# Hoe wordt het tafeltennis balletje gedetecteerd?

Om het balletje terug te kunnen slaan moet het systeem weten waar het balletje zich bevindt. Hiervoor moet het systeem uit een informatiebron (Camera, Infrarood, sonar, etc.) de nodige informatie kunnen halen en verwerken tot informatie die door het systeem toegepast kan worden. Dit heet Object Tracking. Dit proces kan samengevat worden met het volgende diagram(Gebaseerd op A Survey on Object Detection and Tracking Methods (Himani, Darshak, & Udesang, 2014)):

Deze onderdelen worden in het hoofdstuk “Beschikbare beeldherkenning technieken/middelen” (Referentie?) verder toegelicht.

## Nodige eigenschappen voor de beeldherkenning

Om het systeem te correct te laten tafeltennissen zijn een aantal aspecten van de beeldherkenning van toepassing:

### Uitvoer tijd

Doordat het systeem maar een beperkte tijd heeft om te reageren zal het balletje snel getraceerd moeten worden. Wanneer de beelden niet snel genoeg verwerkt worden zal de robotarm niet op tijd kunnen reageren.

### Accuraatheid

Het systeem moet weten waar het balletje zich bevindt en zal hiermee moeten uitrekenen waar het balletje zich zal bevinden. Echter hoeft dit niet volledig accuraat uitgevoerd te worden. Dit komt doordat de arm zal reageren met een batje waardoor een verschil van enkele centimeters weinig verschil zal maken.

### Resistentie tegen verschillen

Tussen de beelden in zullen een aantal verschillen ontstaan. Deze kunnen veroorzaakt worden door beweging op de achtergrond, verschil in lichtsterkte of het vallen van schaduw. Hierdoor zal het mogelijk zijn dat het balletje niet correct wordt gevonden of dat het systeem het balletje op een andere locatie verwacht (false positive).

### Herstelmogelijkheid wanneer het balletje buiten beeld valt

Wanneer een speler (of de arm) een punt scoort zal het balletje van de tafel vallen en waarschijnlijk buiten het beeld terecht komen. Ook kan een van de spelers zich zo positioneren dat zijn batje (of arm) het beeld van de camera blokkeert. In deze situaties moet het systeem het balletje snel terug kunnen vinden om te zorgen dat door gespeeld kan worden.

## Beschikbare beeldherkenning technieken/middelen

Er zijn een aantal hulpmiddelen en algoritmen beschikbaar om te helpen met het vaststellen van de positie van de bal. Hieronder worden deze per categorie weergegeven:

### Opvangen beeld materiaal (Invoer beeldmateriaal)

De eerste stap in het implementeren van Object Tracking is een invoerbron waaruit informatie word toegediend. Dit zijn ruwe beelden die meerdere malen per seconde worden opgenomen. Hiervoor bestaan de volgende technieken:

#### Sonar/Echo

Een manier om het balletje te detecteren is met behulp van Sonar of Echo. Dit houdt in dat een sensor een geluidsignaal verzend en de tijd meet totdat hij dit signaal terug ontvangt. Hiermee kan een afstand worden uitgerekend tot de sensor. Echter kan dit signaal worden verstoord door andere geluidsbronnen in de omgeving, zoals het contact van het balletje op een batje, het stuiteren van het balletje of het spreken van mensen in de omgeving.

#### Infrarood

Een infrarood sensor kijkt naar de warmte die een object uitstraalt. Echter zal het balletje een soortgelijke temperatuur hebben als de omgeving waar deze zich in bevindt waardoor het balletje slecht zichtbaar zal zijn op een infrarood beeld.

#### Camera(kleur of grijswaarden)

Ten slotte kan worden gewerkt met beelden uit een camera. Iedere camera zal de nodige beelden kunnen produceren, echter zullen camera’s met meer beelden per seconden een accuratere herkenning opleveren doordat het verschil tussen de beelden minder groot is.

#### Vergelijking

Voor het opvangen van het beeld materiaal moet gelet worden op het feit dat het resultaat in 3D zal moeten zijn. Geen van de beschreven methode zal dit met een enkel meetpunt een 3D locatie kunnen berekenen.

Om dit correct te laten werken met een 3D omgeving zullen 2 meet punten nodig zijn om de locatie vast te stellen.

Een manier om dit te realiseren is een opstelling met een camera recht boven de tafel, waar een breedte en lengte locatie mee gedetecteerd kan worden. Vervolgens is er een zijcamera nodig om de hoogte te registreren. De beelden van de camera’s kunnen op een andere manier verwerkt worden. //PLAATJE?

### Detecteren van de mogelijke object positie(Object Detectie)

Vervolgens moet een techniek worden toegepast om een mogelijke positie van het balletje te verkrijgen. Het is mogelijk dat de technieken meerdere mogelijkheden retourneren, dit zal in een volgende stap worden opgelost.

#### Frame Differencing

Met Frame Differencing wordt het huidige beeld vergeleken met een voorgaand beeld. Ieder verschil dat ontstaat is een mogelijke positie van een bewegend object. Doordat deze methode zeer eenvoudig is, is dit ook een vrij snelle methode om mogelijke locaties vast te stellen. Echter kunnen verschillen ontstaan door veranderingen in het licht en mogelijke bewegingen op een achtergrond.

#### Optical Flow

Optical flow processing is een algoritme dat ogenschijnlijke beweging van een object waarneemt door iedere wijziging tussen beelden te registreren en te verwerken met een serie formules. Deze methode is in staat beweging zeer gedetailleerd op te vangen, echter is deze methode zeer intensief en niet goed in staat om in variërende omstandigheden te werken.

#### Background substraction

Dit model verwerkt beelden aan de hand van een vastgestelde achtergrond (template). Vervolgens wordt in iedere frame deze achtergrond uit het beeld gehaald. Hierdoor blijven alleen de bewegende objecten over in het beeld. Echter is deze methode zwak tegen verschillen in de omgeving zoals licht en beweging op de achtergrond. Hierdoor zal deze methode een stabiele achtergrond nodig hebben met een vastgestelde lichtinbreng. Ook moet de template gekalibreerd worden aan de positie van de camera.

#### Kleur herkenning

Ten slotte kan de locatie worden ingeschat op basis van een kleur. Dit houdt in dat ieder deel van het beeld, dat niet aan de kleurvereisten voldoet, genegeerd zal worden. De resterende informatie zal een mogelijke positie zijn van een object. Deze methode vereist ook dat er weinig objecten met een soortgelijke kleur aanwezig zijn in de omgeving. Ook kan een verschil in het lichtniveau een probleem opleveren voor het systeem.

#### Canny Edge Detection

Objecten als een geheel herkennen is een ingewikkeld process voor een computer. Een manier om het eenvoudiger te maken is het toepassen van Canny Edge Detection (Canny, 1986). Dit algoritme gebruikt een Gaussian Filter (Blur) om eventuele verstoringen en minieme veranderingen in het beeld te verwijderen. Vervolgens wordt de intensiteit van iedere pixel berekend(grijswaarde) en deze worden vergeleken met nabije pixels. Waar de intensiteit sterk verschilt van de aangrenzende pixels kan gesproken worden over een rand. Deze gegevens worden in het zwart-wit getekend op een afbeelding van dezelfde grootte als het origineel. Hierdoor raken de kleuren uit het beeld verloren. Daarentegen is het vrij ongevoelig voor verschillen in lichtsterkte.

#### Vergelijking

De detectie methode is van invloed op de mogelijke stappen die genomen kunnen worden om deze te verwerken. De methoden zijn hieronder in een tabel geplaats. Hierbij is gekeken naar de accuraatheid van de methode, de gevoeligheid voor verstoringen en de complexiteit van de methode:

|  |  |  |  |
| --- | --- | --- | --- |
| Methode | Accuraatheid | Verstoringsgevoelig | Complexiteit |
| Frame Differencing | Laag (Alle verschillen worden opgevangen, zoals belichting) | Zeer gevoelig (elke verandering is een nieuwe locatie) | Zeer laag (aantal operaties gelijk aan de hoeveelheid pixels) |
| Optical Flow | Zeer Hoog (Iedere detail, waaronder rotatie, wordt waargenomen) | Extreem gevoelig (Iedere verandering is een beweging die meegerekend wordt) | Zeer hoog (Meerdere wiskundige formules per pixel) |
| Background Subtraction | Hoog (Alleen de locatie van het object blijft over) | Zeer gevoelig(elke verandering is een nieuwe locatie) | Laag (Voor elke pixel een controle of het achtergrond is en een mogelijke reset) |
| Kleur Herkenning | Gemiddeld (Objecten met een soort gelijke kleur worden opgevangen) | Gevoelig (Licht kan de kleur van het object buiten bereik duwen) | Zeer laag (aantal operaties gelijk aan de hoeveelheid pixels) |
| Canny Edge Detection | Hoog (Alleen vormen blijven over) | Laag (Door het toepassen van een gaussian zijn de meeste verstoringen verwerkt) | Hoog (5 operaties zijn benodigd) |

In de methoden die weergegeven zijn is te zien dat het oplossen van de verstoringsgevoeligheid leidt tot een hogere complexiteit. Echter biedt deze uitkomst nog geen garanties. Dit komt doordat een aantal van de hogere tracking algoritmes afhankelijk zijn van deze stap.

### Herkenning van het object (Object Classificatie) en vaststellen positie

Rafeal Nieto heeft een overzicht gemaakt van mogelijke Object classificatie technieken die worden gebruikt in zijn Master Thesis (Nieto, 2013) heeft omschreven. Hieronder staat een beknopt overzicht van deze methoden.

#### Template Matching (TM)

Bij Template Matching wordt op een beeld het juiste object opgespoord door deze te vergelijken met een vooraf gesteld beeld (template). Dit wordt gedaan door een convolutie (Berekening van de overlap van twee signalen, of beelden (Convolution, 2015)) uit te rekenen en de locatie met de hoogste convolutie waarde is het object dat gevonden dient te worden.

Doordat het algoritme uit weinig stappen bestaat is het eenvoudig toe te passen. Deze methode verwerkt zijn gegevens snel genoeg om in een real-time applicatie te kunnen draaien.

Echter kan het algoritme niet goed tegen transformaties van het doelobject (Vervormingen, rotaties en verandering in formaat). Deze problemen zullen niet veel voorkomen bij de ping pong bal doordat deze ten alle tijden rond zullen zijn.

Verder kunnen problemen ontstaan wanneer de kleur van het object afwijkt van de template. Dit kan komen door bijvoorbeeld de belichting van het object. Dit probleem kan echter opgelost worden in de Object Detectie stap.

#### Mean-Shift (MS)

Mean Shift beschrijft een proces waarbij de nieuwe locatie wordt uitgerekend aan de hand van een eerdere positie in combinatie met een herkenningspunt. Dit herkenningspunt kan een template, een kleur combinatie of een andere herkenningspunt zijn.

Het algoritme rekent locaties uit die overeenkomen met het herkenningspunt. Echter is deze herkenning niet zo strikt als bij template matches, waarbij iedere de volledige overeenkomst telt, maar in plaats hiervan zullen alle overeenkomsten gemarkeerd worden. Vervolgens wordt de Epanechnikov Kernel methode (Struijker) toegepast om de overeenkomsten dicht bij de eerdere locatie prioriteit te geven over de anderen overeenkomsten. Vervolgens wordt het algoritme iteratief uitgevoerd over het resultaat totdat de locaties van de objecten samenvallen.

Deze methode kan ook doorberekend worden wanneer het object buiten beeld valt (Obstructie) door met de eerder uitgerekende snelheid en locatie de verwachte locatie van het object aan te passen. Hierdoor kan ieder frame een voorspelling gemaakt worden totdat een nieuwe cluster overeenkomsten in de buurt van de verwachte locatie komt waardoor het object teruggevonden kan worden.

Het algoritme presteert goed wanneer een specifiek kenmerk het object omschrijft. Hierdoor kan het object eenvoudig herkend worden.

Dit algoritme presteert echter minder goed wanneer het object buiten het beeldbereik valt. Dit komt doordat het algoritme zal aannemen dat het object door beweegt met als gevolg dat de voorspelde locatie ver buiten het beeld zal vallen.

#### Particle Filter-based Color Tracking (PFC)

Dit algoritme werkt op een verglijkbare manier als de bovenstaande Mean-Shift methode, echter werkt deze alleen met kleur. Vervolgens wordt hier ook de bovenstaande Epanechnikov Kernel methode (Struijker) toegepast om de overeenkomsten dicht bij de vorige positie een hogere waarde te geven.

Echter gebruikt deze methode het gemiddelde van alle potentiele locaties om de daadwerkelijke locatie te bepalen. Deze vergelijking van locaties wordt herhaaldelijk toegepast totdat de locaties samenvallen op een locatie. Deze uiteindelijke locatie is het resultaat van het algoritme.

Dit algoritme presteert beter als anderen in complexe situaties doordat iedere mogelijke locatie meegenomen wordt in het eind resultaat. Hierdoor is dit een van de meest gebruikte algoritmes voor beeldherkenning.

#### Lucas-Kanade Tracking (LK)

Lucas-Kanade Tracking is een vorm van Optical Flow herkenning (Rojas). Het werkt door de verschillen in grijswaarden (Intensiteit) te meten. Hierbij worden beelden vergeleken en probeert het algoritme in te schatten welke richting het object op beweegt zodat de verschillen in intensiteit verklaard kunnen worden.

Echter gaat het algoritme er van uit dat het verschil in tijd en afgelegde afstand tussen de beelden niet groot is. Dit wil zeggen dat het algoritme gebouwd is voor langzaam bewegende objecten. Hierdoor is deze niet geschikt voor het bijhouden van snelle objecten.

#### Incremental Learning for Robust Visual Tracking (IVT)

Incremental Learning gebruikt een aantal templates om een inschatting te maken van de mogelijke transformaties van een object. Vervolgens gaat het algoritme van ieder nieuw beeld het object registreren in een nieuwe template. Door het beeld te vergelijken met alle templates zal het algoritme eventuele veranderingen kunnen opvangen doordat hij het nieuwe object zal kennen. Daarentegen kan het algoritme niet goed omgaan met het verlies van het object en zal het algoritme steeds intensiever worden doordat de hoeveelheid templates groter wordt.

#### Tracking Learning Detection (TL)

De Tracking Learning Detection is een tracker die zich aanpast aan het beeld dat deze ontvangt. Dit wordt gerealiseerd door informatie van de frames bij te houden en te gebruiken. Deze tracker gaat er hierdoor vanuit dat de beweging tussen de frames klein is (lage snelheid) en dat het object in beeld is. Wanneer het object buiten beeld valt zal de tracker het object niet meer terug kunnen vinden.

#### Corrected Background-Weighted Histogram Tracker (CBWH)

Het doel van de Background weighted histogram tracker is om de Mean-Shift methode te verbeteren door de invloeden vanuit de achtergrond te verminderen. (Yang, Jia, Rong, Zhu, Wang, & Yue, 2013) Daarentegen is hij door deze methode minder efficiënt wanneer er kleurverschillen optreden. Om dit probleem op te lossen kan een Kalman Filter toegepast worden.

Deze methode kan de hoeveelheid iteraties van de Mean-Shift methode verkleinen. Hierdoor zal deze variant het object sneller kunnen vinden. Ook zal het een hogere precisie kunnen behalen doordat minder objecten meetellen in de berekening.

Het algoritme presteert goed wanneer er een duidelijk verschil is tussen het doel en de achtergrond. Echter kan het algoritme problemen krijgen wanneer er meerdere soortgelijke objecten bij elkaar aanwezig zijn.

#### Scale and Orientation Adaptive Mean-Shift Tracking (SOAMST)

Dit algoritme is wederom een aanpassing op het Mean-Shift systeem (Vojir, Noskova, & Matas, 2013). Hierbij probeert het systeem een schaal van het object in te schatten. Hiermee kan het systeem het object terugvinden wanneer de schaal veranderd. Deze aanpassing presteert beter op beelden waar de schaal van het object veranderd.

Daarentegen creëert het systeem nieuwe problemen bij beelden waarin de schaal niet veranderd. Om dit probleem op te lossen wordt gewerkt met een “back-ward consistency check” (Vojir, Noskova, & Matas, 2013) die de beelden in de omgekeerde volgorde verwerkt om een meetpunt te verkrijgen voor het nieuwe beeld. Hierdoor kost deze algoritme meer operaties als de originele Mean-Shift methode.

#### Vergelijking

De methode van Herkenning heeft twee belangrijke factoren. Ten eerste dient het object op een juiste locatie herkent te worden. Vervolgens dient dit zeer snel te gebeuren, doordat het systeem nog tijd nodig heeft om een tegenaanval in te plannen.

De accuraatheid van deze tracking mechanisme is uitgerekend door Rafael Martin Nieto (Nieto, 2013). Hieronder zijn twee grafieken (Nieto, 2013)) zichtbaar voor de accuraatheid van de tracking mechanisme in twee situaties. De eerste grafiek laat de resultaten zien voor een simpele simulatie (Laboratorium set) van een beweging. De tweede grafiek geeft de resultaten weer voor een serie van complexere sequenties (Simpele Realiteit).



In deze grafiek is de accuraatheid van de tracking algoritme berekend volgens hoofdstuk 3.3 van Rafeal Martin Nieto. Hierin staat vermeld dat een hogere score inhoud dat het algoritme slechter presteert. Echter heeft Rafeal Martin Nieto hierin niet gekeken naar de tijdsduur van de uitvoeringen.

Deze vergelijking heeft een aantal video sequenties vergleken:

1. Complex Movements: Het doel object wisselt snel van richting en snelheid.
2. Global Illumination: De lichtomstandigheden in de omgeving veranderen (zoals bijvoorbeeld bewolking).
3. Local Illumination: De lokale lichtomstandigheden veranderen, zoals wanneer het object door een schaduw beweegt.
4. Noise: Willekeurige variaties in het beeld
5. Occlusion: Het object raakt buiten beeld voor een periode.
6. Scale change: het weergegeven object wordt groter of kleiner.
7. Similiar Object: Een object met een soortgelijke kleur of vorm verschijnt in de buurt van het object.

Vervolgens moet gekeken worden naar de snelheid van de trackers.