Geautomatiseerde beeldanalyse van morfologische kenmerken en de motiliteit van een individuele spermatozoon.

Analyse van de kop, de staart en het afgelegde pad voor mobiele en getrapte cellen.

D.J. Geijs

Examination Committee Prof. dr. ir. Albert van den Berg Dr. ir. Loes Segerink Dr. ir. Ferdi van der Heijden B. de Wagenaar BIOS Lab on a Chip group

Verslagnummer: 2015-2

Bachelor thesis

Voorwoord

De afgelopen periode heb ik bij BIOS gewerkt aan mijn eerste kleine wetenschappelijk onderzoek. Ik heb de BIOS vakgroep als een gezellige ploeg leren kennen. Ik wil Loes Segerink bedanken voor haar begeleiding bij het gehele proces van dit onderzoek. Haar kritische blik heeft geholpen om mijzelf voortdurend te verbeteren. Bjorn de Wagenaar wil ik bedanken voor zijn interesse in dit project. Het was fijn om mijn resultaten te bespreken met iemand die op de hoogte was van mijn onderzoek. Zijn enthousiasme zette mij aan om het beste uit mijn resultaten te halen. Ferdi van der Heijden wil ik bedanken voor zijn uitgebreide hulp bij het programmeren. Als laatste wil ik mijn huisgenoot Tom Lankhorst bedanken voor het aanhoren van mijn problemen en de nuttige tips.

Abstract

Koppels met zwangerschaps problemen kunnen worden geholpen door het toepassen van kunstmatige bevruchting. Een techniek daarvoor is intracytoplasmatische sperma-injectie (ICSI), waarbij een zaadcel direct in de eicel wordt gespoten. Een factor die het succespercentage van ICSI beïnvloed is de kwaliteit van de getransplanteerde embryo zijn. Er wordt verondersteld dat een beschadigde spermacel die gebruikt wordt voor ICSI de kwaliteit van de getransplanteerde embryo negatief beïnvloed en dat de kwaliteitscontrole van spermacellen de succeskansen van ICSI kan verbeteren.

De BIOS Lab on a Chip groep is het gelukt om spermacellen te vangen op fibronectine spots. Er is alleen nog niet onderzocht of een spot de morfologie of de motilteit beïnvloed. Om dit te onderzoeken is er daarom een automatisch beeldanalyse systeem ontwikkelt en getest die zowel vrij zwemmende als gevangen cellen kan analyseren en kan vergelijken.

Het ontworpen analysesysteem kan succesvol zaadcellen detecteren, volgen, kwantificeren en kwalificeren. Hierdoor is het systeem in staat om verschillende parameters te bepalen van de individuele zaadcellen. De zaadcellen zijn gedetecteerd door het toepassen van een drempelwaarde. Het midden van de kop van de zaadcellen is berekend en dit midden is gevolgd gedurende de meting. Het traject dat een zaadcel heeft afgelegd kan worden bepaald en per zaadcel worden onderscheiden. Abnormaliteiten aan de staart kunnen worden gedetecteerd en per zaadcel kan de staart door de tijd worden gevolgd. De maximale hoek die de staart maakt ten opzichte van de kop kan worden bepaald. Op basis van deze hoek kan een vrijzwemmende cel geclassificeerd worden als progressieve of non progressieve cel.

Het verder testen en analyseren van de parameters is noodzakelijk, maar de eerste stap is gemaakt om de fibronectine spots beter te kunnen onderzoeken.

Inhoudsopgave

1	Inleiding en probleemstelling			
2	Theorie			
	2.1	Kenmerken van een spermatozoon	3	
	2.2	Morfologie	4	
	2.3	Motiliteit	5	
3	Met	hoden	7	
	3.1	Data	7	
	3.2	Doel en eisen	7	
	3.3	Software opbouw	9	
	3.4	Kop detectie	9	
	3.5	Pad cel bepalen	12	
	3.6	Oriëntatie cel bepalen	12	
	3.7	Staart detecteren	14	
	3.8	Parameters	14	
4	Resultaten			
	4.1	Kop detectie	19	
	4.2	Morfologie	21	
	4.3	Motiliteit	24	
5	Discussie en aanbevelingen			
	5.1	Data	29	
	5.2	Methode en resultaten	29	
6	Con	clusie	31	
Refe	rentie	es	32	
Appe	Appendix 3			

.

Inleiding en probleemstelling

Inleiding

Koppels met zwangerschaps problemen kunnen worden geholpen door het toepassen van kunstmatige bevruchting, zoals intra-uteriene inseminatie (IUI),in-vitrofertilisatie (IVF) of intracytoplasmatische spermainjectie (ICSI). IUI is in de meeste gevallen de eerste fertiliteitsbehandeling die wordt toegepast. Bij deze voortplantingstechniek worden opgevangen zaadcellen van de partner kunstmatig in de baarmoeder ingebracht. Na ongeveer zes mislukte IUI behandelingen wordt er meestal voor IVF gekozen (UMCG, 2014). IVF wordt ook wel reageerbuisbevruchting genoemd. Dit komt doordat de geïsoleerde eicel van de vrouw samen met de zaadcellen van de man in een medium worden samengevoegd. Na ongeveer 24 uur wordt er gekeken of er een bevruchting heeft plaatsgevonden. ICSI is een meer geavanceerde technologie, die de eicel bevrucht door middel van een punctie in de eicel. Een spermacel wordt direct in de eicel aangebracht. Deze cel wordt handmatig gekozen op basis van een normale morfologie (Kotwicka et al, 2007). Het probleem hierbij is alleen dat de laborant de spermatozoön selecteert voor injectie en hierbij de normaliter natuurlijke selectie barrière omzeilt. Op deze manier kan een beschadigde spermatozoön die normaal niet de oöcyt had bereikt, de eicel bevruchten. Dit kan leiden tot embryo's die na het implanteren een lagere potentie hebben om zich door te ontwikkelen (Wildeling et al, 2011). Wanneer men kijkt naar het slagingspercentage van IVF en ICSI zijn beiden ongeveer 30% (Andersen et al, 2009). De reden dat dit slagingspercentage zo laag ligt kan door een reeks van factoren zoals overgewicht, leeftijd, afwijkend endocrien systeem of door de kwaliteit van de getransplanteerde embryo. Dit laatste punt is lastig te verbeteren bij IVF aangezien de bevruchting willekeurig gebeurd met één van de spermacellen. Bij ICSI is deze mogelijkheid tot verbetering er wel aangezien de spermacel handmatig wordt ingebracht. Motile sperm organelle morphology examination (MSOME) is ontwikkeld om spermacellen te analyseren bij een vergroting van 6000x en eigenschappen te kwantificeren om een potentiële cel te selecteren voor ICSI. (Wildeling et al, 2011). Wanneer een techniek wordt gebruikt die op morfologische basis een cel selecteert voor ICSI spreekt men van Intracytoplasmic Morphologically Selected Sperm Injection (IMSI). Een belangrijke parameter voor de gezondheid van de cel is de DNA-fragmentatie. De gradatie van deze biochemische parameter kan een voorspelling geven over de slagingskans van ICSI

(Speyer et al., .2010). Morfologische kenmerken en de DNA-fragmentatie correleren op basis van verschillende onderzoeken (Setti et al., 2013). Onderzoek naar IMSI toont aan dat er een verhoogde kans op zwangerschap is en dat abrupt zwangerschapsverlies verlaagd wordt (Berkovitz et al, 2006; Nadalini et al, 2009; Bartoov et al, 2003; Antinori et al, 2008). Het nadeel van MSOME is dat het 60 tot 210 minuten kan duren om een normale cel te vinden. Verder moeten twee embryologen samen de analyse doen op het zelfde sample om de subjectiviteit te verkleinen. Het is dus een dure en tijdrovende methode. (Darwish, 2012).

Ook zijn er onderzoeken waaruit blijkt dat de motiliteit invers correleert met de DNA-fragmentatie (Huang et al., 2005; Peluso et al., 2013). Om deze motiliteits parameters te meten zijn er automatische systemen op de markt. Computer assisted sperm analysis (CASA) systemen zijn ontwikkeld om de motiliteit van sperma te analyseren die zich vrij kunnen bewegen.

MSOME is helaas niet uit te voeren in combinatie met CASA. Dit komt omdat voor MSOME de cellen geïmmobiliseerd moeten zijn. De BIOS Lab on a Chip group van de Universiteit Twente heeft een platform ontwikkeld die het mogelijk maakt om spermacellen vast te houden op geprinte fibronectine spots zonder dat deze geïmmobiliseerd hoeven te worden (Frimat et al, 2014). De spots geven de cellen voldoende vrijheid om te bewegen, echter dit gebeurd in een circulaire beweging. Dit platform biedt de mogelijk om individuele cellen automatisch te analyseren op zowel morfologie en motiliteit alvorens deze worden geïnjecteerd met ICSI of om onderzoek te doen naar relaties tussen de morfologie, motiliteit en DNAfragmentatie.

Probleemstelling

Er is nog geen bewijs of lineair progressieve cellen zich hetzelfde gedragen wanneer zij getrapt zijn. De fibronectine spot zou dus de motiliteit kunnen beïnvloeden. Ook kan de cel zodanig aan de spot gehecht zijn dat bepaalde morfologische kenmerken niet meer zichtbaar zijn. Het doel is om een softwareplatform te ontwikkelen die zowel vrij zwemmende als getrapte cellen automatisch kan analyseren op morfologie en motiliteit. In vervolgonderzoek kan dan worden onderzocht of er morfologische of motiliteits parameters veranderen wanneer een unieke cel op een fibronectine spot afzwemt om vervolgens in getrapte toestand een roterende baan te vervolgen.

Doel

Het doel van dit onderzoek is om een automatisch image analysis systeem te ontwikkelen en te testen voor de kwantificatie en kwalificatie van de motiliteit en morfologie van individuele spermacellen. De methode moet universeel werken voor getrapte als vrij zwemmende cellen en inzetbaar zijn om de probleemstelling op te lossen.

Dit onderzoek begint met theorie over de morfologie en motiliteit van een spermacel en over de huidige methoden om deze te kwalificeren of te kwantificeren. De methode die hier op volgt begint met een introductie van de beschikbare data en apparatuur gevolgd door een eisenpakket van de software. De methode wordt vervolgens opgesplitst in verschillende onderdelen. Het detecteren van de kop, het volgen van de kop, de orïentatie van de cel en het detecteren van de staart. Vervolgens worden er parameters beschreven die aan de hand van de detectie de cel kan kwalificeren of kwantificeren.

De opgestelde parameters en methoden zijn geanalyseerd en de resultaten worden in dezelfde volgorde als de methode gepresenteerd.

2 Theorie

2.1 Kenmerken van een

spermatozoon

De kop van spermatozoön is ongeveer 4 tot 5 μ m lang en is 3 μ m lang op het breedste stuk. De kop (figuur 2.1) bevat drie onderdelen: de nucleus, acrosoom en het centriool met daaraan het flaggelum(staart). (Saladin, 2010)

Het grootste gebied van de kop wordt door het meest belangrijke onderdeel van de spermacel ingenomen, de nucleus. Deze nucleus bezit de genetische informatie die uiteindelijk gedeeld wordt met de vrouwelijke eicel. Het centriool met daaraan de staart zit in een inkeping aan het einde van de nucleus (Saladin, 2010). Daarnaast bevat de kop een ander element: de acrosoom.

De acrosoom is een lysosoom in de vorm van een dunne schil om de punt van nucleus. Deze lysosoom breekt niet alleen de afvalstoffen in de cel af, het bezit ook de enzymen die nodig zijn voor de penetratie van de eicel (Saladin, 2010; Pinkhof et al., 2012).

Het middenstuk, de staart en het eindstuk vormen in zijn geheel het flagellum (figuur 2.1). Het dikste gedeelte is het middenstuk, een cilinder die ongeveer 3 tot 5 µm lang is en half zo breed als de kop. Om het axiale filament zitten veel mitochondria. Deze produceren de adenosinetrifosfaat (ATP) die nodig is om de staart heen en weer te laten bewegen. De staart, 40 tot 45 µm lang, is het langste deel van het flagellum. Het axiale filament is hier





omgeven door ondersteunende fibers. Het eindstuk, 4 tot 5 μ m lang, en bestaat alleen uit de uitloper van het axiale filament en vormt zo de punt van de staart (Saladin, 2010). In dit onderzoek wordt gebruik gemaakt van varkens spermatozoa, omdat dit makkelijk te verkrijgen is en sterk op dat van mensen lijkt. Ter vergelijking zijn de afmetingen

Tabel 2.1:	Tabel met de afmetingen	van menselijk spermatoz	oa en varkens spermatozoa.
		<i>,</i> ,	•

Afmeting	Menselijk spermatozoa (µm, 95% be- trouwbaarheidsinterval) (World Health Organization, 2010)	Varkens spermatozoa (µm) (Kondracki et al, 2012)
Kop - lengte	3.7 - 4.7	9.2 - 9.7
Kop - breedte	2.5 - 3.2	4.4 - 5.0
Middenstuk - lengte	3.3 - 5.2	-
Middenstuk - breedte	0.5- 0.7	-
Staart - lengte	~45	~45

A. Head Defects



Figuur 2.2: Schematische tekening van verschillende abnormale vormen van menselijk spermatozoa (World Health Organization, 2012).

naast elkaar gezet in tabel 2.1.

2.2 Morfologie

Standaard methode

De World Health Organization [WHO] (2010) heeft op basis van de morfologie een classificatie gemaakt tussen normale en abnormale cellen. Hiervoor is een gefixeerde uitstrijk van het sperma nodig die vervolgens wordt aangekleurd met diverse kleuringen. De preparaten worden onder de microscoop bekeken bij een vergroting van 40X of 100X (Setti et al., 2013) en elke cel wordt individueel geïnspecteerd. De laborant classificeert de cel aan de hand van de volgende opgestelde richtlijnen.

Een normale zaadcel heeft een egale, gelijkmatige verdeelde celwand en is ovaal van vorm. De acrosoom neemt ongeveer 40-70% van de totale oppervlakte van de kop in en mag geen grote vacuolen bevatten.

Het middenstuk moet gelijkmatig en dun zijn en heeft ongeveer dezelfde lengte als die van de kop. De as van het middenstuk moet in het verlengde liggen van die van de kop.

De staart is dunner dan het middenstuk, overal gelijkmatig van dikte en ongeveer 45 μ m lang. De staart mag gebogen zijn zolang er geen scherpe hoek in zit. (World Health Organization, 2010).

Een zaadcel kan op verschillende plekken abnormaliteiten vertonen. De kop kan te groot of te klein, rond, puntvormig of zelfs dubbel zijn. De nek en het middenstuk kunnen asymmetrisch aan de kop zitten. Ook kunnen ze te dik, te dun, of te scherp gebogen zijn. De staart kan defecten vertonen zoals een kleine, dubbele, gekrulde of scherp gebogen staart. In figuur 2-2 zijn verschillende defecten illustratief weergegeven.

Motile sperm organelle morphology examination

MSOME is een andere methode die gebruikt wordt om defecten op te sporen zonder daarbij een kleuring te gebruiken. Dit heeft als voordeel dat de cellen niet beschadigd worden door een kleuring en dat ze nog bruikbaar zijn voor een injectie in de eicel. Wanneer met MSOME een cel wordt gekozen voor injectie spreekt men van intracytoplasmic morphologically selected sperm injection(IMSI). Voor de observatie van verschillende organellen wordt gebruik gemaakt van een hoge magnificatie van 6600X in plaats van de standaard 40x tot 100x vergroting (Setti et al., 2013). De volgende organellen zijn onder deze vergroting te zien: het acrosoom, de nek, de staart, de mitochondriën en de nucleus. Aan afwijkingen van de nucleus wordt op dit moment het meeste aandacht geschonken. Dit komt doordat de aanwezigheid, hoeveelheid en grote van vacuolen(figuur 2.3) in de nucleus gecorreleerd lijkt te zijn met het percentage DNA fragmentatie in de cel (Perdix et al., 2011; Garolla et al., 2008; de Almeida Ferreira Braga et al., 2011; Hammoud et al., 2013). Studies tonen aan dat DNA-beschadigd sperma dat wordt gebruikt voor ICSI in verband wordt gebracht met het mislukken van de embryonale ontwikkeling tijdens of na het implanteren van de embryo (Borini et al., 2006; Tesarik et al., 2004).

Echter het vinden van spermatozoa zonder vacuolen in een sample is een moeilijk en tijdrovende procedure en hiervoor is een laborant nodig.

2.3 Motiliteit

Standaard methode

De WHO heeft richtlijnen opgesteld voor het handmatig classificeren van de motiliteit van spermatozoa. De motiliteit kan opgedeeld worden in drie categorieën: progressieve motiliteit(PR), non-progressieve motiliteit(NP) en immotility(IM). Cellen met een progressieve motiliteit beweging actief in lineaire baan of in een grote cirkel. Alle andere patronen vallen in de NP, dit zijn cellen waar geen sprake is van progressie zoals het zwemmen in kleine cirkels. De laatste categorie (IM) zijn cellen die geen beweging hebben en waarvan de staart ook niet beweegt.

Minimaal 200 cellen worden handmatig onder een vergroting van 200x of 400x geanalyseerd en verdeeld in één van de drie hierboven genoemde categorieën. De verdeling van de cellen over de drie categorieën wordt vervolgens gebruikt om de totale motiliteit (PR+NP) en de progressieve(PR) motiliteit te bepalen. Respectievelijk iis de ondergrens hiervan 40% en 32%. (World Health Organization, 2011).

Computer assisted sperm analysis

Het analyseren van de motiliteit wordt ook gedaan met behulp van geautomatiseerde afbeeldingsanalyse. Computer assisted sperm analysis (CASA) wordt uitgevoerd door systemen zoals de Hamilton's IVOS II Sperm Analyzer en de CEROS II Sperm Analyzer. CASA gebruikt digitale afbeeldingen om de baan van elke spermacel te bepalen door middel van verschillende algoritmes(figuur 2.4). Wanneer de baan van een cel bepaald is door het CASAsysteem kunnen hiervan verschillende parameters worden bepaald. Deze CASA-parameters zijn mathematisch opgesteld om zo precies mogelijk de beweging van een cel te beschrijven wanneer deze zich door het microscopische veld beweegt (Verstegen et al., 2002). De IVOS II Sperm Analyzer maakt gebruik van een fasecontrast microscoop met objectieven tussen de 4x en 100x. De video's die worden gebruikt hebben een resolutie van 640x480 pixels en een framerate van 7.5 tot 60 frames per seconde (Origio, 2011). Concurrerende systemen hebben soortgelijke specificaties (Microoptic,2013).

In tabel 2.2 zijn de parameters te zien die de meeste CASAsystemen bepalen van een sample. Deze parameters worden gebruikt in onderzoeken naar spermatozoa. Zoals bijvoorbeeld bij onderzoek naar de DNA-fragmentatie van een spermacel. Een onderzoek van Huang et al. (2005) toont aan dat er mogelijk een negatieve correlatie is tussen de straight line velocity (VLS) en de DNA-fragmentatie. Cohen-Bacrie et al. (2009) heeft de amplitude of lateral head displacement (ALH) onderzocht en daarbij geen significante correlatie gevonden. Peluso et al. (2013) heeft onderzoek gedaan naar lineair progressieve cellen (cellen die daadwerkelijk een verplaatsing hebben) en stelt dat er een correlatie is tussen de DNA-fragmentatie en de lineair progressieve motiliteit.



Figuur 2.3: Aanwezigheid van vacuoles in de nucleus bij een vergroting van 6000x opgedeeld in drie groepen: a,b,c) cellen met enkele of geen vacuole daarin. d,e,f) cellen met te grote of meerdere vacuoles. g,h,i) Cellen met te grote én meerdere vacuoles (Knez et al., 2011).



Figuur 2.4: Afbeelding waarin de verschillende CASA afkortingen uit tabel 2-2 grafisch zijn weergegeven. (World Health Organization, 2010)

Tabel 2.2: Omschrijving van verschillende veelgebruikte CASA parameters. * Verschillende CASA systemen gebruiken verschillende mathematische algoritmes om deze parameters te bepalen. Er zijn nog geen gegevens bekend over de verschillen tussen deze systemen (World Health Organization, 2010).

Afkorting	Betekenis afkorting	Omschrijving (World Health Organization, 2010)	Grootheid
VCL	curvilinear velocity	Tijd-gemiddelde snelheid van een sperma kop van het daadwerkelijk afgelegde pad(curvilinear path).	µm/s
VSL	straight-line velocity	Tijd-gemiddelde snelheid van een sperma kop tussen het eerste en laatste gedetecteerde punt.	µm/s
VAP*	average path velocity	Tljd-gemiddelde snelheid van de sperma kop van zijn gemiddelde pad.	µm/s
ALH*	amplitude of lateral head displacement	Magnitude van de laterale uitwijking van de sperma kop ten opzichte van het gemiddelde pad(VAP).	μm
LIN	lineariteit	De lineariteit van het curvilinear path, VSL/VCL	-
WOB*	wobble	Oscilitatie van het curvilinear path over het gemiddelde pad. VAP/VCL	-
STR*	straightness	Lineariteit van het gemiddelde pad. VSL/VAP	-
BCF*	beat-cross frequency	Gemiddelde frequentie waarmee het curvilinear path het gemiddelde pad doorkruist.	Hz
MAD	mean angular displacement	De tijd-gemiddelde absolute waarde van het hoek verschil van het curvilineare path.	graden



3.1 Data

Voor dit onderzoek zijn in totaal dertig video's gebruikt (Appendix 1). Hiervoor is een Nikon TE-2000 microscoop gebruikt met meerdere camera's. Vierentwintig video's zijn opgenomen met een Nikon DS-Ri1 camera en zes met een Basler ACA780-75 camera, beide met verschillende instellingen. Er zijn video's beschikbaar met een framerate van 16.6 fps tot en met 50 fps. De kleinste resolutie van de video is 320x256 en de grootste is 1280x1024. Vierentwintig video's zijn opgenomen met zowel een groen filter als zonder filter. Op zes video's zijn getrapte spermatozoa te zien en op vierentwintig video's vrij zwemmende. In tabel 3-1 is een overzicht te zien van de gebruikte video dataset. Alle video's zijn opgenomen met een vergroting van 10x.

3.2 Doel en eisen

Het doel van dit onderzoek is om een automatisch image analysis systeem te ontwikkelen en te testen voor de kwantificatie en kwalificatie van de motiliteit en morfologie van individuele spermacellen. De methode moet universeel werken voor getrapte als vrij zwemmende cellen en inzetbaar zijn om de probleemstelling op te lossen. Om dit doel te bereiken wordt het doel opgedeeld in een aantal subproblemen. Allereerst moet de cellen worden gedetecteerd in ieder frame van de video en moet het afgelegde pad van iedere cel worden bepaald. Met deze informatie weet men op ieder tijdstip waar iedere cel zich bevindt. Dit is nodig om metingen te doen aan de cel gedurende een tijd. Op basis van het doel, de theorie en de beschikbare data zijn de volgende eisen opgesteld waaraan de software moet voldoen:

- Sestructureerde en gebruikersvriendelijke code
 - » Code moet makkelijk uit te breiden zijn.
 - » Code moet makkelijk te begrijpen zijn.
 - » Analyse moet vlot verlopen(1 ~ 2 minuten).
- > Detectie van spermatozoa
 - » Ongevoelig zijn voor contrast en helderheid.
 - » Dode cellen moeten gefilterd kunnen worden.
 - » Moet getrapte en vrij zwemmende spermatozoa kunnen detecteren.
- > Volgen van spermatozoa over de tijd
 - » De locatie en oriëntatie moet worden opgeslagen.
 - » Het afgelegde pad moet per cel worden bepaald.
 - » Moment van trappen registreren.
- > Detectie van de staart
 - » Moet de staart van getrapte en vrij zwemmende spermatozoa kunnen detecteren.
- > Volgen van de staart over de tijd

Tabel 3.1: Overzicht van totale video database bestaande uit 30 video's. Opgenomen met twee verschillende camera's.

Video database (30)					
Basler ACA780-75 (6)	Nikon DS-I	Ri1 (24)			
Vrij zwemmend (6)	Trapped (6)	Vrij zwemmend (18)			
 ▶ Filter Groen (3) Geen (3) ▶ FPS 25 (2) 37 (2) 50 (2) 	 ▶ Filter Groen (6) Geen (0) ▶ FPS 16.6 (6) 	 ▶ Filter Groen (9) Geen (9) ▶ FPS 16.6 (18) 			
► Resolutie 780x572 (6)	▶ Resolutie 720x480 (6)	 ▶ Resolutie 1280x1024 (6) 640x512 (6) 320x256 (6) 			

Box 1: Function Flow Chart



- » Meerdere locatie punten moeten worden opgeslagen.
- » De gedetecteerde staart moet gekoppeld worden aan de juiste cel.
- > Parameters opstellen
 - » De grootte van de kop en de lengte van de staart moet worden bepaald.
 - » De VLS moet worden bepaald.
 - » Wanneer getrapt moet de rotatiesnelheid worden bepaald.
 - » De maximale uitwijking van de staart moet worden bepaald.

3.3 Software opbouw

De applicatie wordt geschreven in Matlab. Het voordeel van Matlab is dat deze een grote toolbox heeft en het dus makkelijk maakt om snel concepten te testen. Eén van de eisen is dat code van de software gestructureerd en gebruikersvriendelijk is. Er is daarom gekozen om de software in zoveel mogelijk losstaande functies te schrijven. Dit heeft een tal van voordelen. Losse functies maken de code overzichtelijk doordat er een hiërarchie ontstaat. Ze werken als blokken en kunnen toegevoegd worden wanneer deze daadwerkelijk nodig zijn. Op deze manier worden alleen berekeningen gedaan die daadwerkelijk nodig zijn. De rekentijd blijft zo optimaal.

In box 1 is een flowchart weergegeven van de functies die zijn ontwikkeld. Deze zijn in vijf verschillende categorieën opgedeeld en de methode achter de functies zal in dezelfde volgorde worden toegelicht. (De categorie video inladen zal niet worden toegelicht. De code hiervan is te vinden in appendix B en C.) Bij elke methode beschrijving wordt de functie genoemd die daar betrekking op heeft.

3.4 Kop detectie

Om een bewegende spermacel door de tijd heen te volgen is het belangrijk om de kop te detecteren in een stilstaand frame. Wanneer namelijk in elk frame de positie van de kop kan worden bepaald kan het pad dat de cel aflegt door de tijd heen worden bepaald. Voorafgaand aan de detectie van de kop wordt de statische achtergrond bepaald. De statische achtergrond wordt gebruikt om de video te filteren voor helderheid verschillen en dode cellen. Met behulp van een drempelwaarde wordt de video omgezet naar een binaire video met daarin slechts de koppen van de cellen. De koppen in de binaire video kunnen vervolgens worden gedetecteerd. De locatie en de afmetingen van de gedetecteerde cellen worden opgeslagen.







Figuur 3.1: Resultaat (c) na het verwijderen van de statische achtergrond (b) van het originele frame (a). De rode cirkels geven de cellen aan die verdwijnen door deze methode.

Achtergrond verwijderen

f_{x} isolateheads (Appendix C), findtail (Appendix J)

Om de statische achtergrond te bepalen wordt van elke pixel in de video de gemiddelde waarde genomen. Dit is het gemiddelde van alle waarden die de pixel heeft gedurende de video. Deze statische achtergrond kan vervolgens worden afgetrokken van elk frame om alleen de dynamische objecten over te houden (figuur 3.1). Dit heeft als voordeel dat ruis en vervuiling van het sample

wordt weg gefilterd. Drempelwaarde

$f_{\rm x}$ isolateheads (Appendix C), showheads (Appendix E)

Nadat de statische achtergrond is verwijderd uit het frame wordt de afbeelding omgezet naar een zwart-wit afbeelding. Om deze conversie mogelijk te maken is er een drempelwaarde nodig, om te bepalen welke grijswaarden zwart worden en welke wit. Deze drempelwaarde kan zo gekozen worden dat alleen de grijswaarden die aanwezig zijn in de kop wit worden (figuur 3.2). Dit is gunstig aangezien we alleen geïnteresseerd zijn in de kop.

De drempelwaarde kan handmatig worden gekozen door te kijken wat de grijswaarde is aan de rand van de kop, dit is de onder drempel. Vervolgens kan worden gekeken wat de boven drempel is, dit is de hoogste grijswaarde die voorkomt in de kop van de cel. Over het algemeen is het instellen van alleen een onder drempel voldoende om alleen de kop te detecteren, een boven drempel wordt aangeraden wanneer er last is van ruis.

De drempelwaarde kan ook worden bepaald door gebruik te maken van een algoritme, zodat de gebruiker niet elke keer handmatig de drempelwaarden voor de cel hoeft te detecteren. In de dit systeem is gebruik gemaakt van de Matlab functie 'graythresh' (Otsu, 1975) die statistisch de drempelwaarde bepaald aan de hand van het histogram van de grijswaarden. Deze automatisch bepaalde drempelwaarde wordt vermenigvuldigd met een parameter die van 0 tot 1 loopt. Deze parameter kan door de gebruiker ingesteld worden om hiermee de drempelwaarde te corrigeren. Dit is nodig, omdat het verschil in grijswaarde tussen de randen van de cel en de achtergrond soms erg klein is. Zo klein dat de 'graytresh' functie hier ongevoelig voor is.

Wanneer een threshhold is ingesteld en deze is toegepast op alle frames van de video kan het nuttig zijn om te kijken wat het resultaat is van de ingestelde threshhold. De functie showheads is hiervoor ontwikkeld. De functie laat de video zien zoals te zien is in figuur 3.3a. Deze beelden kunnen weer op de orginele video worden gelegd om te zien wat het resultaat is (figuur 3.3b).





Figuur 3.2: A) Afbeeldingen die vijf verschillende grijswaarden bevat. B) Door het instellen van een onder drempel van 1 en een boven drempel van 4 kan het object van interesse geïsoleerd worden.

Kop detecteren

$f_{\rm x}$ regionprops (Matlab)

Nu het frame een binaire afbeelding is geworden kan er gezocht worden naar ' eilandjes' van enen. Matlab heeft hier een functie voor genaamd regionprops. Deze functie geeft een set van eigenschappen van elk aaneengesloten object in de binaire afbeelding, zoals het zwaartepunt, breedte, lengte en de oppervlakte. Deze informatie zal worden gebruikt voor het analyseren van de celgrootte.





Figuur 3.3: A) Resultaat na het toepassen van een threshhold. Deze kan boven op de orginele video (B) worden gelegd om te kijken of het gewenste resultaat is gehaald.

3.5 Pad cel bepalen

$f_{\mathbf{x}}$ showpaths (Appendix G), findmatch (Appendix F)

De functie regionprops kan objecten detecteren in een frame, maar kan dit object niet volgen door de tijd. Dit laatse is van belang aangezien parameters aan een unieke cel moeten worden gekoppeld. Door gebruik te maken van de coördinaten van de gevonden zwaartepunten kan dit probleem worden opgelost.

Om het pad van een cel te volgen door de tijd heen is het van belang om een uniek pad te identificeren. Hiervoor is de functie findmatch geschreven die een afstandsmatrix maakt tussen de punten van het huidige frame en het komende frame. Per bekend punt wordt een aansluitend punt gezocht die het dichtst bij ligt, deze wordt gekoppeld aan het al afgelegde pad en vormt het nieuwe punt waar van uit gezocht wordt in het volgende frame. Dit wordt gedaan door een afstandsmatrix op te stellen voor twee opeenvolgende frames. In deze matrix staan de afstanden tussen de punten gevonden in het eerste frame en het tweede frame.

In box 2 twee is een afstandsmatrix te zien van de punten gevonden in frame 1(B2-1) en frame 2 (B2-2). De afstanden (in pixels) zijn uitgezet in een matrix (B2-3). De functie findmatch gaat nu in de volgende stappen door de afstandsmatrix heen.

- Stap 1: De bovenste rij van de matrix wordt geselecteerd.
- Stap 2: De kleinste waarde in deze rij wordt gezocht(horizontale rode lijn)

Het kan voorkomen dat een punt in frame 2 heel dicht

bij twee punten in frame 1 ligt. Daarom wordt er ook gecontroleerd welk punt het dichtst bij het punt in frame 2 ligt.

- Stap 3: De kleinste rij van de kolom wordt gezocht(verticale rode lijn).
- Stap 4: De rij en kolom waarin gezocht is worden uit de matrix verwijderd. Er zijn namelijk succesvol twee punten aan elkaar gekoppeld.

Het voordeel van deze methode is dat zowel lineaire als circulair bewegende cellen zijn te volgen door de tijd en zelfs een combinatie van beide. Het pad wat de cel volgt door tijd staat aan de basis van de rest van de te bepalen eigenschappen. De locatie van de kop wordt namelijk gebruikt als startpunt voor de detectie van de staart en de cel oriëntatie. Het totale pad is nodig om de gevonden data te kunnen scheiden voor elke unieke cel.

3.6 Oriëntatie cel bepalen

 $f_{\rm x}$ tailstart (Appendix H), findangle (Appendix I)

Om metingen aan de staart te doen is het handig om een de oriëntatie van de cel te weten, hiermee kan ook een lokaal assenstelsel worden opgesteld. Het lokale assenstelsel wordt zo gekozen dat de lengtedoorsnede van de kop de x-as vormt (figuur 3.5). Om dit lokale assenstelsel te bepalen moet de hoek van het object in het globale assenstelsel bekend zijn. Het is dus van belang dat de cel oriëntatie in het globale assenstelsel wordt bepaald. Om deze oriëntatie te bepalen zijn twee vectoren nodig waarvan de hoek wordt berekend. In het geval van figuur 3.5 is dat de vector *a* en vector *b*. Om vector *a* te verkrijgen moet er eerst een



Figuur 3.4: Tracking resultaten uit vier verschillende video's. A) en B) zijn video's gemaakt met de Basler camera. C) en D) video's gemaakt met de Nikon camera.

Box 2: Afstandsmatrix



Voorbeeld* De gebruiker heeft opgegven dat punten een match vormen wanneer ze maximaal 20 pixels van elkaar verwijderd zijn. Voor punt #1 wordt gekeken of er punt in de buurt ligt. In de afstandmatrix wordt eerst horizontaal gekeken. Punt #a heeft met 1 de kleinste afstand. Er wordt voor punt #a gekeken of deze niet toevallig dichter bij een ander punt ligt. Dit is niet het geval. Punt #1 en #a worden aan elkaar gekoppeld. Ze kunnen maar één keer gekoppeld worden dus ze worden uit de matrix verwijderd. De matrix die overblijft is te zien in B2-3B.

Voor punt #2 (B2-3B) wordt er nu een match gezocht. Twee punten vallen binnen de opgegeven marge van de gebruiker, namelijk #b en #c. #c ligt het dichtste bij en wordt gekozen. Weer wordt er gekeken of #c ook daadwerkelijk het dichst bij #2 ligt. Dit is maar net het geval, want #3 ligt met 13 pixels ook erg dicht in de buurt. Punt #2 en #c worden gekoppeld en uit de afstandmatrix verwijderd(B2-3C).

In deze volgorde wordt de hele matrix rij voor rij afgewerkt, tot er geen combinaties meer mogelijk zijn. Wanneer er een punt uit frame 2 overblijft zonder te zijn gekoppeld met een punt van frame 1, dan zal deze als een nieuwe cel worden gedetecteerd en toegevoegd worden als een nieuw laatst bekend punt. Als er een punt uit frame 1 overblijft dan zal het laatst bekende punt ongewijzigd blijven. Dit is te zien in frame 1 en frame 2 voor punt #6. Doordat deze uit focus ligt is hij niet gedetecteerd in frame 2.

* De afstanden zijn verzonnen en komen niet overeen met de afbeelding.



B2-3 Afstandsmatrix Nieuw gevonden punten #e А #1 261 247 241 303 #2 247 15 б 120 126 -aatst bekend #3 3 260 13 121 120 2 #4 67 239 118 114 #5 304 67 118 121 1 #6 139 363 351 276 339 #b #d #c #e В #2 15 6 120 126 #3 3 13 121 120 Laatst bekend 114 2 67 118 #4 67 #5 118 121 1 351 363 276 339 #6 #b #d #e С #3 121 120 3 Laatst bekend #4 118 2 67 67 1 #5 118 339 363 276 #6 #d #e D #4 67 Laatst bekend 67 #5 1 276 339 #6



globale x-as

Figuur 3.5: Een grafische weergave van een cel in het globale assenstelsel. Het punt k is het punt wat gedetecteerd wordt door regionsprops. I wordt gevonden door tailstart. Met deze twee punten kan de celoriëntatie worden bepaald.

punt worden gezocht dat op de aanhechting van de staart ligt. Dit gebeurd door de functie tailstart toe te passen. De methode hierachter wordt in box 3 beschreven. Wanneer dit punt is gevonden kan de hoek worden uitgerekend (vergelijking 1) door het inproduct van beide vectoren te delen door de vermenigvuldiging van de magnitude van beide vectoren.

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} \tag{1}$$

De informatie van de hoek van de cel kan tevens gebruikt worden om te voorspellen wanneer een cel roteert of niet en wat de rotatiesnelheid is. Deze kennis kan worden gebruikt om te voorspellen wanneer een vrij zwemmende cel naar een getrapte toestand gaat.

3.7 Staart detecteren

 $f_{\rm x}$ showtail (Appendix K), findtail (Appendix J)

Staart isoleren

Om de rekentijd en het aantal fouten te reduceren worden eerst de staarten uit het frame gefilterd. Hiervoor wordt gebruik gemaakt van de video waarin de achtergrond is verwijderd (paragraaf 3.4). De randen van de staarten worden gedetecteerd door middel van een kernel. Hiervoor zijn verschillende kernels beschikbaar, zoals die van Sobel, Roberts of Prewitt. Er is gekozen voor Prewitt omdat deze gevoelig is voor randen gelegen in elke oriëntatie. Dit is gunstig aangezien de cellen veranderen van oriëntatie door de tijd heen. De gevonden randen worden nu met behulp van de Matlab Image Processing Toolbox bewerkt zodat deze een 'mask' vormen voor de staart. Dit gebeurd in de volgende stappen en het resultaat hiervan is in figuur 3.6 te zien.

- > 1.Videoframe wordt ingeladen.
- 2. Een Sobel kernel wordt gebruikt om de randen te detecteren
- 3. Objecten kleiner dan een door de gebruiker op gegeven waarden worden verwijderd.
- 4. Dilatie wordt toegepast om de gedetecteerde randen te verdikken
- 5. Eventuele gaten worden opgevuld.

Het verkregen masker (figuur 3.6-5) is een binaire afbeelding dus door deze te vermenigvuldigen met de originele video kunnen de staarten uitgeknipt worden (figuur 3.6-6).

Staart zoeken

De staart wordt per cel per frame individueel gezocht. De locatie van de kop wordt gebruikt als startpunt. Vanaf dit punt worden in een toenemende straal punten geselecteerd die in een cirkel om het startpunt liggen. De maximale waarde in een cirkel wordt gemarkeerd als een punt dat zich in de staart bevindt. De maximale waarde wordt zo voor elke straal bepaald en op deze wijze worden de punten van de staart geselecteerd.

In box 3 stapsgewijs te zien hoe de staart wordt gevonden. De coördinaten van de gevonden punten worden opgeslagen en zijn tevens de output van findtail. Deze coordinaten kunnen met behulp van findangle omgezet worden naar hoeken ten opzichte van de lokale x-as die door kop van de cel loopt zoals te zien in figuur 3.5.

3.8 Parameters

De detectie van de staart en de kop levert datapunten op waarvoor parameters kunnen worden opgesteld. Deze parameters geven objectieve informatie over de morfologie of motiliteit van de cel. De snelheid kan worden bepaald van progressieve cellen. Voor non-progressieve

Box 3: Staart zoeken



Vanaf de kop wordt er in een toenemende straal gezocht naar de staart. Dit gebeurd door punten te selecteren om de kop heen om hier vervolgens de maximale waarde van te berekenen. De staart zal namelijk het meest intensieve punt zijn wat op deze cirkel ligt. De gebruiker kan met *cr* (cirkel radiatie) opgeven hoeveel punten er om de kop heen moeten worden geselecteerd. De stapgrootte van de hoek wordt vervolgens opgedeeld in θ =(2 π /cr). Om de punten te selecteren wordt er voor de x- en de y-coördinaten van de kop (A) een extra afstand toegevoegd (B3-1). Voor de *x* is dat $x_{\theta_i} = x + r \sin(\theta)$ en voor de *y* is dat $y_{\alpha_i} = y + r \cos(\theta)$. De coördinaten voor het punt op de cirkel (*B*) kunnen nu gebruikt worden om de intensiteit van dit punt in de video te bepalen. Op deze manier wordt de intensiteit van alle punten op cirkel bepaald. In B3-2 is illustratief weergeven hoe de intensiteit verdeeld is over de cirkel van B3-1. Het meest intensieve punt stuk van de staart is aangegeven met een * en de videocoördinaten van dit punt worden opgeslagen. Deze methode wordt vervolgens iteratief toegepast voor een toenemende straal. In B3-3 is te zien hoe de staart wordt opgebouwd uit de verschillende gevonden coördinaten. De maximale straal waarin gezocht wordt moet de gebruiker handmatig opgeven.

cellen en getrapte cellen kan er een rotatiesnelheid worden opgesteld. Voor zowel progressieve, non-progressieve en getrapte cellen kan de kop worden opgemeten en kan de maximale hoekuitwijking die de staart maakt worden bepaald.

VSL

DeVSL kan worden uitgerekend door de afstand tussen het eerste en het laatste bekende punt van de cel op te meten. Deze snelheid is in pixels/s en kan worden geconverteerd naar μ m/s wanneer er een kalibratiemeting is gedaan. Deze parameter wordt gebruikt voor progressieve cellen.

VCL

De VCL wordt uitgerekend door de afstand tussen elk achtereenvolgde punt uit te rekenen. Wanneer er een punt ontbreekt zal deze genegeerd worden en het volgende punt worden geselecteerd. Deze snelheid is in pixels/s en kan worden geconverteerd naar µm/s wanneer er een kalibratiemeting is gedaan. Deze parameter wordt gebruikt voor progressieve en non-progressieve cellen.

Rotatie snelheid

De hoek van de celoriëntatie door de tijd heen kan gebruikt worden voor de rotatiesnelheid. Eerst is de functie unwrap nodig deze corrigeert de fase hoeken om zo een betere plot te produceren (figuur 3.9). Wanneer de data is gecorrigeerd kan er een eerste graads polynoom worden opgesteld om zo een lineaire fit te maken. Een afgeleide van deze fit geeft ons vervolgens de snelheid in *rad/frame*. De snelheid kan omgezet worden in *rad/s* door de snelheid te vermenigvuldigen met de framerate van de video. De rotatie snelheid kan worden gebruikt om te bepalen of de cel wel of niet roteert. Als de cel roteert kan ook worden bepaald in elke richting dit is. Deze parameters wordt gebruikt voor non-progressieve en getrapte cellen.

Maximale hoekuitwijking staart

Figuur 3.7 geeft illustratief weer hoe de staart zich verplaatst door de tijd heen. De maximale uitwijking die de staart maakt θ_1 en θ_2 zijn hierin aangegeven. Deze hoeken worden berekend ten opzichte van de lokale x-as zoals in figuur 3.5 gedefinieerd. In figuur 3.8 zijn de hoeken van de staart door de tijd heen geplot. Hierin is te zien hoe θ_1 en θ_2 hieruit kunnen worden bepaald.

Kop afmetingen

De functie regionprops geeft drie parameters van de gevonden objecten; de lengte, de breedte en de oppervlakte. Om de objecten wordt door de functie een ellips gevormd. Van deze ellips worden de twee hoofdassen bepaald. De langste as wordt als lengte van de cel kop gebruikt, de kleinste as als de breedte en de oppervlakte als oppervlakte van de kop.



Figuur 3.6: 1) Orgineel 2) Prewitt edge detection 3) Kleine objecten worden gefilterd 4) Dilatie 5) Gaten worden opgevuld 6Soalleen het mask



Figuur 3.7: Illustratieve weergave van de beweging van de staart door de tijd heen.



Figuur 3.8: Plot met daarin meerdere posities van de staart door de tijd heen. De maximale hoek is aan gegeven met beide stippellijnen en komen overeen met figuur 3.7.



Figuur 3.9: Oriëntatie van vier verschillende cellen door de tijd heen. De functie unwrap is gebruikt om te corrigeren voor de fase sprongen. De zwarte stippellijnen zijn eerstegraads polynomen dit gefit zijn. De afgeleide van deze polynoom geeft de rotatie snelheid.

Resultaten

De methoden en de parameters zijn getest en geanalyseerd. De resultaten hiervan zullen worden gepresenteerd. De methoden en parameters die betrekking hebben op de morfologie of motiliteit worden besproken in paragraaf 4.2 en 4.3 Allereerst zal de detectie van de kop worden besproken. De detectie van de kop is afhankelijk van twee parameters. De invloed van deze parameters op de detectie is geanalyseerd.

4.1 Kop detectie

De functie isolateheads bevat twee parameters. De minimale detectiegrootte (*p*) van een kop en de threshold (*t*). Het is belangrijk dat deze parameters niet teveel variëren binnen een dataset (video's in zelfde condities opgenomen). Dit heeft namelijk als voordeel dat video's achter elkaar kunnen worden geanalyseerd zonder dat hiervoor telkens de parameters aangepast hoeven te worden. Verder is het ook belangrijk dat de twee parameters niet te gevoelig zijn. Een hele gevoelige parameter heeft als nadeel dat de gebruiker veel pogingen moet doen om het gewenste resultaat te verkrijgen. In afbeelding 4.1 is gekeken hoe de gemiddelde kopgrootte veranderd wanneer de twee parameters veranderen. Dit is gedaan voor één dataset, namelijk die van de Basler (tabel 3.1). Deze dataset bestaat

uit zes video's die met dezelfde vergroting en resolutie zijn gemaakt. Op deze manier is het verschil in kopgrootte tussen verschillende video's minimaal. Deze dataset bezit in totaal 23 bewegende cellen. Op de x-as is de threshold te zien en op de y-as de gemiddelde kopgrootte van alle 23 cellen uit de Basler dataset. De dataset is voor p=1, p=5, p=10, p=15 en p=20 uitgerekend. Te zien is dat de minimale detectiegrootte nauwelijks invloed heeft op de kopgrootte en dat de kopgrootte lineair afneemt tussen de 0.3 en 0.7. Dat de kopgrootte afneemt is te verklaren doordat de threshold steeds groter wordt. Het midden van de spermakop is in de video het meest intense punt en naarmate de threshold stijgt vallen er steeds meer pixels die om dit punt liggen onder deze threshold (figuur 4.2).

In figuur 4.2a is de gekeken naar wat de optimale threshhold is voor verschillende parameters p. De optimale threshhold is wanneer er zoveel mogelijk cellen positief gedetecteerd worden. Te zien is dat wanneer p stijgt de optimale threshhold bij een steeds lagere t wordt gevonden. Verder valt het op dat de threshold optimaal blijft in een gebied van 0.10-0.15.

In figuur 4.2b is de invloed van t en p te zien op het totaal



Figuur 4.1: Threshold vs kopgrootte in pixels



Figuur 4.2: a) % correct gedetecteerde cellen in de Basler dataset (100% is 23 cellen). b) Het totaal aantal gedecteerde objecten.

aantal gedetecteerde objecten. Hoe lager t hoe meer objecten er worden gedetecteerd. Een drempel is waar te nemen bij elke p waarna het aantal gedetecteerde objecten drastisch omlaag gaat. Bij een minimale objectgrootte van 1px (p=1) is er een hoge threshhold nodig om de ruis te verwijderen. Naarmate p groter wordt is er steeds minder thresholding nodig aangezien het aantal de objecten die ruis vormen niet erg groot zijn.

4.2 Morfologie

De software is in staat om drie morfologische kenmerken te detecteren of te kwantificeren. Het eerste kenmerk is de kop. Hiervan kan de lengte, breedte en oppervlakte worden opgemeten. Een tweede gedetecteerde kenmerk zijn cytoplasmatische druppels op de staart. Een derde kenmerk is de detectie van de staart. Deze zal in het begin van paragraaf 4.3 worden besproken.

Kop afmetingen

Van twee verschillende datasets zijn de afmetingen van de cellen bepaald. De Basler dataset met een resolutie van 780x572 en de Nikon dataset met een resolutie van 640x512. In de Basler dataset zijn in totaal 23 cellen gedetecteerd en opgemeten en in de Nikon dataset 61 cellen. Zoals in paragraaf 4.1 is vast gesteld heeft de minimale detectiegrootte weinig invloed, deze is op 25 ingesteld. Uit figuur 4.1 blijkt dat de afmetingen van de cel beïnvloed wordt door de threshold. Het is daarom van belang om een threshold te zoeken die de dimensies van de kop zo goed mogelijk in stand houdt, dit is gedaan door de beste overlay te vinden voor de video zoals te zien in figuur 3.2. De cellen zijn ook met de hand opgemeten met het programma ImageJ.

In tabel 4.1 zijn de resultaten van de metingen van de Basler en Nikon dataset. De cellen uit de Basler dataset zijn ook handmatig opgeteld. Tevens zijn er waardes toegevoegd uit een onderzoek van Kondreciki et al. (2012). Dit zijn afmetingen van de spermatozoa van het varkensras Duroc.

Tabel 4.1: Gemiddelde afmetingen van de spermakop.

Dataset	Lengte (µm)	Breedte (µm)	Oppervlakte (µm)	Aantal cellen	р	t
Nikon (640x512)	10.49 ± 2.08	3.72 ± 1.01	30.42 ± 7.70	61	15	0.22
Basler (780x572)	15.32 ± 4.77	4.79 ± 1.46	55.59 ± 17.89	23	25	0.3
Nikon (640x512) hand gemeten	7.99 ± 0.74	4.18 ± 0.40	32.10 ± 4.76	10	-	-
Basler (780x572) hand gemeten	8.67 ± 1.20	4.53 ± 0.71	36.05 ± 8.05	10	-	-
Kondreciki et al, 2012	9.31 ± 0.39	4.69 ± 0.18	40.64 ± 11.18	90.81 x 10 ⁹	-	-



Figuur 4.3: Consecutieve reeks van een rechts roterende vrijzemmende cel na het uitvoeren van de functie findtail. Te zien is dat de cel een assymetrische slag maakt met zijn staart.

Cytoplasmatische druppels

Tijdens de analyses van de video's is het opgevallen dat na het tracken van een cel er soms twee gelijkvormige paden door elkaar lopen(Figuur 4.4C). Eerst werd aangenomen dat dit door de detectie van ruis kwam. De *p* werd dan vervolgens verhoogd om de ruis te negeren en het resultaat was dat er vervolgens nog maar één pad overbleef.

Gedurende het onderzoek is er geconstateerd dat er cellen met cytoplasmatische druppels op de staart aanwezig zijn in de samples (figuur 2.1n) die de dubbele tracking paden veroorzaakten. Dit komt doordat de intensiteit van de druppels erg hoog is en vergelijkbaar is met die van de kop van de cel. Het tracking algoritme detecteert de druppel als een individuele cel en ken deze daarom een eigen pad toe. Met deze kennis is er gekeken of deze morfologische afwijking te detecteren is door gebruik te maken van het feit dat de druppel een eigen pad krijgt toegewezen die sterk op het pad van de kop lijkt.

De druppel van de staart is erg klein dus de parameter

p zal voorzichtig moeten worden gekozen. Een kleine p heeft als nadeel dat er veel last is van ruis. De druppel op de staart heeft een hoge intensiteit en daarom kan er een hoge threshold worden gekozen. Dit reduceert het aantal gedetecteerde objecten enorm (figuur 4.2B). In figuur 4.4 zijn twee video's te zien waar zich cellen in bevinden met en zonder druppels op de staart. Vervolgens is er een kopdetectie uitgevoerd met parameters t=0.60 en p=5 (figuur 4.4B). Te zien is dat de druppels gedetecteerd worden en dat deze vervolgens getrackt kunnen worden (figuur 4.4C en D). Door de gevonden paden met elkaar te vergelijken kan er worden bepaald of twee paden over elkaar heen liggen. Op deze manier zijn er 9 cellen gedetecteerd met een cytoplasmatische druppel in de Basler dataset. Na handmatige analyse is vastgesteld dat er 8 druppels positief zijn geanalyseerd. Eén cel is foutief gedetecteerd, omdat niet vast te stellen was of de cel een druppel bevatte. Deze cel is in figuur 4.4 aangeven met een '*'. Van de zes video's was er in elke video minimaal 1 cel met een druppel. De parameters t en p zijn onveranderd gebleven voor alle video's.

Tabel 4.2:Resultaten na analyse van cytoplasmatichedruppel detectie

Cellen met	Positief	Foutief	Niet
druppel	gedetecteerd	gedetecteerd	gedetecteerd
8	8	1	0



Figuur 4.4: Frame uit twee verschillende video's van de Basler dataset waar cellen met druppels op de staart zijn te zien. B) Resultaat na een kopdetectie met t=0.60 en p=5; de druppels zijn nog steeds zichtbaar. De videoframes in A bevatten nog de statische ruis. Hierdoor zijn er dode cellen te zien die in B verdwijnen. C) De druppels worden succesvol getrackt. Hierdoor onstaan er twee paden die op elkaar liggen. D) Uitvergroting van de paden van de cellen die een druppel op de staart hebben. Een cel bevat geen druppel en is gemarkeerd met een '*'.



Figuur 4.5: A) Links roterende cel B) Rechts roterende cel

4.3 Motiliteit

Maximale hoekuitwijking van de staart

In figuur 4.3 zijn 10 consecutieve frames te zien waarin de gevonden punten van de staart zijn geplot. De staart wordt niet volledig tot aan de punt gedetecteerd dit komt doordat de maximale straal is ingesteld op 40. Dit is kwalitatief bepaald, omdat hoe groter de straal wordt des te minder intensief de staart. De verlaging in intensiteit zorgt ervoor dat de kans op fouten steeds groter wordt.

Twaalf cellen uit de Basler dataset zijn gebruikt voor de analyse van de maximale hoekuitwijking. Deze cellen zijn opgedeeld in drie categorieën: lineair progressief, rechts roterend en link roterend. De maximale hoekuitwijking is bepaald door 60 frames lang de hoek van de staart te plotten. In totaal bevat iedere staart 30 data punten. De straal waarin gezocht is *rmin=10* en *rmax= 40*. Voor elk punt op de staart is de hoek ten opzichte van het lokale assenstelsel berekend. Vervolgens is handmatig vastgesteld wat de maximale uitwijking θ_1 en θ_2 is.

Het bepalen van de maximale uitwijking levert op dat er verschillen zijn waar te nemen tussen de drie categorieën(tabel 4.3). Zo heeft een rechts roterende cel een grote θ_1 en een kleine θ_2 . Wanneer men nogmaals goed naar figuur 4.5 kijkt is dit ook te zien. In frame 3, 6, 9 en 10 slaat de staart veel verder uit dan in frame 1, 4, en 7. Omgekeerd heeft een links roterende cel een grote θ_2 en een kleine θ_1 . Een lineair progressieve cel heeft vrijwel een even grote θ_1 als θ_2 . Dit doet de lineaire cel niet door gelijkmatig de staart in beide richtingen even ver uit te slaan. Waargenomen is dat een lineaire cel een bepaalde tijd zich gedraagt als een links- of rechtsom roterende cel. Dat wil zeggen dat waar te nemen is dat de cel gedurende een tijd een grote θ_1 heeft en een kleine θ_2 . Na een periode wisselt de cel van slag en draaien de waarde van θ_2 en θ_1 om. Het patroon lijkt op een slag die men in een kano

maakt. Hierbij roeit men eerst een aantal slagen aan de linker kant van de kano om vervolgens te wisselen en even veel slagen aan de rechter kant te maken. Hierdoor blijft de kano rechtuit varen.

Het patroon van de lineaire cel is weer gegeven in figuur 4.7 voor 60 frames. Duidelijk is te zien dat in de eerste 30 frames de cel een grotere θ_1 heeft. Na 30 frames wisselt het patroon en krijgt de cel een vrijwel evengrote θ_2 . Ook de intermediaire hoeken θ_1 en θ_2 zijn vrijwel even groot. Deze manier van zwemmen kan de zigzaggende beweging van lineaire cellen verklaren zoals te zien in de figuren van 3.4 en 4.4. Er is niet waargenomen of een klein verschil tussen $|\theta_1|$ of $|\theta_2|$ er voor zorgt dat een lineaire cel langzaam afbuigt. Verder is er gekeken wat gemiddeld de

totale hoek van de drie categorieën is: $|\theta_1| + |\theta_2|$.

Cel tracking

Het is belangrijk dat wanneer een cel is gedetecteerd dat deze door de tijd heen in zoveel mogelijk frames wordt getrackt. Er is daarom gekeken voor positief gedetecteerde cellen in hoeveel frames deze zijn gedetecteerd en getrackt. Hier voor zijn de thresholds gebruikt die een zo hoog mogelijk percentage positief gedecteerde cellen opleveren. Dit is gedaan op basis van de resultaten van figuur 4.2a. De resultaten zijn weergeven in tabel 4-3. Uit de tabel is op te maken dat wanneer een cel eenmaal gedetecteerd is dat deze in elk frame te volgen is en met behulp van findmatch succesvol gekoppeld kan worden.

De tracking is verder vergeleken met bestaande opensource CASA software (Wilson-Leedy, Jonas G and Ingermann, 2007). Deze is als een plugin voor ImageJ te downloaden en geeft als output de gemiddelde VCL en VSL van de cellen in de video en een afbeelding met daarin de getrackde cellen. In de figuur 4-4 is te zien dat de tracking van de eigen software en die van de open-source dezelfde baan beschrijven Niet alleen is de baan hetzelfde,



Figuur 4.6: Hoekuitwijking van een lineair progresieve cel. A) frame 1-30 B)frame 30-60 C) frame 1-60

Tabel 4.3: Maximale hoek uitwijking θ_1 en θ_2 voor cellen die link- of rechtsom roteren en lineair progressieve cellen.

Zwem gedrag	Gemeten cellen	$\theta_1 \pm std$	$\theta_2 \pm std$	$ \theta_1 + \theta_2 \pm std$	
Links roterend	3	0.36 ± 0.15	-1.18 ±0.17	1.55	
Rechts roterend	4	1.51 ±0.19	-0.43 ±0.15	1.94	
Lineair	5	0.83 ±0.15	-0.81 ±0.28	1.64	

ook de hoge frequentie die op de lijn van het pad te zien is het zelfde.

Met behulp van de tracking is het ook mogelijk om de standaard CASA parameters VSL, VCL en LIN te bepalen. Dit is gedaan voor de Nikon en de Basler dataset. De parameters van de halve Basler dataset zijn ook berekend door de open-source CASA software. De resultaten hiervan zijn te zien in tabel 4.6

Transitie detectie

Een eis van het systeem is dat het zowel getrapte als mobiele cellen kan tracken. In figuur 4.7 zijn de paden te zien van verschillende getrapte cellen. Te zien is dat cel nummer zeven eerst vrij rondzwemt en op een bepaald tijdstip op de trap blijft zitten en begint te roteren. In de data is het moment van trappen te zien. In figuur 4.8 is de hoek van de cel over de tijd geplot. Wanneer deze tijd vergeleken wordt met de daadwerkelijke video wordt duidelijk dat de cel inderdaad rond frame 400 getrapt wordt. De parameters die de cel heeft tijdens deze twee fasen zijn te zien in tabel 4.4. Wat opvalt is dat de VLC en de VLS in de eerste fase ongeveer twee keer zo groot is als die van de tweede fase. De lineariteit blijft hierdoor ook vrijwel hetzelfde.

Rekentijd

Er is bijgehouden voor elke resolutie wat de gemiddelde rekentijd is voor één seconde video. Deze tijden zijn berekend met een systeem dat beschikt over een Intel Core 2 Duo E8400 processor, EVGA GTX-480 videokaart en 4GB werkgeheugen (figuur 4.10). Logischerwijs neemt de rekentijd toe naarmate de resolutie toeneemt. Op alle video's zijn cellen te detecteren en te tracken. Het voordeel van een hoge resolutie is dat metingen aan de kop worden nauwkeuriger worden naarmate er meer pixels zijn. Voor het gebruikte systeem zijn voor alle metingen video's gebruikt met een resolutie van 640x512 en 780x572, omdat deze een goed compromis vormen tussen rekentijd en de hoeveelheid pixels. Een systeem met betere specificaties zou een hogere video's met een hogere resolutie sneller kunnen uitrekenen alleen de meerwaarde hiervan is niet onderzocht.

Kalibratie

Alle bewerkingen van de software zijn in pixels, omdat de video's hieruit zijn opgebouwd. Om de software te kalibreren is er een meting gedaan. Voor beide camera's is er voorafgaand een foto gemaakt van een objectglas met daarop een schaalverdeling van in totaal 1mm. De kleinste verdeling op de schaal was 1 micron. Tien maal is een stap van 1 micron opgemeten met de software ImageJ en hiervan is het gemiddelde genomen. Voor de Basler camera is bepaald dat 10 µm overeenkomt met 8.044 pixels. Voor de Nikon is bepaald dat 10 µm overeenkomt met 7.625 pixels. Tabel 4.4: Tabel van Basler dataset. Voor diverse p is onder een optimale t het tracking percentage bepaald. (100% is 1225 frames)

Minimale kopgrootte(px)	Cellen gedetecteerd	Tracking %
p=1	20/23	0.9956
p=5	23/23	0.9949
p=10	23/23	0.9994
p=15	23/23	0.9984
p=20	23/23	0.9982



Figuur 4.7: Twee cellen (5) en (10) die vast zitten op een fibronectine spot. Cel 7 zwemt eerst even vrij rond voordat deze uiteindelijke ook getrapt raakt.



Figuur 4.8: Orientatie van de cel door de tijd heen. De plot is opgedeeld in twee gedeeltes; vrij en getrapt.

Tabel 4.5: Parameters van de cel tijdens de vrije fase (1-400)en de getrapte fase.

Frame	VSL (µm/s)	VCL (µm/s)	LIN	Rotatiesnelheid (°/s)
1-400	2.20	55.92	0.0493	9.85
400-800	1.09	28.04	0.0392	152.8



Figuur 4.9: Overlay van tracking resultaten. In kleur zijn de verschillende paden te zien die met behulp van de eigen methode zijn bepaald. In zwart zijn de paden te zien die bepaald zijn met behulp van open-source CASA software.

Tabel 4.6: Standaard CASA parameters van de motiliteit voor de Nikon en de Basler dataset.

	VSL (µm/s) ± std	VCL (μm/s) ± std	LIN	Aantal cellen	Р	t
Nikon (640x512)	8.52 ± 7.87	20.71 ± 11.29	0.30	61	15	0.22
Basler (780x572)	14.64 ± 14.25	73.69 ± 33.07	0.17	23	25	0.3
Open-source CASA software voor Basler (780x572) dataset	27.06 ± 7.17	76.19 ± 5.41	0.28	12	-	-



Rekentijd voor totale analyse van 1 seconde video

Figuur 4.10: Grafiek met daarin de rekentijd voor een totale analyse van 1 seconde video. Dit is gedaan voor drie resoluties.

5 Discussie en aanbevelingen

5.1 Data

De video's zijn gemaakt met een vergroting van 10x. Dit maakt de detectie van vacuolen in de nucleus niet mogelijk terwijl onderzoek aantoont dat er een correlatie is tussen de hoeveelheid vacuolen en de DNA-fragmentatie. Om een vergroting te halen die dit wel kan is een andere opstelling nodig. Echter een opstelling zoals de huidige is noodzakelijk, omdat deze veel cellen tegelijkertijd kan analyseren en een groot deel kan afkeuren op basis van de motiliteit. De ideale situatie is een opstelling die beide analyses tegelijk kan uitvoeren. Het systeem met de lage vergroting zou het andere systeem kunnen vertellen welke cellen de moeite waard zijn om onder een hoge vergroting te analyseren.

5.2 Methode en resultaten

Kop detectie

De huidige methode is in staat om de koppen van spermacellen te detecteren. Voor één dataset is onderzocht hoe parameters *p* en *t* de detectie beïnvloeden. Om het systeem volautomatisch te maken zal deze handmatige invoer van de threshold moeten verdwijnen. Er zal dan automatische thresholding moeten plaatsvinden. Dit kan door de gebruiker in één video willekeurige cellen te laten selecteren. Het algoritme zal dan thresholds blijven proberen tot alle opgegeven cellen met een zo optimaal mogelijke threshold zijn gedetecteerd.

Kop afmetingen

De automatisch gevonden lengte, oppervlakte en standaard deviaties van de Basler dataset wijken af van de met hand opgemeten cellen. Dit kan komen doordat de threshold te laag is ingesteld. Hierdoor worden stukken staart meegerekend met de kop. Dit kan verklaren waarom de lengte en de oppervlakte groter is terwijl de breedte niet afwijkt. De afmetingen van de Nikon dataset komen overeen met die van de literatuur. De standaard deviaties zijn alleen hoog, maar dit komt doordat er maar 61 cellen zijn gemeten in tegenstelling tot de 9 miljard van de literatuur. Ditzelfde geldt voor de met de hand opgemeten cellen. Een nadeel van het systeem is dat de afmetingen van de cel afhankelijk zijn van de ingestelde threshold. Hierdoor zijn de metingen niet subjectief, omdat de gebruiker op het oog moet inschatten of het gewenste resultaat is bereikt. Zoals hierboven aangeven kan dit opgelost worden door het systeem volautomatisch de threshold te laten bepalen, hierdoor wordt de menselijk fout geëlimineerd en kan het systeem gekalibreerd worden.

Staart detectie

De staart kan worden gedetecteerd en worden opgeslagen in coördinaten. Deze data punten worden gebruikt om de maximale hoekuitwijking te bepalen. De detectie is alleen afhankelijk van een opgeven maximale straal waarin gezocht wordt. Hierdoor is het meten van de staart niet mogelijk. De methode selecteert slechts één punt per straal. Wanneer de staart een boog beschrijft van 180 graden liggen punten van de staart tweemaal op dezelfde straal. Hiervan wordt slechts één geselecteerd. Ook is de kans op een foutief gedetecteerd punt mogelijk. De toenemende straal waarin gezocht wordt maakt de kans groter en ook de magnitude van de fout. Een foutief geselecteerd punt kan namelijk steeds verder van de staart afliggen naarmate de straal groter wordt. De kans op een fout is verkleint door de staart te isoleren van de video, echter de huidige methode is niet in staat om fouten te filteren. Er is gewerkt aan een methode waarin de foutenmarges kleiner zijn en die automatisch de punt van de staart zoekt. Deze methode is gebaseerd op een dynamic programming algoritme. Dit algoritme berekend de optimale route tussen twee punten. Voor de staart zal dan een route moeten worden gezocht waarvan de som van de intensiteit het hoogst is.

Cytoplasmatische druppels

Druppels hebben een hoge intensiteit en worden daarom als koppen gedetecteerd. Dit heeft als gevolg dat er twee keer een getrackt pad te zien is. Door gebruik te maken van dit 'ongewenste' effect kunnen druppels op de staart worden gedetecteerd. Dit sluit niet uit dat ook andere abnormaliteiten dit effect kunnen creëren. Eén cel is gedetecteerd zonder een druppel op de staart. De cel bleek wel een ander defect te vertonen. De methode zal dus misschien niet nauwkeurig genoeg zijn om alleen cellen met druppels te detecteren maar wel om abnormale cellen te detecteren. Cellen met enige afwijking worden namelijk als niet geschikt gezien voor ICSI of IVF.

Maximale hoekuitwijking van de staart

Er is aangetoond dat de maximale hoek van de staart cellen in drie categorieën kan opdelen. Rechts roterende, links roterende en lineair progressieve. Een lineaire cel heeft eerst een sterke uitwijking naar één kant om na een periode te wisselen naar een sterke uitwijking naar de andere kant. Er zijn te weinig cellen gemeten om te bepalen of de totale hoek ($\theta_1 \text{ en } \theta_2$) correleert met de hoeksnelheid of de VCL. De resultaten hebben ook niet laten zien of een kleine verschil tussen θ_1 en θ_2 bepaald of een lineaire cel langzaam afbuigt.

De methode kon niet worden toegepast op de video's waarop getrapte cellen zijn te zien. De cellen lagen vaak niet in focus, de intensiteit van het licht veranderde gedurende de opname. Hierdoor kon de statische achtergrond niet geheel worden verwijderd. Dit probleem kan worden verholpen door nieuwe opnames te maken. Getrapte cellen kunnen een uitwijking hebben die overeenkomt met één van de drie categorieën. Als deze classificatie ook toe te passen is voor getrapte cellen kan men bepalen wat het zwempatroon in een vrije oplossing is. Ook kan er worden gekeken wat de gemiddelde rotatiesnelheid per categorie is. Onderzoek van Frimat et al. (2012) suggereert namelijk dat een hoge rotatiesnelheid een indicatie kan zijn voor hoog motiele cel.

Cel tracking

Het tracken van de cellen gebeurd vrijwel in elk frame. Hierbij worden telkens de juiste cellen aan elkaar gekoppeld.

De video met de meeste cellen binnen een frame bevatte 14 cellen. Een commercieel CASA-systeem detecteert en trackt meer dan 100 cellen binnen een frame (Origo, 2011). De detectie op een hogere concentratie cellen zal moeten worden onderzocht. Spermacellen die door elkaar zwemmen of vlak naast elkaar zwemmen kunnen namelijk foutief toegewezen worden aan een bestaand pad. Verder is het systeem niet in staat om een cel die het frame verlaat en later terugkeert te identificeren als dezelfde cel. Beide problemen kunnen worden gereduceerd door de richting van het pad te bepalen. Hierdoor zullen cellen die elkaar kruizen niet verkeerd worden toegewezen en kan worden voorspeld wanneer de cel weer het frame in zal komen. Cellen kunnen getrackt worden in zowel getrapte als vrije toestand. Ook is een combinatie van beiden mogelijk. Aangetoond is dat te detecteren valt wanneer een cel van toestand verandert. Het biedt de mogelijkheid om te cel te analyseren in twee verschillende fasen. Dit kan worden gebruikt om te bepalen of de getrapte cel een roterende of lineair progressieve cel is. Deze informatie is belangrijk voor verder onderzoek naar de maximale hoekuitwijking.

6 Conclusie

Een systeem is ontwikkeld dat in staat is spermacellen te detecteren en te tracken in zowel mobiele als getrapte toestand.

(A) Het systeem is in functies opgedeeld en is makkelijk uit te breiden. De code is gemakkelijk te begrijpen aangezien de gebruiker een overzichtelijke flowchart heeft van de functies. De totale berekening inclusief het laden van de video duurt 2.3s per seconde video. Functies geven feedback wanneer ze bezig zijn zodat de gebruiker kan zien wanneer de functie klaar is.

(B) Met juist gekozen parameters is er een celdetectie van 100% behaald en een tracking van ~100% in alle frames van een dataset bestaande uit zes video's met in totaal 23 cellen. Door achtergrondsubstractie is er geen last van helderheid verschillen of statische ruis en worden doden cellen gefilterd.

C) De cellen zijn individueel te detecteren en de orientatie kan worden bepaald door het begin van de staart te zoeken. De celorientatie kan worden gebruikt om te registreren wanneer een cel getrapt wordt.

(D) Staartdetectie is mogelijk en de maximale uitwijking van de staart is te bepalen. Er is vastgesteld dat met de maximale uitwijking aangetoond kan worden of een cel zich lineair of circulair verplaatst. Er is kwalitatief bepaald dat deze methode nog fouten bevat en dat er een fouten filter zal moeten worden ontwikkeld.

(E) De staart is opgebouwd uit verschillende datapunten en kan per cel individueel worden bepaald.

(F) Drie standaard CASA parameters (VLC, VLS en LIN) zijn te berekenen uit de tracking resultaten. De lengte, breedte en oppervlakte van de kop is op te meten. Echter deze is afhankelijk van de ingestelde threshhold **t**en de afmetingen worden dus niet objectief bepaald. De rotatiesnelheid kan worden berekend van een cel wanneer deze getrapt is of wanneer deze een circulaire baan aflegd. Ook kan de richting waarin de cel draait worden bepaald.

A. Gestructureerde en gebruikersvriendelijke code

- » Code moet makkelijk uit te breiden zijn.
- » Code moet makkelijk te begrijpen zijn.
- » Analyse moet vlot verlopen(1 ~ 2 minuten).

B. Detectie van spermatozoa

- » Ongevoelig zijn voor contrast en helderheid.
- » Dode cellen moeten gefilterd kunnen worden.
- » Moet getrapte en vrij zwemmende spermatozoa kunnen detecteren.

C. Volgen van spermatozoa over de tijd

- » De locatie en oriëntatie moet worden opgeslagen.
- » Het afgelegde pad moet per cel worden bepaald.
- » Moment van trappen registreren.

D. Detectie van de staart

- » De vorm van de staart moet zo goed mogelijk behouden blijven.
- » Moet de staart van getrapte en vrij zwemmende spermatozoa kunnen detecteren.

E. Volgen van de staart over de tijd

- » Meerdere locatiepunten moeten worden opgeslagen.
- » De gedetecteerde staart moet gekoppeld worden aan de juiste cel.

F. Parameters opstellen

- » De grootte van de kop en de lengte van de staart moet worden bepaald.
- » De VLS moet worden bepaald.
- » Wanneer getrapt moet de rotatiesnelheid worden bepaald.
- » De maximale uitwijking van de staart moet worden bepaald.

Referenties

Antinori, M., et al. (2008). "Intracytoplasmic morphologically selected sperm injection: a prospective randomized trial." Reproductive biomedicine online 16(6): 835-841.

Auger, J. (2010). "Assessing human sperm morphology: top models, underdogs or biometrics." Asian J Androl 12(1): 36-46.

Bartoov, B., et al. (2003). "Pregnancy rates are higher with intracytoplasmic morphologically selected sperm injection than with conventional intracytoplasmic injection." Fertility and sterility 80(6): 1413-1419.

Berkovitz, A., et al. (2006). "Does the presence of nuclear vacuoles in human sperm selected for ICSI affect pregnancy outcome?" Human Reproduction 21(7): 1787-1790.

Bijar, A., et al. (2012). "Fully automatic identification and discrimination of sperm's parts in microscopic images of stained human semen smear." Journal of Biomedical Science and Engineering 5: 384-384.

Cohen-Bacrie, P., et al. (2009). "Correlation between DNA damage and sperm parameters: a prospective study of 1,633 patients." Fertility and sterility 91(5): 1801-1805.

Cooper, T. G., et al. (2009). "World Health Organization reference values for human semen characteristics." Human reproduction update: dmp048 %@ 1355-4786-4786.

Darwish, A. M. M. (2012). Enhancing Success of Assisted Reproduction, Intech.

Doyle, P. (1999). "The UK Human Fertilisation and Embryology Authority." International journal of technology assessment in health care 15(01): 3-10.

Friedrich, B. M., et al. (2010). "High-precision tracking of sperm swimming fine structure provides strong test of resistive force theory." The Journal of experimental biology 213(8): 1226-1234.

Frimat, J. P., et al. (2014). "Make it spin: individual trapping of sperm for analysis and recovery using micro-contact printing." Lab on a Chip.

Gaffney, E. A., et al. (2011). "Mammalian sperm motility: observation and theory." Annual Review of Fluid Mechanics 43: 501-528.

Garolla, A., et al. (2008). "High-power microscopy for selecting spermatozoa for ICSI by physiological status." Reproductive biomedicine online 17(5): 610-616.

Huang, C.-C., et al. (2005). "Sperm DNA fragmentation

negatively correlates with velocity and fertilization rates but might not affect pregnancy rates." Fertility and sterility 84(1): 130-140.

Hughes, E. G. (1997). "The effectiveness of ovulation induction and intrauterine insemination in the treatment of persistent infertility: a meta-analysis." Human Reproduction 12(9): 1865-1872.

Jouannet, P., et al. (1988). "Male factors and the likelihood of pregnancy in infertile couples. I. Study of sperm characteristics." International journal of andrology 11(5): 379-394.

Knez, K., et al. (2011). "The IMSI procedure improves poor embryo development in the same infertile couples with poor semen quality: a comparative prospective randomized study." Reprod Biol Endocrinol 9: 123-123.

Kondracki, S., et al. (2012). "Comparative analysis of Duroc and Pietrain boar sperm morphology." Acta Veterinaria Brno 81(2): 195-199.

Kotwicka, M., et al. (2007). "Human spermatozoa ultrastructure assessment in the infertility treatment by assisted reproduction technique." Systems Biology in Reproductive Medicine 53(6): 297-302.

Larsen, L., et al. (2000). "Computer-assisted semen analysis parameters as predictors for fertility of men from the general population." Human Reproduction 15(7): 1562-1567.

Nadalini, M., et al. (2009). "Impact of intracytoplasmic morphologically selected sperm injection on assisted reproduction outcome: a review." Reproductive biomedicine online 19: 45-55.

Nieschlag, E., et al. (2001). Andrology: Male reproductive health and dysfunction, Springer.

Organization, W. H. and others (2010). "WHO laboratory manual for the examination and processing of human semen."

Otsu, N. (1975). "A threshold selection method from gray-level histograms." Automatica 11(285-296): 23-27.

Peluso, G., et al. (2013). "The study of spermatic DNA fragmentation and sperm motility in infertile subjects." Archivio Italiano di Urologia e Andrologia 85(1): 8-13.

Pinkhof, H., et al. (2012). Pinkhof geneeskundig woordenboek, Bohn Stafleu van Loghum.

Piomboni, P., et al. (2012). "The role of mitochondria in

energy production for human sperm motility." International journal of andrology 35(2): 109-124.

Rozeboom, K. J. (2000). "Evaluating boar semen quality." Animal Science Facts. Extension Swine Husbandry. College of Agriculture & amp; Life Sciences. North Carolina State University: 1-8.

Saladin, K. S. (2010). Anatomy & amp; physiology: the unity of form and function, McGraw-Hill.

Setti, A. S., et al. (2013). "Twelve years of MSOME and IMSI: a review." Reproductive biomedicine online 27(4): 338-352.

Smith, D. J., et al. (2009). "Bend propagation in the flagella of migrating human sperm, and its modulation by viscosity." Cell Motil Cytoskeleton 66(4): 220-236.

Stauss, C. R., et al. (1995). "Sperm motility hyperactivation facilitates penetration of the hamster zona pellucida." Biology of reproduction 53(6): 1280-1285.

Suarez, S. S., et al. (1991). "Evidence for the function of hyperactivated motility in sperm." Biology of reproduction 44(2): 375-381.

Turner, R. M. (2005). "Moving to the beat: a review of mammalian sperm motility regulation." Reproduction, Fertility and Development 18(2): 25-38.

Umcg (2010). IVF en ICSI.

Van der Zwalmen, P., et al. (1991). "Sperm morphology and IVF pregnancy rate: comparison between Percoll gradient centrifugation and swim-up procedures." Human Reproduction 6(4): 581-588.

Verstegen, J., et al. (2002). "Computer assisted semen analyzers in andrology research and veterinary practice." Theriogenology 57(1): 149-179.

Wilding, M., et al. (2011). "Intracytoplasmic injection of morphologically selected spermatozoa (IMSI) improves outcome after assisted reproduction by deselecting physiologically poor quality spermatozoa." Journal of assisted reproduction and genetics 28(3): 253-262.

Yang, H.-F., et al. (2014). "Head tracking and flagellum tracing for sperm motility analysis."

Zinaman, M. J., et al. (2000). "Semen quality and human fertility: a prospective study with healthy couples." Journal of Andrology 21(1): 145-153.

Appendix

A. Tabel met videodata

#	Camera	Gevraagde FPS	Daadwerkelijke FPS	Filter	Resolutie	Naam	Beweging
	Nikon	25	16.6	Groen	1280x1024	25fps 1280-1024 GF002	Vrij
2	Nikon	25	16.6	Geen	1280x1024	25fps_1280-1026	Vrij
3	Nikon	50	16.6	Groen	1280x1024	50fps 1280-1024 GF001	Vrij
4	Nikon	50	16.6	Geen	1280x1024	50fps 1280-1025	Vrij
5	Nikon	max	16.6	Geen	1280x1024	maxfps 1280-1024	Vrij
6	Nikon	max	16.6	Groen	1280x1024	maxfps_1280-1024_GF	Vrij
7	Nikon	25	16.6	Groen	320x256	25fps 320-256 GF008	Vrij
8	Nikon	25	16.6	Geen	320x256	25fps_320-256	Vrij
9	Nikon	50	16.6	Geen	320x256	50fps 320-257	Vrij
10	Nikon	50	16.6	Groen	320x256	50fps_320-256_GF007	Vrij
11	Nikon	max	16.6	Groen	320x256	maxfps 320-256 GF006	Vrij
12	Nikon	max	16.6	Geen	320x256	maxfps 320-256	Vrij
13	Nikon	25	16.6	Groen	640x512	25fps_640-512_GF005	Vrij
14	Nikon	25	16.6	Geen	640x512	25fps_640-514	Vrij
15	Nikon	50	16.6	Groen	640x512	50fps_640-512_GF004	Vrij
16	Nikon	50	16.6	Geen	640x512	50fps_640-513	Vrij
17	Nikon	max	16.6	Geen	640x512	maxfps_640-512	Vrij
18	Nikon	max	16.6	Groen	640x512	maxfps_640-512_GF003	Vrij
19	Nikon	-	16.6	Groen	720x480	Field1.avi	Trapped
20	Nikon	-	16.6	Groen	720x480	Field2.avi	Trapped
21	Nikon	-	16.6	Groen	720x480	Field3.avi	Trapped
22	Nikon	-	16.6	Groen	720x480	Field4.avi	Trapped
23	Nikon	-	16.6	Groen	720x480	Field5.avi	Trapped
24	Nikon	-	16.6	Groen	720x480	Field6.avi	Trapped
25	Basler	25	25	Geen	780x572	25_fps_exp_time_1000_us	Vrij
26	Basler	25	25	Groen	780x572	25_fps_exp_time_1000_us_GF	Vrij
27	Basler	37	37	Geen	780x572	37_fps_exp_time_1000_us	Vrij
28	Basler	37	37	Groen	780x572	37_fps_exp_time_1000_us_GF	Vrij
29	Basler	50	50	Geen	780x572	50_fps_exp_time_1000_us	Vrij
30	Basler	50	50	Groen	780x572	50_fps_exp_time_1000_us_GF	Vrij

B. loadvideo

function [im,frames,fps]=loadvideo(path,actualframes); mov=VideoReader(path); %video inladen frames=mov.NumberOfFrames; %aantal frames uitlezen fps=mov.FrameRate; % framrate uitlezen im=zeros(mov.Height,mov.Width,ceil(frames/actualframes)); %pre-allocatie videomatrix for i=1:actualframes:frames im(:,:,ceil(i/actualframes))=rgb2gray(read(mov,i)); %frame uitlezen met read en omzetten met rgb2gray clc disp(['Loading video ' num2str(ceil(i/frames*100)) '% done']) end

end

C. testvideo

```
function [video2,actualframes,dropped,fps]=testvideo(video,fps);
frames=size(video,3);
j=1;
for i=1:frames
frame=video(:,:,i);
value(:,:,i)=sum(frame(:)); % som van alle waarden voor elk frame bepalen. Vrijwel
altijd uniek.
end
%video testen op dubbele frames
for i=2:frames-1
    check1=abs(value(:,:,i-1)-value(:,:,i)); %som van frame met vorig frame vergelijken
    check2=abs(value(:,:,i)-value(:,:,i+1)); %som van frame met volgende frame
vergelijken
if (check1>1)&&(check2>1) %frame toelaten als er een verschil zit in het linker en
rechterframe
video2(:,:,j)=video(:,:,i);
j=j+1;
end
clc
disp(['Testing video ' num2str(ceil(i/frames*100)) '% done'])
end
actualframes=size(video2,3); %nieuwe hoeveelheid frames bepalen
dropped=frames-2-actualframes; %aantal 'dropped' (verloren) frames bepalen
interval=floor(dropped/actualframes)+1; %keyframe interval bepalen
fps=fps/interval; %nieuwe fps uitrekenen
disp(['Dropped frames found:' num2str(dropped)])
disp(['Key frame interval:' num2str(interval) ])
end
```

D. isolateheads

```
function [bw heads]=isolateheads(im,min blobs,threshadjust)
sum=zeros(size(im,1),size(im,2)); %pre allocatie voor snelheid
frames=size(im,3); %frames bepalen
for i=1:frames
  sum=sum+im(:,:,i); %ieder frame bij elkaar optellen
end
background=sum/frames; %om vervolgens te delen door het aantal frames zodat de
gemiddelde achtergrond bekend is
for i=1:frames
ims=abs(im(:,:,i)-background); %statische achtergrond wordt afgetrokken van ieder frame
th=graythresh(ims)*255*threshadjust; %255 nodig omdat graythresh een getal geeft tussen
de 0-1 wij willen de grijswaarde weten
bw heads(:,:,i) = bwareaopen(ims>th/2, min blobs); %het frame met ingestelde threshold
wordt gefilterd op te kleine objecten(ruis of rommel)
%imshow(bw heads(:,:,i),[]);
clc
disp(['Isolating heads ' num2str(ceil(i/frames*100)) '% done'])
end
end
```

E. showheads

```
function []=showheads(im, bw heads, box, center)
frames=size(bw heads,3);
for i=1:frames
 stats=regionprops(bw_heads(:,:,i));
drawnow; %nodig voor realtime tekenen.
imshow(im(:,:,i),[]);
for k = 1 : size(stats)
    hold on;
  if (stats(k).Area>0) && (box==1) %boundingbox tekenen
  thisBB = stats(k).BoundingBox;
  rectangle('Position', [thisBB(1),thisBB(2),thisBB(3),thisBB(4)],...
  'EdgeColor', 'r', 'LineWidth',2 )
  end
  if (stats(k).Area>0) && (center==1)%middenpunt tekenen
  thisBB = stats(k).Centroid;
 plot(thisBB(1),thisBB(2),'b.')
  end
  hold off;
end
end
end
```

F. findmatch

```
function [new,idx2,idx] = findmatch(history,next,var1)
```

```
dist=@(x1,y1,x2,y2) sqrt((x2-x1).^2 + (y2-y1).^2); %functie voor afstand opstellen
(pythagoras)
sizehistory=size(history); %kijken hoeveel frames(rijen) er al zijn.
newsize=sizehistory(1)+1; %rij opschuiven
new=history; %history gaan we aanvullen met de nieuw gevonden punten en vormt zo new.
numbers=[1:size(next,1)]; %hoeveel nieuwe punten zijn er gevonden
for i=2:2:size(history,2) %afstand berekenen tot elk punt
    for j=1:size(next,1)
       afstanden(i/2,j)=dist(history(max(find(history(:,i-1)>0)),i-1),history(max(find(hi
story(:,i)>0)),i),next(j,1),next(j,2));
    end
end
for i=2:2:size(history,2) %afstanden met elkaar vergelijken
 if size(numbers,1)>0
    if min(afstanden(i/2, numbers))<var1</pre>
      [val,place1]=min(afstanden(i/2,numbers)); % kleinste getal in rij zoeken
      [val,place2]=min(afstanden(:,numbers(place1))); %kleinste getal in kolom zoeken
       new(newsize,place2*2-1:place2*2)=next(numbers(place1),:); %data wegschrijven
naar new.
       idx(1,i/2)=numbers(place1); % geeft het nummer van het nieuwe punt
       idx2(1,i/2)=place2; % geeft het nummer van het bestaande punt waarmee het nieuwe
punt is gekoppeld.
       numbers(place1) = [];
    end
end
end
for k=1:size(next) %nieuwe punten die niet zijn gekoppeld wegschrijven naar eigen
nieuwe kolom.
if min(afstanden(:,k))>var1
    new(newsize, sizehistory(2)+1:sizehistory(2)+2)=next(k,:);
    sizehistory(2) = sizehistory(2) + 2;
    idx(1, (sizehistory(2)/2)) = k;
    idx2(1,(sizehistory(2)/2))=(sizehistory(2)/2);
end
end
end
```

G. showpaths

```
function [] =showpaths(points,video,timeinframe,frame)
%frame=size(video,3);
frames=frame;
imshow(video,[]);
hold on
cmap = hsv(size(points,2)); %kleurengradient opstellen op basis van aantal punten
next=1;
for p=2:2:size(points,2)
    if Count (points (:, p), '>0') > time inframe * frames; % filteren op aantal positiepunten
plot(points(:,p-1),points(:,p),'.','Color',cmap(p,:)) %punten tekenen
drawcoord=max(find(points(:,p-1)>1));
text(points(drawcoord,p-1)+20, points(drawcoord,p)+20, sprintf('%i', next), 'color', cmap(p,
:), 'Fontsize', 20)%getal tekenen
    end
    next=next+1;
end
```

H. tailstart

```
function [tailcoord,tailangle]=tailstart(frame,headcenters)
points=headcenters;
im=frame;
cres=100;
for i=2:2:size(points,2)+1
    if (points(1,i-1)>0) | | points(1,i)>0
    xcenter=points(1,i-1);
    ycenter=points(1,i);
   %cellrad=sqrt(max(cat(1,stats1.Area)))/pi;
   cellrad=2;
   for j=1:cres
   angle=(2*pi/cres)*j;
   x=round(xcenter+cellrad*5*sin(angle));
   y=round(ycenter+5*cellrad*cos(angle));
   if (x>0) && (y>0) && (x<size(im, 2)) && (y<size(im, 1))</pre>
   graph(j, 1) = im(y, x);
   graphcoord(j,1)=x;
   graphcoord(j,2)=y;
   end
   end
   %coordinaten gevonden staart wegschrijven naar maxstaart
   [val pos]=max(graph(:,1));
   %tailangle(:,i/2)=((360/(pi))*(pi/cres)*pos);
   tailcoord(1,i-1:i) = graphcoord(pos,1:2);
    end
end
end
```

I. findangle

```
function [ angle ] = findAngle( center_x, center_y, point_x, point_y)
%bepaald hoek ten opzichte van (1,0)

dx = point_x - center_x;
dy = point_y - center_y;

v_point = [dx dy];
v_ref = [1 0];

cos_a = dot(v_point, v_ref)./(norm(v_point) * norm(v_ref));
angle = acos(cos_a);

if(dy < 0)
    angle = 2*pi-angle;</pre>
```

end

J. findtail

```
function [maxmaxcoord,video2] = findtail(video,head,cres,r,th)
frames=size(video,3);
%% mediaan afbeelding bepalen
medianimage=zeros(size(video(:,:,1)));
video2=zeros(size(video(:,:,1)));
for l=1:size(video,1)
    medianimage(l,:)=median(rot90(squeeze(video(l,:,:))));
end
```

```
close all
%% mask maken voor de staarten
for i=1:frames
videofilt1=edge(abs(video(:,:,i)-medianimage),'sobel',1);
videofilt2=bwareaopen(videofilt1,10);
videofilt3=imdilate(videofilt2, strel('disk', th, 8));
videofilt4=imfill(videofilt3,4,'holes');
% staart uitknippen met de gevonden masks uit de orginele video
video2(:,:,i)=videofilt4.*video(:,:,i);
%imshow(video(:,:,i),[])
clc
disp(['Extracting tail ' num2str(ceil(i/frames*100)) '% done'])
end
%% cirkel maken
for it=1:1:r
crad=it;
m=1;
n=1;
graph=zeros(cres, frames);
for frame=1:frames
   %ims=abs(im(:,:,frame)-medianimage);
```

xcenter=head(frame,1);

```
ycenter=head(frame, 2);
   for j=1:cres
   angle=(2*pi/cres)*j;
   x=round(xcenter+crad*sin(angle));
   y=round(ycenter+crad*cos(angle));
   if (x>0) && (y>0) && (x<size(video,2)) && (y<size(video,1)) % anders probeerd hij buiten
het frame te zoeken
   graph(j,frame)=video2(y,x,frame);
   graphcoord(j,m)=x;
   graphcoord(j,m+1)=y;
   end
   end
   m=m+2;
   %imshow(graph,[])
   end
%% filteren en maximale waarde bepalen uit 1 frame
meancol=mean(graph');
graph2=graph;
for i=1:frames
    graph2(:,i)=graph(:,i)-meancol';
end
graph2=abs(graph2);
graph2=graph2(1:size(graph,1),1:size(graph,2));
%imshow(graph2,[])
%% maxima per lijn eruit halen
n=1;
maxding = zeros(size(graph2));
for i=1:frames
  [C,I]=max(graph2(:,i));
   if C>1
   maxding(I,i)=1;
   maxcoord(i,1) = graphcoord(I,n);
   maxcoord(i,2)=graphcoord(I,n+1);
   else
    maxcoord(i,1)=nan;
  maxcoord(i,2)=nan;
   end
   n=n+2;
end
maxmaxcoord(:,:,it)=maxcoord;
end
end
```

K. showtail

```
function [anglerange,
tdistance]=showtail(video,head,tailbase,tail,cellnumber,framesize)
nummer=cellnumber;
range=size(tail,3);
path=head(:,2*nummer-1:2*nummer);
cmap = hsv(range+1);
kleur=1;
for frame=1:200
for i=10:1:range
  [staart]=[tail(frame,:,i)];
  [kop] = [head(frame, nummer*2-1:nummer*2)];
  [staartpunt]=tailbase(frame,nummer*2-1:nummer*2);
  tangle(1, i, frame) =- ((findAngle(kop(1), kop(2), staart(1), staart(2))) - (findAngle(kop(1), ko
p(2),staartpunt(1),staartpunt(2)));
  tdistance(1, i, frame) = tangle(1, i, frame);
end
%tdistance(1,:,frame)=-unwrap(tdistancewrap(1,:,frame));
p=polyfit(1:range,tdistance(:,:,frame),4)
subplot(4,1,1);
plot(1:range,polyval(p,1:range))
hold on
plot(1:range,tdistance(1,:,frame),'*')
axis([ 10 range -pi pi])
hold off
h=subplot(4,1,2);
point1=round(path(frame,2)-60);
point2=round(path(frame,2)+60);
point3=round(path(frame, 1)-60);
point4=round(path(frame, 1)+60);
drawnow;
imshow(video(point1:point2,point3:point4,frame),[])
hold on
plot(squeeze(tail(frame,1,:))-path(frame,1)+60, squeeze(tail(frame,2,:))-
path(frame, 2) + 60, 'r.')
tailangle(frame)=findAngle(0,0,staartpunt(1)-kop(1),staartpunt(2)-kop(2));
v=[cos(tailangle(frame)), sin(tailangle(frame))];
basis=[60 60];
v=round(50.*v);
line([60 60+v(1)],[60 60+v(2)],'Color','g','LineWidth',1)
plot(staartpunt(1)-kop(1)+60, staartpunt(2)-kop(2)+60, 'y*');
plot(60,60,'c*');
hold off;
ax=get(h, 'Position');
ax(4)=ax(4)*1;
set(h, 'Position', ax);
subplot(4,1,3)
drawnow;
hold on
%plot(frame,tailangle(frame));
plot(tailangle)
axis([1 200 0 2*pi])
```

```
hold off;
subplot(4,1,4);
drawnow;
hold on
plot(1:range,tdistance(1,:,frame),'*')
anglerange(frame)=tdistance(1,30,frame);
hold off
axis([ 10 range -pi pi])
frame
waitforbuttonpress;
end
end
```

```
Analyse voor enkele video
L.
8{
Casa Using Matlab
by Daan Geijs
Version 1.0
daangeijs@gmail.com
november 2014
8}
%% load video
addpath('Functions')
clear all; close all;
[video,frames,fps]=videoload('Video/Basler/25 fps exp time 1000 us.avi',1);
%[video,actualframes,dropped,fps]=testvideo(video(:,:,60),fps);
%% setup variables
d points=20; % dit is voor de findmatch functie. Dit is de maximale afstand waarin een
volgend punt mag liggen.
t=0.35;
p=15;
88
[heads]=isolateheads(video,p,t); %getal geeft minimale detecteerbare blobs en de
threshold is aan te passen
showheads(heads,heads,1,1); % laat de gevonden koppen zien.
88
startdata=regionprops(heads(:,:,1)); % de locaties van de gevonden objecten in het
eerste frame
history=cat(2,startdata.Centroid);
for frame=1:frames-1
newdata=regionprops(heads(:,:,frame+1));
newpoints=cat(1, newdata.Centroid);
if size(newpoints)>0
[history,id]=findmatch(history,newpoints,d points);% vult history aan met de nieuwe
gevonden coordinaten, deze coordinaten vormen weer de basis voor de volgende iteratie.
[maxstaart]=tailstart(video(:,:,frame),history(frame,:));
maxmaxstaart(frame,1:numel(maxstaart))=maxstaart;
%showheads(video(:,:,frame),heads(:,:,frame),1,1) %laat de locatie van de koppen in de
video zien
hold on
for r=2:2:size(maxstaart,2)
 % plot(maxstaart(1,r-1),maxstaart(r),'g*') %laat de gevonden staart zien in de video
end
hold off;
end
end
showpaths (history, video (:,:, frame), 0.9, frames) %laat het uiteindelijk pad zien
%% VSL,VCL en LIN bepalen
motil param=motility(history,400,fps);
%% rotatie bepalen voor specifieke cel, nummer komt overeen met het nummer uit de figuur
van showpaths
celnum=3;
frames=210;
```

```
for i=1:size(maxmaxstaart,1)
```

```
hoeken(i)=findAngle(history(i,celnum*2-1),history(i,celnum*2),maxmaxstaart(i,celn
um*2-1), maxmaxstaart(i, celnum*2));
end
uwrap=unwrap(hoeken);
[p]=polyfit(1:frames-1,uwrap(1:frames-1),1);
speed=mean(diff(polyval(p,1:frames-1)));
rotspeed rad=speed*fps
rotspeed degree=speed*fps*(360/(2*pi));
%faketraps(video,history,celnum,60); %knipt een roterende cel uit en maakt er een
getrapte cel van
figure;
hold on; plot(polyval(p,1:frames-1));
plot(uwrap);hold off;
%% 3d plot
x=maxmaxstaart(:,celnum*2-1);
y=maxmaxstaart(:,celnum*2);
hold on
plot3(x, y, 1: frames - 1, ' - ')
%% staart detectie
```

```
close all
[tailcoords]=findtail(video, history(:, celnum*2-1:celnum*2), 200, 40, 4);
%% staart laten zien
startframe=130;
[~,hoekstelsel,]=showtail(video(:,:,startframe:frames), history(startframe:frames,:),max
maxstaart(startframe:frames-1,:),tailcoords(startframe:frames,:,:),celnum,60);
```

M. Fake traps

```
function [mov]=faketraps(video,pointsofheads,numbers,framesize)
frames=size(video,3);
%framesize=50;
%numbers=[2,3,6,7];
mov=zeros(framesize*2+1, size(numbers, 2)*framesize*2+1);
start=1;
for frame=1:frames-1
for i=1:size(numbers,2)
    drawnow:
   path=pointsofheads(frame, 2*numbers(i)-1:2*numbers(i));
   point1=round(path(1,2)-framesize);
   point2=round(path(1,2)+framesize);
   point3=round(path(1,1)-framesize);
   point4=round(path(1,1)+framesize);
   if (point1>0) & (point2>0) & (point3>0) & (point4>0)
   cut=video(point1:point2,point3:point4,frame);
   %total(1:size(total,1),size(total,2)+1:size(cut,2)*i)=cut;
   mov(1:framesize*2+1,start:start+framesize*2,frame)=cut;
   end
   start=start+framesize*2;
   imshow(mov(:,:,frame),[])
end
start=1;
end
end
```

N. Analyse en testen voor meerdere video's

```
%%testbench
addpath('Functions')
addpath('Video')
clear all; close all;
counter=1;
videofolder='Video/Nikon/640/';
files = dir(sprintf('%s*.avi',videofolder));
name = {files.name}.';
titels={'Filename', 'Total detected objects', 'Cells of interest', 'Tracking
succes', 'Frames', 'Dropped', 'Computation Time', 'CT/frame', 'CT/s'};
for tres=0.22
areadb=nan;
ldb=nan;
bdb=nan;
teller=1;
for i=1:size(files,1)
tic;
[video, frames, fps]=videoload(sprintf('%s%s', videofolder, files(i).name), 6);
fps=fps/6; %nodig voor bij de nikon
[heads]=isolateheads(video, 15, tres);
d points=20;
startdata=regionprops(heads(:,:,1), 'Area', 'Centroid', 'MajorAxisLength',
'MinorAxisLength');
history=cat(2,startdata.Centroid);
areahistory=cat(1,startdata.Area).';
bhistory=cat(1,startdata.MinorAxisLength).';
lhistory=cat(1,startdata.MajorAxisLength).';
for frame=1:frames-1
newdata=regionprops(heads(:,:,frame+1),'Area','Centroid',
'MajorAxisLength', 'MinorAxisLength');
newpoints=cat(1, newdata.Centroid);
newarea=cat(1,newdata.Area);
newl=cat(1, newdata.MajorAxisLength);
newb=cat(1, newdata.MinorAxisLength);
if size(newpoints)>0
[history, id his, id new]=findmatch (history, newpoints, d points); % vult history aan met de
nieuwe gevonden coordinaten, deze coordinaten vormen weer de basis voor de volgende
iteratie.
id his=id his(find(id his>0));
id new=id new(find(id new>0));
for a=1:size(id new,2)
    areahistory(frame, id his(a)) = newarea(id new(a));
    lhistory(frame, id his(a)) = newl(id new(a));
    bhistory(frame,id his(a))=newb(id new(a));
end
%showheads(video(:,:,frame),heads(:,:,frame),1,1)
clc
disp(['Find path and angle ' num2str(ceil(frame/frames*100)) '% done'])
end
```

```
%showpaths(history, video(:,:,frame),0.01,1)
end
meanarea=mean(areahistory);
meanl=mean(lhistory);
meanb=mean(bhistory);
if isempty(history) ==1
        avg=0;
else
    for r=1:2:size(history,2)
        obj=find(history(:,r)>0);
        avg(ceil(r/2),1)=size(obj,1)/frames;
    end
end
r=motility(history,frames,fps);
[idx, val]=find(avg(:)>0.90);
for k=1:size(idx)
motil param(:,teller)=r(2:4,idx(k));
teller=teller+1;
end
results(i,1)=size(avg,1);
results(i,2)=size(val,1);
results(i,3)=mean(avg(idx));
results(i,4)=frames;
%results(i,5)=dropped;
results(i,6)=toc;
results(i,7)=results(i,6)/frames;
results(i,8)=results(i,7)*fps;
areadb=[areadb;meanarea(idx)'];
ldb=[ldb;meanl(idx)'];
bdb=[bdb;meanb(idx)'];
close all
showpaths(history, video(:,:,frame),0.9,frames)
mean(areahistory);
mean(bhistory);
mean(lhistory);
clearvars -except files name titels results eindresultaat counter tres areahistory
areadb idx areadbXL motil param teller bdb ldb videofolder fps
% data verzamelen
end
disp(sprintf('%i Cellen gedetecteerd met een: ',size(areadb,1)))
disp(sprintf('Gemiddelde lengte: %f STD %f',nanmean(ldb),nanstd(ldb)))
disp(sprintf('Gemiddelde breedte: %f STD %f',nanmean(bdb),nanstd(bdb)))
disp(sprintf('Gemiddelde oppervlakte: %f STD %f',nanmean(areadb),nanstd(areadb)))
disp(sprintf('Gemiddelde VSL: %f STD %f',nanmean(motil param(1,:)),nanstd(motil
param(1,:))))
disp(sprintf('Gemiddelde VCL: %f STD %f', nanmean(motil param(2,:)), nanstd(motil
param(2,:))))
disp(sprintf('Gemiddelde LIN: %f STD %f',nanmean(motil param(3,:)),nanstd(motil
param(3,:)))
```

```
eindresultaat1=[name,num2cell(results)];
eindresultaat(counter)={[titels;eindresultaat1]};
```

```
areadbXL(1:numel(areadb),counter)=areadb(idx);
counter=counter+1;
end
```

% %

```
for l=1:size(eindresultaat,2)
test=eindresultaat{1,1};
dataarray(:,1)={test{2:size(files,1)+1,3}}.';
dataarray2(:,1)={test{2:size(files,1)+1,4}}.';
end
```

```
data_array=cell2mat(dataarray);
tracking_array=cell2mat(dataarray2);
xaxis=(size(data_array,2)/2)/10;
plot(0.05:0.05:xaxis,data_array')
axis([0 1,0 50])
%%
for l=1:size(data_array,1)
temp=data_array(l,:);
idx=temp>=bound(l,1) & temp<=bound(l,2);
data_array2(l,1:numel(idx))=idx
totaal=sum(bound(:,1))
end
for k=1:size(bound,1)
finaldata(k,:)=data_array2(k,:)*bound(k,1)
```

```
end
```

```
plot(0.05:0.05:xaxis,(sum(finaldata)/23)*100)
```

hold on
plot(0.05:0.05:xaxis,tracking_array*100)
axis([0 1,0 100])

88

median_head=nanmean(areadbXL)
plot(0:0.05:0.85,median_head)