen ook berekend worden met een schaakbordpatroon. Men moet er wel op letten dat voor elke camera het kalibratieobject zich in dezelfde positie bevindt, zodat elke camera met hetzelfde wereldcoördinatenstelsel werkt.

De volledige berekening van de cameraparameters wordt verduidelijkt in [Zha00].

4.6 Epipolaire geometrie

De epipolaire geometrie relateert de projecties van een punt in de scène naar twee verschillende camera's. Het verband tussen de projecties kan men zien in figuur 4.6. Er worden twee cameracentrums (C_l en C_r) met hun bijhorende beeldvlak gedefinieerd. Het 3D-punt M wordt geprojecteerd naar beide camera's in m_l en m_r . De epipolen e_l en e_r stellen de projecties van het andere cameracentrum voor. De lijn tussen de twee cameracentrums wordt ook wel de basislijn genoemd. Deze lijn snijdt niet noodzakelijk de beeldvlakken van de camera's. Het epipolaire vlak wordt gevormd door de twee cameracentrums en het 3D-punt M . Het epipolaire vlak snijdt in dit geval de beide beeldvlakken. Deze snijlijnen l_l en l_r noemen we de epipolaire lijnen. Ze lopen door de epipolen (e_l en e_r) en door de puntprojecties (m_l en m_r).

Wanneer we nu echter enkel de projectie van M in één camerastandpunt kennen, kunnen we de epipolaire lijn berekenen in het andere camerastandpunt. Het cameracentrum C_l en



Figuur 4.6: Epipolaire geometrie

het projectiepunt m_l vormen een rechte in de scène. De projectie van deze rechte in het andere camerastandpunt is de epipolaire lijn l_r . Deze lijn gaat bijgevolg door de epipool e_r en het projectiepunt m_r dat onbekend is.

4.6.1 Fundamentele matrix

Het verband tussen een projectiepunt in het ene camerastandpunt en de corresponderende epipolaire lijn in het andere camerastandpunt noemt men de epipolaire beperking en wordt uitgedrukt door de 3×3 fundamentele matrix F zodat:

$$m_r^T F m_l = 0$$

Het punt m_r ligt dus op de epipolaire lijn $l_r = F m_l$.

Om de epipolaire lijn l_l te berekenen uit m_r kunnen we F^T gebruiken. De fundamentele matrix kan berekend worden aan de hand van de camerakalibratiematrices zoals wordt beschreven in [HZ04].

Hoofdstuk 5

Markerdetectie

In dit hoofdstuk worden kort enkele beeldsegmentatietechnieken besproken. Met behulp van deze technieken kan het gefilmde onderwerp onderscheiden worden van de achtergrond van de scène. Daarna worden dan de markers die op het onderwerp bevestigd zijn gedetecteerd.

Om zo goed mogelijke resultaten te bekomen is het belangrijk om de scène, de markers en de kledij zodanig aan te passen, dat de gebruikte technieken optimaal benut kunnen worden. In [GW01] worden deze en nog andere technieken uitvoerig besproken.

Een vaak terugkomende beeldbewerking is thresholding. Hierbij vertrekt men vanuit een grijswaardenafbeelding. Naargelang de toepassing bepaalt men een goede thresholdwaarde. Alle grijswaarden die boven de thresholdwaarde liggen krijgen de kleurwaarde van wit en alle grijswaarden die onder de thresholdwaarde liggen krijgen de kleurwaarde van zwart. Het resultaat is dus een zwart-witbeeld.

5.1 Achtergrond

De invoer bij de volgende technieken is een origineel opgenomen camerabeeld.

5.1.1 Achtergrondsubtractie

Bij deze techniek wordt vooraf een afbeelding van de lege scène gemaakt en opgeslagen. Deze achtergrondafbeelding wordt dan later vergeleken met de opgenomen beelden waar het onderwerp zich wel in de scène bevindt. Dit wordt gedaan door de pixelwaarden van de twee beelden van elkaar af te trekken. Wanneer het resultaat hiervan voor een bepaalde pixel dichtbij nul ligt, is het zeer waarschijnlijk dat deze pixel tot de achtergrond behoort. Meer concreet kan men door het resulterende beeld te thresholden, een binair beeld bekomen waarbij de achtergrondpixels de waarde nul hebben, en de overige pixels de waarde 1. Dit binaire beeld kan men dan vermenigvuldigen met het originele opgenomen beeld. Het resultaat hiervan is dan dat enkel het onderwerp nog zichtbaar is, de achtergrond is dan

zwart.

Goedkope camera's, beeldruis, schaduwen en wisselende belichting beïnvloeden de nauwkeurigheid van deze techniek. Meestal zal men proberen om eerst de beeldruis te verwijderen door de afbeeldingen te vervagen.

5.1.2 Speciale achtergrond

Het onderwerp staat hierbij voor een volledig blauwe of groene achtergrond zoals ook vaak gebeurt bij televisie en films. Het is eigenlijk een geoptimaliseerde versie van de hierboven beschreven achtergrondsubstractie, waarbij het achtergrondbeeld uit slechts één kleur bestaat. Hierdoor wordt het nog wat eenvoudiger om het onderwerp van de achtergrond te onderscheiden.

5.2 Markers

De invoer bij deze technieken is een origineel opgenomen camerabeeld of een camerabeeld waar de achtergrond reeds uit verwijderd werd.

5.2.1 Intensiteit

Wanneer infraroodreflecterende markers gebruikt worden, is dit in de opgenomen camera-beelden duidelijk zichtbaar door witte stippen. Deze stippen hebben een hoge intensiteit en kunnen door middel van tresholding makkelijk uit het camerabeeld gehaald worden.

5.2.2 Kleur

Bij deze techniek geeft een waarde aan hoe ver een gevonden kleur in het beeld mag afwijken van de gezochte kleur om als correct beschouwd te worden. In het RGB-kleurenmodel¹ moet men dan per kleurkanaal deze waarde gaan vergelijken. In het HSV-kleurenmodel² moet men enkel de hue-waarde vergelijken omdat enkel deze waarde de kleur voorstelt. Door middel van tresholding krijgen we opnieuw een binair beeld waarbij enkel nog witte vlekken te zien zijn op de plaatsen van de gevonden markers.

5.2.3 Vorm

Naargelang de vorm van de markers kan men verscheidene technieken gebruiken. Bijna altijd zullen echter bollen als markers gebruikt worden. Deze bolvormige markers kunnen gedetecteerd worden door middel van een Hough-transformatie [Owe]. Hierbij wordt in een binair beeld gezocht naar cirkels met een bepaalde opgegeven straal.

¹Red-Green-Blue

²Hue-Saturation-Value

5.3 Exacte markerpositie bepalen

Het resultaat van de vorige technieken is een zwart-witbeeld waarin enkel nog witte vlekken te zien zijn op de plaatsen waar er zich mogelijk markers kunnen bevinden. Hieruit kunnen we indien nodig nog ruis verwijderen. De overblijvende vlekken stellen dan de markers voor.

De exacte pixelcoördinaten van de markers kunnen dan bepaald worden door per witte vlek het gemiddelde van de pixelposities te nemen.

Een andere mogelijkheid is opnieuw een Hough-transformatie te gebruiken en het middelpunt van de gevonden cirkels te nemen als markerposities.

5.4 Maximum aantal markers

Het is uiteraard niet de bedoeling dat we meer markers detecteren dan er werkelijk zijn aangebracht op het onderwerp. Daardoor kunnen we de bovenstaande technieken aanpassen door bijvoorbeeld meer ruis te verwijderen uit de beelden of door de treshholdwaarden aan te passen.

5.5 Temporele correlatie

5.5.1 Verwacht aantal markers

De bovenstaande technieken kunnen aangepast worden zodat ze zoeken naar een verwacht aantal markers. Dit verwachte aantal markers kan bepaald worden aan de hand van het aantal gevonden markers in het vorige frame. Het is natuurlijk niet de bedoeling dat we steeds hetzelfde aantal markers vinden voor een camera, maar hierdoor kan er wel voor gezorgd worden dat het aantal gevonden markers in de opeenvolgende frames niet te sterk fluctueert.

5.5.2 Markerposities in het vorige frame

Indien we veronderstellen dat de beelden aan een hoge frequentie opgenomen worden, zullen de markers zich in de opeenvolgende frames niet snel verplaatsen. Dit kunnen we uitbuiten door de gevonden markerposities te vergelijken met de markerposities in het vorige frame. Indien er op een bepaald moment te veel markers werden gevonden kunnen we de markers die niet in de buurt liggen van een marker in het vorige frame elimineren.

Hoofdstuk 6

3D-reconstructie

In dit hoofdstuk wordt besproken hoe we de 2D-markers van de verschillende camerastandpunten kunnen omzetten in 3D-markers in de scène. Dit probleem is opsplitsbaar in het correspondentieprobleem en het reconstructieprobleem.

6.1 Correspondentieprobleem

Wanneer een 3D-punt geprojecteerd wordt naar verschillende camerastandpunten resulteert dit in een aantal corresponderende 2D-punten. De invoer bij een motion capture opname bestaat echter uit camerabeelden waarin de projecties van meerdere 3D-markers voorkomen. We moeten kunnen bepalen welke 2D-markers in de camerabeelden gebruikt moeten worden om een bepaalde 3D-marker te reconstrueren. Dit wordt het correspondentieprobleem genoemd.

Bij het detecteren van de 2D-markers in de beelden kan het gebeuren dat er op bepaalde posities 2D-markers gevonden worden die er in werkelijkheid niet zijn. Ook worden niet alle 2D-markers in de beelden teruggevonden omdat er zich vaak oclusies voordoen. Deze verschijnselen vermoeilijken het correspondentieprobleem.

De correlatiemethodes die beschreven worden zijn vooral belangrijk bij de 3D-reconstructie in het eerste frame van een opname. Dan is het uitermate belangrijk dat alle 3D-markers kunnen gereconstrueerd worden. Deze markers moeten echter later getraceerd worden in de volgende frames zodat we de verschillende markertrajecten van elkaar kunnen onderscheiden. De 3D-markers die ontbreken in het eerste frame kunnen bijgevolg niet getraceerd worden en zijn verloren voor de ganse opname [HFP⁺01].

6.1.1 Epipolaire lijnen

Zoals reeds eerder gezien werd, kan men met de fundamentele matrix en een projectiepunt in het ene camerabeeld een epipolaire lijn in een ander camerabeeld berekenen. Deze lijn stelt de straal voor die vertrekt vanuit de positie van de andere camera en die door het

beschouwde projectiepunt in het beeldvlak gaat. De 2D-markers die dichtbij of op deze lijn liggen, stellen de mogelijk 2D-markers voor die overeenkomen met de eerder gekozen 2D-marker in het andere camerastandpunt. Indien er slechts één 2D-marker gevonden wordt die binnen een vooraf bepaalde afstand van de epipolaire lijn ligt, kunnen we deze gebruiken om de 3D-marker te reconstrueren [HFP⁺01].

6.1.2 Herprojectie

Indien we reeds meerdere corresponderende 2D-markers hebben gevonden, kunnen we deze gebruiken om een eerste positie van de 3D-marker te berekenen. Deze 3D-positie kunnen we dan herprojecteren in de overige camerastandpunten. Indien er zich slechts één 2D-marker in de buurt van dit geprojecteerde punt bevindt, kunnen we deze 2D-marker eveneens gebruiken om een nauwkeurigere positie voor de 3D-marker te berekenen. Door het vinden van deze nieuwe 2D-marker verhoogt de kans aanzienlijk dat de eerder gevonden 2D-markers corresponderen [HFP⁺01].

6.1.3 Foutieve correlaties

Het kan toch voorkomen dat er één of meerdere foutieve 2D-markers gekozen worden om een 3D-marker te reconstrueren. De oorzaken hiervan kunnen zijn:

- **Onnauwkeurige camerakalibratie:** Wanneer de camerakalibratiematrix berekend werd met data die veel ruis bevat, zal ook de fundamentele matrix, die berekend wordt uit de camerakalibratiematrices onnauwkeurig zijn. Dit leidt bijgevolg tot epipolaire lijnen die ver afwijken van de correcte epipolaire lijn. Dit kan er voor zorgen dat de verkeerde 2D-marker gekozen wordt.
- **Foutieve 2D-markerdetectie:** Als we de epipolaire lijn berekenen van een 2D-marker die eigenlijk in werkelijkheid niet aanwezig is, kan het toch gebeuren dat er een correspondentie voor deze foutieve 2D-marker gevonden wordt.

Deze problemen kunnen verholpen worden door enkel de 3D-markers te aanvaarden die gereconstrueerd werden met minstens drie 2D-markers. Een bijkomende mogelijkheid is om enkel de gereconstrueerde 3D-markers te aanvaarden, waarvan de gemiddelde afwijking van de herprojecties op de gebruikte 2D-markers niet groter is dan een bepaalde vooropgestelde waarde.

6.1.4 Resultaat

Na het vinden van de corresponderende 2D-markers is het mogelijk om een aantal 3D-markers te reconstrueren. Er kunnen zich nu drie gevallen voordoen:

1. **Het aantal gereconstrueerde 3D-markers is gelijk aan het aantal werkelijk aangebrachte markers:** Dit betekent dat de 3D-reconstructie hoogstwaarschijnlijk correct is verlopen.
2. **Het aantal gereconstrueerde 3D-markers is kleiner dan het aantal werkelijk aangebrachte markers:** We kunnen proberen extra 3D-markers te reconstrueren door de parameters van de correlatiemethoden te versoepelen. Dit betekent dat we verder van de epipolaire lijn en de herprojectie naar 2D-markers gaan zoeken. Een andere mogelijkheid is om ook 3D-markers toe te staan die gereconstrueerd werden met slechts twee 2D-markers.
3. **Het aantal gereconstrueerde 3D-markers is groter dan het aantal werkelijk aangebrachte markers:** Er komen nu dus zeker foutieve 3D-markers voor. In het volgende hoofdstuk zullen we zien dat deze foutieve 3D-markers niet blijven terugkomen in de opeenvolgende frames, omdat ze niet lang getraceerd kunnen worden. Op deze manier kunnen we deze foutieve 3D-markers wegfilteren.

6.2 Reconstructieprobleem

Een 3D-punt is eenvoudig te projecteren naar de beeldvlakken van de camera's met behulp van de projectiematrix. Het 3D-punt reconstrueren aan de hand van de 2D-projecties is een andere opgave. Dit wordt het reconstructieprobleem genoemd.

Het reconstructieprobleem kan in dit geval worden opgelost met stereotriangulatie. Hierbij worden de projectiepunten in twee camerastandpunten gebruikt. Er zijn ook technieken die de beschikbare projecties in meerdere camerastandpunten in acht nemen.

6.2.1 Twee camerastandpunten

Aan de hand van een projectiepunt kan een straal in de scène berekend worden die vertrekt in het cameracentrum en door het projectiepunt gaat. Indien dit gedaan wordt voor twee corresponderende 2D-markers, resulteert dit in twee stralen in de scène die elkaar zouden moeten snijden op de plaats van de 3D-marker.

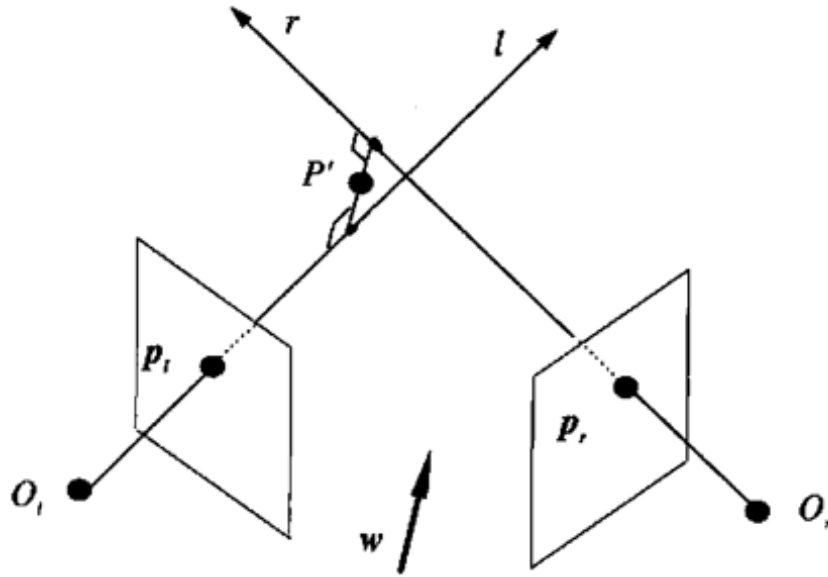
Dit komt in de praktijk echter bijna nooit voor omwille van de ruis die optreedt bij de camerakalibratie en de 2D-markerdetectie. De volgende methodes proberen het gezochte snijpunt te benaderen.

Middelpuntsmethode

Zoals men kan zien in figuur 6.1 kruisen de stralen elkaar. De middelpuntsmethode [Beb] berekent het punt in de scène dat het dichtst bij de beide stralen ligt. Dit punt is het middelpunt van het lijnstuk dat loodrecht staat op de beide stralen.

Vanuit het perspectief van de linkercamera kunnen de stralen als volgt beschreven worden:

$$l = ap_l$$



Figuur 6.1: Stereo triangulatie: middelpuntsmethode

$$r = bR^T p_r + T$$

De eindpunten van het lijnstuk dat de beide stralen verbindt zijn:

$$P_1 = a_0 p_l$$

$$P_2 = b_0 R^T p_r + T$$

De vector die loodrecht op beide stralen staat is:

$$w = p_l \times R^T p_r$$

Het gezochte lijnstuk staat loodrecht op de beide stralen en verbindt ze. Daarom kunnen we stellen dat:

$$a_0 p_l - b_0 R^T p_r - T + c_0 w = 0$$

ofwel:

$$a_0 p_l - b_0 R^T p_r + c_0 (p_l \times R^T p_r) = T$$

Door dit stelsel van drie vergelijkingen in drie onbekenden op te lossen bekommen we a_0 en b_0 waarmee we P_1 en P_2 kunnen berekenen. Het gezochte middelpunt van dit lijnstuk is dan:

$$P' = (P_1 + P_2)/2$$

In plaats van zomaar het middelpunt te kiezen, kan men bijvoorbeeld ook rekening houden met de afstanden van de camera's tot het lijnstuk. Deze afstanden kunnen berekend worden met de gevonden parameters a_0 en b_0 . Er wordt dan een punt op het lijnstuk gekozen dat dichterbij de straal van de meest nabije camera ligt.

Hartley-Sturmmethode

Door de aanwezige ruis voldoen de corresponderende 2D-markers niet meer aan de epipolaire beperking:

$$x_r^T F x_l = 0$$

In [HS97] gaat men de optimale projectiepunten x'_l en x'_r berekenen zodat de epipolaire beperking wel nog geldt. Dit wordt gedaan door de volgende kostfunctie te minimaliseren:

$$d(x_l, x'_l)^2 + d(x_r, x'_r)^2$$

waarbij $d(*, *)$ de Euclidische afstand tussen twee punten voorstelt.

De optimale projectiepunten x'_l en x'_r liggen respectievelijk op de epipolaire lijnen l_l en l_r . De loodrechte projecties van de originele 2D-markers op de epipolaire lijnen leveren de optimale punten x'_l en x'_r . Daarom kunnen we de kostfunctie ook schrijven als:

$$d(x_l, l_l)^2 + d(x_r, l_r)^2$$

waarbij $d(*, *)$ de minimale afstand tussen het projectiepunt en de epipolaire lijn voorstelt. l_l en l_r kunnen alle corresponderende epipolaire lijnen voorstellen. De methode om de kostfunctie te minimaliseren is:

1. Parametriseer alle mogelijke epipolaire lijnen in het linker camerabeeld zodat we een bepaalde epipolaire lijn kunnen definiëren als $l_l(t)$.
2. Bereken het punt op deze epipolaire lijn $l_l(t)$ dat het dichtst bij de originele 2D-marker ligt.
3. Bereken voor dit punt de epipolaire lijn $l_r(t)$ in het andere camerabeeld.
4. Druk de kostfunctie uit als een functie in t .

$$d(x_l, l_l(t))^2 + d(x_r, l_r(t))^2$$

5. Zoek de waarde van t die deze functie minimaliseert.

Om de correcte waarde voor t te vinden moet er nu enkel het minimum bepaald worden van een polynomiale functie van de zesde graad. De parametrisatie van de epipolaire lijnen en de berekening van t worden expliciet beschreven in [HS97]. Als t bepaald is, zijn de optimale projectiepunten x'_l en x'_r ook gevonden. Omdat deze punten wel voldoen aan de epipolaire beperking zullen de stralen die erdoor lopen elkaar in de scène snijden. De positie van de 3D-marker kan vervolgens berekend worden met bijvoorbeeld de middelpuntsmethode.

6.2.2 Meerdere camerastandpunten

Middelpuntsmethode

Voor elk paar corresponderende projectiepunten berekenen we een 3D-punt aan de hand van de middelpuntsmethode voor twee camerastandpunten. Van de resulterende 3D-punten nemen we vervolgens het gemiddelde als positie voor de 3D-marker.

Minimalisering van de fout bij herprojectie

Deze methode neemt alle corresponderende 2D-markers tegelijk in acht en probeert deze kostfunctie te minimaliseren:

$$\sum_{i=0}^n (P_i X - x_i)^2$$

n stelt het aantal corresponderende punten voor, P is de cameraprojectiematrix die de 3D-marker X herprojecteert en x_i is de 2D-marker in het camerabeeld.

Deze functie kan bijvoorbeeld geminimaliseerd worden met de kleinste-kwadratenmethode van Levenberg-Marquardt [DeT]. Als initiële gok kan de 3D-marker gebruikt worden die gereconstrueerd werd uit twee van de corresponderende 2D-markers.

Hoofdstuk 7

Markertracering

Wanneer men het heeft over markertracering, bedoelt men het berekenen van het ganse traject van de reeds gereconstrueerde 3D-markers van het eerste frame. Het resultaat is dat men van elke 3D-marker kan zien waar die zich op een bepaald moment bevindt. Het algoritme om de markers te traceren wordt beschreven in [HFP⁺01] en is gebaseerd op algoritmes die gebruikt worden bij particle tracking velocimetry (PTV). Dit is een beeldverwerkingsmethode waarbij men snelheidsvelden van deeltjes (particles) berekent uit 2D -en 3D-beeldsequenties. Dit wordt gedetailleerd beschreven in [MDP93].

7.1 Marker datastructuur

Om de trajecten van de markers op te slaan en zodoende alle 3D-markers te kunnen identificeren, is er nood aan een degelijke datastructuur.

Voor elke camera wordt er een lijst van alle gevonden 2D-markers per frame bijgehouden. De gereconstrueerde 3D-markers worden eveneens opgeslagen in een lijst.

7.1.1 2D-marker

Een 2D-marker bevat verwijzingen naar een 2D-marker in het vorige en in het volgende frame. Verder wordt ook bijgehouden of de 2D-marker reeds gebruikt is bij een 3D-reconstructie. Indien dit het geval is, wordt ook een verwijzing bijgehouden naar de betreffende 3D-marker.

7.1.2 3D-marker

Net zoals bij de 2D-markers worden ook hier verwijzingen bijgehouden naar een marker in het vorige en in het volgende frame. Alle 2D-markers die aan de 3D-marker gerelateerd worden, worden eveneens opgeslagen. Hierbij wordt er wel een onderscheid gemaakt tussen de twee 2D-markers die gebruikt werden om de positie van de 3D-marker te berekenen, en de andere 2D-markers die gevonden werden door middel van de herprojectie van de 3D-marker in de andere camerastandpunten.

7.2 Glijdende venster van vier frames

Dit algoritme wordt gebruikt om 2D-markers of 3D-markers te traceren. Het opzet is eenvoudig: de verplaatsingen van de markers worden gevolgd in een venster van vier opeenvolgende frames.

Bij het traceren van markers wordt er meestal verondersteld dat:

- er slechts kleine verplaatsingen gebeuren in de opeenvolgende frames. Hierdoor kan men in een gebied rond de positie in het vorige frame zoeken.
- de verplaatsingen van de markers vloeiend gebeuren. Hierdoor kan men de posities van de gevonden markers in de vorige frames gebruiken om de positie in het huidige frame te bepalen en de positie in het volgende frame te voorspellen.

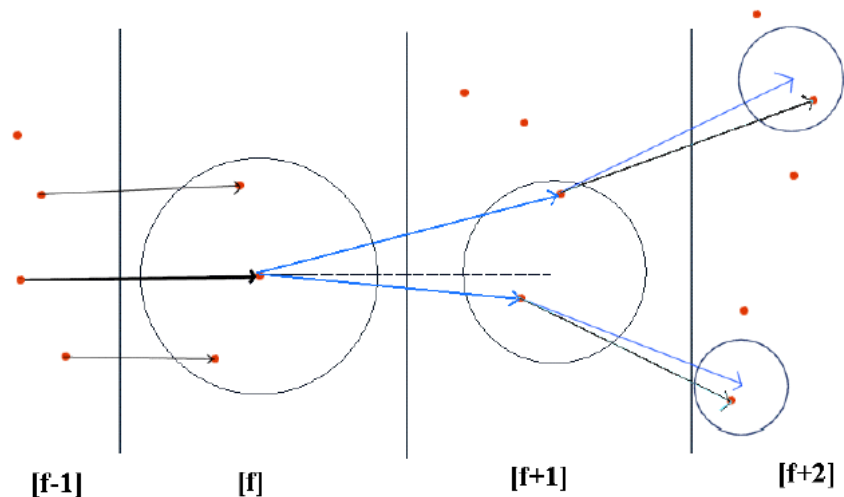
Wanneer we een marker willen traceren van het huidige frame naar het volgende frame, moeten we eerst een verplaatsingsvector bepalen om de positie in het volgende frame te kunnen voorspellen. De berekening van deze verplaatsingsvector kan op drie verschillende manieren gebeuren. Dit hangt af van de beschikbaarheid van verwijzingen naar markers in het vorige frame.

- **Eigen verplaatsingsvector:** De te traceren marker heeft zelf een verwijzing naar een marker in het vorige frame. De verplaatsingsvector is nu het verschil in markerposities in het huidige en het vorige frame.
- **Correlatie:** In dit geval heeft de te traceren marker geen verwijzing naar een marker in het vorige frame. We gaan er nu van uit dat markers in de buurt van de te traceren marker zich op ongeveer dezelfde manier verplaatsen. Daarom wordt in het huidige frame een correlatieomgeving ingesteld op de positie van de te traceren marker. Voor elke marker die zich in deze omgeving bevindt en ook een verwijzing heeft naar het vorige frame, kunnen we een verplaatsingsvector berekenen. Het gemiddelde van deze verplaatsingsvectoren gebruiken we als verplaatsingsvector voor de te traceren marker.
- **Willekeurig:** Omdat we geen informatie hebben om een degelijke verplaatsingsvector te berekenen, kunnen we enkel een vooropgestelde verplaatsingsvector gebruiken. Deze kan aan een verwachtingspatroon beantwoorden of statistisch bepaald zijn.

De gevonden verplaatsingsvector passen we nogmaals toe op de huidige positie van de marker. Zo bekomen we een voorspelling van de nieuwe positie. Gecentreerd op deze positie definiëren we vervolgens een zoekomgeving. Als er zich binnen deze omgeving slechts één marker bevindt, besluiten we dat dit de juiste marker is.

Wanneer er echter meerdere markers gevonden worden in de zoekomgeving, gaan we nog een frame verder kijken om de correcte marker te bepalen. Hiervoor bepalen we de verplaatsingsvectoren van de marker in het huidige frame naar elke gevonden marker in de zoekomgeving. Aan de hand van deze vectoren en de eerder berekende verplaatsingsvector

voorspellen we de markerposities in het daarop volgende frame. Hierbij houden we wel rekening met het verschil van de opeenvolgende vectoren zodat de eventuele versnelling van de markers gemodelleerd wordt. Op deze posities worden nieuwe zoekomgevingen ingesteld. Ditmaal behouden we enkel de omgevingen waarin slechts één marker gevonden wordt. Indien dit het geval is voor meerdere zoekomgevingen, kiezen we de marker waarvoor de opeenvolgende vectoren de gelijkmatigste versnelling vertonen. Dit wordt duidelijk geïllustreerd in figuur 7.1.



Figuur 7.1: vier frames met de correlatie -en zoekomgevingen

7.3 Globale traceringsalgoritme

Het is belangrijk dat alle 3D-markers in het eerste frame aanwezig zijn en gereconstrueerd kunnen worden. Indien dit niet het geval is, zijn de ontbrekende markers ook verloren voor de andere frames.

Zoals eerder gezegd is er nu voor elke 3D-marker informatie beschikbaar over welke 2D-markers gebruikt werden om tot deze reconstructie te komen. Deze 2D-markers kunnen we nu traceren naar het volgende frame aan de hand van de gevonden 2D-markers in de camerabeelden. Indien dit lukt, kan er opnieuw een 3D-reconstructie uitgevoerd worden met deze getraceerde 2D-markers. De nieuwe 3D-marker kan daarna gelinkt worden aan de 3D-marker uit het vorige frame.

Door deze methode te gebruiken wordt het 3D-reconstructieproces versneld.

Wanneer niet alle 3D-markers op deze manier kunnen getraceerd worden in een frame, gaat men achtereenvolgens de volgende stappen proberen:

- **Binoculaire reconstructie:** De eerder gestelde voorwaarde om een 3D-marker pas te aanvaarden wanneer die met minstens drie 2D-markers gerelateerd wordt, vervalt nu. Alle 3D-markers die binoculair (aan de hand van twee 2D-markers) gereconstrueerd kunnen worden, voegen we toe aan de markerlijst.

- **3D-tracering:** Alle 3D-markers uit het vorige frame die nog geen verwijzing hebben naar het huidige frame gaat men traceren zoals beschreven in 7.2. Hierdoor kan men hopelijk nog een aantal anonieme 3D-markers identificeren. Indien dit de eerste maal niet succesvol is, kan men het nog eens proberen met een grotere zoek -en correlatieomgeving.
- **3D-markerpositie voorspellen:** Indien er nog steeds 3D-markers ontbreken, worden deze berekend aan de hand van hun posities in de drie vorige frames. Hierbij wordt rekening gehouden met de eventuele versnelling van de 3D-marker.

Hoofdstuk 8

Skeleton fitting

De gereconstrueerde 3D-markers vormen een soort van uitwendig skelet voor het onderwerp. Het doel van skeleton fitting is om aan de hand van de 3D-markers een inwendig skelet te berekenen.

In [SPB⁺98] en [HFP⁺01] worden twee methodes beschreven die dit inwendige skelet berekenen, namelijk de lokale en de globale methode.

De lokale methode probeert voor elk gewricht apart een positie te bepalen, terwijl bij de globale methode alle gewrichten ineens beschouwd worden. De globale methode heeft nood aan een goede initialisatie. Daarom is het interessant om de resultaten van de lokale methode als invoer voor de globale methode te gebruiken.

Allereerst wordt het model beschreven dat een skelet voorstelt.

8.1 Skeletmodel

Het skeletmodel zoals beschreven in [DPF⁺99, SPB⁺98, FGP⁺98] bestaat uit een aantal gewrichten die met elkaar verbonden worden door botten. De gewrichten worden ondergebracht in een hiërarchische boomstructuur. Naargelang het onderwerp kan deze boomstructuur verschillen.

Bij een menselijk skelet bevinden de handen, de voeten en het hoofd zich onderaan in de boomstructuur en stellen ze de uiteinden van het skelet voor. Om de duidelijkheid te bevorderen, worden ze verder ook gewrichten genoemd.

De botten van het skelet hebben een vaste lengte en elk gewricht heeft een aantal vrijheidsgraden van beweging. Voor elke vrijheidsgraad wordt een rotatiehoek bijgehouden. De lengtes van de botten en de rotatiehoeken bepalen de positie van het skelet in een frame. Dit kan geformuleerd worden als:

$$P = P_{skeleton} = [P_{beweging}, P_{init}]$$

De vaste begintoestand van het skelet wordt aangegeven door P_{init} . Dit is tevens de toestand van het skelet in rust en bevat de rotaties en translaties die elk gewricht positioneren ten opzichte van zijn oudergewricht. Het oudergewricht is het gewricht dat net hoger staat

in de boomstructuur.

$P_{beweging}$ is verschillend voor elk frame en bevat de rotatiehoeken voor elke vrijheidsgraad die het skelet van zijn begintoestand naar de huidige toestand brengt.

Aan de hand van de waarden in P_{skelet} kan voor elk gewricht een lokale transformatiematrix opgesteld worden. Als voorbeeld wordt de elleboog beschouwd. De elleboog heeft twee vrijheidsgraden, namelijk de buiging en de draaiing. De lokale transformatiematrix voor de elleboog wordt bijgevolg:

$$D^{elleboog} = D_{rot}^{draaiing} D_{init}^{draaiing} D_{rot}^{buiging} D_{init}^{buiging}$$

Zoals men kan zien, wordt elke vrijheidsgraad voorgesteld door een matrix van de vorm:

$$D^{vrijheidsgraad} = D_{rot} D_{init}$$

De meeste matrices D_{init} roteren enkel het coördinatenstelsel van de vorige naar de huidige vrijheidsgraad en zien er uit als:

$$D_{init} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

D_{rot} stelt dan de lokale rotatie α rond de Z-as voor:

$$D_{rot} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$D_{init}^{buiging}$ is de eerste vrijheidsgraad van de elleboog en ziet er daarom anders uit. Er moet hier namelijk ook rekening gehouden worden met de verplaatsing van het gewricht ten opzichte van zijn oudergewricht. Deze verplaatsing modelleert in dit geval de lengte van het bot tussen de elleboog en de schouder.

De lokale transformatiematrix van een gewricht is uiteindelijk van de vorm:

$$D_l = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De globale transformatiematrix van een gewricht wordt berekend door recursief de lokale matrices van de voorgaande gewrichten in de boomstructuur te vermenigvuldigen met de lokale transformatiematrix:

$$D_g = \prod_i D_{l,i}(P)$$

De lokale transformatiematrices $D_{l,i}$ van de gewrichten zijn hierbij afhankelijk van de status van het skelet P .

8.2 Lokale methode

De lokale methode maakt gebruik van de volgende twee observaties:

- De afstanden tussen de markers die gelegen zijn op hetzelfde bot zijn ongeveer constant.
- De afstand van een marker tot een aangelegen gewricht is ongeveer constant.

De volgende stappen worden doorlopen:

1. Opdeling en associatie van de markers
2. Berekening van de posities van de gewrichten
3. Schalering en positionering van het skelet

8.2.1 Opdeling en associatie van de markers

Om later de posities van de gewrichten te kunnen berekenen, moet eerst elke marker met een lichaamsdeel geassocieerd worden. Dit lichaamsdeel kan een bot of een uiteinde van het skelet zijn.

De associatie kan uiteraard volledig manueel gebeuren door voor elke marker het bijhorende lichaamsdeel aan te duiden. Het is echter mogelijk om dit proces te automatiseren.

Opdeling

Voor elk paar markers wordt de gemiddelde afstand over alle frames berekend. Daarna worden de minimale en de maximale afwijking op dit gemiddelde bepaald. De markers worden vervolgens in verzamelingen opgedeeld zodat voor elk paar markers in een verzameling geldt dat het verschil tussen de minimale en de maximale afwijking niet meer bedraagt dan een vooropgestelde waarde. Door deze vooropgestelde waarde interactief aan te passen kunnen we een degelijke opdeling van de markers bekomen.

Naargelang de afwijking die een marker vertoont ten opzichte van de andere markers in een verzameling, kan er een gewicht aan de marker toegekend worden. Dit gewicht kan later gebruikt worden bij het bepalen van een lokaal coördinatenstelsel voor een bot.

Associatie

De verzamelingen van markers kunnen nu manueel geassocieerd worden met een lichaamsdeel. Indien de berekende verzamelingen niet correct zijn, kunnen er nadien nog manuele aanpassingen gebeuren.

8.2.2 Berekening van de posities van de gewrichten

De positie van een uiteinde van het skelet wordt bepaald door de gemiddelde positie te berekenen van de geassocieerde markers. Hierbij wordt rekening gehouden met de gewichten die aan de markers toegekend werden.

De posities van de overige gewrichten worden bepaald door rotatiecentrums te berekenen.

Lokaal coördinatenstelsel van een bot

De positie en de oriëntatie van een bot kunnen in de ruimte bepaald worden door minstens drie markers. Hiermee kunnen we een lokaal coördinatenstelsel voor een bot opstellen. Het coördinatenstelsel wordt gevormd door een oorsprong en twee assen te kiezen. De derde as wordt berekend uit de andere twee assen zodat er een rechtshandig coördinatenstelsel ontstaat.

De oorsprong van het coördinatenstelsel is de gewogen gemiddelde positie van de markers. De X-as wordt bepaald door de marker die het verst van de oorsprong verwijderd is. De Y-as wordt dan weer bepaald door de marker die het verst van de X-as verwijderd is. Bij het berekenen van de afstanden van de markers tot de oorsprong en de X-as, wordt er eveneens rekening gehouden met de markergewichten.

Een markergewicht geeft aan hoe stabiel een marker zich gedraagt ten opzichte van de andere markers. Door dit in acht te nemen zal het resulterende coördinatenstelsel zich zo stabiel mogelijk gedragen en minder te lijden hebben onder ruis.

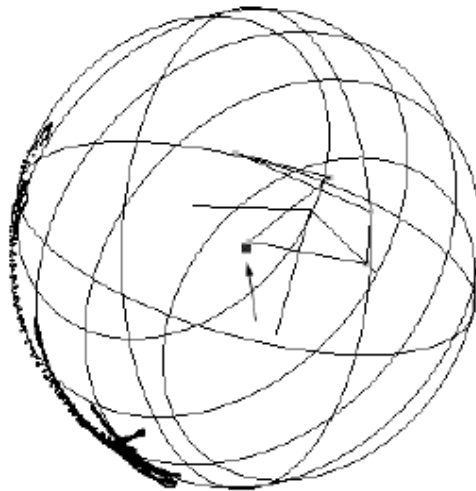
De oorsprong van het coördinatenstelsel wordt in elk frame opnieuw berekend omdat de markers zich kunnen verplaatsen. De keuze van de markers die de assen bepalen, gebeurt slechts eenmalig. Hiervoor worden de markerposities in het eerste frame gebruikt.

Relatieve markertrajecten

Er wordt verondersteld dat de afstand tussen een marker en een aanliggend gewricht constant is. Hieruit volgt dat de markers van aanliggende botten zich op een bol rond het gewricht bewegen. Dit gewricht verbindt dus de aanliggende botten.

Stel dat een bot B geassocieerd wordt met de markers a , b en c . Met deze markers kan in elk frame een lokaal coördinatenstelsel bepaald worden. Op een aanliggend bot bevindt zich de marker m . In elk frame wordt de positie van m in het lokale coördinatenstelsel berekend. Deze posities vormen het traject dat de marker m aflegt ten opzichte van het bot B . Dit traject zal op een bol lopen en het middelpunt van deze bol stelt het rotatiecentrum van de marker voor (figuur 8.1).

Indien een gewricht gemodelleerd wordt zodat het slechts één vrijheidsgraad heeft, is het voldoende om twee markers op de aanliggende botten te hebben. De markers bewegen zich nu niet op een bol maar op een cirkel rond het rotatiecentrum (figuur 8.2). Het lokale coördinatenstelsel wordt berekend aan de hand van de twee markers op het bot en de marker op het aanliggende bot waarvoor het rotatiecentrum gezocht wordt.



Figuur 8.1: Aan de hand van het relatieve markertraject wordt een rotatiecentrum bepaald.

Rotatiecentrum

Het rotatiecentrum van een marker wordt bepaald door het middelpunt te vinden van het traject dat de marker aflegt in het lokale coördinatenstelsel. Door het optreden van ruis zal dit traject niet over een perfecte bol lopen. Daarom wordt er naar de straal r en het middelpunt (x, y, z) gezocht die de volgende kostfunctie minimaliseren:

$$d(r, x, y, z) \times \text{gewicht}(x_0, y_0, z_0)$$

waarbij:

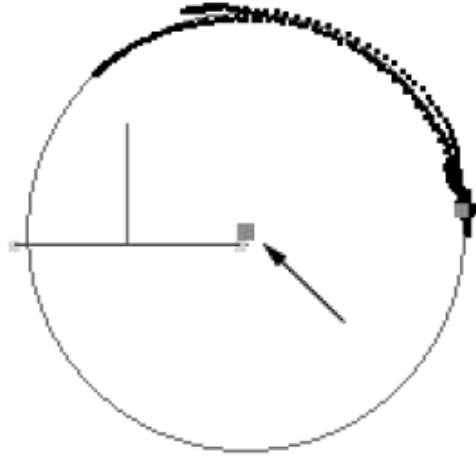
$$d(r, x, y, z) = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} - r$$

Zoals men kan zien wordt er aan elke markerpositie (x_0, y_0, z_0) in het traject een gewicht toegekend. De gewichten zijn omgekeerd evenredig met het aantal keer dat de marker voorkomt in een bepaald gedeelte van het traject. Ze worden als volgt berekend:

1. Bereken de kleinste kubus die alle posities in het traject bevat.
2. Verdeel de kubus in kleinere dealkubussen door elke dimensie in een aantal stukken te delen.
3. Tel het aantal markerposities n_i in elke dealkubus.
4. Het gewicht van een marker is dan: $1/n_i$ waarbij i de dealkubus is waarin de marker zich bevindt.

In figuur 10.6 wordt het traject van een marker getoond waarbij enkel de dealkubussen, die op het traject liggen, zichtbaar zijn.

Het gevonden rotatiecentrum staat nog steeds relatief ten opzichte van het aanliggende bot. Door de omzetting te maken naar globale coördinaten, verkrijgen we de positie van het rotatiecentrum in elk frame.



Figuur 8.2: Speciaal geval waarbij een aanliggend bot slechts twee markers bevat. De marker beweegt zich nu op een cirkel in plaats van op een bol.

Positie van het gewricht

Indien twee aanliggende botten beide geassocieerd worden met drie markers, kunnen we op de bovenstaande manier zes rotatiecentrums met hun bijhorende straal berekenen. De uiteindelijke positie van het gewricht wordt dan het gewogen gemiddelde van de rotatiecentrums. Het gewicht voor een rotatiecentrum wordt bepaald door:

$$gewicht_{rotatiecentrum} = \frac{gewicht_{marker}}{straal}$$

waarbij $gewicht_{marker}$ de stabiliteit van de marker voorstelt zoals bij het berekenen van een lokaal coördinatenstelsel. Het gewicht is omgekeerd evenredig met de straal van de bol omdat de fout bij de berekening van het middelpunt van een kleine bol globaal gezien ook klein zal zijn. De uiteindelijke positie van het gewricht wordt dan gegeven door:

$$x_G = \frac{\sum_{rotatiecentrums} (x_{rotatiecentrum} \times gewicht_{rotatiecentrum})}{\sum_{rotatiecentrums} gewicht_{rotatiecentrum}}$$

8.2.3 Schalering en positionering van het skelet

Nu de posities van de gewrichten in alle frames gekend zijn, kunnen we proberen een skeletmodel in elk frame te positioneren. Om dit te kunnen doen, moet het skeletmodel eerst geschaleerd worden zodat het overeenkomt met het gefilmde onderwerp. Dit houdt concreet in dat we de lengtes van de botten gaan bepalen.

Dit kan het eenvoudigst gedaan worden door de gemiddelde afstanden tussen de gewrichten te gebruiken als lengtes voor de botten. Aan de hand van deze botlengtes wordt het skelet in elk frame gepositioneerd zodat de gewrichten van het model zo goed mogelijk overeenkomen

Hierbij is wel een goede initialisatie nodig om tot een oplossing te komen. De modelparameters die bekomen werden met de vorige methode kunnen hiervoor gebruikt worden.

8.3 Globale methode

Bij de globale methode wordt het skeletmodel gepositioneerd zodat de gewrichten zich op een bepaalde vaste afstand van de markers bevinden. Het gebruikte markermodel houdt dus in dat een marker zich op een bol rond een aanliggend gewricht beweegt. De functie die geminimaliseerd moet worden, berekent het verschil tussen de gemodelleerde en de waargenomen afstanden van de markers tot hun bijhorende gewrichten.

8.3.1 Zonder initialisatie

Omdat we ons niet kunnen baseren op beschikbare gegevens, moeten naast alle skeletparameters ook nog de afstanden tussen de markers en de gewrichten bepaald worden. Daarom wordt de functie geminimaliseerd voor alle frames tegelijk. De parameters die gezocht worden zijn:

- de lengtes van de botten (deze blijven constant voor alle frames)
- de afstanden van de markers tot de aanliggende gewrichten (deze blijven constant voor alle frames)
- voor elk frame: de globale positie van het skelet in de scène
- voor elk frame: de globale oriëntatie van het skelet in de scène
- voor elk frame: de rotatiehoeken voor de vrijheidsgraden van alle gewrichten

Als resultaat krijgen we de positie van het skelet in alle frames, samen met de optimale waarden voor de botlengtes en de afstanden tussen de markers en de gewrichten. Vanwege het hoge aantal gezochte parameters en het ontbreken van enige initialisatie is deze methode traag en komt ze niet altijd tot een oplossing.

8.3.2 Met initialisatie

De resultaten van de lokale methode worden nu gebruikt als initialisatie voor de globale methode. De botlengtes zijn dus reeds bekend en de afstanden van de markers tot hun aanliggende gewrichten kunnen we berekenen. Voor elk gewricht worden deze afstanden opgezocht in alle frames en de medianen ervan worden als initiële afstanden gebruikt.

Het skelet kan nu in een frame gepositioneerd worden door middel van de kleinste-kwadratenmethode zodat de verschillen van de werkelijke afstanden met de gemodelleerde afstanden geminimaliseerd worden. Als initialisatie kunnen de skeletparameters gebruikt worden die bekomen werden met de lokale methode.

De gezochte parameters voor een frame zijn:

- de globale positie van het skelet in de scène
- de globale oriëntatie van het skelet in de scène
- de rotatiehoeken voor de vrijheidsgraden van alle gewrichten

Nadat het skelet in elk frame gepositioneerd werd, kunnen de afstanden van de markers tot hun aanliggende gewrichten herberekend worden. Het skelet kan nu opnieuw gepositioneerd worden aan de hand van deze afstanden. Dit kan men nog enkele malen herhalen totdat de afstanden als voldoende nauwkeurig beschouwd worden.

Hoofdstuk 9

Optimalisatie met het skelet -en markermodel

In dit hoofdstuk wordt besproken hoe de voorgaande algoritmes verbeterd kunnen worden indien er al van in het begin een skelet -en markermodel beschikbaar is ([HFP⁺00]).

Het skeletmodel is geschaleerd zodat het overeenkomt met de acteur. Het markermodel bevat de afstanden van de markers tot hun aanliggende gewrichten.

Eerst wordt het aangepaste motion capture proces beschreven. Daarna wordt er uitgelegd hoe het skelet -en markermodel kan berekend worden en welke aanpassingen er kunnen gedaan worden aan de algoritmes. Tot slot worden de resultaten van deze aanpassingen besproken.

9.1 Het proces

De stappen van het proces worden weergegeven in figuur 9.1 [HFP⁺01].

Eerst wordt het skelet -en markermodel berekend aan de hand van een gymbeweging die de acteur moet uitvoeren.

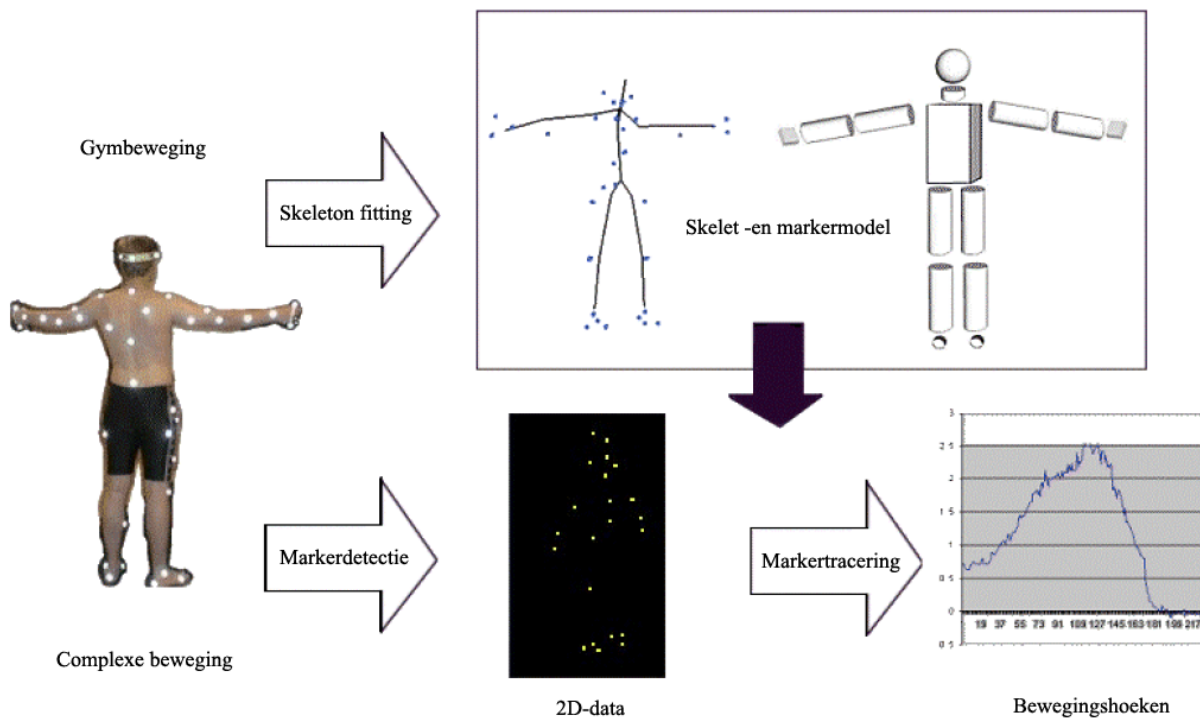
Daarna wordt dit model gebruikt tijdens de markertracering om ervoor te zorgen dat er zo weinig mogelijk markers verloren gaan.

Nadat alle markers in een frame gevonden zijn, kan het skelet er onmiddellijk in gepositioneerd worden. Het resultaat van de markertracering is dus ineens een geanimeerd skelet.

9.2 Gymbeweging

Het skelet -en markermodel dat we nodig hebben, is het resultaat van de globale skeleton fitting methode (zie 8.3). Dit model wordt met de originele algoritmes berekend zoals ze beschreven werden in de vorige hoofdstukken.

Om de invoerbeelden te genereren, wordt er aan de acteur gevraagd om de gymbeweging uit te voeren. De gymbeweging wordt beschreven in [HFP⁺00] en bestaat typisch uit een



Figuur 9.1: Motion capturing met behulp van een skelet -en markermodel dat berekend werd aan de hand van een gymbeweging. Met dit model kunnen nu complexere bewegingen opgenomen worden.

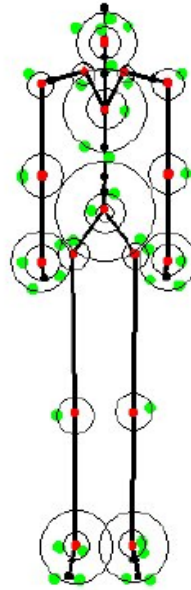
aantal bewegingen waaruit men duidelijk de belangrijke gewrichten kan herkennen. Deze bewegingen zijn tamelijk eenvoudig: bijvoorbeeld het buigen en strekken van een arm. Als resultaat krijgen we een skeletmodel dat geschaleerd is naar de acteur en een markermodel dat de afstanden van de markers tot hun aanliggende gewrichten bevat (figuur 9.2).

9.3 3D-reconstructie

Met het beschikbare skelet -en markermodel kunnen we twee testen uitvoeren vooraleer we besluiten een 3D-reconstructie aan de hand van een bepaald paar 2D-markers te aanvaarden.

9.3.1 Zichtbaarheidstest

Een marker wordt als zichtbaar beschouwd vanuit een bepaald camerastandpunt wanneer de camera er recht op kijkt en niet langs één van de zijkanten. Dit betekent dat de normaalvector van het gewricht naar de 3D-markerpositie en de vector van het gewricht naar het cameracentrum een scherpe hoek vormen.



Figuur 9.2: Het skelet -en markermodel: het skelet is geschaleerd naar de acteur en elke marker beweegt op een bol rond een aanliggend gewricht.

We veronderstellen wederom dat de verplaatsingen van de markers en dus ook die van het skelet in de opeenvolgende frames minimaal zijn. We zoeken de positie van het gewricht op waarmee de betreffende 3D-marker geassocieerd wordt. Deze positie in het vorige frame gebruiken we nu als een benadering voor de positie in het huidige frame.

We bepalen nu twee vectoren:

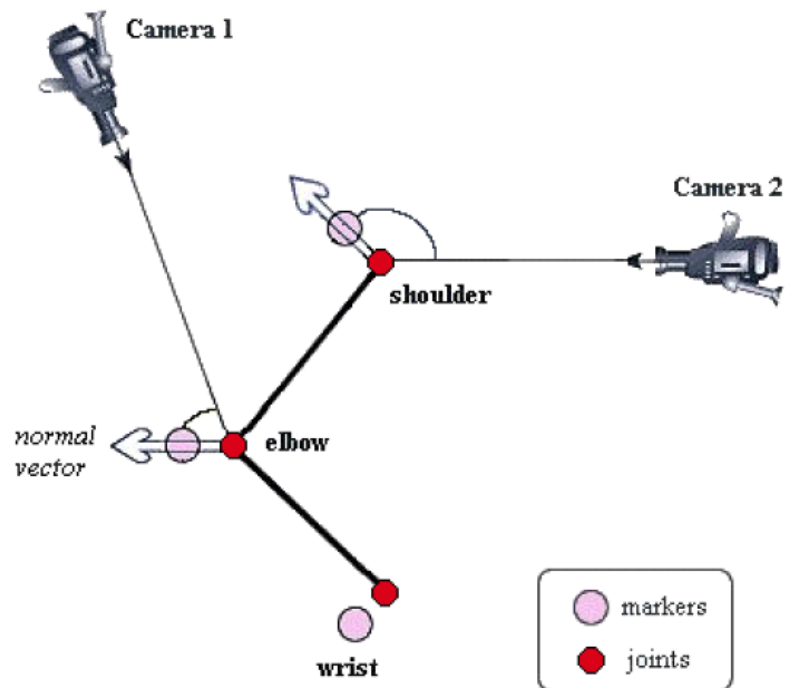
- **De normaalvector:** Dit is de vector die gevormd wordt door de posities van de 3D-marker en het gewricht.
- **De cameravector:** Deze vector wordt gevormd door de posities van de camera en het gewricht.

Indien de hoek die gevormd wordt tussen deze twee vectoren kleiner is dan 90 graden, slaagt de 2D-marker voor de zichtbaarheidstest. Het principe wordt duidelijk geïllustreerd in figuur 9.3.

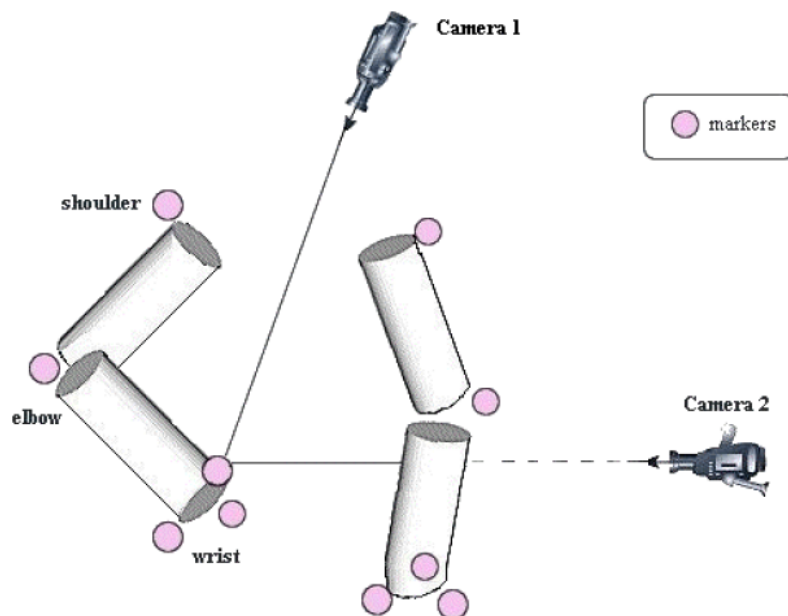
Wanneer een 2D-marker niet slaagt voor deze test, wordt een andere 2D-marker gekozen uit de lijst van 2D-markers, die gevonden werden aan de hand van de herprojecties van de 3D-marker.

9.3.2 Occlusietest

Na de zichtbaarheidstest wordt de occlusietest uitgevoerd. Hierbij wordt er gecontroleerd dat er zich geen lichaamsdelen tussen de 3D-marker en de camera bevinden. Het skelet wordt gepositioneerd zoals in het vorige frame. De lichaamsdelen worden voorgesteld door



Figuur 9.3: Zichtbaarheidstest: de elleboogmarker is zichtbaar voor camera 1, de schoudermarker is niet zichtbaar voor camera 2.



Figuur 9.4: Occlusietest: de marker op de pols is zichtbaar vanuit camera 1, de marker is echter verborgen voor camera 2 aangezien de lijn tussen de marker en het cameracentrum een ander lichaamsdeel snijdt.

cilinders. Het hoofd wordt echter voorgesteld door een bol. De assen van de cilinders liggen uiteraard in de richting van de botten. De straal van een cilinder is de gemiddelde afstand tussen de gewrichten en de markers die met dit lichaamsdeel geassocieerd worden.

Voor elke gereconstrueerde 3D-marker wordt een lijn getrokken tussen de 3D-positie en de camerapositie. Indien deze lijn één van de cilinders of de bol snijdt, dan doet er zich een occlusie voor vanuit dat camerastandpunt (figuur 9.4).

Net zoals bij de zichtbaarheidstest moet er dan een andere 2D-marker gekozen worden.

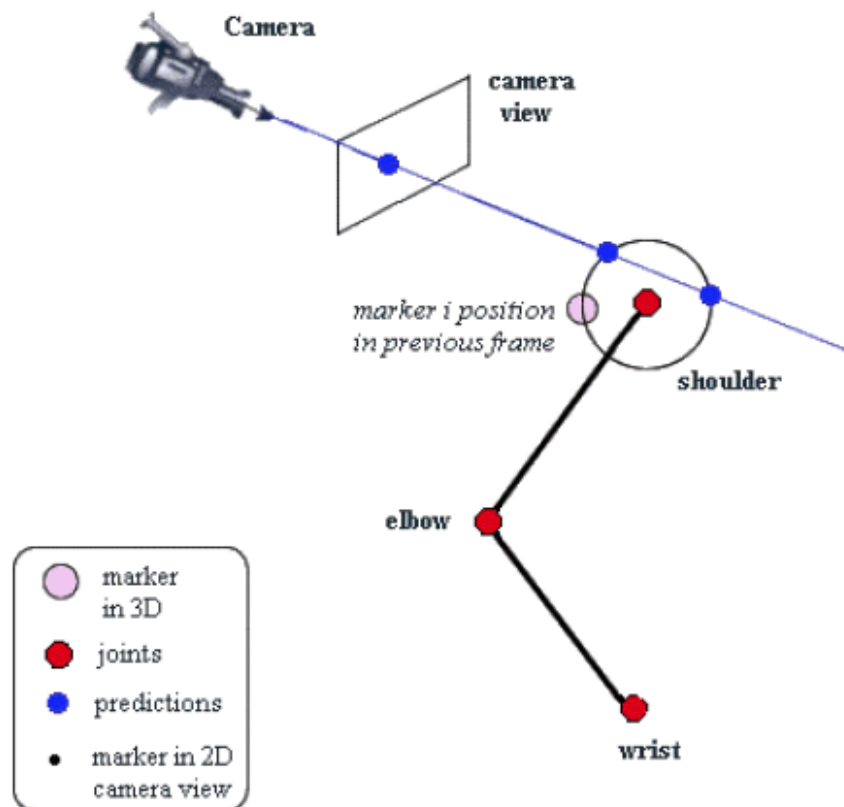
9.4 Markertracering

Wanneer we reeds een skelet -en markermodel ter beschikking hebben en nog steeds niet alle 3D-markers gereconstrueerd werden in een frame, kunnen de volgende stappen nog uitgevoerd worden:

- **Monoculaire reconstructie:** Bij deze methode probeert men aan de hand van één 2D-marker, die nog niet gebruikt werd, een 3D-reconstructie uit te voeren. Voor deze 2D-marker kan men een straal in de scène bepalen aan de hand van de cameramatrix. Deze straal vertrekt vanuit het cameracentrum en gaat door het betreffende beeldpunt waar de 2D-marker zich bevindt (figuur 9.5). Voor elke ontbrekende 3D-marker, zoeken we de positie op van het gewricht waaraan deze 3D-marker geconnecteerd is in het vorige frame. We kennen ook de normale afstand tussen de 3D-marker en dit gewricht. Hiermee kunnen we een bol in 3D voorstellen. Indien de eerder berekende straal deze bol intersecteert, kunnen we de 3D-marker reconstrueren. We kiezen hiervoor het intersectiepunt dat het dichtst bij de vorige positie van de 3D-marker ligt.
- **Identificatie door de afstand tussen het gewricht en de marker:** Voor elk gewricht waarvoor er nog 3D-markers ontbreken gaat men de positie in het vorige frame opzoeken, en die gebruiken als geschatte positie in het huidige frame. Uit het skelet -en markermodel halen we de normale afstand tussen het gewricht en de ontbrekende 3D-marker. Voor elke ongeïdentificeerde 3D-marker wordt de afstand tot het gewricht berekend. De 3D-marker waarvan de afstand het dichtst bij de afstand uit het skelet -en markermodel ligt, wordt gekozen als de ontbrekende 3D-marker.
- **Correctie van positie:** De 3D-markers die niet trinoculair gereconstrueerd werden, ondergaan nog een positiecorrectie. Indien de afstand van een 3D-marker tot het gewricht groter is dan gespecificeerd werd in het skelet -en markermodel, zorgen we ervoor dat de 3D-marker dichter naar het gewricht geplaatst wordt.

9.5 Skeleton fitting

Het probleem van skeleton fitting wordt sterk gereduceerd omdat het skelet -en markermodel reeds aanwezig is. De markers moeten niet meer geassocieerd worden met een bot of



Figuur 9.5: Monoculaire reconstructie

uiteinde van een skelet, maar wel rechtstreeks met de juiste marker van het skeletmodel. Het skelet kan nu in een frame gepositioneerd worden door de verschillen van de werkelijke en de gemodelleerde afstanden tussen de markers en de gewrichten te minimaliseren. Als initialisatie voor de kleinste-kwadratenmethode kunnen de skeletparameters van het vorige frame gebruikt worden.

De gezochte skeletparameters voor een frame zijn:

- de globale positie van het skelet in de scène
- de globale oriëntatie van het skelet in de scène
- de rotatiehoeken voor de vrijheidsgraden van alle gewrichten

9.6 Resultaten

In [HFP⁺01] worden de resultaten van deze werkwijze vergeleken met die van de originele werkwijze. De belangrijkste resultaten van de optimalisaties zijn dat:

- er minder correcties moeten doorgevoerd worden door de gebruiker.

- het algoritme robuuster wordt.
- het algoritme sneller wordt.

Deze resultaten hebben meerdere verklaringen:

- De kans dat de verkeerde 2D-markers gebruikt worden bij een 3D-reconstructie verkleint aanzienlijk.
- Er gaan veel minder markers verloren tijdens de markertracering.
- Indien er toch een marker verloren gaat, kan men een goede schatting geven van de markerpositie.
- Het skelet -en markermodel moet maar één keer berekend worden.

Hoofdstuk 10

Implementatie

In dit hoofdstuk zullen de implementaties van enkele van de reeds besproken optische motion capture technieken aan bod komen. Het doel van de implementatie is om met behulp van camera's en passieve markers synchrone opnames te maken, die dan later, offline, kunnen verwerkt worden tot een geanimeerd skelet. Verder wordt er nog aandacht besteed aan eventuele problemen die aan het licht kwamen tijdens de implementatie en hoe deze opgelost werden. Uiteindelijk worden de resultaten van de implementatie overlopen en voorgesteld.

10.1 Gebruikte bibliotheken en programma's

- Qt [Qt]
 - Grafische gebruikersomgeving
 - Custom widgets (DoubleSpinBox, ColorButton)
- OpenGL: Open Graphics Library [Libb]
 - Genereren van animaties
 - Weergeven van resultaten
- OpenCV: Open Computer Vision Library [Liba]
 - Camerakalibratie
 - 3D-reconstructie
 - Films openen en opslaan
 - Data openen en opslaan (XML)
- Matlab: Matrix Laboratory [Mat]

- Markerdetectie
- Kleur uit een afbeelding selecteren
- Fundamentele matrix berekenen
- Niet-lineaire kleinste-kwadratenmethode (lsqnonlin)
- Camera Calibration Toolbox for Matlab [fM]
 - Om de eigen OpenCV-implementatie mee te vergelijken.
- HuffyuV [Huf]
 - Lossless compressie en decompressie van beelden

10.2 Invoer

10.2.1 Camera's

OpenGL

Met OpenGL is het mogelijk om een accurate pinhole camera voor te stellen. Door deze camera dan op verschillende posities in de scène te plaatsen en een animatie te herhalen, kan men van hetzelfde onderwerp synchrone beelden opnemen. Omdat telkens dezelfde camera gebruikt werd, moesten de intrinsieke kalibratieparameters slechts eenmalig berekend worden. Voor elk camerastandpunt werden dan wel nog de extrinsieke kalibratieparameters berekend.

De beelden die met de OpenGL-camera's gegenereerd werden zijn ideaal om mee te experimenteren omdat men er zeer correcte, snelle en synchrone data mee kan verkrijgen. Ook heeft men amper te maken met ruis of afwijkingen in de kalibratieparameters.

Het programma om de beelden te genereren en op te slaan staat los van het eigenlijke motion capture programma. Het diende initieel om de camerakalibratie te testen en om afbeeldingen van een schaakbordpatroon op te slaan. Later werden er nog andere functies ingestopt om algoritmes uit te testen en om animaties te maken.

De mogelijkheden van dit programma zijn:

- Afbeeldingen en animaties opslaan.
- Camerakalibratie.
- Het zoeken naar cirkels in een afbeelding.
- 3D-reconstructie van punten.
- Herprojectie van 3D-punten.
- Animatie van een robotarm met markers door middel van inverse kinematica.

PGRFlyCapture

In het Expertisecentrum voor Digitale Media (EDM) staat een kooi met een tiental bevestigde camera's van Point Grey Research [Pro]. Als achtergrond hangen er grote groene lakens rond de kooi om een goed contrast te krijgen met het gefilmde onderwerp.

De camera's waren reeds vooraf gekalibreerd, dus moesten enkel nog de kalibratieparameters omgezet worden naar het formaat dat reeds gebruikt werd voor de OpenGL camera's. De kalibratiefuncties van OpenCV gebruiken namelijk een rotatievector in plaats van een rotatiematrix. De richting van deze vector bepaalt de as waar rond geroteerd wordt en de lengte van de vector bepaalt de rotatiehoek.

De opnames werden gemaakt met een reeds bestaand framework in een resolutie van 1024×768 . De beelden werden ongecomprimeerd als een `RgbSequence` opgeslagen en moesten nog geconverteerd worden naar een geschikt formaat om als invoer te kunnen dienen. Daarvoor werd een klein programmaatje geschreven met behulp van OpenCV en het reeds bestaande framework.

Om de opnames synchroon te laten verlopen, moesten alle camera's op het zelfde moment getriggerd worden. Wanneer dit echter gebeurde voor veel camera's tegelijk, viel de framerate al snel terug naar twee tot drie frames per seconde. Wanneer het aantal frames per seconde opgevoerd werd, kon het gebeuren dat sommige camera's één of meerdere frames lieten vallen, wat uiteraard niet gewenst is. Zoals al eerder aangegeven werd, speelt de behaalde framerate een grote rol bij het traceren van de markers. Men gaat er immers van uit dat de markers zich in opeenvolgende beelden slechts minimaal verplaatsen.

Om dit probleem te omzeilen, hebben we enkele opnames gemaakt waarin zeer traag bewogen werd. Op deze manier verplaatsten de markers zich niet te snel, maar nu konden we helaas ook niet meer spreken van het opnemen van een natuurlijke menselijke beweging. Daarom werd er besloten om verder te gaan met zelf gegenereerde OpenGL-beelden om de motion capture algoritmes te implementeren.

Compressie

Ongecomprimeerde beelden opslaan leidt al snel tot zeer grote bestanden. Daarom werden alle beelden lossless gecompriemd met de videocodec Huffyuv. De OpenGL-bestanden werden een factor vier kleiner en de PGRFlyCapture-bestanden een factor twee zonder dat er informatie verloren ging.

10.2.2 Markers

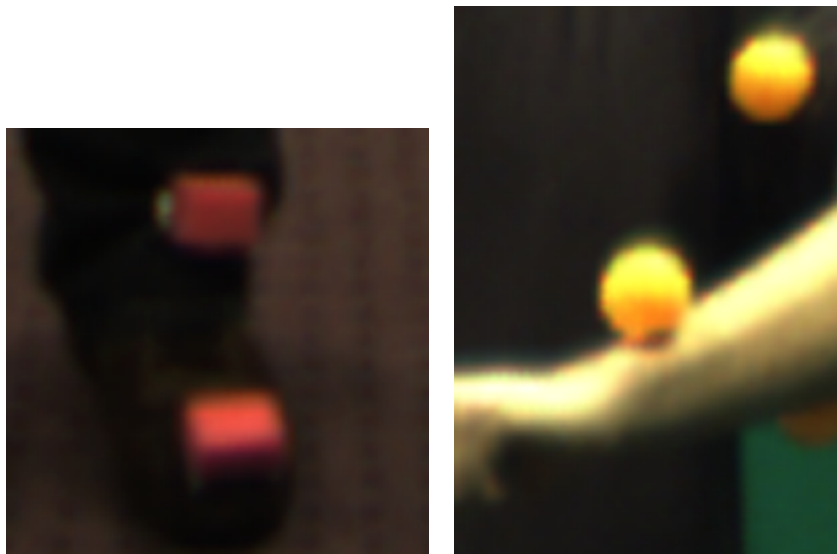
Roze sponsjes

De sponsjes werden met dubbelzijdige kleefband op de kledij of de huid bevestigd. De kleur viel meer dan genoeg op, maar doordat de sponsjes in een balkvorm geknipt waren, reflecteerde het licht anders op de verschillende vlakken van het sponsje. Hierdoor konden de middelpunten van de markers niet nauwkeurig genoeg bepaald worden. Het kwam zelfs vaak voor dat de kleuren zo fel verschilden dat niet alle markers gedetecteerd werden.

Wanneer de detectieparameters dan versoepeld werden, werden er te veel foute markers in het tapijt gedetecteerd.

Oranje tafeltennisballetjes

Deze werden op dezelfde manier bevestigd als de roze sponsjes. Ook hier was er voldoende contrast met de omgeving. Door de bolvorm van de tafeltennisballetjes, worden de markers vanuit elk standpunt als cirkels waargenomen. Dit kan handig zijn bij het zoeken naar de markers. Ook belangrijk is dat de kleur in het midden van de marker er ongeveer hetzelfde zal uitzien voor alle markers die vanuit hetzelfde standpunt bekeken worden. Maar dit is uiteraard afhankelijk van de belichting.



Figuur 10.1: De oranje tafeltennisballetjes zijn duidelijk de betere markers.

Rode punten

Voor de OpenGL-beelden worden rode punten als markers gebruikt. Dit is een ideale situatie vanwege de grote nauwkeurigheid van de gegenereerde beelden.

Verschillende kleuren tegelijk

Het is ook mogelijk om bepaalde markers een andere kleur te geven. Hiermee wordt het simpeler om de verschillende markers van elkaar te onderscheiden. De algoritmes om markers te traceren maakten hier niet specifiek gebruik van en de resultaten met één markerkleur waren al zo goed als perfect. Daarom werd de verdere implementatie hiervan achterwege gelaten. Wanneer er veel meer markers gebruikt worden of meerdere personen tegelijk opgenomen worden, kan dit wel een uitweg bieden.

10.3 Grafische gebruikersomgeving

De grafische gebruikersomgeving (figuur 10.2) bestaat uit vijf delen:

1. De menubar waarmee men de motion capture stappen kan uitvoeren en de scène kan bewerken.
2. De toolbar waarmee men de verschillende instellingendialoogvensters kan bereiken en de parameters kan aanpassen.
3. De drie cameraviews (linksboven, rechtsboven en linksonder) waarmee men verschillende camerabeelden kan selecteren en gebruiken.
4. De scène (rechtsonder) waar de resultaten van de motion capture stappen in 3D weergegeven worden.
5. De schuifbalk (onderaan) waarmee men naar een bepaalde positie kan gaan in de drie cameraviews en de scène tegelijk.

10.4 Camerakalibratie

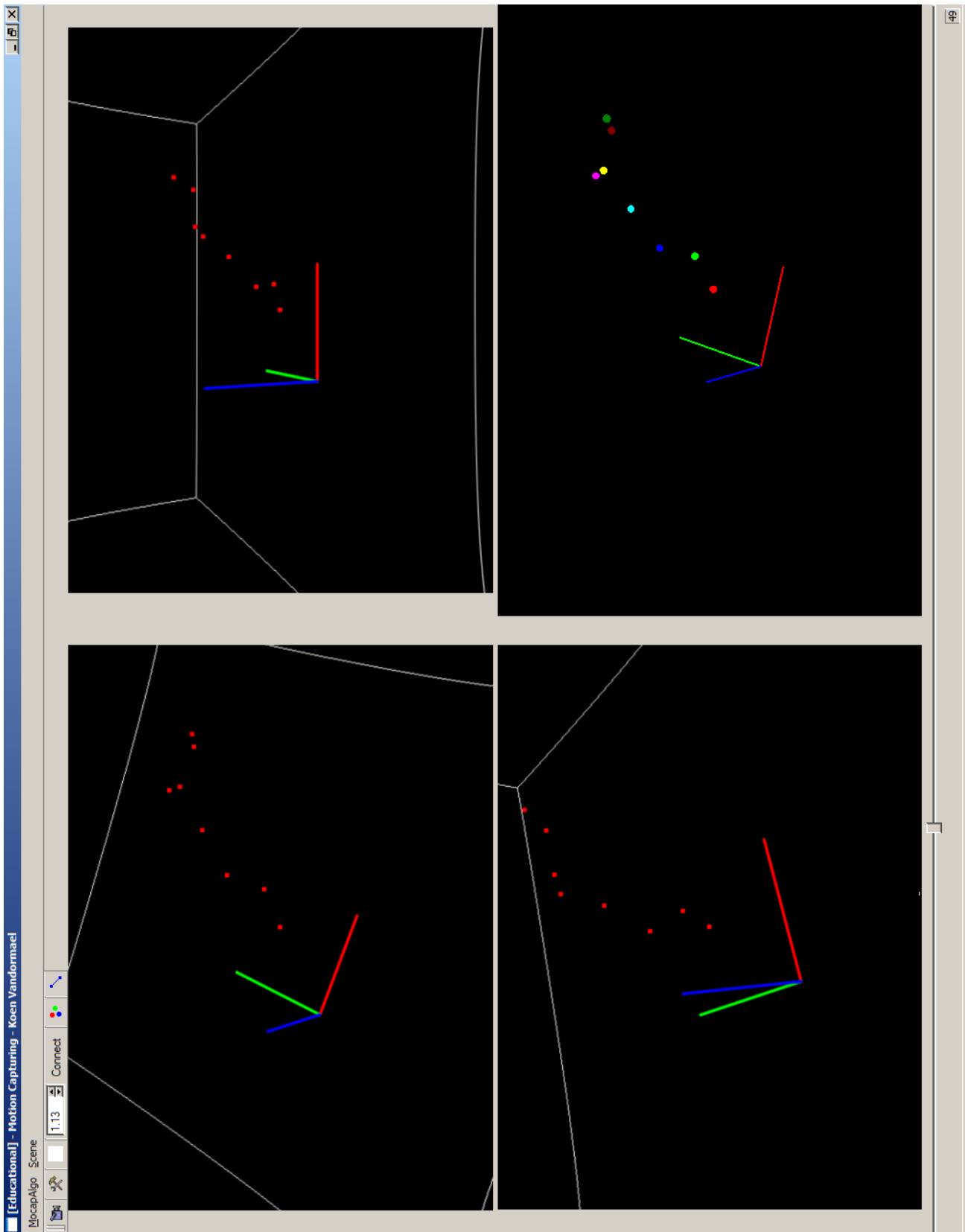
De camera's worden gekalibreerd met de beschikbare functies van de OpenCV bibliotheek. Deze bibliotheek implementeert de methode van Zhang [Zha00].

Om de intrinsieke parameters te berekenen gebruikt men als invoer een aantal afbeeldingen met een schaakbordpatroon dat telkens een andere positie en oriëntatie heeft. Om er zeker van te zijn dat men de juiste afbeeldingen gebruikt, kan men voor elk frame van een opname zoeken naar een schaakbordpatroon (figuur 10.4). Het aantal rijen en kolommen van het gezochte schaakbord kan uiteraard ingesteld worden, evenals de lengte van één blokje. Wanneer men tevreden is over het gevonden schaakbord, kan het huidige frame bewaard worden als een afbeelding. Met de verkregen vervormingscoëfficiënten kunnen we nu de lensvervorming ongedaan maken (figuur 10.3).

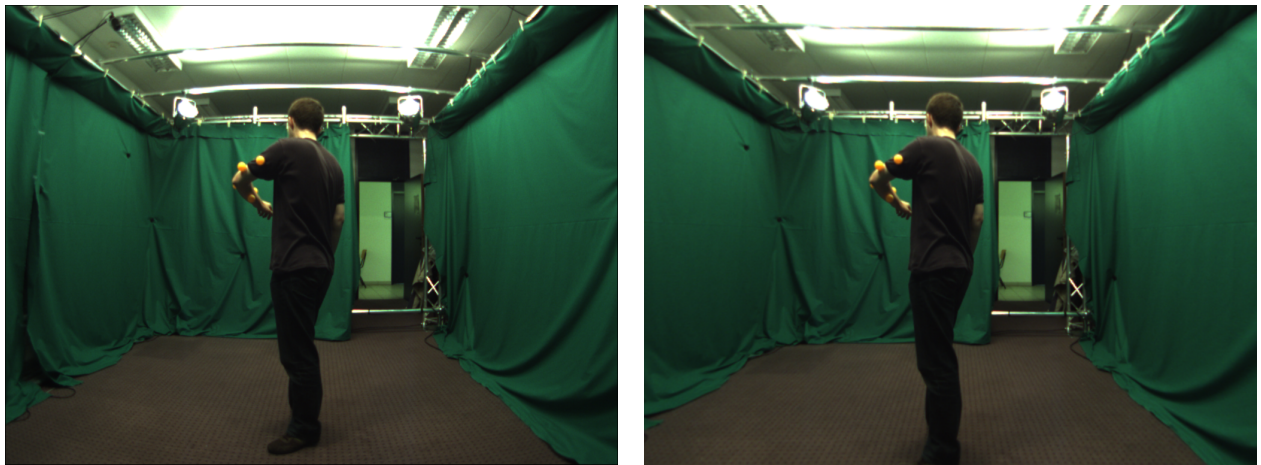
De extrinsieke parameters worden op ongeveer dezelfde manier bekomen. Hier is slechts één afbeelding nodig van het schaakbord patroon. Men moet er wel op letten dat er voor de kalibratie van alle camera's hetzelfde stilstaande schaakbordpatroon gebruikt wordt. Indien dit niet gedaan wordt, zullen de rotatie -en translatieparameters niet overeenkomen met hetzelfde wereldcoördinatenstelsel.

Hier kan men echter wel een oplossing voor vinden. Stel dat camera 1 gekalibreerd is aan de hand van schaakbord 1 en camera 2 aan de hand van schaakbord 2. Indien we dan camera 3 kalibreren aan de hand van de beide schaakborden, kan men uit de verschillende bekomen extrinsieke parameters voor camera 3 afleiden hoe de 2 coördinatenstelsels ten opzichte van elkaar liggen.

Nadat de camera's gekalibreerd zijn kunnen we de volgende operaties uitvoeren:



Figuur 10.2: De grafische gebruikersomgeving: rechtsonder ziet men de gereconstrueerde scène vanuit het standpunt van de camera linksboven.



Figuur 10.3: Links: normaal beeld met lensvervorming, rechts: na het ongedaan maken van de lensvervorming

- Projectie van een 3D-punt naar een 2D-punt voor een camera. Hiermee kan men bijvoorbeeld het gebruikte 3D-coördinatenstelsel in een camerabeeld weergeven.
- Berekenen van de eigenlijke 3D-camerapositie.
- Voor elk paar camera's kan de fundamentele matrix berekend worden. Hiermee kunnen dan de epipolaire lijnen gevonden worden, die de gevonden punten in het andere camerastandpunt voorstellen.

De kalibratieparameters van alle camera's kunnen opgeslagen worden in een XML-bestand (figuur 10.5).

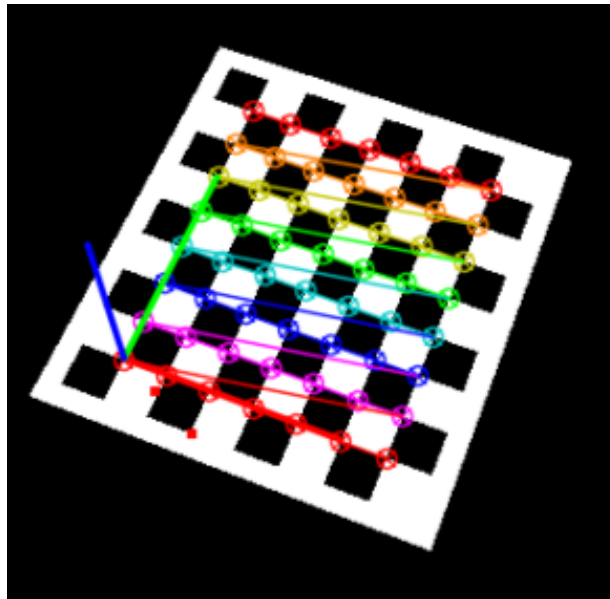
10.5 Markerdetectie

Het zoeken naar de markers gebeurt voor elke camera afzonderlijk. Het is ook mogelijk om frame per frame andere instellingen te gebruiken voor de markerdetectie. In deze sectie worden enkele mogelijkheden besproken.

10.5.1 Achtergrond verwijderen

Met behulp van een beeld van een lege scène is het mogelijk om de achtergrond van het gefilmde object te onderscheiden. Hierdoor kan er voor gezorgd worden dat er geen markers gedetecteerd worden die zich in de achtergrond bevinden.

Deze techniek werd wel geïmplementeerd, maar uiteindelijk niet gebruikt omdat de meeste scènes toch een contrasterende achtergrond bezaten. Dit verhoogt natuurlijk de snelheid van het detectiealgoritme.



Figuur 10.4: Een gedetecteerd 8×7 schaakbordpatroon met de herprojectie van het coördinatenstelsel

10.5.2 Kleur zoeken

De kleur waarnaar gezocht moet worden kan ofwel manueel ingegeven worden, ofwel uit een beeld geselecteerd worden. Een belangrijke parameter bij het zoeken naar de kleur is de „color treshold”. Deze waarde geeft per kleurkanaal aan hoe ver een gevonden kleur in het beeld mag afwijken van de gezochte kleur om als correct beschouwd te worden. Deze waarde gaat uit van positieve 8-bits integer kleurkanaalwaardes (0-255). Een kleur is dus correct wanneer de afwijking in geen enkel kleurkanaal meer is dan de gekozen waarde. Dit gaf een beter resultaat dan wanneer de waarde gebruikt werd om de totale afwijking (de som van de afwijkingen in elk kleurkanaal) te limiteren.

Het resultaat van deze bewerking is een binair beeld waarbij enkel nog witte vlekken te zien zijn op de plaatsen van de gevonden markers.

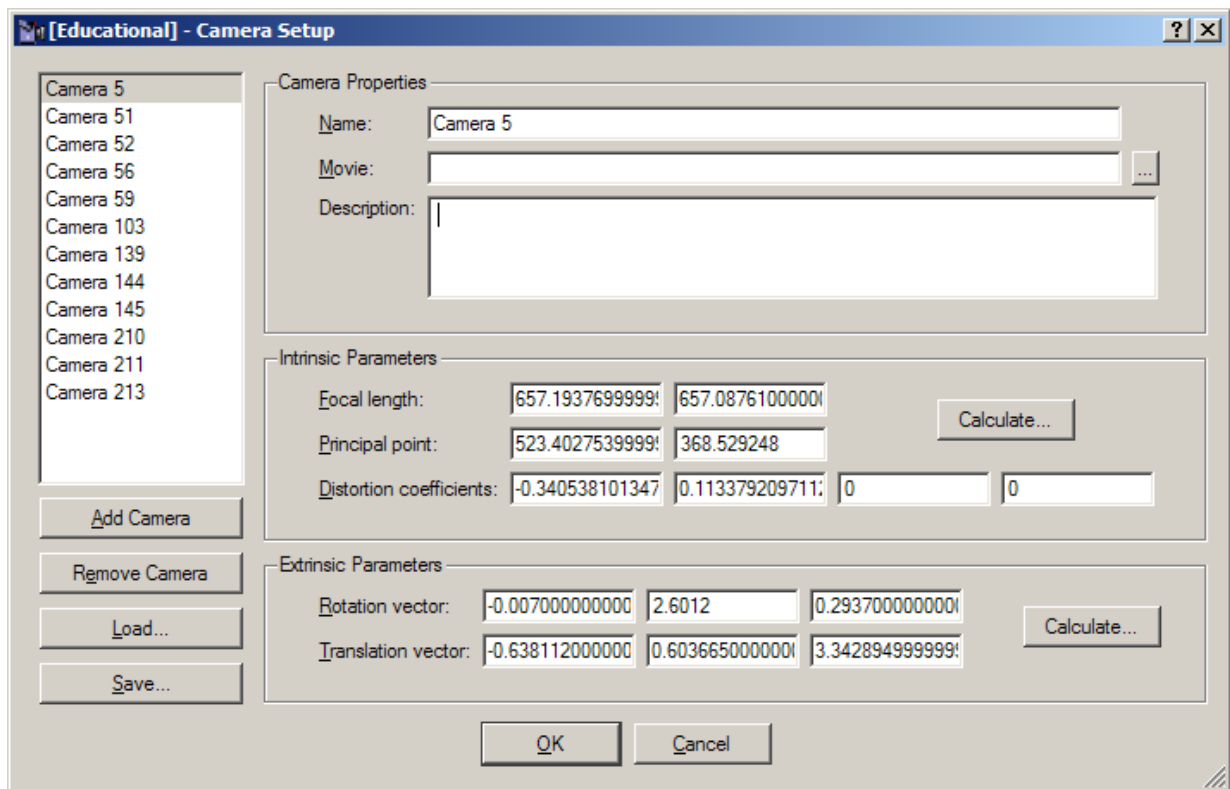
10.5.3 Ruis verwijderen

Om ruis te verwijderen uit dit resulterende beeld, wordt de morfologische operatie erosie gebruikt. Hierbij kan de grootte van het structuring element ingesteld worden.

10.5.4 Cirkels zoeken

Het is ook mogelijk om naar cirkels met een bepaalde straal te zoeken in het beeld door middel van een Hough-transformatie. Dit werd als test geïmplementeerd met OpenCV, maar het gewone zoeken naar kleuren gaf betere resultaten.

Een combinatie van het zoeken naar kleuren en cirkels is ook mogelijk, maar kan te lijden



Figuur 10.5: Enkele gekalibreerde camera's

hebben onder een slechte belichting. In het geval van de oranje tafeltennisballetjes kan je duidelijk zien dat de bovenkant van de markers lichter van kleur is dan de onderkant. Dit levert na het zoeken naar kleuren geen cirkels meer op. Een oplossing hiervoor kan zijn om eerst naar de cirkels te zoeken en daarna pas de kleur te controleren.

Uiteindelijk werd er gekozen om enkel naar de kleur te kijken om markers te detecteren, aangezien de resultaten hiervan al zeer goed waren.

10.6 3D-reconstructie

Het reconstrueren van de 3D-markers aan de hand van de 2D-markers van de verschillende camera's gebeurt met functies uit de OpenCV-bibliotheek. De stereotriangulatie gebeurt met de middelpuntsmethode.

Wanneer er meer dan twee corresponderende 2D-markers zijn, wordt er een 3D-positie berekend voor elke paar opeenvolgende 2D-markers. Het gemiddelde van deze posities bepaalt de 3D-marker.

10.7 Markertracering

Voor het traceren van de markers worden algoritmes uit [HFP⁺01, HFP⁺00] gebruikt. Er zijn enkele aanpassingen gemaakt aan de beschreven algoritmes:

- Om de markers makkelijker te kunnen identificeren, krijgen ze een uniek nummer toegewezen. Aan de hand van dit nummer wordt ook hun kleur bepaald. Dit vergemakkelijkt de weergave van de markers in de scène en in de camerabeelden.
- Voor 2D -en 3D-tracering met behulp van het glijdende venster van vier frames worden voorop twee zoekomgevingen van verschillende grootte ingesteld, in plaats van één zoekomgeving en een schaalfactor.
- Volgens het originele algoritme worden er voor de 3D-reconstructie slechts twee punten gebruikt. Daarna wordt het gevonden 3D-punt opnieuw geprojecteerd naar de andere camerastandpunten. Indien er op die plaatsen ook een marker aangetroffen wordt, wordt dit intern bijgehouden. Hierna is het echter nog mogelijk om de nieuwe gevonden punten samen met de originele twee te gebruiken om een correcter 3D-coördinaat te berekenen. Voor de 3D-reconstructie worden er dus meerdere punten gebruikt indien dit mogelijk is.
- Dit verandert natuurlijk ook het verdere verloop van het algoritme. Bij het traceren van de 2D-punten is er nu meer kans dat er markers behouden blijven, waarmee men opnieuw een 3D-reconstructie kan uitvoeren.

De gevonden 3D-markers kunnen in een XML-bestand bewaard worden. Het resultaat van het traceren van de markers kan gezien worden in figuur 10.2.

10.8 Skeleton fitting

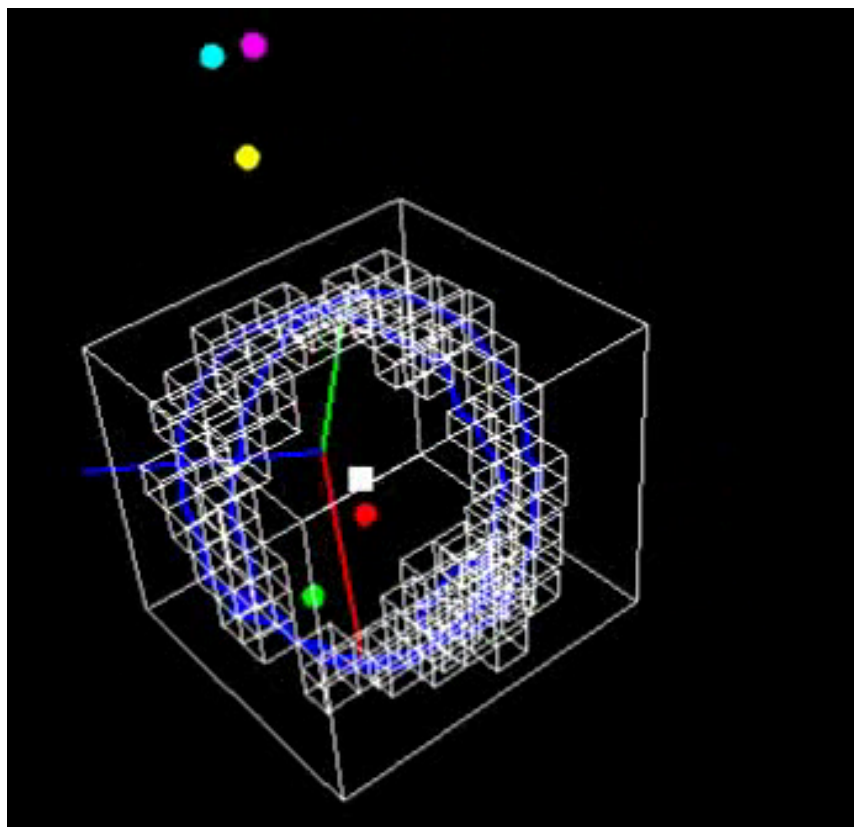
Skeleton fitting wordt geïmplementeerd met algoritmes uit [SPB⁺98, HFP⁺01, HFP⁺00, FGP⁺98]. Het gebruikte skelet kan niet dynamisch in het programma worden gewijzigd.

10.8.1 Lokale methode

Er kan automatisch naar een passende opdeling van de markers gezocht worden. Hiervoor kan men een waarde in de grafische gebruikersomgeving aanpassen en onmiddellijk het resultaat van de aanpassing zien. Vervolgens kunnen de markergroepen toegewezen worden aan een bot (bv. de onderarm), ofwel aan een uiteinde van het skelet (bv. het hoofd). Daarna kunnen de toewijzingen nog manueel aangepast worden.

Om de positie van een uiteinde van het skelet te berekenen wordt het gewogen gemiddelde van de toegewezen markers gebruikt.

De posities van de andere gewrichten worden berekend aan de hand van het rotatiecentrum van de markers van de aanliggende botten. De verschillende posities in het traject van een



Figuur 10.6: Het gewogen relatieve traject van een marker.

marker krijgen een gewicht, zoals beschreven werd in 8.2.2. Dit kan men goed zien in figuur 10.6. Het traject van de blauwe marker ten opzichte van de 3 markers bovenaan wordt weergegeven. De witte kubussen bepalen het gewicht van een positie in het traject. Het grote witte punt is het rotatiecentrum.

Voor de niet-lineaire kleinste-kwadraten methode wordt Matlab gebruikt. Het resultaat van de lokale skeleton fitting van een arm kan gezien worden in figuur 10.7.

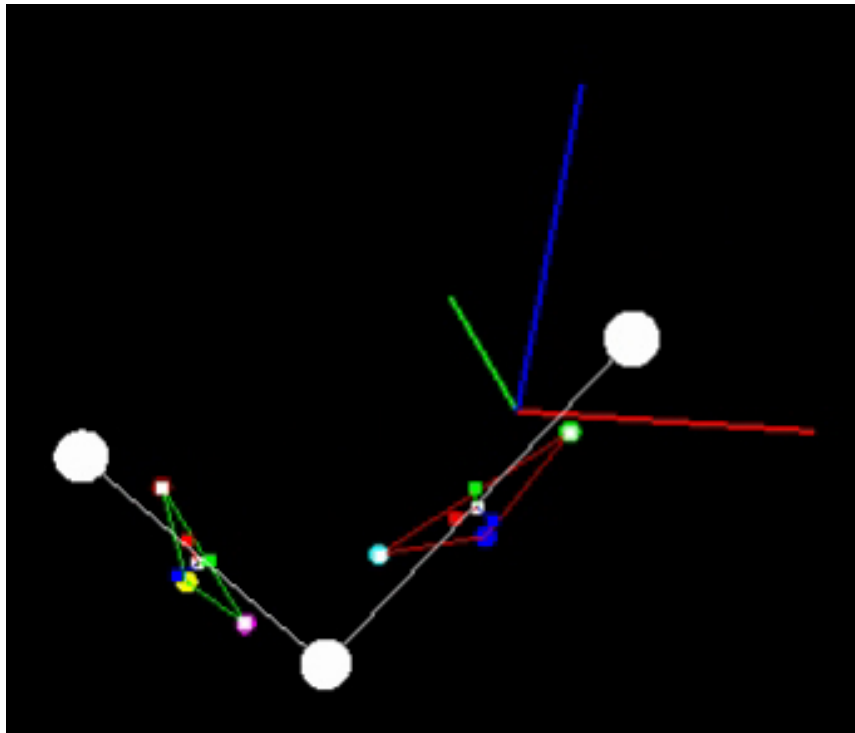
10.8.2 Globale methode

Omdat er Matlab gebruikt wordt voor de niet-lineaire kleinste-kwadraten methode moest het gebruikte model om een skelet voor te stellen twee keer geïmplementeerd worden. Eénmaal in Matlab om de berekeningen te kunnen uitvoeren en éénmaal in de normale code om het skelet te kunnen weergeven in OpenGL.

10.9 Resultaten

In figuur 10.8 worden enkele resultaten van de implementatie getoond.

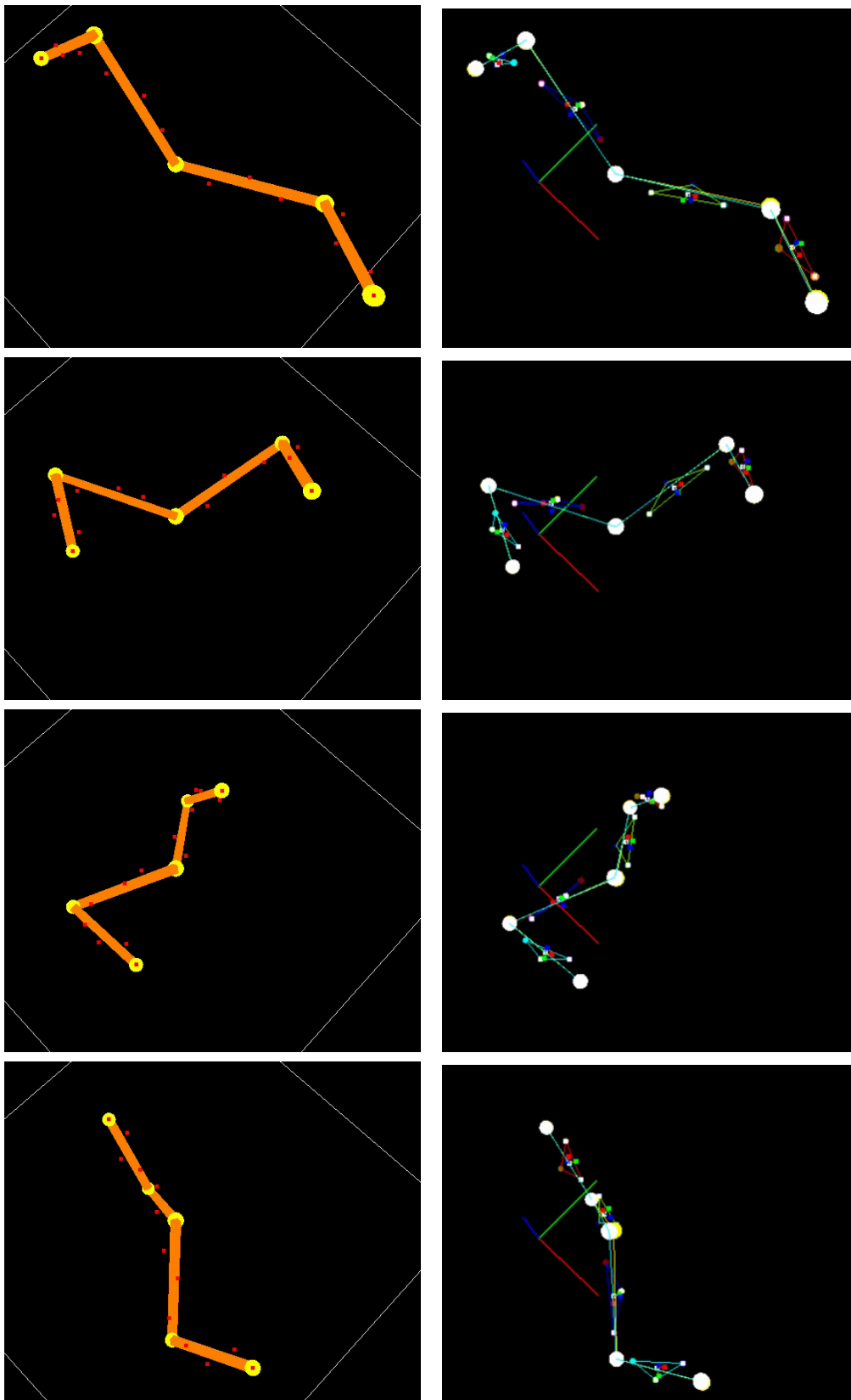
Aan de linkerkant staan de invoerbeelden die door één van de camera's geregistreerd wer-



Figuur 10.7: Een arm waarvan de posities van de gewrichten bepaald werden met local skeleton fitting.

den. De gele bollen en de rode punten stellen respectievelijk de gewrichten en de markers voor.

Aan de rechterkant worden de overeenkomstige gereconstrueerde beelden getoond. De markers hebben ter identificatie elk hun eigen kleur gekregen en werden in groepjes opgedeeld naargelang hun onderlinge bewegingen. De gele bollen stellen de gewrichtsposities voor, die berekend werden met de lokale skeleton fitting methode. In de meeste beelden worden de gele bollen grotendeels overlapt door witten bollen. Dit zijn de gewrichtsposities van het geschaleerde skeletmodel, die bekomen werden met de globale skeleton fitting methode. Het wereldcoördinatenstelsel wordt aangegeven door de rode, groene en blauwe assen.



Figuur 10.8: Links: de invoerbeelden die door één van de camera's opgenomen werden. Rechts: de gereconstrueerde beweging vanuit ongeveer hetzelfde camerastandpunt bekeken.

Hoofdstuk 11

Conclusies

Motion capturing is een techniek om bewegingen digitaal op te nemen en daarna over te zetten naar een animatiemodel. Verscheidene vakgebieden hebben baat bij deze techniek, maar het zijn vooral de film -en gamesindustrieën die er mee naar buiten treden.

Er bestaan verschillende soorten motion capture systemen, met elk hun specifieke voor -en nadelen. De keuze van een systeem hangt meestal af van de prijs, de nauwkeurigheid, de bemonsteringsfrequentie, de latentietijd en natuurlijk de toepassing.

Het motion capture proces van een offline, optisch systeem doorloopt vijf stappen: het aanbrengen van de markers op een acteur, de camera's opstellen en kalibreren, de beweging opnemen en hieruit de markers detecteren, de markers traceren en uiteindelijk de skeleton fitting. Het resultaat van deze stappen is een gedigitaliseerde en geanimeerde skeletbeweging. Men bekomt hier eveneens een skelet -en markermodel.

Wanneer men dit model gebruikt bij de opnames van een nieuwe beweging, kunnen de stappen van het motion capture proces zodanig geoptimaliseerd worden dat het proces sneller en robuuster worden. Er zijn dan ook minder interventies van de gebruiker vereist.

Als implementatie werd een programma geschreven waarmee men al de stappen van het motion capture proces kan doorlopen. Als invoer werden er meestal zelfgegenereerde OpenGL-beelden gebruikt, omdat de behaalde bemonsteringsfrequentie van de camera's in het EDM niet hoog genoeg was.

Uit de resultaten blijkt dat, wanneer de parameters goed ingesteld en de camera's correct gekalibreerd worden, er zeer degelijke resultaten kunnen gehaald worden met de beschreven algoritmes.

Bibliografie

- [BB] Gregory Baratoff and Scott Blanksteen. Tracking devices. <http://www.hitl.washington.edu/scivw/EVE/I.D.1.b.TrackingDevices.html>.
- [Beb] George Bebis. The reconstruction problem. <http://www.cse.unr.edu/~bebis/CS791E/Notes/StereoReconstruction.pdf>.
- [BFH⁺] R. Boulic, P. Fua, L. Herda, M. Silaghi, J. Monzani, L. Nedel, and D. Thalmann. An anatomic human body for motion capture.
- [Cap] Michael Capps. Magnetic tracking technology. <http://www.cs.nps.navy.mil/people/faculty/capps/4473/projects/mag-track/full.html>.
- [CL] Jonathan Cameron and Joan Lasenby. A real-time sequential algorithm for human joint localization.
- [Deb96] Paul E. Debevec. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley CA, 1996.
- [DeT] Carleton DeTar. Levenberg-marquardt method. http://www.physics.utah.edu/~detar/phyics6720/handouts/curve_fit/curve_fit/node7.html.
- [DPF⁺99] N. D'Apuzzo, R. Plänkers, P. Fua, A. Grün, and D. Thalmann. Modeling human bodies from video sequences, 1999.
- [FGP⁺98] P. Fua, A. Grün, R. Plänkers, N. D'Apuzzo, and D. Thalmann. Human body modeling and motion analysis from video sequences. In *Proceedings of the IS-PRS International Symposium on Real Time Imaging and Dynamic Analysis*, Hakodate, Japan, 1998.
- [fM] Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [FR94] Olivier D. Faugeras and Luc Robert. What can two images tell us about a third one? In *ECCV (1)*, pages 485–492, 1994.

- [Fus06] Andrea Fusiello. Elements of geometric computer vision. <http://profs.sci.univr.it/~fusiello/teaching/visione/elementsCV.pdf>, 2006.
- [GW01] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [HFP⁺00] L. Herda, P. Fua, R. Plänkers, D., R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Computer Animation*, Philadelphia, PA, May 2000.
- [HFP⁺01] L. Herda, P. Fua, R. Plänkers, D. Boulic, and D. Thalmann. Using skeleton-based tracking to increase the reliability of optical motion capture, 2001.
- [HS97] Richard I. Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding: CVIU*, 68(2):146–157, 1997.
- [Huf] HuffYuv. <http://neuron2.net/www.math.berkeley.edu/benrg/huffyuv.html>.
- [HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [LF95] Q. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis, 1995.
- [Liba] Open Computer Vision Library. <http://sourceforge.net/projects/opencvlibrary/>.
- [Libb] Open Graphics Library. <http://www.opengl.org>.
- [Mat] Matlab. <http://www.mathworks.com/>.
- [MDP93] N. A. Malik, Th. Dracos, and D. A. Papantoniou. *Particle tracking velocimetry in three-dimensional flows Part II: Particle tracking*, pages 279–294. Springer Berlin / Heidelberg, 1993.
- [Mot] Motion capture. http://en.wikipedia.org/wiki/Motion_capture.
- [Mul94] Axel Mulder. Human movement tracking technology. Technical Report 94-1, School of Kinesiology, Simon Fraser University, July 1994.
- [Owe] Robyn Owens. The hough transform. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT6/node3.html.
- [Pol00] Marc Pollefeys. Visual 3d modeling from images. <http://www.cs.unc.edu/~marc/tutorial>, 2000.

- [Pro] Point Grey Research Imaging Products. <http://www.ptgrey.com/products/imaging.asp>.
- [Qt] Trolltech Qt. <http://www.trolltech.com/products/qt>.
- [SPB⁺98] Marius-Calin Silaghi, Ralf Plänkers, Ronan Boulic, Pascal Fua, and Daniel Thalmann. Local and global skeleton fitting techniques for optical motion capture. *Lecture Notes in Computer Science*, 1537:26–40, 1998.
- [Tra] Wes Trager. A practical approach to motion capture: Acclaim’s optical motion capture system. http://www.siggraph.org/education/materials/HyperGraph/animation/character_animation/motion_capture/motion_optical.htm#An%20overview%20of%20current%20input%20systems.
- [Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

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 en uw akkoord te verlenen.

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

Motion Capturing

Richting: **Licentiaat in de informatica** Jaar: **2006**

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

Deze toekenning van het auteursrecht aan de Universiteit Hasselt houdt in dat ik/wij als auteur de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij kan reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

U bevestigt dat de eindverhandeling uw origineel werk is, en dat u het recht heeft om de rechten te verlenen die in deze overeenkomst worden beschreven. U verklaart tevens dat de eindverhandeling, naar uw weten, het auteursrecht van anderen niet overtreedt.

U verklaart tevens dat u voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen hebt verkregen zodat u 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 u als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze licentie

Ik ga akkoord,

Koen VANDORMAEL

Datum: