

# Labo beeldverwerking

Kristof Teelen & David Van Hamme

2010-11

## Inleiding

De labo's beeldverwerking omvatten 2 componenten: labosessies en een groepsproject.

### Labosessies

De labosessies introduceren verschillende typische beeldverwerkingstechnieken, zoals

- morfologie
- lijn- en randdetectie
- segmentatie door thresholding
- regiogebaseerde segmentatie
- representatie en descriptor
- herkenning en classificatie
- zoekalgoritmes

Alle sessies zijn gebaseerd op 'Digital Image Processing' van Gonzalez en Woods, uitgegeven door Prentice Hall. Deze opgaven zullen op *Dokeos* geplaatst worden.

De uitwerking van de opdrachten uit de labosessies wordt verwacht als Matlab m-files. In de gegeven labo-opdrachten worden er regelmatig vragen gesteld. Probeer deze te beantwoorden als commentaar in de m-files die uiteindelijk ingediend worden. Zorg ervoor dat alle functies aangesproken kunnen worden via 1 duidelijk script. Voeg dus ook de beelden toe!

Deze labosessies moeten per projectgroep opgelost worden, de verdeling onder elkaar is jullie eigen verantwoordelijkheid.

## Groepsprojecten

Tijdens de groepsprojecten zoeken jullie naar een oplossing voor een praktische beeldverwerkingprobleem, zoals

1. Real-time nummerplatherkenning (4)
2. Toegangscontrole op basis van gezichtsherkenning (4)
3. Slijtage-analyse voor koolstoflatten (5)
4. Automatische voetbalanalyse (4)
5. Tram collision avoidance (4)
6. Real-time visual hull-computation (5)

7. Analyse van 3D puntwolken (5)
8. Analyse van loopbeweging (4)
9. 3D trajectreconstructie (pingpong) (5)
10. Surveillance en veiligheid (6)
11. Snapshotgebaseerde kalibratie (4)

Verderop wordt een korte beschrijving van alle projectonderwerpen, en de gestelde problemen gegeven.

Schrijf per groep zelf een projectbeschrijving op basis van deze korte indicatie, aan de hand van een template beschikbaar op Dokeos. In het projectvoorstel moet zowel een korte beschrijving van het totaalproject, als een korte beschrijving van de gebruikte methode/algoritmes/... voor elk onderwerp opgemaakt worden. Elk groepsproject wordt onderverdeeld in verschillende werkpakketten (= WP).

Het is belangrijk dat in de beschrijving voor elk WP een indicatie wordt gegeven van de gebruikte oplossingsmethodes en eventuele alternatieven, waarbij ook duidelijk aangegeven wordt waarom welke methode verkozen wordt boven andere. De beschrijving vermeldt de specificaties voor de software (input, output en werking van de code voor elk WP), en door onderlinge afspraken moeten de verschillende modules dan ook naadloos op elkaar aansluiten.

Aan elk WP wordt de naam van 1 student gekoppeld, die de nodige code schrijft voor de oplossing van zijn deelprobleem. Alle groepsleden zijn samen verantwoordelijk voor de integratie van alle onderdelen tot een werkend geheel.

Voor elk WP wordt code geschreven die op Dokeos ingeleverd moet worden met de overkoepelende naam *projectnaam\_WP#.m* voor de aangegeven deadline. De documentatie is erg belangrijk, zorg ervoor dat voor elke file nogmaals duidelijk inputs, outputs en werking beschreven moeten worden. Elke groep voegt dan de verschillende werkpakketten samen in 1 overkoepelend programma voor elk project.

## Quotatie

### Beoordelingsmomenten

- De definitieve versie van het projectvoorstel wordt beoordeeld per groep (Hierna volgt kort feedback over de voorgestelde oplossingen, methodes en werkdruk).
- De kwaliteit van de ingeleverde labo's wordt beoordeeld met quotatie per groep.
- De groepsprojecten moeten individueel verdedigd worden a.d.h.v een presentatie tijdens de examenperiode. Een verslag is niet nodig, maar de code die je inlevert moet grondig gedocumenteerd zijn: de nodige inputs en de resulterende outputs moeten tot in detail beschreven worden, met daarnaast ook een korte beschrijving van de werking van het geïmplementeerde algoritme.

### In te dienen documenten

**labosessies:** 1 containerfile met 1 matlabscrip *projectnaam\_laboBV.m* dat alle matlabscripts per labo (*laboBV\_#.m*) oproept, en waarin alle nodige files gevoegd zijn.

Deadline: 2 dagen voor het theorie-examen.

**groepsproject:** Dien telkens 1 containerfile in per groep met daarin alle nodige files.

- Een projectvoorstel per groep;
- De goed gedocumenteerde matlabcode voor elk afzonderlijk werkpakket, (*projectnaam\_WP#.m*), samen met een overkoepelende mainfunctie die de verschillende onderdelen integreert (*projectnaam.m*);
- Een presentatie per uitgevoerd werkpakket. Zorg ervoor dat ook de uiteindelijke presentatie op de dropbox in Dokeos terechtkomt.

Deadline: De deadlines voor elk in te leveren item worden tijdig meegedeeld via de kalender op Dokeos. Geef significante namen aan elk item.

# 1 Real-time nummerplaatherkenning

De situatie op de hogeschoolparking bewijst elke dag dat er een hoge nood is aan geautomatiseerde systemen voor toegangscontrole. In dit project willen we in real-time de nummerplaat en het type van de wagen analyseren en vergelijken met entries in een database. Enkel geprivilegieerde gebruikers mogen het onderbezette maar afgezette stuk parking gebruiken.

**Programmeeromgeving & beelden:** Ontwikkel een visiesysteem dat in real-time de video-opnames aan de ingang van een parking (vertragende of stilstaande wagen) analyseert en de nummerplaat en type wagen detecteert. Voor de real-time opnames kan gebruik gemaakt worden van IP/FireWire camera's, en moet een beeldverwerking algoritme geprogrammeerd worden in C/C++ (de OpenCV-library). Neem in een eerste fase de nodige sequenties op om de verschillende deelalgoritmes te testen. Indien mogelijk, ga dan later over naar een real-time applicatie voor zo veel mogelijk onderdelen van het totale programma.

**Werkpakketten:** Om dit systeem te realiseren moeten enkele modules ontwikkeld worden die bestaan uit verschillende werkpakketten:

- WP 1 a + b: Bewegingsdetectie: Zet de real-time videostroom op en detecteer beweging. Gebruik verschillende voorgrond-achtergrond-segmentatietechnieken (ViBe, MoG, first order filter, etc.) om de beweging in een beeld te detecteren, en geef deze door als region-of-interest voor de volgende WPen. Bespreek vooral ook de robuustheid van de algoritmes tegen belichtingsvariaties doorheen de dag. Implementeer 2 verschillende algoritmes, en vergelijk de resultaten met elkaar. Zoek een methode om dit soort algoritmes te vergelijken op in de literatuur, en voer de evaluatie uit voor beide algoritmes.
- WP 2: Detectie van nummerplaat: verifieer voor elke frame of er een nummerplaat in het beeld voorkomt, op basis van typische kleurpatronen (of andere features). Geef ook aan hoe lang deze nummerplaat in de videostroom zichtbaar is (in welke opeenvolgende frames?).
- WP 3: Herkenning van cijferlettercombinatie: Gebruik OCR-technieken (zoals Google tesseract) om de tekens op de nummerplaat te analyseren. Verhoog de robuustheid door de informatie uit meerdere opeenvolgende frames te gebruiken. Koppeling met database: stockeer de detecties in een database (MySQL). Zorg voor de nodige functionaliteit, verschillende soorten queries, etc.
- WP 4: Herkenning van merk/type wagen: Gebruik patroonherkenning (bv. matched filter) om de merkaanduiding terug te vinden en te interpreteren. Gebruik de locatie van de nummerplaat om regions of interest af te bakenen. Koppeling met database: stockeer de detecties in een database (MySQL). Zorg voor de nodige functionaliteit, verschillende soorten queries, etc.

**Testcase:** Het uiteindelijk doel is een real-time werking van het ontwikkelde systeem op de aangegeven locatie in verschillende weersomstandigheden en momenten van de dag.

## 2 Toegangscontrole op basis van gezichtsherkenning

We wensen een systeem te ontwikkelen dat ons zegt of alle studenten die verwacht worden in een les, ook werkelijk aanwezig zijn (ook als ze hun badge of studentenkaart vergeten). Daarom moet er een visiesysteem ontwikkeld worden dat in real-time elk gezicht bij toegang in het lokaal registreert, interpreteert en herkent.

**Programmeeromgeving & beelden:** Ontwikkel een visiesysteem dat in real-time de video-opnames aan de ingang van een lokaal (PE136) analyseert. Voor de real-time opnames kan gebruik gemaakt worden van IP/FireWire camera's, en moet een beeldverwerking algoritme geprogrammeerd worden in C/C++ (de OpenCV-library). Neem in een eerste fase de nodige sequenties op om de verschillende deelalgoritmes te testen. Indien mogelijk, ga dan later over naar een real-time applicatie voor zo veel mogelijk onderdelen van het totale programma.

**Werkpakketten:** Om dit systeem te realiseren moeten enkele modules ontwikkeld worden die bestaan uit verschillende werkpakketten:

- WP 1: Bewegingsdetectie: Zet de real-time videostroom op en detecteer beweging. Gebruik verschillende voorgrond-achtergrond-segmentatietechnieken (ViBe, MoG, first order filter, etc.) om de beweging in een beeld te detecteren, en geef deze door als region-of-interest voor de volgende WP'en. Bespreek vooral ook de robuustheid van de algoritmes tegen belichtingsvariaties (vooral ook aan/uitschakelen licht).
- WP 2: Gezichtsdetectie: verifieer voor elke frame of er een gezicht in het beeld voorkomt, op basis van de Viola-Jones face detector. Geef ook aan hoe lang dit gezicht in de videostroom zichtbaar is (in welke opeenvolgende frames?), en gebruik dit om de detectie robuuster te maken.
- WP 3: Gezichtsdetectie: verifieer voor elke frame of er een gezicht in het beeld voorkomt, op basis van de vorm van het voorgrondobject. Elke mens wordt gekarakteriseerd door een typische omega-vorm van hoofd en schouders. Geef ook aan hoe lang deze vorm in de videostroom zichtbaar is (in welke opeenvolgende frames?), en gebruik dit om de detectie robuuster te maken.
- WP 4: Gezichtsherkenning: Gebruik technieken (zoals eigenfaces of PEM) om de gezichten op de nummerplaat te analyseren. Verhoog de robuustheid door de informatie uit meerdere opeenvolgende frames te gebruiken. Koppeling met database: stockeer de detecties in een database (MySQL). Zorg voor de nodige functionaliteit, verschillende soorten queries, etc.

**Testcase:** Het uiteindelijke doel is een real-time werking van het ontwikkelde systeem op de aangegeven locatie in verschillende omstandigheden en momenten van de dag.

### 3 Slijtage-analyse voor koolstoflatten

Tijdens deze thesis wordt er in samenwerking met De Lijn een controlesysteem voor de energievoorziening van trams ontwikkeld op basis van computervisietechnieken. Dit controlesysteem moet een deel van het huidige manuele inspectiewerk van de railinfrastructuur overnemen en ons in staat stellen om preventief problemen op te sporen zodat de veiligheid en de exploitatie gewaarborgd kunnen blijven.

De slijtage van de koolstoflatten op de pantograaf wordt veroorzaakt door de beweging van de rijdraad over de koolstoflaag op de pantograaf. De Lijn wil elke tram meermaals per dag controleren door bij het buitenrijden van de stelplaats een visiesysteem controles te laten doen. Een eerste camera is gericht op de pantograaf zodat de slijtage geanalyseerd kan worden. Elke controle moet gekoppeld worden aan het identificatienummer van de tram (zichtbaar in de tweede camera, zodat men de uiteindelijke resultaten kan stockeren in een database, en op een gebruiksvriendelijke wijze aanbieden aan de kwaliteitsverantwoordelijke.

**Programmeeromgeving & beelden:** Ontwikkel een visiesysteem dat in real-time de video-opnames van de koolstoflat en de tram analyseert. Voor de real-time opnames kan gebruik gemaakt worden van IP/FireWire camera's, en moet een beeldverwerking algoritme geprogrammeerd worden in C/C++ (de OpenCV-library). Ontwikkel een eerste prototype, en test de verschillende deelalgoritmes uit op de beschikbare videosequenties.

**Werkpakketten:** Om dit systeem te realiseren moeten enkele modules ontwikkeld worden die bestaan uit verschillende werkpakketten:

- WP 1: Bewegingsdetectie: Detecteer beweging in de beschikbare videostromen. Gebruik verschillende voorgrond-achtergrond-segmentatietechnieken (ViBe, MoG, first order filter, etc.) om de beweging in een beeld te detecteren, en geef deze door als region-of-interest voor de volgende WPen. Bespreek vooral ook de robuustheid van de algoritmes tegen belichtingsvariaties (vooral zon/bewolking/verschillende momenten van de dag).
- WP 2: Tramdetectie: verifieer voor elke frame of het bewegend object een tram is. Gebruik patroonherkenningstechnieken (zoals template matching) om de tram te identificeren, en te volgen doorheen de opeenvolgende frames (door het opstellen van motion vectors). Geef ook aan hoe lang 1 tram in de videostroom zichtbaar is (in welke opeenvolgende frames?), en probeer een region-of-interest af te bakenen voor het identificatienummer van de tram.
- WP 3: Herkenning van cijferlettercombinatie: Gebruik OCR-technieken (zoals Google tesseract) om de tekens op de nummerplaat te analyseren. Verhoog de robuustheid door de informatie uit meerdere opeenvolgende frames te gebruiken, en te vergelijken met de database waarin de bestaande tram-IDs beschikbaar zijn. Zorg voor een koppeling met de database (MySQL), en stockeer de detecties met timestamp en eventueel andere nuttige informatie. Zorg voor de nodige functionaliteit, verschillende soorten queries, etc.
- WP 4: Pantograaflocalisatie: Lokaliseer de pantograaf in de beschikbare beelden. Gebruik patroonherkenningstechnieken (zoals template matching) om verschillende herkenbare punten (hoorns, intersecties, etc.) op de pantograaf te detecteren en herkennen. Combineer de detectiescores van de verschillende punten op een intelligente manier om de overall detectie robuuster te maken.
- WP 5: Koolstoflat-analyse: Detecteer en analyseer het profiel van de koolstoflat (op basis van aangepaste randdetectiealgoritmes. Geef aan of er abnormale patronen teruggevonden kunnen worden in dit profiel, en ontwerp een scoresysteem dat automatisch de bruikbaarheid van het profiel aangeeft.

**Testcase:** Het uiteindelijk doel is een eerste prototype visiesysteem dat voldoet aan de gevraagde vereisten voor de gegeven dataset.

## 4 Automatische voetbalanalyse

Er is veel interesse van voetbalclubs naar een visiesysteem voor de automatische beoordeling van de prestaties van voetbalspelers en de kwaliteit ervan tijdens een voetbalmatch. Momenteel zijn er zulke systemen op de markt, maar die zijn erg duur en vereisen nog veel manuele input. Het doel van dit project is de uitbreiding van een systeem dat dit evaluatieproces volledig automatiseert.

Er is tijdens voorbije jaren al software geschreven voor de identificatie van verschillende elementen in de spelsituatie: de spelers van beide ploegen en de scheidsrechter. We slagen er nu grotendeels in om de positie van de spelers vast te leggen op elk tijdstip. Dit is van belang bij een analyse van de veldbezetting, bv. de onderlinge afstand tussen spelers in spelsituaties, zoals stilstaande fases of buitenspel.

Er blijven nog enkele problemen die nog niet opgelost zijn. Nu willen we ook de bal en de lijnen op het speelveld detecteren op basis van kleur- en textuursegmentatie, zodat we de beschrijving van spelersbewegingen kunnen linken aan het baltraject. Daarnaast zijn er nog steeds oclusiesituaties waarbij het zicht op spelers geblokkeerd wordt in een camerastandpunt, daarom willen we nu ook hun rugnummers gaan detecteren zodat we nog meer trajecten kunnen linken.

Het onderzoek was tot nu toe toegespitst op zaalvoetbal, en zou nu ook uitgebreid moeten worden naar buitenvoetbal. Dit vereist bijkomend onderzoek om te evalueren of de gebruikte basistechnieken nog altijd voldoen in de nieuwe situatie, bv. de belichting in een zaal is veel constanter dan die buiten.

**Programmeeromgeving & beelden:** Ontwikkel een visiesysteem dat de video-opnames van een voetbalwedstrijd analyseert. We verwachten dat de beeldverwerkingalgoritmes geprogrammeerd worden in Matlab of C/C++ (de OpenCV-library). Test de ontwikkelde algoritmes op de beschikbare beeldsequenties.

**Werkpakketten:** Om dit systeem te realiseren moeten enkele modules ontwikkeld worden die bestaan uit verschillende werkpakketten:

- WP 1: Bewegingsdetectie: Detecteer de beweging van de speler in een buitenvoetbalsituatie. Gebruik verschillende voorgrond-achtergrond-segmentatietechnieken (ViBe, MoG, first order filter, etc.) om de beweging in een beeld te detecteren, en geef deze door als region-of-interest voor de volgende WPen. Bespreek vooral ook de robuustheid van de algoritmes tegen belichtingsvariaties (vooral de variatie van het buitenlicht).
- WP 2: Bewegingsdetectie: Detecteer de beweging van de speler in een buitenvoetbalsituatie. Gebruik textuur- en kleursegmentatietechnieken om de spelers te onderscheiden van de achtergrond in elk beeld en geef deze door als region-of-interest voor de volgende WPen. Bespreek vooral ook de robuustheid van deze algoritmes tegen belichtingsvariaties (vooral de variatie van het buitenlicht).
- WP 3: Implementeer een methode om de spelertrajecten te construeren. Zoek features (zoals kleur en grootte) in de voorgrondregio's en link de opeenvolgende posities van spelers aan elkaar op basis daarvan.
- WP 4: Baldetectie: Detecteer de positie van de bal in de beeldruimte op elk moment. Gebruik technieken zoals een matched filter of een particle filter om de bal in de opeenvolgende beelden te detecteren, en stel het traject van de bal doorheen de match op.
- WP 5: Rugnummerdetectie: Detecteer en herken zoveel mogelijk rugnummers van spelers doorheen de videosequentie. Kies in een eerste fase een redelijk aantal voorbeelden uit de sequentie om de ontwikkelde algoritmes op uit te proberen. Gebruik later de regions-of-interest van eerdere werkpakketten om enkel die locaties waar zich spelers zouden kunnen bevinden om te analyseren. Pas verschillende correlatie of OCR- technieken toe om de rugnummers te analyseren, en vergelijk de resultaten.

**Testcase:** Het uiteindelijke doel is de ontwikkeling van algoritmes en de verificatie van hun nut op de beschikbare beeldsequenties.

## 5 Tram collision avoidance

Er gebeuren jaarlijks heel wat ongelukken met trams, omdat de remafstand vrij groot is, en er geen uitwijkmogelijkheden zijn. In dit project proberen we dat te vermijden door via computervisietechnieken een schatting proberen maken van de afstand tot andere trams, en tot andere voertuigen of weggebruikers die de sporen blokkeren.

Niet enkel de afstand vanuit de tram is belangrijk, ook de positie van de pantograaf en de leidraad hebben hun invloed op de voortstuwing van de tram, en dan vooral de slijtage. Een onderdeel van dit project is het detecteren van de positie van pantograaf en leidraad in videosequenties opgenomen vanop het dak van een tram.

**Programmeeromgeving & beelden:** Programmeer de algoritmes in Matlab of C/C++ (de OpenCV-library). Maak gebruik van de gegeven dataset aan beeldsequenties.

**Werkpakketten:** Om dit systeem te realiseren moeten enkele modules ontwikkeld worden die bestaan uit verschillende werkpakketten:

- WP 1: Detecteer de sporen voor de tram. Onderzoek technieken (zoals kortste pad) om de sporen te volgen van onderaan het beeld tot bovenaan. Dit vereist een criterium dat aanduidt of sporen lopen of niet. Gebruik de output van een beeldverwerkingoperatie (zoals Sobelfilters, textuurfilters) om een maat te genereren.
- WP 2: Herken de trams in de opeenvolgende frames door template matchingtechnieken, zoals bv. correlatie. Zoek de intelligentere state-of-the-art technieken op, implementeer enkele en vergelijk de resultaten op de gegeven dataset.
- WP 3: Bereken de afstand tot de voorgaande tram op basis van visuele metrology. Veronderstel dat de afstand tussen de sporen gekend is, en relateer daaraan de afstand tot de vorige tram. Houd daarbij rekening met de perspectieffervorming, en gebruik een homografie om het grondoppervlak te rectificeren voor de afstandbepaling. Vergelijk dit resultaat met de terugprojectie op basis van de gegeven kalibratiematrices.
- WP 4: Detecteer in elk frame of een ander voertuig of weggebruiker zoals fietsers of automobilisten de tramsporen blokkeert. Dat kan door een abrupt einde van de sporen te detecteren tegen het blokkerend voertuig. Een tweede mogelijkheid die onderzocht moet worden is de detectie van alle auto's in de frames. Een eerste mogelijke methode is de detectie van vlakke texturen in een beeld. Maak gebruik van de parameters gegeven in een van de labo's om beelden te segmenteren volgens textuur. Onderzoek nog andere mogelijkheden uit de literatuur om een snelle textuursegmentatie uit te voeren.

**Testcase:** Het uiteindelijke doel is een correcte werking van het ontwikkelde systeem op de gegeven sequenties. Zorg voor een duidelijke evaluatie van de algoritmes.

## 6 Real-time visual hull-computation

In PE136 is er een multi-camerasysteem gemonteerd waarvan de zichtpunten elkaar overlappen. Dit laat toe om alle informatie te combineren en 3D-informatie over de scene te extraheren uit de verschillende 2D beelden uit gesynchroniseerde videostreamen. In dit project willen we de 3D-bewegingen van 1 persoon analyseren door het berekenen van een visuele omhullende. Daarnaast voeren we een rudimentaire bewegingsanalyse.

**Programmeeromgeving & beelden:** Ontwikkel een visiesysteem dat in real-time de video-opnames in PE136 verwerkt. Voor de real-time opnames kan gebruik gemaakt worden van IP/FireWire camera's, en moet een beeldverwerking algoritme geprogrammeerd worden in C/C++ (de OpenCV-library). Neem in een eerste fase de nodige sequenties op om de verschillende deelalgoritmes te testen. Indien mogelijk, ga dan later over naar een real-time applicatie voor zo veel mogelijk onderdelen van het totale programma.

**Werkpakketten:** Om dit systeem te realiseren moeten enkele modules ontwikkeld worden die bestaan uit verschillende werkpakketten:

- WP 1: Kalibratie: Zoek een geometrische beschrijving van de optica en de positionering van de camera's in deze ruimte. Multi-camerakalibratie is een moeilijk probleem, zoek dus eerst verschillende bestaande technieken op, en geef een beschrijving van voor- en nadelen. Geef indien mogelijk een bruikbare en werkbare implementatie.
- WP 2: Bewegingsdetectie: Detecteer de beweging van de proefpersoon. Gebruik verschillende voorgrond-achtergrond-segmentatietechnieken (ViBe, MoG, first order filter, etc.) om de beweging in een beeld te detecteren, en geef deze door als region-of-interest voor de volgende WPen. Bespreek vooral ook de robuustheid van de algoritmes tegen belichtingsvariaties (vooral aan/uitknippen van licht).
- WP 3: 2D visual hull: Bereken de omhullende van de proefpersoon op basis van de voorgrond-achtergrondsegmentatie. Zoek technieken om de ruwe vorm uit te vlakken, en een 'smoother' hull te berekenen (bv. door uitmiddeling of door een beschrijving door polynomen), en verifieer wat de voor- en nadelen van deze bewerkingen zijn.
- WP 4: 3D visual hull: Bereken de 3D omhullende voor de proefpersoon op basis van terugprojectie van de verschillende 2D hulls, en de kalibratiegegevens. Implementeer een voxel-gebaseerd algoritme. Vergelijk de resultaten met deze voor WP5.
- WP 5: 3D visual hull: Bereken de 3D omhullende voor de proefpersoon op basis van terugprojectie van de verschillende 2D hulls, en de kalibratiegegevens. Implementeer een algoritme op basis van intersectie van halfvlakken. Vergelijk de resultaten met deze voor WP4.
- WP 6: Bewegingsanalyse en visualisatie: Bereken de distance transform voor alle punten op de 3D omhullende (ten opzichte van zijn kern), en leidt hieruit een flexibel skelet af. Zorg voor een elegante representatie van zowel de visuele omhullende als het skelet in 3D (OpenGL of VTK), en analyseer deze vorm (bv. duid ledematen aan, etc.).

**Testcase:** Het uiteindelijke doel is een real-time werking van het ontwikkelde systeem op de aangegeven locatie in PE136 in verschillende omstandigheden.



## 7 Analyse van 3D puntenwolken

Een 3D ruimte kan tegenwoordig accuraat opgemeten worden met behulp van laserscanners. Een probleem blijft de interpretatie van de opgemeten data, dat is in vele gevallen een puntenwolk, waar elk punt gegeven wordt door een  $(x,y,z)$  coördinaat ten opzichte van een referentie (oorsprong van assenstelsel). Om deze metingen op een hoger niveau te kunnen analyseren is verdere processing vereist. In dit project onderzoeken we verschillende technieken om puntenwolken te analyseren.

Daarnaast zouden we ook willen onderzoeken of het mogelijk is om de 3D informatie van de puntenwolken te combineren met beelddata. Elk punt in de wolk wordt ook gekarakteriseerd aan de hand van een intensiteitwaarde die evenredig is met de reflectantie. We willen proberen onderzoeken of het mogelijk is om deze informatie te registreren met een kleurenbeeld, en aan elk punt in de wolk een kleur toe te kennen.

**Programmeeromgeving & beelden:** Ontwikkel algoritmes om de beschikbare 3D wolken te analyseren. We verwachten dat de beeldverwerkingalgoritmes geprogrammeerd worden in Matlab of C/C++ (de OpenCV-library). Test de ontwikkelde algoritmes op de beschikbare data.

- WP 1: Probeer een aantal algoritmes voor triangulatie uit. Zoek eerst enkele voorbeelden op in de literatuur, en implementeer een werkende oplossing.
- WP 2: Implementeer een algoritme dat zoekt naar de alpha-hull (of alpha shape) van een puntwolk.
- WP 3: Implementeer een versie van het marching cubes algoritme. Denk daarna een oplossing uit om dit algoritme uit te breiden naar marcherende bollen rond punten, of polytopen, of cubes die van grootte veranderen.
- WP 4: Implementeer een algoritme dat geometrische objecten aan de puntenwolk kan fitten, bv. verzamel alle punten die in werkelijkheid op 1 vlak liggen en vervang deze door een gecorrigeerde positie (dus verwijder de meetruis). Hou er rekening mee dat een vlak eindig is, dat objecten elkaar occluderen, etc.
- WP 5: Registratie van puntenwolken met grijswaardendata. Vanop de positie van de laserscanner nemen we ook enkele foto's. Probeer deze te registreren met de 'intensiteits'informatie van de laserdata (vanuit min of meer hetzelfde zichtpunt) zodat aan elk punt in de wolk een natuurlijkere kleur geconnecteerd kan worden.

**Testcase:** Het uiteindelijke doel is de ontwikkeling van algoritmes en de verificatie van hun werking op de beschikbare data.

## 8 Analyse van loopbeweging

De analyse en evaluatie van bewegingen is erg belangrijk aspect van bewegingsleer, voor toepassingen in revalidatie en topsport. In deze thesis willen we enkele deelproblemen analyseren met computervisiealgoritmes. Een eerste probleem is het bepalen van het tijdstip voor hielcontact en toe-off tijdens wandelen, lopen, versnellend wandelen, versnellend lopen, walk-to-run transitie, run-to-walk transitie op loopband en overgronds. Belangrijk is om te kunnen differentiëren tussen links en rechts. Daarnaast moet ook de correcte afstand tussen de coördinaten van hielcontact en toe-off bepaald worden

**Werkpakketten:** Om dit systeem te realiseren moeten enkele modules ontwikkeld worden die bestaan uit verschillende werkpakketten. De beweging moet accuraat beschreven kunnen worden, en de tijdstippen van hielcontact en toe-off moeten tot op een frame accuraat geanalyseerd worden.

- WP 1: Bewegingsdetectie: Detecteer de beweging van de proefpersoon. Gebruik verschillende voorgrond-achtergrond-segmentatietechnieken (ViBe, MoG, first order filter, etc.) om de beweging in een beeld te detecteren. Kijk vooral naar de robuustheid van de algoritmes, het is erg belangrijk om de outline van de voeten op elk moment zo juist mogelijk te bepalen.
- WP 2: Bewegingsdetectie: Detecteer de beweging van de proefpersoon. Gebruik textuur- en intensiteitsegmentatietechnieken om de beweging van de onderbenen te onderscheiden van de achtergrond in elk beeld. Kijk vooral naar de robuustheid van de algoritmes, het is erg belangrijk om de outline van de voeten op elk moment zo juist mogelijk te bepalen.
- WP 3: Bewegingsannotatie: beschrijf de beweging van de voet (vooral van hiel en teen, dus van de zool), en stel een criterium op om de hielcontact en toe-off events uit de beschrijving te extraheren.
- WP 4: Verificatie: Stel een betrouwbaarheidsmaat op zodat elk event een score krijgt al naargelang de betrouwbaarheid van de detectie (in samenspraak met WP3). Ontwerp een gebruiksvriendelijke GUI waarin de video's ingelezen worden en de resultaten geanalyseerd kunnen worden. Laat toe dat (indien nodig) parameters ingesteld kunnen worden zodat op basis van de betrouwbaarheidsmaat de correcte tijdstippen van de events gevonden worden, en ook vergeleken kunnen worden met een grondwaarheid.

**Testcase:** Het uiteindelijk doel is een correcte en accurate werking van het ontwikkelde visiesysteem op de beeldsequenties, opgenomen in verschillende omstandigheden.

## 9 3D trajectreconstructie (pingpong)

In eender welke sport is de analyse van techniek en tactiek op basis van videobeelden een steeds groter aandachtspunt. In dit project onderwerpen we tafeltennis aan een casestudy, als interessante toepassing van vele computervisietechnieken hierin samenkomen.

Beschrijf het traject van een pingpongballetje in 3D op basis van de beeldinhoud uit 2 overlappende camerabeelden. Gebruik een subset van beelden uit het begin van de sequentie om de camera's te kalibreren, detecteer en reconstrueer het traject van het balletje. In een latere fase kan dan overgegaan worden naar een analyse van techniek en tactiek.

**Programmeeromgeving en beelden:** Er moet een beeldverwerking algoritme geprogrammeerd worden in Matlab of C/C++ (de OpenCV-library en eventueel VTK/OpenGL) dat de positie van het balletje trackt en visualiseert. Gebruik de beschikbare datasequenties.

### Werkpakketten:

- WP 1: Kalibreer de camera's volgens methode van Bouguet (matlab of OpenCV) en pas aan naar de wide baseline opstelling gebruikt voor deze beelden. Rectificeer de beelden in de videostroom met de bekomen distortieparameters.
- WP 2: Detecteer het balletje in elk frame, en track het doorheen alle opeenvolgende frames van de sequentie. Maak gebruik van een particle filter approach om dit te verwezenlijken.
- WP 3: Om de positie van de tafel en het balletje in 3D te kennen, moet de beeldpositie teruggeprojecteerd worden naar de ruimte. Voer de terugprojectie uit voor de positie van tafelhoeken en balletje. Waar de teuggeprojecteerde stralen voor de puntpositie van hetzelfde object in verschillende beelden elkaar 'raken', zal de 3D positie van dat punt van het object zich bevinden.
- WP 4: Detecteer en track de palet doorheen de sequentie. Zoek een techniek om de paletten te detecteren en volgen doorheen de sequentie, zodat uiteindelijk het contactmoment gedetecteerd en geanalyseerd kan worden.
- WP 5: Visualiseer positie van tafel, paletten en balletje in 3D in een handige gebruikersinterface. Doe een analyse van het 3D traject van de bal (kan deze bv. beschreven worden door een parabool?). Gebruik eerst matlab, en stap daarna over naar VTK.

**Testcase:** Het uiteindelijk doel is de ontwikkeling van algoritmes en de verificatie van hun werking op de beschikbare beeldsequenties.

## 10 Surveillance en veiligheid

In dit project onderzoeken we de mogelijkheden van multi-camerasystemen in de monitoring van veiligheid op industriële sites. Op elk moment detecteren we de positie van werknemers op de site, en evalueren of ze niet in verboden zones terechtkomen. Daarnaast gebruiken we het cameranetwerk ook om uitslaande brand te detecteren op basis van typische karakteristieken, zoals kleur, flikkering, etc. Indien iemand in een verboden zone terechtkomt, of als er brand uitlaat moet er tijdig alarm geslagen worden.

**Programmeeromgeving en beelden:** Er moet een beeldverwerking algoritme geprogrammeerd worden in Matlab dat de positie van werknemers surveilleert, en tegelijk controleert of er brand ontstaat. Ontwikkel 3 verschillende methodes en evalueer welk van deze optimaal presteert voor de gegeven dataset.

### Werkpakketten:

- WP 1: Kalibreer de camera's (volgens methode van Bouguet (matlab)), en zorg voor de externe kalibratie van de camera's in de wide baseline opstelling gebruikt voor deze beelden.
- WP 2: Bewegingsdetectie: Detecteer de beweging van de werknemers op de site. Gebruik verschillende voorgrond-achtergrond-segmentatietechnieken (ViBe, MoG, first order filter, etc.) om de beweging in een beeld te detecteren, en geef deze door als region-of-interest voor de volgende WPen. Bespreek vooral ook de robuustheid van de algoritmes tegen belichtingsvariëaties (vooral de variatie van het buitenlicht).
- WP 3: Track de positie van de werknemers over de tijd, positioneer ze in 3D op de site op elk tijdstip en geef aan of iemand zich in een verboden zone begeeft.
- WP 4: Gebruik een analyse van de temporele kleurveranderingen om brand te detecteren, en vergelijk met WP5-6.
- WP 5: Ontwikkel een textuurgebaseerde methode (bv. Gabor-filters) om brand te detecteren, en vergelijk met WP4-6.
- WP 6: Ontwikkel een waveletgebaseerde methode om brand te detecteren, en vergelijk met WP4-5.

**Testcase:** Het uiteindelijke doel is de ontwikkeling van algoritmes en de verificatie van hun werking op de beschikbare beeldsequenties.

## 11 Snapshot-gebaseerde kalibratie

Kalibratie houdt in dat zowel de kenmerken van de camera en de optica gemodelleerd worden, als dat de extrinsieke parameters voor positie en oriëntatie van de camera vastgelegd worden. Tegenwoordig worden dikwijls multicameranetwerken gebruikt in ons onderzoeksdomein, waarvoor een bepaalde camera-opstelling op een bepaald tijdstip volledig onderzocht en gekalibreerd wordt. Indien we dit experiment willen herhalen willen we tijd besparen door de gegevens van de voorgaande metingen te herbruiken.

In dit project moet een algoritme ontwikkeld worden dat in een eerste fase de volledige kalibratieprocedure beschrijft, en in een tweede fase toelaat om de extrinsieke cameraparameters te herberekenen door de snapshots vanuit de nieuwe positie van de camera's te vergelijken met de beelden vanuit de volledig gekalibreerde opstelling.

**Programmeeromgeving:** Er moet een beeldverwerkingalgoritme geprogrammeerd worden in Matlab dat de kalibratie uitvoert op basis van een dambordpatroon, en een tweede algoritme dat de beeldinhoud van de snapshots vergelijkt met de voorgaande en daaruit de extrinsieke parameters afleidt.

### Werkpakketten:

- WP 1: Kalibreer de camera's (volgens methode van Bouguet (matlab)), en zorg voor de externe kalibratie van de camera's in de wide baseline opstelling gebruikt voor deze beelden.
- WP 2: Bereken de 3D locatie van elk punt uit de overlapping van de beide standpunten (door terugprojectie op basis van de kalibratiegegevens). Evalueer de nauwkeurigheid van de punten in de teuggeprojecteerde ruimte, en zorg voor een mooie 3D visualisatie van de puntenwolk.
- WP 3: Vergelijk de inhoud van de snapshots gemaakt in het tweede experiment met de beeldinhoud van het eerste door featuredetectie en -matching. Leg zo de overeenkomsten vast en vergelijk de performantie van SIFT, SURF, MSER, etc.
- WP 4: Ontwikkel een algoritme dat de positie van de camera's in het tweede experiment kan bepalen op basis van de kennis uit het vorige WP. Op basis van elke subset van 4 gekende punten in 1 vlak kan de camerapositie en oriëntatie bepaald worden. Zoek een criterium dat de informatie van elke subset combineert, en een optimale positie- en oriëntatieparameters extraheert.

**Testcase:** Het uiteindelijk doel is de ontwikkeling van algoritmes en de verificatie van hun werking op de beschikbare beeldsequenties.