# 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)):

## 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.

### 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 dit algoritme 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. 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.

### 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

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(Verschil tussen waarden (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

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.

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 herkent 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

Lucas-Kanade Tracking is een aanpassing op het template matching algoritme om kleine transformaties in het beeld toe te staan.

#### Incremental Learning for Robust Visual Tracking

#### Tracking Learning Detection

#### Corrected Background-Weighted Histogram Tracker

#### Scale and Orientation Adaptive Mean-Shift Tracking