ABSTRACT

IMPROVING THE PERFORMANCE AND EVALUATION OF COMPUTER-ASSISTED SEMEN ANALYSIS

by
Ji Won Choi

Semen analysis is performed routinely in fertility clinics to analyze the quality of semen and sperm cells of male patients. The analysis is typically performed by trained technicians or by Computer-Assisted Semen Analysis (CASA) systems. Manual semen analysis performed by technicians is subjective, time-consuming, and laborious, and yet most fertility clinics perform semen analysis in this manner. CASA systems, which are designed to perform the same tasks automatically, have a considerable market share, yet many studies still express concerns about their accuracy and consistency. In this dissertation, the focus is on detection, tracking, and classification of sperm cells in semen images, key elements of CASA systems. The objective is to improve existing CASA algorithms and systems by applying validated computer vision, tracking, and computational intelligence algorithms.

The first step of the study is the development of simulation models for generating synthetic images of semen samples. The images enable the assessment of CASA systems and their algorithms. Specifically, the simulation models generate time-lapse images of semen samples for various sperm image categories and include ground truth labels. The models exploit standard image processing operations such as point spread functions and 2D convolutions, as well as new models of sperm cell swimming, developed for this study. They embody multiple studies of sperm motility in the form of parameterized motion equations. Use cases are presented to use the swimming models and the simulated images to assess and compare algorithms for sperm cell segmentation, localization, and tracking.

Second, a digital washing algorithm is presented for unwashed semen samples. Digital washing has the potential to replace the chemical washing techniques used by fertility clinics at present, which are costly, time-consuming, and unfriendly to the environment. The digital washing algorithm extracts features from moving sperm cells in an image, and uses these features to identify all sperm cells (moving and stationary) within each studied image (simulated or real). The effectiveness of the digital washing algorithm is demonstrated by comparing the performance of the proposed algorithm to other cell segmentation and detection techniques.

Third, a classification algorithm for sperm cells is developed, based on their swimming patterns. The classification algorithm uses K-means clustering on a subset of motility parameters of sperm cells selected by the Artificial Bee Colony (ABC) algorithm. Results of classification and clustering are shown, using simulated and real semen images. Swimming pattern classification has the potential to increase understanding of the relationship between the distribution of sperm cell swimming modes in a patient's semen image and the fertility of that patient.

Lastly, a new method is presented to calculate motility parameters from sperm tracks. The movement of sperm cell is modeled as a sinusoidal traveling wave ("traveling sinusoid"). The amplitude and average path of a moving cell are estimated using an extended Kalman filter (EKF). The states estimated by the EKF include position, velocity, amplitude, and frequency of the traveling wave. The motility parameters calculated from this approach are shown to be superior to those calculated by other existing methods in terms of their accuracy and consistency.

CASA developers will find in this study (and in the software made available) new tools to improve the performance of their designs, and to compare and contrast different proposed approaches and algorithms.

IMPROVING THE PERFORMANCE AND EVALUATION OF COMPUTER-ASSISTED SEMEN ANALYSIS

by
Ji Won Choi

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

May 2022

APPROVAL PAGE

IMPROVING THE PERFORMANCE AND EVALUATION OF COMPUTER-ASSISTED SEMEN ANALYSIS

Ji Won Choi

Dr. Moshe Kam, Dissertation Advisor Date
Dean of the Newark College of Engineering, Professor of Electrical and Computer Engineering, NJIT

Dr. Alexander Haimovich, Committee Member Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Ali Abdi, Committee Member Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Seyyedmohsen Azizi, Committee Member Date
Assistant Professor of Engineering Technology, NJIT

Dr. Leonid Hrebien, Committee Member Date
Professor of Electrical and Computer Engineering, Drexel University

BIOGRAPHICAL SKETCH

Author:	Ji Won Choi
Degree:	Doctor of Philosophy
Date:	May 2022

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2022
- Bachelor of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2015

Major: Electrical Engineering

Presentations and Publications:

J. Choi, L. Alkhoury, L. Urbano, P. Masson, M. VerMilyea, and M. Kam, "An Assessment tool for Computer-Assisted Semen Analysis (CASA) Algorithms," Under review, Research Square PREPRINT (Version 1), https://doi.org/10.21203/rs.3.rs-1350853/v1.

C. Iaboni, D. Lobo, J. Choi and P. Abichandani, "Event-Based Motion Capture System for Online Multi-Quadrotor Localization and Tracking," *Sensors*, vol. 22, no. 9, 2022, https://doi.org/10.3390/s22093240

C. Iaboni, H. Patel, D. Lobo, J. Choi and P. Abichandani, "Event Camera Based Real-Time Detection and Tracking of Indoor Ground Robots," *IEEE Access*, vol. 9, pp. 166588-166602, 2021, doi: 10.1109/ACCESS.2021.3133533.

L. Alkhoury, J. Choi, C. Wang, A. Rajasekar, S. Acharya, S. Mahoney, B. S. Shender, L. Hrebien and M. Kam, "Heart-rate Tuned Comb Filters for Processing Photoplethysmogram (PPG) Signals in Pulse Oximetry," *Journal of Clinical Monitoring and Computing*, Vol. 35, pp. 797-813, 2021, https://doi.org/10.1007/s10877-020-00539-2

J. Choi, C. Wang, L. F. Urbano, P. Masson, M. VerMilyea and M. Kam, "Classification and Clustering of Human Sperm Swimming Patterns," *IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, Shenzhen, China, 2018, pp. 192-196. doi: 10.1109/SIPROCESS.2018.8600422.

C. Wang, J. Choi, L. F. Urbano, P. Masson, M. VerMilyea and M. Kam, “Tracking of Human Sperm in Time-Lapse Images,” *IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, Shenzhen, China, 2018, pp. 197-201. doi: 10.1109/SIPROCESS.2018.8600463.

J. Choi, L. Urbano, P. Masson, M. VerMilyea and M. Kam “Clustering of Human Sperm Swimming Patterns in Time-Lapse Images,” *IEEE International Conference on Healthcare Informatics (ICHI)*, New York, NY, 2018, pp. 371-373. doi: 10.1109/ICHI.2018.00060.

But whatever I am now, it is all because God poured out his special favor on me—and not without results. For I have worked harder than any of the other apostles; yet it was not I but God who was working through me by his grace.

1 Corinthians 15:10

ACKNOWLEDGMENT

First and foremost, I would like to thank my dissertation advisor, Dr. Moshe Kam. Over the years that I have been at the New Jersey Institute of Technology, it was due to his guidance and support that I have been able to achieve this goal. I also thank Dr. Urbano, who has provided the necessary means for me to conduct the studies in this dissertation. Without the assistance from Dr. Kam and Dr. Urbano, I would not have been successful in my Ph.D. studies.

Additionally, my thanks go out to Dr. Pramod Abichandani, who has advised me in research studies I was involved in at NJIT. I also thank the other members of my dissertation committee, Dr. Alexander Haimovich, Dr. Ali Abdi, Dr. Seyyedmohsen Azizi, and Dr. Leonid Hrebien, for their time, advice, and insightful comments.

I would like to show my gratitude to the Department of Electrical and Computer Engineering for their assistance and TA support. I also thank the Office of Naval Research (Grant No. N68335-16-C-0085) and the US Department of Army (Grant No. W15QKN-17-D-0029) for the research funding.

My sincere thanks go to Dr. Durgamadhab Misra, Dr. Cong Wang, Sheryl Baker, Angela Vega-Irvin, Carolina Yenez, and Kimberly Dripchak for their guidance and assistance in my years at NJIT.

I am also thankful to have had the opportunity to have shared my Ph.D. experience with my intelligent and bright colleagues, Dr. Weiqiang Dong, Chizhong Wang, Ludvik Alkhoury, and Craig Iaboni. My gratitude goes out to them for all their support and encouragement.

Lastly, I thank my parents, Suck eun and Myung sook, and my brother, Jung won. Our family's journey as immigrants was not easy. As a family, we have been together through many rocky roads. It is for their support and sacrifices that I am who I am today.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Table Page

LIST OF TABLES
(Continued)

Table Page

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure Page

LIST OF FIGURES
(Continued)

Figure Page

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure Page

LIST OF FIGURES
(Continued)

Figure Page

LIST OF SYMBOLS

(x_{H_C}, y_{H_C})	Head position of circular swimming cell
r_c	Radius of the circular path
a	Amplitude of the sinusoid modulated on the circular path
f_s	Frequency of the sinusoid modulated on the circular path (Hz)
f_c	Frequency of the circular cycle (cycle/sec)
(C_{x_c}, C_{y_c})	Vertical and horizontal offset constant
(x_{Tail_C}, y_{Tail_C})	A set of k points along the center of the flagellum of circular swimming cell
λ	Wavelength of flagellum (distance between the start and the end of the flagellum)
$R(\cdot)$	Rotation matrix
$b(\cdot)$	Local variation in beating amplitude along the flagellum
(x_{H_L}, y_{H_L})	Position of the sperm head of linear swimming cell
(C_{x_L}, C_{y_L})	Horizontal and vertical offset constant (pixels)
V	Straight line path velocity (pixels/sec)
f_l	Rate of change in ribbon angle (Hz)
r_h, r_v	Width and height of ribbon (pixels)
A_{har}	User defined ratio between the first and the third harmonics
A_c	Correction constant for defined width of the ribbon
θ_r	Direction of the forward movement (radian)
$(x_{Tail_{LM}}, y_{Tail_{LM}})$	A set of k points along the center of the flagellum of linear mean swimming cell

$b_1(k), b_2(k)$	Local horizontal and vertical variation in beating amplitude along the flagellum
$b_3(k)$	Flagellum position correction function
(μ_x, μ_y)	Drift coefficients of Brownian motion
$(\sigma_{x_b}, \sigma_{y_b})$	Diffusion coefficients of Brownian motion
$W(t)$	standard 1-dimensional Brownian motion ($W(t) \sim N(0, t)$)
(x_{H_H}, y_{H_H})	Position of the sperm head of hyperactive cell
T	the difference in time between each simulation frame
(x_{Tail_H}, y_{Tail_H})	A set of k points along the center of the flagellum of hyperactive cell
(x_{H_I}, y_{H_I})	Position of the sperm head of immotile cell
(x_{Tail_D}, y_{Tail_D})	A set of k points along the center of the flagellum of immotile cell
ω	Weight
μ	Mean
Σ	Covariance
σ^2	variance
T	Period
$F(k)$	State transition matrix at time k
$G(k)$	control input matrix at time k
$H(k)$	Observation matrix at time k
$Q(k)$	Process noise covariance matrix at time k
$R(k)$	Observation noise covariance matrix at time k
$P(k)$	State prediction covariance matrix at time k
$W(k)$	Kalman filter gain at time k
$\upsilon(k)$	Innovation at time k
$S(k)$	Innovation covariance matrix at time k

LIST OF ABBREVIATIONS

CASA	Computer-Assisted Semen Analysis
WHO	World Health Organization
OSPA	Optimal Subpattern Assignment
MOTP	Multiple Object Tracking Precision
MOTA	Multiple Object Tracking Accuracy
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
VCL	Curvilinear velocity
VSL	Straight-line velocity
VAP	Average path velocity
ALH	Amplitude of lateral head displacement
LIN	Linearity
WOB	Wobble
STR	Straightness
MAD	Mean angular displacement
FPS	Frames Per Second
LoG	Laplacian of Gaussian
GMM	Mixture of Gaussian Model
SGM	Single Gaussian Model

NN	Nearest Neighbor
GNN	Global Nearest Neighbor
PDAF	Probabilistic Data Association Filter
JPDAF	Joint Probabilistic Data Association Filter
KF	Kalman Filter
EKF	Extended Kalman Filter
DWNA	Discrete White Noise Acceleration

CHAPTER 1

INTRODUCTION

Semen analysis is an essential assessment tool for evaluating fertility. It is usually performed manually by technicians, or by using an computer-assisted semen analysis (CASA) system. Some of the major parameters collected during semen analysis include sperm concentration (the number of sperm per milliliter of semen), percentage of motile sperm, and the morphology of sperm cells. In the case of CASA systems, motility parameters are also calculated from the sperm cells, which thereby quantifying the characteristics of how the sperm cells move in the semen sample. The motility parameters include curvilinear velocity (VCL), straight linear velocity (VSL), average path velocity (VAP), linearity (LIN), wobble (WOB), straightness (STR), amplitude of lateral head displacement (ALH), and mean angular displacement (MAD) [1].

In manual analysis, the technicians count or label cells in each frame of the image. This process is labor-intensive, time-consuming, and subjective (analysis may differ from technician to technician). On the other hand, computer-assisted semen analysis (CASA) systems are known to be fast, consistent and repeatable [2]. In CASA systems, semen sample video frames are analyzed to evaluate each specimen using image processing and particle tracking. Several studies have shown that CASA systems have comparable results to those of manual analysis by technicians with proper calibration and system parameter selection [3, 4]. However, other studies expressed caution, and suggested to wait for further development of CASA algorithms before automatic algorithms replace manual measurements [5,6].

In this study, several methods and algorithms that tackle existing challenges with CASA systems are presented. The problems include the following:

- Lack of tools to assess and compare different CASA system algorithms.

- Semen samples need to be washed chemically to remove debris and extraneous material, a process which is costly and environmentally unfriendly.
- Lack of methods to classify the sperm cells in the semen sample meaningfully, possibly for clinical use.
- Inconsistency of motility parameter calculation between CASA instruments of different manufacturers.

In the following Chapters 2-5, the study attempts to address these challenges.

Chapter 2 describes a method of simulating a semen sample for assessment of Computer-Assisted Semen Analysis (CASA) systems and algorithms. The simulation is able to generate an image with known qualities and provide the ground truth location of each cell in the image. We believe this is the first simulation tool of its kind available to CASA developers. It required the development of new models for the image and swim modes of sperm cells.

Chapter 3 describes a method of washing an unwashed semen sample digitally. In a typical semen analysis, the semen goes through a washing process, which removes debris and non-sperm cells present in the semen sample. The washing algorithm attempts to ”wash” away the debris and non-sperm cells through digital filtering means, eliminating the need to wash semen samples physically. Moreover, unlike chemical washing, the digital washing process does not wash away many sperm cells (ideally it does not wash away any).

Chapter 4 describes a method of classifying sperm cells in semen samples using motility parameters calculated from the track history. Sperm cells exhibit certain distinct modes of swimming, which include linear mean, circular, hyperactive, or immotile. The study classifies the cells by their swim types. The method involves the use of K-means clustering and the Artificial Bee Colony (ABC) algorithm. The ABC algorithm searches through different combinations of a subset of motility parameters to find the most distinct clusters for the different classes (swim type) of sperm cells.

Chapter 5 describes a method for calculating motility parameters using an extended Kalman filter (EKF). The EKF assumed that the track of sperm cell is a “traveling sinusoid." By using EKF, one is able to obtain estimates of position, velocity, frequency, and amplitude of the traveling wave. From this information, motility parameters for sperm cells can be calculated with high accuracy and consistency. The motility parameters calculated using the "traveling sinusoid" EKF tracking method are compared to the motility parameter calculated from the existing methods and shown to have the highest accuracy and consistency.

Table 1.1 summarizes the challenge/subject and outcome covered in each chapter of the dissertation.

Table 1.1 Summary of each Chapter in the Dissertation

Chapter	Challenge/Subject	Outcome
2	Lack of true ground truth for semen samples for objective testing of CASA systems and algorithms	A new semen sample simulation models are provided for consistent testing and comparison of CASA systems and algorithms
3	Unwashed semen samples require expensive and environmentally unfriendly chemical washing	A new digital washing algorithm is presented for detection of sperm cellsand removing debris and extraneous objects
4	Classification of sperm cells in semen analysis	K-means clustering based classification is performed, driven by iterative search algorithm, Artificial Bee Colony (ABC)
5	Inconsistent motility parameter calculations in existing systems	EKF based tracking for accurate and consistent calculation of motility parameters is presented through the estimation of position, velocity, frequency, and amplitude of swimming of sperm cells
6	End notes	-

Original contributions of this dissertation:

1. A new method of generating sperm cell images.
2. New mathematical models for the progressive swimming modes of sperm cells (linear mean and circular).
3. A new method of digital washing for unwashed semen samples.
4. A new method of classifying sperm cells by their swimming modes.
5. A new model for tracking sperm cells ("traveling sinusoid") and a new method of calculating motility parameters.

CHAPTER 2

AN ASSESSMENT TOOL FOR COMPUTER-ASSISTED SEMEN ANALYSIS SYSTEMS AND ALGORITHMS

In this chapter, a method of simulating a time-lapse image of a semen sample is introduced, along with several use-cases for the simulation, demonstrating how it can help in assessing and comparing Computer-Assisted Semen Analysis (CASA) algorithms and systems.

2.1 Background

The focus of this chapter is on the simulation of semen images for testing and validating Computer-Assisted Semen Analysis (CASA) systems and their algorithms. These systems continue to be of great interest to clinicians and andrology researchers [7]. Modern CASA systems "have been designed to objectively and quantitatively measure several aspects of sperm structure and function, aiming to provide high levels of intra- and inter-laboratory consistency" [7, 8]. To achieve this aim, methods of noise filtering, image segmentation, localization, multi-object tracking, and machine learning were employed [9–21].

A major challenge in developing and validating CASA systems is the accurate assessment and comparison of the resulting semen analysis methods to the ground truth. This process needs to take place across a representative sample of the expected images. For real-life samples, the ground truth is often unknown, motivating the use of high-quality image simulations with modifiable parameters for the validation of CASA systems and CASA algorithms. Simulations of this kind have the potential to help in developing automated semen analysis systems, and comparing different candidate algorithms to each other.

To generate a simulation of a semen sample video for assessment and validation, one needs the following: (1) a model for the image of a sperm cell (to generate sperm cells in the simulated semen image) and (2) a model for the sperm cell movement that defines how the semen image changes over time. The development of sperm cell models in this chapter mirrors similar work that was done for other cell types [22–27]. The development of sperm cell swim models is informed by several existing approaches. Some of these use a set of nonlinear equations of motion ([28] and Chapter 6 in [10]). Others use fluid dynamics to develop a more refined description of cell movements [29–37]. However, most existing simulation models of cell dynamics [31–37] did not integrate multiple cells into a comprehensive image such as the ones used in practice by CASA systems [38, 39]. Integration of these single cell descriptions into full-scale multi-cell images is required before relevant image simulations can be developed and used.

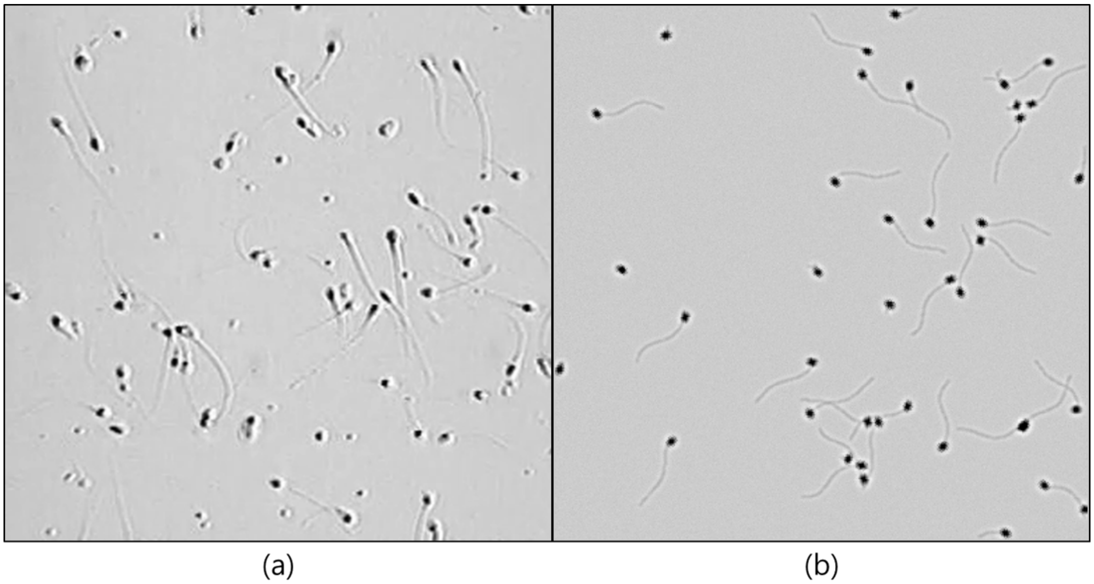


Figure 2.1 (a) Image of real human semen sample. (b) Image of simulated semen sample.

In this chapter, the models for the image of a sperm cell and for four (4) different swimming modes of sperm cells (circular, linear mean, hyperactive, and immotile) are provided. The computational requirements to simulate these models and integrate them into multiple-cell image are relatively low, allowing for the generation of multiple images in a format similar to the images used by the CASA system. As an example, a snapshot of a real semen image and a simulated image are shown side by side in Figure 2.1. Figure 2.1a is an image of semen sample at $200\times$ magnification. Figure 2.1b is a simulated semen image generated by using parameters estimated from the semen image on the left.

The chapter is organized as follows. In Section 2.2, models of a sperm cell and of the swimming modes are presented. In Section 2.3, the experimental setup and the performance metric used for assessment is explained. In Section 2.4, a use case for testing of segmentation and localization is demonstrated. In Section 2.5, use cases for testing of tracking algorithm is demonstrated. This chapter is accompanied by simulation software which is publicly available at [40] (as a stand-alone software and as MATLAB codes).

2.2 Methods

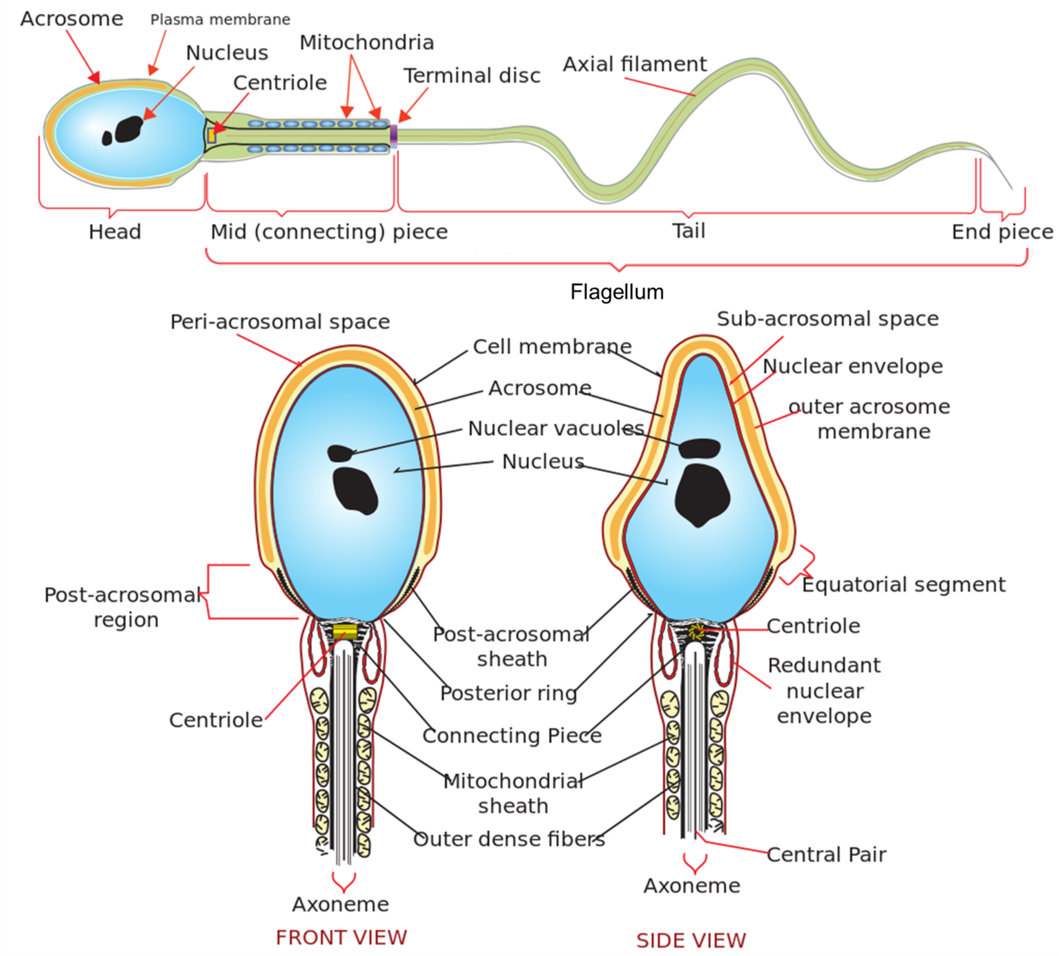


Figure 2.2 Diagram of a human sperm cell. (Top) Main components of a sperm cell. (Bottom) Front and side view of sperm head and midpiece.

The human sperm cell (see Figure 2.2, public domain: [41]) has two main parts, the head and the flagellum. The flagellum is composed of midpiece, tail (also called principal piece), and end piece. The flagellum moves in a wave-like motion to propel the cell [42]. The following sections describe the two-dimensional (2-D) image of the head and the flagellum of a sperm cell, and the models of sperm movements.

2.2.1 Simulating a Sperm Cell

The head and the flagellum of a sperm cell are generated separately and combined afterward. The models of the head and the flagellum of sperm cells follow the morphology of sperm cells discussed in the WHO laboratory manual for the examination and processing of human semen [1]. The normal shape of the human sperm head is generally oval. The tail, or the principal piece, of the flagellum is a thin cylinder of uniform calibre. In the simulation, the flagellum is modeled as the tail only, the simulation does not consider the midpiece or the endpiece for the simulation model since they are often too small to be observed separately.

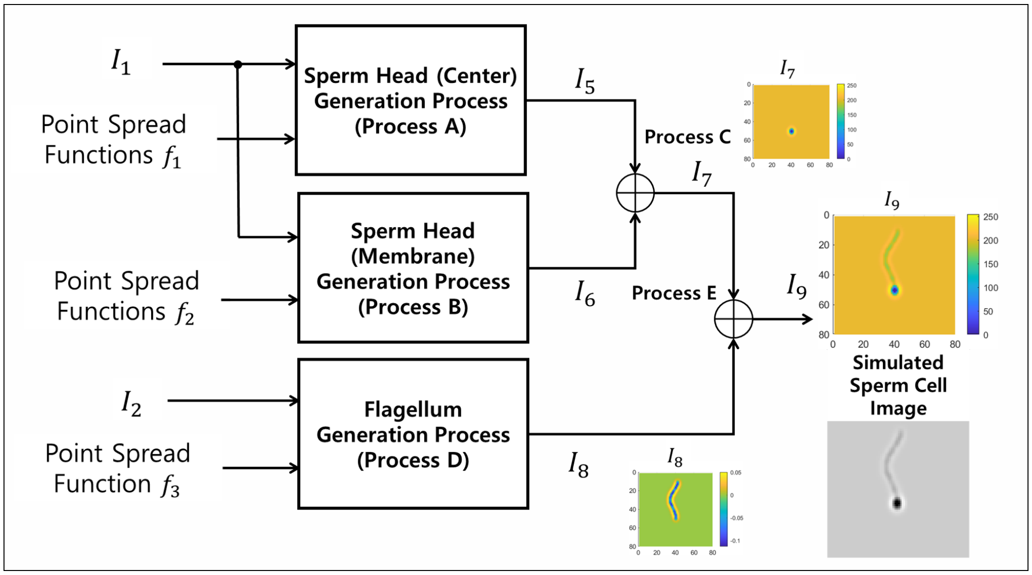

I_1
Point Spread Functions f_1
Sperm Head (Center) Generation Process (Process A)
I_5
Process C
I_7
Sperm Head (Membrane) Generation Process (Process B)
Point Spread Functions f_2
I_6
Process E
I_9
I_2
Point Spread Function f_3
Flagellum Generation Process (Process D)
I_8
Simulated Sperm Cell Image
0
20
40
60
80
250
200
150
100
50
0.05
0
-0.05
-0.1

Figure 2.3 Flowchart of sperm image generation process. Images I_1 and I_2, and point spread functions f_1, f_2, and f_3 are the inputs. The output I_9 is the simulated sperm image. Details of each process is shown in Figures 2.5, 2.7, 2.8, 2.9, and 2.10. Image I_7 shows the image of sperm head (Output of process C). Image I_8 shows the image of flagellum (Output of process D). Image I_9 shows the image of sperm head (Output of process E). Color bars next to I_7-I_9 indicate the corresponding grayscale intensity values. Resulting grayscale image of simulated sperm cell below the colored plot of I_9.

The flowchart of the process of generating the sperm image is shown in Figure 2.3. The inputs of the process are images I_1 (corresponding to the sperm head) and I_2 (corresponding to the curve of the flagellum), and point spread functions f_1, f_2, and f_3. The image of sperm head center I_5 and membrane I_6 are generated using the operations labeled Process A and B, respectively. The inputs for Process A are image I_1 and point spread function f_1, and the inputs for Process B are image I_1 and point spread function f_2. The images of sperm head center I_5 and membrane I_6 are merged in Process C, producing a final image of sperm head I_7. Processes A-C are therefore the sperm head generation process. The final image of the flagellum I_8 is generated using Process D, whose inputs are image I_2 and point spread function f_3. Lastly, the image of sperm head I_7 and flagellum I_8 are merged in Process E, producing the complete image of the sperm cell, I_9. A grayscale image of the simulated sperm cell is shown on just below the full-color simulated cell image in Figure 2.3.

I_1 is an image of N by N pixels where

$$I_1(x,y,t) = \begin{cases} 255 & (x,y) \in \{(x_{H_C}(t), y_{H_C}(t)), (x_{H_L}(t), y_{H_L}(t)), \\ & \qquad (x_{H_H}(t), y_{H_H}(t)), (x_{H_I}(t), y_{H_I}(t))\} \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

N is the size of the frame of the simulation image. Here, $\{(x_{H_C}, y_{H_C})$, (x_{H_L}, y_{H_L}), (x_{H_H}, y_{H_H}), $(x_{H_I}, y_{H_I})\}$ are the locations of sperm heads of cells engaged in circular swimming, linear mean swimming, hyperactive swimming, or no swimming (immotile cells), respectively. I_2 is also an image of N by N pixels where

$$I_2(x,y,t) = \begin{cases} 255 & (x,y) \in \{(x_{tail_C}(k,t), y_{tail_C}(k,t)), (x_{tail_L}(k,t), y_{tail_L}(k,t)), \\ & \quad (x_{tail_H}(k,t), y_{tail_H}(k,t)), (x_{tail_I}(k,t), y_{tail_I}(k,t))\}, \\ & \quad k = 1,2,3,...,M, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

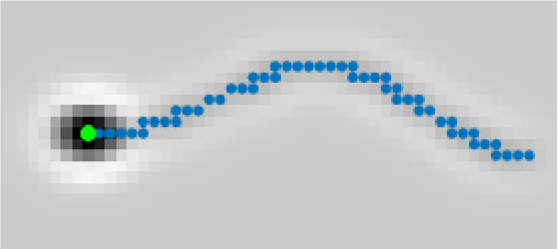


Figure 2.4 Simulated sperm image with points for the location of the head (green) and the arc of the flagellum (blue).

Here, $\{(x_{tail_C}, y_{tail_C}),\ (x_{tail_L}, y_{tail_L}),\ (x_{tail_H}, y_{tail_H}),\ (x_{tail_I}, y_{tail_I})\}$ are the points along the curve of sperm flagellum of circular swimming, linear mean swimming, hyperactive, and immotile cells, respectively. Each flagellum consists of M points (typically $M = 200$). The detailed placing of the sperm head ($\{(x_{H_C}, y_{H_C}), (x_{H_L}, y_{H_L}), (x_{H_H}, y_{H_H}), (x_{H_I}, y_{H_I})\}$) and the points along the curve of sperm flagellum ($\{(x_{tail_C}, y_{tail_C}), (x_{tail_L}, y_{tail_L}), (x_{tail_H}, y_{tail_H}), (x_{tail_I}, y_{tail_I})\}$) are provided in Section 2.2.2.

An image of a simulated sperm cell is shown in Figure 2.4 with the points for the location of the sperm head (where $I_1(x,y) = 255$) and the curve of the flagellum (where $I_2(x,y) = 255$) shown in green and blue, respectively. In the simulation, grayscale values are used in the range between 0 to 255 (256 levels). In the Sections 2.2.1.1-2.2.1.5, the five processes (A-E) are explained in detail. The flowcharts of processes (A-E) are given in

Figures 2.5, 2.7-2.10. Images of I_3-I_9 are accompanied by numbered colorbar that can be used to convert the image to grayscale.

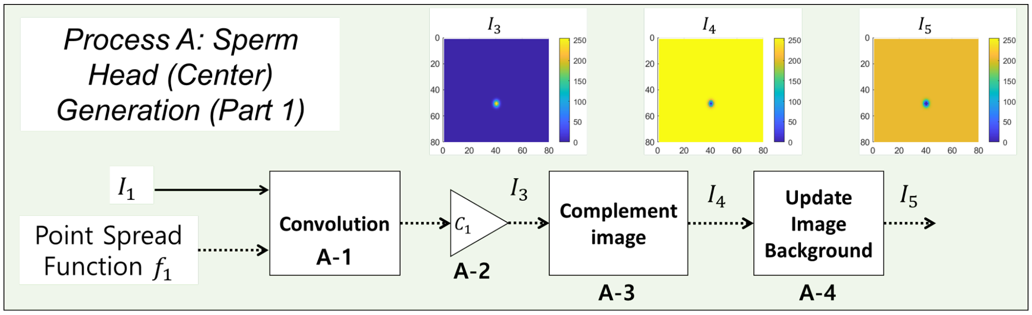


Figure 2.5 Flowchart of sperm head (center) image generation process. Image I_1 and point spread function f_1 are the inputs. The output is the simulated image of sperm head center I_5. I_1 and f_1 is convolved and then scaled by C_1 to generate I_3 (Processes A-1 and A-2). The image I_3 is complemented to generate the image I_4 (Process A-3). Lastly, the background is added to image I_4, generating an image of sperm head center I_5 (Process A-4).

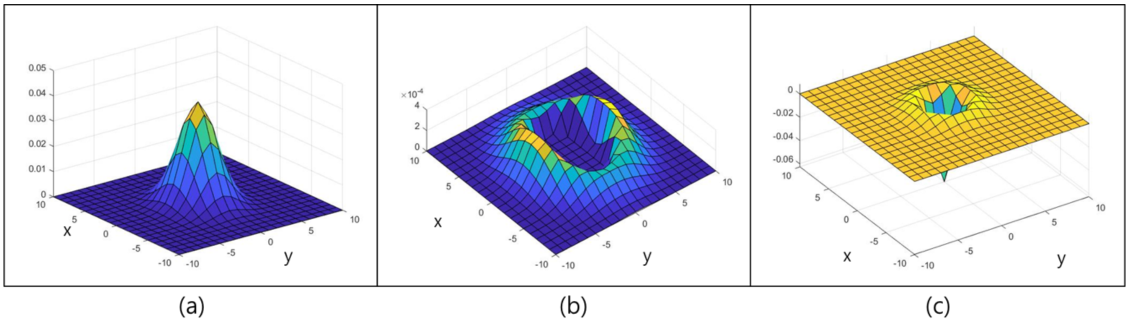


Figure 2.6 Point Spread Functions used in sperm cell image generation process; (a) f_1, (b) f_2, (c) f_3.

2.2.1.1 Process A: Sperm Head (Center) Generation (Part 1). The flowchart of the Process A is shown in Figure 2.5. To generate the oval-like structure of the sperm head, image I_1 is convolved with a 2-D normal distribution filter f_1 (Process A-1); the filter is used as a point spread function (Chapters 3-5 in [43]). The point spread function f_1, a 2-D normal distribution, is defined as

$$f_1(x,y) = \frac{1}{2\pi\sigma_{x_G}\sigma_{y_G}} exp\left(-\left[\frac{(\frac{x}{\sigma_{x_G}})^2 + (\frac{y}{\sigma_{y_G}})^2}{2}\right]\right). \tag{2.3}$$

The 3-D surface plot of f_1 is shown in Figure 2.6a. Standard deviations σ_{x_G} and σ_{y_G} control the length and width of the cell. In the examples shown in this section, $\sigma_{x_G} = 1.86$ px and $\sigma_{y_G} = 2.86$ px. The size of the filter f_1 is 25×25 px (x and y ranges from -12 to 12). The resulting output image is then scaled by a constant C_1 to set the peak intensity value to be 255 and is denoted I_3 (Process A-2 in Figure 2.5). Constant C_1 can be changed to control the intensity value of the cell head. The image I_3 is complemented to produce the image of dark-oval-like center on a white background (Process A-3). In Process A-4, the background for the simulation image is added. Let B_L be the background that will be added to the simulated image. The size of B_L is N by N. In the examples in this section, a uniform background is assumed, where every value in the pixel is equal to 204 ($B_L(x, y) = 204$, 80% of 255). The value $255 - B_L$ is subtracted from the image I_4 and any value below 0 is set to 0. The resulting image is denoted image I_5 ($I_5 = \max[0, I_4 - (255 - B_L)]$).

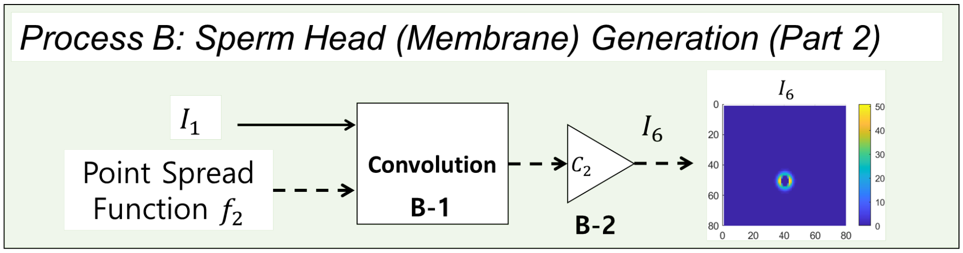

Process B: Sperm Head (Membrane) Generation (Part 2)
I_1
Point Spread Function f_2
Convolution
B-1
C_2
B-2
I_6
I_6
0
20
40
60
80
0
20
40
60
80
50
40
30
20
10
0

Figure 2.7 Flowchart of the image generation process of the sperm membrane. Image I_1 and point spread function f_2 are the inputs. The output is the simulated image of sperm membrane I_6. I_1 and f_1 is convolved and then scaled by C_1 to generate I_6 (Processes B-1 and B-2).

2.2.1.2 Process B: Sperm Head (Membrane) Generation (Part 2).

The flowchart of Process B is shown in Figure 2.7. In typical semen image samples, one can observe a halo-like membrane surrounding the sperm head, as shown on the left of Figure 2.1 (referred to as “a horseshoe-shaped halo” in [9]). To generate this halo-like membrane around the cell, a modified version of Laplacian of the 2-D normal distribution filter is used as the second point spread function on image I_1. The Laplacian of the 2-D normal distribution filter is defined as

$$g(x,y) = \nabla^2 \left(\frac{1}{2\pi\sigma_{x_L}\sigma_{y_L}} exp\left(- \left[\frac{(\frac{x}{\sigma_{x_L}})^2 + (\frac{y}{\sigma_{y_L}})^2}{2} \right] \right) \right). \tag{2.4}$$

The point spread function f_2 is defined as

$$f_2(x,y) = \max(0, g(x,y)), \tag{2.5}$$

The resulting point spread function f_2 is shown in Figure 2.6b. Standard deviations σ_{x_L} and σ_{y_L} control the length and width of the cell membrane. In the examples in this section, $\sigma_{x_L} = 2.79$ px and $\sigma_{y_L} = 4.29$ px. The size of the filter f_2 is 25×25 px (x and y ranges from -12 to 12). The point spread function f_2 is convolved with image I_1, generating the image of a membrane (Process B-1). The resulting output image is then scaled by constant C_2 (Process B-2) to set the peak intensity to be 51 (peak intensity value of cell membrane, 20% of 255). This image of the membrane is denoted I_6. Constant C_2 can be changed to control the intensity value of the cell membrane.

2.2.1.3 Process C: Sperm Head Generation (Part 3).

The flowchart of Process C is shown in Figure 2.8. In Process C, the images of the sperm head center I_5 and the image of the flagellum I_6 are added to complete the image of the sperm head. The resulting sperm head image is shown in Figure 2.8 and is denoted I_7.

2.2.1.4 Process D: Sperm Flagellum Generation.

The flowchart of Process D is shown in Figure 2.9. To generate the flagellum, image I_2 (Equation (2.2)) is convolved with point spread function f_3 (Equation (2.6)) to generate the image of uniform calibre look of the flagellum. The point spread function f_3 is defined as the Laplacian of Gaussian filter (equal to the Equation (2.4) with $\sigma_{x_L} = \sigma_{y_L} = \sigma_f$), namely

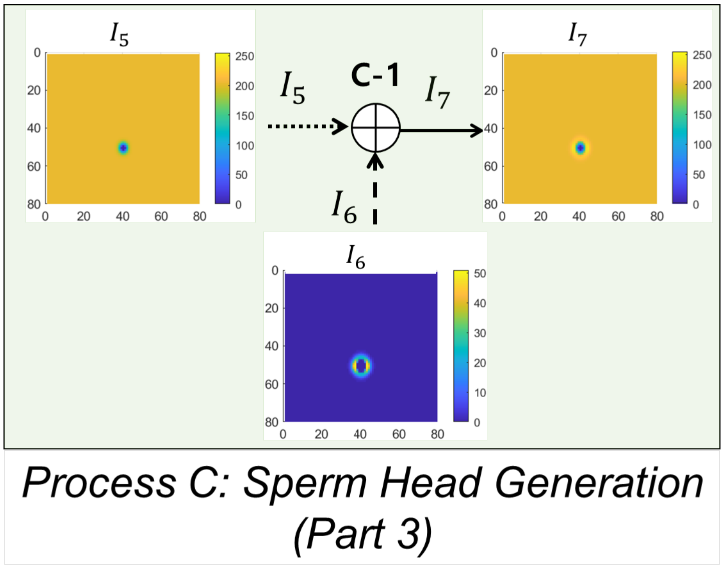


Figure 2.8 Diagram of sperm head image merging process. Images I_5 and I_6 are the inputs. The output is the simulated image of sperm head I_7. I_5 and I_6 are added to generate I_7 (Process C-1).

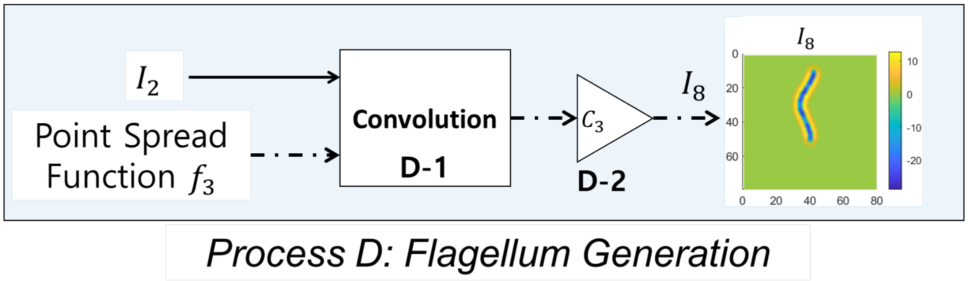


Figure 2.9 Flowchart of flagellum generation process. Image I_2 and point spread function f_3 are the inputs. The output is the simulated image of flagellum I_7. I_2 and f_3 is convolved and then scaled by C_1 to generate I_7 (Processes D-1 and D-2).

$$f_3(x,y) = \nabla^2 \left(\frac{1}{2\pi\sigma_f^2} exp \left(- \left[\frac{(\frac{x}{\sigma_f})^2 + (\frac{y}{\sigma_f})^2}{2} \right] \right) \right). \tag{2.6}$$

The 3-D surface plot of f_3 is shown in Figure 2.6c. Standard deviation σ_f control the width of the flagellum. In the examples in this section, $\sigma_f = 1.5$ px. The size of the filter f_3 is 25×25 px (x and y ranges from -12 to 12). The output image is then scaled by constant C_3 to set the peak intensity of the membrane as 13 (approximately 5% of 255). Constant

C_3 can be changed to control the intensity of the flagellum. The resulting image of the flagellum is a long membrane-like curve. The scaled image of the flagellum is denoted I_8.

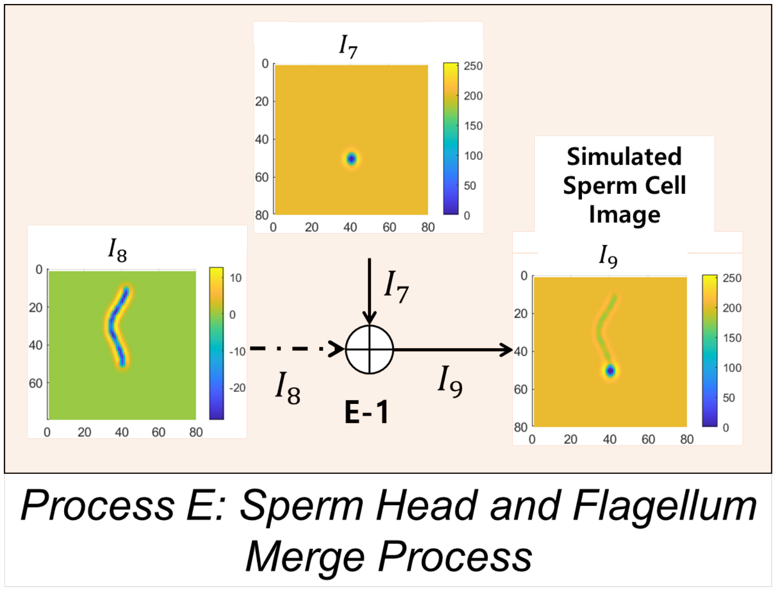

I_7
0
20
40
60
80
250
200
150
100
50
Simulated Sperm Cell Image
I_9
I_8
10
0
-10
-20
E-1
Process E: Sperm Head and Flagellum Merge Process

Figure 2.10 Diagram of sperm head and flagellum image merging process. Images I_7 and I_8 are the inputs. The output is the simulated image of sperm cell I_9. I_7 and I_8 are added to generate I_9 (Process E-1).

2.2.1.5 Process E: Sperm Head and Flagellum Merge. Finally in Process E, the image of cell head, I_7, and the image of the flagellum, I_8, are added to generate the image of a sperm cell (I_9 in Figures 2.3 and 2.10). Process E is shown in Figure 2.10.

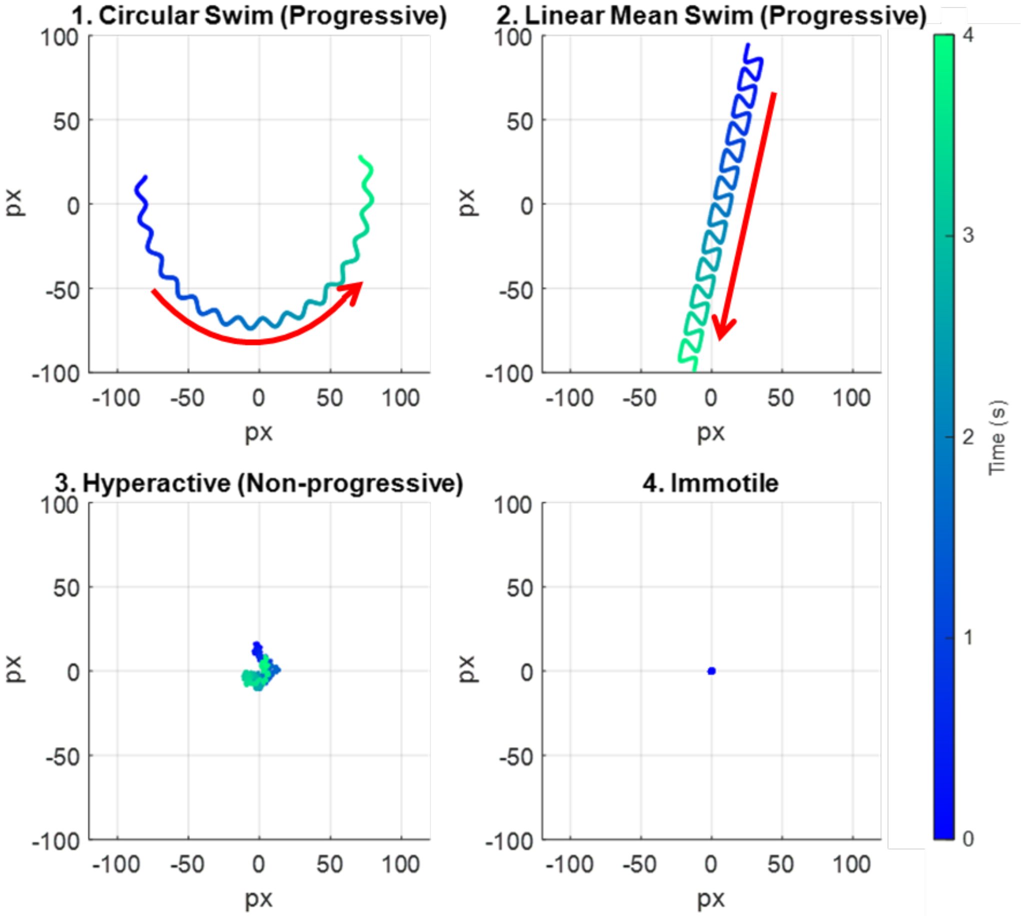


Figure 2.11 Simulated swimming path of the four swimming modes: circular swim, linear mean swim, hyperactive, and immotile. The color bar indicates the color of a track respect to time. The duration of the movement is 4 seconds.

2.2.2 Swimming Models

In this section, the swim/movement models of the head and the flagellum of a sperm cell are described. The swimming models describe how the position of the head and the flagellum change over time. These positions define the points used to generate I_1 (cell head, Equation (2.1)) and I_2 (cell flagellum, Equation (2.2)).

The sperm movements are categorized into these four swimming modes:

1. Circular swim (progressive),

2. Linear mean swim (progressive),

3. hyperactivated (non-progressive),

4. Immotile, or dead.

Figure 2.11 shows simulated tracks for the four swimming modes for about 4 seconds. The tracks indicate the location of the head of a cell and the arrow indicates the direction of movement.

Sperm cell motility in 2-D images is often divided into progressive, non-progressive, and immotile [1]. Cells with progressive movement actively propel themselves forward, either linearly or on a large circle path. Examples of the two types of progressive movements are (1) circular swim and (2) linear mean swim, as shown on top of Figure 2.11. The red arrow indicates the direction of movement for the cell. Non-progressive movement refers to the motility exhibited by (3) hyperactivated sperm cells, which do not travel significantly away from their initial location, and yet are described as vigorous, “whiplash type”, or “frantic” [44, 45]. An example is shown in bottom left of Figure 2.11, where the trace of a hyperactive sperm cell shows active movement, but the cell does not progress much in any direction over time. (4) Immotile movements define cells which show barley any movement at all (shown in the bottom right of Figure 2.11).

In Sections 2.2.2.1-2.2.2.3, each movement are described in detail.

2.2.2.1 Progressive Swimming Models. 2-D progressive sperm swimming patterns were categorized as either circular or linear mean [39]. Based on the progressive sperm tracks recorded in [9, 14, 35, 39, 46], the models to describe circular and linear mean movements were synthesized.

2.2.2.1.1 Circular Swim Model. A sperm cell exhibiting a circular swim moves along a circular path with oscillations about this path caused by the beatings of the flagellum [39]. This circular swim can be represented as a sinusoidal modulated circular path (Figure 2.12). The equations used to model the movement of the head and the flagellum of a circular

Table 2.1 Equations for Simulating Circular Swimming Cell

Circular Swim Model - Head

$$x_{H_C}(t) = (r_c + a\sin(2\pi f_s t))\cos(2\pi f_c t) + C_{x_c} \tag{2.7a}$$

$$y_{H_C}(t) = (r_c + a\sin(2\pi f_s t))\sin(2\pi f_c t) + C_{y_c} \tag{2.7b}$$

Circular Swim Model - Flagellum

$$x_{T_c}(k,t) = b(k)\sin\left[2\pi\left(\frac{k}{M} - f_s t\right)\right] \tag{2.8a}$$

$$y_{T_c}(k) = -\lambda\frac{k}{M} \tag{2.8b}$$

$$\begin{bmatrix} x_{tail_C} \\ y_{tail_C} \end{bmatrix} = R(2\pi f_s t)\begin{bmatrix} x_{T_c} \\ y_{T_c} \end{bmatrix} + \begin{bmatrix} x_{H_C} \\ y_{H_C} \end{bmatrix} \tag{2.9}$$

$$R(\cdot) = \begin{bmatrix} \cos(\cdot) & -\sin(\cdot) \\ \sin(\cdot) & \cos(\cdot) \end{bmatrix} \tag{2.10}$$

$$b(k) = a\left(\alpha\frac{\lambda k}{M} + \beta\right),\ \alpha = 0.02,\ \beta = 0.8 \tag{2.11}$$

Table 2.2 Nomenclature for Circular Swim Model

(x_{H_C}, y_{H_C}) : head position of circular swimming cell
r_c : radius of the circular path
a : amplitude of the sinusoid modulated on the circular path
f_s : frequency of the sinusoid modulated on the circular path (Hz)
f_c : frequency of the circular cycle (cycle/sec)
(C_{x_c}, C_{y_c}) : vertical and horizontal offset constant
(x_{Tail_C}, y_{Tail_C}) : a set of k points along the center of the flagellum of circular swimming cell
λ : wavelength of flagellum (distance between the start and the end of the flagellum)
$R(\cdot)$: rotation matrix
$b(\cdot)$: local variation in beating amplitude along the flagellumm

swimming cell are given in Table 2.1. Detailed nomenclature for the variables used in Table 2.1 are given in Table 2.2. The 2-D position of the cell's head is determined by Equations (2.7a), (2.7b). r_c is the radius of the overall circular path, f_s is the frequency of the sinusoid modulated on the circular path (Hz), a is the amplitude of the sinusoid

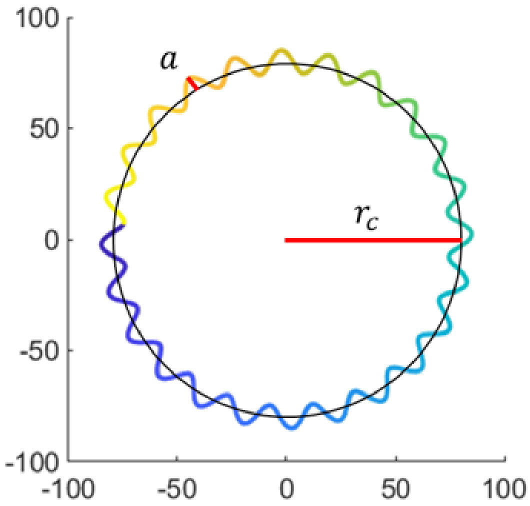


Figure 2.12 Trajectory of simulated circular swimming cell.

modulated on the circular path, and f_c is the frequency of the circular cycle (cycle/sec). C_{x_c} and C_{y_c} are horizontal and vertical offsets, respectively.

$(x_{T_c}(k,t), y_{T_c}(k))$ are cartesian coordinates of the points along the center curve of the flagellum with $k = 1, 2, 3, ..., M$ where $M = 200$. The flagellum of circular swimming cells follow Dresdner's model of sperm flagellum (Equation (2.8a)) [30]. Equation (2.8a) determines the oscillation of the flagellum (x_{T_c}), and Equation (2.8b) determines the end to end length (wavelength λ) of the flagellum (distance between the start and the end of the flagellum). $b(k)$ is the local variation in beating amplitude along the flagellum. The points (x_{T_c}, y_{T_c}) are rotated using the rotation matrix $R(\cdot)$ to match the direction of movement and shifted to the location of the sperm head (x_{H_C}, y_{H_C}) (Equation (2.9)); the points of the flagellum of a circular swimming cell are denoted (x_{tail_C}, y_{tail_C}). In the examples shown in this chapter, the local variation in the beating amplitude, $b(k)$, follows the affine function $b(k) = a\left(\alpha\frac{\lambda k}{M} + \beta\right)$, with $\alpha = 0.02, \ \beta = 0.8$ for circular swimming cell. Here, a is the amplitude of a sinusoid modulated on the circular path, and λ is the wavelength of the flagellum.

An example of a simulated flagellum of a circular swimming cell is shown in Figure 2.13. On the left is a simulated image of a circular swimming cell with the path shown in a

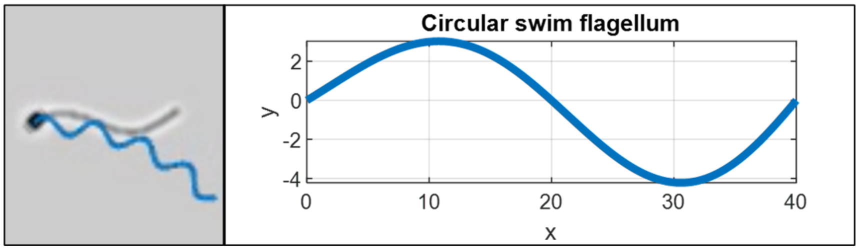


Figure 2.13 (Left) Simulated image of circular swimming cell with past track shown in blue. (Right) Plot of the flagellum of circular swimming cell. The example shows simulated flagellum for wavelength (λ) = 40 pixels and amplitude (a) = 4 pixels.

blue line. On the right is a curve which represents the flagellum of that cell. The example in this image uses amplitude of $a = 4$ pixels and wavelength of $\lambda = 40$ pixels.

Table 2.3 Equations for Simulating Linear Mean Swimming Cell - Head

Linear Mean Swim Model - Head
$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} V\cos(\theta_r)t + C_{x_L} \\ V\sin(\theta_r)t + C_{y_L} \end{bmatrix} \qquad (2.12)$
$P = \begin{bmatrix} P_x(t) \\ P_y(t) \end{bmatrix} = \begin{bmatrix} \frac{r_v}{2}\sin(4\pi f_l t) \\ \frac{r_h A_c}{2}[\sin(2\pi f_l t) + A_{har}\sin(6\pi f_l t)] \end{bmatrix} \qquad (2.13)$
$A_c = \dfrac{1}{\max\limits_{\theta}(\sin(\theta) + A_{har}\sin(3\theta))} \qquad (2.14)$
$\begin{bmatrix} x_{H_L} \\ y_{H_L} \end{bmatrix} = R(\theta_r)P + \begin{bmatrix} x_c \\ y_c \end{bmatrix} \qquad (2.15)$

2.2.2.1.2 Linear Mean Swim Model. In the linear mean swim, a sperm cell moves along a straight-line path, rolling side to side as it propels itself forward. This feature causes a ribbon-like movement about the straight line. The equations used to model the movement of the head and the flagellum of linear mean swimming cell are given in Tables 2.3 and 2.4.

The cell head movement (ribbon-like movement along a linear path) is generated using Equations (2.12)-(2.15). The position along the linear path is denoted as (x_c, y_c) and the position along the ribbon-like movement is denoted as $(P_x(t), P_y(t))$ The resulting path is shown in Figure 2.14a.

Table 2.4 Equations for Simulating Linear Mean Swimming Cell - Flagellum

Linear Mean Swim Model - Flagellum

$$\begin{bmatrix} x_o(k,t) \\ y_o(k,t) \end{bmatrix} = \begin{bmatrix} b_1(k)x_T(k,t) \\ b_2(k)y_T(k,t) \end{bmatrix} \tag{2.16}$$

$$x_T(k,t) = \frac{r_v}{2}\sin\left(4\pi\left(\frac{k}{M} + f_l t\right)\right) \tag{2.17a}$$

$$y_T(k,t) = \frac{r_h}{2}\sin\left(2\pi\left(\frac{k}{M} + f_l t\right)\right) \tag{2.17b}$$

$$\begin{bmatrix} x_{LM}(k,t) \\ y_{LM}(k,t) \end{bmatrix} = \begin{bmatrix} x_o(k,t) \\ y_o(k,t) \end{bmatrix} - \begin{bmatrix} \frac{\lambda k}{M} \\ 0 \end{bmatrix} \tag{2.18}$$

$$y_{LM_2}(k,t) = y_{LM}(k,t) - [P_y(t) - y_T(0,t)]b_3(\psi) \tag{2.19}$$

$$\begin{bmatrix} x_{tail_{LM}} \\ y_{tail_{LM}} \end{bmatrix} = R(\theta_r)\begin{bmatrix} x_{LM} \\ y_{LM_2} \end{bmatrix} + \begin{bmatrix} x_{H_L} \\ y_{H_L} \end{bmatrix} \tag{2.20}$$

$$b_1(k) = \frac{1}{1 + e^{(\alpha k/M + \beta)}} \tag{2.21a}$$

$$b_2(k) = e^{-\gamma_1 k/M} \tag{2.21b}$$

$$b_3(k) = 1 - e^{-\gamma_2 k/M} \tag{2.21c}$$

Table 2.5 Nomenclature for Linear Swim Model

(x_{H_L}, y_{H_L}) : position of the sperm head of linear swimming cell

(C_{x_L}, C_{y_L}) : horizontal and vertical offset constant (pixels)

V : straight line path velocity (pixels/sec)

f_l : rate of change in ribbon angle (Hz)

r_h, r_v : width and height of ribbon (pixels)

A_{har} : user defined ratio between the first and the third harmonics

A_c : correction constant for defined width of the ribbon

θ_r : direction of the forward movement (radian)

$(x_{Tail_{LM}}, y_{Tail_{LM}})$: a set of k points along the center of the flagellum of linear mean swimming cell

$b_1(k), b_2(k)$: local horizontal and vertical variation in beating amplitude along the flagellum

$b_3(k)$: flagellum position correction function

The straight-line path is a linear function of time t with initial positions of (C_{x_L}, C_{y_L}). The straight-line path is a line segment from initial positions (C_{x_L}, C_{y_L}) to the point (x_c, y_c) (as shown in Figure 2.14). The horizontal and vertical positions along the linear path (x_c, y_c) change at the rate of horizontal and vertical velocity in the direction of forward

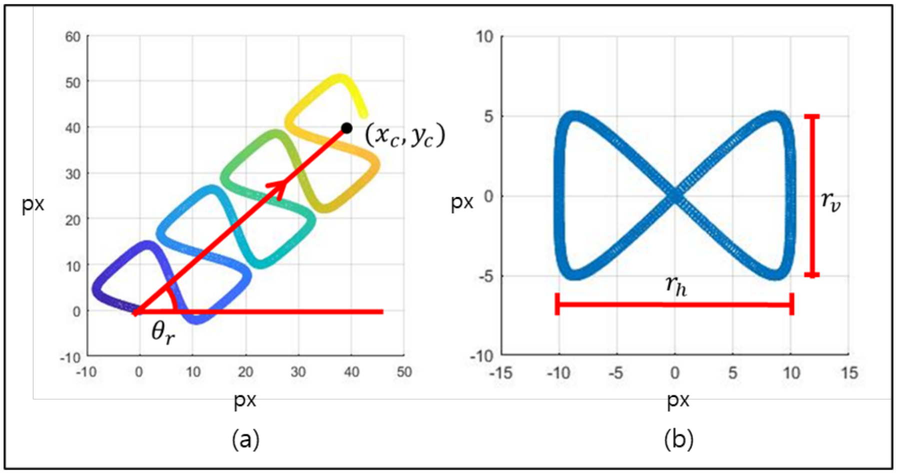


Figure 2.14 (a) Trajectory of simulated linear mean swimming cell (Equations (2.12)-(2.15)). (b) Ribbon-like oscillatory movement along the straight-line path of linear mean swimming cells (Equation (2.13)).

movement θ_r (Equation (2.12)). The position along a “ribbon-like” path $(P_x(t), P_y(t))$ is generated using Equations (2.13) and (2.14). The ribbon is shown in Figure 2.14b, where r_v is the height of the ribbon and r_h is its width. $P_x(t)$ and $P_y(t)$ are periodic functions of time t. $P_x(t)$ is a sinusoid with a frequency of $2f_l$ and amplitude of $\frac{r_v}{2}$. $P_y(t)$ is a sum of two sinusoids of frequencies of f_l and $3f_l$ and amplitudes of $\frac{r_h A_c}{2}$ and A_{har}, respectively. A_c is a correction constant to set the amplitude of function $P_y(t)$ as $\frac{r_h}{2}$ (Equation (2.14)). The resulting ribbon-like path is shown in Figure 2.14b. A_{har} is the ratio between the two sinusoids of function $P_y(t)$. The shapes of the ribbon for varying values of A_{har} are shown in Figure 2.15. The examples in this chapter use $A_{har} = 0.1$.

The position of the cell along the ribbon, $(P_x(t), P_y(t))$, is multiplied by rotation matrix $R(\cdot)$ resulting in a rotation by θ_r (Equation (2.15), $R(\theta_r)P$). Lastly, The position on the ribbon-like path $R(\theta_r)P$ is added to the position along the straight-line path (x_c, y_c) to define the position of the sperm head at all times (Equation (2.15)).

The flagellum is generated using Equations (2.16)-(2.21). Dresdner’s model is used to define the horizontal and vertical oscillations in the flagellum $(x_o(k,t), y_o(k,t))$. $x_o(k,t)$ and $y_o(k,t)$ are discrete set of points for $k = 1, 2, 3, ..., M$ realized at time t. The value of M in the simulation is 200. An example of simulated flagellum of linear mean swimming

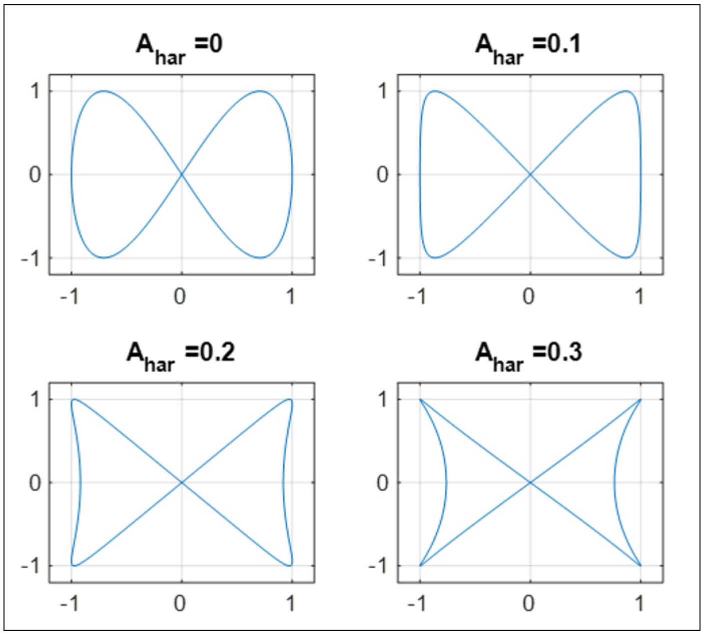


Figure 2.15 Effect of different values of A_{har} on the shape of the ribbon of linear mean swimming cells.

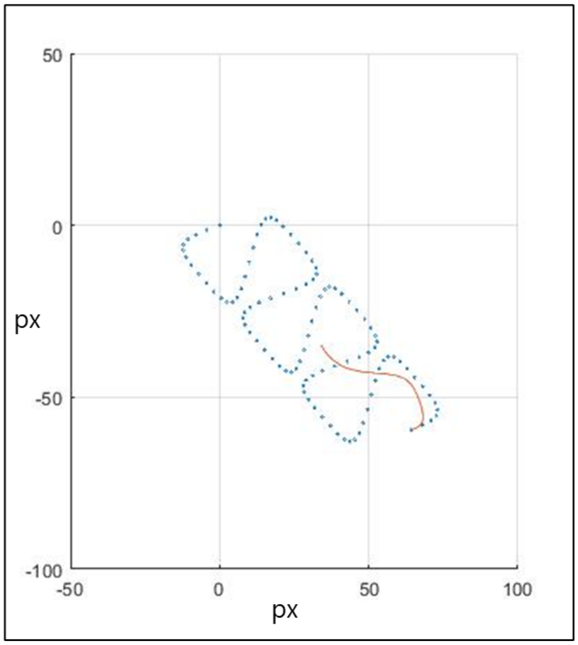


Figure 2.16 Example of sperm flagellum generation for linear mean swimming cell. (Orange) Simulated flagellum. (Blue) Track of linear mean swimming cell.

cell is shown in Figure 2.16. The blue dots represent the successive location of the sperm head. For one of these locations, the associated flagellum is shown in orange. The fundamental frequencies of the ribbon in horizontal ($2f_l$ for $P_x(t)$) and vertical directions

(f_l for $P_y(t)$) are used to determine the movement of the flagellum (Equations (2.13) and (2.17)). The local variation in beating amplitude for $x_T(k,t)$ and $y_T(k,t)$ are $b_1(\psi)$ and $b_2(k)$, respectively (Equations (2.21a, 2.21b)).

The function $b_1(k)$ (Equation (2.21a)) is a transformed sigmoid function, where the values of α and β determine the shift and the compression/expansion of the function; $b_2(k)$ (Equation (2.21b)) is a decaying exponential function. Together, these functions provide the desirable realistic visual effect.

The line segment from the head to the end of the flagellum is defined as $f(k) = -\frac{\lambda k}{M},\ k = 1, 2, 3, ..., M$. The oscillations $x_o(k,t)$ and $y_o(k,t)$ are added to this straight line (Equation (2.18)).

The differences in the vertical location of the head $P_y(t)$ and the vertical position of the flagellum $y_T(k,t)$ cause offset in placement of the flagellum because the model have only considered the fundamental frequency component, f_l, of $P_y(t)$ to generate ($y_T(k,t)$). This is corrected using Equations (2.19) and (2.21c).

Lastly, the flagellum is rotated to match the direction of movement and shifted to the position of the cell head (Equation (2.20)). The constants in Equations (2.21a)-(2.21c) in the example were set to: $\alpha = 22, \beta = -2, \gamma_1 = 5, \gamma_2 = 1.5$.

2.2.2.2 Non-progressive Swim Model. Hyperactive swim, described to be non-progressive, 'frantic', and 'vigorous' is modeled through as Brownian motion to reflect the randomness in the cell movement [47]. The equations used to model the movement of the head and the flagellum of hyperactive swimming cell are given in Table 2.6.

For each dimension (x and y), the location of a cell is given in Equation (2.22), where (μ_x, μ_y) are drift coefficients of Brownian motion, ($\sigma_{x_b}, \sigma_{y_b}$) are diffusion coefficients of Brownian motion, and $W(t)$ is the standard 1-dimensional Brownian motion ($W(t) \sim N(0,t)$). The simulation assumed no drift ($\mu_x = \mu_y = 0$), and the diffusion coefficients are assumed to be the same for the horizontal (x) and the vertical (y) directions ($\sigma^2_{x_b} = \sigma^2_{y_b} =$

Table 2.6 Equations for Hyperactive Cell

Hyperactive Swim Model - Head
$x(t) = \mu_x t + \sigma_{x_b} W(t)$ and (2.22a) $y(t) = \mu_y t + \sigma_{y_b} W(t)$ (2.22b) $x_{H_H}(t) = x_{H_H}(t-T) + \sigma_b W(T)$ and (2.23a) $y_{H_H}(t) = y_{H_H}(t-T) + \sigma_b W(T),$ (2.23b)
Hyperactive Swim Model - Flagellum
$(x_{tail_H}(k,t), y_{tail_H}(k,t)) =$ Linear interpolation on a set of points $[(x_{H_L}(0), y_{H_L}(0)), (x_{H_L}(1), y_{H_L}(1)),$ $(x_{H_L}(2), y_{H_L}(2)), ..., (x_{H_L}(n), y_{H_L}(n))]$ (2.24) where $x_{H_L}(n) = x_{H_H}(t-nT),$ $y_{H_L}(n) = y_{H_H}(t-nT)$

Table 2.7 Nomenclature for Hyperactive Cell Swim

(μ_x, μ_y) : drift coefficients of Brownian motion
$(\sigma_{x_b}, \sigma_{y_b})$: diffusion coefficients of Brownian motion
$W(t)$: standard 1-dimensional Brownian motion ($W(t) \sim N(0,t)$
(x_{H_H}, y_{H_H}) : position of the sperm head of hyperactive cell
T : the difference in time between each simulation frame
(x_{Tail_H}, y_{Tail_H}) : a curve along the center of the flagellum of hyperactive cell

σ_b^2). The location of the sperm head of hyperactive cell is defined by Equation (2.23), where T is the difference in time between each simulation frame ($\frac{sec}{\text{frame}}$).

The flagellum of the hyperactive swimming cell is defined to be the linear interpolation on a set of points $[(x_{H_L}(t), y_{H_L}(t)), (x_{H_L}(t-T), y_{H_L}(t-T)), (x_{H_L}(t-2T), y_{H_L}(t-2T)), ..., (x_{H_L}(t-nT), y_{H_L}(t-nT))]$ (a set of location of the cell from time $t - nT$ to t, Equation (2.24)). In the simulation, the value nT is set to 0.2 seconds.

Table 2.8 Equations for Immotile Cell

Immotile Swim Model - Head
$x_{H_I}(t) = x_{H_I}(0)$ and (2.25a) $y_{H_I}(t) = y_{H_I}(0)$ (2.25b)
Immotile Swim Model - Flagellum
$(x_{tail_I}(k,t), y_{tail_I}(k,t)) =$ Linear interpolation on a set of points $[(x_{I_L}(0), y_{I_L}(0)), (x_{I_L}(1), y_{I_L}(1)),$ $(x_{I_L}(2), y_{I_L}(2)), ..., (x_{I_L}(n), y_{I_L}(n))]$ where $x_{I_L}(n) = x_{H_H}(-nT),$ $y_{I_L}(n) = y_{H_H}(-nT)$ (2.26)
(x_{H_I}, y_{H_I}) : position of the sperm head of immotile cell (x_{Tail_I}, y_{Tail_I}) : a curve along the center of the flagellum of immotile cell

2.2.2.3 Immotile or Dead cell Model. Immotile or dead cells are simulated as non-moving points. The equations used to model the movement of the head and the flagellum of an immotile cell are given in Table 2.8.

Throughout the simulation, the head and the flagellum of the cell will not move (Equations (2.25) and (2.26)). The flagellum is generated by taking a snapshot of the flagellum of a hyperactive cell. The flagellum of the immotile cell is generated using Equation (2.26).

In Figure 2.17, an example of the simulated semen sample is shown, where the cells of four different swimming modes can be seen. The trajectories of each cell are shown in blue lines.

2.3 Testing Setup and Assessment Metric

For testing, simulate images are based on real semen image samples. The parameters extracted to simulate semen images were the image background B_L, the noise variance σ_N^2,

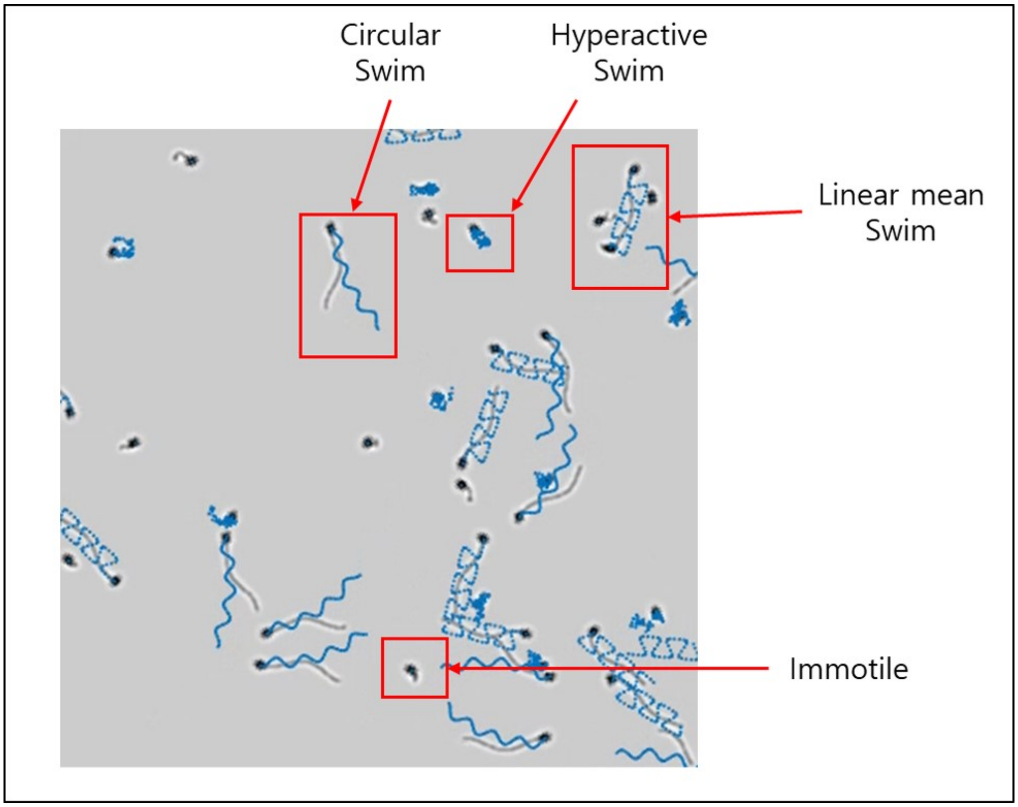


Figure 2.17 Example of simulated image with track of each cell shown in blue.

the size of sperm head, the number of cells in the image N_C, and the number of non-moving cells N_D. The semen samples used for testing are available in [48].

Two possible use cases of the simulation were explored. First, five segmentation and localization techniques are tested and their performances are assessed. The performance of segmentation and localization was evaluated in terms of optimal subpattern assignment (OSPA) distance ($c = 20,\ p = 2$) [49], precision and recall rates [50].

Second, the performance of four (4) different tracking algorithms were tested. The four tracking algorithms are: nearest neighbor (NN), global nearest neighbor (GNN), probabilistic data association filter (PDAF), and joint probabilistic data association filter (JPDAF) [51]. The performance of the tracking algorithms were assessed in terms of Multiple Object Tracking Precision (MOTP) and Multiple Object Tracking Accuracy (MOTA) with cutoff distance of $c_T = 5$ px [52].

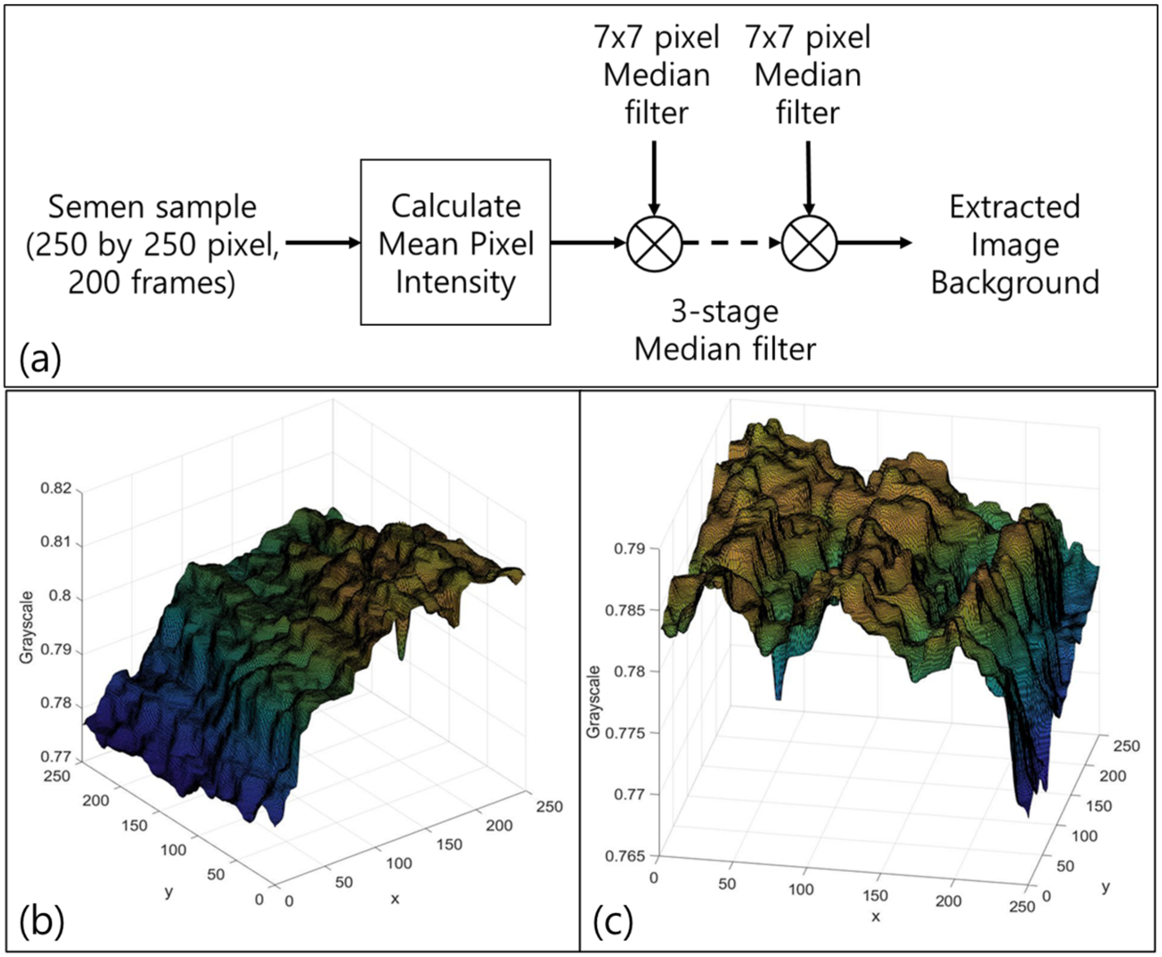
2.3.1 Semen Sample Collection and Preparation

The semen samples used in the study were collected by the In-Vitro Fertilization Laboratories at Penn Fertility Care (part of Penn Medicine) in accordance with the policies of the University of Pennsylvania. Each specimen was allowed to liquefy for 30-40 minutes at room temperature and was washed in media. Washing separates sperm cells from seminal plasma and consists of three cycles of centrifugation. First, ≤ 3 mL of semen is layered onto a 1.0 mL preparation of density gradient media (0.9 mL PureSperm 100, Spectrum Technologies and 0.1 mL Enhanced WG, modified human tubal fluid with HAS and Gentamicin, Conception Technologies) and is centrifuged for 20 min at 400 g. The pellet is resuspended in 2 mL Enhanced WG and centrifuged again for 5 min at 400 g. Lastly, the supernatant is discarded and the pellet is resuspended in 0.5 mL Enhanced WG and 5 μL is pipetted onto a 20 μm deep Vitrolife MicroCell chamber. A microscope stage warmer is used to regulate the temperature of each sample at 37 °C during phase contrast imaging. For a detailed explanation on the preparation process of the samples, please refer to [10]. Specifically, the samples used in this chapter are sample ID005 (denoted as sample 1) and sample ID007 (denoted as sample 2).

2.3.2 Feature Extraction from Human Semen Samples

The key parameters were extracted from two real semen samples of 250 by 250 pixels with a duration of 200 frames. The ground truth locations of each sperm cell in each image were labeled manually by a human.

To obtain the image background B_L, the mean of grayscale intensity values of each pixel was calculated for the duration of 200 frames. Afterwards, the mean grayscale image was convolved with a series of three 7×7 median filters (Chapter 3 in [43]). The output of the 3-stage median filters was used as the image background B_L for the subsequent simulations. The flowchart of the background extraction process is shown in Figure 2.18a



Sample 1 (ID005)

Sample 2 (ID007)

Figure 2.18 (a) Flowchart of the background extraction process. 3D surface plots of the background model for (b) sample 1 - ID005 and (c) sample 2 - ID007.

and the obtained image background for sample 1 and sample 2 are shown in Figures 2.18b and 2.18c, respectively.

To mimic the noise present in each human semen sample, the noise variance σ_N^2 for each sample was calculated. First, the variance of each pixel for 200 frames was calculated. Afterwards, the median value of the variance of all pixels in the sample was selected to be the noise variance σ_N^2 for the simulation. The noise in the images was assumed to be additive Gaussian. After the simulated images were generated, the images were contaminated by additive Gaussian noise with zero mean and variance σ_N^2.

The size of the sperm head was obtained by segmenting the sperm heads in each frame and finding the median length of the major and minor axes of the segmented heads (Chapter 11 in [43]). To find the two radii of the sperm heads along the major (r_M) and minor (r_m) axis, the median length of each one of the major and minor axes was divided by two. The values of σ_{y_G} and σ_{x_G} of the 2-D Gaussian filter (of the form of Equation (2.3)) were equal to the lengths of radii along the major and minor axes ($\sigma_{y_G} = r_M, \sigma_{x_G} = r_m$). The values of σ_{y_L} and σ_{x_L} of the 2-D LoG filter (of the form of Equation (2.4)) were equal to 1.5 times the lengths of radii along major and minor axis ($\sigma_{y_L} = 1.5r_M, \sigma_{x_L} = 1.5r_m$). The value of σ_f for point spread function f_3 was fixed at 1.5.

In each sample, the number of sperm cells was counted across 200 frames. The number of cells N_C in the simulated image was the maximum number of cells found within 200 frames of each semen sample. The number of non-moving cells N_D was the number of dead/immotile cells found in each sample. The simulation parameters that were extracted from the two samples are shown in Table 2.9. As shown in Table 2.9, sample 1 had three non-moving cells and sample 2 had no non-moving cells.

Table 2.9 Simulation Parameters for Segmentation and Localization Testing

	Noise Variance σ_N^2	Radius r_M (Major axis)	Radius r_m (Minor axis)	Number of cells N_C	Number of non-moving cells N_D
Sample 1	8.22×10^{-6}	2.86 px	1.86 px	10	3
Sample 2	1.03×10^{-4}	2.90 px	1.66 px	8	0

2.3.3 Performance Assessment Metrics for Segmentation and Localization

The performance of segmentation and localization algorithm on the images was assessed using the optimal subpattern assignment (OSPA) distance [49], and the values of precision and recall. The equation of OSPA distance is given as

$$\overline{d}_p^{(c)}(X, Y) := \left(\frac{1}{n}\left(\min_{\pi\in\prod_n}\sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m)\right)\right)^{(1/p)}, n \geq m. \tag{2.27}$$

Here, X is the set of true object locations ($X = \{x_1, ..., x_m\}$). Y is the set of estimated object locations ($Y = \{y_1, ..., y_n\}$). The parameter p is the order. The parameter c is the cut-off distance. m is the number of true objects. n is the number of detected objects.

$d^{(c)}(x, y) := \min(c, d(x, y))$ is the distance between x and y with cut-off $c > 0$. For this study, $p = 2$ and $c = 20$ pixels. For $n < m$, $\overline{d}_p^{(c)}(X, Y) := \overline{d}_p^{(c)}(Y, X)$. The OSPA distance is "large" if (1) the distance between the detection and the ground truth is large, and (2) if the difference in number of detections and ground truth objects are large. A perfect segmentation and localization algorithm would have OSPA distance of 0 (no difference between the ground truth and the detections). A poor algorithm would have distance value close to the cut-off distance $c = 20$ (large differences between the ground truth and the detections).

Precision is defined as $TP/(TP + FP)$ and recall is defined as $TP/(TP + FN)$, where TP, FP, and FN refer to true positive, false positive, and false negative, respectively. True positive is defined to be the number of matches in detection and ground truth within the cutoff distance ($c = 20$ pixels), false positive is defined to be the number of false alarms, and false negative is defined to be the number of missed detections. The ground truth locations of real semen samples were manually labeled by human, and the ground truth locations of simulated images were provided by the simulation software.

2.3.4 Performance Assessment Metrics for Tracking

The performance of the tracking was assessed using multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA) metrics proposed by Bernardin [52].

MOTP is defined as

$$MOTP = \frac{\sum_{i,k} d_k^i}{\sum_k c_k}, \tag{2.28}$$

where d_k^i is the distance between the object o_i and its corresponding hypothesis. k is the frame index and c_k is the number of matched pairs in frame k. The cutoff distance for track matching was 20 pixels. MOTP is the total error in estimated position for the matched pairs (distance d) over all frames. If the distance between the estimated track and the ground truth track was beyond the cutoff distance ($c_T = 5$), it was not considered as a matching track. When there were multiple estimated tracks at a given time for a single ground truth track, the Hungarian algorithm was performed to find the most optimal pair based on the distance between the estimate and the ground truth positions [53]. The unit of MOTP is in pixels (px).

MOTA is defined as

$$MOTA = 1 - \frac{\sum_k FP_k + M_k + MME_k}{\sum_k g_k}, \tag{2.29}$$

where M_k, FP_k, and MME_k represents the number of missed sequence, false positives, and mismatches in frame k, respectively. g_k is the number of ground truth objects in frame k. MOTA is derived from three different error ratios: false positive rate $\overline{FP}$, missed detection rate $\overline{M}$, and mismatch rate $\overline{MME}$.

False positive rate $\overline{FP}$ is the total number of false alarms over the total number of objects present over all frames:

$$\overline{FP} = \frac{\sum_k FP_k}{\sum_k g_k}.$$

Missed detection rate $\overline{M}$ is the total number of misses over the total number of objects present over all frames:

$$\overline{M} = \frac{\sum_k M_k}{\sum_k g_k}.$$

Mismatch rate $\overline{MME}$ is the total number of mismatches (change in match of ground truth track to detected track) over the total number of objects present over all frames:

$$\overline{MME} = \frac{\sum_k MME_k}{\sum_k g_k}.$$

2.4 Applying the Simulation to Assess Segmentation, Localization, and Tracking

The real semen image and the simulated images were tested using five (5) different algorithms for detection (segmentation and localization). These algorithms were the following:

1. (Otsu) Binarization of image using Otsu's thresholding followed by morphological enhancements (closing, dilation and erosion) [54].
2. (Adaptive) Binarization of image using the adaptive thresholding method of Bradley [55] with the sensitivity defined to be 0.8 followed by morphological enhancements (closing, dilation and erosion).
3. (Spot-enhancement) Binarization of spot-enhanced image using Otsu's thresholding followed by morphological enhancements (closing, dilation and erosion) (method proposed in [9]).
4. (Edge-detection) Edge detection of median-filtered image using Sobel operator (modified algorithm proposed in [11]) followed by morphological enhancements (dilation, closing, and erosion).

5. (GMM) Motion detection algorithm using Gaussian mixture model (GMM) with the number of training frames set to 20, number of Gaussian modes in the mixture model set to 3, learning rate set to 0.005, and background ratio set to 0.7 [17, 56] followed by by morphological enhancements (closing, dilation and erosion).

The flowchart of five (5) different algorithms is shown in Figure 2.19. The algorithms consist of 3 stages, binarization, morphological enhancement, and localization. Figure 2.19b shows the binarization methods of the five (5) different algorithms. Figure 2.19c shows the morphological enhancement for each algorithm. If a blob (using 8-connectivity) in the morphologically enhanced image consisted of 5 or more pixels, it was considered to be a detected sperm cell.

Corresponding to each sample, 20 different simulated images were generated for testing, for a total of 40 images. In the simulated images, the non-moving cells were modeled as immotile cells. The moving cells ($N_C - N_D$) were modeled as either linear mean swimming (50%) or circular swimming (50%) cells. The algorithms were tested on real and simulated images with zero-mean Gaussian noise with variance ranging from 0 to 1225 (standard deviation of 35 in grayscale).

The OSPA distance, precision, and recall rates for five different algorithms for varying levels of noise are shown in Figures 2.20 and 2.21. Figures 2.20a and 2.21a show the OSPA distances for real images of samples 1 and sample 2, respectively. Figures 2.20d and 2.21d show the mean OSPA distances of 20 simulated images of samples 1 and sample 2, respectively. Figures 2.20b and 2.21b show the precision rates for varying levels of noise of real samples 1 and 2, respectively. Figures 2.20e and 2.21e show the mean precision rates of 20 different simulated images of sample 1 and sample 2, respectively. Figures 2.20c and 2.21c show the recall rates for varying levels of noise of real samples 1 and 2, respectively. Figures 2.20f and 2.21f show the mean recall rates of 20 different simulated images of sample 1 and sample 2, respectively.

As shown in Figures 2.20 and 2.21, the five different algorithms show similar qualitative performance on the real sample and the simulated images for each sample. In

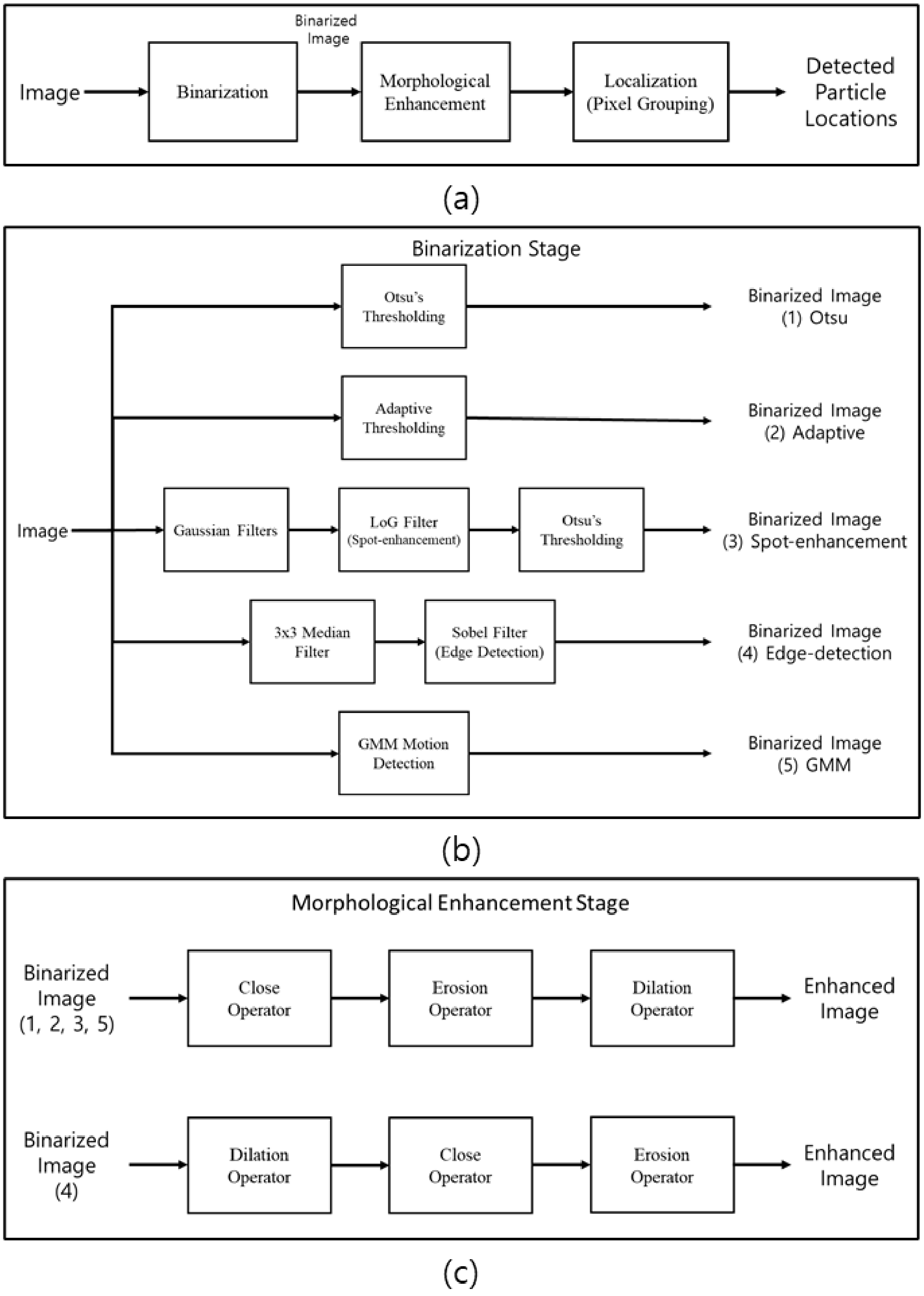


Figure 2.19 Flowcharts segmentation and localization algorithms used in the simulation study. (a) Overall segmentation and localization algorithm. (b) Binarization (segmentation) stage of (1) Otsu, (2) Adaptive, (3) Spot-enhancement, (4) Edge detection, and (5) GMM algorithms. (c) Morphological enhancement stage that precedes localization algorithm.

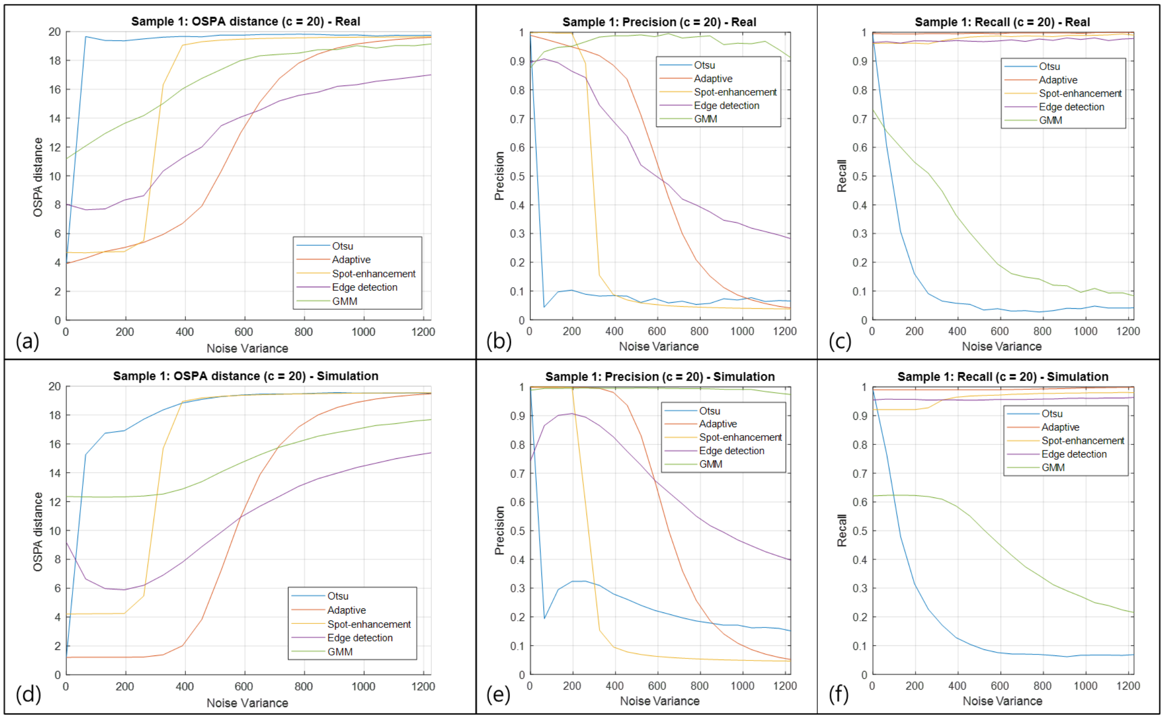


Figure 2.20 OSPA distance, precision and recall rates of sample 1 for varying levels of additive Gaussian noise ((a,b,c) real, (d,e,f) simulation).

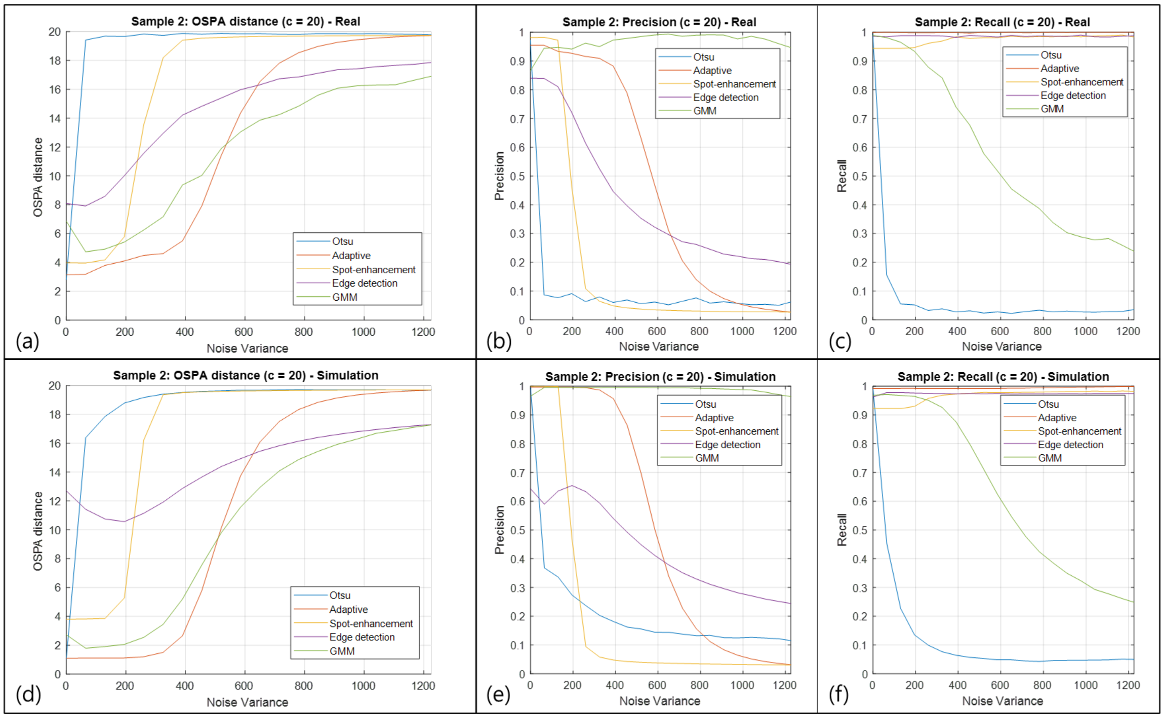


Figure 2.21 OSPA distance, precision and recall rates of sample 2 for varying levels of additive Gaussian noise ((a,b,c) real, (d,e,f) simulation).

sample 1, one can observe that the Adaptive algorithm shows the best performance in terms of OSPA (lowest values) under low noise levels (noise variance approximately below 600), followed by the Spot-enhancement algorithm, the Edge-detection algorithm, the GMM algorithm, and the Otsu algorithm, respectively (Figures 2.20a and 2.20d). In sample 2, one can observe that the Adaptive algorithm shows the best performance in terms of OSPA under low noise level (noise variance approximately below 500), followed by the GMM algorithm, the Spot-enhancement algorithm, the Edge-detection algorithm, and the Otsu algorithm, respectively (Figures 2.21a and 2.21d).

One can observe the differences in OSPA distances for the GMM algorithm on the two samples. The GMM algorithm is a motion detection algorithm. If the object is not in motion, the algorithm would not be able to detect the object. Sample 1 has three non-moving cells. As a result, a higher OSPA distance value and lower recall rate are observed for sample 1 than for sample 2 using the GMM algorithm as shown in Figures 2.20 and 2.21.

In terms of precision, a sharp loss in performance is observed for the Otsu algorithm and the Spot-enhancement algorithm when the noise level increases beyond the noise variance of 100 and 300, respectively. Precision for the Adaptive algorithm and the Edge-detection algorithm also decreases as the noise level increases, where the Edge-detection shows degradation in performance much earlier, but becomes more robust in high levels of noise (approximately above noise variance of 600). The GMM algorithm shows the highest level of robustness, having almost no degradation in performance in precision (small number of false alarms).

In terms of recall, almost no change is observed between the noise level of 0 to 1225 for the Adaptive algorithm, the Spot-enhancement algorithm, and the Edge-detection algorithm. These algorithms were less likely to have missed detections. On the other hand, the Otsu algorithm and the GMM algorithm showed degradation in performance when the noise level increased, the Otsu algorithm showing a sharp decline around noise variance of

0.01 and the GMM algorithm showing a gradual decline between noise variance value of 0 to 1225.

In conclusion, the Adaptive algorithm shows the overall best performance in localization and segmentation for low noise levels (below noise variance of 600). The Otsu algorithm showed the worst performance in the presence of noise. The study concludes that the GMM algorithm is the best performing algorithm in terms of precision (least likely to make false alarms).

Observing the precision and recall rates of real and simulated samples in Figures 2.20 and 2.21, one can also observe that the precision and recall rates of the five different algorithms show similar trends for the real sample image and its corresponding simulated images. As shown in the results, the simulation shows strong potential to be used as a testing ground for segmentation and localization algorithms in place of real semen samples that do not have ground truth labels.

2.5 Applying the Simulation to Assess Tracking

Using the simulation, the performances of the four tracking algorithms (NN, GNN, PDAF, and JPDAF) were tested. The codes for tracking algorithms were written by Urbano and are provided in [57]. The output of each tracking algorithm is the track information of all the detected cells in the semen image.

The performances of NN, GNN, PDAF, and JPDAF algorithms were evaluated for varying numbers of cells. Each simulated image consisted of 20, 40, 100, or 200 cells. Each cell in each image was assigned to a swim type using equal probabilities: linear mean swim ($\frac{1}{4}$), circular swim ($\frac{1}{4}$), hyperactive swim ($\frac{1}{4}$), and immotile ($\frac{1}{4}$). The size of the frame was 500×500 px. The framerate was set at 15 FPS and a total of 10 seconds of each image was used for tracking. 20 different scenarios were generated for each number of cells for a total of 100 images. The background intensity level was 204 (80% of 255) and no noise was added to the video images. Ground truth tracks for each image were provided by the simulation and were compared to the estimated tracks calculated by NN, GNN, PDAF, and JPDAF algorithms.

For the test cases, the parameters used for the swimming types are shown in Table 2.10. The background intensity level was 204 (80% of 255) and no noise was added to the video. The values of variables of standard deviations for the each of the point spread functions f_1, f_2, and f_3 used in track algorithm testing are the same as the values for sample 1 in Section 2.3.2. Ground truth tracks were provided by the simulation and the estimated tracks were provided by the tracking algorithms.

Table 2.10 Simulation Parameters for Tracking Assessment

Parameter	value
r_c	80 pixel
f_c	50 deg/s
f_s	4 Hz
a	3 pixel
f_l	3 Hz
r_h	12 pixel
r_v	8 pixel
A_{har}	0.1
V	50 pixel/s
$\sigma_{x_b}, \sigma_{y_b}$	10 pixel

The performance of the algorithms was assessed for varying numbers of cells. Each simulated image consisted of 20, 40, 100, or 200 cells. The cells in each image were assigned to a swim type: linear mean swim, circular swim, hyperactive swim, and immotile. The swim types were identically distributed among the cells (1:1:1:1). 20 different scenarios were generated for the different number of cells for a total of 100 images. The framerate was set at 15 FPS and a total of 10 seconds of each image was used for tracking.

Table 2.11 Tracking Performance of NN Algorithm on Varying Number of Cells

NN					
# of cells	MOTP	$\overline{FP}$	$\overline{M}$	$\overline{MME}$	MOTA
20	1.4 px	0.2101	0.0941	0.0003	0.6955
40	1.4 px	0.2935	0.1143	0.0012	0.5910
100	1.4 px	0.3593	0.1609	0.0030	0.4768
200	1.4 px	0.4140	0.2163	0.0061	0.3635

Table 2.12 Tracking Performance of GNN Algorithm on Varying Number of Cells

GNN					
# of cells	MOTP	$\overline{FP}$	$\overline{M}$	$\overline{MME}$	MOTA
20	1.4 px	0.0690	0.0812	0.0012	0.8487
40	1.4 px	0.0920	0.0886	0.0028	0.8165
100	1.4 px	0.1317	0.1090	0.0070	0.7524
200	1.5 px	0.1810	0.1343	0.0144	0.6703

Table 2.13 Tracking Performance of PDAF Algorithm on Varying Number of Cells

PDAF					
# of cells	MOTP	$\overline{FP}$	$\overline{M}$	$\overline{MME}$	MOTA
20	1.4 px	0.2076	0.0994	0.0003	0.6927
40	1.4 px	0.2934	0.1292	0.0009	0.5765
100	1.4 px	0.3636	0.1956	0.0027	0.4381
200	1.5 px	0.4448	0.2748	0.0049	0.2754

Table 2.14 Tracking Performance of JPDAF Algorithm on Varying Number of Cells

JPDAF					
# of cells	MOTP	$\overline{FP}$	$\overline{M}$	$\overline{MME}$	MOTA
20	1.4 px	0.0714	0.0817	0.0012	0.8457
40	1.4 px	0.0969	0.0907	0.0026	0.8098
100	1.4 px	0.1291	0.1099	0.0058	0.7553
200	1.5 px	0.1717	0.1352	0.0118	0.6813

The mean MOTP, $\overline{FP}$, $\overline{M}$, $\overline{MME}$, and MOTA values for 20 different scenarios for each type of images (20, 40, 100, and 200 cells) are shown in Tables 2.11, 2.12, 2.13, and 2.14 for NN, GNN, PDAF, and JPDAF, respectively. The values of multi-object tracking precision (MOTP) for the four tracking algorithms were approximately equal regardless of the number of cells in the image. As the number of cells increased, the value of multi-object tracking accuracy (MOTA) decreased and the false positive rate ($\overline{FP}$), miss detection rate ($\overline{M}$), and mismatch rate ($\overline{MME}$) increased.

For a large number of cells, GNN and JPDAF algorithms performed better than NN and PDAF tracking algorithms in terms of MOTA. The major factor for this difference was in the false positive rate ($\overline{FP}$). Comparing NN and PDAF, MOTA for PDAF was lower than NN for all of the testing cases (20, 40, 100, and 200 cells). The GNN and JDPAF tracking algorithms showed similar performance in terms of MOTA. Overall, GNN and JPDAF showed the best performance, which was followed by NN. The PDAF tracking algorithm had the worst performance.

2.6 Conclusion

This chapter presented a model of a 2-D (top-down) view of the sperm cell and the model of 4 different swimming modes generated by observing the swimming paths of real human sperm cells. The simulation opens up various opportunities to study and test different sperm processing algorithms, including segmentation, localization, and tracking algorithms. In the examples, five different segmentation and localization algorithms were tested. The simulation was used to rank the algorithms by their performance and the obtained ranking with simulation was similar to the ranking using real images. In addition, the performance of four different tracking algorithms (NN, GNN, PDAF, and JPDAF) were tested on the simulated images using MOT metrics and were ranked by their performance. The simulation models and the software presented in this chapter serve as a powerful new tool for developing and enhancing CASA systems and algorithms. Using this new tool, stronger and more robust CASA systems can be developed. The use of such systems can remove the necessity of manual semen assessments, which are time-consuming, labor intensive, and subjective.

CHAPTER 3
DIGITAL WASHING OF UNWASHED HUMAN SEMEN SAMPLE

In this chapter, a digital washing algorithm for processing of unwashed semen samples for the detection of sperm cells is given.[1]*It offers an alternative to chemical washing of semen samples, the currently used procedure. Digital washing is quicker, less expensive, and more environmentally friendly than chemical washing. Also, unlike chemical washing, it does not wash away many sperm cells (ideally it does not wash away any). To assess the new digital washing algorithm, we compared its performance (segmentation and localization) on the same unwashed samples to the performance of several other segmentation and localization algorithms.*

3.1 Background

Semen analysis may be performed on unwashed or washed semen samples. In unwashed semen samples, non-sperm cells can be present in the fluid. Some examples of non-sperm cells are immature germ cells, leukocytes (also known as white blood cells, WBC), and epithelial cells. The concentration of the non-sperm cells is normally less than 15% of the sperm concentration [58]. In specific cases, such as pyospermia, one can have a high concentration of non-sperm cells that may affect male fertility. In addition, immature sperm cells can also be found in semen, and possible causes for their presence range from infection to imbalance of hormones [59].

To get rid of these debris (non-sperm cells) before performing sperm analysis, the semen samples can go through a process of washing. On chemically washed samples that are free from debris and extraneous objects, Computer-Assisted Semen Analysis (CASA) systems have shown to perform very well [5]. However, analyzing unwashed samples

[1] This work is a collaboratory work between Ji Won Choi (the author of dissertation) and Ludvik Alkhoury. Both are PhD candidates in NJIT's ECE department under the advisement of Dr. Moshe Kam.

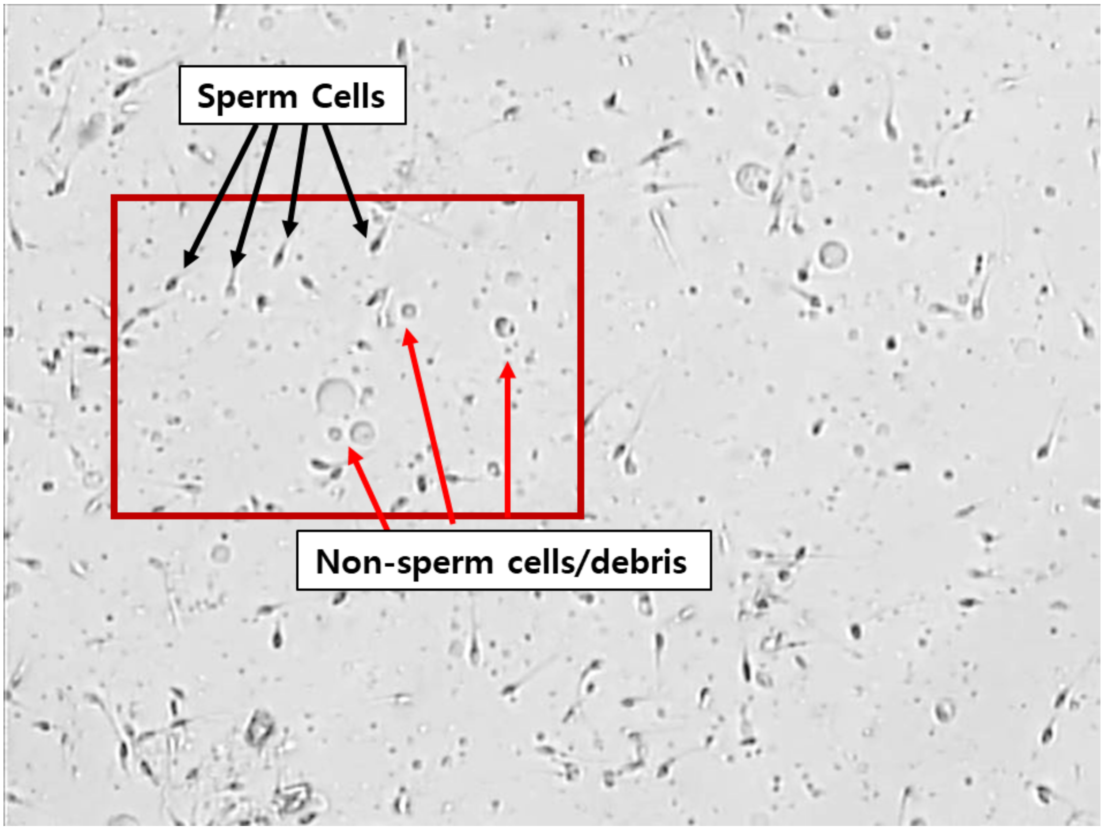


Figure 3.1 An image of an unwashed semen sample that contains sperm and non-sperm cells.

is problematic. The presence of non-sperm cells or immature sperm cells can cause difficulties in assessing the movements of sperm cells. An image of an unwashed semen sample with non-sperm cells is shown in Figure 3.1 as an example where some sperm cells are marked in black arrows and the non-sperm cells/debris are marked with red arrows. Clearly, for computer systems, the task of segmenting, localizing, and classifying sperm cells from the unwashed sample would be a challenge.

In this chapter, a digital washing algorithm is presented, which takes the image of an unwashed semen sample and ”washes” the sample. As a result, After washing, one can obtain the location of sperm cells with significantly fewer false alarms (detecting non-sperm cells as sperm cells).

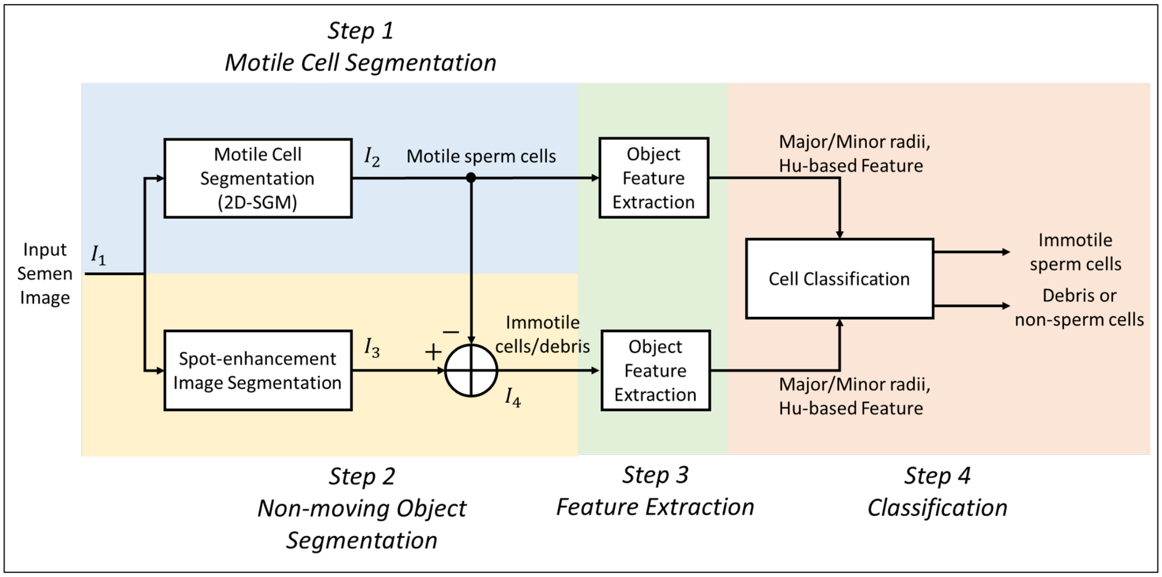

Step 1
Motile Cell Segmentation
Motile Cell Segmentation (2D-SGM)
I_2
Motile sperm cells
Object Feature Extraction
Major/Minor radii, Hu-based Feature
Input Semen Image
I_1
Cell Classification
Immotile sperm cells
Debris or non-sperm cells
−
Immotile cells/debris
Spot-enhancement Image Segmentation
I_3
+
Object Feature Extraction
I_4
Major/Minor radii, Hu-based Feature
Step 2
Non-moving Object Segmentation
Step 3
Feature Extraction
Step 4
Classification

Figure 3.2 Flowchart of digital washing algorithm. (Blue) Step 1: Motile Cell Segmentation (2D-SGM). (Yellow) Step 2: Non-moving Object Segmentation. (Green) Step 3: Object Feature Extraction. (Red) Step 4: Classification.

3.2 Methods

Figure 3.2 shows the overall flowchart of the digital washing algorithm. The algorithm performs of following steps:

- Step 1 - Motile Cell Segmentation (2D-SGM),
- Step 2 - Non-moving Object Segmentation (immotile cells or debris),
- Step 3 - Feature Extraction, and
- Step 4 - Classification of Immotile Cells.

The input to the system is the semen image. The input image is used to detect (1) motile cells and (2) non-moving objects (Steps 1 and 2). The motile cells are segmented using an algorithm named 2D-SGM algorithm (modified version of the Gaussian mixture model (GMM) algorithm) [56]. The features from the motile cells and the non-moving objects are calculated (Step 3). The features that are calculated from each object are radii length of the object along the major and minor axes [43] and a feature calculated from the Hu moments [60]. The calculated features from the motile cells are then used to identify non-moving sperm cells among the non-moving objects in the image (Step

4). In Sections 3.2.1-3.2.4, each step is described in detail: (1) motile cell segmentation, (2) non-moving object segmentation, (3) feature extraction, and (4) classification.

3.2.1 Step 1: Motile Sperm Cell Detection

In step 1 of digital washing algorithm, the input is the semen image and the outputs are (1) segmented regions of moving sperm cells and (2) locations of detected moving sperm cells. The digital washing algorithm uses a motion detection algorithm to find the moving objects in the semen image sample. The algorithm assumes that the sperm cells are the only active moving objects in the image. By detecting the moving objects, the algorithm will be able to detect the locations of sperm cells with high precision. In the motion detection algorithm, the foreground refers to the segmented regions of moving objects and the background refers to the approximately static image environment (non-moving).

3.2.1.1 Gaussian Mixture Model. The developed motion detection algorithm is based on the idea of the Gaussian mixture model (GMM) proposed by Stauffer [56]. The GMM algorithm models the values of each pixel instead of modeling all the pixels together. The recent history of each pixel $\{X(0), X(T), ..., X(t-T), X(t)\}$ is modeled as a mixture of K Gaussians:

$$P(X(t)) = \sum_{i=1}^{K} \omega(i,t) * \eta(X(t), \mu(i,t), \Sigma(i,t)) \tag{3.1}$$

where K : the number of distributions,

$\omega_i(t)$: an estimate of the weight of i^{th} Gaussian in the mixture at time t,

$\mu_i(t)$: the mean value of i^{th} Gaussian in the mixture at time t,

$\Sigma_i(t)$: the covariance matrix of the i^{th} Gaussian in the mixture at time t (assume $\Sigma_i(t) = \sigma_i^2(t)\mathbf{I}$ (approximation)),

$\sigma_i^2(t)$: the variance of the pixel value of the i^{th} Gaussian in the mixture at time t,

$\mathbf{I}$: identity matrix of size $D \times D$, and

T : period between each frame (sec).

$\eta(X(t), \mu, \Sigma)$ is a Gaussian probability density function given as:

$$\eta(X(t), \mu, \Sigma) = \frac{1}{(2\pi^{D/2}|\Sigma|^{1/2})} e^{\frac{1}{2}(X(t)-\mu)^T \Sigma^{-1}(X(t)-\mu)} \tag{3.2}$$

and D is the dimension of pixel value $X(t)$ ($D = 3$ if using red, green, and blue (RGB)). Typical value of K is between 3 and 5. The pixel values in each dimension are assumed to be independent and have equal variance ($\Sigma_i(t) = \sigma_i^2 \mathbf{I}$). A large weight $\omega_i(t)$ infers that the Gaussian distribution belongs to the background, a small weight infers that the Gaussian distribution belongs to the foreground (moving object).

With each new frame of an image, a new pixel value is checked to see if the value is within 2.5 standard deviations away from the mean of each Gaussian. If a match is found, the prior weights of each Gaussian distributions at time t are updated as follows:

$$\omega_i(t) = (1 - \alpha)\omega_i(t - T) + \alpha(M_i(t)) \tag{3.3}$$

where α is the learning rate and $M_i(t)$ is an one-hot encoded variable (equal to 1 for the matched distribution and 0 for other distributions). Additionally, the mean and variances of the matching distribution are updated as follows:

$$\mu_i(t) = (1 - \rho_i)\mu_i(t - T) + \rho_i X(t), \text{and} \tag{3.4}$$

$$\sigma_i^2(t) = (1 - \rho_i)\sigma_i^2(t - 1) + \rho_i(X(t) - \mu_i(t))^T(X(t) - \mu_i(t)), \tag{3.5}$$

where

$$\rho_i = \alpha\eta(X(t)|\mu_i, \sigma_i). \tag{3.6}$$

If the new pixel does not belong to any distribution, then the Gaussian distribution with the smallest weight is replaced. The new Gaussian distribution has the mean μ equal to the current value of the pixel and the variance set to a user specified value $\sigma^2_{Init_{GMM}}$.

Afterwards, the Gaussians are ordered by the value of the ratio between the weight ω_i and the standard deviation σ_i (ω_i/σ_i) in descending order. The first B distributions are chosen as the background model, where B is defined as

$$B = argmin_b \left(\sum_{i=1}^{b} \omega_i > \omega_T \right). \tag{3.7}$$

ω_T is the threshold for the minimum background ratio. If the value of ω_T is very small, then the Gaussian with the highest ratio (ω_i/σ_i) is the only Gaussian distribution that will

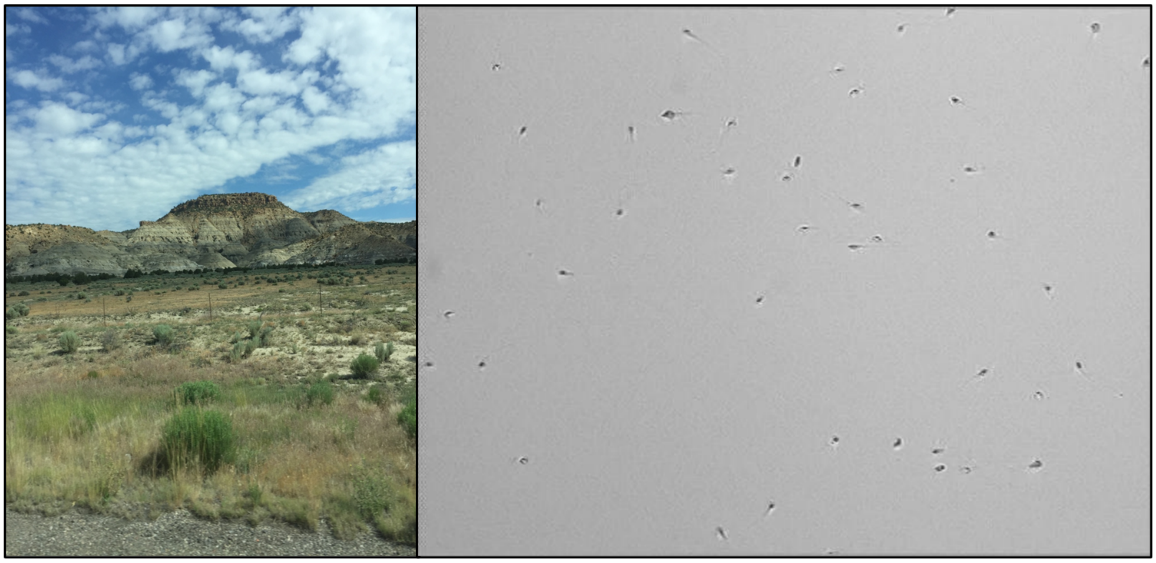


Figure 3.3 (Left) A scenery image taken by a standard camera. (Right) A microscopy image of human semen sample.

represent the background (assumes that the background model is unimodal). All the other distributions will be considered a foreground. If ω_T is large, more distributions are included in the background model.

3.2.1.2 Motion Detection using Single Gaussian Model (2D-SGM). The use of multiple Gaussian distributions for background modeling in images can deal with changing or moving backgrounds such as lighting changes, moving clouds, or swaying trees. This characteristic makes the GMM algorithm very powerful. However, the GMM algorithm was not designed for detecting motion in microscopy images. The GMM algorithm has therefore been modified by us to account for the differences in standard surveillance images and microscopy images.

Figure 3.3 shows two images. On the left, an image of a scene is taken, where many complex objects are present. This image is an example in which GMM algorithm was intended to be used for motion detection. On the right, an microscopy image of a human semen sample is shown. Unlike the image on the left of Figure 3.3, the background of the microscopy image is simple. In standard images, many different sources of background exist (moving clouds, moving grass/trees, changes in lighting). However, in microscopy images, only one source of background exists. In the modified algorithm, the expected number of Gaussians in the GMM model is reduced to one. This simplifies the GMM algorithm to be a t-test. The modified motion detection algorithm proposed here is shown in Figure 3.4. The motion detection algorithm has two main components denoted as operation A (segmentation) and B (localization).

The input of the motion detection algorithm is a semen image. The output of the algorithm is the segmented regions and locations of moving objects in the semen image. In operation (A), segmentation is performed (produces image I_B). In operation (A-1), a median filtered image is generated (using 3×3 median filter [43]). In 3×3 median filtering, the value of the pixel is replaced with the median value within 3 by 3 window

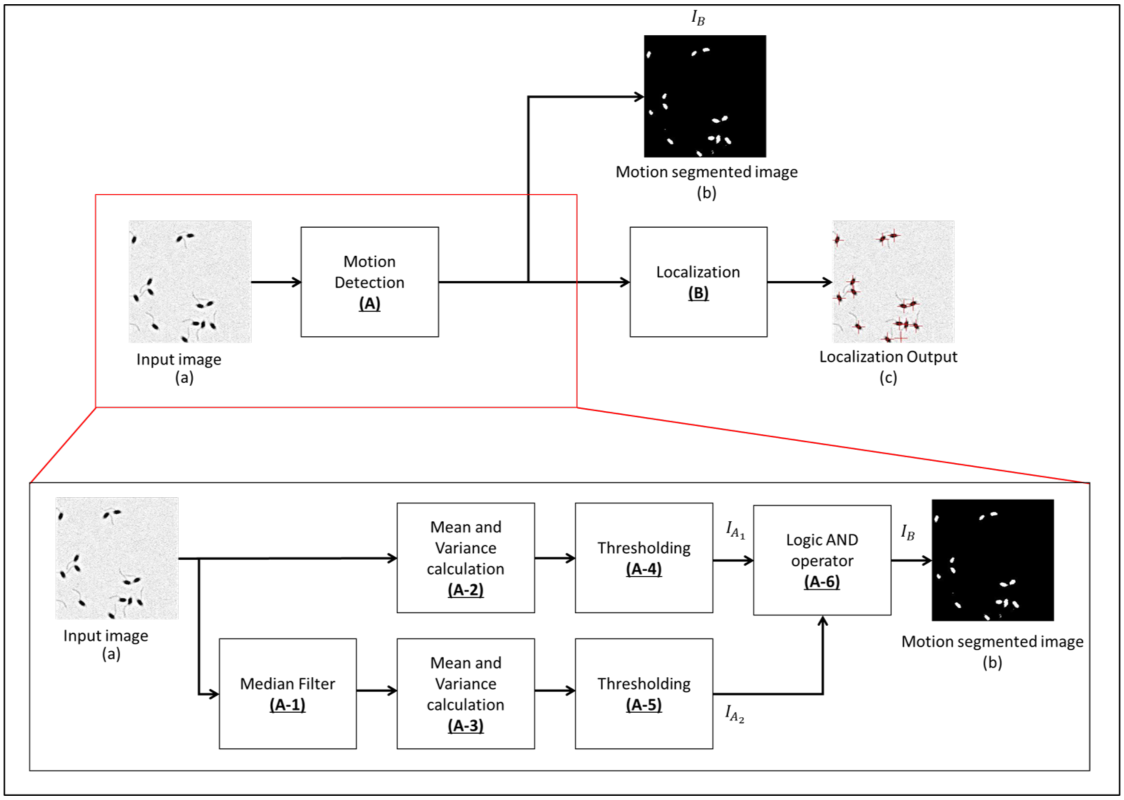
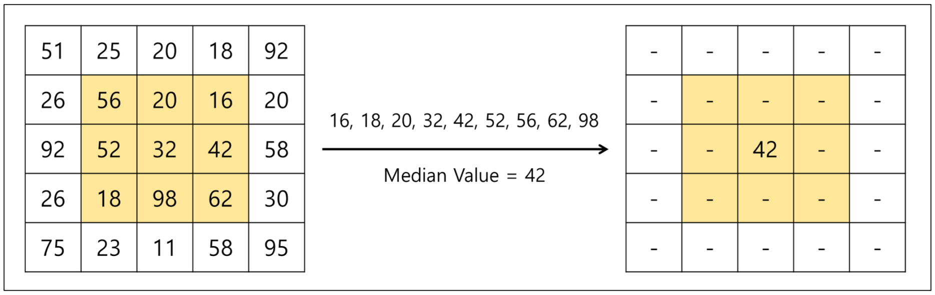


Figure 3.4 Motion detection algorithm (2D-SGM) for segmenting and localizing motile sperm cells.

around the pixel. An example of median filtering is shown in Figure 3.5. On the left, there are nine values within the shaded 3 by 3 region. When ordered from lowest to the highest value, the values are: 16, 18 ,20, 32, 42, 52, 56, 62, 98. The median value of this set is 42. The center of the shaded region is replaced with the median value, 42.



51	25	20	18	92
26	56	20	16	20
92	52	32	42	58
26	18	98	62	30
75	23	11	58	95

-	-	-	-	-
-	-	-	-	-
-	-	42	-	-
-	-	-	-	-
-	-	-	-	-

Figure 3.5 Example of 3 × 3 median filtering.

In operations (A-2) and (A-3), the mean and the variance of 20 past grayscale values of each pixel are calculated from the original image and the median filtered image.

Calculating the mean and variance for every new frame are computationally costly (if the image is 480 by 480 pixels, then one needs to calculate the mean and variance for a total of 230,400 pixels every frame). To reduce the computational load, Welford's online algorithm has been implemented to update the mean and variance [61].

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n} \tag{3.8}$$

$$s_n^2 = s_{n-1}^2 + \frac{(x_n - \bar{x}_{n-1})^2}{n} - \frac{x_{n-1}^2}{n-1} \tag{3.9}$$

x_n is the n^{th} value of the pixel. $\bar{x}_n$ is the mean of n values of the pixel. s_n^2 is the variance of n values of the pixel. When a new value of a pixel is introduced, the oldest value is removed and the new value is added to calculate the new mean and variance by using Equations (3.8) and (3.9) (maintaining data length of 20).

In operations (A-4) and (A-5), the current value of the pixel is compared to the means and variances calculated in operations (A-2) and (A-3). If the new value of the pixel is more than C standard deviations away from the mean, then the pixel is considered to be part of the foreground (moving object). If the pixel is considered to be a part of the foreground, it is 1, and if the pixel is considered to be a part of the background, it is 0. The performance of the algorithm with a small value of C would be more sensitive to finding values away from the mean. The performance of the algorithm with a large value of C would be less sensitive. For example, $C = \infty$ will result in all pixels to be part of the background and $C = 0$ will result in all pixels that are not equal to the mean to be the foreground. The foreground image of the original image is notated as I_{A_1}, and the foreground image of the median filtered image is notated as I_{A_2}. In operation (A-6), an AND operator is used on I_{A_1} and I_{A_2} to generate the output of the motion segmented image $I_B = I_{A_1} I_{A_2}$. AND operator is a logic operator that produces the output of 1 only if both of the inputs are 1. Otherwise, the output is 0. The use of the AND operator on I_{A_1} and I_{A_2} lower the amount of noise and false detection.

In operation (B), the segmented foreground image is enhanced using dilation and erosion morphological operators with a disk structuring element with a radius of 7 [43] and the locations of the moving cells/objects are labeled. The groups of pixels are labeled assuming eight-connectivity (if any surrounding pixel has the value 1, the pixels are in the same group). Any group that contained 5 or more pixels was considered to be a sperm cell. The motion detected image is denoted as I_2 in Figure 3.2. This motion detection algorithm is denoted as 2D-SGM (2-Dimensional Single Gaussian Model).

To demonstrate the effectiveness of the 2D-SGM algorithm in detecting moving sperm cells, the 2D-SGM algorithm was tested for three real human semen images. These three samples were samples ID005, ID006, and ID007 (description of image collection process is given in Section 2.3.1). The frame rate for samples ID005 and ID006 was 15 FPS and the frame rate for sample ID007 was 30 FPS. A total of six scenes was taken for testing. Three (3) scenes were from 15 FPS samples (two from ID005 and one from ID006) and the other three (3) scenes were from 30 FPS sample (ID007). Each scene was 200 frames long and all of the samples used in this example were washed samples (no debris or non-sperm cells).

A total of six (6) different algorithms were tested for their ability to detect and localize sperm cells. The six algorithms were: (1) 2D-SGM, (2) Otsu's thresholding [54], (3) Adapative algorithm by Bradley [55], (4) Urbano's segmentation algorithm [9], (5) Abbiramy's segmentation algorithm [11], and (6) Gaussian Mixture Model (GMM) [56] detection algorithm. The performance of detection was assessed in terms of precision and recall. The ground truth locations of sperm cells in each sample was labeled by human.

Precision is defined as $TP/(TP + FP)$ and recall is defined as $TP/(TP + FN)$, where TP, FP, and FN refer to true positive, false positive, and false negative, respectively.

True positive is defined to be the number of matches in detection and ground truth within 10 pixels, false positive is defined to be the number of false alarms, and false negative is defined to be the number of missed detections (the same definition used in Chapter 2). The 2D-SGM algorithm was designed to detect moving cells with high precision (small number of false alarms). Recall rates of 2D-SGM are expected to be lower than the other cell segmentation and localization algorithms due to the 2D-SGM algorithm's inability to detect non-moving sperm cells.

The performance of the 2D-SGM algorithm for varying values of C for three samples (ID005, ID006, and ID007) are shown in Figures 3.6-3.11. Samples contained different numbers of immotile sperm cells. For example, sample ID005 contained immotile cells and Sample ID007 contained very few immotile cells. Top of Figures 3.6-3.11 shows scatter plot of precision and recall values of motion detection algorithm for varying values of C. The bottom left shows the curve for precision and C (notated as std threshold) and the bottom right shows the curve for recall and C. For the GMM algorithm, different operating points for varying values of the threshold ω_T, minimum background ratio, are shown. The values of ω_T were between 0 to 1. The GMM algorithm was applied using the vision.ForegroundDetector function in MATLAB. The learning rate was 0.005, the number of Gaussians was 5, the number of training frames was set to 20. For the adaptive algorithm, adaptthresh function in MATLAB was used with the sensitivity value of 0.8.

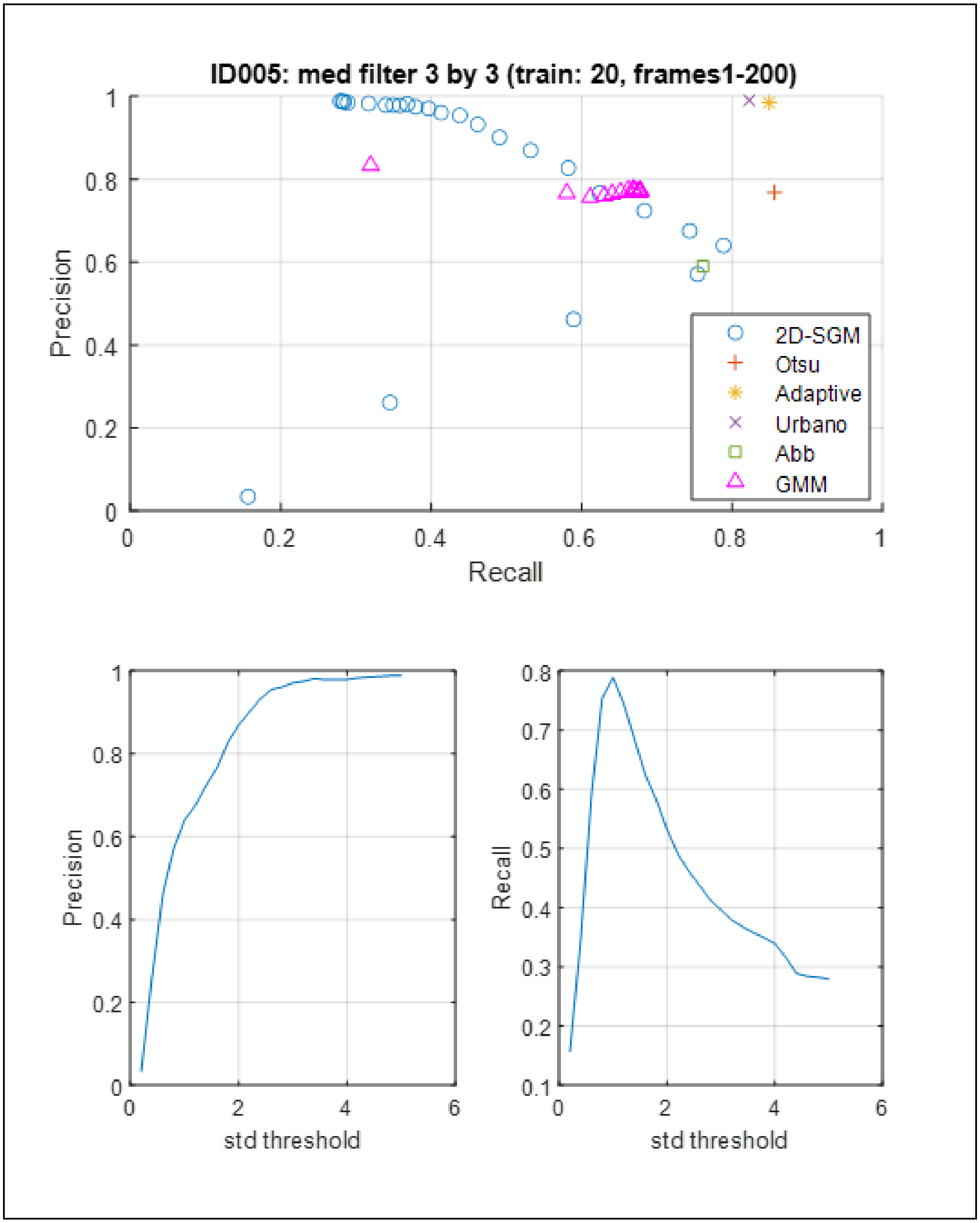


Figure 3.6 (Top) Precision and recall plot for the six different segmentation algorithms for sample ID005 (Scene 1). (Bottom left) Precision and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2. (Bottom right) Recall and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2.

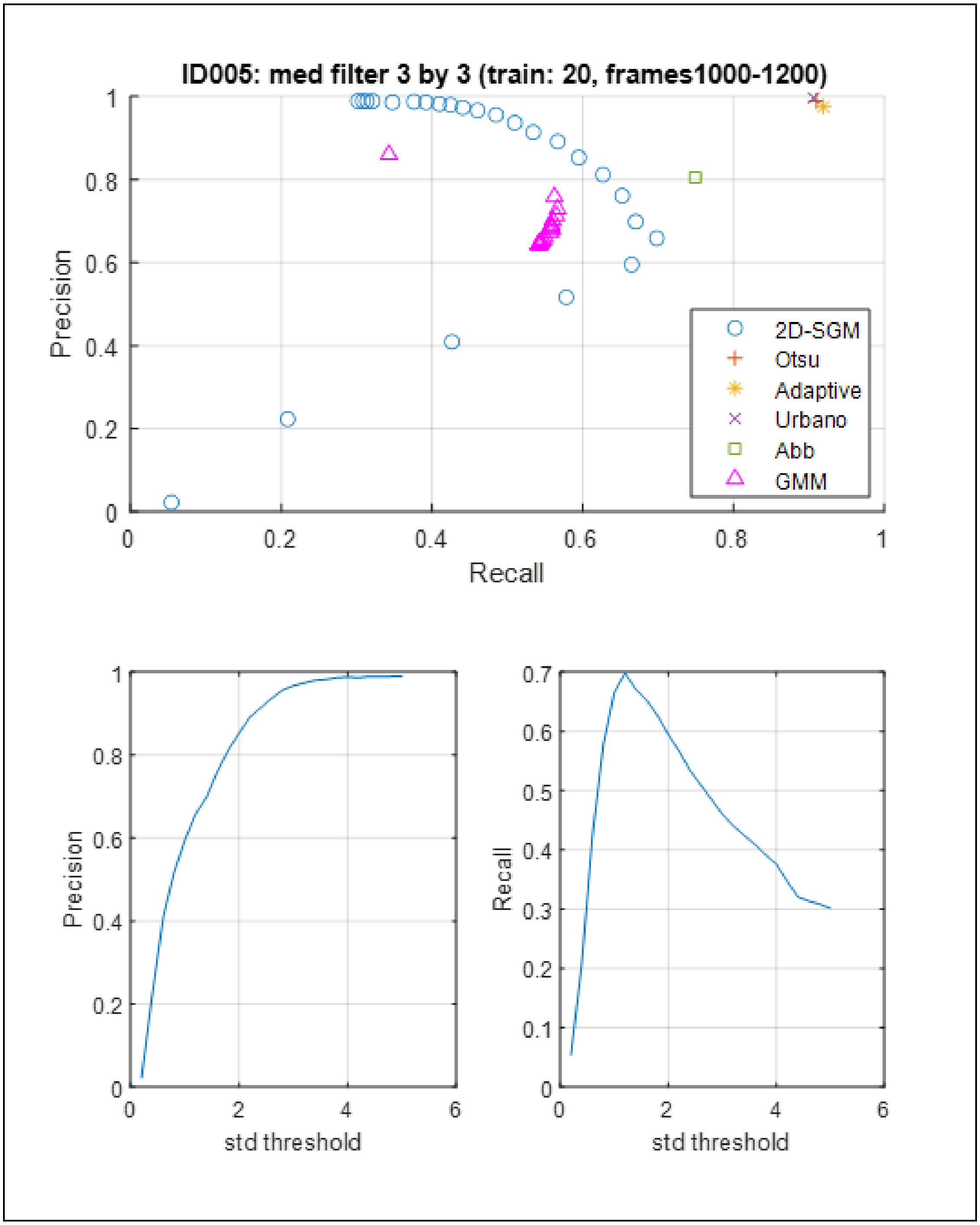


Figure 3.7 (Top) Precision and recall plot for the six different segmentation algorithms for sample ID005 (Scene 2). (Bottom left) Precision and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2. (Bottom right) Recall and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2.

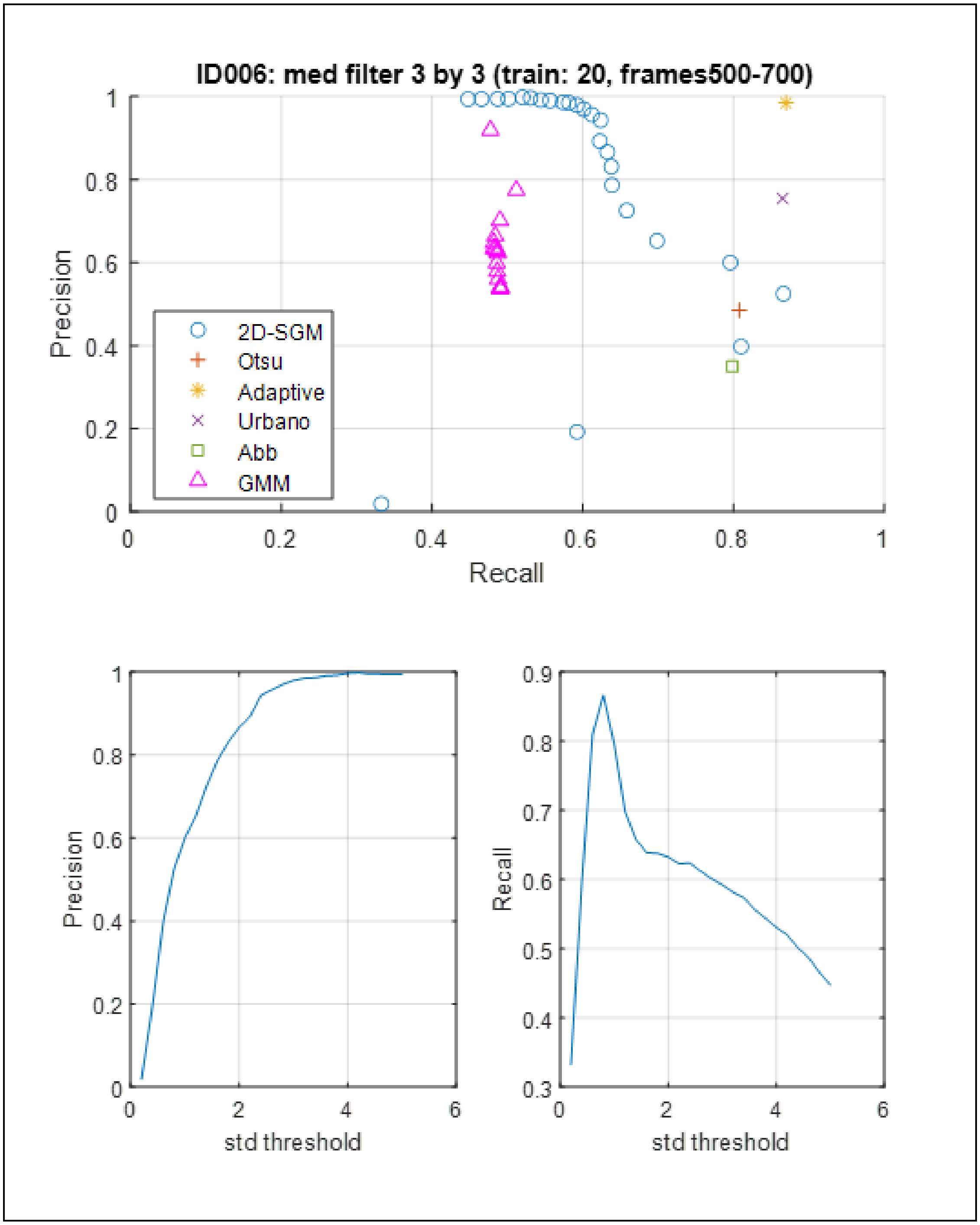


Figure 3.8 (Top) Precision and recall plot for the six different segmentation algorithms for sample ID006 (Scene 1). (Bottom left) Precision and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2. (Bottom right) Recall and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2.

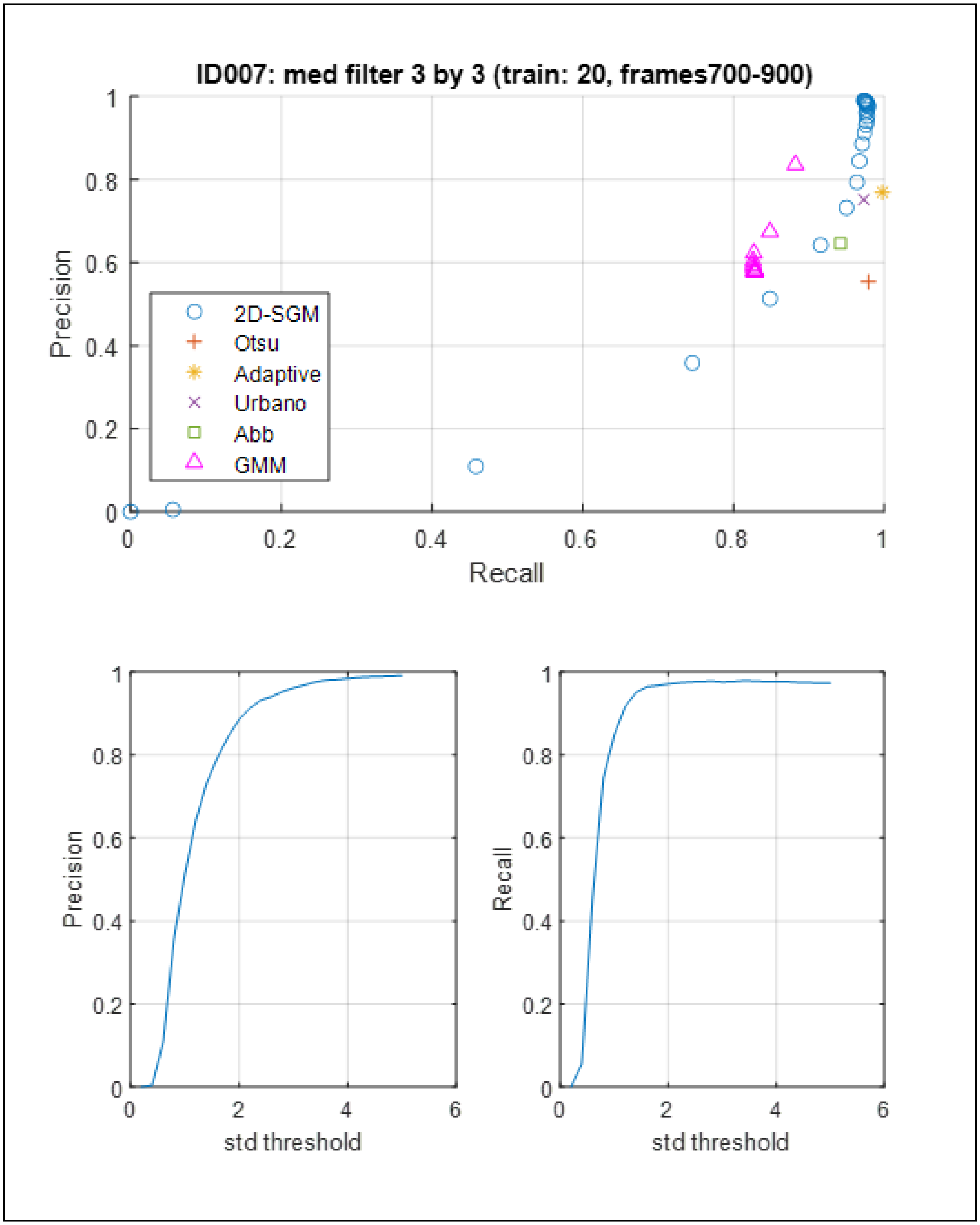


Figure 3.9 (Top) Precision and recall plot for the six different segmentation algorithms for sample ID007 (Scene 1). (Bottom left) Precision and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2. (Bottom right) Recall and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2.

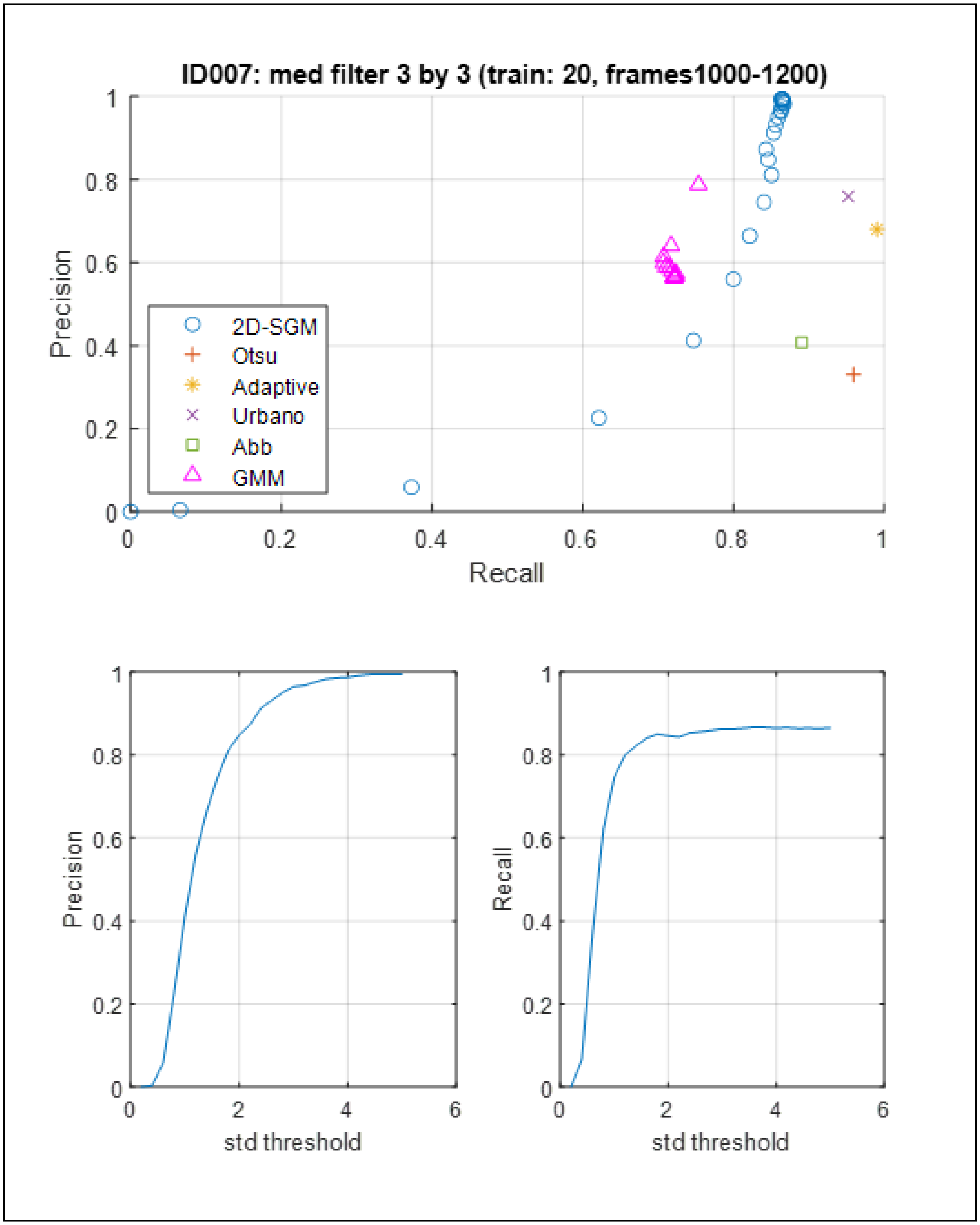


Figure 3.10 (Top) Precision and recall plot for the six different segmentation algorithms for sample ID007 (Scene 2). (Bottom left) Precision and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2. (Bottom right) Recall and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2.

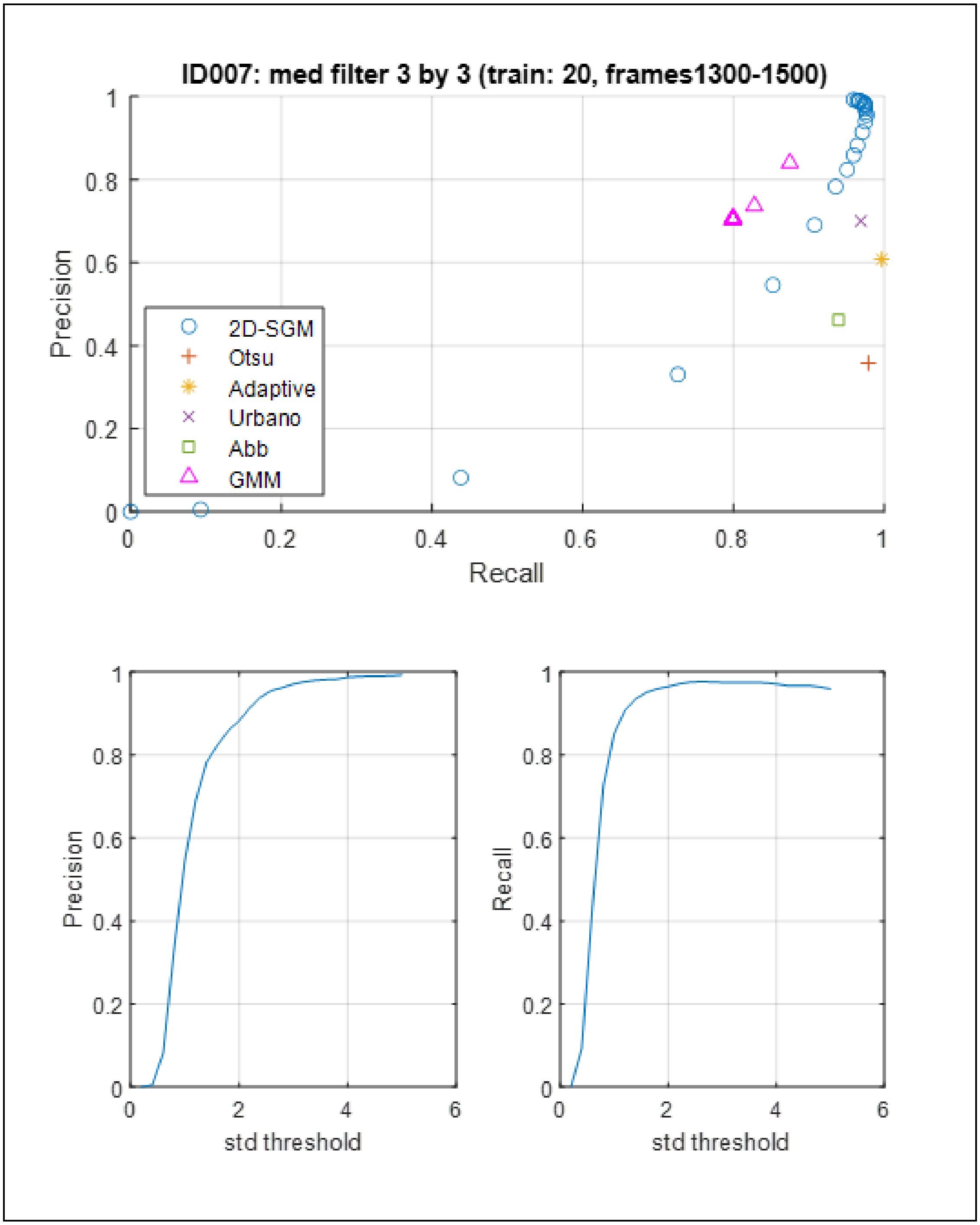


Figure 3.11 (Top) Precision and recall plot for the six different segmentation algorithms for sample ID007 (Scene 3). (Bottom left) Precision and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2. (Bottom right) Recall and standard deviation threshold C curve for motion detection algorithm in Section 3.2.1.2.

Table 3.1 Precision and Recall Rates of the Six Different Segmentation Algorithms for Six Different Scenarios (Real Human Samples)

ID005:1-200			ID005:1000-1200		
Algorithm Name	Precision	Recall	Algorithm Name	Precision	Recall
2D-SGM	0.9709	0.3963	2D-SGM	0.9667	0.4604
Otsu	0.7684	0.8563	Otsu	0.9899	0.9101
Adaptive	0.9849	0.8494	Adaptive	0.9754	0.9194
Urbano	0.9903	0.8229	Urbano	0.9964	0.9064
Abbiramy	0.5902	0.7613	Abbiramy	0.8056	0.7492
GMM	0.7692	0.6517	GMM	0.6911	0.5582
ID006:500-700			ID007:700-900		
Algorithm Name	Precision	Recall	Algorithm Name	Precision	Recall
2D-SGM	0.9794	0.5923	2D-SGM	0.9611	0.9759
Otsu	0.4853	0.8074	Otsu	0.5538	0.9790
Adaptive	0.9849	0.8697	Adaptive	0.7696	0.9977
Urbano	0.7544	0.8652	Urbano	0.7518	0.9728
Abbiramy	0.3500	0.7985	Abbiramy	0.6461	0.9418
GMM	0.6343	0.4819	GMM	0.5872	0.8275
ID007:1000-1200			ID007:1300-1500		
Algorithm Name	Precision	Recall	Algorithm Name	Precision	Recall
2D-SGM	0.9641	0.8630	2D-SGM	0.9707	0.9746
Otsu	0.3302	0.9593	Otsu	0.3585	0.9789
Adaptive	0.6802	0.9906	Adaptive	0.6079	0.9967
Urbano	0.7596	0.9524	Urbano	0.7002	0.9691
Abbiramy	0.4071	0.8905	Abbiramy	0.4623	0.9395
GMM	0.5875	0.7151	GMM	0.7048	0.7993

The precision and recall rates for the six different scenarios are shown in Table 3.1. In this table, $C = 3$ and $\omega_T = 0.7$. The best performing values for each scenario were underlined. In most of the testing cases, the adaptive algorithm and Urbano's algorithm showed the best recall rates. In the presence of immotile cells (ID005 and ID006), the 2D-SGM algorithm and the GMM algorithm contained missed detections since these algorithms are based on detecting motion (low recall rates). However, the 2D-SGM algorithm has shown high precision rates in all of the six cases (precision rate above 0.96). Unlike the other segmentation algorithms, the 2D-SGM algorithm is less likely to make false alarms due to noise. In all of the cases, one can also observe that the proposed 2D-SGM algorithm performs better than GMM algorithm in terms of precision. By using the 2D-SGM algorithm, one is able to extract the images of motile cells with low false alarms (non-sperm cells). In the following steps of the digital washing algorithm, the image of the motile cell will be used to identify the sperm cells from the non-moving cells/debris in the image (resulting in an increase in recall rate, or a lower number of missed detections).

3.2.2 Step 2: Non-moving Object Segmentation using Spot-enhancement

In step 2 of the digital washing algorithm, the input is the semen image and the output is the segmented regions of objects (cells/debris) in the semen image. For step 2, the digital washing algorithm uses Urbano's segmentation algorithm [9]. After segmenting out all the object in the image, the regions of motile cells is subtracted to obtain the regions of non-moving objects in the image.

The Urbano's segmentation algorithm consists of the following stages: (1) noise reduction, (2) spot-enhancement, (3) intensity thresholding/binarization, and (4) morphological enhancement. The flowchart of the segmentation algorithm is shown in Figure 3.12.

3.2.2.1 Noise Reduction. The noise present in the original image is removed by using a series of Gaussian filters. The filter is applied by using 2D convolution on the image $f(x,y)$ and the filter $h(x,y)$:

$$g(x,y) = f(x,y) \otimes h(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(x,y)h(x-m,y-n), \tag{3.10}$$

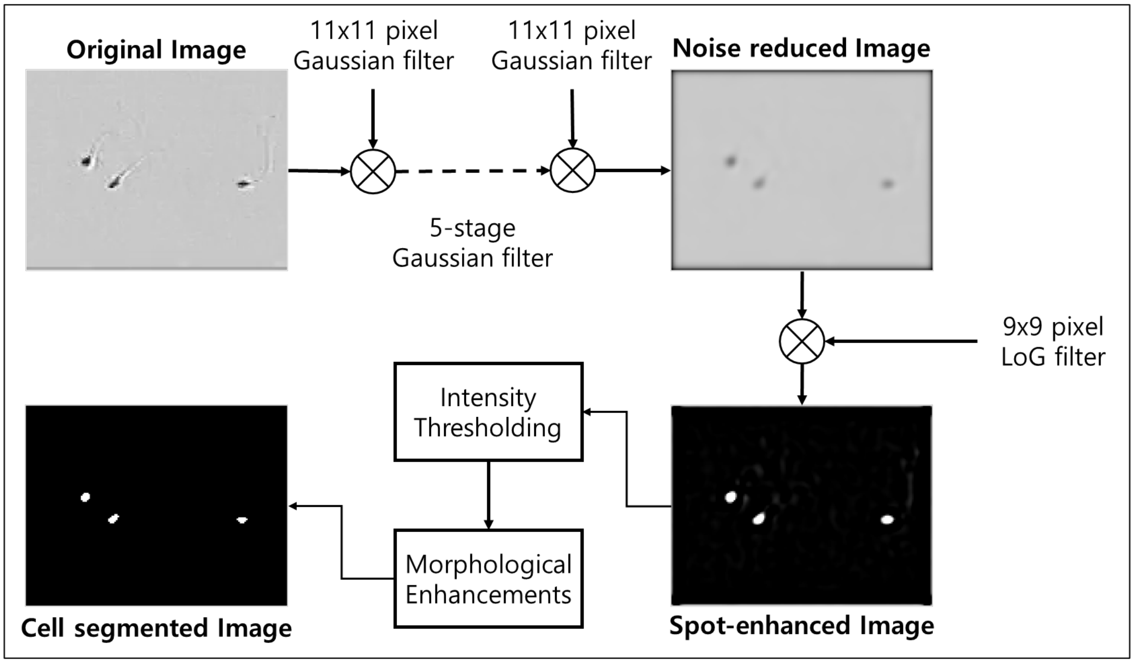


Figure 3.12 Cell segmentation algorithm by Urbano. The original image is filtered with series of five Gaussian filters to reduce the noise present in the image. The nosie reduced image is then filtered using Laplacian-of-Gaussian (LoG) filter to generate the spot-enhanced image. The spot-enhanced image is then thresholded using the Otsu's method and morphologically enhanced. The output of the system is the cell segmented image.

where $g(x, y)$ is the filtered image. The Gaussian filter $h_G(x, y)$ is defined as

$$h_G(x, y) = \frac{1}{2\pi\sigma_G^2} e^{-\frac{x^2+y^2}{2\sigma_G^2}} \tag{3.11}$$

in 2D, where σ_G is the standard deviation of the Gaussian. The original image is filtered 5 times in sequence using the Gaussian filter as shown in Figure 3.12. The value of the standard deviation used in the study was $\sigma_G = 1$.

3.2.2.2 Spot-enhancement. The noise reduced image was filtered using the Laplacian of Gaussian (LoG) filter. The Laplacian of Gaussian filter $h_L(x, y)$ is defined as

$$h_L(x, y) = -\frac{1}{\pi\sigma_L^4} \left[1 - \frac{x^2 + y^2}{2\sigma_L^2}\right] e^{-\frac{x^2+y^2}{2\sigma_L^2}} \tag{3.12}$$

where σ_L is the standard deviation of the LoG. The LoG filter highlights the differences between the background, cell membrane, and the sperm head. As shown in the bottom right of Figure 3.12, the location of the cell heads contains large values, where the background pixels show small values. The value of the standard deviation used in the study was $\sigma_L = 0.9$.

3.2.2.3 Intensity Thresholding. The spot-enhanced image was binarized using the threshold calculated using the Otsu's thresholding algorithm [54]. The Otsu's algorithm is an algorithm that searches for the threshold that minimizes intra-class variance σ_ω^2, which is the weighted sum of the variances of the two classes (background and foreground):

$$\sigma_\omega^2(k) = \omega_0(k)\sigma_0(k) + \omega_1(k)\sigma_1^2(k). \tag{3.13}$$

Here, k is the value of the threshold (boundary that determines the two classes). The group of pixels that contains grayscale values under or equal to k is class 0 and the group of pixels that contains grayscale values above k is class 1. The weights of the two classes are ω_0 and ω_1. The weights are the probabilities of class occurrence,

$$\omega_0(k) = \text{Pr}(C_0) = \sum_{i=1}^{k} p_i = \omega(k), \quad \text{(class 0)}, and \tag{3.14}$$

$$\omega_1(k) = \text{Pr}(C_1) = \sum_{i=k+1}^{L} p_i = 1 - \omega(k), \quad \text{(class 1)}, \tag{3.15}$$

where p_i is defined as the ratio of the number of pixels with i^{th} grayscale value to the total number of pixels in the frame (grayscale value in this case are assumed to be between 0 to 255 for a total of 256 grayscale levels)

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=1}^{L} p_i = 1. \tag{3.16}$$

Here, n_i is the number of pixels with i^{th} grayscale value, and N is the total number of pixels in the frame; the total number of grayscale levels is L ($L = 256$); the variances of the two classes are σ_0^2 and σ_1^2. In Urbano's algorithm, this threshold k is multiplied by a constant, where value is 1.1 ($k_{new} = 1.1k$). This study follows the same steps.

3.2.2.4 Morphological Enhancement. The regions that were segmented by thresholding were enhanced by morphological enhancement techniques. This enhancement includes erosion and dilation with 5×5 and 3×3 diamond structuring element, respectively [43]. As a result, the segmented image of cells in the image is obtained (example shown in the bottom left of Figure 3.12). The resulting image is I_3 marked in Figure 3.2.

3.2.2.5 Region Subtraction for Segmentation of Immotile Cells/debris. In step 3 of the digital washing algorithm, the image of segmented regions of moving sperm cells I_2 and the image of segmented regions of cells/debris in a semen sample I_3 are the inputs and the outputs are (1) the segmented regions of non-moving cells/debris in a semen image I_4 and (2) locations of non-moving cells/debris.

Figure 3.13 shows each steps of immotile cell or debris segmentation. Figure 3.13a is the image of semen sample. From the semen sample, the regions of motile cell I_2 and regions of all possible cells and debris I_3 are obtained (Figure 3.13a and Figure 3.13b, respectively). By subtracting I_2 from I_3, the regions of immotile cells and debris I_4 is found (Figure 3.13d). Figure 3.13e shows the original image with the detected motile

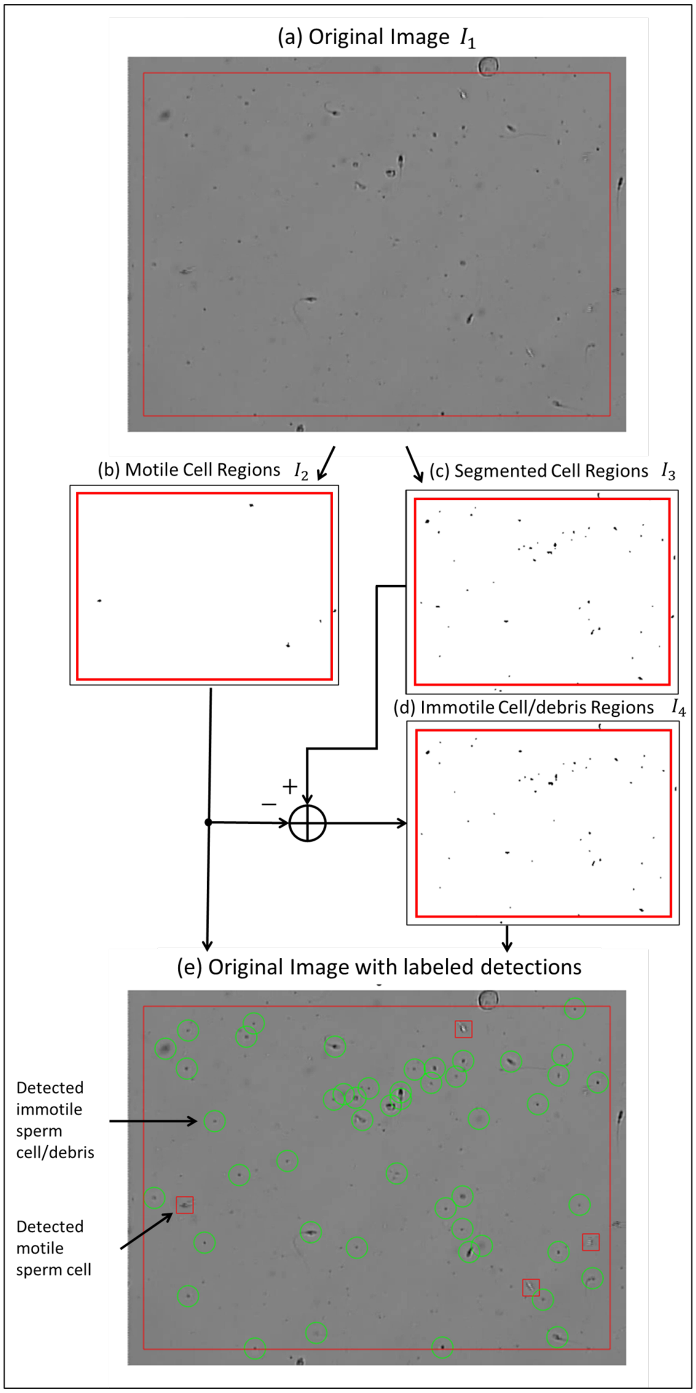

(a) Original Image I_1
(b) Motile Cell Regions I_2
(c) Segmented Cell Regions I_3
(d) Immotile Cell/debris Regions I_4
+
−
(e) Original Image with labeled detections
Detected immotile sperm cell/debris
Detected motile sperm cell

Figure 3.13 Segmentation of cell/debris in unwashed human semen sample. (a) Frame of semen sample image. (b) Regions of motile sperm cells segmented using motion detection algorithm in Section 3.2.1.2. (c) Segmented Cell regions using Urbano's segmentation algorithm. (d) Regions of immmotile cell and debris obtained from subtracting motile cell region (b) from the segmented cell regions (c). (e) Original image with detections labeled. (Green circle) Detected immotile cells and debris. (Red square) Detected motile sperm cells.

sperm cells and detected immotile cells and debris marked by red squares and green circles, respectively. Afterwards, the groups of pixels in image I_4 were labeled assuming eight-connectivity. Any group that contained 5 or more pixels was labeled as an immotile cell or debris.

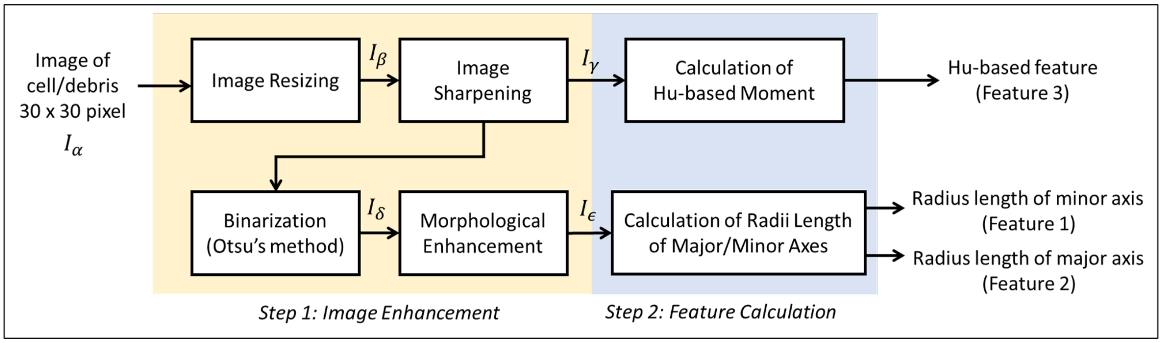


Image of cell/debris 30 x 30 pixel
I_α
Image Resizing
I_β
Image Sharpening
I_γ
Calculation of Hu-based Moment
Hu-based feature (Feature 3)
Binarization (Otsu's method)
I_δ
Morphological Enhancement
I_ϵ
Calculation of Radii Length of Major/Minor Axes
Radius length of minor axis (Feature 1)
Radius length of major axis (Feature 2)
Step 1: Image Enhancement
Step 2: Feature Calculation

Figure 3.14 Feature extraction process. Input is 30×30 pixel image of a cell or debris. The first step of the process is image enhancement. This consists of (1) image resizing, (2) image sharpening, (3) binarization, and (4) morphological enhancement.

3.2.3 Step 3: Feature Extraction

In step 3, the inputs are (1) segmented regions of moving sperm cells (I_2) and (2) segmented regions of immotile cells/debris (I_4). The outputs of step 3 are features of the each segmented object in images I_2 and I_4. The feature extraction from the detected cells comprised of two steps:

1. Image enhancement of detected region of cell/debris,
2. Calculation of features: radii length of detected blob along the minor and major axes (features 1 and 2), and Hu-moments-based feature (feature 3).

The flowchart of the feature extraction process is shown in Figure 3.14. The results within the stages of feature extraction are shown in Figure 3.15.

3.2.3.1 Image Enhancement of Detected Region of Cell/debris. The 30×30 pixel around each of the detected cell/debris was presented for feature calculation. Let the $30 \times$

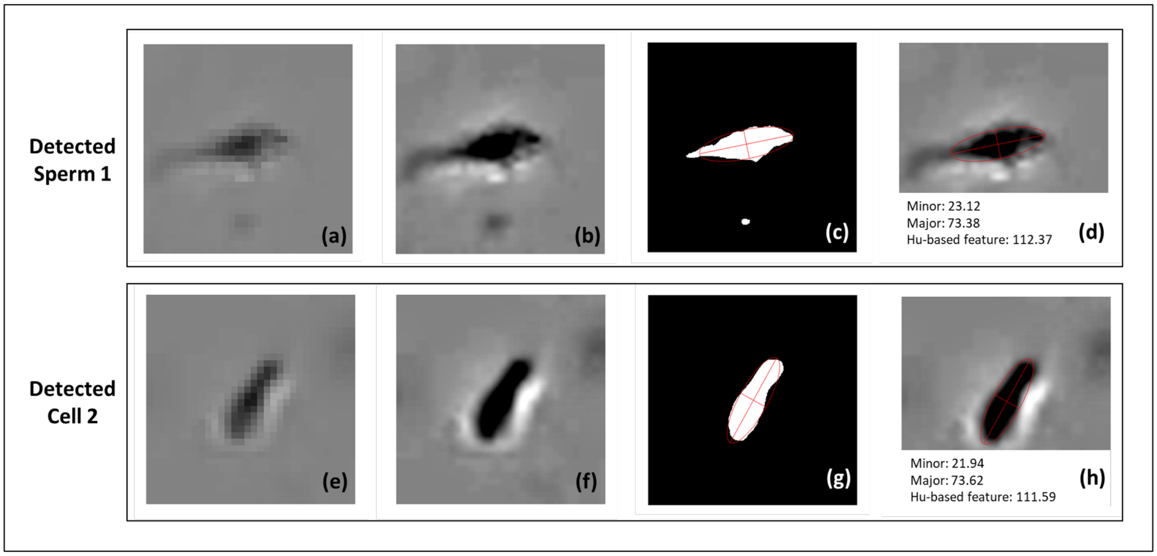


Figure 3.15 Results of stages within the Feature extraction process. (a,e) 30 × 30 pixel image of a cell. (b,f) Enlarged and sharpened image of (a,e). (c,g) Binarized and morphologically enhanced image of (b,f) with fitted ellipse shown in red. (d,h) Original image with red ellipse plotted on top of the detected cell. The calculated features listed below.

30 image be denoted I_α. Prior to feature extraction, the images of the cell/debris were enhanced. The enhancement step consisted of (1) image resizing, (2) image sharpening, (3) binarization, and (4) morphological enhancement. The process is shown in Figure 3.14.

1. Image resizing: the images were resized by a factor of 10 using bicubic interpolation (image interpolation algorithm, Section 2.4 in [43]). The bicubic interpolation was performed using *imresize* function in MATLAB. The resulting image is 300 × 300 pixel image and is denoted as I_β.

2. Image sharpening: the enlarged image is sharpened using the unsharp mask [43]. First, the low-pass filtered image $I_{\beta_{lp}}$ is generated by filtering the image I_B by a Gaussian filter. The equations of the operation are given in Equations (3.10) and (3.11). The unsharp mask is defined as

$$I_{mask} = I_\beta - I_{\beta_{lp}}. \tag{3.17}$$

I_{mask} is an image that contains high-pass (frequency) content of the image. The mask is scaled by a factor k and added to the original image I, generating a sharpened image:

$$I_\gamma = I_{\beta_{sharp}} = I_B + kI_{mask}. \tag{3.18}$$

The value of k determines how much of the high-pass content one is adding to the original image. In this study, $\sigma = 50$ and $k = 2$. The sharpened image is denoted as I_γ. Figures 3.15b and 3.15f show a result of resizing and sharpening of Figures 3.15a and 3.15e.

3. Image binarization: the sharpened image is then binarized using the Otsu's thresholding method [54]. This image is denoted as I_δ.

4. Morphological enhancement: the binarized image is dilated and eroded with 15×15 diamond structuring element [43]. This process preserves the overall size of the detected region, but connects the regions that may have been disconnected during the binarization phase. Figures 3.15c and 3.15g show the result of binarization and morphological enhancement of Figures 3.15b and 3.15f. This image is denoted as I_ϵ.

3.2.3.2 Calculation of Features. Three features are calculated from I_γ and I_ϵ in the image enhancement process. These features are a Hu-moments-based feature and length of radii along the major and minor axes of ellipsoid drawn on the detected cell.

- Hu-moment-based feature: Hu moments are features of a two-dimensional image which are invariant with respect to scale, translation, and rotation [60]. Hu moments are derived from moments and central moments of the image. The Hu moments are calculated from I_γ (sharpened image). The moments are defined as

  $$M_{ij} = \sum_x \sum_y x^i y^i I_\gamma(x, y), \tag{3.19}$$

  where $I_\gamma(x, y)$ is the image intensity value at pixel at location (x,y). The central moments are defined as

  $$\eta_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y), \tag{3.20}$$

  where $\bar{x}$ and $\bar{y}$ are the coordinates of the centroid, computed as:

  $$\bar{x} = \frac{M_{10}}{M_{00}}, \tag{3.21}$$

  $$\bar{y} = \frac{M_{01}}{M_{00}}. \tag{3.22}$$

  The normalized central moments can be obtained as follows:

  $$\mu_{pq} = \frac{\eta_{pq}}{\eta_{00}^{(\frac{p+q}{2})}}. \tag{3.23}$$

  The seven Hu moments are defined as:
  $H_1 = \mu_{20} + \mu_{02}$
  $H_2 = (\mu_{20} - \mu_{02})^2 + 4(\mu_{11})^2$
  $H_3 = (\mu_{30} - 3\mu_{12})^2 + (\mu_{03} - 3\mu_{21})^2$

$H_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{03} + \mu_{21})^2$
$H_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2)$
$H_6 = (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$
$H_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2)$

In [62], Al-Azzo et al. used the sum of the seven Hu moments for classification of human actions. The similar approach is taken here for classification of sperm cells. The Hu-moments-based feature P_{Hu} used for classification is defined as

$$P_{Hu} = \sum_{i=1}^{7} \log_{10}(|H_i|), \tag{3.24}$$

where $|\cdot|$ is the absolute value function.

- Length of radii: Each detected region is approximated by an ellipse using I_η (enhanced image). This approximation is performed by finding an ellipse that has the same second central moments as the region that is being approximated (Section 11.4 of [43]). In Figure 3.15c and Figure 3.15g, the approximated ellipse is shown in red. The length of major and minor axes is divided by 2 to calculate the length of the radii (r_{Major} and r_{Minor}).

These three features (length of radii and Hu-moments-based feature) are calculated for the moving sperm cells and non-sperm cells/debris. The mean and standard deviation for the three features of the moving sperm cells are calculated. The mean and the standard deviation of r_{Minor} are denoted as μ_1 and σ_1, respectively. The mean and the standard deviation of r_{Major} are denoted as μ_2 and σ_2, respectively. The mean and the standard deviation of P_{Hu} are denoted as μ_3 and σ_3, respectively.

Figure 3.16 shows the histograms of the three features. The red vertical line represents the mean (μ_n) of the distribution of each histogram and the black dotted vertical lines represent different standard deviation (σ_n) values away from the mean ($\mu_n \pm N\sigma_n$ for $N = \{1, 2, 3, 4\}$).

3.2.4 Step 4: Classification - Immotile Sperm Cell Detection

The three features (length of radii and Hu-moments-based feature) are calculated from each of the detected non-moving sperm cell/debris. There are three local detectors for each

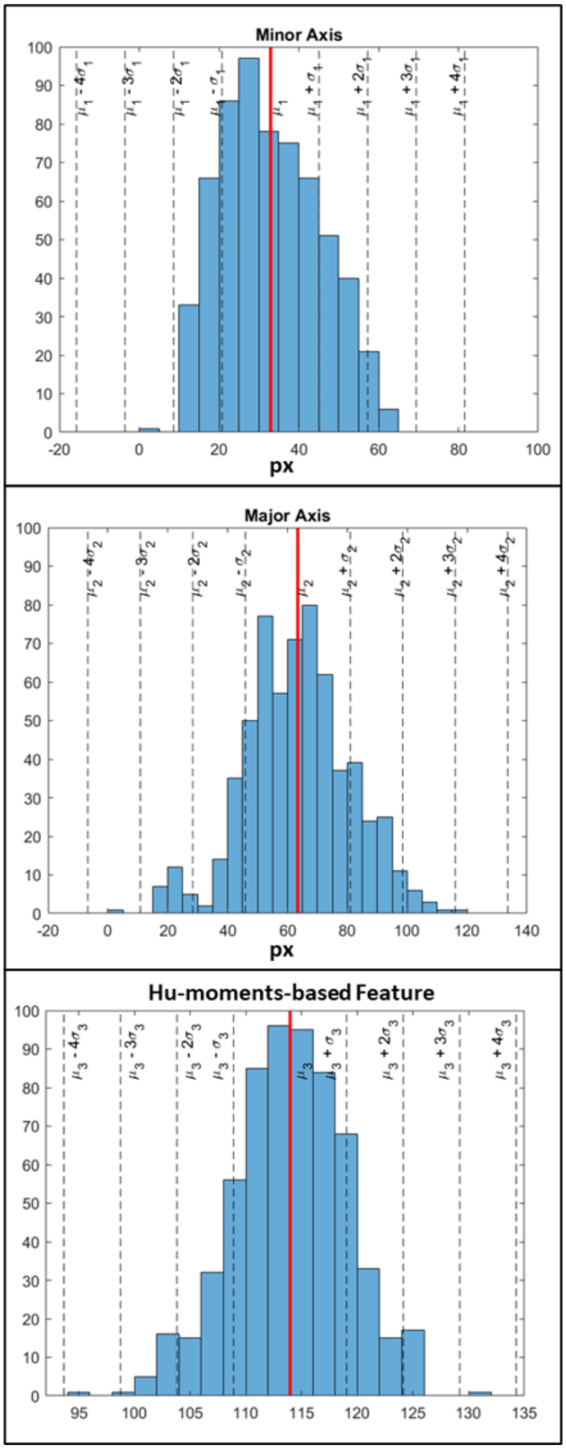


Figure 3.16 Histogram of three features of moving sperm cells. Mean value shown with vertical red line. (Top) Histogram of radius along the minor axis r_{Minor}. (Middle) Histogram of radius along the major axis r_{Major}. (Bottom) Histogram of Hu-moments-based feature P_{Hu}.

feature. Each local detector checks if the values of three features are within the range of $[\mu_n - N\sigma_n, \mu_n + N\sigma_n]$ as shown in Figure 3.17. The local decisions, D_1, D_2, and D_3, are defined as

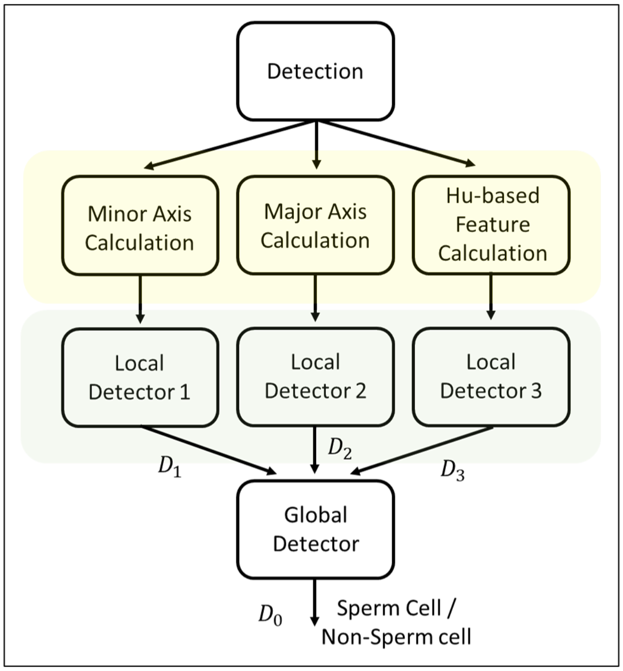


Figure 3.17 Sperm cell classification process.

$$D_1 = \begin{cases} 1, & \text{if } \mu_1 - N\sigma_1 \leq r_{Minor} \leq \mu_1 + N\sigma_1 \\ 0, & \text{otherwise} \end{cases}, \tag{3.25}$$

$$D_2 = \begin{cases} 1, & \text{if } \mu_2 - N\sigma_2 \leq r_{Major} \leq \mu_2 + N\sigma_2 \\ 0, & \text{otherwise} \end{cases}, \tag{3.26}$$

$$D_3 = \begin{cases} 1, & \text{if } \mu_3 - N\sigma_3 \leq P_{Hu} \leq \mu_3 + N\sigma_3 \\ 0, & \text{otherwise} \end{cases}, \tag{3.27}$$

The global decision D_0 is defined as

$$D_0 = D1 \wedge D2 \wedge D3, \tag{3.28}$$

where $\wedge$ is an AND operation. The global decision, D_0, is "1" (or sperm cell) if all local detectors decide that the detected cell is a sperm cell and "0" (or non-sperm cell) otherwise. The value of the decision boundary parameter "N" is critical to the overall performance of the algorithm. If N is too large, then all the detected non-moving objects will be labeled as sperm cells, which will cause false detections. If N is too small, then none of the detected non-moving objects will be labeled as sperm cells, which will cause missed detections. In our studies, the value of $N = 1.2$ was shown to be the most suitable (see Section 3.2.6).

3.2.5 Performance Metric

Precision, recall, and F$_\beta$-score are used to assess the performance of detection for the five (5) algorithms. The five algorithms are Otsu, Adaptive, Urbano, Abbiramy, and Digital Washing (proposed algorithm). The definition of precision and recall are the same as in Section 3.2.1.2. F$_\beta$-scores are calculated as

$$F_\beta = \frac{(1+\beta^2)(Precision)(Recall)}{\beta^2(Precision) + Recall}, \tag{3.29}$$

Similar to our previous testing with the motion detection algorithm in Section 3.2.1.2, the cutoff distance between the ground truth and detection location is $c = 10$ pixels. F$_\beta$-score is a weighted average of precision and recall. For example, if $\beta = 2$, the weight of the precision is lower than the recall, if $\beta = 0.5$, the weight of the recall is lower than the precision, and if $\beta = 1$, the weight of the recall and precision are equal.

3.2.6 Parameter Selection for Digital Washing

An important parameter for the digital washing algorithm is the decision boundary parameter N. N determines the range that identifies the non-moving object as a sperm cell or a non-sperm cell. In our experiments, the ground truth locations of each cell were labeled manually. Using the ground truth, the value of precision and recall were calculated for different values of N. The primary goal of the digital washing algorithm is to reduce the number of false detections (considering debris or other cells as sperm cells). While not eliminating genuine sperm cells, the performance of the algorithm is shown for the value of N that the maximizes value of $F_{0.5}$ (higher weight for precision). The value of the boundary decision parameter is $N = 1.2$ as shown in Figure 3.18. If the three features computed on a non-moving cell object within the range $[\mu_n - 1.2\sigma_n,\ \mu_n + 1.2\sigma_n]$, then the non-moving object is classified as sperm cell. Otherwise, the non-moving object is classified as non-sperm cell. In practice, the ground truth locations of sperm cells are not given and we recommend $N = 1.2$ based on our experiments.

3.3 Experimental Results

The performance of the digital washing algorithm was compared to that of other cell segmentation and localization operating on the same images. We studied Otsu's algorithm [54], adaptive algorithm by Bradley [55], Abbiramy's algorithm [11], and Urbano's algorithm [9]. We compared precision, recall, and F$_\beta$-score for $\beta = 0.5, 1,$ and 2. The three metrics (precision, recall, and F$_\beta$-score) were computed for two samples (Sample ID019 and ID021). The collection process of these samples are the same as the process described in the Section 2.3.1 without the semen sample washing process (unwashed samples). There are five (5) scenes for sample ID019 and four (4) scenes for sample ID021. The length of each scene is 200 frames.

The mean and standard deviation for the nine (9) different testing sets are shown in Table 3.2. The best performing average values are underlined. In this table, one can

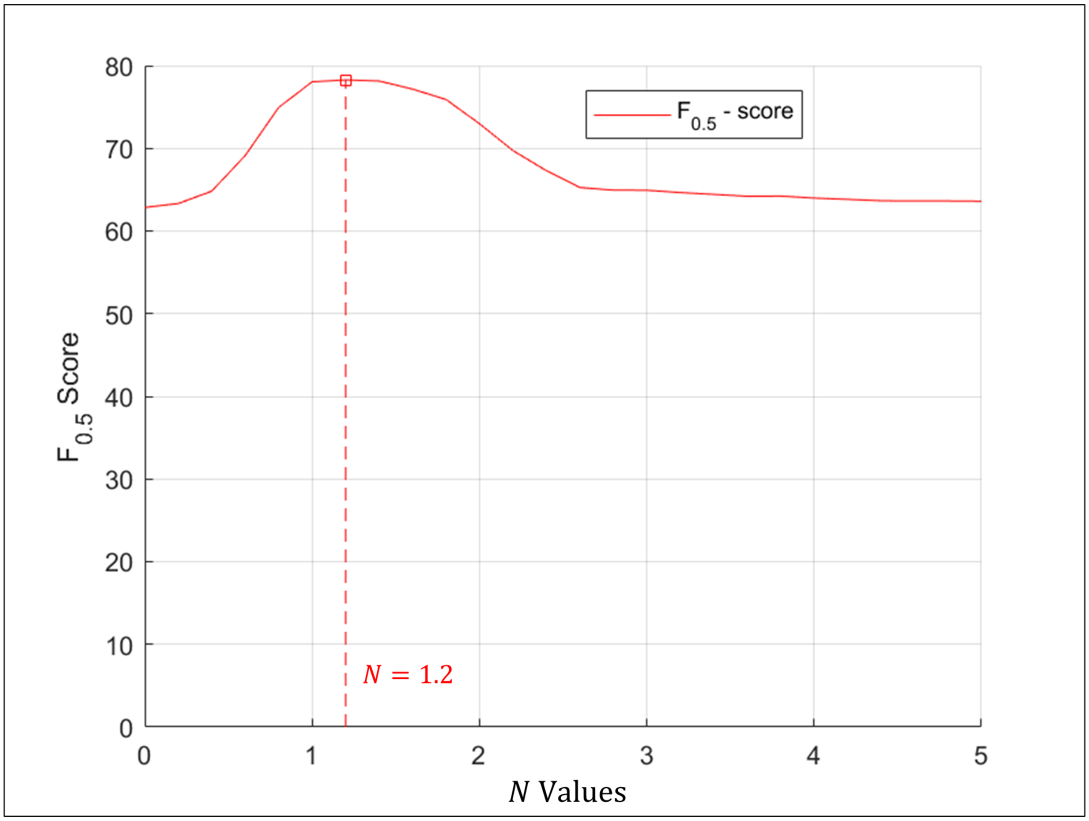


Figure 3.18 Values of $F_{0.5}$ for varying values of boundary decision parameter N.

observe that the digital washing algorithm obtains the highest precision, $F_{0.5}$ score, and F_1 score. For the recall rate, Abbiramy's algorithm showed the highest performance at the rate of 92.55%. For F_2 score, Urbano's algorithm showed the highest performance at the value of 78.60. While the recall rate for the digital washing algorithm was lower compared to the recall rates of the other segmentation and localization algorithms, the digital washing algorithm was shown to be the best algorithm to reduce the rate of false alarm (removing the detection of non-sperm cells and debris by "washing" the semen sample).

In addition, two-sample t-tests are performed for the performance of the digital washing algorithm to the four cell segmentation algorithm on precision and $F_{0.5}$-score ($\alpha = 0.1$). The tested hypotheses on precision and $F_{0.5}$-score with their corresponding p-values are shown in Tables 3.3 and 3.4, respectively. As shown in Tables 3.3 and 3.4, the performances of the digital washing algorithm is higher than the four cell segmentation algorithms in terms of precision and $F_{0.5}$ scores (statistically significant).

Table 3.2 Precision, Recall, and F$_\beta$-score for the Five Different Cell Segmentation Algorithm (Values of Mean μ and Standard Deviation σ for Nine (9) Different Scenes of Unwashed Samples ($N = 1.2$))

	Otsu		Adaptive		Urbano		Abbiramy		Digital Washing	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Precision	71.03	15.36	73.36	14.10	57.88	13.70	46.20	11.73	$\underline{86.21}$	8.81
Recall	68.82	11.87	63.99	16.62	88.19	7.45	$\underline{92.55}$	4.34	59.86	11.76
$F_{0.5}$	69.64	11.14	70.31	12.24	61.64	12.71	51.01	11.59	$\underline{78.27}$	4.94
F_1	68.53	7.72	67.09	12.65	68.72	10.61	60.7	10.49	$\underline{69.55}$	6.34
F_2	68.39	9.03	64.97	14.84	$\underline{78.60}$	7.40	75.85	6.65	63.21	9.70

Table 3.3 One Sided Hypothesis Testing on Precision on the Performance of Digital Washing Algorithm to Four Other Cell Segmentation Algorithms ($N = 1.2$)

	Hypothesis	p-value
Test I:	H_0: Prec(Digital Washing) - Prec(Otsu) = 0 H_a: Prec(Digital Washing) - Prec(Otsu) > 0	0.01652
Test II:	H_0: Prec(Digital Washing) - Prec(Adaptive) = 0 H_a: Prec(Digital Washing) - Prec(Adaptive) > 0	0.02451
Test III:	H_0: Prec(Digital Washing) - Prec(Urbano) = 0 H_a: Prec(Digital Washing) -Prec(Urbano) > 0	0.0004
Test IV:	H_0: Prec(Digital Washing) - Prec(Abbiramy) = 0 H_a: Prec(Digital Washing) -Prec(Abbiramy) > 0	0.00002

To show the different operating points for the digital washing algorithm, precision, recall, and F$_\beta$ scores are calculated for different boundary decision values N (F$_\beta$ as a function of N). The values of N are $\arg\max_N \; F_\beta(N)$, $\beta = 0.5, 1$, and 2. The results are provided in Table 3.5 and the highest value in each row has been underlined. In Table 3.5, one can observe the performance of the digital washing algorithm for different weights for precision and recall. When the weight for precision is higher than recall, the digital washing algorithm reaches a precision rate of 86.21% ($\beta = 0.5$). When the weight for the recall is higher than precision, the digital washing algorithm can reach a recall rate of

Table 3.4 One Sided Hypothesis Testing on $F_{0.5}$ Scores on the Performance of Digital Washing Algorithm to Four Other Cell Segmentation Algorithms ($N = 1.2$)

	Hypothesis	p-value
Test I:	H_0: $F_{0.5}$(Digital Washing) - $F_{0.5}$(Otsu) = 0 H_a: $F_{0.5}$(Digital Washing) - $F_{0.5}$(Otsu) > 0	0.03318
Test II:	H_0: $F_{0.5}$(Digital Washing) - $F_{0.5}$(Adaptive) = 0 H_a: $F_{0.5}$(Digital Washing) - $F_{0.5}$(Adaptive) > 0	0.05401
Test III:	H_0: $F_{0.5}$(Digital Washing) - $F_{0.5}$(Urbano) = 0 H_a: $F_{0.5}$(Digital Washing) - $F_{0.5}$(Urbano) > 0	0.00321
Test IV:	H_0: $F_{0.5}$(Digital Washing) - $F_{0.5}$(Abbiramy) = 0 H_a: $F_{0.5}$(Digital Washing) - $F_{0.5}$(Abbiramy) > 0	0.00009

Table 3.5 Precision, Recall, and F$_\beta$ Scores of Digital Washing Algorithm for Different Values of N (Boundary Decision Parameter) for Maximizing F$_\beta$-score for $\beta = 0.5, 1, 2$

	$\arg\max_N F_{0.5}(N)$	$\arg\max_N F_1(N)$	$\arg\max_N F_2(N)$
Precision	$\underline{86.21}$	75.96	60.01
Recall	59.86	79.46	$\underline{90.51}$
$F_{0.5}$	$\underline{78.27}$	75.90	63.82
F_1	59.55	$\underline{76.50}$	70.97
F_2	63.21	77.95	$\underline{80.89}$

90.51% ($\beta = 2$). When the weights are equal, the algorithm is able to reach precision and recall rates of 75.96% and 79.46%, respectively. The maximum values of $F_\beta(N)$ for $\beta = 0.5, 1,$ and 2 are 78.27%, 76.50%, and 80.89%, respectively.

3.4 Conclusion

In this chapter, an algorithm for digital washing of unwashed semen samples was presented. The algorithm consists of four parts, motile cell segmentation, non-moving object segmentation, feature extraction, and classification of sperm cells.

The algorithm is based on the assumption that sperm cells are the only cells that will be actively moving in the semen sample. The motion detection (2D-SGM) algorithm based on single Gaussian distribution for each pixel is proposed for the detection of motile sperm cells. Using the washed semen samples, the ability of 2D-SGM algorithm to achieve high precision (or low false alarm rate) was compared to other segmentation algorithms and to a well-studied motion detection algorithm. The digital washing algorithm uses the similarities in the features of the immotile objects and the features of motile sperm cells to identify the immotile objects. The algorithm segments and classifies the immotile sperm cells from the scene.

The digital washing algorithm was compared to other algorithms that perform segmentation and detection (namely, Otsu [54], adaptive algorithm by Bradley [55], Abbiramy [11], and Urbano [9]) in terms of precision, recall, and F_β scores. Our study concludes that at a significance level α = 0.1, the digital washing algorithm is statistically better than the four cell segmentation algorithms for detecting sperm cells in unwashed semen samples in terms of precision and $F_{0.5}$ score. It should be noted that the four cell segmentation algorithms we used for comparison were not designed for washing of unwashed semen samples. The digital washing algorithm proposed in this chapter may enhance the capabilities of computer-assisted semen analysis systems by making them able to process unwashed semen samples. This capability removes therefore removing the need to pre-process/wash the semen samples.

For future studies, one can explore the methods to classify the cell/debris in the unwashed sample to get a count of not only sperm cells but other non-sperm cells in the semen sample. As the concentration of non-sperm cells can be an indicator of a cause of infertility, the study of non-sperm objects may hold some value. In addition, the algorithm suggests a new method of segmentation and localization of the sperm cells. For the calculation of motility, one is able to explore different methods of tracking to obtain the tracks of detected sperm cells in unwashed samples.

CHAPTER 4

CLASSIFICATION OF SPERM CELL SWIM TYPE

In this chapter, a method of classifying a sperm cell by performing a parameter subset search is presented. The Artificial Bee Colony (ABC) algorithm and the K-means clustering algorithm were used on motility parameters of sperm cells in a sample (simulated and real) to classify and cluster the different swim types.

4.1 Background

The significance of sperm concentration and percent of motile sperm is clear in male fertility analysis. However, the significance of some sperm motility parameters is a subject of on-going research [63]. Several studies have used clustering methods to assist in this line of investigation. The clustering methodology proposed in this study is motivated by the review of clustering on motility parameters of CASA systems by Martinez-Pastor et al. [63]. The fully automated multi-sperm tracking algorithm introduced by Urbano et al. [9, 10] is used to explore the clustering task of human sperm swim subpopulation.

Sperm cell movements can be divided into three major categories: progressive, non-progressive, and immotile [1]. Immotile movements define cells which show no movement at all, and non-progressive movements are defined by motility that lacks progression, where the sperm head shows little displacement over time. Progressive movements characterize actively moving sperm cells. Two-dimensional progressive sperm swimming patterns observed by Babcock et al. [39] were categorized as *linear mean* and *circular*.

The two progressive movements, linear mean and circular, differ by the presence and absence of head rolling. The examples can be seen in Figures 4.1 and 4.2. In Figures 4.1a and 4.1b, which show snapshots of linear mean swim, one can observe the orientation of the head is changing during swimming, representing head roll during

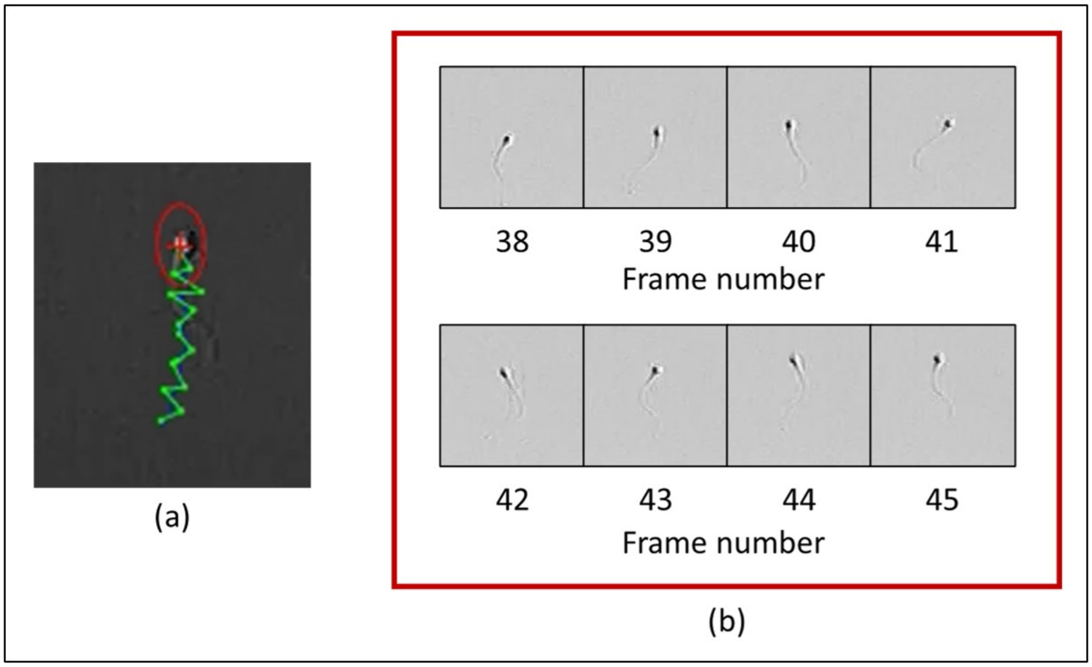


Figure 4.1 (a) Snapshot of sperm A during linear mean swim. (b) Frame-by-frame snapshots of sperm A during linear mean swimming from frame 38 to 45 (observed at 30 FPS).

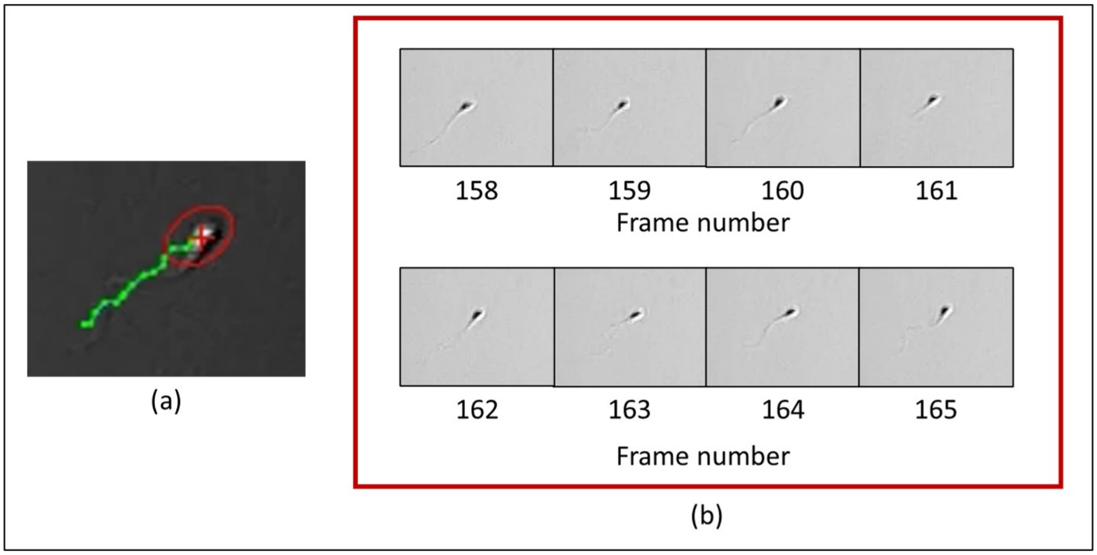


Figure 4.2 (a) Snapshot of sperm B during circular swim. (b) Frame-by-frame snapshots of sperm B during circular swimming from frame 158 to 165 (observed at 30 FPS).

the swim. In Figures 4.2a and 4.2b, which show snapshots of circular swim, one can observe that the orientation of the head stays the same. From this set of observations, the sperm cells of different movements are assumed to have a different range of values of motility parameters in the proposed approach. Using this assumption, K-means clustering is performed on the motility parameters to identify and classify the sperm swimming types (subpopulation analysis). To increase the accuracy of classification, a computational intelligence algorithm, the Artificial Bee Colony (ABC) algorithm, is used to search through different subsets of motility parameters that provide a distinct set of clusters for classification.

This chapter is organized as follows. In Section 4.2, the motility parameter calculation, sperm track data extraction, and sperm track clustering of simulated and real sperm swimming data are described. In Sections 4.3.1 and 4.3.2, results of clustering on simulated data and on two (2) human sperm samples are shown, respectively.

4.2 Methods

4.2.1 Specimen Video Collection and Track Data Extraction

Video clips of human semen specimens prepared and collected by the In-Vitro Fertilization laboratories at Penn Fertility Care were used in this study. The video clips are 200× magnified images of 640×480 pixel resolution (0.857 μm/pixel). The sperm swim track data have been obtained from the video clips by segmentation, localization, and track data association algorithms. The flowchart of the system is shown in Figure 4.3.

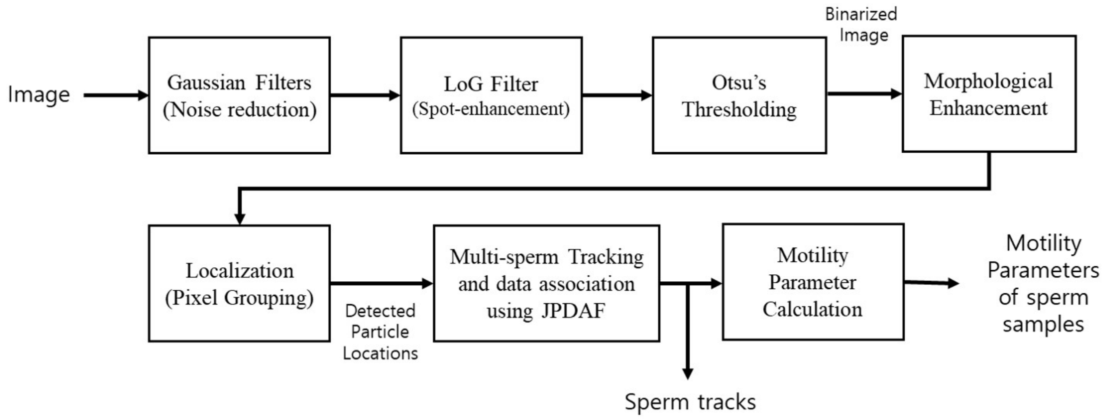


Image
Gaussian Filters (Noise reduction)
LoG Filter (Spot-enhancement)
Otsu's Thresholding
Binarized Image
Morphological Enhancement
Localization (Pixel Grouping)
Detected Particle Locations
Multi-sperm Tracking and data association using JPDAF
Motility Parameter Calculation
Motility Parameters of sperm samples
Sperm tracks

Figure 4.3 Flowchart of the automated sperm cell detection and tracking system proposed by Urbano et al.

Following [9], in the segmentation phase, the image was convolved with 11×11 pixel Gaussian filters five times to reduce image noise. Then, the image was convolved with 9×9 Laplacian of Gaussian (LoG) filter to produce spot-enhance image. Using Otsu's method [54], the intensity threshold was calculated for the image, and the image was binarized. The binarized image then went through morphological enhancements, erosion, and dilation, to reduce spurious detection due to noise. Finally, the sperm cells were "localized" by labeling any group with more than or equal to 5 pixels as a detected sperm head.

As the final step of track data extraction, using the joint probability density association filter (JPDAF) [51], the particle location data were used to track the particles throughout the video clip. By using JPDAF, the probability of missed detection and track mismatch during head collisions in sperm cells was reduced, giving more accurate tracking results compared to other popular track association methods such as the Nearest Neighbor algorithm [9, 10]. For detailed information regarding the procedures of segmentation, localization, tracking, and track data association, see [9].

4.2.2 Motility Parameter Calculation

In the standard CASA systems, the eight different motility parameters are calculated from a track segment, which is the observed path over time T_s.

1. VCL: Curvilinear velocity.
2. VSL: Straight-line velocity.
3. VAP: Average path velocity.
4. ALH: Amplitude of lateral head displacement.
5. LIN: Linearity of the curvilinear path the sperm head (VSL/VCL).
6. WOB: Wobble (VAP/VCL).
7. STR: Straightness (VSL/VAP).
8. MAD: Mean angular displacement.

The curvilinear velocity (VCL) is a measure of rate of travel of sperm cell over time T_s.

$$VCL(k) = \frac{1}{T_s} \sum_{i=k-n}^{k} ||z(i+1) - z(i)|| \tag{4.1}$$

$|| \cdot ||$ is denotes the magnitude (length) of a vector. n is the number of points in the track segment. The value of n is given as

$$n = \frac{T_s}{T}, \tag{4.2}$$

where T is the period or time interval of data (time between each frame or data). Straight line velocity (VSL) is a measure of rate of travel between the first point ($z(k-n)$) and the last point ($z(k)$) in the track segment.

$$VSL(k) = \frac{1}{T_s} ||z(k) - z(k-n)|| \tag{4.3}$$

Average path velocity (VAP) is a measure of rate of travel of a sperm cell over the "average path." Here, the average path ($\hat{z}(k)$) is an estimated path traveled by the sperm cell without the oscillations caused by the beating of the flagellum. The value of VAP is given by

$$VAP(k) = \frac{1}{T_s} \sum_{i=k-n}^{k} ||\hat{z}(i+1) - \hat{z}(i)||. \tag{4.4}$$

The progression ratios that are used to described the motility of sperm cells are Linearity (LIN), Strighness (STR), and Wobble (WOB). These ratios are calculated as integer percentages:

$$LIN = \frac{VSL}{VCL} \times 100, \tag{4.5}$$

$$STR = \frac{VSL}{VAP} \times 100, \tag{4.6}$$

$$WOB = \frac{VAP}{VCL} \times 100. \tag{4.7}$$

Mathematical estimation of amplitude of lateral head displacement (ALH) is given by

$$ALH(k) = \frac{2}{T_s} \sum_{i=k-n}^{k} ||z(i+1) - \hat{z}(i)|| \tag{4.8}$$

where the distance between the smoothed path and the corresponding position along the curvilinear path is averaged and doubled. Lastly, the mean angular displacement (MAD) is defined as average angle between successive locations along the curvilinear path of the sperm head. The angle θ is between two segment ($\underline{a}$ and $\underline{b}$) can be found as

$$\theta = \cos^{-1} \frac{\underline{a} \cdot \underline{b}}{||\underline{a}||||\underline{b}||} \tag{4.9}$$

where $\cdot$ is a dot product operator.

The motility parameters are obtained by processing successive 1-second long track segments ($T_s = 1$). For the task of classification and clustering, the parameters of each particle are then averaged. The median of the parameter measurements was used, as the median has shown better performance in clustering than the mean; the median is less impacted by outliers.

4.2.3 K-means Clustering

In a clustering problem, there is a set of N data points $\{x_1, x_2, ..., x_N\}$. The goal of a clustering algorithm is to find a method of identifying points that would belong to K clusters. K-means clustering is a type of unsupervised learning algorithm that is popularly used for the task of clustering [64]. In K-means clustering, the algorithm initializes a set of points $\{\mu_1, \mu_2, ..., \mu_K\}$, which are "initial" proposed "centers" of K clusters. Each data point in a set $\{x_1, x_2, ..., x_N\}$ is assigned to one of K clusters. Usually, this assignment is based on the least squared Euclidean distance

$$r_{nk} = \begin{cases} 1 & \text{if } k = \text{arg min}_j ||x_n - \mu_j||^2 \\ 0 & otherwise, \end{cases} \tag{4.10}$$

where $k = 1, 2, .., K$. Afterwards, the "center" points are updated by calculating the mean of each clusters

$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}. \tag{4.11}$$

This process of assignment (data point to cluster) and update (center cluster) continues until convergence (no change is observed).

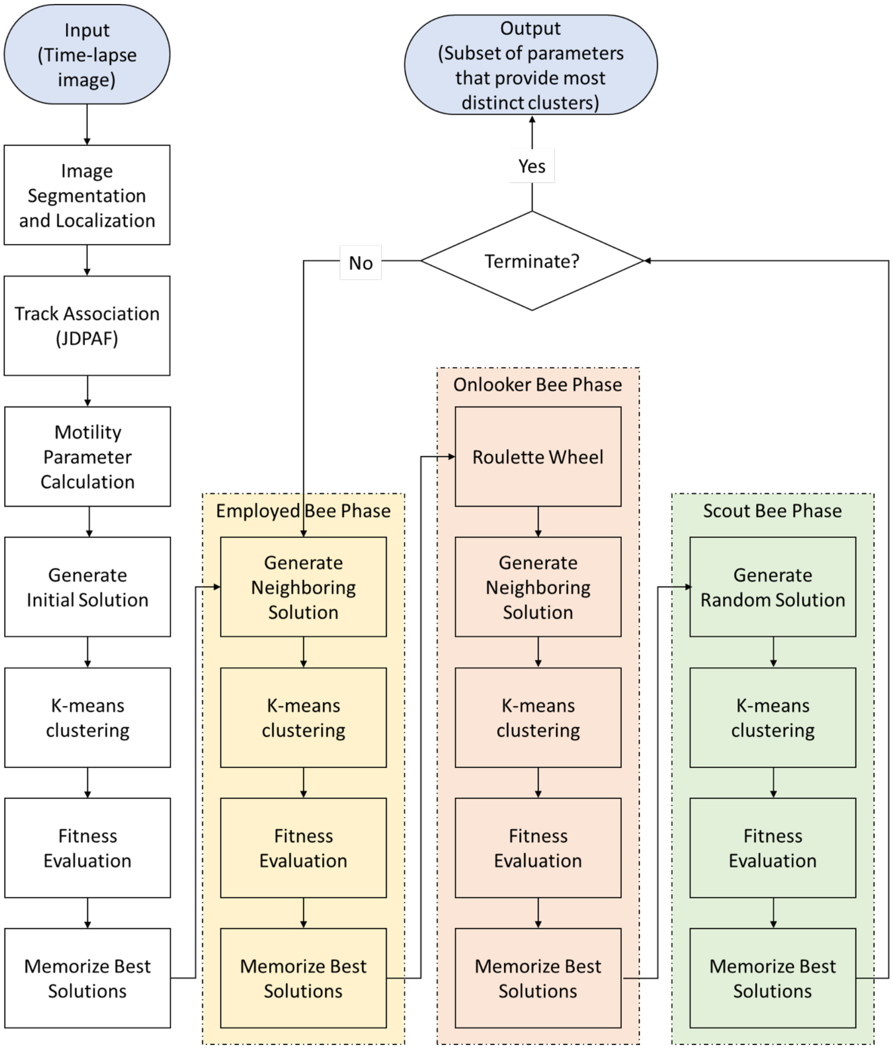


Figure 4.4 Process Flowchart. Time-lapse image is the input and the subset of parameters that provide most distinct clusters is the output.

4.2.4 Sperm Track Clustering and Artificial Bee Colony Algorithm Design

In attempt to find the subset of motility parameters for K-means clustering that will provide distinct clusters of each swimming type, the Artificial Bee Colony (ABC) algorithm was used. ABC algorithm is a computational intelligence algorithm based on the foraging behavior of the honeybee swarm, proposed by Karaboga [65]. The main goal of the ABC algorithm is to find the optimal “food source” or optimal solution for a given problem. ABC algorithm accomplishes this task by using three types of bees: employed bees, onlooker bees, and scout bees.

After gathering food at a food source, bees return to a “dance floor” within the hive. On the “dance floor,” each of the bees performs a waggle dance. The dance provides information to the other bees, such as the direction, distance, and “richness” of the food source [66]. The duration of the dance is longer for richer food sources. In the ABC algorithm, the richness of a food source is referred as the *fitness* of a solution.

For the design of the ABC algorithm, each of the possible subsets of motility parameters (VCL, VSL, VAP, ALH, LIN, WOB, STR, and MAD) is considered a food source, which will be evaluated by bees. The aim of the algorithm would be to find a subset of motility parameters that will produce the most distinct clusters by K-means clustering. Distinctiveness of clusters is measured by the *silhouette values* [67].

A “silhouette value” is a measure of similarity between different data points within the same cluster. The silhouette value for particle i is

$$S_n = \frac{b_n - a_n}{max(a_n, b_n)}, \tag{4.12}$$

where a_n stands for the average distance from data point n to other data points within the same cluster, and b_n is the minimum distance from data point n to other data points in other clusters.

The sum of silhouette values is high for “good” clustering, and low for “poor” clustering. For the ABC algorithm, fitness value (the “richness” of the food source) was defined as the sum of silhouette value for all the data used in clustering (Equation (4.13)). The fitness function for “food source” s for K clusters, with cluster j containing n_j data points is defined as

$$F(s) = \sum_{j=1}^{c}(\sum_{n=1}^{n_j} S_n). \tag{4.13}$$

The first type of bees, employed bees, exploit the known food sources. These employed bees return to their hive, and "dance" on the dance floor. Onlooker bees observe the dance, and decide on their destination. The duration of the dance is proportional to the richness of the food source; the bees that have been to a rich food source will dance longer than the bees who have been to a poor food source. Therefore, it is more likely for the onlooker bees to sight the bee that has been to a rich food source. As a result, the onlooker bees have more chance of going to a rich food source. Scout bees perform a random search in order to find even better food sources. These three types of behavior of bees drive the ABC algorithm.

In the ABC algorithm, the bees are first sent to random locations to search for food. Each bee evaluates each food source, and returns to the hive with a fitness value. Few of the "best" sources are chosen as candidate locations for employed bees and onlooker bees to exploit. Next, the algorithm goes through three phases (employed, onlooker, and scout bees), and repeats until a termination requirement has been met. For the design, the ABC algorithm is terminated at the user-defined iteration.

In the proposed application of the ABC algorithm, possible initial solutions or food sources are proposed in the first step of the ABC algorithm. These food sources are assessed and ranked based on their "fitness" or richness of the food source. Based on these rankings, a list of the best food sources is generated (initialization stage). The employed bees are sent to the food sources that are on the list of best food sources. The employed bees look at the neighboring site to see if any of the neighboring sites contain a better food source than the food sources listed in the current list of best food sources. If any of the bees find a food source that is better, the new site replaces the worst food source listed in the best food source list. Next, the onlooker bees choose their destination by observing the dances of the employed bees. For the onlooker bees, a roulette wheel with the selections proportional to the fitness values in the best food source list is used to select a destination. The probability of choosing food source s_m from the total number of best food sources s_o is defined to as

$$p_t(s_m) = \frac{F(s_m)}{\sum_{k=1}^{s_0} F(s_k)}. \tag{4.14}$$

Afterwards, the onlooker bees observe the neighboring sites, and update the best food source list if any better food source was found. Lastly, during the scout bee phase, the scout bees are sent to random locations to evaluate food sources, and the best food source list is updated if any better food source was found. The scout bees enable a global search, creating a possibility of avoiding convergence to a local maximum. The process is represented as a flowchart in Figure 4.4.

In the context of this problem, every possible subset of motility parameters is a possible food source for the bees to exploit. In the design, an 8-bit binary word was generated to indicate which motility parameters are used. Only the motility parameters corresponding to a "1" are selected for the subset used later in K-means clustering. Initial random food sources are selected by choosing a random binary number between 00000000 and 11111111. The neighboring "food source" is determined by flipping one of the bits in the food source word randomly.

Clustering of the subset of parameters was performed through the unsupervised-learning based K-means clustering method. Specifically, the built-in function for K-means clustering in MATLAB was used, which is the K-means++ algorithm by Arthur [68]. It was assumed that there were three (3) distinct clusters in the processed data, with the three (3) clusters representing each of the sperm swim categories, namely the linear mean, circular, and non-progressive (including immotile). Each K-means clustering has been repeated five times to reduce the probability of finding a local minimum.

4.3 Results

In this section, the results of classification and clustering is shown using the simulated images and real human semen images. The simulated images were generated using the swim tracks observed in the real human semen images. The classification and clustering results (simulated and real) are shown for two samples (sample ID005 and sample ID007). Samples ID005 and ID007 are denoted as sample 1 and sample 2, respectively. The frame rate of sample 1 is 15 FPS and the frame rate of sample 2 is 30 FPS. The details of the semen image collection process are described in the Section 2.3.1.

4.3.1 K-means Clustering of Synthetic Sperm Swim Tracks with the ABC Algorithm

Using the mathematical models described in Section 2.2.2, simulated images of sperm cells were generated. The tracks were categorized as (1) circular, (2) linear mean, or (3) immotile/dead. From tracks of each category of samples 1 and 2, swimming parameters were extracted and averaged. The height of the ribbon r_v in linear mean swim was approximated to be $\frac{2}{3}$ of width of the ribbon r_h (see Section 2.2.2). The simulation parameters extracted from the two samples are found in Table 4.1.

Table 4.1 Simulation Parameters Extracted from Samples 1 and 2

Swim Type	Circular (Table 2.1)				Linear mean (Table 2.3)			
Variables	r_c	f_c (Hz)	f_s (Hz)	a	V	f_l (Hz)	r_h	r_v
Sample 1	180.88	0.1030	11.51	2.89	91.85	10.55	9.00	6.00
Sample 2	251.59	0.0368	3.54	6.62	29.11	1.87	12.11	8.07

Synthetic swim track data for 100 cells for the three swimming types ((1) circular, (2) linear mean, and (3) immotile/dead) was generated. For testing, the simulation parameters for each swim type were randomly selected from a normal distribution with the parameters specified in Table 4.1 as the mean. Multiple scenes were generated at which the standard deviation for the normal distribution ranged from $p_\sigma = 0$ to 25 percent of the mean value. In addition, it was assumed that all the particles were disturbed by a small additive Brownian motion (to model the influence of the movement of the fluid). The Brownian motion was equal to $\sigma_B W(t)$ where $W(t)$ is a standard 1-dimensional Brownian motion ($W(t) \sim N(0, t)$). The value of σ_B was set to 5 ($\sigma_B = 5$). The duration of the simulated swim was for 15 seconds. The frame rate for data for sample 1 was set to 15 FPS and the frame rate for sample 2 was set to 30 FPS to match the sample that the simulation parameters were extracted from. A total of 20 different scenarios for samples 1 and 2 were generated.

For the ABC algorithm, ten (10) employed bees, five (5) onlooker bees, and five (5) scout bees were used for 50 iterations. Accuracy was used to assess the performance of the classification algorithm. Accuracy is defined to be the sum of correct classifications divided by the total number of classifications. To observe the effectiveness of the classification algorithm based on ABC search, the accuracy of classifying swim types based on K-means clustering using additional three motility parameter subsets is shown. These three subsets are:

1. {VCL, VSL} (subset 1),
2. {VCL, VSL, ALH} (subset 2), and
3. {VCL, VSL, ALH, LIN} (subset 3).

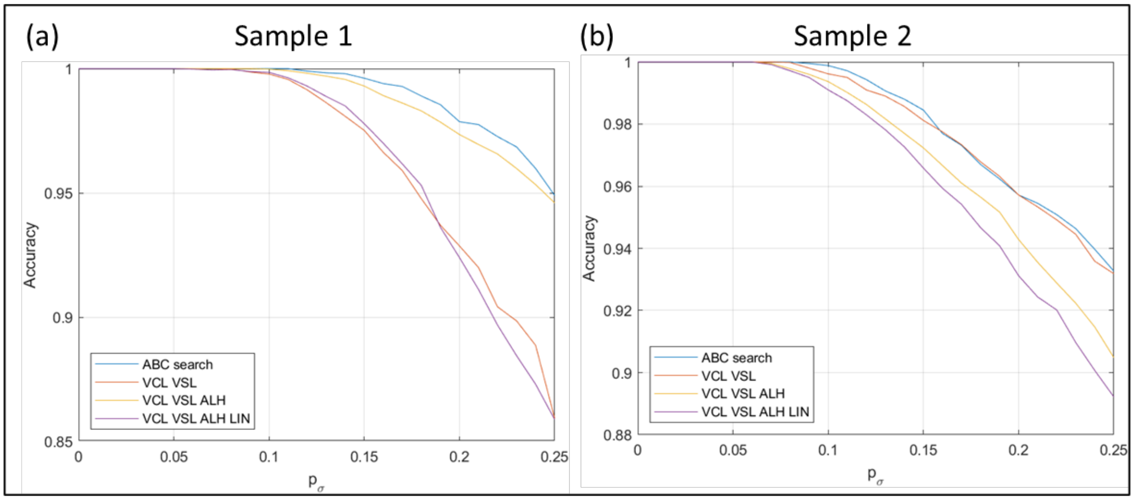


Figure 4.5 Accuracy of clustering for synthetic tracks generated using sample 1 and sample 2. (Blue) Subset selected using ABC search. (Orange) VCL and VSL used for clustering. (Yellow) VCL, VSL, and ALH used for clustering. (Purple) VCL, VSL, ALH, LIN used for clustering.

The results are shown in Figure 4.5. The curves show mean accuracy for 20 scenarios generated for both samples. For samples 1 and 2, the ABC search based clustering have shown the best performance out of the four methods. Clustering based on {VCL, VSL, ALH, LIN} has performed the worst for both samples, suggesting that the increase in the number of parameters for clustering does not always improve clustering performance.

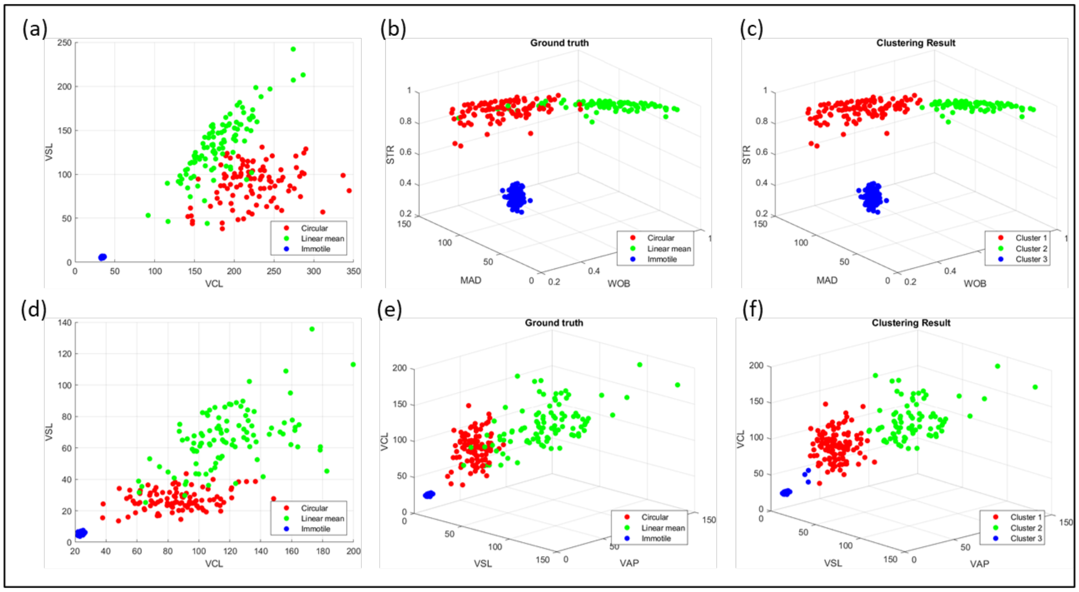


Figure 4.6 Clustering results on synthetic tracks generated parameters extracted from sample 1. (a, d) Scatter plots (VCL vs. VSL) of three swim types (ground truth) for $p_\sigma = 0.25$ for simulated images of samples 1 and 2, respectively. (b, e) Scatter plots (subset parameters selected using ABC search) of three swim types (ground truth) for $p_\sigma = 0.25$ for simulated images of samples 1 and 2, respectively. (c, f) Scatter plots (subset parameters selected using ABC search) of three swim types (result of clustering) for $p_\sigma = 0.25$ for simulated images of samples 1 and 2, respectively.

To show the clustering results using the ABC search algorithm, scatter plots of ground truth clusters and ABC search based clusters are shown in Figure 4.6 for samples 1 and 2. Parts (a, d) of Figure 4.6 show scatter plots (VCL vs. VSL) of three swim types for $p_\sigma = 0.25$ for simulated images of samples 1 and 2, respectively. Parts (b, e) of Figure 4.6 show scatter plots (subset parameters selected using ABC search) of three swim types for $p_\sigma = 0.25$ for simulated images of samples 1 and 2, respectively. The scatter plots of parts (a, b, d, e) of Figure 4.6 have been labeled with ground truth. Points belonging to circular swimming tracks are represented in red, points belonging to linear mean swimming tracks are represented in green, and points belonging to immotile swimming tracks are represented in blue. Part (c, f) of Figure 4.6 show scatter plots (subset parameters selected using ABC search) of three swim types for $p_\sigma = 0.25$ for simulated images of samples 1 and 2, respectively. The colors of each point represent the assignments to one of the clusters. The ABC search algorithm searches for the subsets that would result in high fitness value (distinct clusters). As shown in Figures 4.5 and 4.6, ABC search based clustering has shown the ability to classify swimming types with high accuracy in existence high variation in swimming behaviors.

4.3.2 K-means Clustering of Human Sperm Swim Tracks with ABC Algorithm

Ten (10) employed bees, five (5) onlooker bees, and five (5) scout bees were used in the ABC algorithm. For real sperm data, 100 iterations are used to reduce the likelihood of converging to a local maximum. Clustering results are shown for two different samples, one at 15 FPS (frames per second) and the other at 30 FPS (both observed for 10 seconds).

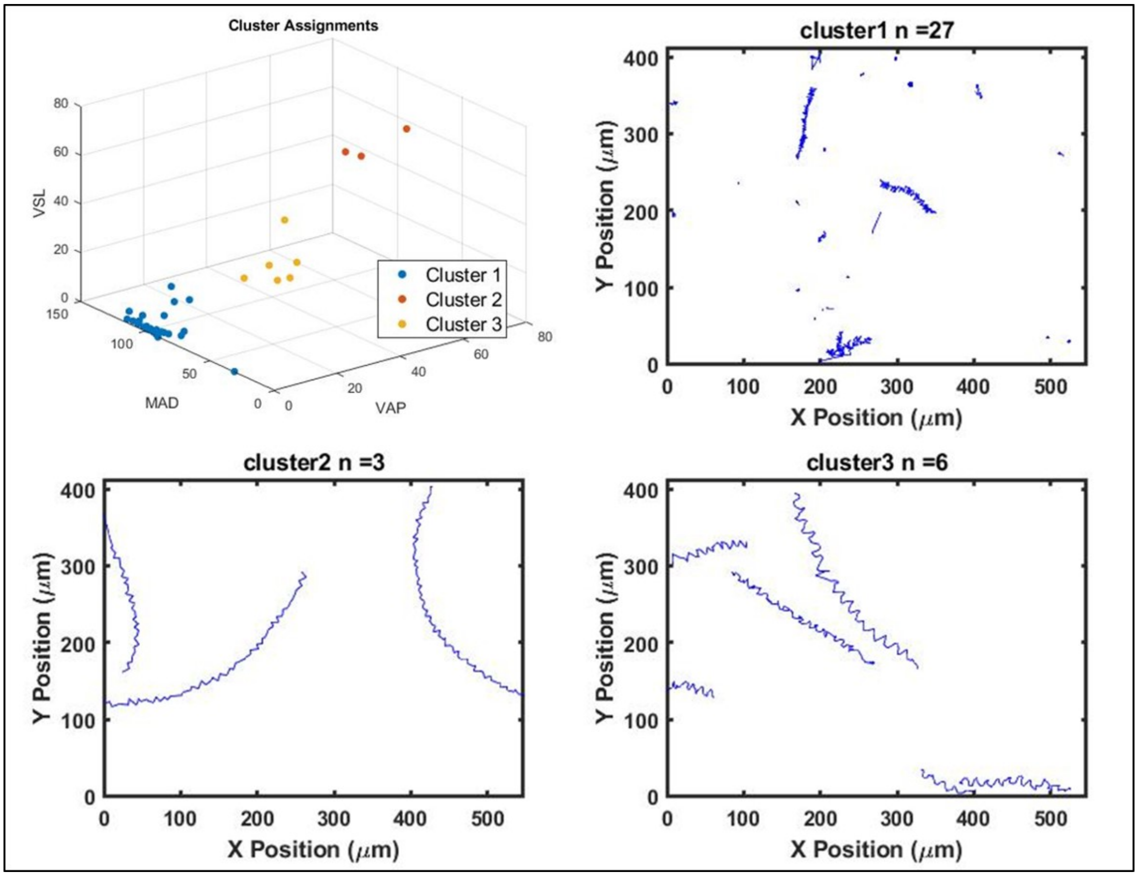


Figure 4.7 Sample 1 clustering result (15 FPS, 10 seconds)

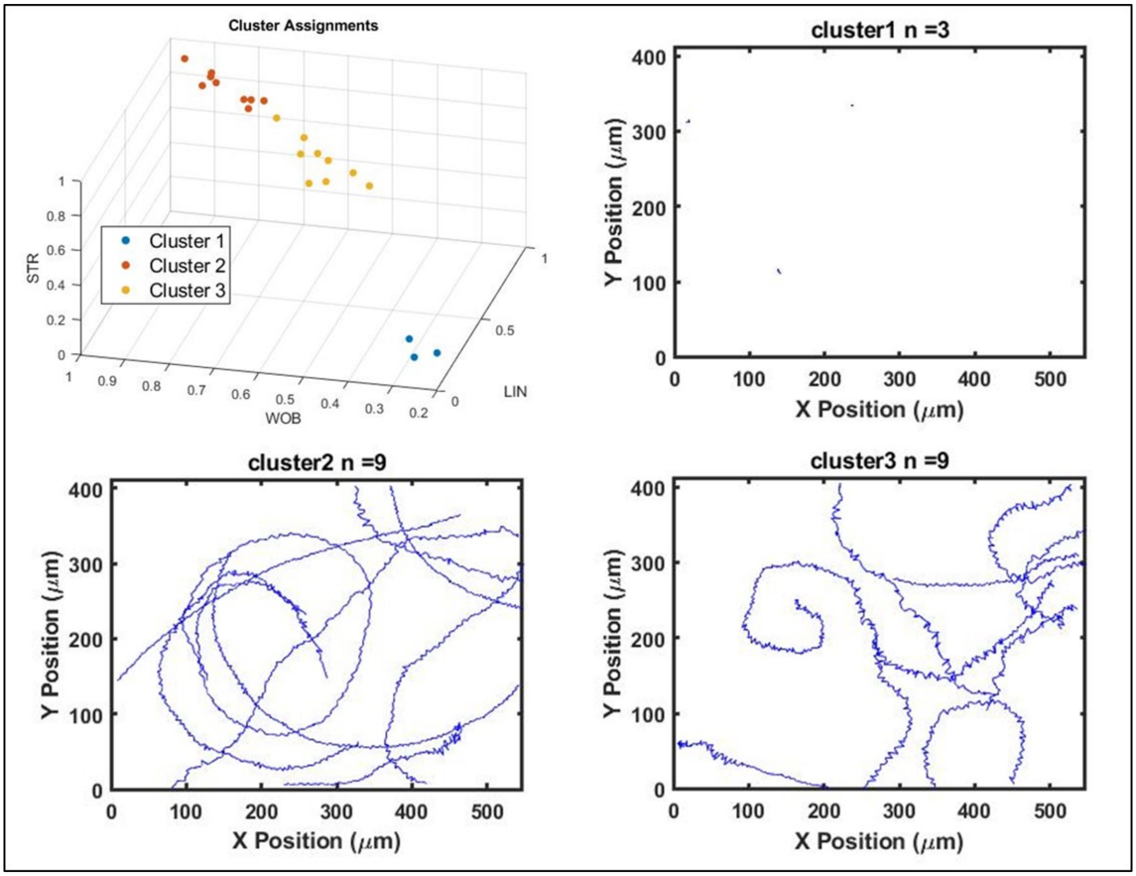


Figure 4.8 Sample 2 clustering result (30 FPS, 10 seconds)

4.3.2.1 Clustering Results - Sample 1.

In sample 1, 36 sperm swim tracks were observed. Motility parameters used for this clustering were VSL, VAP, and MAD.

- Cluster 1: 27 tracks with low straight-line velocity and average path velocity. Linear mean progressive tracks can be observed in this cluster. Few tracks show small linear progression. Due to the very small speed, they have been clustered with the dead or immotile cells.
- Cluster 2 (circular): 3 tracks with high value of straight-line velocity and average path velocity. Tracks show circular swimming path, with small lateral displacement.
- Cluster 3 (linear mean): 6 tracks with smaller straight-line and average path velocity compared to Cluster 2. Tracks show approximate linear swimming paths with jagged-like patterns.

4.3.2.2 Clustering Results - Sample 2.

In sample 2, 21 sperm swim tracks were observed. Motility parameters used for this clustering were LIN, WOB, and STR.

- Cluster 1: 3 tracks with very low values of LIN, WOB, and STR.
- Cluster 2 (circular): 9 tracks with high values of LIN, WOB, and STR. Tracks show circular swimming path, with small lateral displacement.
- Cluster 3 (linear mean): 9 tracks with high values of STR and smaller LIN and WOB values compared to Cluster 2. Tracks show approximate linear swimming path with jagged-like patterns.

4.4 Discussion

To assess the accuracy of the proposed clustering and classification method, simulated and real human samples were tested. In the simulation testing, sperm swim tracks were generated using a real human sample. The performance of ABC algorithm driven clustering was compared to predefined subsets of motility parameters. In the result, the proposed method has shown the best classification performance. In addition, it was observed that using more parameters did not provide better clusters (shown in Figure 4.5). It was found that clustering on a "good" subset of motility parameters showed better classification results than using a large number of motility parameters. It was also noted that each image used a subset of parameters that was different based on the swimming parameters of each simulated sample (shown in Figure 4.6). The proposed method was able to propose a unique subset of parameters that were used to cluster and classify the sperm cells based on their swimming types. The ABC algorithm driven clustering was also applied to images of human semen samples where the algorithm has also shown high accuracy in clustering and classification based on swimming modes (shown in Figures 4.7 and 4.8). The proposed method, the ABC driven clustering, has shown to be an effective method to identify the sperm cells based on the types of the swimming movements.

4.5 Conclusion

K-means clustering guided by the ABC algorithm (for parameter selection) provided distinct clusters of the different sperm swimming types. The algorithm produced clusters of moving sperm swim types characterized by the presence or absence of sperm head roll.

The ability to categorize sperm swim movement could provide a tool that will lead to a better understanding of the qualities of a semen sample.

CHAPTER 5

ROBUST EXTRACTION OF MOTILITY PARAMETERS FROM TIME-LAPSE SEMEN IMAGES

In this chapter, a method of calculating the motility parameters is presented. The Extended Kalman filter (EKF) uses the track of sperm cell, assuming that the cell exhibit sinusoidal traveling wave behavior. Motility parameters are calculated from the estimates of the EKF. The method is compared to existing motility parameter calculation methods using simulated sperm swim tracks.

5.1 Background

In Computer-Assisted Semen Analysis (CASA) systems, parameters that describe the motility of sperm cell are calculated and provided. The main motility parameters include [1]:

1. VCL: Curvilinear velocity. Velocity of a sperm head moving along the curvilinear path.
2. VSL: Straight-line velocity. Velocity of a sperm head along the straight line path between the first detected point to its last.
3. VAP: Average path velocity. Velocity of a sperm head along its average path [69].
4. ALH: Amplitude of lateral head displacement about its average path.
5. LIN: Linearity of the curvilinear path the sperm head (VSL/VCL).
6. WOB: Wobble. Measure of oscillation of the curilinear path about the average path (VAP/VCL).
7. STR: Straightness. Linearity of the average path of the sperm head (VSL/VAP).
8. MAD: Mean angular displacement. Average angle between successive locations along the curvilinear path the moving sperm head.

An illustration in Figure 5.1 shows visual representation of four motility parameters (VCL, VSL, VAP, and ALH).

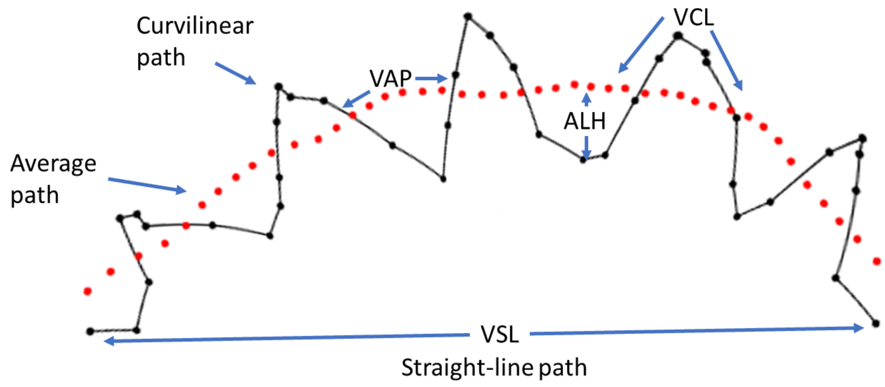


Figure 5.1 Illustration of motility parameters (VCL, VSL, VAP, ALH) measured by CASA systems. (Black) Curvilinear path of sperm head. (Red) Average path of sperm head.

Motility parameters provide important insights on how sperm cells move. The motility parameters can be used to quantify useful parameters to assess the quality of the semen sample. These include percent of motile cells, percent of non-progressive cells, percent of immotile cells, percent of rapid moving cells, and percent of slow moving cells. In this dissertation, we have also provided a use case of motility parameters, which is the classification and clustering algorithm proposed in Chapter 4. In this classification and clustering algorithm, the motility parameters are used to classify sperm cells based on their swimming modes.

While these motility parameters provide valuable information on the semen sample, the parameters calculated from some commercial CASA systems have been known to be inconsistent [70, 71]. The primary factors causing the inconsistencies in the calculation of motility parameters are erroneous estimations of the average path and the frame rate.

In order to calculate the motility parameters accurately and consistently, an accurate estimation of calculation of the average path is needed. Several methods of calculating the average path were proposed in the literature. Fixed length running average (FLRA) and adaptive running average (ADRA) are the most common methods [69, 72]. The 5-point averaging method using FLRA is often suggested for images of 30 FPS and the 11-point averaging method is suggested for images of 60 FPS.

However, these average path calculation algorithms often lead to errors because the frequency of beating in sperm cells varies from a sperm cell to another. To properly calculate the average path, the length of the fixed length averaging method should be approximately the same as the period of the beating of a sperm cell ($period = \frac{1}{frequency}$). The suggested methods (5-point averaging method for 30 FPS or 11-point averaging method for 60 FPS) assume that the frequency of beating in sperm cells is approximately 6 Hz ($\frac{30}{5}$ or $\frac{60}{11}$) or higher. When this assumption is wrong, the estimation of the average path is incorrect.

In Figures 5.2 and 5.3, examples of average path using 5-point averaging method and 11-point averaging method are shown. The blue line shows the original path (curvilinear path) of the sperm head; the red line shows the “average path” calculated from 5-point averaging method; the black line shows the “average path” calculated from 11-point averaging method. The original tracks shown in Figures 5.2 and 5.3 were taken from images with framerate of 15 FPS. Figure 5.2 shows a good example where both averaging methods were able to estimate the average path correctly (shows the overall progressive path without the oscillations). Figure 5.3 shows an example where the 5-point average path calculation algorithm fails. With 5-point smoothing, the oscillations in the sperm track are still present in the average path (red line). Here, the 11-point averaging method seems to provide a reasonable estimation of the average path.

To resolve this type of error, namely the inability to suppress the oscillations, the adaptive running average (ADRA) was proposed and used [72]. In the adaptive running average method, one estimates the period of the oscillation and provides an approximate length of the average filter for average path calculation. However, distortions, noise in observation, and changes in frequency of the oscillations can cause difficulties in estimating the period of the path (causing difficulties in estimating the length by the ADRA). Both

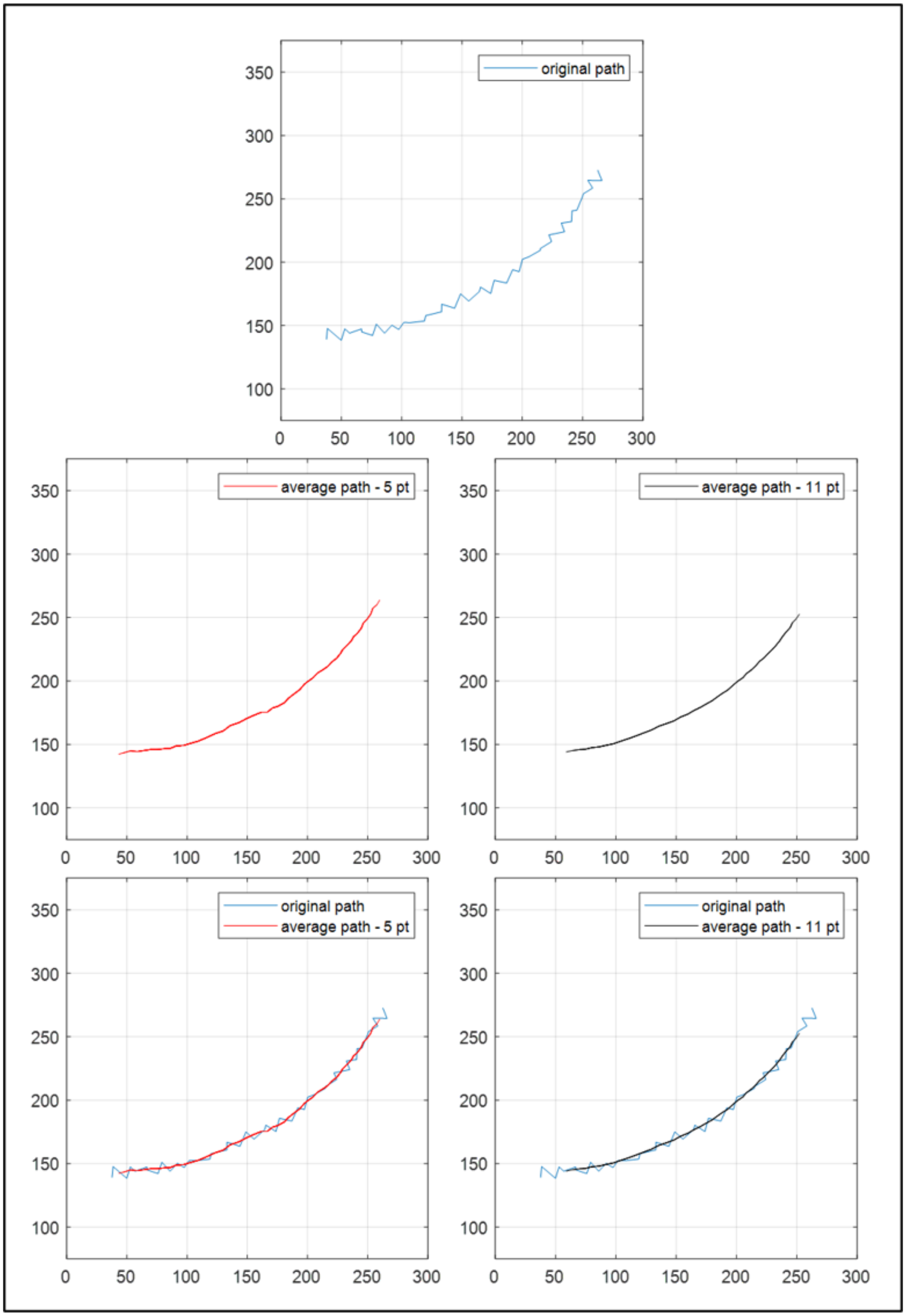


Figure 5.2 Example 1 of 5-point and 11-point average path calculation methods. (Blue) Original path of a sperm cell observed at 15 FPS. (Red) Average path calculated using 5-point averaging method. (Black) Average path calculated using 11-point averaging method.

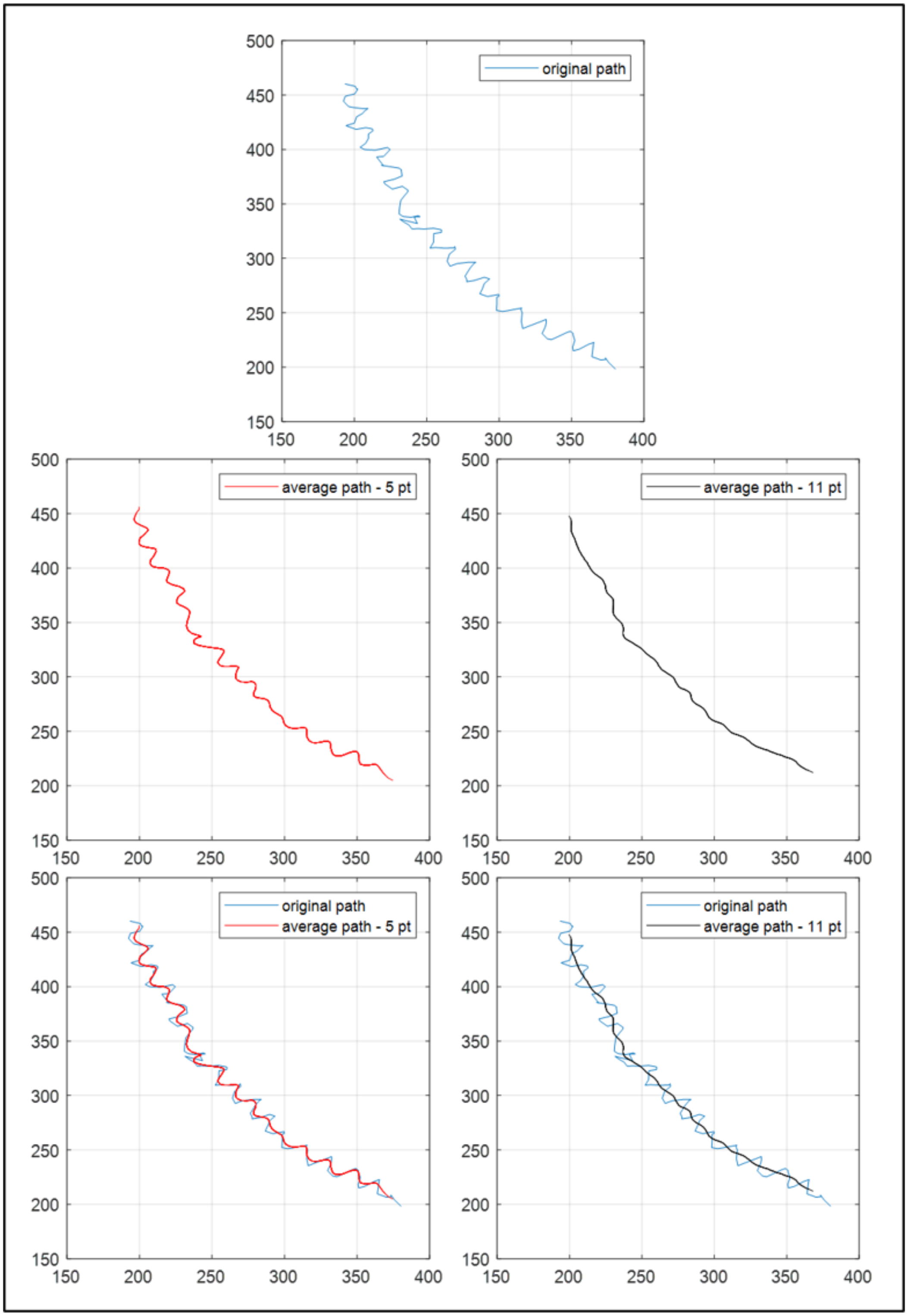


Figure 5.3 Example 2 of 5-point and 11-point average path calculation methods. (Blue) Original path of a sperm cell observed at 15 FPS. (Red) Average path calculated using 5-point averaging method. (Black) Average path calculated using 11-point averaging method.

FLRA and ADRA may therefore provide incorrect estimations of the average path under some circumstances.

The incorrect estimation of the average path affects the value of average path velocity (VAP), amplitude of lateral head displacement (ALH), wobble (WOB), and straightness (STR). The average tracks shown in Figure 5.3 for 5-point averaging and 11-point averaging methods were different; the estimated average path calculated using 5-point averaging was much longer than the average path calculated using the 11-point averaging. This difference results in an overestimate of VAP when VAP is calculated through the using 5-point averaging method. WOB is a ratio between VAP and VCL and STR is a ratio between VSL and VAP. When the value of VAP is incorrect, the values of WOB and STR are also incorrect. The value of ALH, amplitude of the lateral head displacement, is calculated by observing the differences between the curvilinear path and the average path. When ALH was calculated through the average path calculated using the 5-point averaging method (in example shown in Figure 5.3), the value of ALH was much smaller than the true amplitude of lateral head displacement (the height of the oscillations in the sperm track) of the original track (oscillations are still present in the average path).

Another source of estimation error is the differences in frame rates, which causes the values of motility parameters VCL and MAD to be inconsistent. In Figure 5.4, a sinusoid with a frequency of 1 Hz is shown. The top plot shows a sinusoid sampled at 5 Hz (FPS) and the bottom plot shows a sinusoid sampled at 100 Hz (FPS). To connect the discrete set of points, linear interpolation has been used to estimate the curve of the sinusoid. The curve obtained from a sinusoid sampled at 5 FPS is shorter than the curve obtained from a sinusoid sampled at 100 FPS. As a result, the curvilinear velocity (VCL) calculated from 5 FPS data will be smaller than VCL calculated from 100 FPS data. In addition, the mean angular displacement (MAD) is much larger for the curve observed at 5 FPS compared to the curve observed at 100 FPS, as the direction of change is going to be much larger for curves observed at the lower sampling rate.

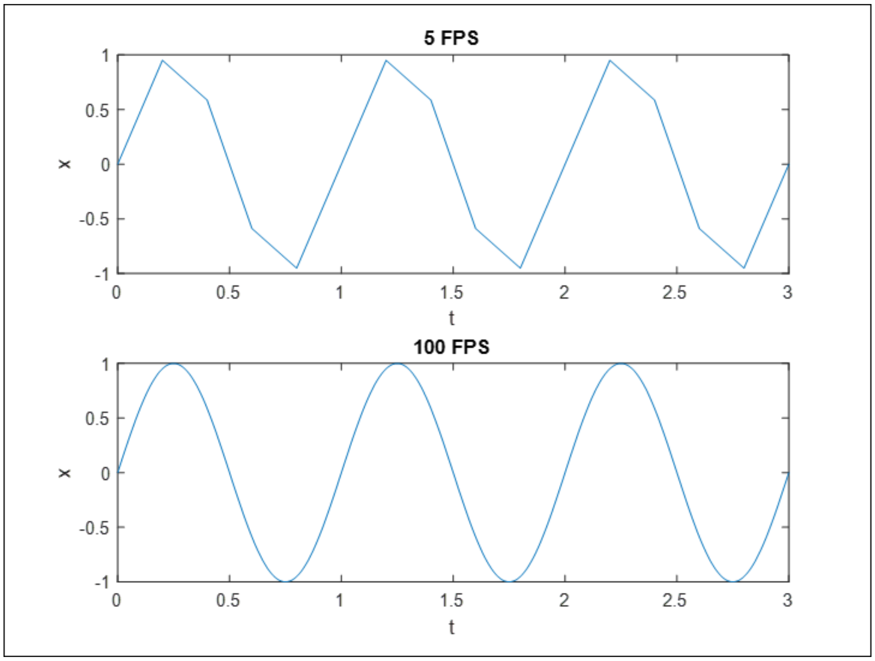


Figure 5.4 Sinusoids with frequency of 1 Hz captured at two different sampling rate. (Top) A sinusoid captured at 5 FPS (5 Hz). (Bottom) A Sinusoid captured at 100 FPS (100 Hz).

To address the difficulties of existing methods for path estimation, a new calculation method for the motility parameter is presented in this chapter. The swim path of a sperm cell is assumed to be a "traveling sinusoid." The "traveling sinusoid" model approximates the track as a forward-progressive moving cell (approximately linear) with oscillations, which are assumed to be sinusoidal. An Extended Kalman filter (EKF) is used to estimate the average path, velocity along the average path, frequency, and amplitude of the traveling sinusoid. This new proposed algorithm can calculate the 8 different motility parameters with accuracy and consistency. The new EKF-based method approach is compared to existing tracking and average path estimation algorithms and is shown to outperform them.

5.2 Methods

In this section, existing methods of tracking and tracking using the "traveling sinusoid" model are explained. In addition, a method of extracting motility parameters from the estimated track using the "traveling sinusoid" model is presented.

5.2.1 Kalman Filter (KF)

The state-space model of a linear time invariant discrete-time system is

$$x(k+1) = Fx(k) + Gu(k) + v(k), \tag{5.1}$$

where $x(k)$ is an n-dimensional state vector, $u(k)$ is an n-dimensional input vector, and $v(k)$ is a zero-mean white Gaussian process noise. F is state-transition matrix and G is defined to be control-input matrix. The covariance matrix of the process noise is

$$E[v(k)v(k)'] = Q(k). \tag{5.2}$$

where $'$ denotes matrix transpose operation. The measurement available from the system is

$$z(k) = H(k)x(k) + w(k), \tag{5.3}$$

where $z(k)$ is an m-dimensional observation vector and $w(k)$ is assumed to be a zero-mean white Gaussian observation noise. H is the observation matrix. The covariance matrix of the observation noise is

$$E[w(k)w(k)'] = R(k). \tag{5.4}$$

In most target tracking scenarios, the states are the position, velocity, and acceleration. If one can assume a linear kinetic target model, the Kalman filter can be used to effectively estimate the "states" $\hat{x}$ of a target. The Kalman filter is the optimal minimum mean square error (MMSE) state estimator for linear kinematic models [73]. The state prediction (error) covariance matrix of the Kalman Filter is denoted here as P. A flowchart of a single iteration of the Kalman filter is shown in Figure 5.5.

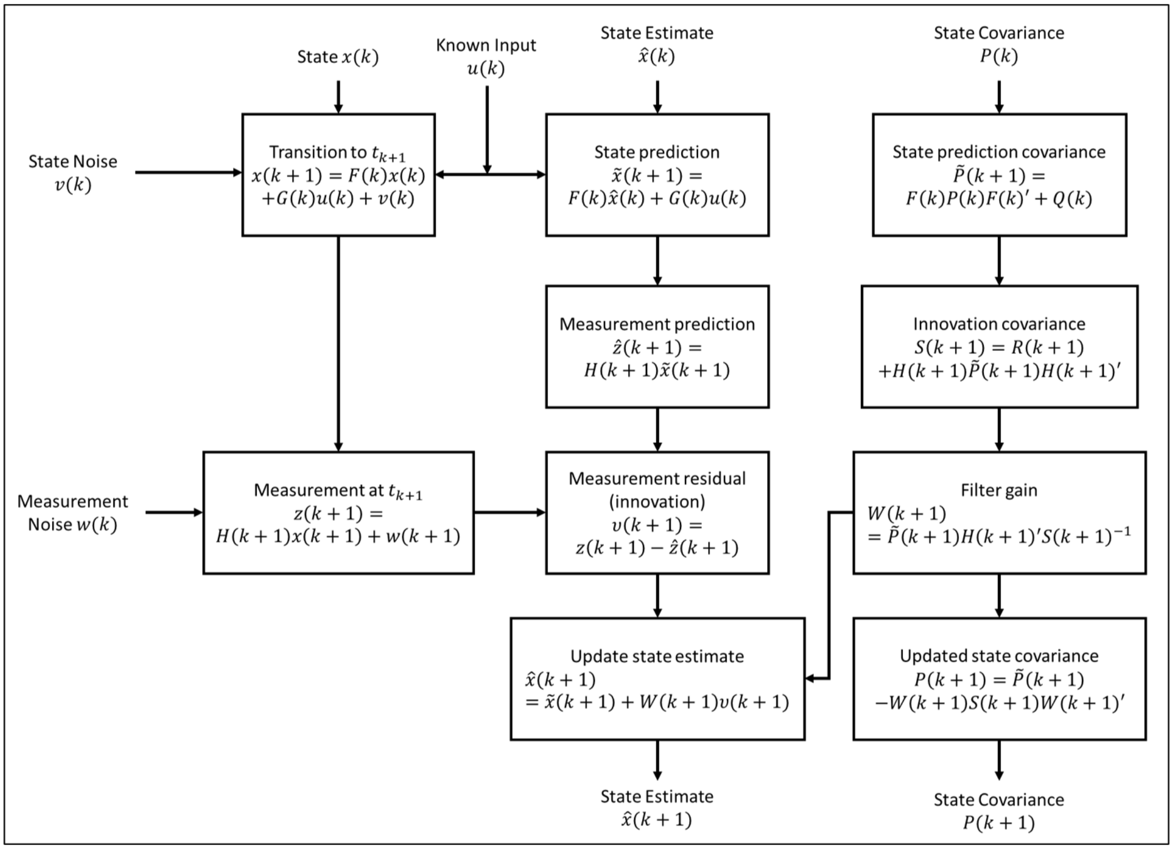

State $x(k)$
Known Input $u(k)$
State Estimate $\hat{x}(k)$
State Covariance $P(k)$
State Noise $v(k)$
Transition to t_{k+1} $x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$
State prediction $\tilde{x}(k+1) = F(k)\hat{x}(k) + G(k)u(k)$
State prediction covariance $\tilde{P}(k+1) = F(k)P(k)F(k)' + Q(k)$
Measurement prediction $\hat{z}(k+1) = H(k+1)\tilde{x}(k+1)$
Innovation covariance $S(k+1) = R(k+1) + H(k+1)\tilde{P}(k+1)H(k+1)'$
Measurement Noise $w(k)$
Measurement at t_{k+1} $z(k+1) = H(k+1)x(k+1) + w(k+1)$
Measurement residual (innovation) $\upsilon(k+1) = z(k+1) - \hat{z}(k+1)$
Filter gain $W(k+1) = \tilde{P}(k+1)H(k+1)'S(k+1)^{-1}$
Update state estimate $\hat{x}(k+1) = \tilde{x}(k+1) + W(k+1)\upsilon(k+1)$
Updated state covariance $P(k+1) = \tilde{P}(k+1) - W(k+1)S(k+1)W(k+1)'$
State Estimate $\hat{x}(k+1)$
State Covariance $P(k+1)$

Figure 5.5 Flowchart of a single iteration of a Kalman Filter.

One can divide the single step of the Kalman filter into two major steps: (1) prediction and (2) update. In the prediction step, the estimated state of a target is calculated as

$$\tilde{x}(k+1) = F(k)\hat{x}(k) + G(k)u(k). \tag{5.5}$$

At this stage, the estimated state $\tilde{x}(k+1)$ is a state estimate calculated from the previous estimate without taking the consideration of the observation $z(k)$. This estimated state is often denoted as $\hat{x}(k+1|k)$ to state that this is the intermediate estimate before the state estimate update stage.

From the estimated state $\tilde{x}(k+1)$, the estimate of the observation is calculated

$$\hat{z}(k+1) = H(k+1)\tilde{x}(k+1). \tag{5.6}$$

The state prediction (error) covariance matrix is estimated as

$$\tilde{P}(k+1) = F(k)P(k)F(k)' + Q(k). \tag{5.7}$$

where $\tilde{P}(k+1)$ is the state prediction covariance matrix without taking account of the observation $z(k)$. It is also often denoted as $P(k+1|k)$.

In the update step, the observation $z(k+1)$ is used to update the state estimate. The innovation $\upsilon(k+1)$, or measurement residual, is the difference between the observation and the estimated observation

$$\upsilon(k+1) = z(k+1) - \hat{z}(k+1). \tag{5.8}$$

The updated state estimate is given by

$$\hat{x}(k+1) = \tilde{x}(k+1) + W(k+1)\upsilon(k+1), \tag{5.9}$$

where $W(k+1)$ is the Kalman gain used to update the state estimate. The Kalman gain $W(k+1)$ is defined as

$$W(k+1) = \tilde{P}(k+1)H(k+1)'S(k+1)^{-1}, \tag{5.10}$$

where $S(k+1)$ is the innovation covariance defined as

$$S(k+1) = R(k+1) + H(k+1)\tilde{P}(k+1)H(k+1)'. \tag{5.11}$$

Along with the state estimate, the updated state prediction covariance matrix is given by

$$P(k+1) = \tilde{P}(k+1) - W(k+1)S(k+1)W(k+1)'. \tag{5.12}$$

5.2.2 Linear Motion Models for Tracking Sperm Cells

Several models for the movement of sperm and other cells were proposed. Among them, the Brownian motion model (random walk) and constant velocity (with white noise acceleration, WNA) models are the most popular. In this section, we use the subscripts k and $k+1$ to denote that the values are of time k and $k+1$, respectively.

5.2.2.1 Brownian Motion Model. The Brownian motion model is a common model for cell tracking and simulation of cell motion [28, 74]. The 2-dimensional Brownian motion model assumes that the displacement of a target v_k (2×1 process noise) is a zero-mean white sequence.

$$X_k = [x_k \; y_k]' \tag{5.13}$$

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = FX_k + v_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + v_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + v_k \tag{5.14}$$

There is no input $v(k)$ ($v(k) = 0$). The covariance matrix of the process noise Q is

$$Q = E[v_k v_k'] = \begin{bmatrix} \sigma_{v_x}^2 & 0 \\ 0 & \sigma_{v_y}^2 \end{bmatrix}, \tag{5.15}$$

where the process noise in the horizontal direction x and y are independent of each other. The variances of process noise are $\sigma_{v_x}^2$ and $\sigma_{v_y}^2$ for horizontal and vertical positions, respectively. The observation Z_k is defined to be

$$Z_k = \begin{bmatrix} z_{k_1} \\ z_{k_2} \end{bmatrix} = HX_k + w_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + w_k. \tag{5.16}$$

Here, z_{k_1} and z_{k_2} are horizontal and vertical position of the observation, respectively, and w_k is the 2×1 observation noise. The observation noise covariance matrix R is defined as

$$R = E[w_k w_k'] = \begin{bmatrix} \sigma_{w_1}^2 & 0 \\ 0 & \sigma_{w_2}^2 \end{bmatrix}, \tag{5.17}$$

where $\sigma_{w_1}^2$ and $\sigma_{w_2}^2$ is the noise variance for the observation (z_{k_1} and z_{k_2}). The observation noise in the horizontal position is assumed independent of the observation noise in the vertical position.

5.2.2.2 Discrete White Noise Acceleration (DWNA) Model. Sperm cells and aquatic animals move by passing waves of lateral displacement down the body. As a result, the sperm cells and aquatic animals show sinusoidal-like movement as they propel themselves forward. To track these types of movement, constant velocity (also known as white noise acceleration (WNA)) models have been used in the literature [9, 10, 75, 76]. The discrete white noise acceleration (DWNA) model is often suggested for discrete time data. In the 2-dimensional DWNA model, the state vector X_k holds the positions and velocities along the horizontal and vertical directions,

$$X_k = [x_k \; \dot{x}_k \; y_k \; \dot{x}_k]'. \tag{5.18}$$

The model assumes constant velocity, where the process noise is zero-mean white acceleration sequence. The change in the state X_k is defined as

$$\begin{aligned} X_{k+1} = \begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ y_{k+1} \\ \dot{y}_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \dot{x}_k T \\ \dot{x}_k \\ y_k + \dot{y}_k T \\ \dot{y}_k \end{bmatrix} + v_k = FX_k + \begin{bmatrix} \Gamma v_{k_1} \\ \Gamma v_{k_2} \end{bmatrix} = \\ \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix} + \begin{bmatrix} v_{k_1} T^2/2 \\ v_{k_1} T \\ v_{k_2} T^2/2 \\ v_{k_2} T \end{bmatrix} \end{aligned} \tag{5.19}$$

where the process noise covariance matrix Q is defined as

$$Q = E[v_k v_k'] = \begin{bmatrix} \Gamma \sigma_{v_x}^2 \Gamma' & 0_{2,2} \\ 0_{2,2} & \Gamma \sigma_{v_y}^2 \Gamma' \end{bmatrix}, \; \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \tag{5.20}$$

There is no input $v(k)$ ($v(k) = 0$). The observation Z_k is defined as

$$Z_k = \begin{bmatrix} z_{k_1} \\ z_{k_2} \end{bmatrix} = HX_k + w_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix} + w_k \tag{5.21}$$

where the observation noise w_k is a 2×1 vector with covariance matrix R is defined as

$$R = E[w_k w_k'] = \begin{bmatrix} \sigma_{w_1}^2 & 0 \\ 0 & \sigma_{w_2}^2 \end{bmatrix}. \tag{5.22}$$

5.2.3 Extended Kalman Filter (EKF)

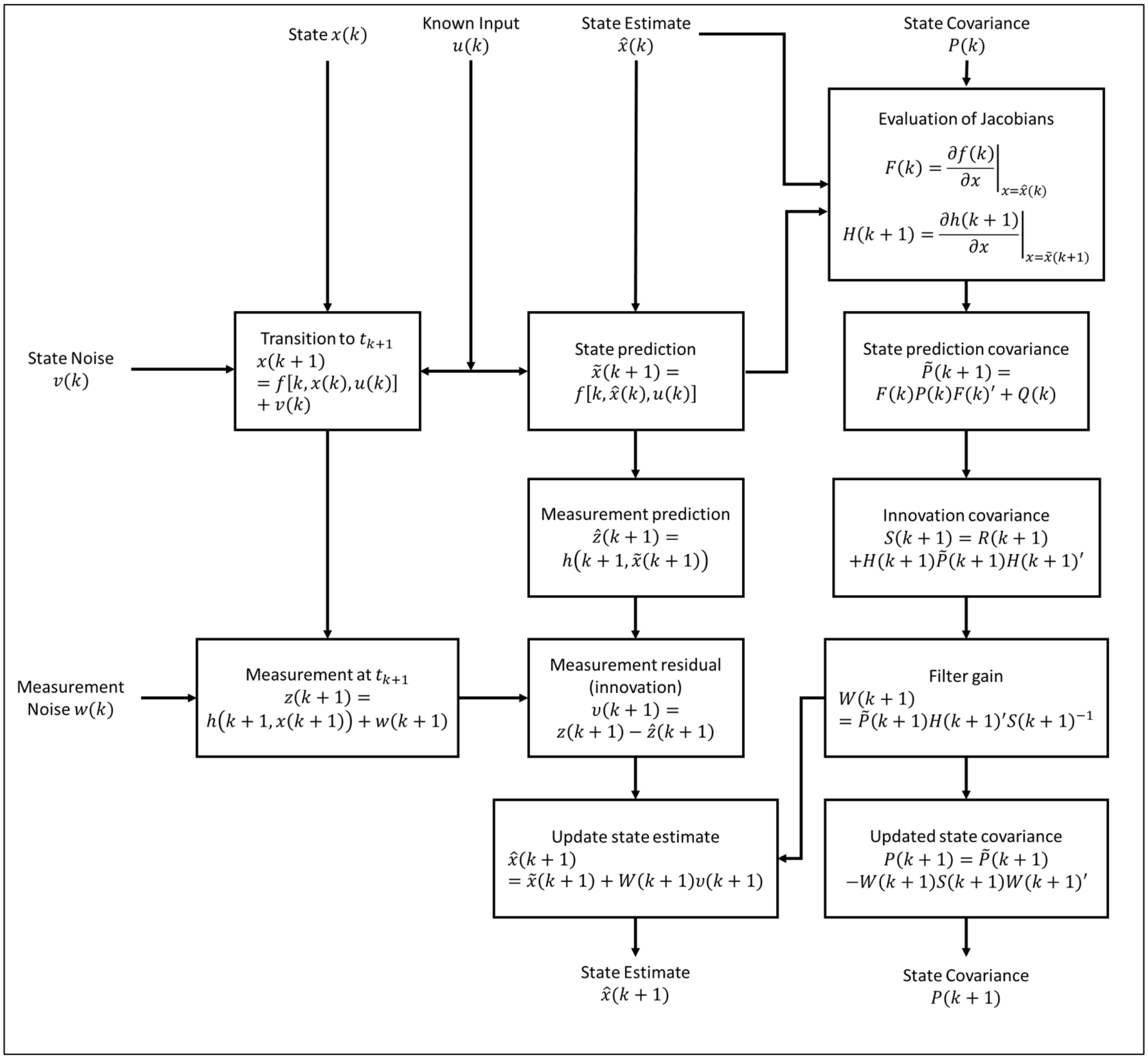
The kinematic model of an object will not always be linear. In such case, one can sometimes write the state equation as

$$x(k+1) = f[k, x(k), u(k)] + v(k), \tag{5.23}$$

where $f[k, x(k), u(k)]$ is an assumed function of $k, x(k), u(k)$, differentiable in x and u. The observation (measurement) equation can be written as

$$z(k+1) = h[k+1, x(k+1)] + w(k), \tag{5.24}$$

where $h[k+1, x(k+1)]$ is a function of $k+1$ and $x(k+1)$, assumed differentiable in x. When the state equation and the observation equation are non-linear, one can sometimes linearize the state equation and the observation equation at each time step k. This approach leads to the Extended Kalman filter (EKF). One iteration of the EKF is shown in Figure 5.6.



State $x(k)$
Known Input $u(k)$
State Estimate $\hat{x}(k)$
State Covariance $P(k)$
Evaluation of Jacobians
$F(k) = \frac{\partial f(k)}{\partial x}\Big|_{x=\hat{x}(k)}$
$H(k+1) = \frac{\partial h(k+1)}{\partial x}\Big|_{x=\tilde{x}(k+1)}$
State Noise $v(k)$
Transition to t_{k+1} $x(k+1) = f[k, x(k), u(k)] + v(k)$
State prediction $\tilde{x}(k+1) = f[k, \hat{x}(k), u(k)]$
State prediction covariance $\tilde{P}(k+1) = F(k)P(k)F(k)' + Q(k)$
Measurement prediction $\hat{z}(k+1) = h(k+1, \tilde{x}(k+1))$
Innovation covariance $S(k+1) = R(k+1) + H(k+1)\tilde{P}(k+1)H(k+1)'$
Measurement Noise $w(k)$
Measurement at t_{k+1} $z(k+1) = h(k+1, x(k+1)) + w(k+1)$
Measurement residual (innovation) $\upsilon(k+1) = z(k+1) - \hat{z}(k+1)$
Filter gain $W(k+1) = \tilde{P}(k+1)H(k+1)'S(k+1)^{-1}$
Update state estimate $\hat{x}(k+1) = \tilde{x}(k+1) + W(k+1)\upsilon(k+1)$
Updated state covariance $P(k+1) = \tilde{P}(k+1) - W(k+1)S(k+1)W(k+1)'$
State Estimate $\hat{x}(k+1)$
State Covariance $P(k+1)$

Figure 5.6 Flowchart of a single iteration of an Extended Kalman Filter.

The operation of the EKF is similar to the process of the KF. The difference is that the state prediction covariance ($\tilde{P}$ and P, Equations (5.7) and (5.12)), filter gain (W, Equation 5.10), and innovation covariance (S, Equation (5.11)) are updated with Jacobians of the state equation F and the observation equation H. The Jacobians of the state equation and the observation equation are given as

$$F(k) = \frac{\partial f(k)}{\partial x}|_{x=\hat{x}(k)},\ H(k+1) = \frac{\partial h(k+1)}{\partial x}|_{x=\tilde{x}(k+1)}. \tag{5.25}$$

Many of the theoretical guarantees developed for the Kalman filter do not extend automatically to the EKF, or may require various restrictions on the functions $f(\cdot)$ and $h(\cdot)$.

5.2.3.1 Motion Model. Many studies, through observation and simulation, have shown that aquatic animals and sperm cells, which propel themselves forward by passing waves of lateral head displacement, exhibit sinusoidal-like paths [29, 36, 38, 39, 72, 75–82]. Instead of using the Brownian motion model or the DWNA model for tracking sperm cells, the proper model for tracking of sperm cells should therefore be a "traveling sinusoid." In this model, there are eight (8) stages, namely

$$X_k = [A_k\ \phi_k\ \omega_k\ \theta_k\ r_k\ \dot{r}_k\ l_k\ \dot{l}_k]'. \tag{5.26}$$

Here,

- A is the amplitude of the wave,
- ϕ is the angle of the sinusoid
- ω is the frequency of the sinusoid
- θ is the angle of rotation
- r and l are the horizontal and vertical positions of the average path
- $\dot{r}$ and $\dot{l}$ are the horizontal and vertical velocity along the average path

The observation Z_k is assumed to be a function of the states X_k, of the form

$$
\begin{aligned}
Z_k = \begin{bmatrix} z_{k_1} \\ z_{k_2} \end{bmatrix} = h(X_k) + w_k = \begin{bmatrix} r_k \\ l_k \end{bmatrix} + \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix} \begin{bmatrix} 0 \\ A_k \sin(\phi_k) \end{bmatrix} + w_k \\
= \begin{bmatrix} r_k - A_k \sin(\phi_k)\sin\theta_k \\ l_k + A_k \sin(\phi_k)\cos\theta_k \end{bmatrix} + w_k
\end{aligned}
\quad . \qquad (5.27)
$$

In this model, the sinusoid $A_k \sin\phi_k$ is rotated by angle θ_k and added to the position r_k and l_k. The state equation is

$$
\begin{aligned}
X_{k+1} = \begin{bmatrix} A_{k+1} \\ \phi_{k+1} \\ \omega_{k+1} \\ \theta_{k+1} \\ r_{k+1} \\ \dot{r}_{k+1} \\ l_{k+1} \\ \dot{l}_{k+1} \end{bmatrix} = FX_k + v_k = \begin{bmatrix} A_k \\ \phi_k + \omega_k T \\ \omega_k \\ \theta_k \\ r_k + \dot{r}_k T \\ \dot{r}_k \\ l_k + \dot{l}_k T \\ \dot{l}_k \end{bmatrix} + \begin{bmatrix} v_{k_A} \\ \Gamma v_{k_\phi} \\ v_{k_\theta} \\ \Gamma v_{k_r} \\ \Gamma v_{k_l} \end{bmatrix} \\
= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_k \\ \phi_k \\ \omega_k \\ \theta_k \\ r_k \\ \dot{r}_k \\ l_k \\ \dot{l}_k \end{bmatrix} + \begin{bmatrix} v_{k_A} \\ v_{k_\phi} T^2/2 \\ v_{k_\phi} T \\ v_{k_\theta} \\ v_{k_r} T^2/2 \\ v_{k_r} T \\ v_{k_l} T^2/2 \\ v_{k_l} T \end{bmatrix} , \ \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}
\end{aligned}
\quad . \qquad (5.28)
$$

In this state equation, the amplitude A_k and the rotation angle θ_k follow the Brownian motion model, and the angle of sinusoid ϕ and position r, l follow the DWNA model (constant velocity and frequency). The state noise covariance matrix Q is defined as

$$Q = E[v_k v_k'] = \begin{bmatrix} \sigma_A^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\phi^2 T^4/4 & \sigma_\phi^2 T^3/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\phi^2 T^3/2 & \sigma_\phi^2 T^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\theta^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_r^2 T^4/4 & \sigma_r^2 T^3/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_r^2 T^3/2 & \sigma_r^2 T^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_l^2 T^4/4 & \sigma_l^2 T^3/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_l^2 T^3/2 & \sigma_l^2 T^2 \end{bmatrix}, \tag{5.29}$$

The model assumes that the amplitude A, the rotation angle θ, the frequency ω, and the velocity $(\dot{r}, \dot{l})$ are independent of each other and of any combination of the other variables. The amplitude A and rotation angle θ follow state noise covariance of the Brownian motion model. The frequency ω and velocity $(\dot{r}, \dot{l})$ follow state noise covariance of the DWNA model. The model assumes the observation noise in the horizontal position z_{k_1} is independent of the observation noise in the vertical position z_{k_2}.

$$R = E[w_k w_k'] = \begin{bmatrix} \sigma_{w_1}^2 & 0 \\ 0 & \sigma_{w_2}^2 \end{bmatrix} \tag{5.30}$$

The state equation of the model is linear (Equations (5.23) and (5.28), $f[k, x(k), u(k)] = F(k)x(k)$). Therefore, Jacobian of $f[k, x(k), u(k)]$ is matrix F in Equation (5.28). The observation equation is nonlinear. The Jacobian of $h[k, x(k)]$ is

$$h'(X_k) = \frac{\partial h(X_k)}{\partial X_k} = H_k = \begin{bmatrix} -\sin(\phi_k)\sin\theta_k & \sin(\phi_k)\cos\theta_k \\ -A_k\cos(\phi_k)\sin\theta_k & A_k\cos(\phi_k)\cos\theta_k \\ 0 & 0 \\ -A_k\sin(\phi_k)\cos\theta_k & -A_k\sin(\phi_k)\sin\theta_k \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}' . \qquad (5.31)$$

In this chapter, the Extended Kalman filter assuming "traveling sinusoid" model is referred as the traveling sinusoid EKF. Accordingly, the Kalman filter assuming DWNA model is referred as the DWNA KF and the Kalman filter assuming the Brownian motion model is referred as the Brownian KF.

5.2.4 Advantages of using "Traveling Sinusoid" EKF Model

In this section, the advantages of using the "traveling sinusoid" model are presented. In an image of a semen sample, a large number of sperm cells can be observed. As the sperm cells swim, a sperm cell can collide with the other cells. Figure 5.7 shows an example where two different particles (cells) moving as "traveling sinusoids." Figure 5.7a shows a result of tracking using traveling sinusoid EKF, Figure 5.7b shows a result of tracking using DWNA KF, Figure 5.7c shows a result of tracking using Brownian motion KF, and Figure 5.7d shows the mean error (distance) between the ground truth tracks and the estimated tracks.

In multi-object tracking, a decision process is required to choose which observation or observations belong to each track. Among the methods to accomplish this association

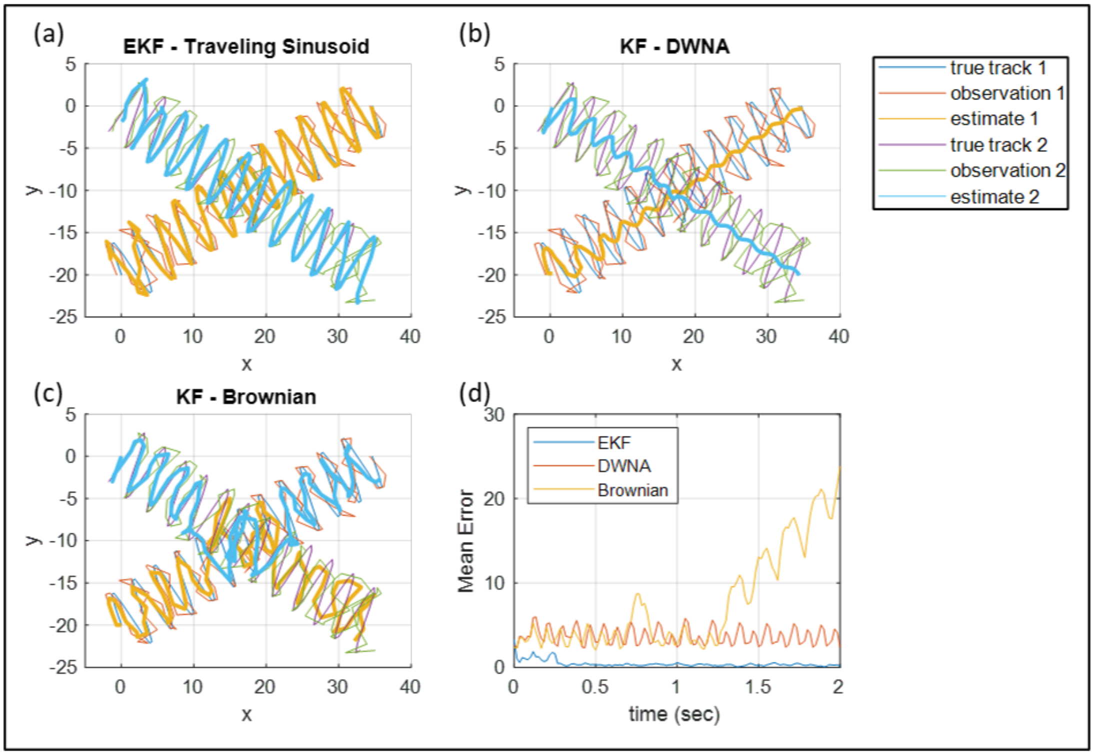


Figure 5.7 An example of tracking two traveling sinusoids with different types of Kalman filters. (a) “Traveling Sinusoid” EKF tracking. (b) DWNA KF tracking. (c) Brownian motion KF tracking. (d) Mean error between ground truth tracks and estimated tracks.

task, the nearest-neighbor (NN) is one of the simplest; the observation closest to a track is used to update the KF or the EKF. For the DWNA KF and traveling sinusoid EKF, track mismatches are less likely to occur, because the estimates are updated based on the expected direction of travel (velocity). However, the Brownian motion KF does not take into consideration the direction of movement of the target. The estimates of the Brownian model, therefore, have a higher likelihood of mismatch between the observation and the tracks.

Figure 5.7 shows a case of mismatch of tracks occurring for the Brownian motion KF. The mean errors between the estimated track and the ground truth for the DWNA KF and the traveling sinusoid EKF remain at a similar level. The mean error between the estimated track and the ground truth for the Brownian KF, on the other hand, shows a gradual increase at the index of approximately 80. This increase in mean error is caused by a track mismatch that occurs when the two tracks collide with each other. Due to the phenomenon of track mismatch, the use of Brownian KF should be avoided for sperm tracking.

The DWNA KF models the track as a target with constant velocity. This assumption causes the estimated track to be approximately linear. For sperm cells that move as "traveling sinusoids," the oscillations are considered to be part of the noise. However, the oscillations are not caused by noise and as a result, the DWNA KF loses valuable information about the sperm track. The "traveling sinusoid" EKF retains the oscillations in the sperm track model, unlike the DWNA KF. This difference is illustrated in Figure 5.8. On the left hand side, Figure 5.8 shows the result of DWNA KF tracking. On the right side of the Figure 5.8 shows the result of traveling sinusoid EKF. The estimated track of DWNA KF was unable to track the oscillation of the traveling sinusoid, where as the traveling sinusoid EKF retained the oscillation in the original track.

5.2.5 Motility Parameter Extraction from EKF Estimates

There are two main changes in the motility parameter calculations methods (previous motility parameter calculation methods shown in Section 4.2.2). The traveling sinusoid EKF estimates the amplitude of the track oscillation. This value of the amplitude can be directly used as the estimate of ALH (amplitude of lateral head displacement). In addition, the traveling sinusoid EKF estimates the positions r_k and l_k, which are the position of the average path at time k. The availability of this estimate removes the need to use track averaging algorithms such as the FLRA and ADRA. The average path estimated using the EKF will directly impact the calculation of VAP (average path velocity), WOB (wobble), and STR (straightness).

Lastly, the proposed method for track estimation also uses a linear interpolation of the state estimates X_k for values between samples at time steps k and $k + 1$. The observation estimate can be found as $\hat{z}(n) = h(n, \hat{x}(n))$, where n is the time step between time steps k and $k + 1$. The interpolation is done to minimize the differences in the value of VCL

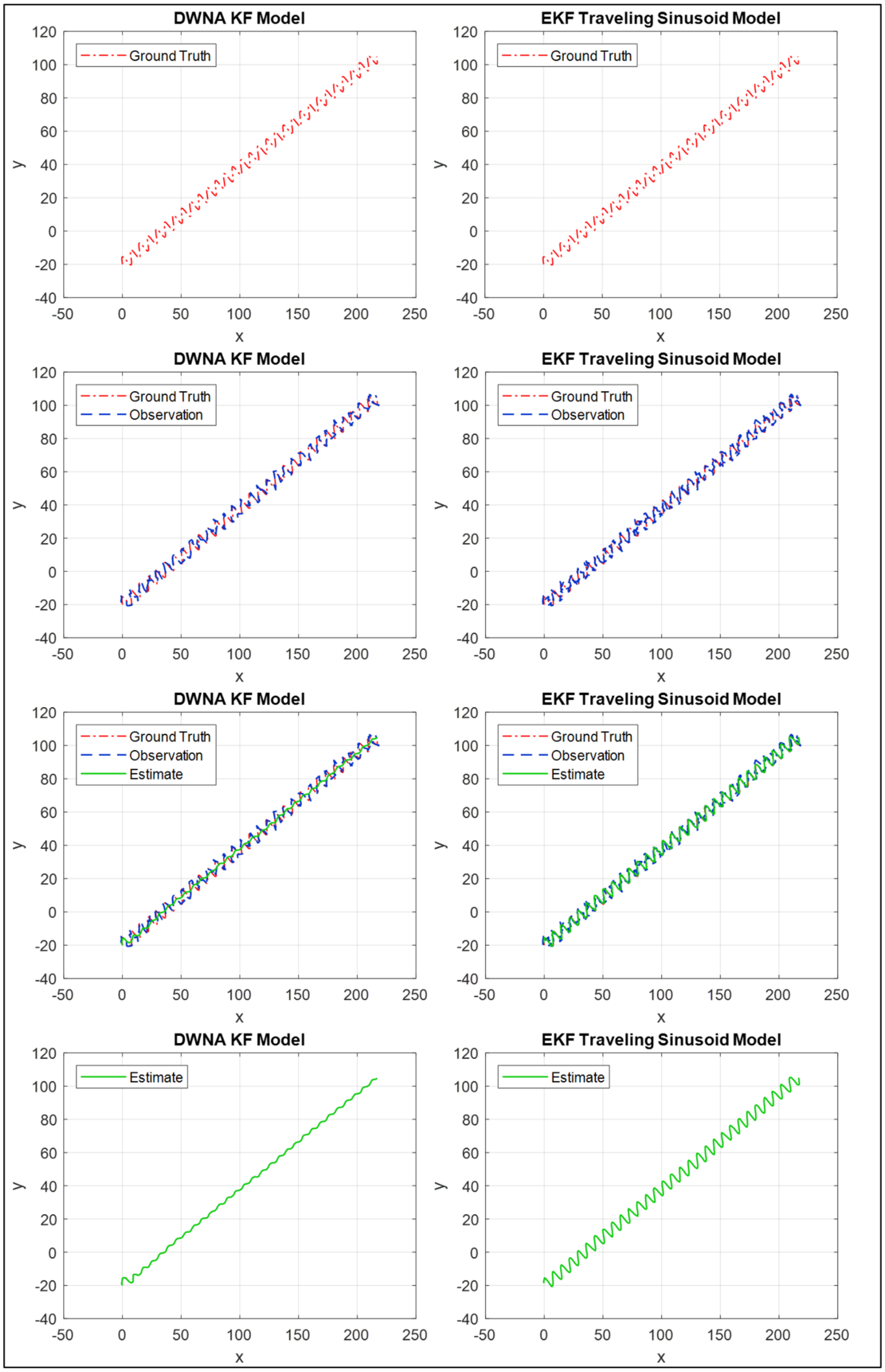


Figure 5.8 An example of tracking traveling sinusoid using DWNA KF and traveling sinusoid EKF. (Red) The ground truth track. (Blue) The observed track. (Green) The estimated track. (Left) DWNA KF. (Right) Traveling sinusoid EKF.

(curvilinear velocity) and MAD (mean angular displacement) for data with different frame rates.

5.3 Test Setup

To assess the performance of the traveling sinusoid EKF for accurate and consistent calculation of motility parameters, four (4) test cases were generated. The four test cases were:

1. "Traveling sinusoid" with observation noise
($X(k+1) = Fx(k)$ and $Z(k) = h(k, x(k)) + w(k)$),
2. Rotating "traveling sinusoid" (similar to circular swim described in Chapter 2) with observation noise
($X(k+1) = Fx(k) + [0\ 0\ 0\ 0.05\pi T\ 0\ 0\ 0\ 0]'$ and $Z(k) = h(k, x(k)) + w(k)$),
3. Rotating "traveling sinusoid" with process noise and observation noise
($X(k+1) = Fx(k) + v(k) + [0\ 0\ 0\ 0.05\pi T\ 0\ 0\ 0\ 0]'$ and $Z(k) = h(k, x(k)) + w(k)$), and
4. Linear mean swimming cell (for the equations of the swim model see Section 2.2.2.1.2) with observation noise ($Z(k) = h(k, x(k)) + w(k)$).

The ground truth track and the observed track used for testing are shown in Figure 5.9. The observation noise is assumed to be zero-mean white noise with a variance of 1 px. The values of parameters used to generate the simulated paths are shown in Tables 5.1 and 5.2. For test cases that involved rotation, the rotation was added to the track with a rate of 0.05π rad/s.

Table 5.1 Initial State Values for Test Cases 1-3

Variable	A	ϕ	ω	θ	r	$\dot{r}$	l	$\dot{l}$
Value	4 px	0 rad	12π rad/s	$\frac{\pi}{6}$ rad	0 px	$50\cos(\frac{\pi}{6})$ px/s	-20 px	$50\sin(\frac{\pi}{6})$ px/s

A total of six (6) different tracking and track averaging methods were tested in their ability to provide accurate and consistent calculations of motility parameters. The six tracking and track averaging methods are:

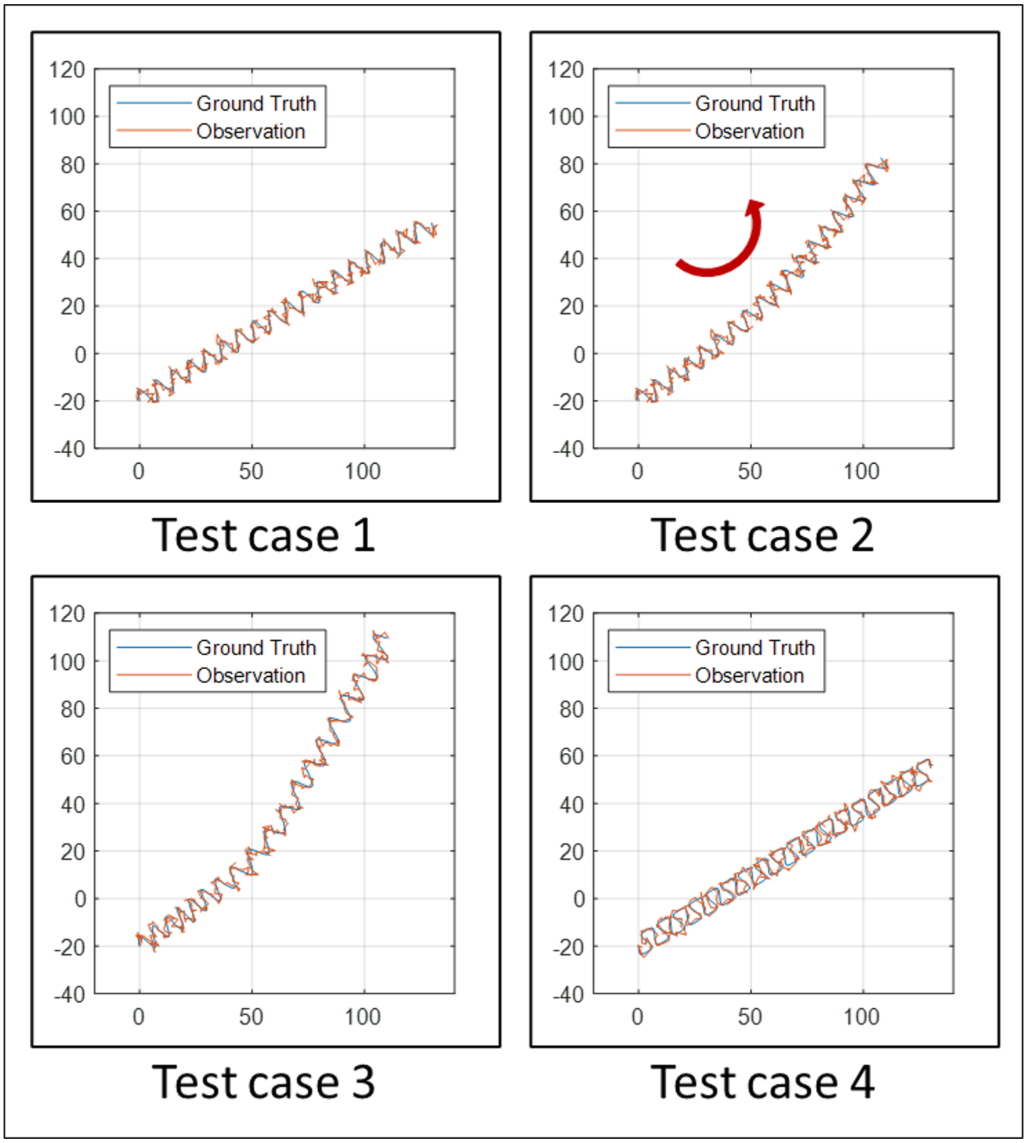


Figure 5.9 Test cases for motility parameter calculation performance assessment. (1) "Traveling sinusoid." (2) Rotating "traveling sinusoid." (3) Noisy rotating "traveling sinusoid. (4) Linear mean swim.

Table 5.2 Parameter Values Used to Generate Test Case 4 (Linear Mean Swimming Path)

Variable	r_v	r_h	f_l	θ_r	V	A_{har}
Value	4 px	8 px	12π rad/s	$\frac{\pi}{6}$ rad	50 px/s	0.1

1. Raw measurements (denoted as nearest-neighbor (NN) in the results) with 5-point averaging,
2. Raw measurements (NN) with 11-point averaging,
3. DWNA KF tracking with 5-point averaging,

4. DWNA KF tracking with 11-point averaging,
5. “Traveling sinusoid” EKF tracking, and
6. “Traveling sinusoid” EKF tracking with interpolated states.

Tracking and track averaging methods are tested were 30, 40, 60, and 120 FPS. The ground truth motility parameters were calculated from the ground truth track (no observation noise) at 120 FPS.

5.4 Results

The motility parameters calculated from the six tracking and track averaging methods are shown in Figures 5.10, 5.12, 5.14, and 5.16. The percent error between the calculated values of the motility parameters and the true values of the motility parameters (calculated from the ground truth tracks at 120 FPS) for the four test cases are shown in Figures 5.11, 5.13, 5.15, and 5.17. Percent error is defined as

$$\text{Percent Error} = \frac{|Calculated - Truth|}{Truth} \times 100\%. \tag{5.32}$$

A good tracking and track averaging methods will (1) have small error between the ground truth and the calculated motility parameter values and (2) have small or no change in motility parameter values for different frame rates.

Figure 5.10 shows the values of motility parameters (VCL, VSL, LIN, ALH, VAP, WOB, STR, and MAD) for the six tracking and track averaging algorithms for test case 1 at frame rates 30, 40, 60, and 120. Figure 5.11 shows the percent error between the ground truth motility parameters to calculated motility parameters from the six tracking and tracking algorithms. Observations:

- For VCL, one can observe that the NN tracking algorithms show change in the values of VCL with the frame rate. Specifically, the value of VCL increases as frame rate increases. For the DWNA KF tracking, one can observe VCL values close to 50, which is far from the true value of VCL (close to 100). On the other hand, the VCL values calculated using the estimated track from traveling sinusoid EKF are very close to the ground truth VCL values (small percent error).

- All the methods were able to accurately calculate the values of VSL (close to the ground truth value). However, other than the traveling sinusoid EKF with interpolation method, all the other algorithms seem to show small differences in the VSL values calculated at different frame rates.
- The traveling sinusoid EKF was able to provide the most accurate values to the ground truth values of LIN. One can observe that the LIN value for the DWNA KF model was very close to 1. This is due to the model assuming a constant velocity (movement close to a line) where the oscillations in the track are considered to be noise. Similar to VCL, the NN tracking method also showed changes in the calculated values of LIN when the frame rate changed.
- Values of ALH are inconsistent for the NN based tracking and DWNA KF based tracking methods. The differences in the framerate and the averaging method cause differences in the ALH values. In contrast, the value of ALH is close to the ground truth for the traveling sinusoid EKF. The value of ALH calculated from traveling sinusoid EKF is also less affected by frame rate compared to the other tracking and track averaging methods.
- The calculated values of VAP are close to ground truth for DWNA KF and traveling sinusoid based tracking methods. For NN methods, the values of VAP are heavily affected by frame rate. The 11-point averaging method provides results that are close to the ground truth up to the frame rate of 60. The 5-point averaging method provides correct results only for 30 FPS.
- The WOB values calculated from DWNA KF and traveling sinusoid EKF tracking methods are consistent for the frame rates 30, 40, 60, and 120. For NN method the values of WOB change for with the frame rates. The values of WOB for traveling sinusoid EKF methods are closest to the ground truth compared to the other methods.
- The STR values calculated from DWNA KF and traveling sinusoid EKF tracking methods are consistent for the frame rates 30, 40, 60, and 120 with small percent error. For NN methods the values of STR changes with the frame rates.
- THE MAD values calculated from DWNA KF and traveling sinusoid EKF - interpolated tracking methods are consistent for the frame rates 30, 40, 60, and 120, with small percent error. NN tracking and traveling sinusoid EKF without interpolation show large percent error and the value of MAD varies with the frame rate.

Figure 5.12 shows the values of motility parameters (VCL, VSL, LIN, ALH, VAP, WOB, STR, and MAD) for the six tracking and track averaging algorithms for test case

2 at frame rates 30, 40, 60, and 120. Figure 5.13 shows the percent error between the ground truth motility parameters to calculated motility parameters from the six tracking and tracking algorithms. Observations are similar to the observations made for test case 1.

Figure 5.14 shows the values of motility parameters (VCL, VSL, LIN, ALH, VAP, WOB, STR, and MAD) for the six tracking and track averaging algorithms for test case 3 at frame rates 30, 40, 60, and 120. Figure 5.15 shows the percent error between the ground truth motility parameters to calculated motility parameters from the six tracking and tracking algorithms. Observations are similar to the observations made for test case 1 for NN tracking and DWNA tracking methods. Compared to the percent errors observed in test cases 1-2, the percent errors of motility parameters (VCL, LIN, ALH, and WOB) for traveling sinusoid EKF are larger in test case 3.

Figure 5.16 shows the values of motility parameters (VCL, VSL, LIN, ALH, VAP, WOB, STR, and MAD) for the six tracking and track averaging algorithms for test case 4 at frame rates 30, 40, 60, and 120. Figure 5.17 shows the percent error between the ground truth motility parameters to calculated motility parameters from the six tracking and tracking algorithms. Observations are similar to the observations made for test case 1 for NN tracking and DWNA tracking methods. Compared to the percent error observed in test cases 1-3, the percent errors of motility parameters (VCL, LIN, WOB, and MAD) for traveling sinusoid EKF are larger in test case 4.

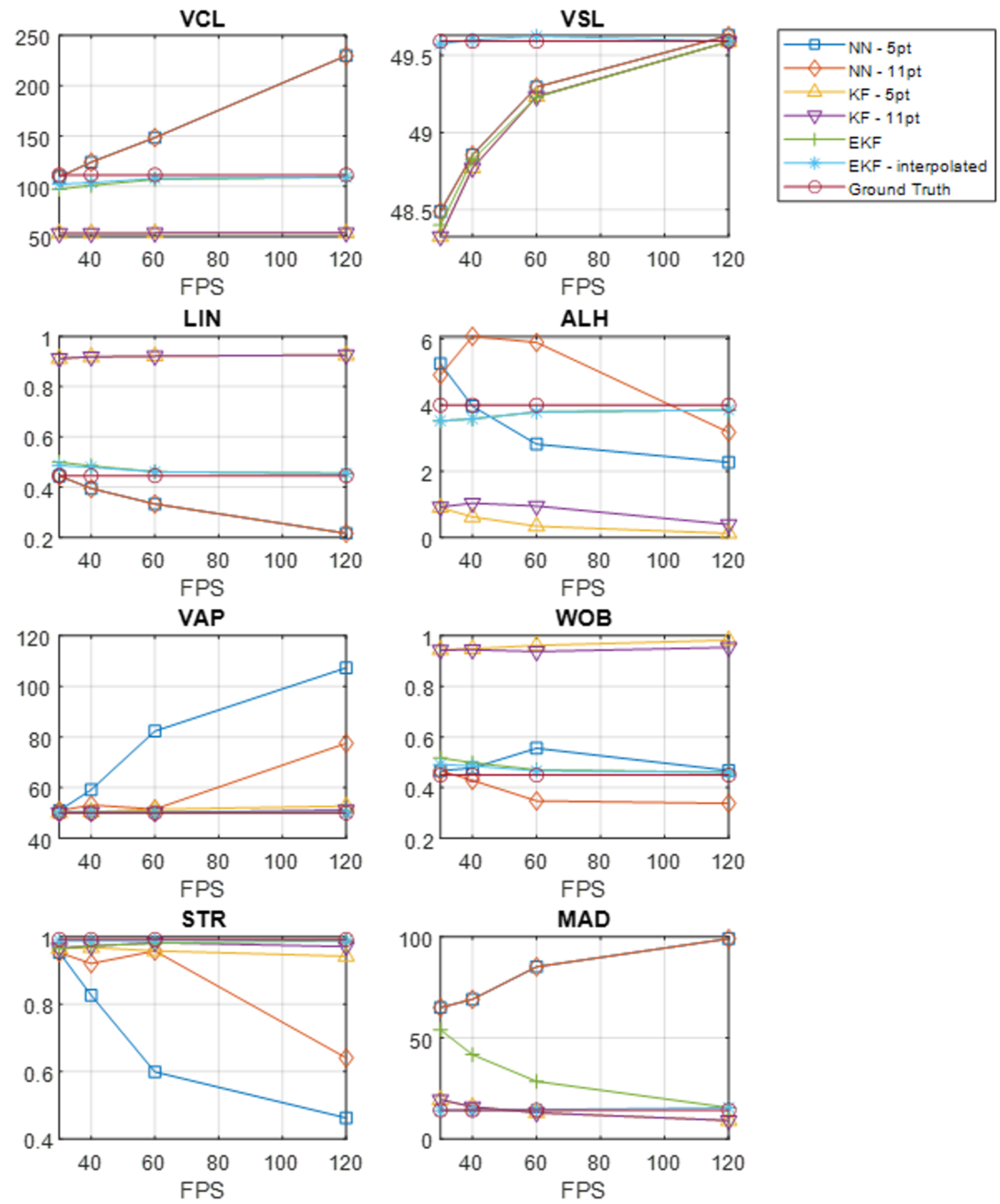


Figure 5.10 Motility parameters calculated from the six tracking and track averaging methods for test 1.

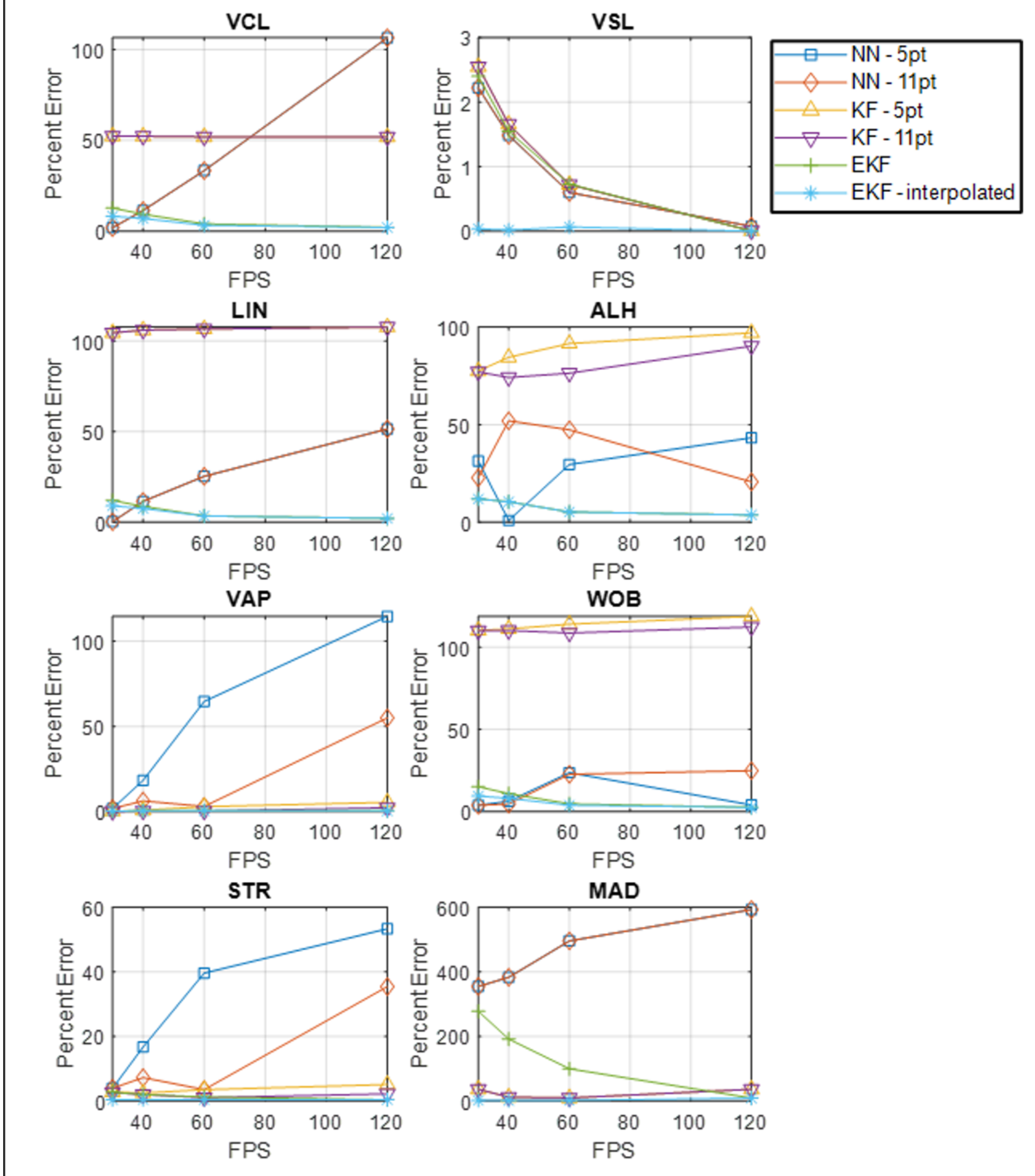


Figure 5.11 Percent error of motility parameters calculated from the six tracking and track averaging methods for test 1.

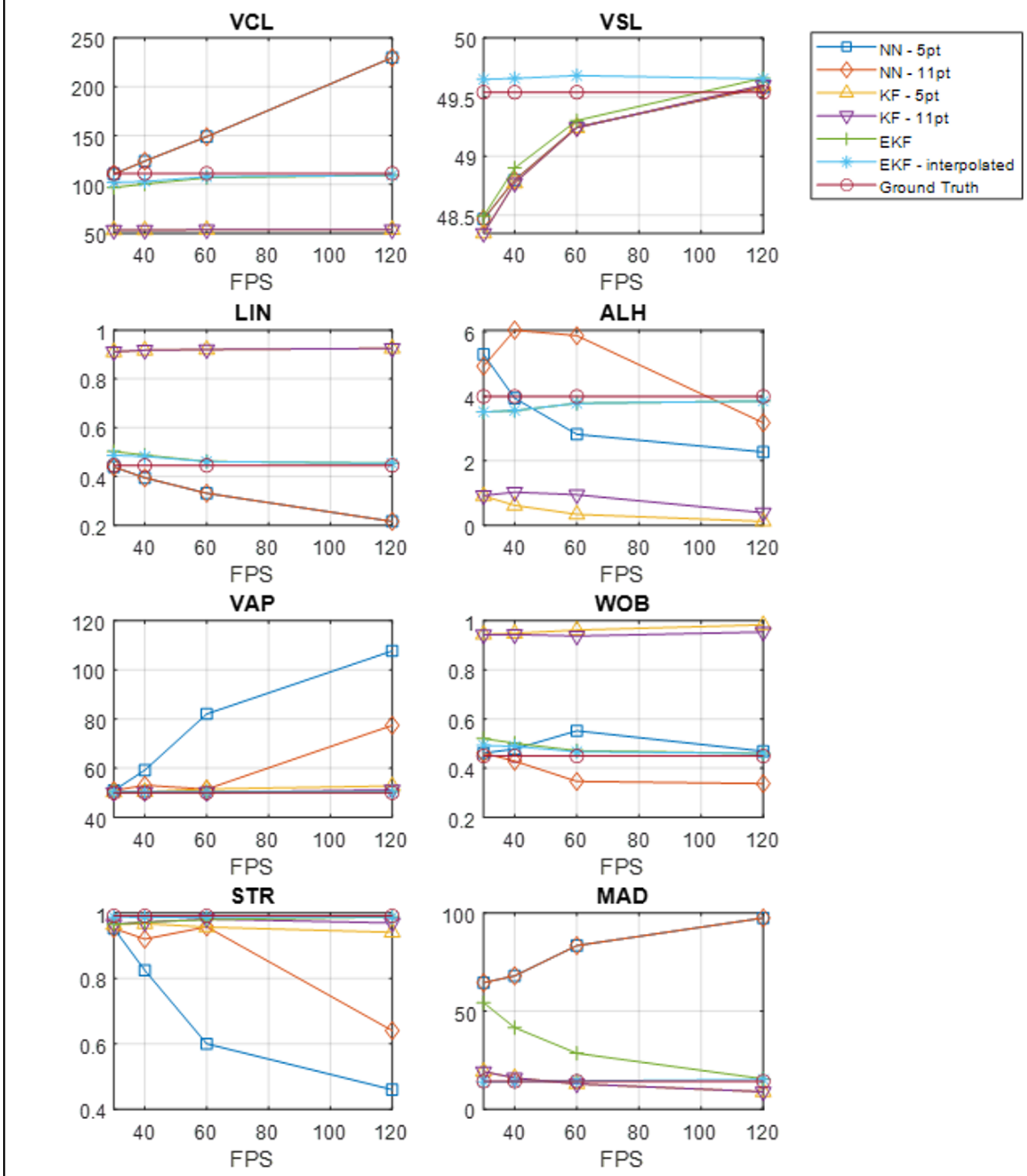


Figure 5.12 Motility parameters calculated from the six tracking and track averaging methods for test 2.

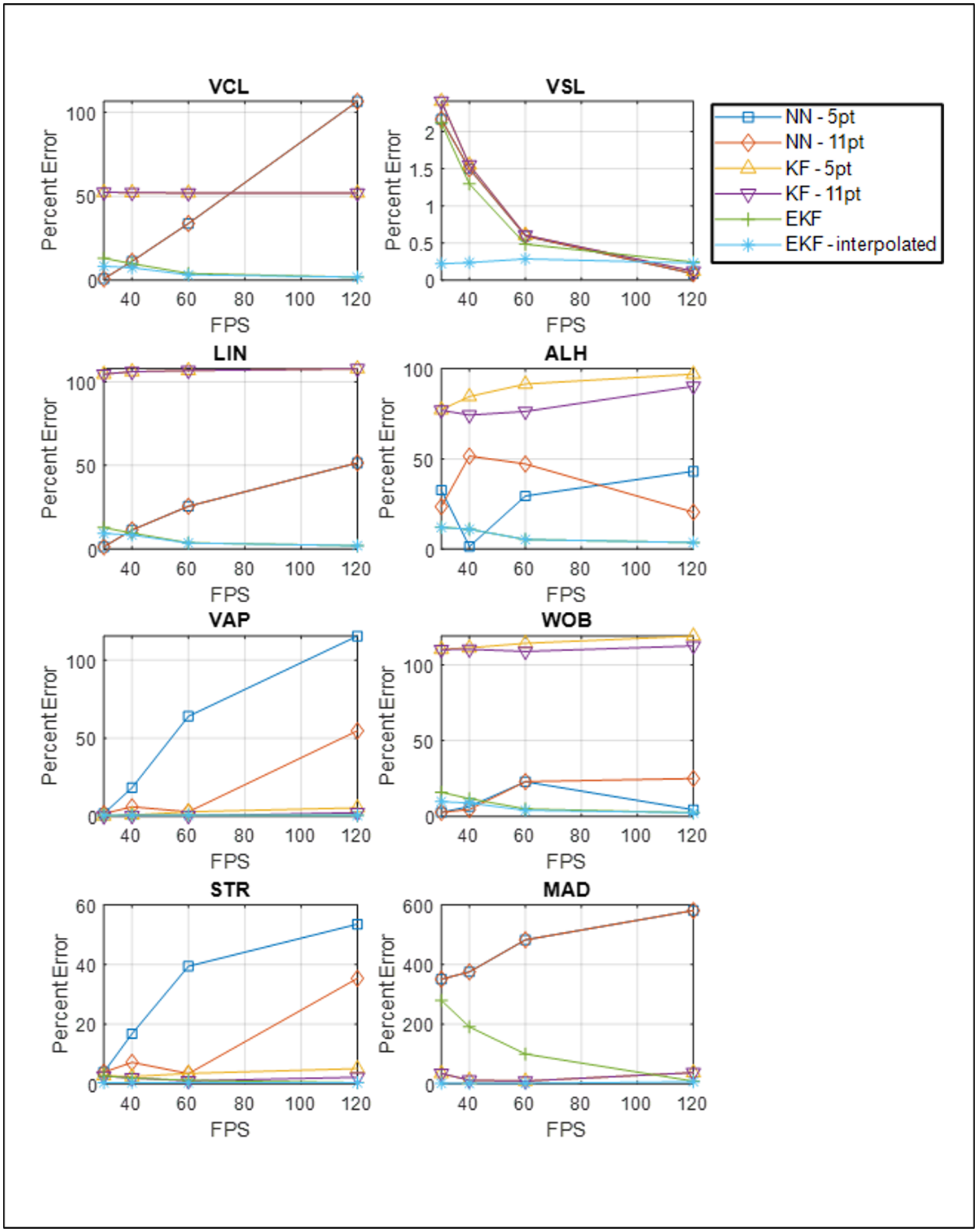


Figure 5.13 Percent error of motility parameters calculated from the six tracking and track averaging methods for test 2.

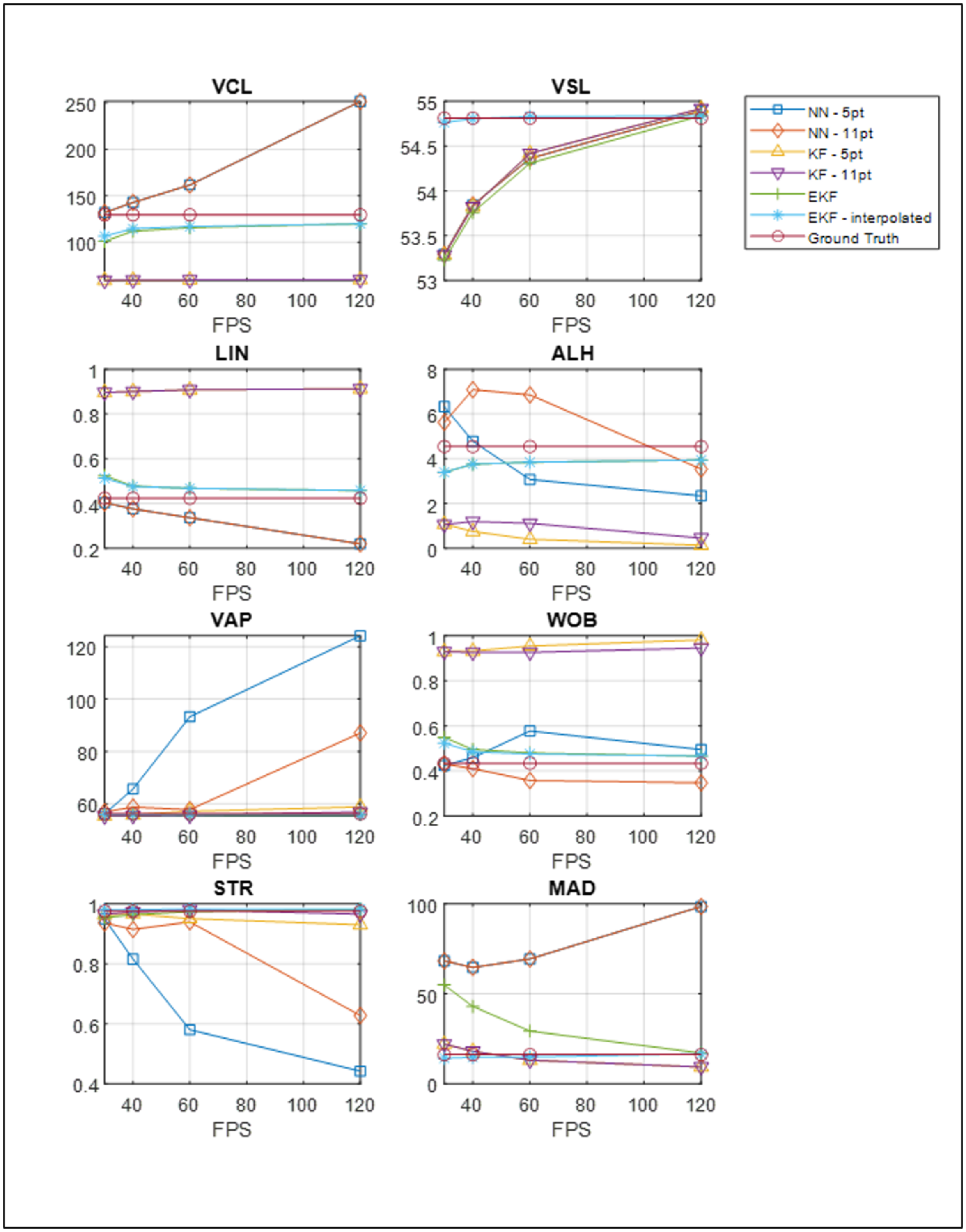


Figure 5.14 Motility parameters calculated from the six tracking and track averaging methods for test 3.

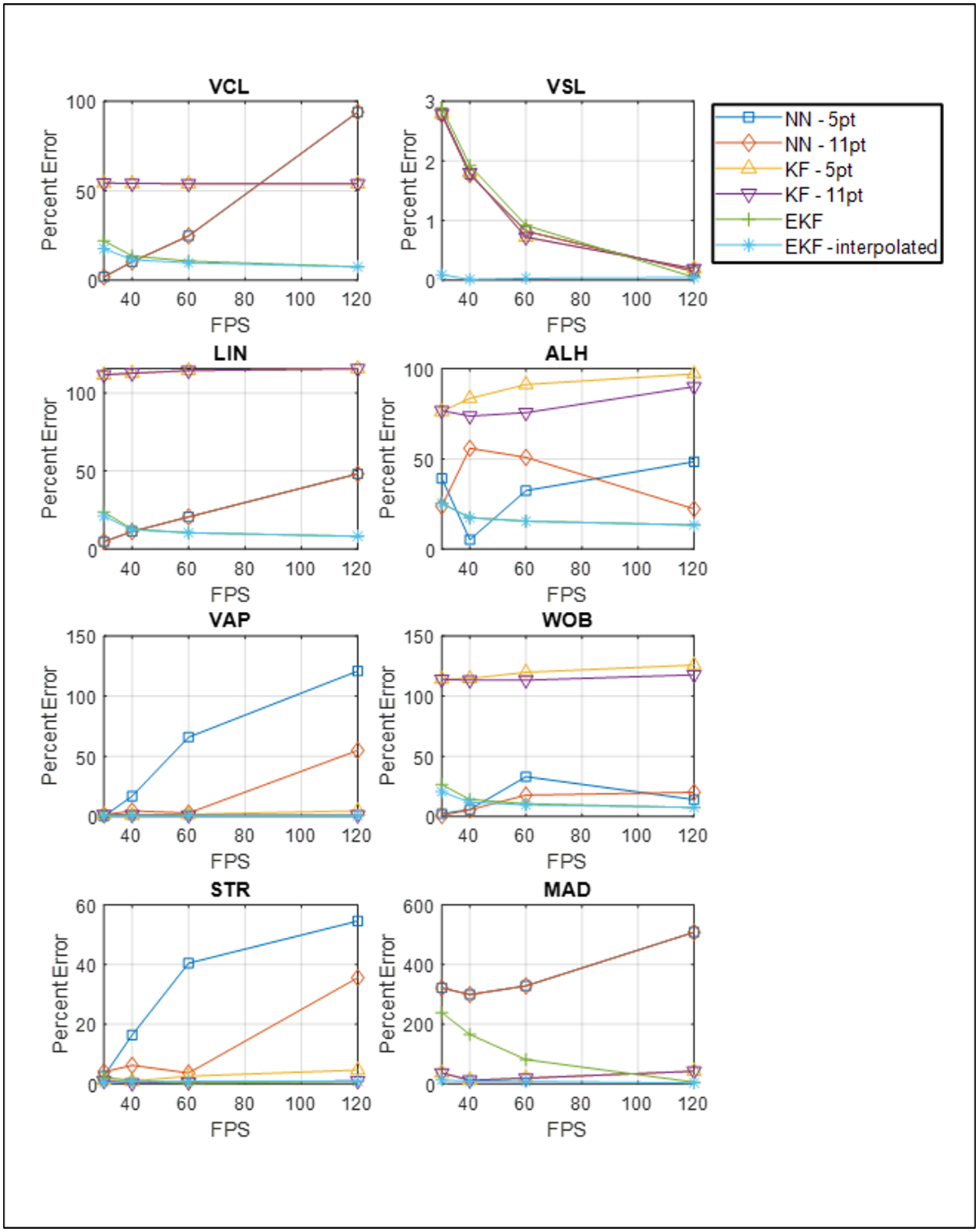


Figure 5.15 Percent error of motility parameters calculated from the six tracking and track averaging methods for test 3.

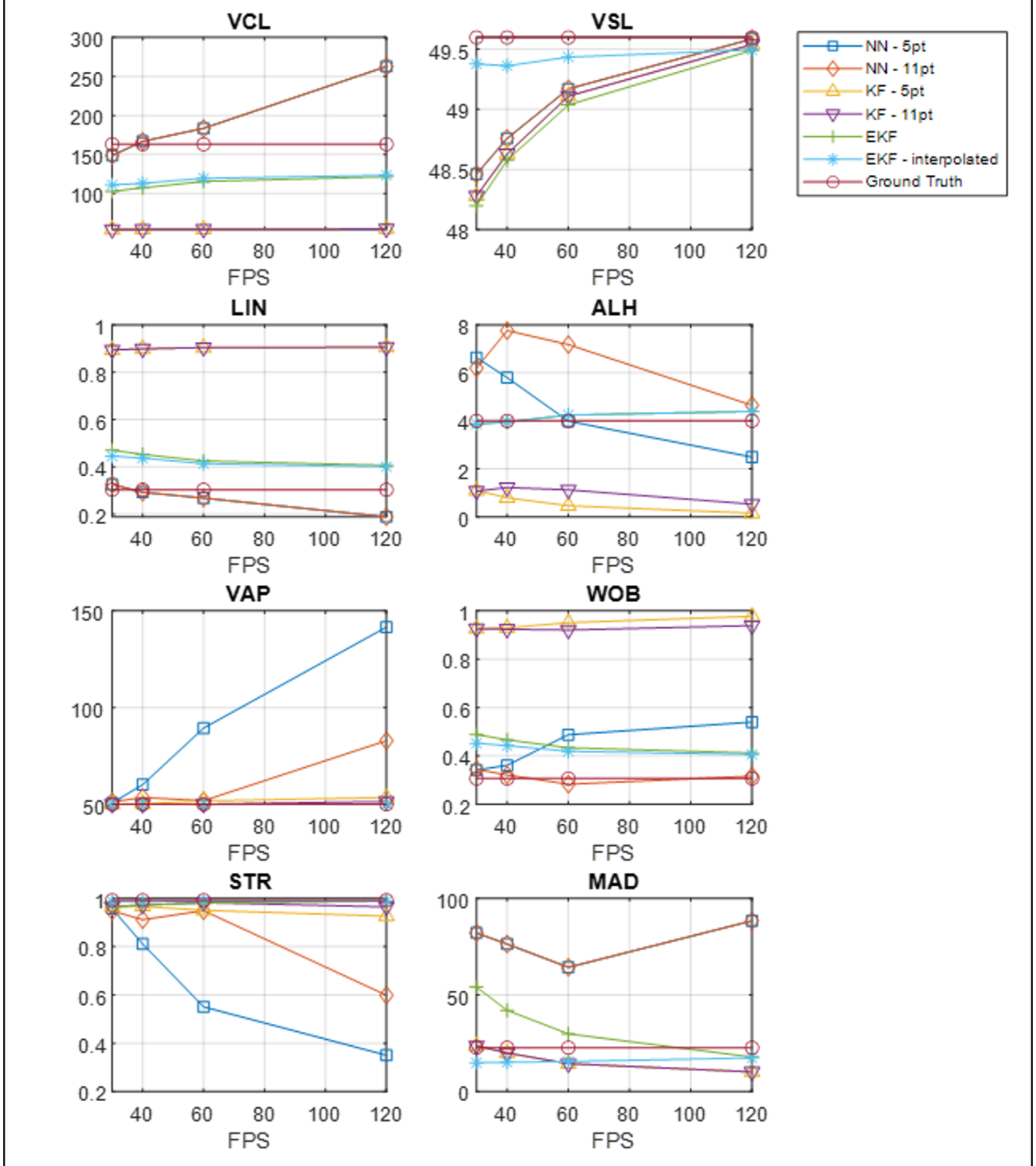


Figure 5.16 Motility parameters calculated from the six tracking and track averaging methods for test 4.

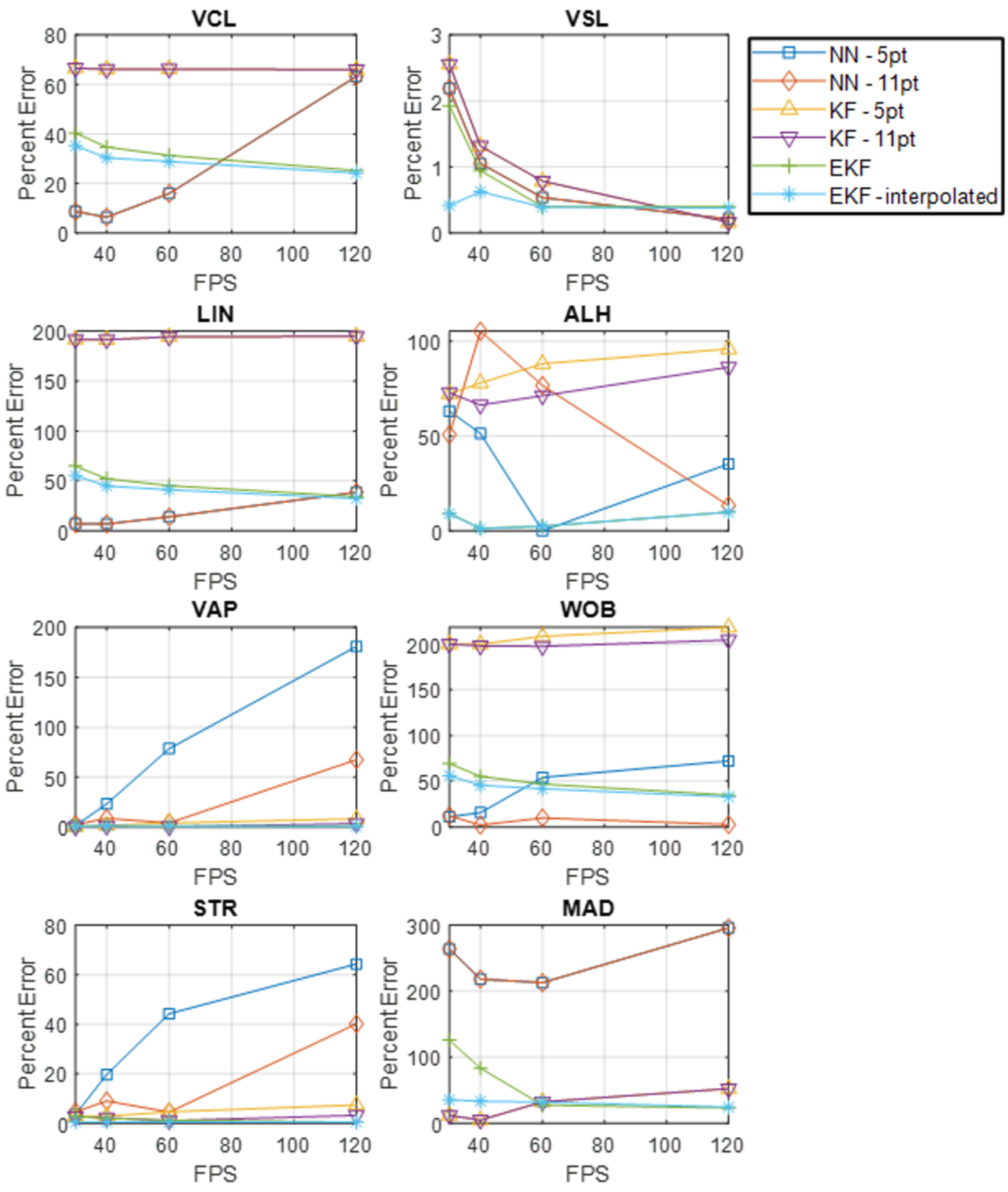


Figure 5.17 Percent error of motility parameters calculated from the six tracking and track averaging methods for test 4.

Overall, the “traveling sinusoid” EKF with interpolated states have shown the highest level of consistency in motility parameter calculations in varying frame rate data with low levels of percent error compared to the other methods. This fact is also illustrated in Figure 5.18, where a box and whiskers plots of the percent error between the motility parameters calculated at 120 FPS to the motility parameters calculated at 30, 40, and 60 FPS for 50 scenarios of test case 3 (noisy traveling sinusoid). Percent error between the values A and B are defined to be

$$\text{Percent Difference} = \frac{|A - B|}{\frac{(A+B)}{2}}. \tag{5.33}$$

For VCL and LIN, the DWNA KF tracking methods show very small percent difference, but this is because the tracks were modeled as a straight line (constant velocity model). Although it is more consistent than the proposed traveling sinusoid EKF model, the percent difference between the ground truth and the calculated values of VCL and LIN are large for DWNA KF methods.

5.5 Conclusion

The use of Extended Kalman filter to estimate the sperm tracks enables consistent and accurate motility parameters compared to existing methods of tracking. The sperm tracks are modeled as a “traveling sinusoid” for tracking and the states are interpolated to reduce the amount of error in calculation of motility parameters due to frame rate. Through the test cases, the robustness of the approach was validated by comparing the result of motility parameter calculation to other methods for tracking and track averaging. The ability to calculate motility parameters with high consistency and accuracy provides a way for existing and future CASA systems to provide more reliable values of motility parameters and semen quality assessments.

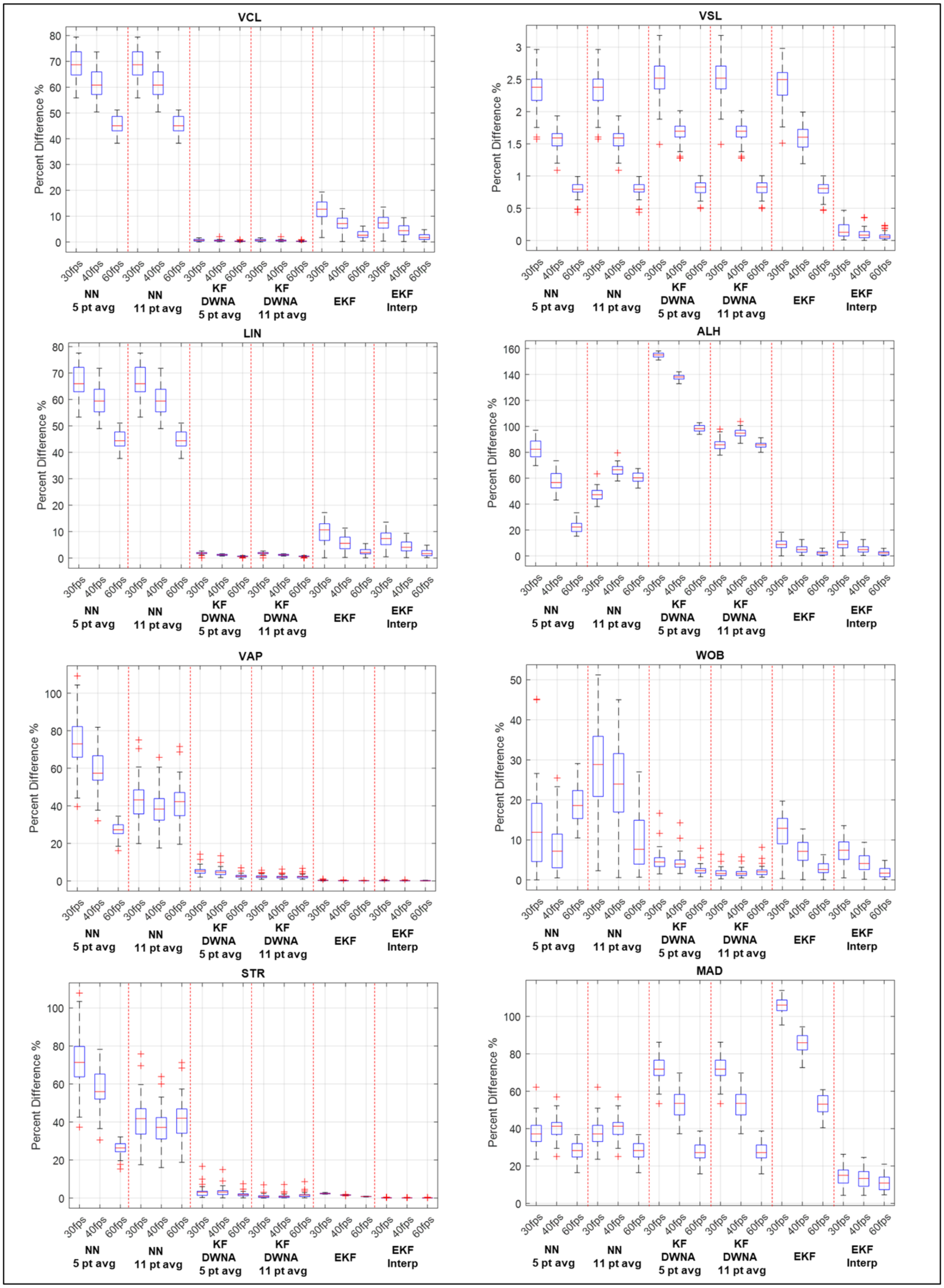


Figure 5.18 Box plots for the percent differences between the motility parameters calculated at 120 FPS to the motility parameters calculated at 30, 40, and 60 FPS.

CHAPTER 6

END NOTES

Four existing challenges in Computer-Assisted Semen Analysis (CASA) systems were addressed. The new solutions include the development of an assessment tool for CASA systems, a digital washing algorithm for unwashed semen samples, a classification technique for sperm cell by swimming patterns, and calculation of motility parameters in semen analysis.

Other challenges in the field exist, calling for further research. In semen analysis, subpopulation analysis is of interest, namely detailed studies of subpopulation of a sperm cohort, where the subpopulation is characterized by distinct characteristics, typically determined through clustering analysis. There are so far no significant findings about subpopulations [63]. Using the classification algorithm developed in this paper, it would be worthwhile to quantify the distribution of swim types within subpopulations and observe the characteristics of subpopulations whose members are known to share a swimming type. Findings from a study of this kind may lead to a deeper understanding of the importance of sperm swim types in semen analysis and in understanding symptoms and maybe even causes for infertility.

Additionally, our study on digital washing can be expanded. In the digital washing algorithm, we have used features of moving sperm cells to detect all sperm cells out of all the objects in the semen image frame. Based on a database of different cell types, it would be interesting to classify different objects or other cells in semen. In specific cases, such as pyospermia (unusual amount of white blood cells in semen), a patient may have a very high concentration of non-sperm cells. By identifying the different non-sperm objects or cells in semen, one may be able to assist clinicians in understanding better the conditions of their patients.

REFERENCES

[1] *Laboratory Manual for the Examination and Processing of Human Semen*, 5th ed., World Health Org., Geneva, Switzerland, 2010.

[2] A. M. Crespilho, L. Chiaradia, A. Cortez, M. Dinelli, F. O. Papa, G. M. Gomes, and K. da Cunha Peixoto Junior, “Sensitivity evaluation of the computer-assisted sperm analysis (CASA) in the determination of frozen-thawed bull semen concentration,” *Brazilian Journal of Veterinary Research and Animal Science*, vol. 54, no. 3, pp. 247–252, 2017.

[3] R. P. Amann and D. F. Katz, “Reflections on CASA after 25 years,” *Journal of Andrology*, vol. 25, no. 3, May 2004.

[4] J. E. Johnson, W. R. Boone, and D. W. Blackhurst, “Manual versus computer-automated semen analyses. part i. comparison of counting chambers,” *Fertility and Sterility*, vol. 65, no. 1, Jan 1996.

[5] S. T. Mortimer, H. van der Horst, and D. Mortimer, “The future of computer-aided sperm analysis,” *Asian Journal of Andrology*, vol. 17, no. 4, Aug 2015.

[6] J. Talarczyk-Desole, A. Berger, G. Taszarek-Hauke, J. Hauke, L. Pawelczyk, and P. Jedrzejczak, “Manual vs. computer-assisted sperm analysis: can CASA replace manual assessment of human semen in clinical practice?” *Ginekologia Polska*, vol. 88, no. 2, 2017.

[7] A. Agarwal, R. Henkel, and A. Majzoub, *Manual of sperm function testing in human assisted reproduction*. Cambridge, United Kingdom: Cambridge University Press, 2021, ch. Standard Semen Analysis: Computer-Assisted Semen Analysis.

[8] M. Tomlinson, K. Pooley, T. Simpson, T. Newton, J. Hopkisson, K. Jayaprakasan, A. Naeem, and T. Pridmore, “Validation of a novel computer-assisted sperm analysis (CASA) system using multitarget-tracking algorithms,” *Fertility and sterility*, vol. 93, pp. 1911–20, 03 2009.

[9] L. F. Urbano, P. Masson, M. VerMilyea, and M. Kam, “Automatic tracking and motility analysis of human sperm in time-lapse images,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 3, Mar 2017.

[10] L. F. Urbano, “Roboust automatic multi-sperm tracking in time-lapse images,” Ph.D. dissertation, Drexel University, 2015.

[11] V. S. Abbiramy and V. Shanthi, “Spermatozoa segmentation and morphological parameter analysis based detection of teratozoospermia,” *International Journal of Computer Applications*, vol. 3, Jun 2010.

[12] F. Mostajer Kheirkhah, H. R. Sadegh Mohammadi, and A. Shahverdi, “Histogram non-linear transform for sperm cells image detection enhancement,” in *Eighth International Conference on Information and Knowledge Technology (IKT)*, 2016.

[13] ——, “Efficient and robust segmentation and tracking of sperm cells in microscopic image sequences,” *IET Computer Vision*, vol. 13, pp. 489–499, 2019.

[14] G. Liu, C. Liu, X. Liu, and B. Liu, “Automatic tracking and activity evaluation system of sperm motions,” in *Chinese Automation Congress (CAC)*, 2017.

[15] G. Luo, W. Huang, and S. Li, “2-d maximum entropy spermatozoa image segmentation based on canny operator,” in *International Conference on Intelligent Computing and Integrated Systems*, Oct 2010.

[16] R. A. Movahed, E. Mohammadi, and M. Orooji, “Automatic segmentation of sperm’s parts in microscopic images of human semen smears using concatenated learning approaches,” *Computers in Biology and Medicine*, vol. 109, pp. 242–253, 2019.

[17] S. Qi, T. Nie, Q. Li, Z. He, D. Xu, and Q. Chen, “A sperm cell tracking recognition and classification method,” in *International Conference on Systems, Signals and Image Processing (IWSSIP)*, Aug 2019.

[18] Y. Ren, P. Wen, S. Li, Y. Liang, and W. Huang, “An improved algorithm of rat sperm image segmentation and counting,” in *International Conference on Intelligent Computing and Integrated Systems*, Oct 2010.

[19] Y. Wang, Y. Jia, M. Yuchi, and M. Ding, “The computer-assisted sperm analysis technique for sperm morphology evaluation,” in *International Conference on Intelligent Computation and Bio-Medical Instrumentation*, Dec 2011.

[20] Z. L. Xuan and W. Z. Yan, “The sperm video segmentation based on dynamic threshold,” in *International Conference on Machine Learning and Cybernetics*, Jul 2010.

[21] Y. Zhang, “Animal sperm morphology analysis system based on computer vision,” in *Eighth International Conference on Intelligent Control and Information Processing (ICICIP)*, Nov 2017.

[22] V. Wiesmann, M. Bergler, R. Palmisano, M. Prinzen, D. Franz, and T. Wittenberg, “Using simulated fluorescence cell micrographs for the evaluation of cell image segmentation algorithms,” *BioMed Central Bioinformatics*, vol. 18, 2017.

[23] V. Wiesmann, T. Sauer, C. Held, R. Palmisano, and T. Wittenberg, “Cell simulation for validation of cell micrograph evaluation algorithms,” *Biomedizinische Technik/Biomedical engineering*, vol. 58, 09 2013.

[24] R. F. Murphy, “Building cell models and simulations from microscope images,” *Methods*, vol. 96, pp. 33–39, 2016.

[25] A. Lehmussola, J. Selinummi, P. Ruusuvuori, A. Niemisto, and O. Yli-Harja, “Simulating fluorescent microscope images of cell populations,” in *IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, pp. 3153–3156.

[26] T. Nattkemper, A. Saalbach, and T. Twellmann, “Evaluation of multiparameter micrograph analysis with synthetical benchmark images,” *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, vol. 1, pp. 667 – 670, Oct 2003.

[27] P. Ruusuvuori, A. Lehmussola, J. Selinummi, T. Rajala, H. Huttunen, and O. Yli-Harja, “Benchmark set of synthetic images for validating cell image analysis algorithms,” in *16th European Signal Processing Conference*, 2008, pp. 1–5.

[28] L. Armon, S. R. Caplan, M. Eisenbach, and B. M. Friedrich, “Testing human sperm chemotaxis: How to detect biased motion in population assays,” *PLoS ONE*, vol. 7, no. 3, Mar 2012.

[29] L. J. Fauci and C. S. Peskin, “A computational model of aquatic animal locomotion,” *Journal of Computational Physics*, vol. 77, no. 1, pp. 85–108, Jul 1988.

[30] R. D. Dresdner and D. F. Katz, “Relationships of mammalian sperm motility and morphology to hydrodynamic aspects of cell function,” *Biology of Reproduction*, vol. 25, pp. 920–930, Jul 1981.

[31] M. P. Curtis, J. C. Kirkman-Brown, T. J. Connolly, and E. A. Gaffney, “Modelling a tethered mammalian sperm cell undergoing hyperactivation,” *Journal of Theoretical Biology*, vol. 309, pp. 1–10, Sept 2012.

[32] K. Ishimoto and E. A. Gaffney, “A study of spermatozoan swimming stability near a surface,” *Journal of Theoretical Biology*, vol. 360, pp. 187–199, Jul 2014.

[33] K. Ishimoto, J. Cosson, and E. A. Gaffney, “A simulation study of sperm motility hydrodynamics near fish eggs and spheres,” *Journal of Theoretical Biology*, vol. 389, pp. 187–197, Nov 2015.

[34] K. Ishimoto and E. A. Gaffney, “Fluid flow and sperm guidance: a simulation study of hydrodynamic sperm rheotaxis,” *Journal of the Royal Society Interface*, vol. 12, May 2015.

[35] K. Ishimoto, H. Gadelha, E. A. Gaffney, D. J. Smith, and J. Kirkman-Brown, “Coarse-graining the fluid flow around a human sperm,” *Physical Review Letter*, vol. 118, no. 12, Mar 2017.

[36] J. Simons, L. Fauci, and R. Cortez, “A fully three-dimensional model of interaction of driven elastic filaments in a stokes flow with application to sperm motility,” *Journal of Biomechanics*, vol. 48, Jan 2015.

[37] D. J. Smith, E. A. Gaffney, J. R. Blake, and J. C. Kirkman-Brown, “Human sperm accumulation near surfaces: a simulation study,” *Journal of Fluid Mechanics*, vol. 621, Feb 2009.

[38] T. Su, L. Xue, and A. Ozcan, “High-throughput lensfree 3d tracking of human sperms reveals rare statistics of helical trajectories,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, no. 40, Oct 2012.

[39] D. F. Babcock, P. M. Wandernoth, and G. Wennemuth, “Episodic rolling and transient attachments create diversity in sperm swimming behavior,” *BioMed Central Biology*, vol. 12, no. 67, 2014.

[40] J. Choi, L. Alkhoury, and M. Kam, “Njit data fusion lab: Sperm simulator,” https://github.com/JiwonChoi-NJIT/NJIT_sperm_simulator, 2021 (accessed November 24, 2021).

[41] M. R. Villarreal, “File:complete diagram of a human spermatozoa en.svg,” Apr 2006. [Online]. Available: https://commons.wikimedia.org/wiki/File:Complete_diagram_of_a_human_spermatozoa_en.svg

[42] E. Gaffney, H. Gadêlha, D. Smith, J. Blake, and J. Kirkman-Brown, “Mammalian sperm motility: Observation and theory,” *Annual Review of Fluid Mechanics*, vol. 43, pp. 501–528, 2011.

[43] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. New York, NY: Pearson, 2018.

[44] E. de Lamirande, P. Leclerc, and C. Gagnon, “Capacitation as a regulatory event that primes spermatozoa for the acrosome reaction and fertilization,” *Molecular Human Reproduction*, vol. 3, no. 3, 1997.

[45] V. J. Kay and L. Robertson, “Hyperactivated motility of human spermatozoa: a review of physiological function and application in assisted reproduction,” *Human Reproduction Update*, vol. 4, no. 6, 1998.

[46] M. U. Daloglu and A. Ozcan, “Computational imaging of sperm locomotion,” *Biology of Reproduction*, vol. 97, no. 2, pp. 182–188, Jul 2017. [Online]. Available: https://doi.org/10.1093/biolre/iox086

[47] P. Glasserman, “Brownian motion,” in *Monte Carlo Methods in Financial Engineering*. New York, NY: Springer-Verlag, 2004, ch. 3.

[48] L. F. Urbano, P. Masson, M. VerMilyea, and M. Kam, “Njit data fusion lab: Real and simulated semen images,” https://github.com/moshekam/NJIT-Semen-Images-Data-Fusion-Lab, 2021 (accessed November 24, 2021).

[49] D. Schuhmacher, B. Vo, and B. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug 2008.

[50] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786550500303X

[51] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.

[52] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, 2008.

[53] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, pp. 83–97, 1955.

[54] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[55] D. Bradley and G. Roth, "Adaptive thresholding using the integral image," *Journal of Graphic Tools*, vol. 12, no. 2, pp. 13–21, 2007.

[56] C. Stauffer and G. W. E. L, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, Feb 1999.

[57] L. F. Urbano, P. Masson, M. VerMilyea, and M. Kam, "Spermtrackingproject," https://github.com/JiwonChoi-NJIT/SpermTrackingProject, 2021 (accessed November 24, 2021).

[58] J. Fedder, "Nonsperm cells in human semen: with special reference to seminal leukocytes and their possible influence on fertility," *Archives of Andrology*, vol. 36, no. 1, pp. 41–65, 1996.

[59] "What is immature sperm? - definition from fertilitysmarts," Apr 2020. [Online]. Available: https://www.fertilitysmarts.com/definition/335/immature-sperm

[60] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[61] B. P. Welford and B. P. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, pp. 419–420, 1962.

[62] "3D Human Action Recognition using Hu Moment Invariants and Euclidean Distance Classifier," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, pp. 13–21, 2017.

[63] F. Martinez-Pastor, J. J. Jorge Tizado, E Garde, L. Anel, and P. De Paz, "Statistical series: opportunities and challenges of sperm motility subpopulation analysis," *Theriogeneology*, vol. 75, no. 5, pp. 783–795, 2011.

[64] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[65] D. Karaboga, “An idea based on honey bee swarm for numerical optimization, technical report - tr06,” *Technical Report, Erciyes University*, Jan 2005.

[66] V. Tereshko and A. Loengarov, “Collective decision-making in honey bee foraging dynamics,” *Computing Information Systems*, vol. 9, Nov 2004.

[67] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901257

[68] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” in *Proceedings of the Annual Association for Computing Machinery - Society for Industrial and Applied Mathematics (ACM-SIAM) Symposium on Discrete Algorithms*, vol. 8, Jan 2007, pp. 1027–1035.

[69] D. Mortimer, *Practical laboratory andrology*. New York, NY: Oxford University Press, 1994.

[70] A. Morris, J. Coutts, and L. Robertson, “A detailed study of the effect of videoframe rates of 25, 30 and 60 Hertz on human sperm movement characteristics,” *Human Reproduction*, vol. 11, no. 2, pp. 304–310, Feb 1996. [Online]. Available: https://doi.org/10.1093/HUMREP/11.2.304

[71] C. Castellini, A. Bosco, S. Ruggeri, and G. Collodel, “What is the best frame rate for evaluation of sperm motility in different species by computer-assisted sperm analysis?” *Fertility and sterility*, vol. 96, pp. 24–7, Jul 2011.

[72] R. O. Davis, P. W. Niswander, and D. F. Katz, “New measures of sperm motion. i. adaptive smoothing and harmonic analysis.” *Journal of andrology*, vol. 13, no. 2, pp. 139–152, 1992.

[73] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation*. New York, NY: John Wiley & Sons, inc, 2001.

[74] E. Meijering, O. Dzyubachyk, and I. Smal, “Chapter nine - methods for cell and particle tracking,” in *Imaging and Spectroscopic Analysis of Living Cells*, ser. Methods in Enzymology, P. M. conn, Ed. Academic Press, 2012, vol. 504, pp. 183–200. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123918574000094

[75] X. Tong and C. Tang, “Robotic fish tracking method based on suboptimal interval Kalman filter,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 10605, Nov 2017, p. 106051J.

[76] Z.-M. Qian, X. E. Cheng, and Y. Q. Chen, “Automatically detect and track multiple fish swimming in shallow water with frequent occlusion,” *PLoS ONE*, vol. 9, no. 9, pp. 1–12, Sept 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0106506

[77] F. Boyer, M. Porez, A. Leroyer, and M. Visonneau, “Fast dynamics of an eel-like robot—comparisons with navier–stokes simulations,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1274–1288, 2008.

[78] M. Bergmann and A. Iollo, “Modeling and simulation of fish-like swimming,” *Journal of Computational Physics*, vol. 230, no. 2, pp. 329–348, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999110005115

[79] J. F. Jikeli, L. Alvarez, B. M. Friedrich, L. G. Wilson, R. Pascal, R. Colin, M. Pichlo, A. Rennhack, C. Brenker, and U. B. Kaupp, “Sperm navigation along helical paths in 3d chemoattractant landscapes,” *Nature Communications*, vol. 6, 2015.

[80] C. J. Voesenek, F. T. Muijres, and J. L. van Leeuwen, “Biomechanics of swimming in developing larval fish,” *Journal of Experimental Biology*, vol. 221, no. 1, 2018, jeb149583. [Online]. Available: https://doi.org/10.1242/jeb.149583

[81] U. Müller, J. Boogaart, and J. L. Van Leeuwen, “Flow patterns of larval fish: Undulatory swimming in the intermediate flow regime,” *The Journal of Experimental Biology*, vol. 211, pp. 196–205, Feb 2008.

[82] G. Ozmen Koca, C. Bal, D. Korkmaz, M. C. Bingol, M. Ay, Z. H. Akpolat, and S. Yetkin, “Three-dimensional modeling of a robotic fish based on real carp locomotion,” *Applied Sciences*, vol. 8, no. 2, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/2/180