

Voetbaldetectie met behulp van radar

De ontwikkeling en evaluatie van software voor een prototype

TU Delft - Elektrotechniek

Ruud van de Bovenkamp | Martijn Steenwijk

Voetbaldetectie met behulp van radar

De ontwikkeling en evaluatie van software voor een prototype

R. van de Bovenkamp
M.D. Steenwijk

juni 2008



Inhoudsopgave

Figurenlijst.....	I
Tabellenlijst.....	III
Voorwoord	V
Samenvatting	VII
1 Inleiding	1
Deel I: Achtergrond	3
2 Maxad: ‘Scoring Technologies’.....	3
2.1 Wat is Maxad?	3
2.2 MaxRadar, MaxSoft en MaxImage	3
2.3 MaxSoft	4
3 Programma van eisen.....	5
3.1 Eisen vanuit het beoogde gebruik.....	5
3.2 Eisen vanuit de ecologische situering.....	6
3.3 Eisen met betrekking tot het ontwerpen	6
3.3.1 Gebruikskennmerken	6
3.3.2 Productie- en ingebruikstellingkennmerken.....	7
3.4 Eisen met betrekking tot het produceren	7
3.5 Eisen met betrekking tot het liquidatiesysteem	7
3.6 Eisen met betrekking tot de bedrijfsstrategie	7
Deel II: Ontwerp.....	9
4 Deelconcepten	9
4.1 Functioneel blokschema	9
4.2 Alternatieven voor de radaropstelling	11
4.3 Alternatieven voor het detecteren van objecten	11
4.3.1 Crosscorrelatie.....	12
4.3.2 Piekdetectie	12
4.4 Alternatieven voor het volgen van doelen.....	15
4.4.1 Dichtstbijzijnde nabuurmethode	15
4.4.2 Het Kalman filter.....	16
4.5 Alternatieven voor het onderscheiden van speler en bal	19
4.5.1 Polarisatie	20
4.5.2 Snelheid.....	20
4.5.3 Radardoorsnede.....	20
4.5.4 RFID-tags	21
4.6 Alternatieven voor programmeertalen.....	22
4.6.1 Matlab.....	22
4.6.2 C	22
4.7 Alternatieven voor de bediening van het systeem	22
4.7.1 Textuele interface	22
4.7.2 Grafische interface	22

5	Verwant onderzoek	25
5.1	Meetopstelling	25
5.2	Bepaling van de parameters van de piekdetectie	26
5.2.1	Bepaling vaste grenswaarde	26
5.2.2	Bepaling optimale parameters OS-CFAR detector	28
5.2.3	Bepaling optimale parameters CML-CFAR detector	29
5.3	Objectclassificatie onderzocht	30
5.3.1	Polarisatiemetingen	30
5.3.2	Mate van constant zijn van reflectie	32
5.4	Conclusie	33
	Deel III: Implementatie	35
6	Implementatie	35
6.1	Programmeertaal	35
6.2	Bal zichtbaar maken	36
6.3	Objectdetectie	36
6.4	Objecttracking	37
6.4.1	Objecttracking met behulp van het dichtstbijzijnde nabuuralgoritme	37
6.4.2	Objecttracking met behulp van het Kalman filter	39
6.5	Objectclassificatie	40
6.6	Balsnelheid	41
6.7	Besluitvorming voor de inschakeling van beeldverwerkingsysteem	42
6.8	Communicatie	43
6.9	Graphical User Interface (GUI)	43
7	Evaluatie van het prototype	45
7.1	Evaluatie Tool	45
7.1.1	Dataloggedeelte	46
7.1.2	Simulatiegedeelte	46
7.1.3	Visualisatiegedeelte	46
7.2	Spelsituaties waarop de evaluatie gebaseerd is	47
7.3	Evaluatie deelsystemen	48
7.3.1	Evaluatie van objectdetectie	49
7.3.2	Evaluatie van objecttracking	50
7.3.3	Evaluatie van objectclassificatie	53
7.3.4	Evaluatie van de besluitvorming van het systeem	54
8	Conclusies en aanbevelingen	55
8.1	Conclusies en toetsing	55
8.1.1	Objectdetectie	55
8.1.2	Objecttracking	56
8.1.3	Objectclassificatie	56
8.1.4	Besluitvorming	57
8.1.5	De snelheid van het prototype	57
8.1.6	Toetsing aan het programma van eisen	58
8.2	Aanbevelingen voor de uiteindelijke implementatie	59
	Literatuurlijst	61
	Appendix A Pseudo-code	63

A1	Pseudo-code dichtstbijzijnde nabuuralgoritme.....	63
A2	Pseudo-code Kalman filter	64
English summary.....		65

Figurenlijst

Figuur 4.1	Functioneel blokschema.	10
Figuur 4.2	Kansverdeling van de reflectiesterkte van een object.	13
Figuur 4.3	Het CFAR detector algoritme. Overgenomen en bewerkt uit [1].	14
Figuur 4.4	Voorbeeld standaardmodus.	24
Figuur 4.5	Voorbeeld onderhoudsmodus.	24
Figuur 5.1	Meetopstelling.	26
Figuur 5.2	Close-up van de bal.	26
Figuur 5.3	Gemiddelde reflectie per afstandsbin.	27
Figuur 5.4	P_d en P_{fa} versus T voor een OS-CFAR detector.	28
Figuur 5.5	P_d per parameterset voor OS-CFAR, $P_{fa}=0,01$	29
Figuur 5.6	P_d per parameterset voor CML-CFAR, $P_{fa}=0,01$	30
Figuur 5.7	Verschil in reflectie van een verticale draad bij verschillende polarisatie.	31
Figuur 6.1	Links de ongelijmde baan, rechts het resultaat na lijmen.	38
Figuur 6.2	De tracks gevormd met behulp van het Kalman filter, gebruikmakend van de drie dynamische modellen. In frames waar geen piek werd gedetecteerd, is de positie van de piek voorspeld op basis van de vorige metingen.	40
Figuur 6.3	Screenshot van de GUI.	43
Figuur 7.1	Screenshot van de Evaluatie Tool.	45
Figuur 7.2	Overzicht van de verschillende situaties.	48
Figuur 7.3	Combinaties voor evaluatie objecttracking.	50
Figuur 7.4	Het kruisen van banen in situatie 4, dataset 2. De nabuurmethode verwisselt de bal en de speler, het Kalman filter verwisselt de banen niet. ...	51

Tabellenlijst

Tabel 5.1	Kans op een gemiste bal per afstandsbins.....	28
Tabel 5.2	P_d en T per parameterset voor OS-CFAR, $P_{fa}=0,01$	29
Tabel 5.3	P_d en T per parameterset voor CML-CFAR, $P_{fa}=0,01$	30
Tabel 5.4	Gemeten reflecties bij verschillende polarisaties.	30
Tabel 5.5	Verskil in reflectie tussen bal en mens bij verschillende polarisaties.	31
Tabel 5.6	Geschaalde standaarddeviatie van een stilstaand object.	32
Tabel 5.7	Geschaalde standaarddeviatie van een bewegend object.	32
Tabel 7.1	Gemiddeld foutenpercentage per type.....	49
Tabel 7.2	Het gemiddeld aantal keer per 100 frames dat de bal werd kwijtgeraakt.	52
Tabel 7.3	Het gemiddeld aantal keer per 100 frames dat de bal kwijtgeraakt werd doordat een speler en een bal verwisseld werden.....	52
Tabel 7.4	Het gemiddeld aantal keer per 100 frames dat de bal kwijtgeraakt werd door valse pieken.....	52
Tabel 7.5	Het gemiddeld aantal gevormde valse banen als gevolg van valse pieken per 100 frames.	52
Tabel 7.6	Resultaten per combinatie in procenten voor situaties waarin de bal zich niet dichtbij de speler bevindt.	53
Tabel 7.7	Percentage gedetecteerde ballen.....	54
Tabel 7.8	Percentage dat het beeldverwerkingsysteem terecht werd ingeschakeld voor een naderende bal.....	54

Voorwoord

Deze thesis maakt deel uit van de afsluiting van de bacheloropleiding Elektrotechniek. De thesis is geschreven binnen het bachelorafstudeerproject. In dit project is het fictieve bedrijf Maxad opgezet waarvoor een doelpuntendetectiesysteem ontwikkeld is. Wij vormen MaxSoft, een onderdeel van Maxad. Deze thesis beschrijft het werk dat voor MaxSoft verricht is.

Gedurende het bachelorproject moest Martijn helaas twee weken opgenomen worden in het ziekenhuis. Na de opname volgde een zware gewenningsperiode aan nieuwe medicijnen. We zijn blij dat het ondanks deze tegenslag toch is gelukt om samen tot het behaalde resultaat te komen.

De opbouw van de thesis is als volgt: achtergrondinformatie over het systeem en het bedrijf Maxad is te lezen in hoofdstuk 2. De eisen aan het ontwerp worden beschreven in hoofdstuk 3. Lezers die geïnteresseerd zijn in de verschillende deelconcepten waarin het ontwerp verdeeld kan worden, worden verwezen naar hoofdstuk 4 en 5. Lezers die geïnteresseerd zijn in het prototype en de prestaties van het prototype, worden verwezen naar hoofdstuk 6 en 7. Lezers die enkel geïnteresseerd zijn in de conclusies en aanbevelingen voor een uiteindelijke implementatie kunnen in hoofdstuk 8 terecht.

Wij willen de volgende personen bedanken voor hun waardevolle bijdrage aan het tot stand komen van dit rapport:

- Dr. ing. I.E. Lager (TU Delft), voor zijn fantastische begeleiding en adviezen;
- J. Figueras Ventura (TU Delft), voor zijn adviezen op het gebied van radar algoritmie;
- Ir. P.J. Aubry (TU Delft), voor zijn adviezen bij het onderscheiden van de deelconcepten;
- Prof. ir. P. van Genderen (TU Delft), voor de introductie die hij ons gaf in de wereld van radar en het gebruik van zijn apparatuur;
- Mevr. J. Kuenen (TU Delft), voor het doornemen en verbeteren van dit rapport
- De medewerkers van IRCTR en iedereen die op enige overige wijze een bijdrage heeft geleverd aan de totstandkoming van dit rapport.

Delft, 13 juni 2008
R. van de Bovenkamp
M.D. Steenwijk

Samenvatting

Maxad is een bedrijf dat reclame-opbrengsten maximaliseert voor adverteerders in het professionele voetbal. Maxad doet dat door een systeem te ontwikkelen dat op de belangrijkste momenten in een voetbalwedstrijd advertenties onder de aandacht brengt. Het systeem detecteert wanneer er gescoord wordt en toont op exact dat moment één of meerdere advertenties op de LED-boardings achter het doel.

Het systeem is gebaseerd op radar- en beeldverwerkingstechnologie. Maxad heeft het systeem opgesplitst in drie onderdelen, namelijk MaxRadar, MaxSoft en MaxImage. MaxRadar doet onderzoek naar methoden om een voetbal voor een radar goed zichtbaar te maken en ontwikkelt de radarhardware van het systeem. MaxSoft ontwikkelt software die uit de radargegevens kan bepalen of een voetbal het doel nadert. MaxImage ontwikkelt een systeem dat met behulp van beeldverwerking exact kan bepalen of de bal de doellijn passeert. Wanneer MaxSoft een voetbal die het doel nadert detecteert, wordt MaxImage ingeschakeld.

Deze thesis beschrijft het onderdeel van MaxSoft. De opdracht voor MaxSoft is: ontwikkel de software die op basis van FMCW-radargegevens een voetbal kan detecteren en volgen. De problemen die zich daarbij voordoen zijn onder andere het maken van onderscheid tussen voetballen en andere objecten, het volgen van een voetbal en het omgaan met complexe situaties zoals die zich in het voetbalspel voordoen.

De software is opgesplitst in drie delen, namelijk objectdetectie, objecttracking en objectclassificatie. Goede objectdetectie is essentieel voor het kunnen tracken van objecten, en goede tracking is belangrijk om objecten juist te kunnen classificeren. Tot slot vindt er besluitvorming plaats op basis van de door de objecttracking en objectclassificatie verzamelde gegevens.

Een studie is gedaan naar deelconcepten die mogelijke oplossingen vormen voor de verschillende delen in het eindontwerp. Tevens zijn er enkele deelconcepten onderzocht en getoetst.

Op basis van het gedane onderzoek en de gevonden resultaten is er software voor een prototype in Matlab ontwikkeld en geëvalueerd. Het prototype is vooral bedoeld om verschillende oplossingen voor de verschillende deelsystemen te kunnen evalueren.

Het detecteren van objecten vindt in het prototype plaats door middel van piekdetectie. De drempel die hierbij toegepast wordt, kan vast ingesteld worden maar kan ook adaptief bepaald worden met behulp van een CFAR detector. Voor het volgen van objecten kan gebruik gemaakt worden van zowel de dichtstbijzijnde nabuurmethode als een methode die gebruik maakt van een Kalman filter.

Objectclassificatie vindt in het prototype plaats op basis van de mate van constantheid van de reflectie van een object. Andere alternatieven zoals het detecteren van een verschil in reflectie bij verschillende polarisatie-richtingen, of het gebruik van RFID-chips zijn niet toegepast, omdat deze niet geschikt leken in de gebruikte opstelling of niet beschikbaar waren.

Uit de evaluatie van het prototype blijkt dat in sterk vereenvoudigde spelsituaties het systeem aan één van de twee gestelde nauwkeurigheidseisen voldoet. Het beeldverwerkingsysteem wordt in meer dan 75% van de gevallen terecht voor een naderende bal ingeschakeld en niet voor een ander naderend object. Het komt echter te vaak voor dat het beeldverwerkingsysteem niet ingeschakeld wordt terwijl er wel een bal aan komt.

De beste combinatie om objecten te volgen bestaat uit het Kalman filter met een CML-CFAR detector.

De gebruikte classificatiemethode blijkt niet goed genoeg te zijn om gebruikt te worden in de uiteindelijke implementatie. Het prototype classificeert 80% van de ballen juist in situaties waarin de bal duidelijk voor de radar zichtbaar is. Wanneer de bal zich echter in de buurt van een ander object bevindt, treden er classificatiefouten op.

Er wordt aanbevolen om voor de ontwikkeling van het eindproduct aanvullend onderzoek te verrichten naar andere eigenschappen voor de classificatie. Ook moet er nog onderzoek gedaan worden naar het combineren van meerdere radars. Het wordt aangeraden om de uiteindelijke implementatie niet in Matlab te ontwikkelen maar in een gecompileerde taal zoals C.

1 Inleiding

Maxad is een bedrijf dat zijn klanten helpt met het maximaliseren van reclame-inkomsten. Omdat in het professionele voetbal erg veel geld voor reclame- en sponsordoeleinden uitgegeven wordt, is dit een interessante markt voor Maxad. Om deze markt te kunnen bedienen, ontwikkelt Maxad een systeem dat doelpunten detecteert. Zodra het systeem een doelpunt detecteert, wordt de inhoud van de LED reclamepanelen rond het veld aangepast. Aangezien doelpunten de momenten zijn die het meest uitgezonden en herhaald worden op televisie zijn dit de momenten waarop adverteerders graag hun boodschap aan het publiek tonen. Reclameborden hebben dus een meerwaarde tijdens doelpunten. Maxad biedt voetbalclubs de kans om deze meerwaarde direct te vertalen in hogere reclame-inkomsten doordat reclametijd tijdens doelpunten aan een exclusieve adverteerder toegewezen kan worden.

Het te ontwerpen detectiesysteem zal bestaan uit twee onderdelen. Ten eerste is er een radarsysteem dat een ruwe indicatie geeft van de positie en richting van de bal. Ten tweede is er een beeldverwerkingsysteem dat uit fotobeelden de exacte locatie en richting van de bal bepaalt. Het radarsysteem heeft als functie het nauwkeurige detectiesysteem in te schakelen op momenten waarop de bal naar het doel beweegt. Dit is nodig omdat het beeldverwerkingsysteem zoveel rekenkracht vraagt dat het telkens maar korte periodes actief kan zijn.

Deze thesis beschrijft de ontwikkeling van de software voor het detecteren van een bal met behulp van een radarsysteem. Het systeem moet naast het detecteren van de bal ook in staat zijn de baan die de bal volgt vast te leggen en te voorspellen of die baan tot een doelpunt leidt of niet.

De software is opgebouwd uit de vier hoofdcomponenten objectdetectie, objecttracking, objectclassificatie en besluitvorming. Het detectiedeel moet objecten detecteren uit radarreflecties. Vervolgens moet het trackinggedeelte de baan van de gevonden objecten bepalen. Het classificatiedeel moet de gevonden objecten classificeren in spelers, de voetbal en achtergrond. Ten slotte moet besloten worden of het beeldverwerkingsysteem aangezet moet worden.

De opbouw van dit rapport is als volgt. Hoofdstuk 2 beschrijft het bedrijf Maxad en de opbouw van het te ontwerpen systeem. Dit wordt gevolgd door een programma van eisen in hoofdstuk 3. In hoofdstuk 4 worden de deelconcepten van het systeem onderzocht. Hierin worden onder andere methoden beschreven om objecten te kunnen detecteren en te volgen, maar er worden ook manieren beschreven om de bal beter zichtbaar en onderscheidbaar van mensen te maken. Vervolgens wordt in hoofdstuk 5 het uitgevoerde onderzoek naar een aantal deelconcepten beschreven. Hoofdstuk 6 behandelt de implementatie van het prototype. Deze implementatie wordt in hoofdstuk 7 geëvalueerd. De evaluatie onderzoekt het gedrag en de prestaties van het prototype in tien verschillende situaties die in het voetbal voor kunnen komen. Vervolgens wordt de snelheid van het systeem bepaald. De resultaten van de evaluatie wordt ten slotte getoetst aan het programma van eisen. Het rapport wordt in hoofdstuk 8 afgesloten met de

conclusies die te trekken zijn uit de evaluatie van het prototype en met aanbevelingen voor een uiteindelijke implementatie van het systeem.

Deel I: Achtergrond

2 Maxad: ‘Scoring Technologies’

‘Scoring Technologies’ is de slogan van het bedrijf dat opdracht gaf voor het ontwikkelen van het doelpuntdetectiesysteem. In dit hoofdstuk wordt het doel van de opdrachtgever en de functie van deze thesis voor de opdrachtgever toegelicht. Tevens wordt de opdracht en het doel van deze thesis afgebakend.

In paragraaf 2.1 wordt de opdrachtgever voor het systeem, het bedrijf Maxad, geïntroduceerd. Maxad is te verdelen in een aantal onderdelen, deze onderdelen worden beschreven in paragraaf 2.2. Tot slot wordt in paragraaf 2.3 de opdracht van MaxSoft beschreven.

2.1 Wat is Maxad?

Maxad is een internationaal opererend bedrijf dat zijn klanten helpt reclameopbrengsten te maximaliseren. Omdat er veel geld omgaat in marketing rond professionele voetbalwedstrijden, ontwikkelt Maxad een systeem dat op de belangrijkste momenten in een wedstrijd de reclameboodschap van de adverteerder aan een grote groep mensen zal tonen: de momenten waarop gescoord wordt. De boodschap wordt getoond op een LED-board dat achter het doel opgesteld staat en de adverteerder zal zijn advertentie terugvinden op krantenfoto's en in televisiesamenvattingen van de wedstrijd.

2.2 MaxRadar, MaxSoft en MaxImage

Het systeem dat Maxad ontwikkelt bestaat uit twee delen. Een deel dat werkt met radar om de bal in het veld te kunnen detecteren en een deel dat met behulp van beeldverwerking bepaalt of de bal de doellijn passeert. Het radargedeelte zelf is op te splitsen in twee delen, een deel hardware en een deel software. Maxad heeft besloten om de ontwikkeling van het systeem uit te besteden aan drie subgroepen, namelijk MaxRadar, MaxSoft en MaxImage.

- MaxRadar verzorgt de ontwikkeling van de hardware die nodig is om een voetbal zichtbaar te maken op een voetbalveld. Dit omvat zowel de ontwikkeling van radarhardware en positionering van antennes als het onderzoek naar andere technieken om een bal zo goed mogelijk zichtbaar te maken voor een radarsysteem.
- MaxSoft verzorgt de ontwikkeling van een softwaresysteem dat met behulp van het door MaxRadar geleverde systeem een voetbal kan detecteren en volgen. Om de software te ontwikkelen doet MaxSoft onderzoek naar methoden voor het detecteren, tracken en classificeren van objecten. Uiteindelijk moet het systeem besluiten of het beeldverwerkingsysteem moet worden ingeschakeld.

- MaxImage verzorgt de ontwikkeling van een systeem dat bepaalt of de voetbal de doellijn is gepasseerd. Het systeem bepaalt dit met behulp van beeldverwerking. MaxSoft ontwikkelt in de eerste plaats efficiënte algoritmes om een bal in fotobeelden te kunnen detecteren, maar onderzoekt ook verschillende mogelijkheden als het gaat om de keuze en positionering van de camera's.

In dit rapport wordt zowel naar MaxRadar als naar MaxImage verwezen. Het noemen van MaxRadar staat gelijk aan het noemen van het hardwaredeel van het radarsysteem en het noemen van MaxImage staat gelijk aan het noemen van het beeldverwerkingsysteem.

2.3 MaxSoft

Deze thesis beschrijft de bevindingen van MaxSoft. MaxSoft onderzoekt de mogelijkheden voor een softwaresysteem dat een voetbal detecteert met behulp van een FMCW-radar. Dit radarsysteem wordt geleverd door MaxRadar. Daarnaast wordt een prototype ontwikkeld dat dient als raamwerk om de prestaties van de verschillende gevonden mogelijkheden te onderzoeken. Op basis van de resultaten van het prototype worden er aanbevelingen gedaan voor de uiteindelijke implementatie.

Er is voor gekozen om in deze thesis een groot deel van het onderzoek te beschrijven zoals dat voor het eindproduct van toepassing is. Tot en met hoofdstuk 4 worden concepten beschreven die van toepassing zouden kunnen zijn op het eindproduct. Hoofdstuk 5 tot en met 7 beschrijven het onderzoek, de implementatie en evaluatie van het prototype. In hoofdstuk 8 worden er conclusies getrokken uit de evaluatie van het prototype en worden er aanbevelingen gedaan voor de uiteindelijke implementatie. De aanbevelingen zullen gebaseerd zijn op de concepten beschreven in hoofdstuk 4 en de conclusies die volgden uit de evaluatie van het ontwikkelde prototype.

3 Programma van eisen

Zoals beschreven in hoofdstuk 2 ontwikkelt MaxSoft de softwarekant van een radarsysteem dat de bal in een voetbalspel detecteert, volgt en een beeldverwerkingsysteem inschakelt wanneer die bal richting het doel gaat. De eisen die aan het ontwerp gesteld worden zijn in dit hoofdstuk samengebracht in een programma van eisen. De opbouw van dit hoofdstuk is als volgt. In paragraaf 3.1 worden de eisen vanuit het beoogd gebruik omschreven. Hierna volgen in paragraaf 3.2 de eisen vanuit de ecologische situering. In de paragrafen 3.4, 3.5 en 3.6 worden de eisen met betrekking tot het ontwerp, de productie en liquidatie van het systeem behandeld. Ten slotte beschrijft paragraaf 3.6 de eisen met betrekking tot de bedrijfsvoering.

3.1 Eisen vanuit het beoogde gebruik

- *Algemene beschrijving*
Het te ontwikkelen softwaresysteem moet op basis van gedigitaliseerde beatsignalen van Frequency Modulated Continuous Wave (FMCW)-radars [15], [20] real-time kunnen bepalen of een voetbal het doel nadert.
Wanneer een voetbal het doel nadert moet een beeldverwerkingsysteem ingeschakeld worden. Het inschakelen van het beeldverwerkingsysteem gebeurt via de parallelle poort. Het systeem is bedoeld voor de professionele markt.
- *Gebruiksgemak / Interface*
Het systeem moet beschikken over twee modi, een standaardmodus en een onderhoudsmodus. De bediening van het systeem vindt plaats via gangbare invoerapparaten zoals een toetsenbord en muis. Visualisatie vindt plaats via een beeldscherm.
- *Platform*
De software wordt ontwikkeld voor gebruik op een normale computer met een Windows of Linux besturingssysteem.
- *Maximale trackingsnelheid*
Het systeem moet objecten kunnen volgen die het doel naderen met een snelheid tot 30 m/s. Deze eis volgt uit het feit dat een professionele voetballer een voetbal een snelheid van maximaal 30 m/s kan meegeven.
- *Gevoeligheid*
Voor het te ontwikkelen systeem geldt dat het beeldverwerkingsysteem liever te veel gewaarschuwd moet worden dan dat er een bal gemist wordt. Het systeem moet kunnen functioneren bij alle weersomstandigheden waarbij gevoetbald wordt, dus zowel bij droog als bij regenachtig of mistig weer.
- *Fouten*
Het systeem moet voor 98% van de ballen die op het doel geschoten worden het beeldverwerkingsysteem waarschuwen.

Wanneer de software het beeldverwerkingsysteem waarschuwt voor een naderende bal, moet er in minimaal 75% van de gevallen ook daadwerkelijk een bal het doel naderen.

Het systeem moet binnen 33 ms kunnen bepalen of een gedetecteerd object een voetbal is. Binnen deze tijd legt een voetbal met maximale snelheid een meter af.

- *Inschakelgrens beeldverwerkingsysteem*

Uit het programma van eisen van MaxImage [13] volgt dat het beeldverwerkingsysteem een inschakelvertraging van 0,01 s heeft. Daaruit volgt dat het systeem in het uiterste geval moet worden ingeschakeld wanneer een voetbal zich op 0,3 m van de doellijn bevindt.

- *Input*

Voor elke aangesloten radar heeft het systeem een input in de vorm van een gedigitaliseerd beatsignaal.

Gebruikersinvoer vindt plaats met behulp van gangbare invoerapparaten zoals een muis en een toetsenbord.

- *Output*

De communicatie met het beeldverwerkingsysteem verloopt via de parallele poort en voldoet aan de IEEE 1284.4-2000 standaard voor bi-directionele datacommunicatie [6].

Visualisatie van de gebruikersinterface vindt plaats via een beeldscherm.

3.2 Eisen vanuit de ecologische situering

De hardware waarop de software geïnstalleerd is moet voldoen aan de CE richtlijn laagspanning (73/23/EG) [3] en richtlijn elektromagnetische compatibiliteit (EMC) (2004/108/EG) [2]. Tevens moet de hardware voldoen aan de norm IEC 60950-1 Information Technology Equipment – Safety – Part 1 General Requirements [7]. Daarnaast moet de omgeving waarin de software wordt geproduceerd voldoen aan de kaderrichtlijn Arbeidsveiligheid 89/391/EEG [8].

3.3 Eisen met betrekking tot het ontwerpen

3.3.1 Gebruikskenmerken

- De standaardmodus is de modus waarin het systeem normaal operationeel is. Het systeem verwerkt in deze modus de inkomende beatsignalen en besluit of er een voetbal nadert en of het beeldverwerkingsysteem gewaarschuwd dient te worden. De standaardmodus biedt tevens de mogelijkheid om het huidige weertype in te stellen. Er worden drie weertypen onderscheiden: helder, regenachtig en mistig.
- De onderhoudsmodus is de modus bedoeld voor probleemoplossing en onderhoud. In de onderhoudsmodus zijn de volgende onderdelen beschikbaar:
 - o *Systeeminformatie*
In dit onderdeel wordt informatie over de huidige toestand van het systeem weergegeven, zoals de inhoud van buffers in het systeem.
 - o *Systeeminstellingen*

In dit onderdeel kunnen verschillende parameters van het systeem worden ingesteld. Instelbare parameters zijn: de drempel van het triggersignaal van de radar, de grootte van de gebruikte buffers in verschillende systeemonderdelen en de drempelafstand waarop het beeldverwerkingsysteem gewaarschuwd wordt. Alle parameters zijn per weertype instelbaar.

- *Systeemdiagnostiek*

Het onderdeel systeemdiagnostiek biedt de mogelijkheid om uitvoer van verschillende systeemonderdelen te analyseren.

- *Updates*

Het onderdeel updates biedt de mogelijkheid om zowel algehele als aanvullende software-updates te installeren.

3.3.2 Productie- en ingebruikstellingkenmerken

Installatie

De software wordt eenmalig geïnstalleerd op het platform.

Ingebruikname van het systeem door de klant

Na productie wordt het systeem door een monteur bij de klant geïnstalleerd. Installatie houdt in dat het systeem naar de wensen van de klant geconfigureerd en getest wordt.

3.4 Eisen met betrekking tot het produceren

De operationele software zal ontwikkeld worden gebruikmakend van ANSI- standaard C-code. De compilers zullen gevalideerd zijn naar de ANSI/ISO/IEC standards.

3.5 Eisen met betrekking tot het liquidatiesysteem

Nadat de gebruikperiode van het systeem is verstreken, moet de software volledig verwijderbaar zijn.

3.6 Eisen met betrekking tot de bedrijfsstrategie

Levering

De software wordt geleverd aan Maxad. Maxad voegt de hardware, software en het beeldverwerkingsysteem samen en levert de combinatie aan zijn klanten.

Uitbreidingsmogelijkheden

De software moet bestaan uit een standaardsysteem wat aan te passen is aan de wensen van de klanten van Maxad. Aanpassingen vinden plaats door middel van het installeren van aanvullende pakketten. Het installeren van aanvullende pakketten vindt plaats in de onderhoudsmodus.

Deel II: Ontwerp

4 Deelconcepten

Het systeem dat radarreflecties moet interpreteren om de positie en baan van de bal te bepalen bestaat uit verschillende onderdelen. Voor elk van de onderdelen zijn verschillende invullingen mogelijk. Dit hoofdstuk beschrijft wat de verschillende deelconcepten zijn en wat de voor- en nadelen zijn van het toepassen van een bepaalde invulling voor een deelconcept.

De deelconcepten die onderscheiden kunnen worden zijn het aantal en de opstelling van de radars, de verschillende mogelijkheden om objecten te detecteren, de verschillende mogelijkheden om objecten te volgen, de verschillende mogelijkheden om de gedetecteerde objecten te classificeren, en de taal waarin de software geschreven wordt. Tenslotte is er het deelconcept dat de bediening van het systeem beschrijft.

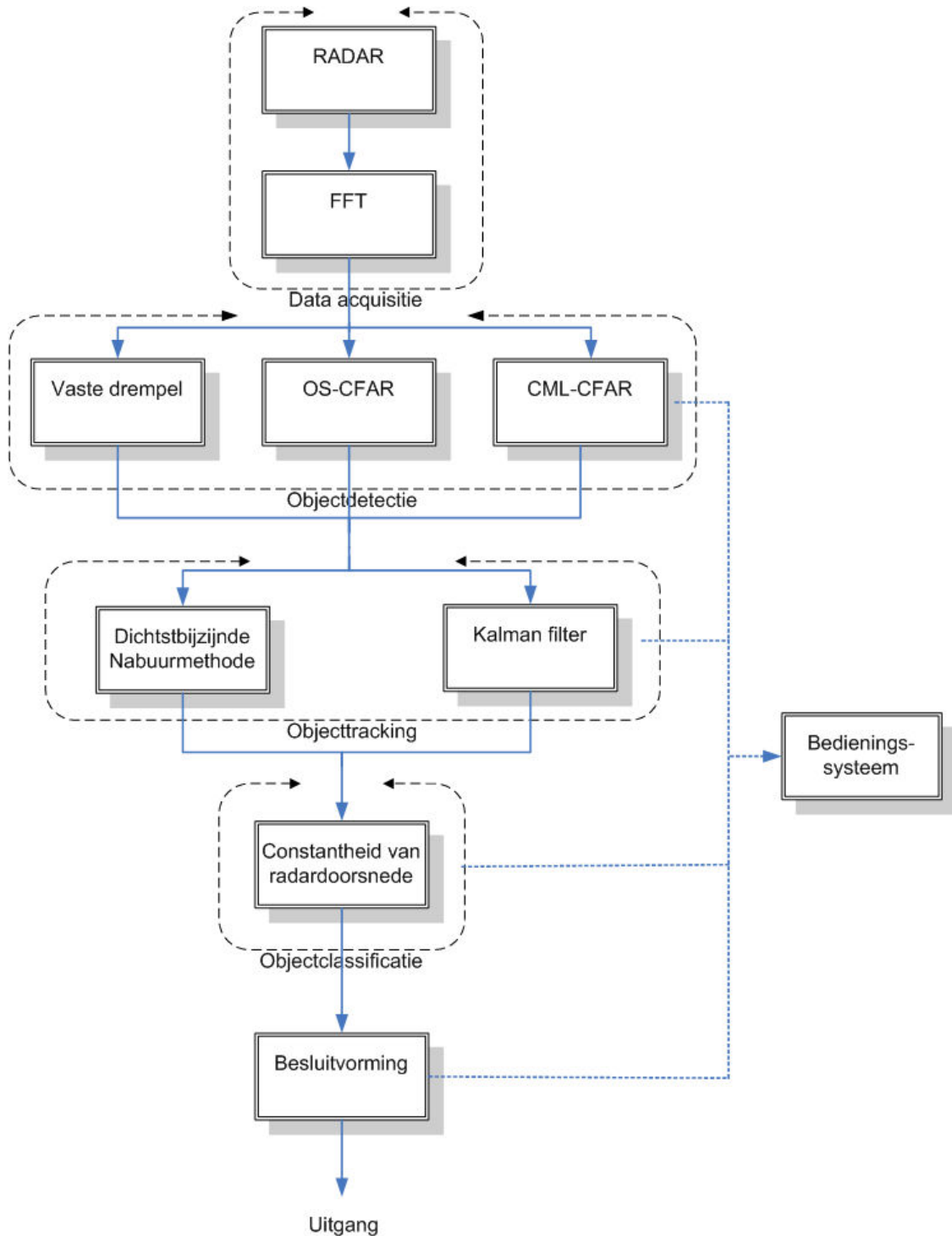
De verschillende mogelijkheden zijn onderzocht op nauwkeurigheid, complexiteit en benodigde rekenkracht van de computer. Bij de ontwikkeling van een prototype is vooral de benodigde rekenkracht een belangrijk punt, omdat tijdens de presentatie van het prototype een normale pc het rekenwerk moet doen. De rekenkracht van een normale pc is beperkt. In het uiteindelijke ontwerp weegt deze eis minder zwaar, alhoewel grotere rekenkundige complexiteit tot duurdere hardware leidt.

De opbouw van dit hoofdstuk is als volgt. In paragraaf 4.1 wordt een overzicht gegeven van het systeem aan de hand van het functioneel blokschema. In paragraaf 4.2 wordt de opstelling van de radars en het aantal benodigde radars onderzocht. Vervolgens worden in paragraaf 4.3 de alternatieven voor het detecteren van objecten gepresenteerd. In paragraaf 4.4 wordt besproken hoe de gevonden objecten gevolgd kunnen worden. Daarna wordt in paragraaf 4.5 ingegaan op de verschillende methodes om de gevonden objecten te verdelen in spelers en ballen. In paragraaf 4.6 wordt de te gebruiken programmeertaal besproken en tenslotte wordt in paragraaf 4.7 ingegaan op een oplossing voor de bediening van het systeem.

4.1 Functioneel blokschema

In figuur 4.1 is het functioneel blokschema van het softwaregedeelte van het systeem weergegeven. Het hoogste element in het schema is de radar zelf. De gegevens die de radar produceert, worden in het blokje FFT verwerkt. De ingang van dit deelsysteem is het beatsignaal en de uitgang is een spectrumanalyse van dat signaal. De spectrale componenten representeren een reflectie op een bepaalde afstand van de radar. Om de frequentieanalyse goed uit te kunnen voeren moet het systeem getriggered worden. Het triggeren wordt ook in dit blokje uitgevoerd. Na de FFT wordt er een piekdetectie uitgevoerd om objecten te detecteren, voor de piekdetectie kan uit verschillende drempelmethodes gekozen worden. Vervolgens worden de gevonden objecten getrackt

met behulp van of het dichtstbijzijnde nabuuralgoritme [10] of een Kalman filter [10], [24]. De objecten worden dan geïdentificeerd aan de hand van de mate van constantheid van de reflecties. Hierna wordt besloten of MaxImage ingeschakeld moet worden. Dit signaal is de uitgang van het systeem. Alle subsystemen hebben ook een uitgang naar het bedieningsysteem.



Figuur 4.1 Functioneel blokschema.

4.2 Alternatieven voor de radaropstelling

Hoewel het principe van een radarsysteem altijd hetzelfde is, bestaan er zeer veel verschillende soorten radars. Enkele voorbeelden zijn pulse radars, Frequency Modulated Continious Wave (FMCW)-radars en Doppler-radars [20]. Elke implementatie van een radarsysteem heeft specifieke voor- en nadelen. Naast het type radar dat gebruikt wordt moet ook het type antenne gekozen worden. Het antennetype is bepalend voor het stralingspatroon van de antenne. Antennes met een breed stralingspatroon belichten een groot gedeelte van de ruimte rond de antenne, smalle bundels juist een klein gedeelte. Om met een brede bundel veel energie van objecten te laten reflecteren is meer vermogen nodig dan bij gebruik van een smalle bundel. Een smalle bundel belicht daarentegen maar een klein gedeelte van de ruimte en heeft dus een kleiner blikveld.

Een keuze die nauw samenhangt met de bundelgrootte is de hoeveelheid radars die gebruikt worden. Met meerdere radars met een smalle bundel is toch een groter gebied te belichten. Daarnaast kan een radar alleen maar de afstand tussen een object en de antenne bepalen. Eén radar ziet objecten in een vlak. Twee radars zien gezamenlijk objecten in een lijn. Om een punt in de ruimte te kunnen zien zijn dus minimaal drie radars nodig.

Een andere mogelijkheid is om gebruik te maken van een scanning radar [20]. Een dergelijke radar heeft een bewegende bundel die telkens een ander deel van de ruimte belicht. Het bewegen van de bundel kan door de antenne fysiek te bewegen of door gebruik te maken van de beam-forming mogelijkheden van een array antenne.

4.3 Alternatieven voor het detecteren van objecten

Het detecteren van doelen komt neer op het interpreteren van de radarreflecties. De radar die gebruikt wordt in het prototype is van het type FMCW, waarvan de werking wordt behandeld in [17]. De amplitude van het signaal dat een FMCW-radar uitzendt, is constant. De frequentie van het uitgezonden signaal wordt gemoduleerd. Vaak is het modulerende signaal een zaagtand. Tijdens een sweep neemt de frequentie lineair toe tot een maximale frequentie en valt daarna weer terug naar de startfrequentie. Als het uitgezonden signaal weerkaatst wordt door een object, valt een deel van het signaal weer terug op de antenne. Een eventuele dopplerverschuiving buiten beschouwing latend, heeft het ontvangen signaal dezelfde frequentie als toen het uitgezonden werd. Het signaal dat op het moment van aankomst uitgezonden wordt door de radar heeft echter een andere frequentie omdat de modulatie verandert in de tijd. De afstand van de radar tot het doel is te bepalen uit het verschil van de twee frequenties. Hiertoe wordt het in- en uitgangssignaal gemixt en geanalyseerd met een Fouriertransformatie.

Het spectrum dat met de Fouriertransformatie verkregen wordt, wordt in deze thesis een frame genoemd. Elk frame moet geïnterpreteerd worden om doelen te detecteren. Voor het interpreteren zijn twee alternatieven onderzocht. Eén alternatief maakt gebruik van de crosscorrelatiefunctie en een ander maakt gebruik van piekdetectie.

4.3.1 Crosscorrelatie

De crosscorrelatie wordt in signaalverwerking veel gebruikt om te bepalen hoeveel twee signalen op elkaar lijken [4], [31]. Aangezien de signalen die verwerkt moeten worden in het systeem bestaan uit reflecties van objecten en ruis lijkt het een goed idee om de crosscorrelatie van de signalen te bepalen. De objecten in twee opeenvolgende frames zullen bij een hoge frame-rate niet veel veranderen terwijl ruis dit juist wel doet. Het signaal in een frame lijkt dus veel op het signaal in het vorige frame op de punten waar zich objecten bevinden.

Voor het bepalen van de crosscorrelatiefunctie zijn efficiënte algoritmes bekend. Het kost de computer dus niet veel rekenkracht om deze te bepalen. In dit opzicht is de crosscorrelatiemethode dus een goede methode. De nauwkeurigheid van de methode is niet zo groot.

4.3.2 Piekdetectie

Een tweede methode om objecten in de radarreflecties te detecteren is met behulp van piekdetectie. Bij deze methode worden de lokale maxima in het spectrum bepaald door te kijken naar het 3-punt maximum. Bij het bepalen van het 3-punt maximum wordt gekeken of een punt in de reeks groter is dan zowel het vorige als het volgende punt. Als dat zo is, dan is een lokaal maximum gevonden.

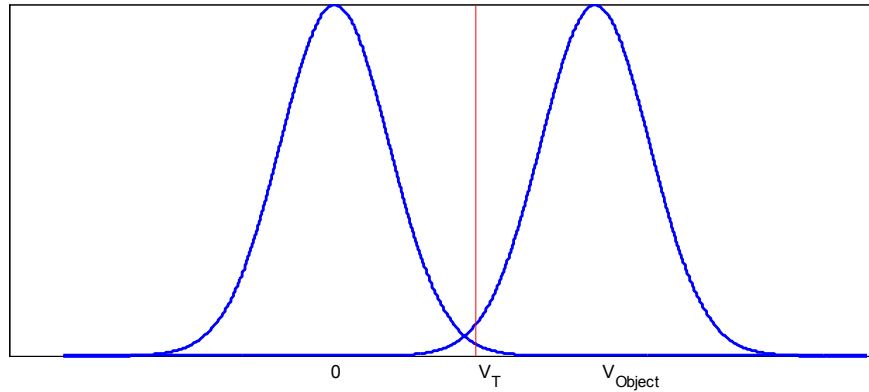
Om niet alle lokale maxima die in de ruis zitten als objecten aan te duiden wordt een drempel toegepast. Alleen pieken die boven de drempel uitkomen worden als object aangemerkt. Het bepalen van het 3-punts maximum in een reeks is rekenkundig eenvoudig. Het bepalen van de grens voor de piekdetectie kan op verschillende manieren. Het is mogelijk om de grens statisch te maken of adaptief. Hier zullen de twee verschillende manieren besproken worden.

Een statische grens dient zo gekozen te worden dat de kans op een vals alarm niet al te groot is, maar ook zodanig dat er geen doelen gemist worden. Een dergelijk probleem is een beslissingsprobleem. In het programma van eisen is gesteld dat er een bepaalde kans mag zijn dat een object niet gedetecteerd wordt. Het bepalen van de miskans is afhankelijk van het type object. De meest belangrijke eigenschap van het object is of de reflectie veranderlijk is of niet [22]. De reflectie van een perfecte bol is niet veranderlijk, aangenomen dat een bal perfect bolvormig is kan een bal ook als niet veranderlijk gezien worden. Als er alleen sprake is van thermische ruis kan de fluctuatie in de reflectie als Gaussisch gezien worden. In dit geval is het beslissingsprobleem goed gedocumenteerd en eenvoudig op te lossen. De miskans is het oppervlakte onder de rechter verdeling in figuur 4.2 dat links van de drempel waarde V_T ligt. De kans op een vals alarm is de oppervlakte onder de linker verdeling in figuur 4.2 dat rechts van de drempel ligt.

Deze kansen zijn te bepalen met behulp van de Q functie [12]. De Q functie is gedefinieerd als

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} e^{-\lambda^2/2} d\lambda \quad (4.1)$$

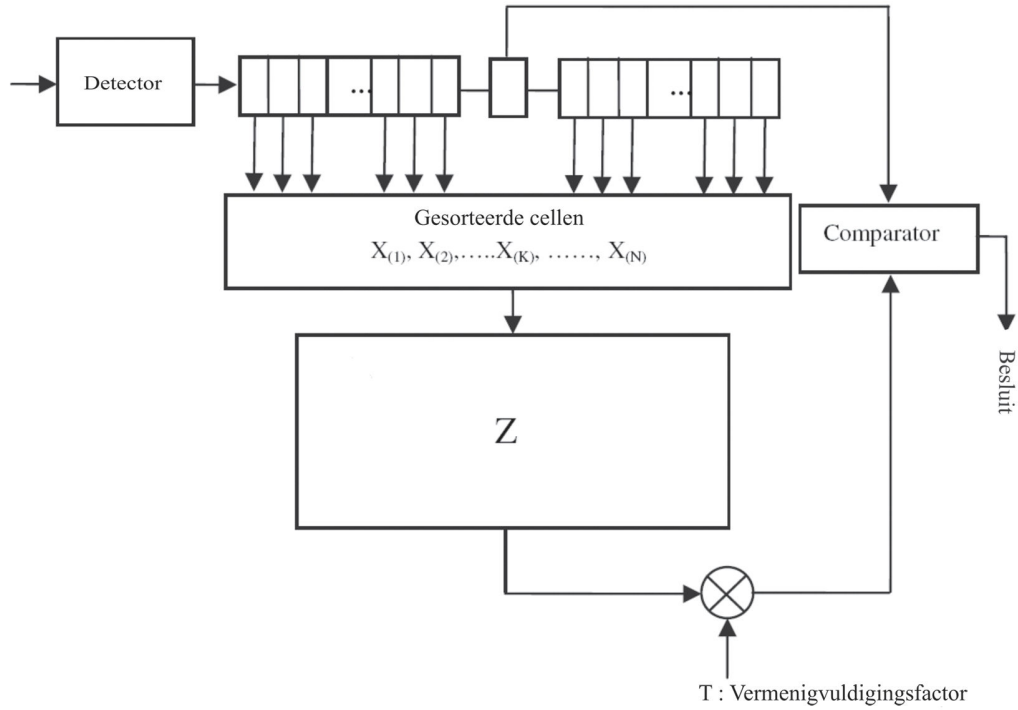
Voor het bepalen van de vals alarm kans wordt $Q\left(\frac{V_T}{\sigma_0}\right)$ berekend, voor het bepalen van de miskans wordt $Q\left(\frac{V_{Object} - V_T}{\sigma_0}\right)$ berekend. In deze vergelijkingen is V_T de drempelwaarde, V_{Object} de gemiddelde waarde van de reflectie van de bal en σ_0 de standaarddeviatie van de ruis.



Figuur 4.2 Kansverdeling van de reflectiesterkte van een object.

Een alternatief voor de hierboven beschreven piekdetectie met een vaste grenswaarde is een piekdetectie met een adaptieve grenswaarde. Veelgebruikte detectors met een adaptieve grenswaarde zijn Constant False Alarm Rate (CFAR)-detectors [19]. De False Alarm rate, en dus de vals alarm kans P_{fa} , van een CFAR-detector is constant.

Een CFAR-detector vergelijkt de waarde van een lokaal maximum, ook wel de Cell Under Test (CUT) genoemd, met N referentiewaarden in een vector X die verdeeld zijn over twee even grote referentiegebieden aan beide zijden van het lokale maximum. De CFAR-detector probeert op deze manier aan de hand van de referentiegebieden een schatting te maken van het lokale ruisniveau en op grond daarvan een drempel te bepalen. Een algoritme verwerkt de waarden uit de referentiegebieden tot een waarde die het lokale gemiddelde ruisniveau Z representeert, waarna de grenswaarde G berekend wordt door de hoogte van de uitkomst te vermenigvuldigen met een factor T . Wanneer de waarde in de CUT hoger ligt dan G wordt het lokale maximum als piek gedetecteerd. Het model voor een CFAR-detector is weergegeven in figuur 4.3.



Figuur 4.3 Het CFAR detector algoritme. Overgenomen en bewerkt uit [1].

Er zijn verschillende algoritmes bekend om de N referentiewaarden in X te verwerken. Voorbeelden zijn Cell Averaging CFAR (CA-CFAR), Greatest-of CFAR (GO-CFAR) en Least-of CFAR (LO-CFAR), waarvoor respectievelijk geldt

$$Z_{CA-CFAR} = \frac{1}{N} \sum_{i=1}^N X_i \quad (4.2)$$

$$Z_{GO-CFAR} = \max(X) \quad (4.3)$$

$$Z_{LO-CFAR} = \min(X) \quad (4.4)$$

Deze algoritmes nemen aan dat de ruisniveaus in de cellen om de CUT homogeen verdeeld zijn. Ze zijn dus goed toepasbaar wanneer het ruisniveau om de pieken homogeen verdeeld is, maar zijn minder goed toepasbaar wanneer de omgeving van de piek bestaat uit clutter of andere pieken. Om dit probleem op te lossen heeft Rohling [26] de Ordered Statistics CFAR (OS-CFAR) detector voorgesteld:

$$Z_{OS-CFAR} = X_k \quad (4.5)$$

De OS-CFAR detector probeert het ruisniveau van de achtergrond af te schatten door de waarden in X te ordenen op grootte en daarna de K^{de} waarde in X te nemen voor Z .

Een andere detector, de Censored Mean Level CFAR (CML-CFAR) detector, is een combinatie tussen de CA-CFAR en de OS-CFAR detector:

$$Z_{CML-CFAR} = \frac{1}{K} \sum_{i=1}^K X_i \quad (4.6)$$

Studies [14], [18], [27] hebben aangetoond dat de CML-CFAR detector beter presteert in omgevingen waar meerdere objecten aanwezig dan de OS-CFAR detector. Voor zowel OS-CFAR als CML-CFAR kan de detectiekans P_d en de vals alarm kans P_{fa} berekend worden. Hierbij wordt aangenomen dat het object in de CUT gemodelleerd kan worden volgens het Swirling I model, en de achtergrondruis Gaussisch verdeeld is [1], [26]. Voor de OS-CFAR detector geldt in dat geval dat

$$P_d = \prod_{j=0}^{K-1} \frac{(N-j)}{\left(N-j + \frac{T}{1+S}\right)} \quad (4.7)$$

$$P_{fa} = \prod_{j=0}^{K-1} \frac{(N-j)}{(N-j+T)} \quad (4.8)$$

waarin S de signaal-ruisverhouding van het te detecteren object is, T is de vermenigvuldigingsfactor en N het aantal referentie cellen. Voor de CML-CFAR detector geldt [1], [14], [18] dat

$$P_d = \binom{N}{K} \prod_{j=0}^{K-1} \left[\frac{T}{1+S} + \frac{N-j+1}{K-j+1} \right]^{-1} \quad (4.9)$$

$$P_{fa} = \binom{N}{K} \prod_{j=0}^{K-1} \left[T + \frac{N-j+1}{K-j+1} \right]^{-1} \quad (4.10)$$

Voor zowel OS-CFAR als CML-CFAR bestaat een vuistregel voor het kiezen van K [25], [26].

4.4 Alternatieven voor het volgen van doelen

Nadat er objecten gedetecteerd zijn, moeten ze ook gevolgd worden om ze te kunnen classificeren. Hiervoor zijn verschillende methodes denkbaar. De rekenkundige complexiteit van de methodes verschilt veel. De meest simpele methode is de dichtstbijzijnde nabuurmethode [10], deze wordt beschreven in paragraaf 4.4.1. Een iets complexere methode maakt gebruik van het Kalman filter [10], [24]. Deze methode wordt beschreven in paragraaf 4.4.2.

4.4.1 Dichtstbijzijnde nabuurmethode

In de dichtstbijzijnde nabuurmethode wordt alleen het vorige frame gebruikt voor het volgen van de baan van een object. De objecten in ieder nieuw frame worden gematcht

met het dichtstbijzijnde object in het vorige frame. De baan van de gevonden objecten bestaat uit de opeenvolgende matches met telkens het vorige frame. Het voordeel van deze methode is dat het relatief weinig rekenwerk kost. Daar staat echter tegenover dat er veel situaties zijn waarin de methode niet erg betrouwbaar is. Voorbeelden hiervan zijn: botsende objecten, dicht bij elkaar bewegende objecten en objecten die moeilijk zichtbaar zijn en daardoor niet in ieder frame gedetecteerd worden. Deze drie situaties komen in het voetbal veel voor. Hierdoor is de dichtstbijzijnde nabuurmethode niet ideaal in termen van nauwkeurigheid.

4.4.2 Het Kalman filter

Een tweede methode om objecten te volgen is door gebruik te maken van een Kalman filter. Het gebruik van het Kalman filter bij radar objecttracking is in de literatuur uitgebreid beschreven. In een toepassing met een radar kunnen er verschillende eendimensionale modellen worden gebruikt. Een eendimensionaal Kalman filter met twee toestanden wordt bijvoorbeeld beschreven door Friedland's model [16]. In het model wordt er vanuit gegaan dat de positie van het object wordt gemeten op tijdstippen met een constant interval T . In het model wordt er vanuit gegaan dat het object beweegt in slechts één richtingscomponent.

Voor elke positie wordt de dynamica van het object beschreven als

$$\begin{aligned}x_{n+1} &= x_n + \dot{x}_n T + \frac{1}{2} a_n T^2 \\ \dot{x}_{n+1} &= \dot{x}_n + a_n T\end{aligned}\tag{4.11}$$

waarin x_n = de positie van het object op meting n

$\dot{x}_n$ = de snelheid van het object op meting n

a_n = de acceleratie van het object op meting n

T = het interval tussen twee metingen

In het model beschreven door (4.11) wordt aangenomen dat de acceleratie een ongecorreleerde random constante is, met een verwachte waarde $E[a_n]$ van nul, dus

$$\begin{aligned}E[a_n] &= 0 \\ E[a_n^2] &= \sigma^2 = \text{constant} \quad \text{voor alle } n \\ E[a_n a_k] &= 0 \quad \text{voor alle } n \neq k\end{aligned}\tag{4.12}$$

In vector-matrix vorm kan (4.11) geschreven worden als

$$X_{n+1} = F X_n + G a_n\tag{4.13}$$

waarbij geldt dat

$$X_n = \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix}\tag{4.14}$$

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (4.15)$$

en

$$G = \begin{bmatrix} T^2 / 2 \\ T \end{bmatrix} \quad (4.16)$$

De positie van een object wordt gemeten met een ruisende radar op uniforme intervallen van T seconden. De meetvergelijking wordt dus gegeven door

$$x_m(n) = x_n + v_n \quad (4.17)$$

waarbij geldt dat $x_m(n)$ = de gemeten positie op meting n
 x_n = de ware positie op meting n
 v_n = ruis die meting n verstoort

De statistische eigenschappen van de ruis worden aangenomen als

$$\begin{aligned} E[v_n] &= 0 \\ E[v_n^2] &= \sigma^2 = \text{constant} \quad \text{voor alle } n \\ E[v_n v_k] &= 0 \quad \text{voor alle } n \neq k \end{aligned} \quad (4.18)$$

In termen van de toestandsvector X_n kan (4.17) ook geschreven worden als

$$x_m(n) = HX_n + v_n \quad (4.19)$$

met

$$H = [1 \quad 0] \quad (4.20)$$

Nu zijn (4.13) en (4.19) in de standaardvorm voor het toepassen van de Kalman filtertheorie. Met deze theorie kan een optimale schatting worden gemaakt van de huidige positie van het object, die voldoet aan Friedland's model.

Modellen voor Kalman filters met drie toestanden kunnen bijvoorbeeld worden beschreven door Ramachandra's eerste en tweede model [24]. Ramachandra's model I en II gaan uit van een bewegend object met een constante acceleratie die verstoord wordt door een ruisbron a_n veroorzaakt door bijvoorbeeld externe krachten. Beide modellen nemen aan dat de ruisbron a_n een ongecorrleerde random constante is met een gemiddelde waarde van 0 en een variantie σ_0^2 . De dynamica van Ramachandra's eerste model wordt beschreven als

$$\begin{aligned}
x_{n+1} &= x_n + \dot{x}_n T + \frac{1}{2} \ddot{x}_n T^2 \\
\dot{x}_{n+1} &= \dot{x}_n + \ddot{x}_n T \\
\ddot{x}_{n+1} &= \ddot{x}_n + a_n
\end{aligned} \tag{4.21}$$

waarin x_n = de positie van het object op meting n

$\dot{x}_n$ = de snelheid van het object op meting n

$\ddot{x}_n$ = de acceleratie van het object op meting n

a_n = ruisbron die de acceleratie verstoort

T = het interval tussen twee metingen

Het dynamisch model genoemd in (4.21) kan in vector-matrix vorm beschreven worden als

$$X_{n+1} = FX_n + A_n \tag{4.22}$$

waarbij geldt dat

$$X_n = \begin{bmatrix} x_n \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} \tag{4.23}$$

$$F = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{4.24}$$

en

$$A_n = \begin{bmatrix} 0 \\ 0 \\ a_n \end{bmatrix} \tag{4.25}$$

De dynamica van Ramachandra's model II wordt beschreven door

$$\begin{aligned}
x_{n+1} &= x_n + \dot{x}_n T + \frac{1}{2} \ddot{x}_n T^2 + a_n T^3 / 6 \\
\dot{x}_{n+1} &= \dot{x}_n + \ddot{x}_n T + a_n T^2 / 2 \\
\ddot{x}_{n+1} &= \ddot{x}_n + a_n T
\end{aligned} \tag{4.26}$$

waarin x_n = de positie van het object op meting n

$\dot{x}_n$ = de snelheid van het object op meting n

$\ddot{x}_n$ = de acceleratie van het object op meting n

a_n = ruisbron die de acceleratie verstoort

T = het interval tussen twee metingen

De meetvergelijking voor Ramachandra's model I en II wordt gegeven door (4.19), waarbij H nu gelijk is aan

$$H = [1 \quad 0 \quad 0] \quad (4.27)$$

De statistische eigenschappen van het meetmodel worden hetzelfde verondersteld als voor (4.19).

Ook Ramachandra's model II kan worden beschreven in vector-matrix vorm,

$$X_{n+1} = FX_n + Ga_n \quad (4.28)$$

waarbij X_n en F gedefinieerd worden als in (4.22). G wordt gegeven door

$$G = \begin{bmatrix} T^3 / 6 \\ T^2 / 2 \\ T \end{bmatrix} \quad (4.29)$$

Door met behulp van de modellen beschreven in (4.11), (4.21) of (4.26) een voorspelling te maken van de volgende positie in een track, en deze informatie te gebruiken bij het koppelen van opeenvolgende pieken kan eenvoudig een track gevormd worden. Voordelen die optreden wanneer gebruik gemaakt wordt van het Kalman filter ten opzichte van de in paragraaf 4.4.1 beschreven dichtstbijzijnde nabuurmethode zijn dat het systeem beter overweg kan met passerende objecten en dichtbij elkaar bewegende objecten. Daarnaast is het systeem minder gevoelig voor wegvallende pieken, omdat nu op basis van de dynamica van een object pieken aan elkaar gekoppeld worden tot een baan. Een nadeel van het Kalman filter is dat het systeem ernstig ontregeld wordt wanneer er botsingen optreden. Immers, er wordt verondersteld dat afhankelijk van het gekozen model de gemiddelde snelheid of versnelling nul is. Een ander nadeel is dat het algoritme rekenkundig complexer is dan de dichtstbijzijnde nabuurmethode.

4.5 Alternatieven voor het onderscheiden van speler en bal

Nadat de objecten zowel gevonden als gevolgd kunnen worden, moet er bepaald worden of een van de objecten de bal is. Een bal ziet er voor een radar op een aantal punten anders uit dan een mens. De meest voorname verschillen tussen een bal en een mens zijn de polarisatie, de snelheid, en de mate waarin de radardoorsnede constant is. Deze drie verschillen worden in deze paragraaf beschreven, daarnaast wordt ook een methode beschreven om de bal kunstmatig beter zichtbaar te maken.

4.5.1 Polarisatie

Een elektromagnetische golf kan op verschillende manieren gepolariseerd zijn [28]. De polarisatie is de baan die het eindpunt van het elektrische veld beschrijft in de tijd. De baan kan een cirkel, ellips of een lijn zijn. De manier waarop een elektromagnetische golf gepolariseerd is, heeft invloed op de mate waarin die gereflecteerd wordt door een object. Over het algemeen wordt een lineair gepolariseerde golf het beste gereflecteerd door een object dat in dezelfde richting als de golf gepolariseerd is. Een rechtopstaande metalen staaf bijvoorbeeld, reflecteert een verticaal gepolariseerde golf vele malen beter dan een horizontaal gepolariseerde golf. Dit verschijnsel kan gebruikt worden bij het onderscheiden van een mens en een bal.

Een bal reflecteert golven van iedere polarisatie even goed omdat hij rond is. Een mens is daarentegen langer dan dat hij breed is. Hierdoor reflecteert een mens golven die langs het lichaam gepolariseerd zijn beter [30]. Een radar die kan schakelen tussen horizontale en verticale polarisatie kan gebruikt worden om mensen uit de reflectie te filteren. Pieken die significant in hoogte verschillen bij verticale polarisatie ten opzichte van horizontale gaan pulseren als de radar afwisselend in horizontale en verticale polarisatie geschakeld wordt. Dit pulseren kan gedetecteerd worden door de software. Deze objecten worden dan als potentiële mensen geclassificeerd. Het is ook mogelijk om twee verschillende gepolariseerde radars te gebruiken. De pieken die op de ene radar wel te zien zijn, maar op de andere radar veel minder te zien zijn, hoeven dan niet verder verwerkt te worden. Een bijkomend voordeel van het toepassen van deze techniek is dat ook het doel geclassificeerd kan worden. De doelpalen zijn net als mensen verticaal gepolariseerd terwijl de lat juist horizontaal gepolariseerd is.

4.5.2 Snelheid

De bal en spelers kunnen in sommige situaties eenvoudig uit elkaar gehouden worden door naar de snelheid te kijken. Kort gesteld kan in het spel alleen de bal sneller bewegen dan een mens kan rennen. Een professionele voetballer kan de bal een snelheid tot ongeveer 30 m/s meegeven terwijl de gemiddelde snelheid op de 60 meter sprint tijdens het huidige wereldrecord op ongeveer 10 m/s ligt [9], [11]. Er is dus een snelheidsgrens te bepalen waarboven bewegende objecten een bal moeten zijn. Deze situatie doet zich natuurlijk niet altijd voor, maar kan in combinatie met een goed trackingsysteem weldegelijk waardevol zijn. Wanneer het trackingsysteem in staat is om de bal goed te volgen, hoeft de bal maar heel even boven de snelheidsgrens uit te komen om de bal te kunnen classificeren gedurende de rest van de actie. Daarnaast vallen veel doelpunten uit schoten op doel. Tijdens een schot op doel is de kans erg groot dat de bal sneller gaat dan de vastgestelde grens en is dus goed te classificeren. Het voordeel van deze methode is dat hij weinig rekenkracht kost en in sommige gevallen met grote zekerheid kan beslissen wat de bal is. Het grootste nadeel is dat de methode niet in alle situaties werkt.

4.5.3 Radardoorsnede

De radardoorsnede van een object wordt bepaald door de hoeveelheid teruggekaatste energie van een object te relateren aan die van een perfecte bol. De radardoorsnede hangt af van de drie factoren: het frontaal oppervlak, de reflectivity en de directivity. De reflectivity is het vermogen dat door het object gereflecteerd wordt. De directivity is het vermogen dat in de richting van de radar uitgezonden wordt [21].

Aangezien de radardoorsnede gerelateerd wordt aan een perfecte bol is het aan te nemen dat de radardoorsnede van een voetbal vrijwel constant is. De bal wordt immers zo gemaakt dat hij zo rond mogelijk is. Om de reflectivity van de bal groter te maken kan de bal voorzien worden van een aluminium laagje. De radardoorsnede van de bal zal niet geheel constant zijn omdat de bal tijdens het schieten vervormt. Deze vervorming is echter niet heel groot en van zeer tijdelijke aard.

De radardoorsnede van een mens is verre van constant. Zeker tijdens het rennen zullen zowel het frontaal oppervlak als de directivity sterk veranderen in de tijd. Het frontaal oppervlak en de directivity variëren tijdens het rennen omdat benen en armen bewegen. De directivity varieert daarnaast ook door de hoek waaronder het bovenlichaam staat.

Als de radardoorsnede van een object fluctueert kan dat op de radar gezien worden als een piek die in hoogte varieert. De hoeveelheid energie die het object reflecteert, hangt immers onder andere van de radardoorsnede af. Als de software in staat is deze variaties te detecteren, kunnen objecten die veel fluctueren als mens geïdentificeerd worden. Het detecteren van deze variaties kost op zich niet heel veel rekenkracht, wat een voordeel is, maar heeft wel een goed trackingsysteem nodig.

4.5.4 RFID-tags

Alle hierboven beschreven methodes zijn gebaseerd op de eigenschappen van de objecten zoals ze in het spel voorkomen. In deze paragraaf zal ingegaan worden op een methode om de bal beter zicht- en onderscheidbaar te maken met behulp van RFID-tags.

Radio-frequency identification (RFID) [20] is een identificatietechniek die gebruik maakt van RFID-tags of transponders. Een RFID-tag vangt de energie van de continue golf waarmee hij wordt belicht op, voert een modulatie uit en stuurt de gemoduleerde respons de ether in. Er bestaan verschillende types RFID-tags; passieve, actieve en semi-passieve tags. Passieve RFID-tags gebruiken geen externe energievoorziening. Alle energie die ze nodig hebben wordt gehaald uit het ingevangen signaal. Actieve RFID-tags maken wel gebruik van externe energievoorziening, voor zowel het verzenden en ontvangen van het signaal als het moduleren van het signaal. Tot slot bestaan er semi-passieve RFID-tags, deze RFID-tags maken alleen gebruik van externe energievoorziening voor het verwerken van het signaal [23], [29].

Eén of meerdere passieve RFID-tags zouden op een bal geplaatst kunnen worden. Een passieve RFID-tag is klein, flexibel en niet zwaar en hoeft de eigenschappen van de voetbal dus niet te verstoren. De RFID-tag werkt in deze opstelling op dezelfde frequentie als de radar, en stuurt wanneer hij belicht wordt door een radar een gemoduleerd signaal terug de ether in. Met behulp van radarsoftware is het mogelijk om dit gemoduleerde signaal te herkennen, waardoor de voetbal herkend kan worden. Een groot voordeel van deze techniek is dat de bal goed herkenbaar is. Een nadeel van gebruik van RFID-tags is dat de kosten om ze te op kleine schaal te produceren erg hoog zijn. Ook moet de frequentie waarop de RFID werkt, gematcht worden met de frequentie waarop de radar werkt.

4.6 Alternatieven voor programmeertalen

De prestaties en ontwikkeltijd van de software hangt sterk af van de taal waarin ze geschreven is. In deze paragraaf worden twee alternatieven voor de te gebruiken taal weergegeven. Hierbij zal gekeken worden naar de snelheid van de implementatie en de ontwikkeltijd. De twee software alternatieven zijn Matlab en C.

4.6.1 Matlab

Matlab is een veel gebruikt programma om technisch rekenwerk mee te doen. De zeer uitgebreide toolkits en beschikbare routines maken het gemakkelijk om snel veel werk gedaan te krijgen. De routines zijn veelal geoptimaliseerd op snelheid. Het grootste voordeel van Matlab is dat het gemaakt is om rekenwerk te doen. Een programma kan snel geschreven worden omdat er al veel routines beschikbaar zijn. Voor het ontwikkelen van een prototype en voor het bewijzen van concepten is Matlab bij uitstek geschikt. Het grootste nadeel van Matlab is dat het, ondanks dat veel routines op snelheid geoptimaliseerd zijn, niet snel genoeg is om real-time zware taken uit te voeren.

4.6.2 C

C is een krachtige en flexibele programmeertaal. Een goed geschreven en geoptimaliseerde C code kan erg snel uitgevoerd worden. Er zijn veel compilers beschikbaar waardoor het mogelijk is C code op verschillende processors en systemen uit te voeren. De snelheid van C komt echter niet zonder nadelen. Ondanks het feit dat er veel voorgeprogrammeerde routines beschikbaar zijn in C zal er veel programmeerwerk gedaan moeten worden. Het ontwikkelen van C code is alles behalve gemakkelijk als het programma ingewikkelder wordt. De ontwikkeltijd neemt aanzienlijk toe ten opzichte van Matlab. Hoewel het programma sneller zal zijn als het in C geschreven is, duurt het ontwikkelen veel langer.

4.7 Alternatieven voor de bediening van het systeem

Het programma van eisen stelt dat er twee modi moeten zijn waarin het systeem opereert, de standaardmodus en de onderhoudsmodus. Deze paragraaf gaat in op twee alternatieven om het systeem te bedienen: een textuele en een grafische oplossing.

4.7.1 Textuele interface

Een mogelijkheid om het systeem te bedienen is door gebruik te maken van een textuele interface, vergelijkbaar met een command line. Als het systeem is opgestart verschijnen er menu's op het scherm die de standaardmodus en de onderhoudsmodus voorstellen. Een voordeel van een textuele interface is dat alle niet-grafische delen binnen de interface heel eenvoudig te ontwikkelen zijn. Een groot nadeel is echter dat een dergelijke interface niet echt gebruiksvriendelijk is en niet eenvoudig om zal kunnen gaan met afbeeldingen.

4.7.2 Grafische interface

Een alternatief voor de textuele interface is een grafische interface. Een oplossing voor de bediening in de standaardmodus wordt gegeven in figuur 4.4. Een eenvoudige en

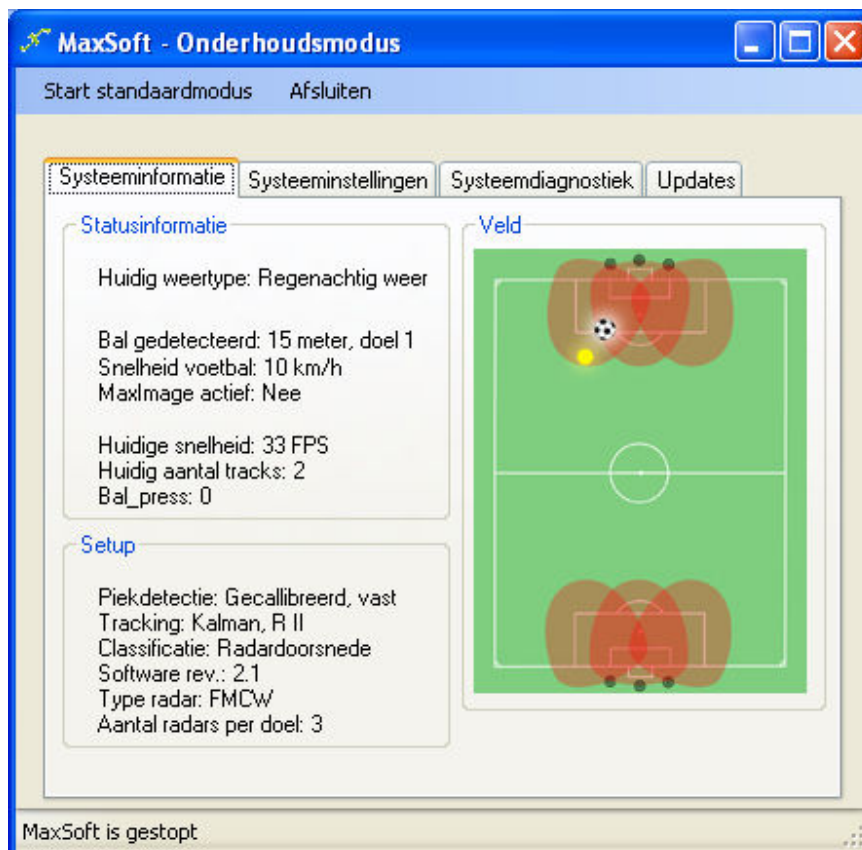
toegankelijke interface geeft basismeetgegevens weer. In hetzelfde venster is er de mogelijkheid om het systeem in te stellen op een ander weertype.

Door te klikken op 'Start onderhoudsmodus' belandt de gebruiker in de onderhoudsmodus. Een voorbeeld van een grafische interface van de onderhoudsmodus wordt weergegeven in figuur 4.5. De verschillende onderdelen van de onderhoudsmodus kunnen overzichtelijk in tabbladen worden weergegeven en afbeeldingen kunnen eenvoudig in de interface worden verwerkt.

Een groot voordeel van de grafische interface ten opzichte van de textuele interface is dat hij veel gebruiksvriendelijker is. Daarnaast kan in een grafische interface de informatie beter gevisualiseerd worden voor gebruikers.



Figuur 4.4 Voorbeeld standaardmodus.



Figuur 4.5 Voorbeeld onderhoudsmodus.

5 Verwant onderzoek

Om te bepalen of de concepten die in hoofdstuk 4 beschreven zijn in het prototype te testen zijn was nader onderzoek nodig. Vooral de verschillende deelconcepten die te maken hebben met piekdetectie en objectclassificatie zijn onderzocht. Op het gebied van piekdetectie is gekeken naar zowel de bepaling van een vaste drempelwaarde als naar parametersets voor adaptieve drempels met de beschikbare radar. Daarnaast is op het gebied van objectclassificatie gekeken naar de vraag of verschillende methodes om een mens en een voetbal te onderscheiden haalbaar zijn met de beschikbare radar. Het polarisatie verschil is onderzocht voor een mens en een voetbal, en er is gekeken naar constantheid van de reflectie voor een mens en een voetbal.

De opbouw van dit hoofdstuk is als volgt. In paragraaf 5.1 wordt de meetopstelling beschreven. Vervolgens worden in paragraaf 5.2 verschillende methoden van piekdetectie behandeld, en in paragraaf 5.3 worden een aantal van de reeds genoemde methodes voor objectclassificatie onderzocht. Tenslotte volgen de conclusies uit het onderzoek in paragraaf 5.4.

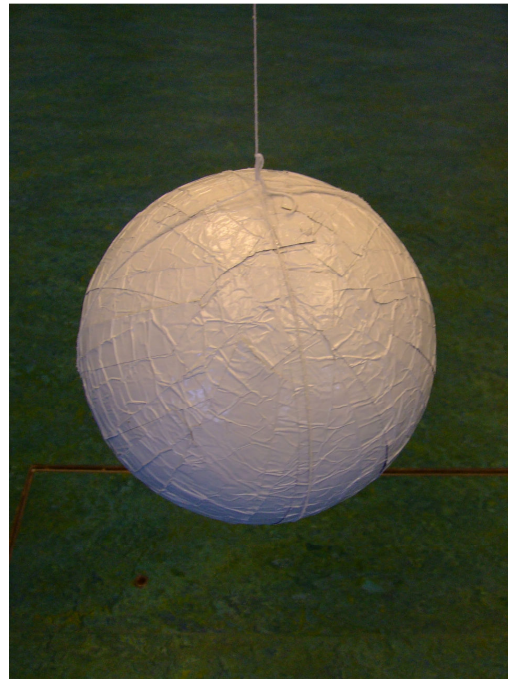
5.1 Meetopstelling

De metingen vonden allemaal plaats op de 23^e verdieping van het EWI gebouw. De metingen vonden dus binnen plaats. De gebruikte radar is een FMCW-radar op een werkfrequentie van 11 GHz. De frequentiesweep is 1 GHz en de sweeptijd 11,7 ms. De gebruikte antenne is een halve paraboolantenne met een gain van 32 dB en een -3 dB bundelbreedte van 5,4 graden in azimuth en 4,9 graden in elevatie. De afstandsresolutie bij de 1 GHz sweep is 20 cm. Het zendvermogen is ongeveer 4 mW.

Om de bal in een rechte lijn en op een reproduceerbare wijze naar de radar toe te kunnen bewegen is aan het plafond een rails bevestigd. De bal hangt aan een touwtje van de rails en kan over de rails bewogen worden, de baan van de bal ligt dan volledig binnen de radarbundel. De bal is bekleed met aluminium om hem voor de radar beter zichtbaar te maken. Het begin van de rails bevindt zich op 3,45 meter van de radar, de rails is 3 meter lang. In figuur 5.1 is een overzicht gegeven van de meetopstelling.



Figuur 5.1 Meetopstelling.



Figuur 5.2 Close-up van de bal.

5.2 Bepaling van de parameters van de piekdetectie

Om de verschillende methodes voor de drempelwaarde van de piekdetectie te kunnen vergelijken moeten eerst de parameters voor de systemen vastgesteld worden. Voor de vaste drempel betekent dat de waarde van die drempel. Voor de CFAR detectors zijn dat het aantal referentiecellen, de methode van schatting van het ruisniveau en de vermenigvuldigingsfactor. Alleen OS-CFAR en CML-CFAR worden behandeld, in paragraaf 4.3.2 wordt immers bevonden dat de andere detectors minder geschikt zijn om meerdere objecten te detecteren. Dit wordt in onze toepassing wel van het piekdetectie algoritme verwacht.

In paragraaf 5.2.1 wordt beschreven hoe een vaste grenswaarde bepaald is. Voor de OS-CFAR en CML-CFAR detector zijn optimale parameters voor het prototype bepaald, gebaseerd op verschillende metingen die uitgevoerd werden met de opstelling beschreven in paragraaf 5.1. De verantwoording van de keuze van de parameters is te lezen in paragraaf 5.2.2 en 5.2.3.

5.2.1 Bepaling vaste grenswaarde

Voor het bepalen van de vaste grenswaarde wordt uitgegaan van een vals alarm kans, P_{fa} , van 0,01. Deze waarde is tot stand gekomen tijdens het bepalen van de parameters voor de CFAR detectors, die hieronder beschreven worden. Zoals beschreven in hoofdstuk vier geldt dat

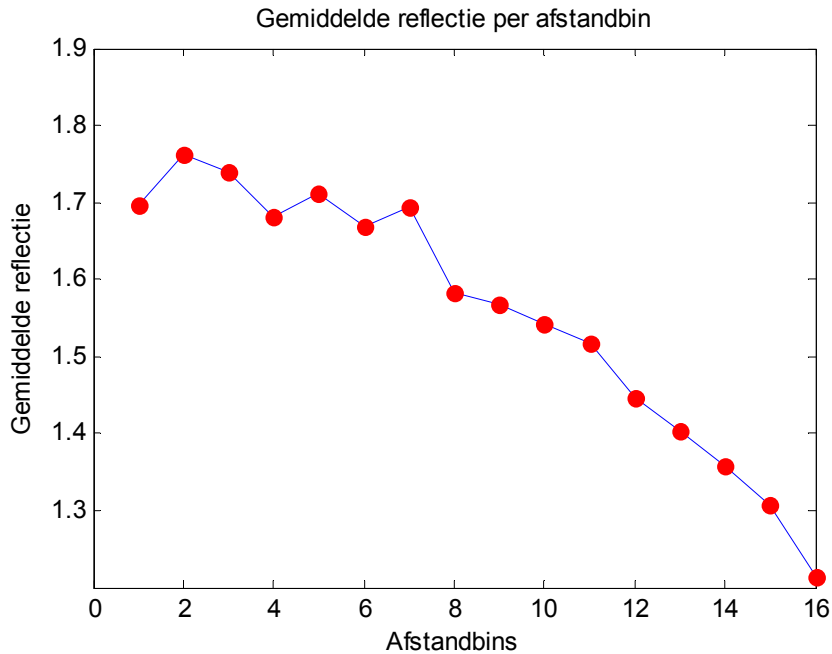
$$P_{fa} = Q\left(\frac{V_T}{\sigma_0}\right) \quad (5.1)$$

Om de grenswaarde te kunnen bepalen moet dus eerst σ_0 bekend zijn. Deze waarde is te bepalen uit het gemiddelde ruisvermogen, omdat geldt dat $P_{av,N} = \sigma_0^2$ [12]. Het gemiddelde ruisvermogen is vastgesteld op -17,6 dB door in de GUI te kijken naar de waarden waar de grafiek rond springt op punten waar geen objecten zijn. Met het gemiddelde ruisvermogen en de vastgestelde P_{fa} is de drempelwaarde V_T berekend op 0,34.

Om te weten of de drempelwaarde die bepaald is niet leidt tot veel gemiste objecten is het van belang om de kans op een gemist object, P_{mo} , te bepalen. De kans op een gemist object wordt gegeven door vergelijking (5.2)

$$P_{mo} = Q\left(\frac{V_{Object} - V_T}{\sigma_0}\right) \quad (5.2)$$

In deze vergelijking is alleen V_{Object} onbekend. Om de gemiddelde reflectie van de bal te verkrijgen zijn vijftientig metingen uitgevoerd waarin de bal telkens langzaam naar de radar toe beweegt. Vervolgens is voor iedere afstandsbin de waarde van de reflectie bepaald op het moment dat de bal zich in die bin bevond. Deze waarden zijn gemiddeld voor iedere bin en weergegeven in figuur 5.3.



Figuur 5.3 Gemiddelde reflectie per afstandbin.

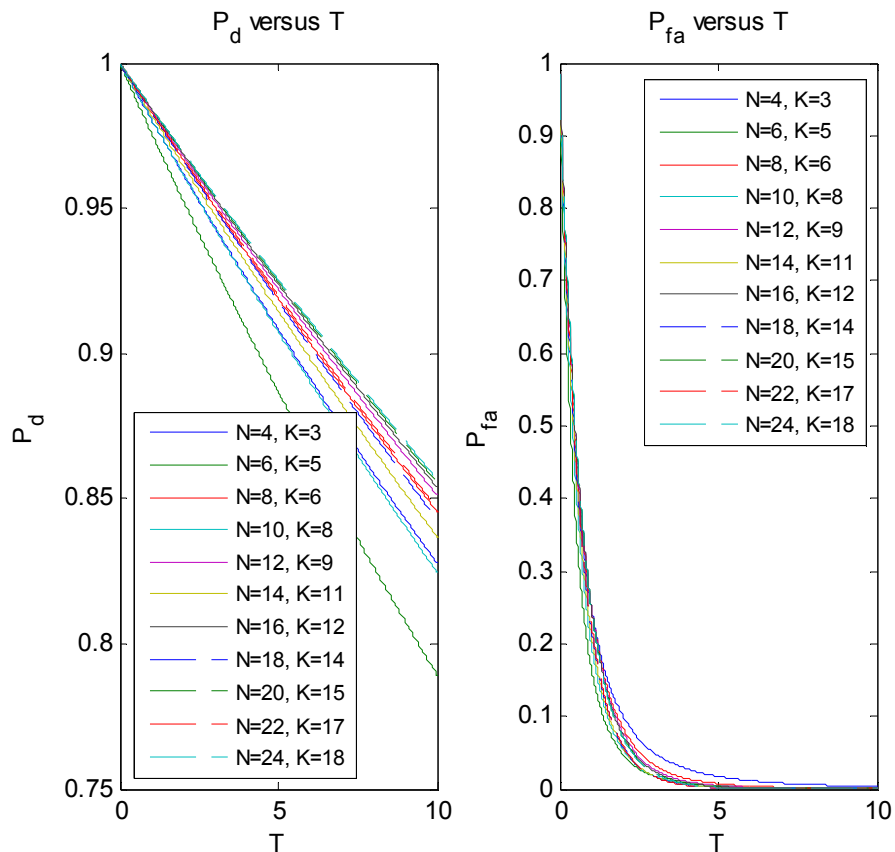
Zoals verwacht neemt de reflectie met de kwadraat van de afstand af. Om te kijken of er een kans bestaat dat er ballen gemist worden met de hierboven vastgestelde drempel is P_{mo} berekend voor zowel de maximale als minimale reflectie. De resultaten hiervan zijn gegeven in tabel 5.1. Het is duidelijk dat in een situatie waarin er zich niets tussen de radar en de bal bevindt de kans dat de bal gemist wordt vrijwel nul is.

Tabel 5.1 Kans op een gemiste bal per afstandsbin.

	P_{mo}
Maximale reflectie	$1,97 \cdot 10^{-9}$
Minimale reflectie	$3,71 \cdot 10^{-22}$

5.2.2 Bepaling optimale parameters OS-CFAR detector

De parameters van de OS-CFAR detector die bepaald moeten worden zijn het aantal referentiewaardes N , welke waarde K in de gesorteerde reeks van referentiewaardes wordt gebruikt voor Z en de vermenigvuldigingsfactor T . Een vuistregel voorgesteld in [25] is om K gelijk aan $3N/4$ te kiezen. De gemiddelde signaal-ruisverhouding S van de piek van een voetbal is vastgesteld op 19,65 dB, deze S volgt uit verschillende metingen gedaan met de opstelling beschreven in paragraaf 5.1.



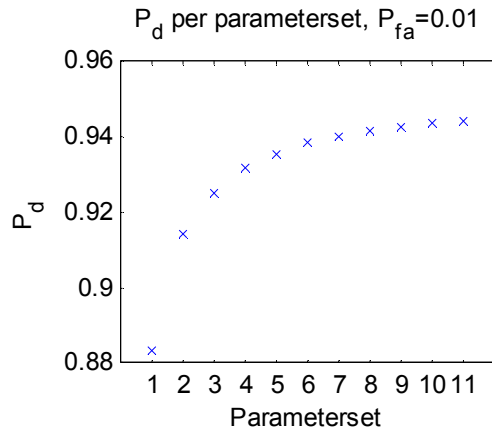
Figuur 5.4 P_d en P_{fa} versus T voor een OS-CFAR detector.

Voor elke even waarde van N op het interval $[4,24]$ zijn P_d en P_{fa} voor verschillende waarden van T berekend. K is telkens gekozen op de afgeronde waarde van $3N/4$, zoals beschreven in [25]. De resultaten zijn weergegeven in figuur 5.4. Een zo hoog mogelijke P_d bij een acceptabele P_{fa} geeft de beste parameterkeuze. Wanneer er immers teveel valse detecties plaatsvinden zal dit het trackinggedeelte van het systeem onnodig belasten. Experimenteel is bepaald dat $P_{fa} = 0.01$ een goede waarde is voor de ‘False alarm’-rate. De keuze van een grote N zal het sorteer-algoritme aanzienlijk vertragen. Het is daarom ook van belang om een zo goed mogelijk resultaat te halen met een minimale N .

In tabel 5.2 en figuur 5.5 zijn de kans op detectie en de bijbehorende T per parameterset weergegeven, waarbij $P_{fa} = 0,01$. De waarde van T is af te lezen uit de grafiek in figuur 5.4.

Tabel 5.2 P_d en T per parameterset voor OS-CFAR, $P_{fa}=0,01$.

Parameter set	N	K	P_d	T
1	4	3	0,8833	6,48
2	6	5	0,9145	3,70
3	8	6	0,9252	4,59
4	10	8	0,9319	3,62
5	12	9	0,9356	4,11
6	14	11	0,9383	3,55
7	16	12	0,9400	3,90
8	18	14	0,9415	3,51
9	20	15	0,9426	3,78
10	22	17	0,9435	3,48
11	24	18	0,9442	3,70



Figuur 5.5 P_d per parameterset voor OS-CFAR, $P_{fa}=0,01$.

Op grond van bovenstaande resultaten is gekozen voor de parameterset $N=12$, $K=9$ en $T=4,11$. Een grotere N zal het systeem onnodig vertragen en draagt slechts weinig bij aan een verbetering van P_d .

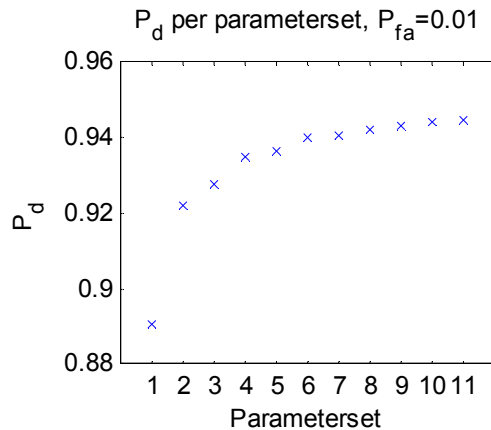
5.2.3 Bepaling optimale parameters CML-CFAR detector

De parameters van de CML-CFAR detector die bepaald moeten worden zijn het aantal referentiewaarden N , het aantal K waarden die worden gemiddeld om Z te bepalen en de vermenigvuldigingsfactor T . De signaal-ruisverhouding S is hetzelfde genomen als in 5.2.2.

In [26] wordt beschreven dat de optimale waarde voor K gelijk is aan $3N/4$. Net als in paragraaf 5.2.1 wordt P_{fa} vastgesteld op 0,01 en wordt P_d berekend, de resultaten zijn weergegeven in tabel 5.3 en figuur 5.6.

Tabel 5.3 P_d en T per parameterset voor CML-CFAR, $P_{fa}=0,01$.

Parameter set	N	K	P_d	T
1	$N=4$	3	0,8904	5,77
2	$N=6$	5	0,9221	2,15
3	$N=8$	6	0,9277	1,98
4	$N=10$	8	0,9348	1,23
5	$N=12$	9	0,9367	1,18
6	$N=14$	11	0,9399	0,86
7	$N=16$	12	0,9407	0,84
8	$N=18$	14	0,9419	0,67
9	$N=20$	15	0,9431	0,65
10	$N=22$	17	0,9441	0,54
11	$N=24$	18	0,9447	0,53



Figuur 5.6 P_d per parameterset voor CML-CFAR, $P_{fa}=0,01$.

Op grond van bovenstaande resultaten zijn de parameters voor de CML-CFAR detector gekozen als $N=14$, $K=11$ en $T = 0,86$. Een grotere N zal immers bijdragen aan een trager systeem en draagt slechts weinig bij aan een verbetering van P_d .

5.3 Objectclassificatie onderzocht

Om na te gaan welke methodes in het prototype bruikbaar zijn bij het classificeren van objecten zijn een aantal deelconcepten onderzocht. Er is gekeken naar de bruikbaarheid van polarisatie van objecten en de mate van constantheid van objectreflecties. De resultaten zijn respectievelijk in paragraaf 5.3.1 en paragraaf 5.3.2 te vinden.

5.3.1 Polarisatiemetingen

Om er zeker van te zijn dat de radar die beschikbaar is voor het prototype lineair gepolariseerd is, is er een meting uitgevoerd op een metalen draad. De draad is verticaal gespannen voor de radar op een afstand van 3,5 meter. Vervolgens is de feedhorn van de antenne gedraaid om te zien of er verandering in de reflectie optreedt. Dit bleekt het geval te zijn. De hoek tussen de maximale en minimale reflectie is 90 graden, er kan dus geconcludeerd worden dat de antenne lineair gepolariseerd is.

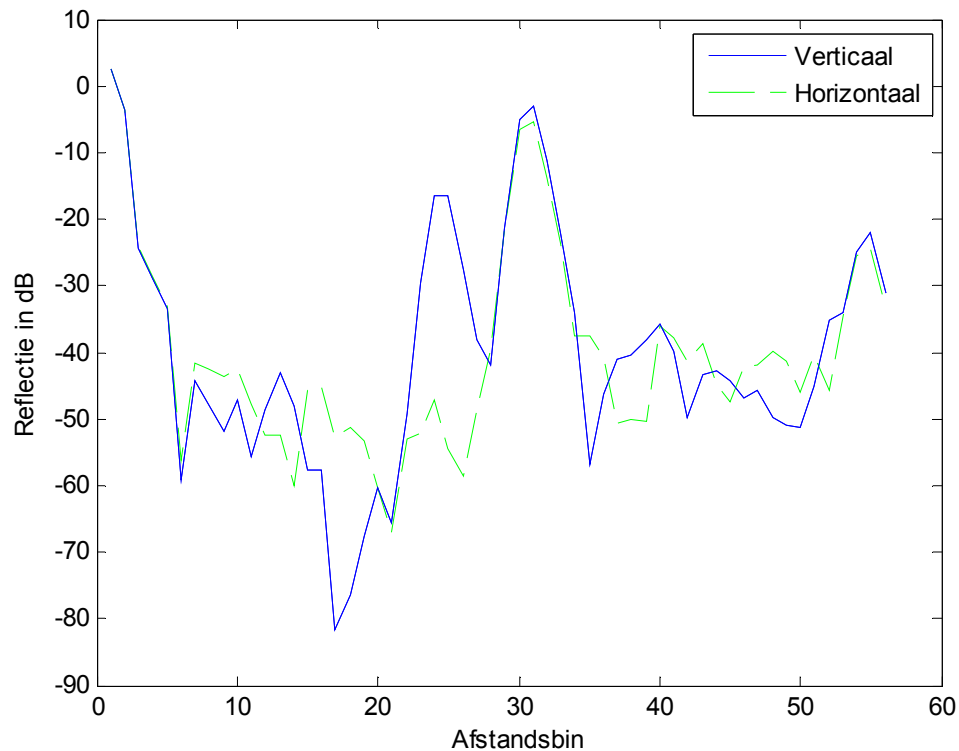
Vervolgens zijn er twee metingen van 30 frames lang gedaan. De eerste meting was met de antenne in verticale polarisatie, de tweede met de antenne in horizontale polarisatie. De reflecties binnen de afstandsbin waarin de draad zich bevond, zijn voor beide metingen gemiddeld en weergegeven in tabel 5.4.

Tabel 5.4 Gemeten reflecties bij verschillende polarisaties.

Polarisatie	Gemiddelde reflectie (in dB)
Verticaal	-7,08
Horizontaal	-22,22

Het is duidelijk te zien dat de verticale geleidende draad voor een horizontaal gepolariseerde radar niet te zien is, de reflectie is 15,14 dB lager dan bij verticale

polarisatie. Dit is te zien in figuur 5.7. In deze figuur is een enkele meting van de draad met zowel verticale (doorgetrokken lijn) en horizontale polarisatie (gestippelde lijn) getekend. De draad bevindt zich in afstandsbin 24.



Figuur 5.7 Verschil in reflectie van een verticale draad bij verschillende polarisatie.

Het is duidelijk dat er zich in de ene grafiek wel een piek bevindt in afstandsbin 24 en in de ander niet. De piek achter de draad is de bal.

Nadat gebleken was dat de radarbundel lineair gepolariseerd is, zijn ook metingen gedaan met de bal en een mens. De metingen zijn op precies dezelfde manier verlopen als hierboven voor de draad beschreven staat. De resultaten zijn gegeven in tabel 5.5.

Tabel 5.5 Verschil in reflectie tussen bal en mens bij verschillende polarisaties.

Object / Polarisatie	Gemiddelde reflectie (in dB)
Bal / Verticaal	-1,08
Bal / Horizontaal	-2,13
Mens / Verticaal	2,14
Mens / Horizontaal	0,24

Het verschil tussen horizontale en verticale polarisatie voor de bal is 1,05 dB en voor een mens 1,9 dB. Het verschil is niet zo groot als bij de draad. Het was ook niet verwacht dat reflecties van de bal zouden verschillen bij ongelijk gepolariseerde bundels. Een verklaring hiervoor is dat de aluminiumbekleding van de bal niet uniform verdeeld is. De aluminium bekleding is in verticale strips aangebracht en niet perfect bolvormig. Dit is te

zien in figuur 5.2. Dat uit deze metingen niet blijkt dat een mens overduidelijk in een bepaalde richting gepolariseerd is, zou verklaard kunnen worden door het feit dat de bundel maar een klein deel van de persoon belicht. De persoon staat op een afstand van 3,5 meter en de bundelbreedte is 5,4 graden in azimuth. Hieruit volgt dat de bundel op 3,5 meter slechts 33 cm breed is. Het frontaal oppervlak van een mens is meer dan 45 cm breed. Omdat de bundel kleiner is dan de persoon in plaats van andersom is de invloed van polarisatie veel minder groot.

5.3.2 Mate van constant zijn van reflectie

Uit hoofdstuk 4 kwam naar voren dat de radardoorsnede van een voetbal constanter zou zijn dan die van een mens. Om te controleren of dit waarneembaar is met de radar, zijn de volgende twee metingen uitgevoerd.

In de eerste meting is de reflectie gemeten van zowel de bal als een persoon terwijl deze stil staat. Vervolgens is van telkens vijf opvolgende frames de standaarddeviatie van de pieken van de objecten bepaald en geschaald naar de gemiddelde waarde van piek over die frames. Dit is voor vijf opvolgende sets van vijf frames gedaan. De resultaten van deze meting zijn gegeven in tabel 5.6.

Tabel 5.6 Geschaalde standaarddeviatie van een stilstaand object.

Object	Frame					Gemiddeld
	15-20	20-25	25-30	30-35	35-40	
Mens	0,0168	0,0455	0,0697	0,0700	0,0442	0,0492
Bal	0,0083	0,0130	0,0009	0,0024	0,0018	0,0053

Uit de gegevens blijkt dat de geschaalde standaarddeviatie van de bal een factor tien kleiner is dan die van de persoon. Dit is volgens verwachting. De verklaring voor het verschil in de variatie tussen een stilstaande persoon en een bal moet gezocht worden in het feit dat een persoon nooit helemaal stil staat. Een persoon zal altijd ademen en onwillekeurige bewegingen maken. Dit resulteert allemaal in variatie in de reflectie.

De tweede meting is wat methode betreft gelijk aan de eerste behalve dat de bal en persoon nu naar de radar toe bewegen. Van deze meting wordt weer voor de pieken de standaarddeviatie geschaald met de gemiddelde waarde. De resultaten hiervan zijn gegeven in tabel 5.7.

Tabel 5.7 Geschaalde standaarddeviatie van een bewegend object.

Object	Frame					Gemiddeld
	15-20	20-25	25-30	30-35	35-40	
Mens	0,5465	0,4143	0,3809	0,6206	0,6244	0,5173
Bal	0,0615	0,0710	0,0874	0,0580	0,0629	0,0682

Ook bij bewegende objecten is het verschil tussen de bal en een mens ongeveer een factor tien. Wel is de variatie met een eveneens een factor tien hoger dan bij de meting met stilstaande objecten. Voor de bewegende mens is dit te verwachten. De loopbeweging van een mens zorgt immers voor een telkens veranderende houding en dus reflectie. Ook voor de bal is een hogere waarde wel te verwachten, omdat ondanks dat de bal via een rails in een stabiele baan naar de radar toe beweegt en de bal als perfect rond beschouwd wordt, door de versnelling de bal omhoog komt. Ook is de rails niet perfect glad

waardoor de bal toch enigszins op en neer beweegt. Daarnaast is het oppervlakte van de bal niet perfect glad, zoals te zien is in figuur 5.2. Dit leidt tot de niet ideale situatie waarin een stilstaand persoon in dezelfde ordegrootte van geschaalde standaarddeviatie komt als een bewegende bal. Dit zal een zwak punt zijn in de classificatie.

5.4 Conclusie

In paragraaf 5.2 zijn voor de verschillende methodes voor objectdetectie de parameters bepaald uitgaande van een vals alarm kans P_{fa} van 0,01. De vaste drempel is voor deze P_{fa} berekend op 0,34. De optimale parameterset voor OS-CFAR is voor deze vals alarm kans in paragraaf 5.3.2 bepaald op $N=12$, $K=9$ en $T=4,11$. Ook voor een CML-CFAR detector is de optimale parameterset bepaald, deze bestaat uit $N=14$, $K=11$ en $T=0,86$.

Uit het onderzoek met de draad is gebleken dat de radar die voor het prototype gebruikt wordt, lineair gepolariseerd is en dus gebruikt kan worden om objecten met een langwerpige vorm te onderscheiden van objecten met een ronde vorm. Omdat uit literatuuronderzoek in hoofdstuk 4 gebleken is dat mensen hogere reflecties veroorzaken wanneer de polarisatie van de radar langs het lichaam gericht is in plaats van haaks daarop, is de polarisatierichting van de radar potentieel inzetbaar in het systeem.

Uit de metingen naar de polarisatie van de mens en de bal blijkt echter dat doordat de bundel van de radar kleiner is dan een mens in plaats van andersom het verschil in reflecties tegenvalt. Ook is de bal die in het prototype gebruikt wordt niet egaal genoeg om onafhankelijk van de polarisatierichting van de radar te zijn. Deze bevindingen maken het gebruik van polarisatie ongeschikt om toegepast te worden in het prototype. Met een bredere radarbundel zou het wel toepasbaar zijn.

De metingen naar de mate van constant zijn van de reflecties van mensen en ballen heeft uitgewezen dat de reflectie van een bal tien maal constanter is dan die van een mens. Dit geldt zowel voor bewegende als voor stilstaande mensen en ballen. Wel is het zo dat de reflecties van bewegende objecten weer tien keer minder constant zijn dan die van stilstaande objecten. Toch is het verschil zo groot dat hier gebruik van gemaakt kan worden in het prototype.

Deel III: Implementatie

6 Implementatie

Aan de hand van het programma van eisen en de deelconcepten is er een prototype ontworpen. Het prototype is vooral bedoeld om de verschillende oplossingen voor de drempelbepaling van de piekdetectie en de twee alternatieven voor het trackinggedeelte te kunnen testen. Daarnaast moet het prototype bewijzen dat het concept van het detecteren van een voetbal met radar en het aansturen van een beeldverwerkingsysteem, uitvoerbaar is. Een van de voornaamste eisen aan het prototype is dat de rekenkundige complexiteit van de implementatie niet al te groot is en de ontwikkeltijd kort. Dit is nodig omdat de onderzochte methodes in een demonstratie op kleine schaal real-time getoond moeten kunnen worden. De meest beperkende factor daarin is de snelheid van de hardware waarop de software draait. Dit hoofdstuk beschrijft de ontwikkeling van het prototype en de gemaakte keuzes.

Het ontwikkelde prototype maakt gebruik van slechts één radar. Het uiteindelijke systeem zal of gebruik maken van meerdere radars of van een scannende radar. Hierdoor is de software die voor het prototype ontwikkeld is te eenvoudig om direct verbeterd te worden tot een inzetbaar systeem. Voor een op volledige schaal inzetbaar systeem moet het beeld van meerdere radars gecombineerd worden of de hoekverandering van de scannende radar geïnterpreteerd worden. In het prototype is aan dergelijke zaken geen aandacht geschonken.

Uit test- en ontwikkeloogpunt is een bestaande Graphical User Interface (GUI) uitgebreid en aangepast om de resultaten weer te geven. Voor het prototype is niet de uiteindelijke gebruikers GUI geschreven maar een interface die de voor het ontwikkelen nodige informatie weergeeft.

Dit hoofdstuk is als volgt ingedeeld. In paragraaf 6.1 wordt besproken voor welke programmeertaal er is gekozen bij het ontwikkelen van het prototype. In paragraaf 6.2 wordt uitgelegd voor welk alternatief is gekozen om de voetbal zo goed mogelijk zichtbaar te maken voor de radar. In paragraaf 6.3 wordt de objectdetectie behandeld. Vervolgens wordt in paragraaf 6.4 de objecttracking behandeld en in paragraaf 6.5 de objectclassificatie. In paragraaf 6.7 wordt de beslissing of het beeldverwerkingsysteem ingeschakeld moet worden behandeld. Dan wordt in paragraaf 6.8 de communicatie tussen MaxSoft en MaxImage besproken. Tenslotte komt in paragraaf 6.9 de GUI aan bod.

6.1 Programmeertaal

Het prototype is volledig in Matlab ontwikkeld. Er is voor Matlab gekozen omdat de standaard toolboxen en functies het mogelijk maken om snel code te ontwikkelen. Ook kon er gebruik gemaakt worden van bestaande Matlab code van Peter de Vrijer [5] voor het eerste deel van de dataverwerking, te weten het triggeren van het beatsignaal en de

FFT. Daarnaast heeft de GUI van Peter de Vrijer model gestaan voor de GUI van het prototype zoals in paragraaf 6.9 beschreven staat. De mogelijkheden van Matlab om grafische interfaces te schrijven heeft ook meegewogen in de keuze. Omdat de beschikbare ontwikkeltijd vrij kort was, is gekozen voor de programmeeromgeving waarbinnen de ontwikkeling het snelst zou verlopen.

6.2 Bal zichtbaar maken

Een bal op zich is voor een radar niet erg goed zichtbaar. Om in het prototype toch een duidelijke reflectie van de bal te krijgen is deze bekleed met aluminium. Het laagje aluminium op de bal geleidt veel beter dan de bal zelf en reflecteert daardoor beter. Het laagje aluminium is aangebracht in de vorm van aluminiumtape. Dit is een goedkope en makkelijke manier om de bal beter te laten geleiden. In het prototype zijn geen RFID-tags gebruikt omdat het niet mogelijk was om op korte termijn aan RFID-tags te komen in de frequentieband van de radar die voor het prototype gebruikt wordt. Als de chips wel voor handen zouden zijn was het waarschijnlijk te gecompliceerd om binnen de ontwikkeltijd van het systeem een werkend prototype te bouwen.

6.3 Objectdetectie

Voor objectdetectie maakt het prototype gebruik van piekdetectie. Zoals beschreven in hoofdstuk 4 is piekdetectie de meest eenvoudige methode om objecten te herkennen in de radarreflecties. Het detectiealgoritme wordt toegepast op de Fouriertransformatie van het beatsignaal. Het spectrum van het beatsignaal heeft pieken op de punten waar echo's van objecten zitten. De plaats van de pieken in het spectrum kunnen omgerekend worden naar afstanden tussen de objecten en de radar.

De piekdetectie bepaalt over het gehele spectrum alle lokale maxima. Aangezien de bewerking van het signaal in een computer plaats vindt is het spectrum beschikbaar als een vector van getallen. Het bepalen van de posities van de lokale maxima komt dan neer op het lokaliseren van die elementen in de vector die groter zijn dan zowel het vorige als het volgende element. Deze elementen worden in een nieuwe vector gezet. Dit zijn de waardes die bij de maxima horen. Ook de positie index in de oude vector van de elementen die in de nieuwe vector opgenomen zijn, worden als vector opgeslagen. Dit is nodig omdat juist deze vector de afstanden van de objecten tot de radar bevat.

De twee aangemaakte vectoren bevatten nu de locatie en de hoogte van de maxima. De gevonden lokale maxima bevatten echter ook veel pieken die het gevolg van ruis zijn. Het is daarom nodig om een drempel toe te passen zodat deze pieken niet verder verwerkt worden. Hiertoe wordt voor ieder element in de vector die de waardes van de lokale maxima bevat gekeken of de waarde boven een in te stellen grens uitkomt. Als dat niet het geval is, wordt het element uit zowel de vector met waardes als die met indices gehaald.

Om de hoogte van de drempel te bepalen zijn twee verschillende methodes getest, een vaste drempel en een adaptieve. Binnen de adaptieve drempel zijn ook weer twee mogelijkheden, een OS-CFAR en een CML-CFAR detector. De waarde van de vaste drempel en de parameters van de CFAR detectors zijn gekozen zoals bepaald in paragraaf 5.2.

6.4 Objecttracking

In het prototype zijn twee methoden geïmplementeerd om objecten te kunnen volgen. In deze paragraaf wordt de implementatie van de methoden besproken. Beide alternatieven zijn voorgesteld in paragraaf 4.4. Het dichtstbijzijnde nabuuralgoritme is een rekenkundig eenvoudige methode om objecten te volgen, de implementatie van dit algoritme wordt in paragraaf 6.4.1 besproken. De tweede methode is rekenkundig complexer dan het dichtstbijzijnde nabuuralgoritme en maakt gebruik van een Kalman filter. De implementatie van het Kalman filter wordt beschreven in paragraaf 6.4.2.

6.4.1 Objecttracking met behulp van het dichtstbijzijnde nabuuralgoritme

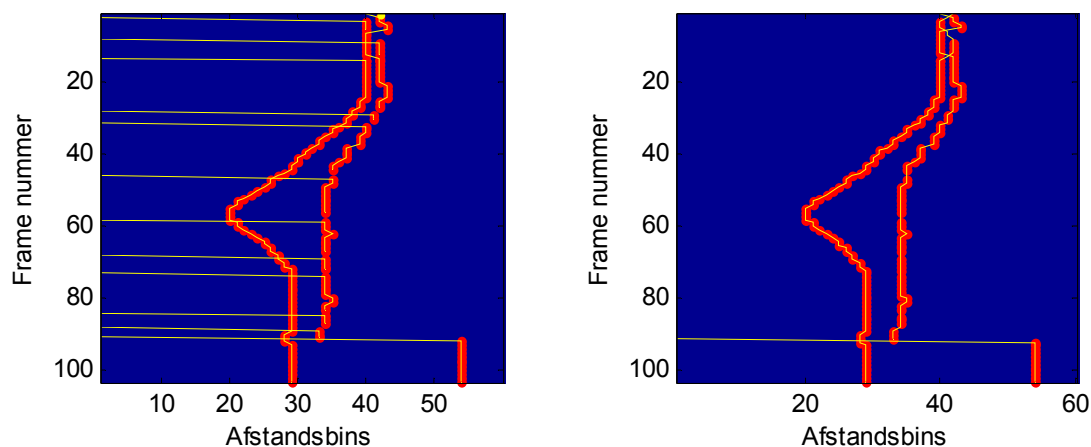
Het dichtstbijzijnde nabuuralgoritme [10] maakt gebruik van de vorige meting om de nieuwe meting daaraan te relateren. Het is dus nodig om tenminste één meting te bewaren. Om een baan te bepalen uit meerdere metingen moet een buffer bijgehouden worden. De buffergrootte kan vrij gekozen worden en is in het prototype op 5 frames gekozen. In deze buffer worden van iedere meting de locaties van de pieken opgeslagen

Het algoritme bestaat uit een initialisatie en zes stappen die bij elke toevoeging van een nieuw frame worden uitgevoerd.¹ De initialisatie wordt alleen uitgevoerd wanneer het systeem wordt ingeschakeld. In de initialisatie wordt een nieuwe matrix aangemaakt. Op de eerste rij van deze nieuwe matrix wordt vervolgens de eerste rij van de matrix met pieken gekopieerd. De matrix waarin de banen worden bijgehouden noemen we de ID-matrix. Zolang de ID-matrix nog niet op volledige lengte is, wordt bij iedere nieuwe meting een rij aan de matrix toegevoegd. Zodra de ID-matrix op lengte is, wordt aan het begin van iedere frame de eerste rij van de matrix weggehaald en een nieuwe rij met nullen onder aan de matrix toegevoegd. Op deze manier zijn telkens de laatste 5 posities van de banen opgeslagen in de ID-matrix. De eerste stap in het algoritme is het bepalen van de nieuwe pieken. Hiertoe wordt de piekdetectie aangeroepen. Vervolgens wordt voor iedere piek de dichtstbijzijnde piek uit het vorige frame gezocht. Als de piek zich binnen een ingestelde afstand van de vorige piek bevindt, wordt die aan de kolom toegevoegd. Doordat voor elke piek alleen de kortste afstand naar een piek uit de vorige frame wordt bepaald kan het voorkomen dat een piek gematcht wordt met een piek uit een vorig frame die dichterbij een andere piek ligt. Dit kan onwenselijke toewijzingen opleveren maar scheelt wel rekenwerk. Om te voorkomen dat niet één nieuwe piek aan twee oude gekoppeld wordt, of twee oude aan één nieuwe, wordt in een vector bijgehouden welke oude en nieuwe pieken al toegewezen zijn in deze frame. Als alle nieuwe pieken doorlopen zijn worden de pieken die geplaatst konden worden uit de vector met nieuwe pieken gehaald.

Als er na de eerste stap nog ongeplaatste pieken over zijn, wordt in de tweede stap gekeken of er afgebroken banen gelijmd moeten worden. Om te weten of er gelijmd moet worden wordt gekeken of er elementen met de waarde 'inf' in de vorige rij zitten. Er wordt een oneindige waarde toegekend aan afgebroken reeksen zoals hieronder beschreven staat. Als dit zo is, wordt de laatst bekende waarde uit die kolom opgehaald

¹ Zie appendix A voor een pseudocode van dit algoritme.

en gekeken of deze binnen een bepaalde grens van de overgebleven pieken ligt. Wanneer dat het geval is, wordt de piek in deze kolom geplaatst. Omdat het algoritme in staat is om banen te lijmen wordt voorkomen dat er veel korte banen gevormd worden. Dit is te zien in figuur 6.1. In deze figuur worden links de gevonden banen weergegeven zonder dat er gelijmd wordt, en rechts wanneer er wel gelijmd wordt. In de figuur is twee keer dezelfde dataset te zien. De twee rode banen zijn de opeenvolgende gedetecteerde pieken van een bal en een mens. De linker baan is die van de bal, de rechter die van de mens. De frames lopen van boven naar beneden op. Een lijn van links naar rechts geeft aan dat er een nieuwe baan begonnen is. In de figuur is te zien dat een situatie waarin maar drie gelijmde banen nodig zijn om alle gevonden pieken te groeperen, kan bestaan uit twaalf banen als er niet gelijmd zou worden. De aanwezigheid van veel banen kan er voor zorgen dat het dataverwerkingsysteem van de radar overbelast raakt [22].



Figuur 6.1 Links de ongelijmde baan, rechts het resultaat na lijmen.

Als er een piek in een afgebroken reeks geplaatst is, moeten de waarden tussen de nieuw geplaatste piek en de laatst bekende piek geïnterpoleerd worden. Dit wordt gedaan door van de laatst geplaatste waarde de laatst bekende waarde af te trekken en dit resultaat door het aantal missende waarden plus één te delen. Vervolgens worden de ontbrekende waarden lineair met dit getal opgehoogd.

Het is mogelijk dat er na deze twee stappen nog steeds ongeplaatste pieken over zijn. Als dit het geval is, wordt in de vierde stap een kolom aan de ID-matrix toegevoegd. Deze kolom bestaat uit nullen met uitzondering van de laatste waarde, deze bestaat uit de nieuwe piek. Zo wordt er een nieuwe baan toegevoegd aan de ID-matrix.

Na deze drie stappen worden er nog twee bewerkingen op de ID-matrix gedaan. Ten eerste wordt er gekeken of er elementen met de waarde nul in de laatste rij zitten. Deze elementen zitten in een baan waaraan geen nieuwe piek toegevoegd is, dit zijn gebroken reeksen. De nullen worden nu vervangen voor de waarde 'inf', hierdoor zal er nooit een piek zomaar aan gematcht worden behalve in de lijm procedure zoals hierboven beschreven. De tweede bewerking kijkt of er kolommen in de ID-matrix zitten die volledig uit elementen met de waarde 'inf' bestaan. Deze kolommen bevatten banen die niet meer te lijmen zijn en dus definitief afgebroken. Aangezien ze geen informatie meer bevatten worden deze kolommen uit de matrix verwijderd.

6.4.2 Objecttracking met behulp van het Kalman filter

Het Kalman objecttracking algoritme¹ start met een initialisatie waarin een vector wordt aangemaakt waarin later de verschillende tracks zullen worden opgeslagen. Wanneer er nog geen tracks bestaan, maakt het algoritme voor ieder gevonden object een nieuwe track aan. Bij het aanmaken van een nieuwe track wordt gekozen voor het gebruik van één van de drie modellen voorgesteld in (4.11), (4.21) of (4.26). Aan de nieuwe track worden verschillende variabelen toegekend, zoals een vector waarin de posities van de baan kunnen worden opgeslagen en verschillende variabelen die nodig zijn voor het Kalman filter.

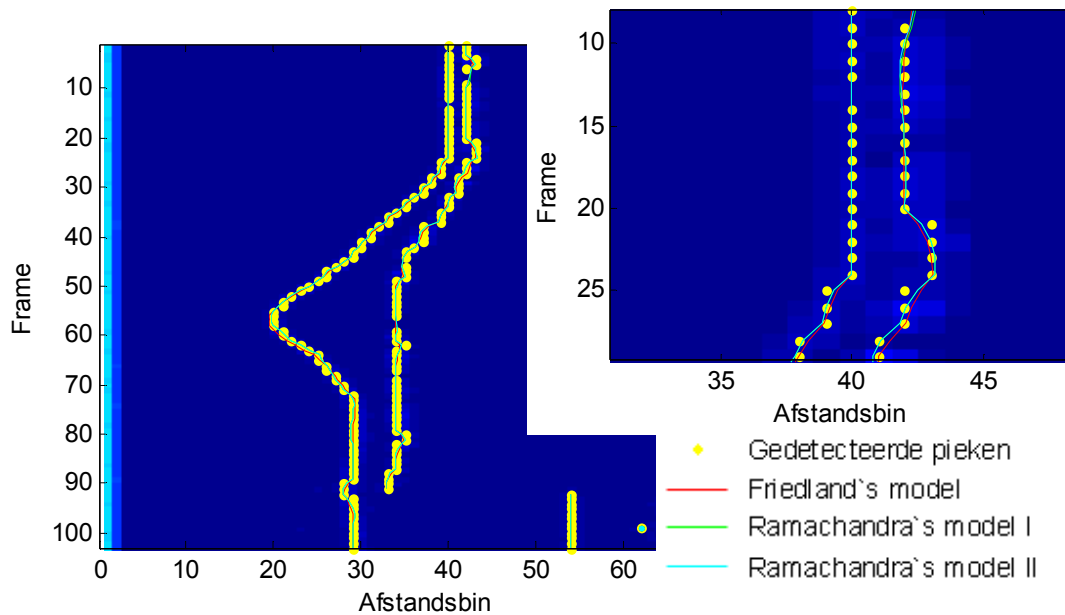
Elke keer voordat een nieuw frame wordt toegevoegd, wordt voor elke al bekende track op basis van één van de modellen een voorspelling gemaakt van de nieuwe positie x_{n+1} in de baan. De voorspelde nieuwe posities x_{n+1} in de tracks worden dan vergeleken met de gevonden pieken in het nieuwe frame. Wanneer een gevonden piek past bij de voorspelling van de nieuwe positie in een track wordt de piek in de track verwerkt met behulp van het Kalman filter zoals beschreven in paragraaf 4.4.2. Wanneer een gevonden piek bij geen enkele voorspelde positie van een track past, wordt een nieuwe track aangemaakt.

Zoals beschreven in paragraaf 6.4.1 zullen er ook nu frames zijn waarin pieken zijn weggefallen. Het algoritme anticipeert hierop door de geschatte volgende positie in een track toe te voegen aan de track. Het aantal keer dat een schatting wordt toegevoegd aan een track wanneer er geen passende piek gevonden is, is beperkt tot k . Het kan immers zo zijn dat het object inderdaad verdwenen is en de piek dus terecht weggefallen is. Tot slot wordt er een ID-matrix aangemaakt.

De resultaten van het Kalman filter zijn per model weergegeven in figuur 6.2. Er is een duidelijk verschil te zien tussen (4.11) met twee toestanden en (4.21) en (4.26) met beiden drie toestanden. Model (4.11) gaat uit van een gemiddelde acceleratie gelijk aan nul, wat zich in de linkerfiguur vertaalt in de rode baan die zich rond frame 60 slechts traag met de bal meebeweegt. De banen gevormd met behulp van model (4.21) en (4.26) passen zich veel sneller aan. Dit is ook te zien in de rechter afbeelding, waarin een gedeelte van de linkerfiguur is uitvergroot.

In figuur 6.2 zijn de gedetecteerde pieken aangegeven door middel van een gele punt. In frame 14 is er een piek weggefallen. Hier is de verwachte baan van het object bepaald op basis van de vorige positie. De modellen (4.21) en (4.26) presteren hierin zoals verwacht beter dan model (4.21). Het verschil in resultaten van model (4.21) en (4.26) bleken echter in het prototype weinig verschil te maken. Daarom is gekozen om in het prototype gebruik te maken van Ramachandra's model I (4.21), dat minder reken capaciteit vraagt dan (4.26).

¹ Zie appendix A voor een pseudocode van dit algoritme.



Figuur 6.2 De tracks gevormd met behulp van het Kalman filter, gebruikmakend van de drie dynamische modellen. In frames waar geen piek werd gedetecteerd, is de positie van de piek voorspeld op basis van de vorige metingen.

6.5 Objectclassificatie

De objectclassificatie die in het prototype toegepast wordt, is gebaseerd op het statistische gedrag van de reflecties van de objecten. Zoals beschreven in hoofdstuk 4 is de radardoorsnede van een bal veel constanter dan die van een mens. Het wel of niet constant zijn van de radardoorsnede vertaalt zich in het wel of niet constant zijn van de amplitude van de reflectie. Het model dat voor de ontwikkeling van dit algoritme gebruikt is, gaat ervan uit dat de reflectie van een object een Gaussische random variabele is. De kansdichtheidsfunctie van een bal heeft een veel kleinere standaarddeviatie dan die van een mens. Het algoritme bepaalt de standaarddeviatie van de hoogte van de pieken van de gevonden objecten en beslist op grond van die informatie of een object een bal is of niet. De ID-matrix die door het trackingalgoritme gemaakt is, bevat in iedere kolom de baan van een object. Het algoritme dat objecten classificeert kan opgedeeld worden in drie delen. Ten eerste worden de pieken uit de frames gehaald. Ten tweede wordt de variatie van de pieken berekend en geschaald. Ten derde wordt besloten of de track van een bal is of niet.

Het eerste deel van het werk is het uitsnijden van de pieken uit de data. Hiertoe wordt voor ieder track in de ID-matrix de baan afgelopen om te kijken of het geen afgebroken baan is. Afgebroken banen bevatten elementen waar geen geldige verwijzing naar een element in de databuffer in staat. Deze verwijzingen leveren fouten in de uitvoer op. De baan wordt in een tijdelijke vector gekopieerd. Vervolgens worden alle waarden die bij de gekopieerde elementen in de baan horen uit de framebuffer gehaald en in een nieuwe vector gezet. Alle amplitudes van de pieken zijn nu uit de data gehaald. Door de

kans op afgebroken banen hoeven de vectoren voor de verschillende pieken niet even lang te zijn, hierdoor moeten ze afzonderlijk behandeld worden.

Van iedere afzonderlijke vector van een track wordt de standaarddeviatie bepaald. De uitkomst van de berekende standaarddeviatie wordt geschaald door het te delen door de gemiddelde waarde van de reflecties over de periode waar de standaarddeviatie van genomen is.

Ten slotte wordt besloten of tussen de gevonden objecten de bal aanwezig is. De beslissing komt erop neer dat als de geschaalde standaarddeviatie onder een gestelde grens komt het object een bal is. Wanneer er meerder objecten aan de eis voldoen wordt het dichtstbijzijnde object als bal gekenmerkt. De keuze om het dichtstbijzijnde object wat voldoet aan de criteria van een bal tot bal te bestempelen, is gebaseerd op het feit dat het dichtstbijzijnde object het gevaarlijkst is.

Wanneer een object als bal geclassificeerd is, heeft het object een grotere kans dat het object bij de volgende beslissing weer als bal geclassificeerd wordt. Hier is voor gekozen om te voorkomen dat wanneer een bal tijdelijk minder op een bal lijkt meteen een andere track als bij de bal horend gekozen wordt. Het systeem past twee methodes toe om hier voor te zorgen. Ten eerste wordt, zodra een ander object dan het vorige als bal geclassificeerd wordt, het vorige object geforceerd als bal aangemerkt wordt. Het aantal keer dat dit gebeurt is in te stellen en staat in het prototype op drie. Ten tweede is de grens waaronder de geschaalde standaarddeviatie moet liggen voor het object dat de vorige keer als bal werd geclassificeerd een factor hoger dan voor de andere objecten. Hierdoor is de kans groter dat het object weer als bal gezien wordt. Hoewel deze systemen er voor zorgen dat een gevonden bal niet snel kwijt raakt, hebben ze ook een nadelig effect. Het duurt namelijk langer voordat een object dat foutief als bal gezien werd, vervangen wordt door de echte bal. Bij een hoge frame-rate is dit probleem niet erg groot, het duurt immers een aantal frames voordat de foute beslissing recht gezet wordt, maar bij een lage frame-rate, zoals in het prototype het geval is, is er een merkbare vertraging na een foute beslissing.

6.6 Balsnelheid

Als de bal gevonden is, is het voor de besluitvorming van belang om te weten wat de snelheid en richting van de bal is. Omdat alleen afstandsgegevens beschikbaar is kan alleen een snelheidscomponent in de richting van de radar bepaald worden. Of de bal naar de radar toe beweegt, of er vanaf, wordt aangegeven in het teken van de snelheid. Als de snelheid negatief is, beweegt de bal van de radar af. Als de snelheid positief is beweegt de bal juist naar de radar toe. Om de snelheid te bepalen wordt eerst de gemiddelde afstand tussen de pieken in de baan van de bal bepaald. Het resultaat hiervan is een snelheid in afstandsbins per frame. Vervolgens wordt dit resultaat vermenigvuldigd met de grootte van een afstandsbin om tot een snelheid in meter per frame te komen. Tenslotte wordt dit resultaat weer vermenigvuldigd met de frame-rate. Het eindresultaat is dan een snelheid in meter per seconde waarin het teken de richting ten opzichte van de radar aangeeft.

6.7 Besluitvorming voor de inschakeling van beeldverwerkingsysteem

Nu het systeem objecten kan detecteren, volgen en classificeren moet er iets gedaan worden met de informatie die verzameld is. Het uiteindelijke doel van het systeem is immers het aansturen van het beeldverwerkingsysteem. Om te bepalen of er een kans op een doelpunt bestaat zijn er twee gebieden ingesteld. Het eerste gebied is het gebied dicht bij het doel. Wanneer de bal zich in dit gebied bevindt, wordt het beeldverwerkingsysteem altijd ingeschakeld. In de uiteindelijke versie zal dit een gebied van ongeveer vijf meter van het doel zijn. Als de bal gevonden wordt binnen dit gebied wordt het beeldverwerkingsysteem aangezet. Dit gebied is op vijf meter van het doel gekozen. Als de bal zich zo dicht bij het doel bevindt, is de kans dat er een doelpunt valt groot. Daarnaast is het zo dat als van deze korte afstand hard op doel geschoten wordt de kans bestaat dat het beeldverwerkingsysteem niet snel genoeg opgestart is om een eventueel doelpunt te detecteren. Om te voorkomen dat er een doelpunt gemist wordt moet het systeem dus ingeschakeld zijn als de bal dichtbij is.

Het tweede gebied is alles buiten de eerste vijf meter. De beslissing om het systeem aan te zetten in dit gebied is gebaseerd op zowel de baan als de snelheid van de bal. In de baan kunnen zich drie situaties voordoen. Of de bal ligt stil, of de bal beweegt zich van het doel af, of de bal beweegt naar het doel toe. In de eerste twee gevallen wordt het systeem vanzelfsprekend niet aangezet. In het laatste geval moet de situatie beter bekeken worden. Het besluit hangt in dit geval af van waar de bal zich bevindt en met welke snelheid de bal naar het doel komt. Om te beslissen of het beeldverwerkingsysteem ingeschakeld moet worden wordt er een schuivende grens bepaald waar de bal voorbij moet komen om het beeldverwerkingsysteem in te schakelen. De afstand vanaf de radar waarop de grens ligt, is afhankelijk van de snelheid van de bal:

$$x_{grens} = x_{gebied} + v_{voetbal} t_{vertraging} \quad (5.3)$$

waarin x_{grens} = de afstand in meters waarop de grens ligt

x_{gebied} = de afstand in meters waarbinnen het beeldverwerkingsysteem altijd ingeschakeld moet worden

$v_{voetbal}$ = de snelheid in meter per seconde van de bal

$t_{vertraging}$ = de inschakelvertraging in seconden van MaxSoft

Een bal die snel op het doel afkomt, zet dus op een grotere afstand het beeldverwerkingsysteem aan dan een bal die langzaam op het doel afkomt. Op deze manier heeft het beeldverwerkingsysteem ongeacht de snelheid van de bal telkens even veel tijd om op te starten, tenzij de bal langzaam richting het doel gebracht wordt en plotseling hard op doel geschoten.

Om te voorkomen dat het beeldverwerkingsysteem te vaak aan en uit gezet wordt in het geval van een bal die een paar keer afwisselend naar het doel en van het doel af beweegt is er een vertraging ingebouwd in het uitschakelen van het beeldverwerkingsysteem toe. Binnen de vertraging kan het signaal niet veranderd worden.

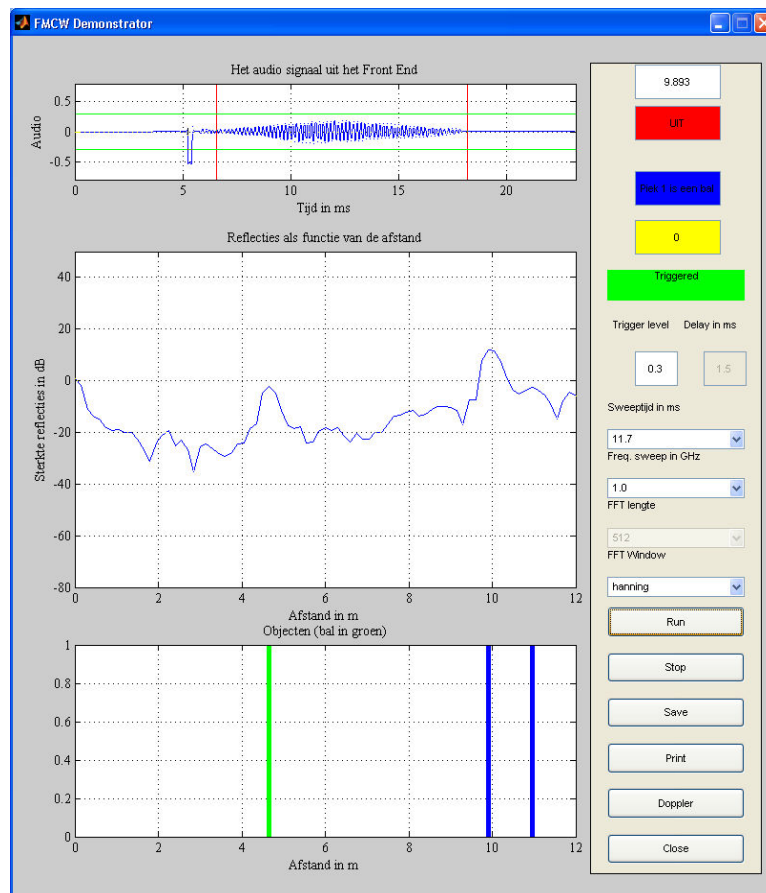
6.8 Communicatie

De software moet, wanneer het besloten heeft dat er een bal het doel nadert, dit communiceren met MaxImage. Omdat het prototype op een gewone PC draait onder Matlab wordt er gebruik gemaakt van de parallelle poort om te communiceren met de buitenwereld. Het aan of uit signaal wordt op een pin van de parallelle poort gezet. Een nul betekent dat het systeem uit moet staan, een één dat het systeem aan moet staan.

6.9 Graphical User Interface (GUI)

Om tijdens de ontwikkeling van het prototype en tijdens het testen ervan relevante data op het scherm te kunnen laten zien is er een GUI geschreven. De GUI is een uitbreiding op de GUI van Peter de Vrijer. In figuur 6.3 is een screenshot van de GUI weergegeven. De uitbreiding op de GUI van Peter de Vrijer bestaat onder andere uit de extra figuur onder de weergave van de reflecties. In deze figuur wordt met blauwe lijnen aangegeven waar objecten gevonden zijn. Als de software een van de objecten als bal geassocieerd heeft, is dat aangegeven met een groene lijn.

Daarnaast wordt in de GUI aangegeven wat de frame-rate en de snelheid van de bal zijn. Er is ook een vlakje dat aangeeft of het beeldverwerkingsysteem in- of uitgeschakeld is, en of de dichtstbijzijnde piek een mens of een bal is.



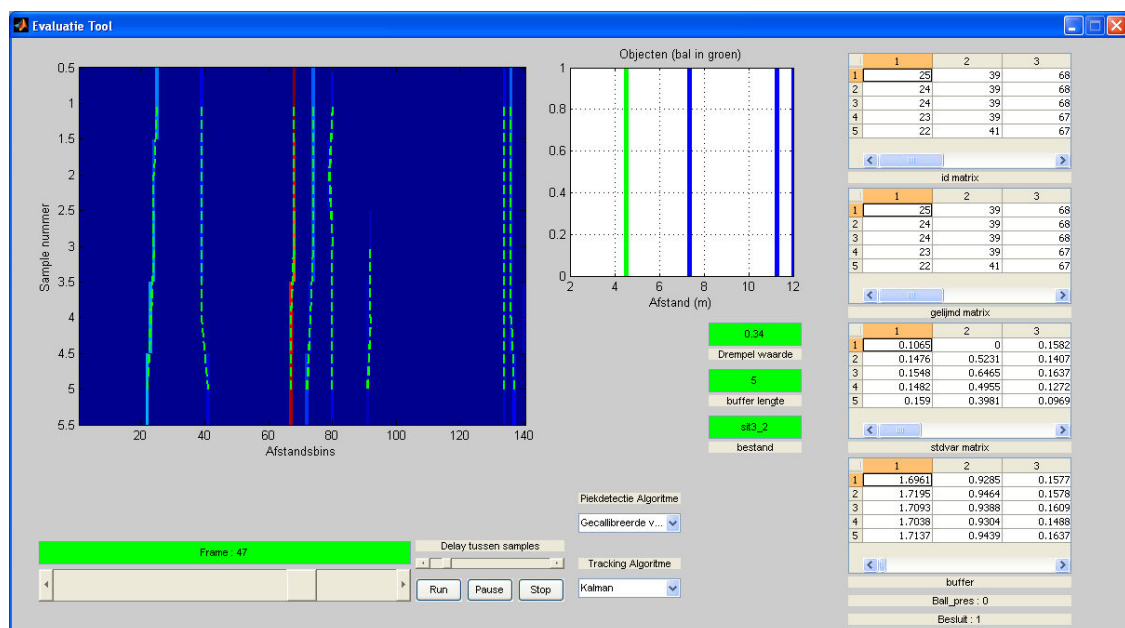
Figuur 6.3 Screenshot van de GUI.

7 Evaluatie van het prototype

Op het ontwikkelde prototype is per deelsysteem een uitgebreide evaluatie uitgevoerd. Het doel van deze evaluatie is enerzijds om te bepalen welke combinatie van deelsystemen het beste presteert, anderzijds om het prototype te kunnen toetsen aan het programma van eisen in hoofdstuk 3. Om de evaluatie te kunnen uitvoeren zijn er evaluatietools ontwikkeld, spelsituaties gesimuleerd en spelsituaties geanalyseerd. Dit hoofdstuk beschrijft de manier waarop de evaluatie is uitgevoerd. Onderdelen die daarbij aan bod komen zijn de ontwikkeling van een Evaluatie Tool in paragraaf 7.1, het beschrijven van de nagebootste spelsituaties in paragraaf 7.2 en de evaluatieresultaten van de verschillende deelsystemen in paragraaf 7.3.

7.1 Evaluatie Tool

Om het prototype te kunnen evalueren is een evaluatieomgeving ontwikkeld. Deze evaluatieomgeving moest voldoen aan de volgende eisen. In de evaluatieomgeving moet het mogelijk zijn de berekeningen die real-time gedaan zijn frame per frame te herhalen. Het moet mogelijk zijn om voor- en achteruit door de resultaten te bladeren. Zowel het detectiegedeelte als het trackinggedeelte moeten gevisualiseerd worden. Ook moet het mogelijk zijn om alle data die gegenereerd wordt bij het verwerken van de informatie van de radar te bekijken. Omdat er voor het prototype twee verschillende trackingmethodes ontwikkeld zijn, moet het mogelijk zijn om tussen deze twee methodes te schakelen. Tenslotte moet het mogelijk zijn om bepaalde variabelen in de software te veranderen om de gevolgen daarvan te onderzoeken.



Figuur 7.1 Screenshot van de Evaluatie Tool.

De ontworpen evaluatieomgeving kan in drie delen worden opgedeeld. Ten eerste is er het dataloggedeelte, ten tweede het simulatiegedeelte en ten derde het visualisatiegedeelte. Deze drie onderdelen komen respectievelijk in de paragrafen 7.1.1, 7.1.2 en 7.1.3 aan bod.

7.1.1 Dataloggedeelte

Om de meting nadat die gedaan is te kunnen simuleren en visualiseren moet de data die tijdens het draaien van het prototype beschikbaar is ook voor het evaluatiesysteem beschikbaar zijn. Om dit voor elkaar te krijgen wordt tijdens het draaien van het systeem aan het eind van ieder verwerkt frame de resultaten van de FFT opgeslagen in een matrix. Na het beëindigen van de meting wordt deze matrix op de harde schijf opgeslagen. Voor deze werkwijze is gekozen omdat het weinig tijd kost om de data in het geheugen op te slaan en later weg te schrijven. Het is daarnaast van belang dat alleen de informatie die het systeem tijdens het draaien kon verwerken ook in de evaluatie gebruikt wordt. Omdat de data real-time weggeschreven wordt worden er net zoveel frames per seconde gebruikt in de evaluatie als in het echte systeem.

7.1.2 Simulatiegedeelte

Het simulatiegedeelte is in principe een exacte kopie van het prototype. De enige verschillen zijn dat de data die verwerkt wordt bestaat uit opgeslagen resultaten van de FFT routine en dat bijna alle uitvoer en tussen resultaten opgeslagen wordt. Wanneer de data ingelezen wordt, wordt er een lopende buffer aangemaakt net als in het systeem gebeurt. Bij ieder volgend frame worden de berekeningen gedaan die het prototype normaal ook doet uitgevoerd. Alle tussenresultaten worden nu bewaard, zoals bijvoorbeeld de ID-matrix en de gelijkde matrix. Ook wordt de buffer met geschaalde standaarddeviaties opgeslagen, evenals de positie van de bal en hoe vaak het classificatiegedeelte een object dat als bal geclassificeerd is geforceerd nogmaals tot bal heeft gemaakt. Tenslotte wordt een afbeelding gemaakt met de gedetecteerde pieken.

Het is nodig om de simulatie over de volledige data te laten lopen omdat het prototype gebruik maakt van frames uit het verleden. Als niet de volledige simulatie gedaan zou worden zou het niet mogelijk zijn om naar bepaalde punten in de meting te gaan of de meting in omgekeerde volgorde te doorlopen.

De simulatie biedt ook de mogelijkheid om te kiezen tussen de twee verschillende methodes voor het volgen van objecten en de drie alternatieven voor het bepalen van de drempel in de piekdetectie. Hierdoor is het mogelijk om deze methodes met elkaar te vergelijken op precies dezelfde data en op precies dezelfde momenten. Het is bijvoorbeeld mogelijk om de toestand van het systeem te bekijken op een bepaald frame en die toestand vervolgens te vergelijken met hoe het systeem zou reageren met een andere methode. Ook is het mogelijk om de simulatie uit te voeren met andere parameters voor bijvoorbeeld de drempel van de piekdetectie.

7.1.3 Visualisatiegedeelte

Om het evaluatiesysteem makkelijk te kunnen bedienen en om de data te visualiseren is een GUI ontworpen, waarvan een screenshot is weergegeven in figuur 7.1. De GUI heeft een bedieningsgedeelte en een visualisatiegedeelte. In het bedieningsgedeelte is het mogelijk de data frame voor frame te bekijken door een schuifbalk te verschuiven naar links of naar rechts. Daarnaast is het mogelijk om de hele dataset als een film te bekijken

door op de ‘run’ knop te klikken. Deze film is te pauzeren of te stoppen door op respectievelijk ‘pause’ of ‘stop’ te klikken. De snelheid waarmee de film speelt, is te veranderen door een tweede schuifbalk naar links of rechts te bewegen. Het bedieningsgedeelte bevat ook velden om de drempelwaarde en de bufferlengte te veranderen, alsmede dropdown menu’s om de drempel- en trackingmethode te kiezen. Zodra de gebruiker één van deze waardes verandert, wordt het simulatiegedeelte opnieuw uitgevoerd en wordt alle informatie op het scherm ververs met de nieuwe resultaten. Na het verversen wordt hetzelfde frame als voor het verversen getoond. Dit maakt vergelijken erg gemakkelijk. Het is daardoor mogelijk om in een bepaalde situatie de drempel hoger of lager te kiezen en meteen de gevolgen daarvan te bekijken. Op dezelfde manier kan gekozen worden tussen de twee verschillende trackingmethode.

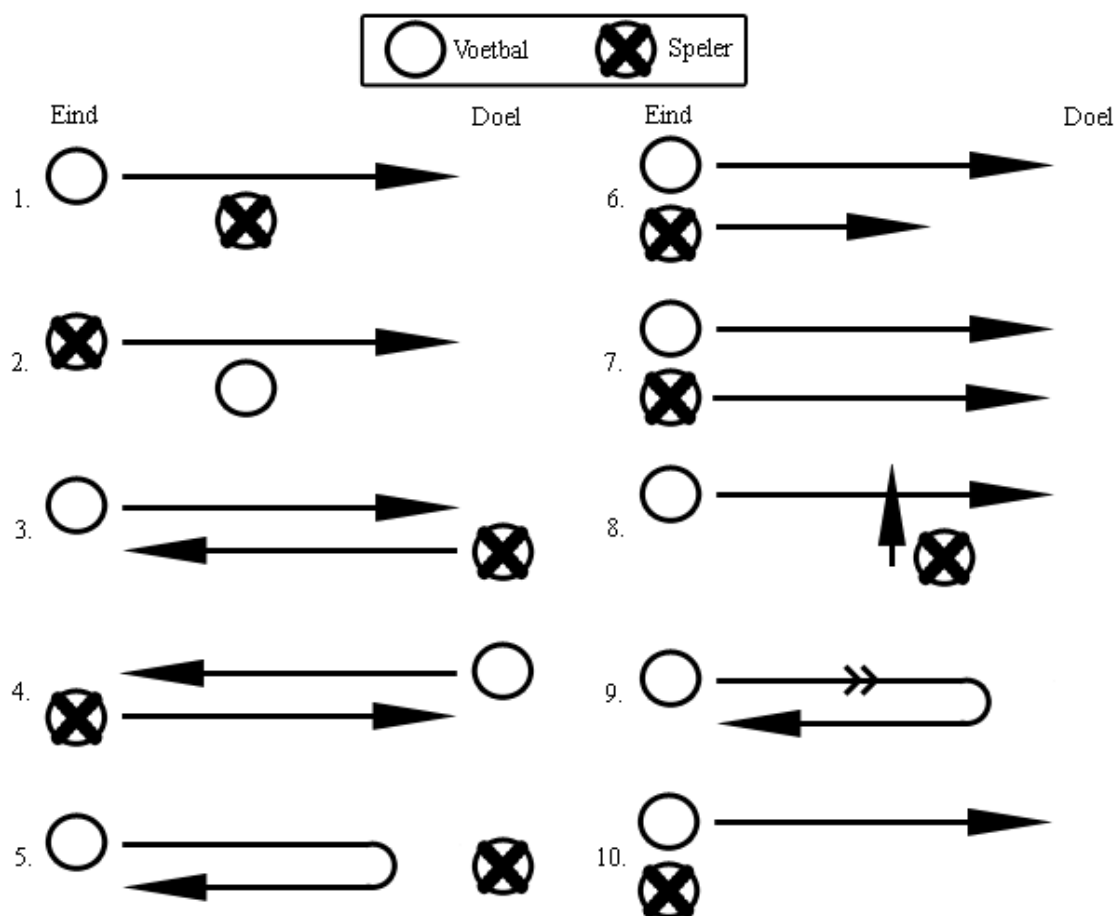
De informatie die op het scherm getoond wordt, bestaat uit een afbeelding van de gevonden pieken met daaroverheen een plot van het trackingmethode. In een tweede figuur worden alle gevonden objecten weergegeven met daarbij de afstand tot de radar. Als er een bal geclassificeerd is wordt deze in groen weergegeven. In vier tabellen worden respectievelijk de ID-matrix, de gelijmde matrix, de standaarddeviatiebuffer en de framebuffer getoond. Door deze informatie in tabellen weer te geven is het mogelijk om na te gaan of er in bepaalde situaties fouten optreden, en waarom. Zo kan gekeken worden of fouten het resultaat zijn van programmeerfouten of tekortkomingen in het ontwerp van het algoritme. Tenslotte is er nog een veld waar weergegeven wordt of de bal geforceerd toegewezen is of niet.

7.2 Spelsituaties waarop de evaluatie gebaseerd is

De evaluatie van de verschillende deelsystemen en het totale systeem is gebaseerd op tien spelsituaties die zich tijdens een voetbalwedstrijd kunnen voordoen. Van elke spelsituatie zijn er vijf datasets gemaakt met de opstelling die in paragraaf 5.1 beschreven is. In elke situatie, behalve in situatie 9, bevinden zich een bal en een speler in de radarbundel. De situaties spelen zich af tussen 3,45 (begin) en 6,45 (einde) meter van de radar. De verschillende spelsituaties zijn geïllustreerd in figuur 7.2 en worden als volgt beschreven:

1. Bal beweegt zich van het einde van de baan naar het begin van de baan.
Speler bevindt zich halverwege de baan.
2. Bal bevindt zich halverwege de baan.
Speler beweegt zich van het einde van de baan naar het begin van de baan.
3. Bal beweegt zich van het einde van de baan naar het begin van de baan.
Speler beweegt zich tegelijkertijd van het begin van de baan naar het einde van de baan.
4. Speler beweegt zich van het einde van de baan naar het begin van de baan.
Bal beweegt zich tegelijkertijd van het begin van de baan naar het einde van de baan.
5. Bal beweegt zich van het einde van de baan naar de speler, en dan weer terug naar het einde van de baan.
Speler bevindt zich aan het begin van de baan.
6. Bal beweegt zich van het einde van de baan naar het begin van de baan.
Speler beweegt zich tegelijkertijd met de bal van het einde van de baan naar halverwege de baan.

7. Bal en speler bewegen zich tegelijkertijd van het einde van de baan naar het begin van de baan. De speler bevindt zich ongeveer een halve meter achter de bal.
8. Bal beweegt zich van het einde van de baan naar het begin van de baan. Speler loopt voor de bal langs wanneer de bal halverwege de baan is.
9. De bal wordt snel van het einde van de baan naar het begin van de baan bewogen, waarna de bal weer terug gaat van het begin van de baan naar het einde van de baan.
10. Bal beweegt zich van het einde van de baan naar het begin van de baan. Speler blijft stilstaan aan het eind van de baan.



Figuur 7.2 Overzicht van de verschillende situaties.

De spelsituaties waarop de evaluatie gebaseerd is, zijn sterk gesimplificeerd in relatie tot echte situaties. Zo bevinden bijvoorbeeld slechts maximaal twee objecten zich in de radarbundel en wordt er geëvalueerd binnen een klein gebied. Het is belangrijk om hier rekening mee te houden wanneer er conclusies uit de evaluatie getrokken worden.

7.3 Evaluatie deelsystemen

Om de invloed van verschillende deelsystemen op het hele systeem te kunnen onderzoeken, is het systeem na verschillende stappen in de dataverwerking geëvalueerd.

Eerst is het deelsysteem waar objectdetectie plaatsvindt geëvalueerd. Daarna is de tweede stap in het dataverwerkingsproces, objecttracking, geëvalueerd. Vervolgens is de objectclassificatie geëvalueerd en tot slot is de uiteindelijke beslissing die het systeem neemt geëvalueerd. Deze paragraaf beschrijft de resultaten van de diverse evaluaties.

7.3.1 Evaluatie van objectdetectie

In deze paragraaf wordt de objectdetectie, de eerste stap in de dataverwerking, geëvalueerd. De objectdetectie is gebaseerd op piekdetectie. De drempel die bij de piekdetectie gebruikt wordt is op drie verschillende manieren bepaald. Ten eerste is er een vaste drempel, ten tweede een OS-CFAR drempel, en ten derde een CML-CFAR drempel. De vaste drempelwaarde en de parameters van de CFAR detectors zijn gebruikt zoals beschreven in paragraaf 5.2.

De evaluatie vindt plaats aan de hand van twee criteria, het aantal valse alarmen en het aantal gemiste objecten. Een gemist object in deze evaluatie is een gemiste bal en wordt geteld voor elk frame waarin de bal niet gedetecteerd wordt. Het missen van een mens wordt niet als een gemist object geteld, omdat mensen weliswaar aanwezig zullen zijn op het veld, maar het systeem gemaakt is om ballen te detecteren. Een vals alarm is ieder gedetecteerd object dat in werkelijkheid niet aanwezig is. Gedetecteerde mensen worden niet als vals alarm geteld. De aantallen worden vervolgens omgezet in procenten. Hiertoe wordt gekeken in een vast afstandsbereik: de eerste zestig afstandsbins. Het aantal bins wordt met het aantal frames dat een dataset lang is vermenigvuldigd om tot een totaal aantal bins te komen. Uit het aantal fouten en het totaal aantal bins is te berekenen in hoeveel procent van de bins een vals alarm optreedt. Voor het bepalen van het percentage gemiste objecten wordt gekeken in hoeveel procent van de frames geen bal gevonden is. Hieronder zullen per drempelmethode de resultaten van de evaluatie besproken worden.

Vaste drempel

De gemiddelde aantallen fouten zijn per type en in procenten weergegeven in tabel 7.1. Het meest opvallende bij de vaste drempel is dat het aantal valse alarmen erg klein is: 0,01 procent. In drie van de tien situaties doet zich helemaal geen vals alarm voor terwijl in de andere zeven situaties het vals alarm percentage maximaal 0,034 procent is.

Het percentage gemiste objecten is een stuk hoger hoewel in drie van de tien situaties er geen objecten gemist worden. Voor situaties 7 en 10 ligt het aantal gemiste objecten onder de 1%, terwijl dat voor situaties 2, 8 en 9 tussen de 1,6 en 6,3% ligt. Voor situatie 3 en 4 zijn de getallen anders. In situatie 3 wordt bijna 20 procent van de ballen gemist. In situatie 4 is dat zelfs ruim 23% met een uitschieter van 42% in dataset 3 van situatie vier. Situaties drie en vier zijn situaties waarin de bal en een persoon elkaar kruisen: de bal komt naar de radar toe en de mens loopt van de radar weg, of andersom. Op het moment dat de bal zich verder van de radar bevindt dan de mens, wordt de bal door het detectiesysteem met de vaste drempel erg slecht gedetecteerd.

Tabel 7.1 Gemiddeld foutenpercentage per type.

Type fout	Drempelmethode		
	Vaste drempel	OS-CFAR	CML-CFAR
Missed Target	5,44%	0,45%	0,45%
Vals alarm	0,01%	1,01%	0,63%

CFAR

Het percentage valse alarmen ligt voor de OS-CFAR detector op gemiddeld 1,01 procent en is voor de CML-CFAR iets meer dan de helft van de OS-CFAR detector.

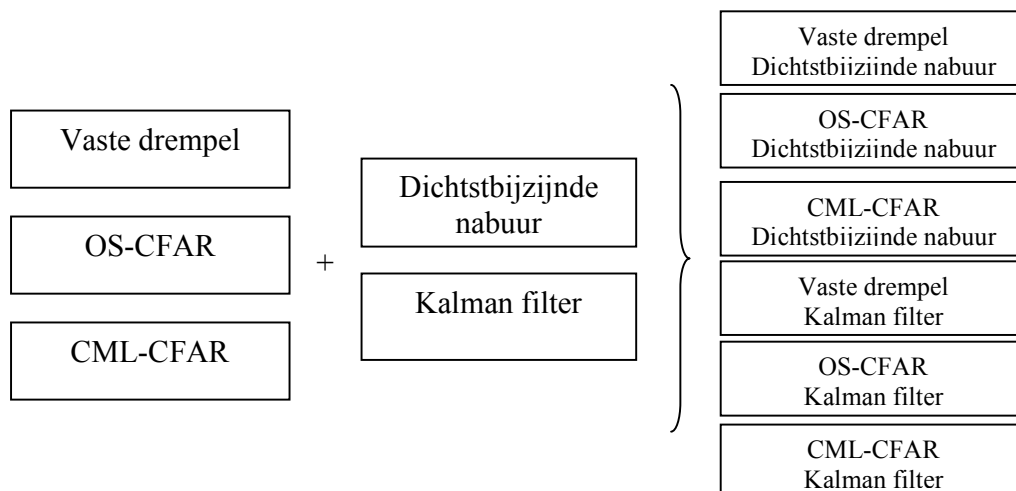
De beide CFAR detectors missen bijna nooit een object. In slechts drie van de tien situaties wordt er een object gemist, dit zijn situaties twee, acht en negen. Voor beide methodes zijn de percentages precies gelijk en liggen onder de 3%. Ook de vaste drempelmethode mist objecten in deze situaties, maar de CFAR detectors missen echter in alle drie de situaties slechts de helft van het aantal objecten dat de vaste drempelmethode mist.

7.3.2 Evaluatie van objecttracking

In deze paragraaf worden verschillende combinaties van het volgen van objecten in combinatie met de in paragraaf 7.3.1 besproken objectdetectiemethodes geëvalueerd. De twee alternatieven voor objecttracking zijn geëvalueerd met behulp van de spelsituaties beschreven in paragraaf 7.2. De evaluatie moet uitwijzen welke van de zes combinaties, zie

figuur 7.3, het beste trackingresultaat oplevert. Er is geëvalueerd aan de hand van vier criteria:

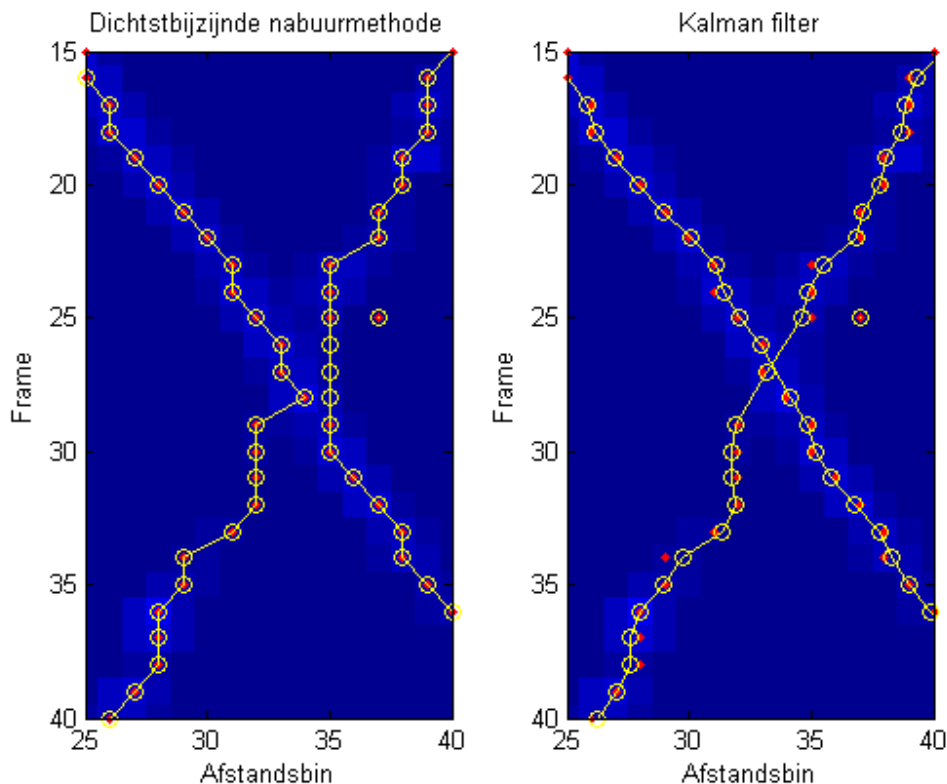
- Het aantal keer dat een track van de bal onderbroken wordt, of verwisseld wordt met een andere track.
- Het aantal keer dat een track van de bal overloopt in de track van een speler, of andersom.
- Het aantal keer dat een track van de bal onderbroken wordt als gevolg van een vals alarm.
- Het aantal tracks die extra ontstonden door vals gedetecteerde pieken.



Figuur 7.3 Combinaties voor evaluatie objecttracking.

Voor het eerste criterium is gekozen omdat het de functie van het trackinggedeelte is om een bal te volgen. Wanneer er onderbrekingen optreden in de track wordt de objectclassificatie ernstig verstoord, onderbrekingen zijn dus niet wenselijk. Onderbrekingen van de baan van de bal kunnen verschillende oorzaken hebben. Het

trackingalgoritme kan een voetbal en de speler verwarren wanneer er de banen kruisen, of wanneer de voetbal en de speler zich heel dicht bij elkaar bevinden. Een voorbeeld van een situatie waarin de banen al dan niet in elkaar overlopen is afgebeeld in figuur 7.4. Deze gebeurtenissen vallen onder het tweede criterium. Daarnaast kan de track van de voetbal ook onderbroken worden doordat er een valse piek gedetecteerd wordt in de buurt van de baan van de bal, en het trackinggedeelte de valse piek ziet als deel van de baan. Deze gebeurtenis valt onder het derde criterium. Het kan ook voorkomen dat het trackinggedeelte banen probeert te vormen van vals gedetecteerde pieken. Deze valse banen vormen een extra belasting voor het trackingsysteem. Het vierde criterium beschrijft het aantal valse banen.



Figuur 7.4 Het kruisen van banen in situatie 4, dataset 2. De nabuurmethode verwisselt de bal en de speler, het Kalman filter verwisselt de banen niet.

In tabel 7.2, tabel 7.3 en tabel 7.4 zijn de resultaten weergegeven van de evaluatie. Wat direct opvalt is dat het Kalman filter veel minder vaak de bal verliest dan de nabuurmethode. Ook blijkt dat het Kalman filter ruim twee keer minder vaak de bal verwisselt met een speler, onafhankelijk van de detectiemethode. Zoals verwacht blijkt uit tabel 7.4 dat de CFAR detectors vaker de bal verliezen als gevolg van een valse piek dan dat dat gebeurt bij een vaste grens.

Tabel 7.2 Het gemiddeld aantal keer per 100 frames dat de bal werd kwijtgeraakt.

		Objectdetectiemethode		
		Vaste drempel	OS-CFAR	CML-CFAR
Objecttrackingmethode	Nabuurmethode	2,99	3,14	2,78
	Kalman filter	1,96	1,67	1,61

Tabel 7.3 Het gemiddeld aantal keer per 100 frames dat de bal kwijtgeraakt werd doordat een speler en een bal verwisseld werden.

		Objectdetectiemethode		
		Vaste drempel	OS-CFAR	CML-CFAR
Objecttrackingmethode	Nabuurmethode	2,39	2,29	2,10
	Kalman filter	0,99	1,03	0,97

Tabel 7.4 Het gemiddeld aantal keer per 100 frames dat de bal kwijtgeraakt werd door valse pieken.

		Objectdetectiemethode		
		Vaste drempel	OS-CFAR	CML-CFAR
Objecttrackingmethode	Nabuurmethode	0,07	0,68	0,58
	Kalman filter	0,06	0,40	0,28

In tabel 7.5 is het gemiddeld aantal gevormde banen als gevolg van vals gedetecteerde pieken weergegeven. Zoals verwacht naar aanleiding van de resultaten van de evaluatie van objectdetectie, is het aantal vals gevormde banen voor de CFAR detectors veel hoger dan voor de vaste drempel. In lijn met deze resultaten blijkt dat de prestaties van CML-CFAR beter zijn dan de prestaties van OS-CFAR.

Tabel 7.5 Het gemiddeld aantal gevormde valse banen als gevolg van valse pieken per 100 frames.

		Objectdetectiemethode		
		Vaste drempel	OS-CFAR	CML-CFAR
Objecttrackingmethode	Nabuurmethode	0,43	12,7	10,8
	Kalman filter	0,51	15,1	11,8

Naast de gemiddelde resultaten waren er een aantal opvallende resultaten. Tijdens de evaluatie bleek dat met name in de situaties 6, 7 en 10 waarbij de bal en speler zich dicht bij elkaar bevonden het moeilijk was voor beide trackingmethodes om een juiste baan te vormen. Het aantal keer dat de bal werd kwijtgeraakt lag in deze situaties ruim twee keer hoger dan het gemiddelde resultaat. Het Kalman filter presteerde ten opzichte van de nabuurmethode ongeveer twee keer beter.

Een andere opvallend resultaat kwam naar voren tijdens de evaluatie van situatie 9. Omdat in deze situatie de bal in sommige datasets even uit de radarbundel verdwijnt, moet het trackingsysteem een nieuwe baan initialiseren voor een bewegende bal. Het Kalman filter blijkt hier grote moeite mee te hebben, waardoor een groot aantal gedetecteerde pieken niet in een baan opgenomen worden. Een verklaring voor dit resultaat is te vinden in de methode van implementatie. In de implementatie wordt er vanuit gegaan dat een object initieel een snelheid van nul heeft.

7.3.3 Evaluatie van objectclassificatie

De volgende stap in de dataverwerking is het classificeren van de gevonden banen als zijnde een voetbal of niet. Het prototype gebruikt, zoals te lezen is in paragraaf 6.5, de mate van constantheid van reflecties van objecten om te bepalen of een gevormde baan een voetbal is of niet. Het al dan niet juist classificeren van de gevonden banen is apart geëvalueerd, de resultaten worden in deze paragraaf besproken.

De evaluatie is verricht aan de hand van de in paragraaf 7.2 besproken spelsituaties. Voor elke dataset is bepaald hoeveel procent van de baan van de bal juist geclassificeerd werd. De evaluatie is gedaan voor alle mogelijke verschillende combinaties van objectdetectie en objecttracking als voorliggende stappen in de dataverwerking, zie figuur 7.3.

Uit de evaluatie blijkt dat de gebruikte classificatiemethode 42% beter functioneert wanneer objecten zich niet dicht bij elkaar bevinden ten opzichte van wanneer ze voor de radar duidelijk onderscheidbaar zijn. Dit kan verklaart worden doordat objecten die zich dicht bij elkaar bevinden slecht door de beide trackingmethodes gevolgd worden, zoals bleek uit de evaluatie van objecttracking in paragraaf 7.3.2. Wanneer de vaste drempel gebruikt wordt, wordt ongeacht de trackingmethode 47% van de baan van de bal juist geclassificeerd. Wanneer er gebruik wordt gemaakt van een CFAR-detector is er wel een aanmerkelijk verschil in de classificatieresultaten bij de verschillende trackingmethodes. Wanneer OS-CFAR en CML-CFAR in combinatie met de nabuurmethode gebruikt worden, wordt respectievelijk 33% en 41% van de baan juist geclassificeerd. In combinatie met het Kalman filter bereiken OS-CFAR en CML-CFAR betere resultaten, respectievelijk 47% en 49% werd juist geclassificeerd.

De gemiddelde classificatieresultaten per combinatie van de situaties waarin de bal zich niet dichtbij de speler bevindt zijn weergegeven in tabel 7.6.

Tabel 7.6 Resultaten per combinatie in procenten voor situaties waarin de bal zich niet dichtbij de speler bevindt.

		Objectdetectiemethode		
		Vaste drempel	OS-CFAR	CML-CFAR
Objecttrackingmethode	Nabuurmethode	79%	63%	74%
	Kalman filter	77%	73%	80%

Uit de resultaten in tabel 7.6 blijkt dat voor situaties waarin de bal zich niet dichtbij de speler bevindt, de CML-CFAR detector in combinatie met het Kalman filter het beste presteert. Opvallend is dat de bevindingen uit paragraaf 7.3.2 worden bevestigd. Wanneer een vaste drempel gebruikt wordt, presteert de objectclassificatie in combinatie met de nabuurmethode beter dan wanneer het Kalman filter gebruikt wordt. Wanneer een CFAR-detector gebruikt wordt, is de objectclassificatie beter wanneer het Kalman filter gebruikt wordt om objecten te detecteren.

Opvallend zijn de aanmerkelijk slechtere resultaten voor beide trackingmethodes wanneer OS-CFAR gebruikt wordt. Er blijkt dat wanneer OS-CFAR gebruikt wordt, het relatief vaker voorkomt dat een valse baan als bal geclassificeerd wordt.

7.3.4 Evaluatie van de besluitvorming van het systeem

Het doel van het totale systeem is om op het juiste moment de beeldverwerkingsoftware in te schakelen. In deze paragraaf worden de prestaties van het systeem besproken op basis van de beslissing die genomen wordt. De prestaties zijn geëvalueerd met behulp van de Evaluatie Tool besproken in paragraaf 7.1 en de spelsituaties beschreven in paragraaf 7.2. Tijdens de evaluatie zijn twee percentages bepaald die vergeleken kunnen worden met de eisen gesteld in het programma van eisen in hoofdstuk 3:

- Het percentage dat het systeem ingeschakeld werd wanneer er zich een bal in het inschakelgebied bevond.
- Het percentage dat het systeem terecht voor een naderende bal ingeschakeld werd.

Tijdens de evaluatie is de grens x_{grens} tussen het eerste en tweede gebied gekozen op 3,75 meter. Objecten die zich aan het begin van de baan bevinden, bevinden zich dus altijd in het inschakelgebied.

De gemiddelde resultaten van de evaluatie zijn weergegeven in tabel 7.7 en tabel 7.8. Omdat in situatie 2 de bal zich niet binnen het doelgebied beweegt, is deze situatie niet meegenomen in de gemiddelde percentages van gedetecteerde ballen.

Tabel 7.7 Percentage gedetecteerde ballen.

		Objectdetectiemethode		
		Vaste drempel	OS-CFAR	CML-CFAR
Objecttrackingmethode	Nabuurmethode	90%	88%	88%
	Kalman filter	87%	85%	86%

Tabel 7.8 Percentage dat het beeldverwerkingsysteem terecht werd ingeschakeld voor een naderende bal.

		Objectdetectiemethode		
		Vaste drempel	OS-CFAR	CML-CFAR
Objecttrackingmethode	Nabuurmethode	97%	85%	93%
	Kalman filter	97%	85%	93%

Het percentage gedetecteerde ballen blijkt ongeacht de objectdetectie- of trackingmethode tussen de 85 en 90% te liggen. Opvallend dat de percentages die aangeven of het systeem inderdaad voor een naderende bal ingeschakeld werd voor beide trackingmethodes gemiddeld gelijk zijn.

Twee situaties die erg van de gemiddelde resultaten afwijken zijn situatie 5 en situatie 9. In situatie 5 wordt de stilstaande persoon in het doelgebied vaak geïdentificeerd als bal. Om deze reden werd vaak een verkeerd besluit genomen. In situatie 9 beweegt de bal in enkele datasets zo snel dat de trackingalgoritmes niet in staat zijn om de bal te volgen. Ook komt het in deze datasets voor dat de bal door een slingering uit de radarbundel verdwijnt. Hierdoor wordt de bal niet altijd volledig gezien in het 'inschakelgebied' en treden er meer fouten op bij het maken van de beslissing of het beeldverwerkingsysteem ingeschakeld moet worden. Omdat de implementatie van het Kalman filter grote moeite heeft om snel bewegende tracks te initialiseren, maakt het Kalman filter ten opzichte van de nabuurmethode in deze situatie meer fouten wanneer de bal weer in de bundel slingert.

8 Conclusies en aanbevelingen

Dit hoofdstuk behandelt in paragraaf 8.1 de conclusies die volgen uit de evaluatie van het ontwikkelde prototype. Ook wordt in deze paragraaf het ontwikkelde prototype getoetst aan het programma van eisen in hoofdstuk 3. Daarna worden in paragraaf 8.2 op basis van het ontwikkelde prototype enkele aanbevelingen gedaan voor de uiteindelijke implementatie.

8.1 Conclusies en toetsing

In deze paragraaf worden conclusies getrokken naar aanleiding van de evaluaties in hoofdstuk 7. Tot slot wordt in paragraaf 8.1.6 het ontwikkelde prototype getoetst aan het programma van eisen in hoofdstuk 3.

8.1.1 Objectdetectie

In het prototype zijn drie verschillende methodes om een drempel te bepalen voor de piekdetectie getest. Twee hiervan zijn CFAR detectors, OS-CFAR en CML-CFAR, en de derde is een vaste drempelwaarde. De methodes zijn getest op het percentage valse alarmen en het percentage gemiste objecten. Uit de evaluatie blijkt dat de vaste drempel een veel lager percentage valse alarmen geeft dan de 1 procent die berekend was. Waarschijnlijk is dit veroorzaakt door een verkeerde schatting van het ruisvermogen in het referentiegebied waar de metingen gedaan zijn. Het ruisvermogen zal in werkelijkheid lager geweest zijn dan wat er in de berekeningen gebruikt is waardoor het percentage valse alarmen lager is dan berekend. Ook wijken de resultaten van de CFAR detectors af van de berekende detectie en vals alarm kansen. Een verklaring hiervoor zou kunnen zijn dat het toegepaste model in de berekening van de kansen niet volledig van toepassing is op de situaties tijdens de evaluatie.

Het opvallendste verschil tussen de vaste drempelmethode en de CFAR detectors is dat in de situaties waar de vaste drempel 20 tot 40% van de ballen in een dataset mist de CFAR detectors geen enkel gemist object hebben. Hieruit blijkt het voordeel van een adaptieve drempel. De vaste drempelmethode kan de bal niet detecteren als die deels achter een ander object schuil gaat. Dit komt omdat een deel van de energie die door de bal terug gekaatst zou worden nu door het object dat zich deels voor de bal bevindt teruggekaatst wordt. De CFAR detectors passen de drempel aan waardoor de bal toch te zien is. Dit maakt de CFAR detectors tot zeer geschikte kandidaten voor de uiteindelijke implementatie.

De miskans voor beide CFAR detectors is gelijk, maar de vals alarm kans bij CML-CFAR is iets meer dan de helft van de OS-CFAR detector. Aangezien valse alarmen tot grotere druk op het trackingsysteem leiden is de CML-CFAR detector het meest geschikt om toegepast te worden in het prototype.

8.1.2 Objecttracking

In het prototype zijn twee verschillende methodes geïmplementeerd om objecten te volgen, een trackingmethode die gebruik maakt van het dichtstbijzijnde nabuuralgoritme en een methode die gebruik maakt van een drie toestanden Kalman filter. De twee methodes zijn geëvalueerd in combinatie met de verschillende beschikbare objectdetectiesystemen. Tijdens de evaluatie is gekeken naar het aantal keer dat de baan van de bal onderbroken werd, en waardoor dit kwam, en het aantal tracks dat ontstond als gevolg van valse detecties.

Uit de evaluatie blijkt dat het Kalman filter ballen beter volgt dan de nabuurmethode. In situaties waarbij twee objecten zich dicht bij elkaar bevinden treden relatief meer fouten op dan in andere situaties. Over het algemeen presteert het Kalman filter beter dan de nabuurmethode. Dit kan verklaart worden door het feit dat het Kalman filter rekening houdt met de dynamica van het object. Het Kalman filter blijkt daarnaast beter te werken in combinatie met een CFAR detector dan met een vaste drempel, ondanks dat er als gevolg van de CFAR detector banen onderbroken zullen worden als gevolg van valse detecties.

Geconcludeerd kan worden dat het Kalman filter in combinatie met de CML-CFAR detector het beste presteert. Een nadeel van deze combinatie is wel dat er ook valse banen worden gevormd.

8.1.3 Objectclassificatie

In het prototype vindt de classificatie plaats op grond van slechts één eigenschap van een object, de mate van verandering in reflectie. Tijdens de classificatie kan slechts één fout optreden en dat is dat een bal niet tot bal geëvalueerd wordt. Deze fout kan verschillende oorzaken hebben. Ten eerste kan een object dat weinig fluctueert zich dicht bij de radar bevinden dan de bal. In dit geval wordt dit object als bal geëvalueerd omdat bij meerdere kandidaten voor de bal het dichtstbijzijnde object bal wordt gemaakt. Ten tweede kan de reflectie van de bal toch fluctueren. Dit kan komen doordat de bal deels achter een ander object schuil gaat en daardoor minder reflectie geeft. Als de oppervlakte van de bal dat niet zichtbaar voor de radar is fluctueert, dan fluctueert de reflectiestrkte van de bal ook. Het kan ook zijn dat het Kalman tracking algoritme de verkeerde plaats van de piek van de bal doorgeeft omdat door afronding een bin naast de werkelijke piek doorgegeven wordt als locatie van de piek. Omdat de reflectie in een naastgelegen bin veel lager is dan op de piek zelf leidt dit tot fluctuaties.

De classificatiemethode zoals toegepast in het prototype werkt alleen op die momenten waarin de bal volledig te zien is. Dat betekent dat de bal en een speler zich niet erg dicht bij elkaar mogen bevinden. Als dit niet het geval is fluctueert de reflectie van de bal teveel. Uit evaluatie in paragraaf 7.3.3 blijkt dat het classificeren van de bal wanneer de speler zich dichtbij de bal bevindt 30% slechter gaat vergeleken met situaties waarin de bal zich verder weg van de speler bevindt.

Uit de evaluatie in paragraaf 7.3.3 kan tevens geconcludeerd worden dat het classificeren van de bal de beste resultaten geeft wanneer de combinatie vaste grens - nabuurmethode of de combinatie CML-CFAR – Kalman filter gebruikt wordt. Bij deze combinaties wordt 80% van de ballen juist geëvalueerd. Een zo hoog mogelijke mate van juiste classificatie is nodig om een zo goed mogelijk besluit te vormen. De mindere

resultaten voor de overige combinaties met een CFAR detector kunnen verklaard worden door het feit dat het prototype veel valse banen classificeert als bal.

8.1.4 Besluitvorming

Voor het prototype is ook de methode van besluitvorming geëvalueerd. Hierbij is gekeken naar het aantal keer waarin het beeldverwerkingsysteem werd ingeschakeld wanneer er zich een bal in het doelgebied bevond. Daarnaast is gekeken of het object waarvoor het systeem werd ingeschakeld inderdaad een bal was.

Uit de evaluatie blijkt dat het prototype voor de verschillende combinaties van deelsystemen tussen de 85 en 90% van de geschoten ballen het beeldverwerkingsysteem waarschuwt. Daarbij moet worden opgemerkt dat resultaten zijn uit een evaluatie in een sterk versimpelde omgeving. De besluitvorming functioneert in het prototype het beste wanneer er gebruik wordt gemaakt van een vaste grens voor objectdetectie en de dichtstbijzijnde nabuurmethode voor objecttracking. In paragraaf 8.1.3 werd geconcludeerd dat ook de combinatie CML-CFAR – Kalman filter vergelijkbare prestaties heeft als combinatie vaste grens – nabuurmethode. De verklaring voor het feit dat deze combinatie slechter presteert als het gaat om besluitvorming ligt voornamelijk in het feit het Kalman filter moeite heeft met het initialiseren van bewegende tracks. Vooral in situatie 9, waarbij de bal in sommige datasets helemaal uit de radarbundel verdwijnt, levert het Kalman filter hierdoor een heel stuk in als het gaat om prestatie.

Uit de evaluatie blijkt dat het beeldverwerkingsysteem de prestaties van het al dan niet terecht inschakelen van het systeem voornamelijk afhangt van de objectdetectiemethode. De beste prestatie worden geleverd door de methode met de vaste grens, het systeem wordt in 97% van de gevallen inderdaad ingeschakeld omdat een bal het doel naderde. Wanneer een OS- of CML-CFAR detector wordt gebruikt lagen de percentages op respectievelijk 85 en 93%. Een verklaring voor het feit dat de besluitvorming bij gebruik van een CFAR-detector slechter presteert kan zijn dat het aantal valse banen dat gevormd wordt als gevolg van valse detecties hoger is dan wanneer een vaste grens gebruikt wordt. Als gevolg hiervan treden er meer verkeerde classificaties op. Deze verkeerd geclassificeerde valse banen kunnen zich ook in het inschakelgebied bevinden, waardoor de juistheid van besluitvorming slechter wordt.

8.1.5 De snelheid van het prototype

De snelheid van het systeem is een belangrijke parameter. Tijdens de evaluatie is gebleken dat het prototype één nieuw frame per 0,13 seconden kan verwerken. De oorzaak hiervan is toe te wijzen aan de beperkte prestaties van de beschikbare computerapparatuur en de programmeertaal waarin de software geschreven is. Als gevolg van de lage frame-rate kan het prototype, wanneer het gebruik maakt van het nabuuralgoritme, ballen tot een snelheid van ongeveer 3,5 m/s volgen. Deze snelheid wordt bepaald door het maximale aantal afstandsbins dat er tussen een nieuw gevonden object en een al bestaand track mag zitten zodat het object aan het track toegevoegd wordt en de frame-rate. In het prototype mag er maximaal 3 bins tussen een bestaande track en een nieuwe piek zitten. Een afstandsbin is 15 cm breed, en dus kan de bal maximaal 45 cm/frame bewegen zonder dat de track breekt. Dit levert de snelheidsgrens van 3,5 m/s op. Voor het Kalman algoritme bestaat een dergelijke grens niet.

8.1.6 Toetsing aan het programma van eisen

In het programma van eisen in hoofdstuk 3 wordt gesteld dat het systeem ballen moet kunnen volgen die met een snelheid van maximaal 30 m/s bewegen. In het geval van het nabuuralgoritme is dat in het prototype niet mogelijk. De frame-rate van het prototype is dermate laag dat bij deze maximale snelheid de bal zich bijna 4 meter verplaatst tussen ieder frame. Dat betekent dat als er van twintig meter geschoten wordt het systeem slechts 5 frames heeft met de bal daarin. Zelfs al zou het systeem deze vijf frames succesvol tot één baan vormen dan zou de reflectie van de bal over de afgelegde twintig meter dermate veranderen dat het classificatiesysteem de baan nooit als bal zou classificeren.

Een andere snelheidsgrens die in het programma van eisen genoemd wordt is dat het systeem binnen 33 ms moet bepalen of een object een bal of een mens is. Voor de classificatie van een nieuw gevonden object, zoals die in het prototype gebruikt wordt, zijn minimaal drie frames nodig. Met een verwerkingstijd van 0,13 seconde per frame is er 0,39 seconde nodig om een nieuw gevonden object te classificeren. Het prototype is dus ruim een factor tien te traag om aan deze eis te voldoen.

Naast de eisen aan de maximale te tracken snelheid zijn er ook twee eisen geformuleerd over de hoeveelheid fouten die mogen optreden. Ten eerste is er geëist dat 98% van de ballen die op het doel geschoten worden gedetecteerd moeten worden. Het prototype detecteert gemiddeld maximaal 90% van de ballen die geschoten werden op het doel. Ten tweede is geëist dat wanneer het systeem ingeschakeld worden, er in minimaal 75% van de gevallen ook daadwerkelijk een bal het doel moet naderen. Het ontwikkelde prototype voldoet wel aan die eis, er blijkt immers dat het besluit om het beeldverwerkingsysteem in te schakelen in 85 tot 97% van de gevallen inderdaad genomen wordt als er daadwerkelijk een bal het inschakelgebied nadert. Er kan dus geconcludeerd worden dat één eis ten aanzien van fouten wel is gehaald en de andere eis niet. Wel moet daarbij opgemerkt worden dat de evaluatie gebaseerd is op eenvoudige spelsituaties, waarbij de radar in het overgrote deel van de situaties vrij zicht had op objecten in het doelgebied.

De conclusies in de paragrafen 8.1.4 en 8.1.5 herleiden het missen van ballen in het inschakelgebied naar zowel de keuze van methode voor objectdetectie als de keuze van de methode voor objecttracking. Om de gestelde eis wel te halen moeten het detectie- en trackingsysteem beter afgesteld worden of er moet er een beter alternatief gevonden worden om ballen te detecteren en volgen. Ondanks dat de eis voor het wel of niet geldig inschakelen van het beeldverwerkingsysteem wel gehaald is, is ook op dit vlak verbetering mogelijk. Wanneer de baan van de bal vaker juist geclassificeerd wordt zal het beeldverwerkingsysteem in meer gevallen alleen op het juiste moment worden ingeschakeld.

8.2 Aanbevelingen voor de uiteindelijke implementatie

Voor verdere ontwikkeling van het prototype naar de uiteindelijke implementatie kunnen op basis van de voorgestelde deelconcepten in hoofdstuk 4 en de conclusies in paragraaf 8.1 een aantal aanbevelingen worden gedaan:

- *Radaropstelling*
In het uiteindelijke systeem wordt waarschijnlijk gekozen voor een oplossing met meerdere wide-angle radars of een scanning radar. Omdat het prototype net als een scanning radar met een smalle bundel werkt, wordt aanbevolen om eerst de mogelijkheden van een scanning radar te onderzoeken.
In het prototype is geen rekening gehouden met het gebruik van meerdere of scanning radars. Het prototype dient op dit vlak dus aangepast te worden om gebruikt te kunnen worden in de uiteindelijke implementatie.
- *Objectdetectie en –tracking*
Op basis van de conclusies over objecttracking in paragraaf 8.1.2 wordt aanbevolen om in het uiteindelijke systeem een verbeterde versie van het Kalman filter in combinatie met een adaptieve CML-CFAR detector op te nemen. Daarnaast wordt aanbevolen om deze combinatie in een meer realistische situatie te evalueren, zoals in op een voetbalveld.
- *Objectclassificatie*
Om de juistheid van besluitvorming te verbeteren is het noodzakelijk aanvullende classificatiemethoden te onderzoeken en te implementeren. Het verdient de aanbeveling om de classificatie te baseren op meerdere eigenschappen. Aanbevolen wordt om eigenschappen als de snelheid van de bal en polarisatie van de bal of de plaatsing van RFID-tags op de bal te onderzoeken.
- *Snelheid*
Om aan de gestelde snelheidseisen te kunnen voldoen, zijn er diverse optimalisaties nodig. Aanbevolen wordt om het systeem te implementeren in C.

Literatuurlijst

- [1] L. Abdou en F. Soltani, “OS-CFAR and CMLD threshold optimization in distributed systems using evolutionary strategies”, Springer-Verlag Londen Limited 2007, 2007.
- [2] Anoniem, “CE-markering: richtlijn elektromagnetische compatibiliteit (EMC)”, (2004/108/EG).
- [3] Anoniem, “CE-markering: richtlijn laagspanning”, (73/23/EG, 93/68/EG).
- [4] Anoniem. (29 april 2008). Cross-correlation. Wikimedia Foundation, Inc. [Online]. Beschikbaar: <http://en.wikipedia.org/wiki/Cross-correlation>.
- [5] Anoniem, “FMCW RADAR – Gebruiksaanwijzing”, ThalesGroup.
- [6] Anoniem, “IEEE standard signaling method for a bidirectional parallel peripheral interface for personal computers 2000”, IEEE 1284-2000, 20 september 2000.
- [7] Anoniem, “Information technology equipment – Safety – Part 1: General requirements”, NEN-EN-IEC 60950-1, 1 december 2005.
- [8] Anoniem, “Richtlijn 89/391/EEG van de Raad van 12 juni 1989 betreffende de tenuitvoerlegging van maatregelen ter bevordering van de verbetering van de veiligheid en de gezondheid van de werknemers op het werk”, (89/391/EEG), 12 juni 1989.
- [9] Anoniem. (1 juni 1998). The physics of football. Physicsworld.com. [Online]. Beschikbaar: <http://physicsworld.com/cws/article/print/1533>.
- [10] Y. Bar-Shalom en Th. E. Fortmann, “Tracking and Data Association”, Orlando: Academic Press, Inc, 1988.
- [11] O. Castellini en C. Fernández Canet. (2 april 2008). 2008 IAAF World Indoor Lists 8. [Online]. Beschikbaar: http://www.iaaf.org/mm/Document/Statistics/TopList/IAAFIndoorListsNo8_1969.pdf.
- [12] L. W. Couch, II, “Digital and Analog Communication Systems”, 7th ed., Upper Saddle River: Pearson Prentice Hall, 2007.
- [13] M. Deuring en J. He, “Doelpuntdetectie met beeldverwerking”, Delft: TU Delft, juni 2008, hoofdstuk 3.
- [14] M.B. El Mashade, “Analysis of the censored-mean level CFAR processor in multiple target and non uniform clutter.”, *IEEE Proc. Radar, Sonar Navig.* 142(5), p. 259–266, 1995.
- [15] I. Faulconbridge, “Radar Fundamentals”, Canberra: Argos Press, 2002.
- [16] B. Friedland, “Optimum steady state position and velocity estimation using noisy sampled position data”, *IEEE Transactions on Aerospace and Electronic systems*, AES-9, p. 906-911, november 1973.
- [17] H.D. Griffiths, “New ideas in FM radar”, *Electronics and Communication Engineering Journal*, vol 2, No. 5, oktober 1990, p. 185-194.
- [18] J.R. Holm en J.A. Ritcey, “The optimality of the censored meanlevel detector”, *IEEE Trans. Inf. Theory* 37(1), p. 206–209, 1991.
- [19] M.O. Kolawole, “Radar Systems, Peak Detection and Tracking”, Oxford: Elsevier Science, 2002.

- [20] M. Meibergen en H. van Zandbrink, “Doelpuntendetectie met behulp van radartechnologie”, Delft: TU Delft, juni 2008.
- [21] I. Nicolaescu en T. Oroian, “Radar cross section”, *TELSIKS 2001*, vol 1, p. 19-21, 2001.
- [22] R. Nitzberg, “Radar Signal Processing and Adaptive Systems”, Boston, London: Artech House, 1999.
- [23] K. Pentilla, M. Keskilammi, L. Syndanheimo en M. Kivikoski, “Radar cross-section analysis for passive RFID systems”, *Microwaves, Antennas and Propagation, IEEE Proceedings*, vol 153, No. 1, p. 103-109, 2006.
- [24] K.V. Ramachandra, “Kalman filtering Techniques for Radar Tracking”, New York: Marcel Dekker, 2000.
- [25] M.A. Richards. Fundamentals of Radar Signal processing. Rank-Based CFAR Techniques. [Online] Beschikbaar: http://users.ece.gatech.edu/~mrichard/Instructor_Resources/Sample%20Lecture%20pdfs/module%2096%20Rank-Based%20CFAR.pdf.
- [26] H. Rohling, “Radar CFAR thresholding in clutter and multiple target situations.”, *IEEE Trans. Aerospace Electronic Systems*. 19, p. 608-621, 1983.
- [27] M.K. Uner en P.K. Varshney, “Distributed CFAR detection in homogeneous and nonhomogeneous backgrounds”, *Aerospace and Electronic Systems, IEEE Transactions on*, p. 84-97, 1996.
- [28] M.D. Verweij, P.M. van den Berg en H. Blok, “Electromagnetic Waves, An Introductory Course”, Delft: Delft University Press, 2001.
- [29] R. Weinstein, “RFID: a technical overview and its application to the enterprise”, *IT Professional*, vol 7, No. 3, p. 27-33, 2005
- [30] A.G. Yarovoy, L.P. Ligthart, J. Matuzas en B. Levitas, “UWB Radar for Human Being Detection”, *Aerospace and Electronic Systems Magazine*, IEEE, vol 21, No. 11, p. 22-26, 2006.
- [31] R.D. Yates en D.J. Goodman, “Probability and Stochastic processes”, Hoboken: John Wiley & Sons, Inc, 2005, p. 379-382.

Appendix A Pseudo-code

A1 Pseudo-code dichtstbijzijnde nabuuralgoritme

```
% Initialisatie
maak buffer aan

% Buffer opschuiven
if (ID nog niet op lengte)
    voeg rij aan ID toe
else
    haal eerste rij ID weg
    voeg rij aan ID toe
end

pieken = haal pieken op

% Plaatsen van de pieken
for r = 1: aantal pieken
    bepaal dichtstbijzijnde track
    if (afstand <= grens) & (piek nog niet geplaatst)
        plaats piek in ID
    end
end
verwijder geplaatste pieken uit pieken array

if (pieken = niet leeg)
% Lijmen indien mogelijk
for r = 1: aantal pieken
    for i = 1:aantal tracks
        if (track gebroken) & (piek sluit aan)
            plaats piek
            interpoleer waardes
        end
    end
end
verwijder geplaatste pieken uit pieken array

if (pieken = niet leeg)
% Nieuwe track aanmaken
for r = 1 : aantal pieken
    voeg nieuwe kolom toe aan ID
    ID(end,end) = piek
end

% Updaten van infs
for i = 1: aantal tracks
    if (id(end,i) == 0)
        id(end,i) = inf;
    end
end

% Weghalen van afgebroken banen
for r = 1: aantal tracks
```

```

    if max(id(:,r) == inf
        haal kolom weg
    end
end
end

```

A2 Pseudo-code Kalman filter

Initialisatie:

maak track-vector aan

Voeg nieuw frame toe:

```

if buffer = leeg
    initialiseer buffer
end
verschuif buffer
voeg nieuw frame toe aan buffer

for i = 1 tot aantal tracks
    verschuif buffers
    verwijder tracks die voorheen bestonden maar nu leeg zijn
end

for j = 1 tot aantal tracks
    voorspel de waarde van track(j) op het tijdstip van het frame wat
    toegevoegd wordt met behulp van huidig model
end

piek = pieken in nieuw frame
for k = 1 tot aantal pieken
    bepaal welke track het dichtst bij piek(k) is voorspeld
    if geen track gevonden of resultaat onwaarschijnlijk ver weg
        initialiseer een nieuwe track
        voeg track toe aan tracks
    end
    verwerk gevonden piek in track met behulp van Kalman filter
    voeg piek toe aan de track
end

verwerk gegevens uit tracks tot ID-matrix

```


English summary

Maxad is a company that maximises advertisement profits for its customers. Maxad does this by developing a system that advertises the customers message during the highlights of a football match. The system detects the precise moment a goal is scored and displays an advertisement on LED-boarding behind the goal.

The goal detection system is based on radar and image processing technology. Maxad is divided into three subdivisions, MaxRadar, MaxSoft and MaxImage. MaxRadar researches ways to make the ball more visible for the radar, and develops the radar hardware. MaxSoft develops software to process the radar data in order to decide whether a ball is approaching the goal. MaxImage develops an image processing system that can determine with high precision whether the ball crosses the goal line. The MaxImage system is switched on when MaxSoft detects a ball approaching the goal.

This thesis describes MaxSoft's part of the project. MaxSoft's assignment is to develop software that can detect a football in FMCW radar data. The challenges of such a system are to distinguish the ball from other objects, to track the ball and doing so in complex situations that arise during the match.

The software is divided into three distinct parts; detection, tracking and classification. Good detection is necessary to track objects while good tracking is essential for good classification. Finally the decision whether to switch on the MaxImage system has to be made based on the tracking and classification results.

Possible solutions to the different parts of the final design have been studied and several concepts have been examined and tested. A prototype has been developed in Matlab based on the results found during the previously described research. The prototype has been designed primarily to test the different ways of determining the threshold for the object detection and the two approaches to track objects.

Objects are detected using a peak detection algorithm. The threshold that is used in the detection process can be either fixed or adaptive. When an adaptive threshold is used, it is possible to choose between two different versions of the CFAR method. To track objects, both the nearest neighbour algorithm and a method based on a Kalman filter can be used.

In the prototype, object classification is performed on the basis of how constant an object's reflection is. Other promising techniques, such as the detection of differences in reflection with changing polarisation angle, or the use of RFID chips have not been implemented because they were either considered to be unsuitable for the prototype or not available during the time of development.

The evaluation of the prototype shows that in greatly simplified game situations the system satisfies one out of two stipulated accuracy demands. The image processing system is switched on without a ball approaching in less than 25% of the time. The prototype, however, does not meet the demand that it is switched on for 98% of the approaching balls.

The best solution to detect and track objects is the combination of a Kalman system and a CML-CFAR detector.

The used classification method performs too poorly to be implemented unaltered in the final version of the system. Classification errors occur especially when the ball is close to an other object. When the ball is clearly visible to the radar the prototype achieves a maximum of 80% accurately classified balls.

For the development of the end product it is recommended to carry out further research into different object properties to base the classification on. Furthermore, it is necessary to study how the data of several radars can be combined. It is recommended to write the end product in a compiled language such as C instead of Matlab.