

Detectie, lokalisatie en vervanging van reclameborden
in een live video-stream van een voetbalwedstrijd



UNIVERSITEIT VAN AMSTERDAM

Bram Stoeller, 0426857
Tjerk Kostelijk, 0418889

15 juli, 2007

Inhoudsopgave

1	Samenvatting	3
2	Inleiding	3
2.1	Eerder werk	3
2.1.1	Real-Time Billboard Substitution in a Video Stream	3
2.1.2	Visual Tracking and Localisation of Billboards in Streamed Soccer Matches	3
3	Achtergrondinformatie	4
3.1	UvA	4
3.2	Synetic	4
3.3	Triple IT	4
4	Probleembeschrijving	4
5	Tools	4
6	Graslijndetectie	4
6.1	Grasdetectie	5
6.2	Houghlines	5
6.3	Houghlinecorrectie	6
7	Tracking	7
7.1	Voorspeller	7
7.1.1	Constraints	8
7.1.2	Multi-resolutie	9
7.2	Hoeklokalisatie op subpixelniveau	10
8	Vervanging	10
8.1	Shearing en interpolatie	10
9	Resultaten	11
9.1	Lokalisatie	11
9.2	Tracking	11
9.3	Vervanging	12
10	Discussie	13
10.1	Voorspeller	13
10.2	Multi-resolutie	13
11	Toekomstwerk	13
11.1	Reclamebord uit beeld	14
11.2	Lichtintensiteit en schaduwen	14
11.2.1	Background estimation	14
11.3	Geheel product	15
12	Conclusie	15

1 Samenvatting

Het doel is om reclameborden langs een voetbalveld te vervangen, zonder enige aanpassingen aan de huidige reclame borden of camera's. De enige informatie die beschikbaar is is basiskennis over wat we verwachte te zien in video beelden van een voetbalwedstrijd. En een constante video-stream van n van de camera's in een voetbalstadion. Het vervangen van de reclameborden is vooral interessant voor adverteerders die doelgericht willen adverteren bij internationale voetbalwedstrijden.

Allereerst zal het reclamebord gevonden moeten worden doormiddel van gras- en hoekdetectie. Vervolgens zal het reclamebord gevolgd moeten worden met behulp van een tracking-algoritme. In ieder frame moeten de hoekpunten gecorrigeerd worden. En de laatste stap is om het bord te vervangen door een andere afbeelding, zodat het lijkt alsof die afbeelding werkelijk op het reclamebord in het voetbalstadion staat. De kunst is om dit op een slimme manier te doen en alle kennis van de geometrie van de omgeving mee te nemen in de functies.

2 Inleiding

Een wereldkampioenschap voetballen word in heel veel landen uitgezonden, maar in ieder land is dezelfde reclame te zien langs de rand van het veld. Zou het niet handiger zijn als je per land zou kunnen beslissen welke reclame er getoond wordt? In 1998 heeft G. Medoni [1] hier voor het eerst over geschreven in zijn paper "Real-Time Billboard Substitution in a Video Stream". F. Aldershof en Th. Gevers [2] hebben eveneens een dergelijke methode geïmplementeerd. Een belangrijk aspect van de methode die beschreven wordt in dit verslag, is om meer rekening te houden met de geometrische eigenschappen van een voetbalveld en de borden er omheen.

2.1 Eerder werk

Het vervangen van reclameborden in voetbalwedstrijden is een zeer specifieke taak. Toch zijn over dit onderwerp een aantal interessante papers verschenen. In dit hoofdstuk worden de twee belangrijkste papers behandeld.

2.1.1 Real-Time Billboard Substitution in a Video Stream

Deze paper beschrijft een conceptuele aanpak van bovenstaand probleem. Het is een globaal systeem en dus ook toepasbaar op b.v. videobeelden van tenniswedstrijden. Er wordt gebruik gemaakt van feature detection om de beweging van de camera te detecteren. Dit is niet robuust voor motion blur. Verder is er een model beschikbaar over het te vervangen reclamebord. In het probleem wat dit verslag beschrijft is het model niet bekend en moet er gehandeld worden met motion blur. Daar het domein bekend is wordt kennis van de geometrie van de te verwachten objecten hiervoor gebruikt. [1]

2.1.2 Visual Tracking and Localisation of Billboards in Streamed Soccer Matches

Deze paper beschrijft een heldere aanpak van een system toegespitst op voetbalwedstrijden. Er wordt niet gesproken over perspectiviteit en wordt er een match gemaakt middels een kleuren-histogram. Het nadeel hiervan is dat dit lokatie onafhankelijk is. In dit verslag is een andere aanpak gehanteerd. [2]

3 Achtergrondinformatie

3.1 UvA

Dit project wordt als invulling van keuzeruimte (10 ECTS) uitgevoerd binnen de opleiding Kunstmatige Intelligentie (AI) aan de Universiteit van Amsterdam (<http://www.uva.nl>). Het project werd begeleid door dr Rein vd Boomgaard (<http://staff.science.uva.nl/~rein>) Verder werd dit project uitgevoerd in opdracht van Synetic (<http://www.synetic.nl>) en Triple IT (<http://www.triple-it.nl>).

3.2 Synetic

Synetic gesitueerd te Heemskerk, houdt zich bezig met diverse toepassingen met betrekking tot het internet, in het bijzonder met het ontwikkelen van websites en on-line applicaties. Synetic is de werkgever van Tjerk Kostelijk en Bram Stoeller en heeft een nauwe samenwerking met het bedrijf Triple IT.

3.3 Triple IT

Triple IT is vooral gespecialiseerd in nieuwe media. Ze werken aan innovatieve oplossingen voor onder andere videostreams op internet en mobiele apparatuur. Ze hebben een aantal grote klanten en staan op goede voet met nationaal en internationaal bekende televisiemakers als Tien, MTV, Endemol, CNN, etcetera.

4 Probleembeschrijving

Het doel van het project is, in een televisie-uitzending van een voetbalwedstrijd, de reclameborden langs de rand van het voetbalveld live te kunnen vervangen door andere reclame. Hierdoor kan de reclame die in beeld komt aangepast worden aan de lokale bevolking. Denk bijvoorbeeld aan uitzendingen van Europese of Wereld Kampioenschappen.

Voor dit probleem zijn door Triple IT een reeks video bestanden aangeleverd van een recente voetbalwedstrijd. De resolutie van de filmpjes is erg laag, (352×288 pixels), wat het project een uitdaging maakt.

5 Tools

De methoden die in dit verslag zijn beschreven zijn geïmplementeerd in *Python 2.5*, waarbij gebruik gemaakt is van de Computer Vision library *OpenCV 1.0*. De frames zijn met behulp van *FFMpeg* geëxtraheerd uit de door Triple IT aangeleverde filmpjes. De resultatenframes zijn weer samen gevoegd tot een filmpje met behulp van *JPGvideo*.

6 Graslijndetectie

De reclameborden langs de rand van een voetbalveld staan altijd op het gras. Dit houdt in dat de onderste rand van de reclameborden in het videobeeld gelijk is aan de rand van het grasveld. De rand van het grasveld is nauwkeurig te bepalen, gras is altijd groen en de pixelwaarden van de graspixels liggen relatief dicht bij elkaar in de *RGB* ruimte.

Wanneer de graspixels onderscheiden zijn van de niet-graspixels, wordt de lijn berekend die de scheiding tussen graspixels en niet-graspixels het beste beschrijft.

6.1 Grasdetectie

Om invloeden van lichtintensiteit en schaduwen te vermijden worden de pixelwaarden (RGB) genormaliseerd (rgb) (Frank Aldershoff en Theo Gevers [2]):

$$r = \frac{R}{R + G + B} \quad (1)$$

$$g = \frac{G}{R + G + B} \quad (2)$$

$$b = \frac{B}{R + G + B} \quad (3)$$

Vervolgens wordt iedere pixel p geclassificeerd, waarbij p_g de genormaliseerde waarde van het groene kanaal representeert.

$$P_{gras}(p) = \begin{cases} 0, & p_g \leq \frac{1}{3} \\ 1, & p_g > \frac{1}{3} \end{cases} \quad (4)$$

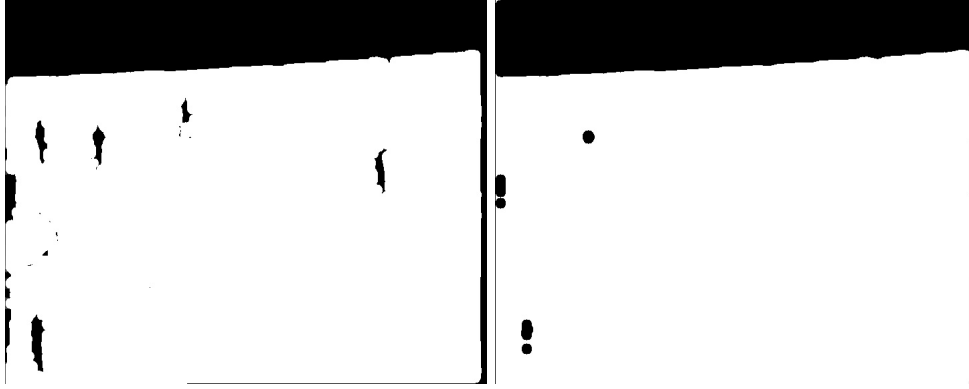
Hierdoor ontstaat een binaire afbeelding (Figuur 1) welke een aantal verkeerd geclassificeerde pixels bevat. Door middel van een dilatie gevolgd door een erosie worden de *false negatives* gecorrigeerd. Door dit proces te herhalen op het negatief van de afbeelding worden ook de *false positives* gecorrigeerd (Figuur 2).



Figuur 1: Iedere pixel is geclassificeerd als zijnde graspixel (*hoog*), danwel niet graspixel (*laag*).

6.2 Houghlines

De scheiding wordt gevonden door gebruik te maken van Houghlines. Het Houghline algoritme vindt rechte lijnstukken door witte pixels. Om deze reden wordt er een Canny-edge detection uitgevoerd op de output van de grasdetectie, een binair plaatje (Figuur 2). Nu de randen op pixelniveau gevonden zijn kan de Hough transformatie een aantal lijnstukken extraheren.



Figuur 2: Correctie van de *false positives* (links) en de *false negatives* (rechts) door middel van erosie en dilatie.

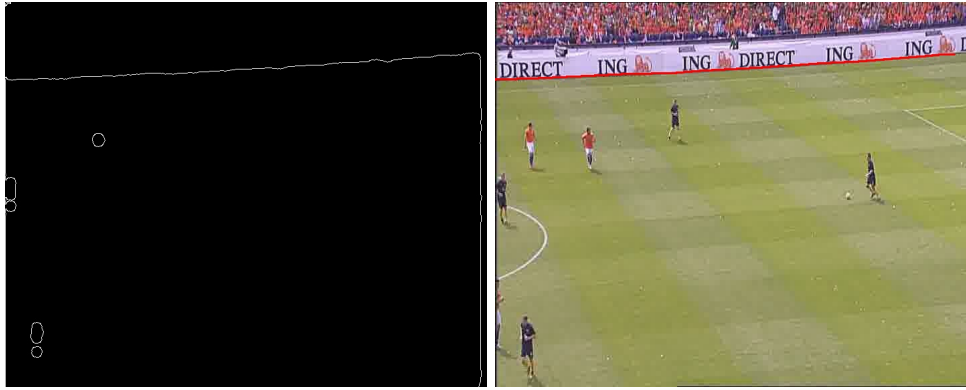
6.3 Houghlinecorrectie

Het houghline algoritme retourneert een aantal lijnstukken. Alleen de relatief horizontale lijnen kunnen de rand van de reclameborden beschrijven. Daarnaast ontstaat er een aantal verticale lijnen doordat er aan de zijkant van de frames soms een zwarte rand ontstaat, wat te maken heeft met de verschillende video conversies. Om deze twee redenen worden de lijnen die meer verticaal zijn dan horizontaal, ofwel met een absolute helling a groter dan 1 (Vergelijking 5), genegeerd.

$$a = \left| \frac{y_2 - y_1}{x_2 - x_1} \right| \quad (5)$$

Sommige lijnen overlappen elkaar (deels), anderen kruisen elkaar. Ook zijn er lijnen die elkaar niet raken of overlappen. Het doel is om een reeks coördinaten te vinden welke de rand van de reclameborden beschrijft (Figuur 3). Hiervoor wordt het volgende algoritme gebruikt (Zie ook Figuur 4):

1. Lijnen waartussen een gat ontstaan is. (paars)
 - Het eindpunt van de eerste lijn wordt verbonden met het begin van de volgende lijn. Er wordt dus een extra lijnstuk toegevoegd.
2. Lijnen die geheel binnen een andere lijn vallen. (rood)
 - Deze worden verwijderd. De reden hiervoor is dat een lange lijn een hogere evidentie heeft dan een kortere lijn.
3. Lijnen die elkaar deels overlappen.
 - Het punt dat precies tussen het eind van het eerste lijnstuk en het begin van het volgende lijnstuk zit (groen) wordt berekend. Vervolgens wordt het eind van het eerste lijnstuk naar dat gemiddelde punt verplaatst, evenals het begin van het volgende lijnstuk.



Figuur 3: Links het resultaat van de Canny edge detector, rechts het resultaat van de Hough-transformatie na correctie door middel van het connected-lines algoritme

7 Tracking

De taak van de trackingsmodule is te lokaliseren waar het reclamebord zich in het huidige frame bevindt. Hierbij wordt de aanname gemaakt dat de initialisatiemodule de correcte startcoördinaten heeft geretourneerd. De trackingsmodule maakt gebruik van informatie over waar het reclamebord zich in de vorige frames bevond. Om dit te verwezenlijken is de trackingsmodule opgesplitst in een aantal onderdelen.

7.1 Voorspeller

Omdat het een voetbalwedstrijd betreft zijn de camerabewegingen zeer vloeiend. Dit betekent dat de bewegingen in het beeld goed te voorspellen is. De voorspeller houdt een geschiedenis bij van drie frames waar de hoekpuntcoördinaten van het reclamebord in worden opgeslagen. De eerste afgeleide (de verplaatsing) wordt als schatter bepaald voor de nieuwe lokatie. Er is gekozen om dit voor elk individueel coördinaat te berekenen omdat de coördinaten niet met precies dezelfde hoeveelheid en richting verplaatsen. Dit komt door de volgende eigenschappen:

1. *Perspectievische vervorming*

Als een reclamebord bijvoorbeeld links in beeld staat is het hoogteverschil tussen de linker boven- en onderhoek kleiner dan dit van de rechter boven- en onderhoek. Dit komt doordat de afstand van de camera tot de linker rand in dit geval groter is dan de afstand tot de rechter rand.

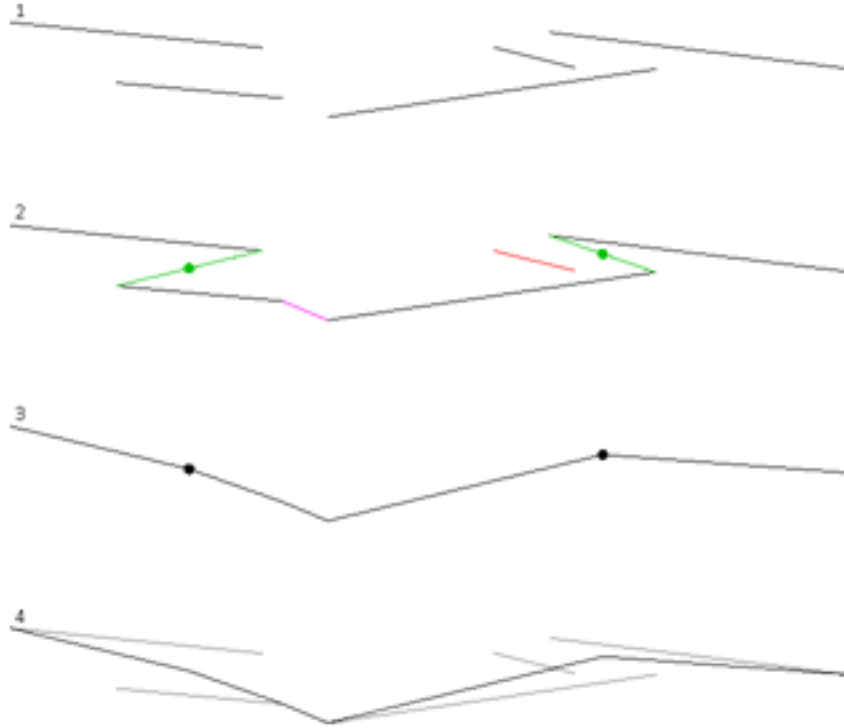
2. *Shearing*

Als een reclamebord in het midden van de camera staat zijn de boven en onderlijn horizontaal, op de lensvervorming na. Als ditzelfde reclamebord links of rechts in beeld staat lopen deze lijnen schuin.

3. *Zooming*

Bij het inzoomen worden coördinaten uit elkaar getrokken, bij het uitzoomen lopen coördinaten naar elkaar toe.

De eerste onbewerkte schatting wordt als volgt bepaald. Voor iedere coördinaat afzonderlijk wordt de gemiddelde verplaatsing in de frames uit de geschiedenis berekend. Deze verplaatsing wordt opgeteld bij de desbetreffende coördinaat.



Figuur 4: De werking van het *smooth connected lines* algoritme.

Doordat we met een discreet beeld te maken hebben en de waarden worden afgerond, komt het voor dat deze schatting een (paar) uitschieter(s) bevat. Om de invloed van de uitschieters te beperken worden er een aantal constraints toegepast.

7.1.1 Constraints

Het elimineren van uitschieters is een belangrijke taak in de trackingsmodule. Doordat er een schatting gemaakt moet worden welke nog moet worden verfijnd door de hoekpunten detector is het mogelijk een aantal constraints toe te passen.

Bij bestudering van de content blijkt dat de linker en rechter bordlijn altijd nagenoeg verticaal zijn. Dit betekent dat de x waarde van bovenliggende hoekpunten niet verschilt. Een uitschieter kan hier een niet verticale lijn van maken wat een ongewenst resultaat geeft. Dit wordt gecorrigeerd door de x waarden van bovenliggende punten te middelen en beide punten hier aan gelijk te stellen. Zie hiervoor vergelijking 6, 7, 8 en 9 Hierdoor zijn uitschieters in de x richting geëlimineerd.

$$topleft'_x \leftarrow \frac{topleft_x + bottomleft_x}{2} \quad (6)$$

$$bottomleft'_x \leftarrow \frac{topleft_x + bottomleft_x}{2} \quad (7)$$

$$topright'_x \leftarrow \frac{topright_x + bottomright_x}{2} \quad (8)$$

$$bottomright'_x \leftarrow \frac{topright_x + bottomright_x}{2} \quad (9)$$

De camera staat op een grote afstand van het reclamebord. Hierdoor is het verschil tussen de afstand van de camera tot de linker bordrand en van de camera tot de rechter bordrand minimaal, waardoor er slechts een zeer kleine perspectivische vervorming optreedt. Om uitschieters in de y richting te beperken worden de horizontale bordlijnen parallel gesteld.

Dit gebeurt op de volgende manier: Het hoogteverschil van de hoekpunten van de linker- en rechterbordrand wordt gemiddeld. Deze nieuwe hoogte wordt gecentreerd rond het oude punt. Zie vergelijking 10 t/m 18

$$\Delta y1 \leftarrow bottomleft_y - topleft_y \quad (10)$$

$$\Delta y2 \leftarrow bottomright_y - topright_y \quad (11)$$

$$\Delta yAverage(\Delta y1 + \Delta y2)/2 \quad (12)$$

$$y1Shift \leftarrow (\Delta y1 - \Delta yAverage)/2 \quad (13)$$

$$y2Shift \leftarrow (\Delta y2 - \Delta yAverage)/2 \quad (14)$$

$$topleft_y \leftarrow topleft_y + y1Shift \quad (15)$$

$$bottomleft_y \leftarrow bottomleft_y - y1Shift \quad (16)$$

$$topright_y \leftarrow topright_y + y2Shift \quad (17)$$

$$bottomright_y \leftarrow bottomright_y - y2Shift \quad (18)$$

7.1.2 Multi-resolutie

Nadat de voorspeller een schatting heeft gemaakt van de lokatie van het reclamebord moet de exacte lokatie worden bepaald. De pixelwaardes van het reclamebord uit het vorige frame zijn bekend en worden opgeslagen in een plaatje. Vervolgens wordt dit plaatje vergeleken met een aantal samples die zich op verschillende plekken op en rond het geschatte punt bevinden. De lokatie van het sample welke het minste verschilt (zie 19 en 20) met het plaatje wordt gezien als beste match.

A = vorig sample van $m \times n$ pixels.

S = set van (9) nieuwe samples (ieder eveneens $m \times n$ pixels).

$$A' = X \in S \frac{\operatorname{argmin} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{c \in \{R,G,B\}} |A_{i,j,c} - X_{i,j,c}|}{m \times n \times 3} \quad (19)$$

In deze vergelijking is $m \times n \times 3$ constant ($3mn$) voor ieder sample X uit S , dus kan deze vergelijking geschreven worden als:

$$A' = X \in S \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{c \in \{R,G,B\}} |A_{i,j,c} - X_{i,j,c}| \quad (20)$$

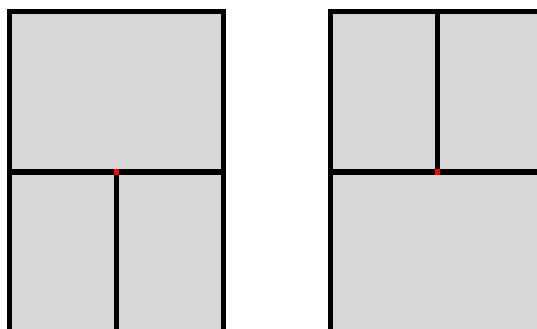
Er is gekozen voor een multi-resolutie aanpak omdat dit de performance aanzienlijk verhoogt. Het multi-resolutie systeem werkt als volgt:

Het frame wordt eerst verkleind met een factor 8. De beste match wordt gezocht zoals hierboven beschreven. Vervolgens wordt het frame met een factor 4 verkleind en wordt op de plek van het vorige resultaat een beste match gezocht. Dit proces wordt herhaald t/m factor 1.

De afstand van de omliggende coördinaten tot het geschatte punt is in het verkleinde frame 1 px. In het originele frame komt dit neer op de verkleinfactor aan pixels. De eerste iteratie wordt er dus gezocht op 8 pixels vanaf de geschatte coördinaat. Bij het kiezen van een startfactor van 8 is rekening gehouden met de maximale verplaatsing tussen twee frames. Dit komt voor bij de snelste camera beweging in de wedstrijd. Merkop dat de startfactor ook afhankelijk is van de resolutie. Bij een hogere resolutie moet er een hogere startfactor worden gekozen.

7.2 Hoeklocalisatie op subpixelniveau

Wanneer de lokatie van de hoekpunten geschat is, wordt deze verfijnd tot op subpixelniveau. Hiervoor wordt er gebruik gemaakt van een raster dat opgedeeld is in drie vlakken (zie Figuur 5). Voor ieder vlak wordt de gemiddelde *RGB*-waarde berekend. Dit raster wordt in een $n \times n$ zoekomgeving, gecentreerd op de verwachte locatie van het hoekpunt, over de originele afbeelding gehaald. Het punt waar de gemiddelden van de drie vlakken het meest van elkaar verschillen wordt gezien als het precieze hoekpunt. Dit gebeurt tot op een halve pixel nauwkeurig. Het raster wordt , zodat zijn horizontale as gelijk staat met de boven- en onder rand van het reclamebord.



Figuur 5: De rasters die gebruikt worden voor de localisatie van de hoekpunten. Links het raster voor de bovenste hoekpunten en rechts het raster voor de onderste hoekpunten.

8 Vervanging

Wanneer de coördinaten gevonden zijn kan er een andere bitmap in het videoframe geplaatst worden. Allereerst moet de bitmap gestreched worden. Vervolgens moet er rekening gehouden worden met lichtintensiteit en tot slot is het van belang dat een reclamebord nooit over een voetballer of over de bal heen geplaatst wordt.

8.1 Shearing en interpolatie

De vier hoekpunten van de bitmap komen overeen met de vier coördinaten in het frame. Echter is de vierhoek in het frame geen rechthoek meer door het perspectief. Een projectieve transformatie zou een geschikte oplossing zijn voor dit probleem. Echter blijkt het voldoende te zijn de

bitmap te herschalen en te shearen. Door middel van bilineaire interpolatie wordt een smooth resultaat behaald. De randen van het nieuwe reclamebord worden tijdens de transformaties gesmooth. Wanneer er een pixel aan de rand van de bitmap opgehaald wordt, wordt de nieuwe pixelwaarde berekend door een gewogen gemiddelde van de originele pixelwaarde in het frame en de pixelwaarde van de bitmap te nemen.

9 Resultaten

Het systeem bestaat feitelijk uit drie belangrijke onderdelen waarvan de eerste twee elkaar gedeeltelijk overlappen. De lokalisatie, de tracking en de vervanging.

9.1 Lokalisatie

Bij de initiële lokalisatie wordt gezocht naar de onderste rand van de reclameborden. Het resultaat is een lijst coördinaten die die rand beschrijven (Figuur 3). Door langs deze coördinaten te lopen is het mogelijk de scheiding tussen twee reclameborden te bepalen. Wanneer deze scheidingen gevonden zijn is het mogelijk de bovenrand van de reclameborden te vinden. Deze laatste twee stappen zijn nog niet geïmplementeerd, maar er zijn verschillende methoden waarmee dit gedaan zou kunnen worden.

9.2 Tracking

Tijdens de tracking wordt op verschillende schalen gezocht naar de plek in een nieuw frame waar het reclamebord van het vorige frame zich bevindt. Hiervoor worden de absolute verschillen per pixel berekend. In Figuur 6 is duidelijk te zien dat er steeds één van de negen mogelijkheden het beste matcht. Uiteindelijk worden de gevonden coördinaten gecorrigeerd, waardoor er vier nieuwe coördinaten ontstaan die de hoekpunten van de reclameborden aangeven. Dit zijn coördinaten op subpixel niveau (Figuur 8). In Figuur 7 is het resultaat van de tracking te zien. In deze figuur is zowel frame 1 weergegeven als frame 200, waardoor te zien is dat de tracker na 200 frames nog steeds accuraat is.

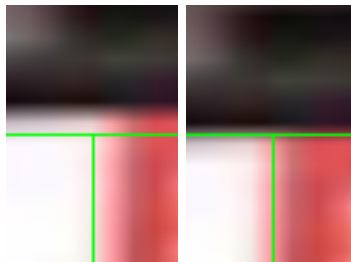


Figuur 6: Boven: In deze figuur zijn de 9 samples, die uit een frame zijn geëxtraheerd, afgebeeld. Onder: De absolute differences van de samples en het sample uit het vorige frame. Het sample links-midden is in dit geval de beste match.

0.0297	0.0330	0.0366
0.0083	0.0132	0.0192
0.0164	0.0197	0.0242



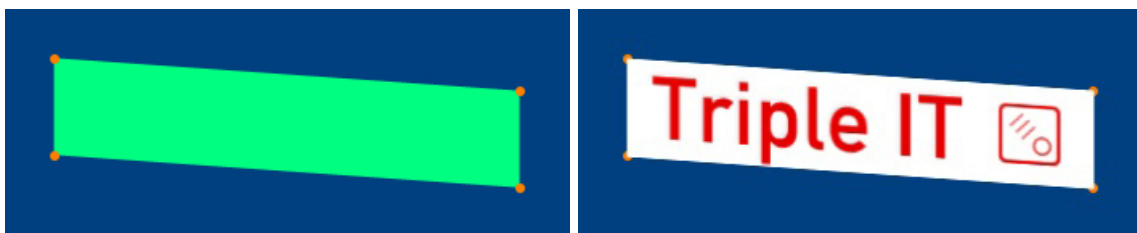
Figuur 7: Resultaat van tracking. Links frame 1 (initialisatie), rechts frame 200.



Figuur 8: Ingezoomed op een masker van 8 bij 12 pixels. Links het resultaat van de schatting, rechts de gecorrigeerde schatting.

9.3 Vervanging

Wanneer de nieuwe coördinaten gevonden zijn kan de nieuwe bitmap erin geplaatst worden. Een demonstratie van het algoritme is weergegeven in Figuur 9. In de werkelijke frames wordt het uiteindelijke reclamebord vele malen kleiner (Figuur 10). Doordat er gebruik gemaakt wordt van subpixels in plaats van pixels wordt zogenaamde *jittering*, het 'stuiteren' van een virtueel reclamebord, voorkomen.



Figuur 9: Links een voorbeeld plaatje met daarin een parallelogram. Rechts hetzelfde plaatjes, maar hier is het parallelogram door vervangen door het virtuele reclamebord door middel van het algoritme dat ook voor de video-stream gebruikt wordt.



Figuur 10: Het resultaat; Het ING-logo is vervangen door het logo van Triple IT.

10 Discussie

10.1 Voorspeller

Er is geëxperimenteerd met de voorspeller om de tweede afgeleide te nemen. Dit zorgte te vaak voor een te grote sprong. Dit kan komen doordat er te weinig frames bijgehouden worden in de geschiedenis.

Een mooiere oplossing zou zijn om meer frames bij te houden en een functie proberen te fitten om zo de coördinaten beter te schatten.

10.2 Multi-resolutie

De uiteindelijke bedoeling is om de tracker op meerdere cameras te draaien. Op het moment dat een camera niet uitzendt zal de cameraman geen rekening houden met vloeiende bewegingen. Dit zou er voor kunnen zorgen dat de beweging meer dan 8 pixels per frame is waardoor de tracking faalt. Op dit moment zou een herinitialisatie kunnen worden toegepast.

11 Toekomstwerk

Het systeem wat in dit verslag beschreven staat is in zeer korte tijd ontwikkeld en heeft als doel een demo te vormen. Hierdoor is het enkel een basis, waarin vele uitbreidingen denkbaar zijn:

11.1 Reclamebord uit beeld

Het systeem kan enkel een reclamebord tracken wat in beeld blijft. Om een reclamebord wat (gedeeltelijk) uit beeld gaat te tracken zijn er twee uitbreidingen mogelijk:

- Als een reclamebord uit beeld verdwijnt de tracking van dat reclamebord opgeven. Op het moment dat er een nieuw reclamebord in beeld komt de initialisatie stap opnieuw uitvoeren op dat reclamebord.
- Als er een reclamebord voor een deel verdwijnt en weer terugkomt kan er alleen focus gelegd worden op de hoekpunten die nog in beeld zijn. De implementatie van dit verslag ondersteund dit nog niet.
- Omdat de camera zich op een punt bevindt is het mogelijk de gehele scene bij te houden. De gehele scene kan dan op een kubus of bol worden geprojecteerd [3]. Hierna kunnen de camerabeweging en zoom worden geschat. Dit is dan voldoende om te tracken omdat er informatie van buiten het beeld beschikbaar is. Het is verstandig, in verband met meetfouten en afrondingen, op reguliere basis een calibratie uit te voeren.

11.2 Lichtintensiteit en schaduwen

De lichtintensiteit kan nogal verschillen per frame. Niet alleen doordat de lichtintensiteit in de werkelijke wereld verschilt, zoals dat gebeurt wanneer er een wolk voorbij komt, maar voornamelijk door de automatische whitebalance in de camera. Wanneer er meer donkere vlakken in beeld komen, zal de camera zich bijstellen en het totale beeld wat lichter maken. Dit heeft als gevolg dat de werkelijke reclameborden lichter worden opgenomen door de camera. Deze verandering moet in het nieuwe reclamebord meegenomen worden. In de implementatie die beschreven wordt in dit verslag wordt hier nog geen rekening mee gehouden. Een mogelijke oplossing is om te kijken naar de gemiddelde pixelwaarde van alle pixels in het werkelijke reclamebord en de verschillen in dit gemiddelde door te voeren in het virtuele reclamebord.

Wanneer er slagschaduw voorkomt, zal er op een lokaler niveau gekeken moeten worden naar de lichtintensiteit per pixel. Dit vergt een wat ingewikkelder algoritme waar nog onderzoek naar gedaan moet worden.

11.2.1 Background estimation

Uiteraard is het niet de bedoeling dat het virtuele reclamebord over voetballers, lijnrechters, scheidsrechters, de bal etcetera geplaatst wordt. Hiervoor zal een background estimation geïmplementeerd moeten worden. Met behulp van de background estimation kan er voor iedere pixel een schatting gemaakt worden of de pixel wel of niet tot de achtergrond (het reclamebord) behoort en of de pixel dus wel of niet vervangen moet worden door een pixel uit het virtuele reclamebord. Ook in dit geval is het weer handig om gebruik te maken van de kennis over het domein. Namelijk dat er altijd een groep pixels zal zijn die samen niet tot de achtergrond behoren. Er zijn verschillende technieken denkbaar om dit soort constraints toe te passen. Een combinatie van erosie en dilatie is een mogelijkheid, of het toepassen van *connected components*. Vervolgens worden de pixels waar zich bijvoorbeeld een voetballer bevindt niet vervangen. Het is eenvoudig om langs de rand antialiasing toe te passen, waardoor het geheel er natuurlijker uit zal zien.

11.3 Geheel product

Als het systeem robuust genoeg is om reclameborden te tracken gedurende een hele wedstrijd kan er een volledig product van worden gemaakt.

Hier kan gedacht worden aan een 'drag and drop' userinterface waar de gebruiker op een gebruiksvriendelijke manier aan kan geven welke reclameborden moeten vervangen worden. De uitzendingenregio's zouden kunnen worden weergegeven met behulp van een interactieve kaart waardoor er per regio een ander reclamebord kan worden geselecteerd.

12 Conclusie

Het is slim om gebruik te maken van de geometrie binnen een videostream van een voetbalwedstrijd. Hierdoor is het mogelijk om met een hogere accuracy te 'schatten' waar de borden staan. Het is gelukt om een reclamebord te vervangen op een manier die er zeer natuurlijk uitziet, echter is hier nog geen rekening gehouden met lichtintensiteit en de situatie dat een object zich voor het bord bevindt. Deze laatste punten zijn interessant genoeg om het project te vervolgen en de hier beschreven methode te verbeteren.

Lijst van figuren

1	Iedere pixel is geclassificeerd als zijnde graspixel (<i>hoog</i>), danwel niet graspixel (<i>laag</i>).	5
2	Correctie van de <i>false positives</i> (links) en de <i>false negatives</i> (rechts) door middel van erosie en dilatie.	6
3	Links het resultaat van de Canny edge detector, rechts het resultaat van de Hough-transformatie na correctie door middel van het connected-lines algoritme	7
4	De werking van het <i>smooth connected lines</i> algoritme.	8
5	De rasters die gebruikt worden voor de localisatie van de hoekpunten. Links het raster voor de bovenste hoekpunten en rechts het raster voor de onderste hoekpunten.	10
6	Boven: In deze figuur zijn de 9 samples, die uit een frame zijn geëxtraheerd, afgebeeld. Onder: De absolute differences van de samples en het sample uit het vorige frame. Het sample links-midden is in dit geval de beste match.	11
7	Resultaat van tracking. Links frame 1 (initialisatie), rechts frame 200.	12
8	Ingezoomd op een masker van 8 bij 12 pixels. Links het resultaat van de schatting, rechts de gecorrigeerde schatting.	12
9	Links een voorbeeld plaatje met daarin een parallelogram. Rechts hetzelfde plaatjes, maar hier is het parallelogram door vervangen door het virtuele reclamebord door middel van het algoritme dat ook voor de video-stream gebruikt wordt. . .	12
10	Het resultaat; Het ING-logo is vervangen door het logo van Triple IT.	13

Referenties

- [1] G. Medioni & G. Guy & H. Rom & A. Franois. Real-time billboard substitution in a video stream. 1998.
- [2] Frank Aldershoff & Theo Gevers. Visual tracking and localisation of billboards in streamed soccer matches. 2004.
- [3] Marc Pollefeys & Sudipta N. Sinha & Seon Joo Kim. High-resolution multiscale panoramic mosaics from pan-tilt-zoom cameras. 2004.