June 13, 2021

Dr. W. Craig Scratchley
School of Engineering Science
Simon Fraser University
Burnaby, BC
V5A 1S6

Re: ENSC 405W/440 Requirement Specifications for Smart Swim by Smart Swim Analytics

Dear Dr. Scratchley,

Attached to this letter is a document outlining the requirement specifications for the Smart Swim as requested for ENSC 405W. The Smart Swim relies upon machine learning models to detect and track a swimmer using swimming competition footage or in real-time using a tilt-pan camera, and to provide an estimation of the swimmer's stroke rate. It provides valuable data for competitive swimmers and trainers.

The requirements specifications report will analyze the problem and specify requirements that will be necessary for the alpha phase development of the product. The problem is split into four systems, namely the Data Collection System, the Model Training System, the Real-Time Swimmer Tracking System, and the Swimmer Positioning System. Under each system, the report outlines the corresponding functional requirements such as hardware and software specifications. Moreover, the document will also outline requirements for meeting engineering standards as well as requirements that pertain to environmental sustainability and user safety.

Smart Swim Analytics is a team of 5 senior Computer Engineering students, Tim Woinoski, Kiran Brar, Kuro Chen, Ethan Cai, and Ray Kim, and Systems Engineering student Kudus Elbo-Iswadi. Together, we aim to utilize our combined knowledge to make the concept of Smart Swim a reality.

Thank you for taking the time to read the Smart Swim requirement specifications. For any questions or concerns, please feel free to contact Tim Woinoski at tim_woinoski@sfu.ca.

Kind Regards,

Tim D

Tim Woinoski
Project Lead
Smart Swim Analytics

REQUIREMENTS SPECIFICATION

for

Smart Swim Analytics System

Version 1.0

Prepared by

Kudus Elbo-Iswadi, Gurkiran Brar, Ethan Cai, Kuro Chen, Ray Kim, Tim Woinoski

Smart Swim Analytics

June 14, 2021

1. Abstract

Smart Swim Analytics aims to help Canadian swimmers improve their performance as cheaply as possible. Thus we have designed the Automated Swimming Analytics System, inspired from [1]. This system will determine a swimmer's stroke length and stroke rate, which are important metrics for measuring performance. Not only will the system be quicker than the current method of measuring these metrics by hand, it will also be cheaper [2].

The product will consist of an annotation tool that lets users easily annotate videos for use in training the model. This task will only need to be performed once at a given pool, after which the model will be ready to use with the Real-time Swimmer Tracking System and Swimmer Positioning System, which detect the swimmers in real time and determine their position in the pool, respectively. This document outlines the various requirements that the previously mentioned components of the product must meet, as well as addressing sustainability and safety concerns.

Contents

List of Figures

List of Tables

2. Introduction

2.1. Purpose

This documentation outlines the requirements for a complete Automated Swimming Analytics System. The details of the scope of this system is outlined in section 2.4. This documentation is the first of many for the creation of an Automated Swimming Analytics System. That being said, at this point, all of the requirements for this project can be found in this single document. No other documents for requirements are needed for the Automated swimming Analytics system proposed here.

2.2. Document Conventions

This documentation was created using LATEXbook style. Its format was copied from the templated given by [4]. Each requirement is defined as follows, "RX.X.X.X", in which each number "X" represents the chapter, section, sub-section and item (in that order) in which the requirement was defined. Each requirement is given a phase in which that requirement will be implemented; such phases are, alpha, beta and production.The requirements that depend on other requirements will not necessarily inherit the importance of the parent requirement.

2.3. Intended Audience and Reading Suggestions

This document is for developers, project managers, users, testers, and documentation writers. The document is organized as follows. Chapter 3 gives a general overview of the automated swimming analytics system. Chaptergives detail on how the Hardware and Software systems interface with each other and the users. Chapter 4 gives the expected features of each subsystem. Lastly the next chapters 4 to 7 detail additional requirements such as nonfunctional requirements like, performance, safety, security, and quality. Appendix A is given as a reference to people who are unfamiliar with swimming. For readers who wish to get a good over of the proposed system, Chapter 3 will give all the required detail. For more in-depth information Chapters 4-8 should be explored.

2.4. Project Scope

This system allows for the collection of swimmer analytics from general non-static footage of swimmers, also known as general overhead race video (ORV). This system makes almost no assumptions about the manner in which the footage is or was collected. In other words it is very robust. Many analytics can be collected by humans utilizing ORV, however only two will be collected with this system. Such metrics will be stroke length and stroke rate, defined in Appendix A. There are four main sub-systems to this system, more detail is given in Chapter 4. Each part works together in order to allow the analytics system to be utilized in any swim competition setting while maintaining robust analytics results quickly and efficiently.

2.5. References

References are given as IEEE in-text-citations and can be referred to in the Bibliography of this document.

3. Overall Description

3.1. Product Perspective

In swimming, data is collected for swimmers at competitions utilizing ORV of swimmers, by a variety of people. Of particular note, RaceTek [2] provides Canadian swimmers with racing data, as seen in Figure 3.1, for every major and some minor competitions. RaceTek has been doing so for many years now but unfortunately, their services are very expensive, and data can not be released to the public. There are other groups that provide similar services in practice environments, such as Form Swim goggles and TritonWare [5, 6].

The basic services provided by RaceTek include; stroke rate analysis, swim velocity analysis and turn analysis collected from videos recorded at swim competitions. All calculations are done by hand using video footage to analyze swimming. These tasks are time consuming, but have the potential to be automated by a computer. The goal of this system is to create a system that will automate these tasks while also making more accurate measurements of the swimmer. As such this system will greatly reduce the cost of collecting swimmer data. In addition Such a system would save coaches and athletes across the world many hours of analyzing post-race videos manually. Additionally, the ability to process swimming footage in real-time is more beneficial to a swimmer than if they are given feedback days later.

This automated swimming analytics system is a new self-contained project. Automated swimming analytics has been attempted by a few organizations [7, 8] and its completion would put Canada ahead in terms of swimming data analytics. In addition to this, the assumptions made in this proposed system will be even less than the ones made in [7, 8]. As such this project has additional complexities added to it. But, in turn, has some additional benefits such as flexibility in terms of recording footage.

3.2. Product Functions

The motivation for this project is to develop a system that will automate the collection of swimming analytics in swim competition videos using image-based processing methods and tracking algorithms. Specifically, those analytics will be the stroke rate and length of a particular swimmer selected before the commencement of a race. In addition to the collection of these metrics the analytics system must work for a general competition setting. Referencing figure 3.2 which gives a high level overview of the proposed systems. The following functions will be preformed to facilitate this goal in a general competition setting. When given a competition venue for data collection a real-time stroke rate

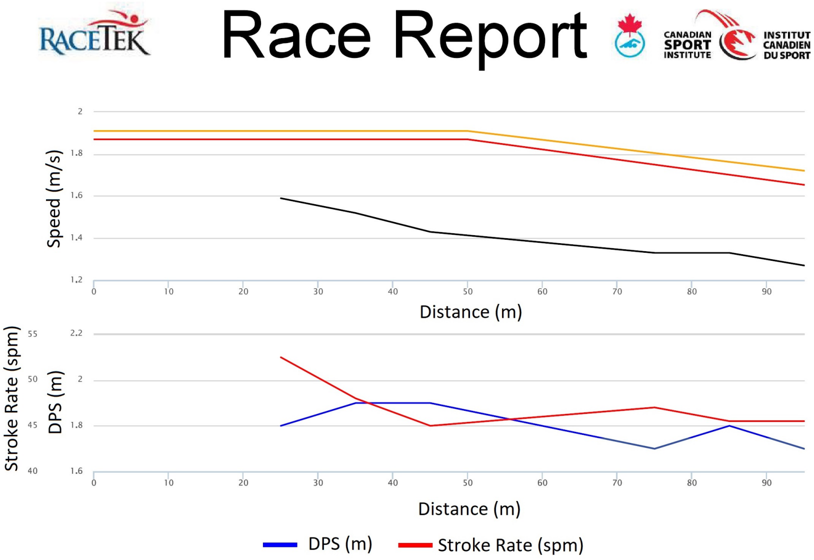

Figure 3.1.: A modified example report from RaceTek [2], found under Video Race Analysis (VRA)

estimation and positioning system that estimates the metrics mentioned will estimate the stroke rate and position of the swimmer of interest. If the competition venue of interest is new, a system that allows for the collection of new data for that particular competition of interest is required in order for the mentioned systems to work. As such, necessary data is collected with the data collection system. This data will be used for updating the required swimmer models so that the tracking and positioning system will work in a general setting.

3.3. User Classes and Characteristics

This product will be used by the following user classes: Athletes, Sports Analysts, Swimmer Data Collection Specialists, and Footage Collectors.

3.3.1. Athletes

While this product directly benefits the Athletes, they are the least important to satisfy in terms of product development as they are only interested in the resulting data produced by the system, they do not have to interact with the system.

3.3.2. Data Collection Specialist

The Swimmer Data Collection specialist mainly interacts with the Data collection system, thus they are moderately important. They are required to annotate the swimmers in a new competition venue. They must be able to create high quality data for the model

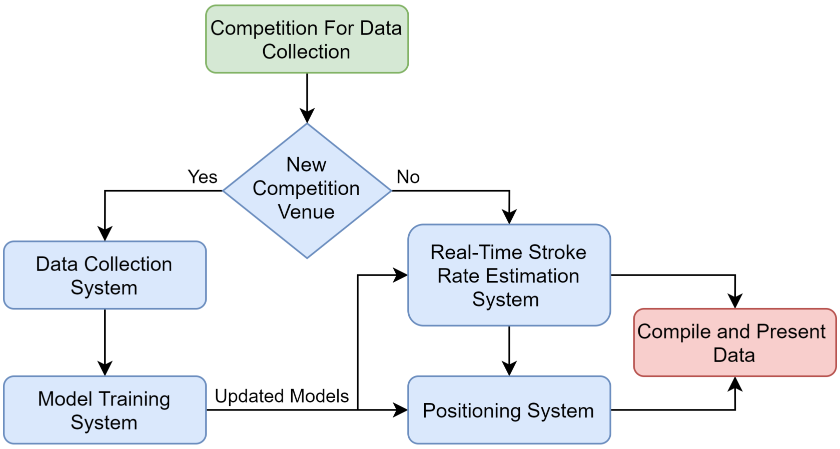

Figure 3.2.: The main system functions and how they work together

training system to learn from. The Data Collection specialist must be trained no how to annotate footage of swimmers such that they are consistent with the already existing data.

3.3.3. Footage Collectors

The Footage Collectors must place the footage collection system in a location for which the swimmer models have been trained on, thus they are moderately important. They must make sure the Real-time system is working correctly and select the swimmer of interest for each race. Foremost, they must be trained on the setup of the tracking system. Additionally they should know how to select the swimmer of interest and troubleshoot any issues that arise.

3.3.4. Sports Analysts

The Sports Analysts, is the most important. They must orchestrate the users of the data collection system and the footage collectors. They also utilize the collected data from the data specialist to train or update swimmer models for the stroke rate estimation system and positioning systems. They may also have to interact with output data in order to get it in a format that is convenient for them. They must be trained on how to update models and install them into the new system when necessary. They must also be trained on how to extract data from the system and how to interpret it.

3.4. Design and Implementation Constraints

The main issues that will limit the options available will be hardware. The more powerful the hardware the more accurate processing can occur. This is especially for the case of the real-time tracking system.

3.5. User Documentation

The user documentation will consist of an programming API reference for use of the training, real-time detection and positioning systems. Finally there will be user documentation for the data creation system. As far as API references go there are a lot of ways to write an API reference. The API reference will follow a similar format to that of the OpenCV library [9].

3.6. Assumptions and Dependencies

We assume we can get all hardware specified, that being a NVIDIA Jetson board [10] and a tilt pan hardware from [11]. We assume that all the hardware can communicate with each-other as well as any additional computers.

4. System Features

The following section outlines the major system features in the swimming analytics requirements. They are split into four sections 4.1 Data Collection, 4.2 Model Training, 4.3 Real-time Swimmer Tracking, and 4.4 Swimmer Positioning.

4.1. Data collection System

This section highlights the requirements of the data collection system. The data collection system is broken down into three parts: detection data, stroke data, and position data collection systems.

4.1.1. Functional Requirements: Detection Data

This section details the requirements for the system that collects detection data.

R4.1.1.1 The system must have the ability to read a video file of maximum 30 minutes in length

a) Must be able to handle MP4 and AVI formats.

Stage: alpha

R4.1.1.2 The system must have the ability to do image annotation from a video

a) system allows user to create boxes around objects,

Stage: alpha

R4.1.1.3 The system must have the ability to set the frame rate for annotation

a) The frame rate must be at least one.

b) The frame rate at most can be the number of frames in the video.

Stage: alpha

R4.1.1.4 The system must have the ability to define the swimmer's class

a) The classes are Diving, On-blocks, Underwater, Swimming, Turning, and Finishing

Stage: alpha

R4.1.1.5 The system must keep track of the lane each swimmer is in

a) The system must allow for annotation of more than one swimmer in a lane (particularly for the diving phase of competition)

b) The system must be able to specify if a swimmer is absent in a lane in a given frame

Stage: alpha

R4.1.1.6 The system must allow the user to select the lane being annotated at a given moment
Stage: alpha

R4.1.1.7 The system must have a mode that allows for “quick annotation” which automatically moves to the next frame when an annotation is made
Stage: alpha

R4.1.1.8 The system must generate data through image annotation made by the system user

a) Once the annotation session is completed data output is automatically generated

b) Data output is data representation of the annotations completed on all frames in the session

c) A backup file must be generated to save the file being worked on to reduce data loss if the program crashes

Stage: alpha

R4.1.1.9 The system must have the ability to display a frame of a given video to annotate
Stage: alpha

R4.1.1.10 The system must be able to load created data and display it on its corresponding frames

a) The system must be able to update the loaded data

b) The system must be able to check the loaded data for missing annotations (relative to the rate of annotation)

c) Once the rate of annotation is set it can not be changed

Stage: alpha

R4.1.1.11 The system must allow the user to move from frame to frame forwards and backwards
Stage: alpha

R4.1.1.12 The system must allow the user to be able to select a frame number for which to jump to
Stage: alpha

R4.1.1.13 The system must display information about how the user is annotating at a given moment

a) The Current frame number must be displayed

b) The annotation class must be displayed

c) The Lane number being annotated must be displayed

d) A backup file must be generated to save the file being worked on to reduce data loss if the program crashes

Stage: alpha

R4.1.1.14 The system must have the ability to store and archive the data output of an annotation
Stage: alpha

R4.1.1.15 The system must have the ability to clear and redo annotations done in the session
Stage: beta

R4.1.1.16 The system must have the ability to load the data into a database
Stage: beta

4.1.2. Functional Requirements: Stroke Data

This section details the requirements for the system that collects stroke data.

R4.1.2.1 The system must allow the user to annotate strokes

a) The system must allow the user to specify position in time of the top of a swimmers stroke as defined in section A

b) The system allows the user to specify the position in time of the swimmers stroke as the video is playing for fast annotation

c) The system must interpolate a complete sinusoidal signal between each top of stroke with amplitude 1 and max value 1

d) The frequency of the sinusoidal signal is $\frac{1}{T}$ where T is the period of time between the top of one stroke and the next

Stage: alpha

R4.1.2.2 There must be an edit mode for which the user can annotate footage

a) If no data exists for the video being annotated this mode allows the user to annotate strokes

b) If data exists this mode asks user if they want to clear data and re-annotate the current video

Stage: alpha

R4.1.2.3 The user must be able to specify if the swimmer being annotated is swimming or not at any given time
Stage: alpha

R4.1.2.4 The user must be able to specify the stroke of the swimmer being annotated
Stage: alpha

R4.1.2.5 The user must be able to pause when annotating strokes
Stage: alpha

R4.1.2.6 The user must be able to go back 5 seconds to fix any errors when annotating
Stage: alpha

R4.1.2.7 The user must be able to slow down and speed up the rate at which video plays for more accurate annotation
Stage: alpha

R4.1.2.8 There must be a graph which displays interpolated annotated strokes as a function of time

a) If no data exists for the video being annotate then this graph is blank and is filled as annotations occur

b) If data exists this graph is filled with the data

Stage: alpha

R4.1.2.9 There must be a viewing mode for which the user may view an annotated video along with its corresponding graph

a) The video only starts playing after the user presses a button

Stage: alpha

R4.1.2.10 They system must allow the user to give it permission to search for videos to apply stroke annotation to streamline the annotation process
Stage: alpha

R4.1.2.11 All annotated data must be saved in a text file at the end of the annotation session
Stage: alpha

4.1.3. Functional Requirements: Position Data

This section details the requirements for the system that collects position data.

R4.1.3.1 The system must allow the user to label the position of a swimmer for every frame in a video of interest

a) key position points can be input and then the frames values between them can be interpolated

b) The distance between key points will be 0.5m

Stage: beta

R4.1.3.2 Once the annotation process is completed the resulting data can be reloaded and edited if needed and an updated position file will be created
Stage: beta

R4.1.3.3 Once the annotation process is completed a text file is produced holding the position of the swimmer of interest for every frame of the corresponding video
Stage: beta

R4.1.3.4 The system must allow the user to input the lane number of the swimmer of interest
Stage: beta

4.1.4. Non-functional Requirements

This section outlines the Non-functional Requirements of the data collection system.

R4.1.4.1 The API is executable on Windows operating system
Stage: alpha

R4.1.4.2 The API is executable on Linux and Mac OS operating system
Stage: production

R4.1.4.3 The data used and generated by the API must be secure and cannot be leaked
Stage: production

R4.1.4.4 The API would have Web GUI for better user readability.
Stage: production

4.2. Model Training System

This section outlines the model training system requirements.

4.2.1. Functional Requirements

This section outlines the functional requirements of the model training API.

R4.2.1.1 The API must have the ability to read input datasets
Stage: alpha

R4.2.1.2 The API must have the ability to distinguish training and testing datasets
Stage:beta

R4.2.1.3 The API must have the ability to train the model with training datasets

a) The API must allow the user train a model from scratch

b) The API must allow the user to update a model utilizing new data

Stage: alpha

R4.2.1.4 The API must have the ability to produce analytics results with testing datasets
Stage: beta

R4.2.1.5 A model is completed training when

a) at least 6000 iterations in total are completed

b) after meeting (a), the model continues until the metrics calculated by YOLO Darknet do not change

[12]
Stage: alpha

4.2.2. Non-functional Requirements

This section outlines the non-functional requirements of the model training API.

R4.2.2.1 The API is executable on Linux operating system and kernel.
Stage: alpha

R4.2.2.2 The training process must not take more than 72 hours
Stage: production

4.3. Real-time Swimmer Tracking System

This section outlines the requirements of the real-time swimmer tracking system.

4.3.1. Functional Requirements

This section outlines the functional requirements of the real-time tracking system.

R4.3.1.1 A Tilt functionality that moves the camera
Stage:alpha

R4.3.1.2 A Pan functionality that moves the camera
Stage: alpha

R4.3.1.3 The user of the system must be able to select a swimmer of interest to be tracked
Stage: alpha

R4.3.1.4 The user of the system must specify the stroke the swimmer of interest is swimming
stage: alpha

R4.3.1.5 A method of detecting swimmers in real-time in a pool setting

a) Detection model must use the 6 swimmer state classification to detect position of swimmer in the frame in real-time

b) Detection model must create box around the swimmer in the frame in real-time

Stage: alpha

R4.3.1.6 A method of temporally linking swimmer detections

a) Tracking model must interpret data from detection model

b) Tracking model must send command to camera to tilt or pan in such a way that centers the swimmer within the frame to a tolerance of 10% the frame size in pixels.

Stage: alpha

R4.3.1.7 At the completion of the race the system must produce a video of the swimmers race with a unique name.
Stage: alpha

R4.3.1.8 At the completion of the race the system must produce a text file with a name the same as the video produced by the system containing data

a) The data encodes the swimmers position in the frame of the corresponding video for every frame in the video

b) The data encodes if the swimmer is either taking a stroke or not for every given frame of the corresponding video

c) The data encodes the continues stroke rate (in strokes cycles per minute) of the swimmer as a function of frames

Stage: alpha

4.3.2. Non-functional Requirements

This section outlines the non-functional requirements of the real-time tracking system.

R4.3.2.1 This system must work in close to real time and the specified files must be produced before the start of the next event.
Stage: production

4.4. Swimmer Positioning System

This section outlines the requirements of the swimmer positioning system.

4.4.1. Functional Requirements

This section outlines the functional requirements of the swimmer positioning system.

R4.4.1.1 The positioning software must not lose the swimmer's position during the whole process
Stage: alpha

R4.4.1.2 The positioning software must estimate the position where the swimmer standing at the diving platform.
Stage: alpha

R4.4.1.3 The positioning software must estimate the position where the swimmer diving into water
Stage: alpha

R4.4.1.4 The positioning software must estimate the position where the swimmer is under the water surface in the midway
Stage: alpha

R4.4.1.5 The positioning software must estimate the position where the swimmer is beyond the water surface in the midway
Stage: alpha

R4.4.1.6 The positioning software must estimate the point where the swimmer makes a turn at one end
Stage: alpha

R4.4.1.7 The system must have the length of the pool as one source of input
Stage: alpha

R4.4.1.8 The system is given non-static video with tracking data of one swimmer from a single camera as one source of input
Stage: alpha

R4.4.1.9 The positioning software must handle MP4 or AVI format videos as input file
Stage: alpha

R4.4.1.10 The positioning software must output the 1D position of swimmer as function of time as output
Stage: alpha

R4.4.1.11 The positioning software must have a post processing sequence of the position of the swimmer that removes impossibilities such as non-continuous velocity and acceleration
Stage: alpha

R4.4.1.12 The positioning software must have a tendency analysis, like velocity, acceleration and deceleration rate of swimmer for performance analysis
Stage: alpha

R4.4.1.13 The positioning software must produce a text file with the same name as the input video.

a) The text file encodes the position of the swimmer as a function of frames

b) The text file encodes the velocity of the swimmer as a function of frames

Stage: alpha

R4.4.1.14 The positioning software must have an evaluation method that gives the accuracy of the system given the ground truth position of a swimmer
Stage: beta

4.4.2. Non-Functional Requirements

This section outlines the non-functional requirements of the swimmer positioning system.

R4.4.2.1 This positioning software must be faster than 20 minutes per race.
Stage: beta

R4.4.2.2 This positioning software shall respond within reasonable time
Stage: beta

R4.4.2.3 This positioning software shall be running in Linux OS environment.
Stage: beta

R4.4.2.4 Data between different categories must not corrupt or mismatch
Stage: beta

R4.4.2.5 The output files generated from the positioning software should be clear and easy-understanding
Stage: production

5. Other Non-functional Requirements

5.1. Safety Requirements

R4.4.1.1 Switch the system OFF before connecting them to a power supply.

R4.4.1.2 Disconnect and lockout the power supply before completing any maintenance work tasks or making adjustments.

R4.4.1.3 The system is not designed to be on the pool deck or in close proximity to the water.

R4.4.1.4 If you smell anything burning, immediately disconnect the power and contact us.

R4.4.1.5 Never try repairing the system with yourself.

R4.4.1.6 Be careful not to place or accidentally sprinkle some acid or alkaline things on the casing of its electronic products to avoid corrosion.

R4.4.1.7 Do not touch a person or electrical apparatus if an electrical incident happened. Always disconnect the power source first.

5.2. Sustainability Requirements

Sustainability issues for this product are limited as this project is mainly software based. The required hardware can be recycled at electronics recycling departments [13]. The biggest environmental concern is computational costs. Deep learning is exceptionally computationally expensive. In [14] the computational costs of models are measured in Trillions of floating point operations. As such the processing required could have an impact on the environment. There is not much that can be done in order to reduce this cost other than to aim for computational efficiency.

6. Acceptance Test Plan - Alpha

Table 6.1 details the acceptance test plan to be evaluated at the end of the ENSC405W semester. The first row details a system of interest and the other details the necessary items each system must complete in order to be considered done.

Table 6.1

Test Plan	Acceptance Criteria
Validate the swimmer detection data annotation system	- A user is able to open the application on Windows - A user is able to open a video to do data annotation with - A user is able to set the annotation frame rate after opening the video - A user is able to change and track swimmer's position and classes in the video during the annotation - The annotated data is produced upon completion
Validate the swimmer stroke data collection system	- A user can successfully annotate the position of a swimmers stroke as the video plays - A user is able to move forward and backwards through the video at an adjustable speed while annotating - A user can successfully clear annotations done on a video when entering edit mode while data exists for that video - When annotating, a user can specify if a swimmer is swimming or not as well as which stroke is being performed - For an annotated video, a user can view the graph of the sinusoidal signal and access it's frequency
Validate the model training system	- A user is able to input data-sets and the API has the ability to read them - After reading the input source given from the user, the API is able to distinguish training and test data-sets - After reading the input given source, The API has the ability to produce analytic results with testing data-sets - The API has the ability to train the model with given training data-sets

Continuation of Table 6.1	
Test Plan	**Acceptance Criteria**
Validate the camera control	- The camera should perform tilt functionality - The camera should perform pan functionality - The camera should perform real-time image capturing and video recording functionality - The camera should perform correct movements corresponding to the commands from the tracking system - The error of the movements should be in a reasonable calculation range
Validate the real-time system	- The real-time tracking system should let the user to select a swimmer of interest to be tracked - The detection model should classify different state of the swimmer in the frame in real-time, and comparing the result with correct sample - The detection model should create box around the swimmer in the frame in real-time, and comparing the result with correct sample - The tracking model should send correct commands to the camera in order for performing correct movement - The system should generate various correct output files with unique names

Continuation of Table 6.1	
Test Plan	**Acceptance Criteria**
Validate the positioning system	- A user is able to follow the swimmer's position during the whole process - A user is able to view the position where the swimmer standing at the diving platform. - A user is able to view the position where the swimmer diving into water - A user is able to view the position where the swimmer is under the water surface in the midway - A user is able to view the position where the swimmer is beyond the water surface in the midway - A user is able to give non-static video with tracking data of one swimmer from a single camera as one source of input - After reading the input source, the system is able to represent a post processing sequence of the position of the swimmer that removes impossibilities such as non-continuous velocity and acceleration - After evaluating input source data, the system is able to output a tendency analysis, like velocity, acceleration and deceleration rate of swimmer for performance analysis - After evaluating input source data, the system is able to output the accuracy of the system given the ground truth position of a swimmer
End of Table 6.1	

Table 6.1.: The table for the acceptance test plan to be presented at the end of the ENSC405W

7. Engineering Standards

The product, Swimming Analytics, is to be implemented based on following engineering standards.

- **ISO/IEC TR 24028:2020 Information technology** — Artificial intelligence — Overview of trustworthiness in artificial intelligence [15]
- **ISO/IEC TR 24029:2021 Artificial Intelligence (AI)** — Assessment of the robustness of neural networks [16]
- **ISO/IEC 27001:2013 Information technology** — Security techniques — Information security management systems — Requirements [17]

8. Conclusion

Smart Swim Analytics' Automated Swimming Analytics System will quickly provide swimmers with accurate metrics to measure their performance as defined by the requirements presented in this document. Users will perform annotations via the API outlined in this document which will be used for training the model. The model will then be used with the tilt-pan hardware, a camera and the Jetson board to accurately determine each swimmer's stroke length and stroke rate. This will help determine each swimmer's speed much more efficiently and cheaply than it would be if done by hand as it currently is. The system's lack of reliance on consistency in the environment ensures it works in a wide variety of settings (different styles of swimming, different length pools, etc). The requirements outlined in this document have defined which aspects of the product will be completed in the proof of concept (by August 2021) and which are to be present for the final product (by December 2021). We have also addressed sustainability and safety concerns, and acknowledged Engineering Standards that our product must meet.

Bibliography

[1] W. Woinoski, "Automated swimming analytics using deep neural networks," *SFU Thesis Library*, 2020, unpublished.

[2] RaceTeck, "Raceteck," 2018, [Online], Available: https://racetek.ca/ [Accessed: Nov 19 2019].

[3] S. U. Swim USA productions, "Usa swimming," 2009, [Online], Available: https://www.youtube.com/user/USASwimmingOrg/featured [Accessed: Sept 10 2019].

[4] J.-P. Eisenbarth, "Srs-ieee," 2016, [Online], Available: https://github.com/jpeisenbarth/SRS-Tex [Accessed: June 06 2021].

[5] F. Form Swim, "Form," 2019, [Online], Available: https://www.formswim.com/ [Accessed: Nov 19 2019].

[6] Tritonwear, "Tritonwear, take the guesswork out of swimming faster," 2015, [Online], Available: https://www.tritonwear.com/ [Accessed: Nov 19 2019].

[7] C. Hudson, "Automated tracking of swimmers in the clean swimming phase of a race," Ph.D. dissertation, Sheffield Hallam University, 2015.

[8] A. Hall, B. Victor, Z. He, M. Langer, M. Elipot, A. Nibali, and S. Morgan, "The detection, tracking, and temporal action localisation of swimmers for automated analysis," *Neural Comput. Appl.*, pp. 1–19, 2020.

[9] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

[10] NVIDIA, "Embedded system with jetson," 2014, [Online], Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/ [Accessed: June 10 2021].

[11] J. Frye, "Lynxmotion, imagine it. built it. control it," 1995, [Online], Available: http://www.lynxmotion.com/ [Accessed: June 10 2021].

[12] AlexeyAB, "When should i stop training," 2014, [Online], Available: https://github.com/AlexeyAB/darknet#when-should-i-stop-training [Accessed: Jun 13 2021].

[13] R. It, "Electronics recycling," 1994, [Online], Available: https://www.return-it.ca/electronics/ [Accessed: June 13 2021].

[14] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[15] ISO and Comity, “Information technology — artificial intelligence — overview of trustworthiness in artificial intelligence,” International Organization for Standardization, Geneva, Switzerland, Standard, 2020, [Online], Available: https://www.iso.org/standard/77608.html [Accessed: Jun 13 2021].

[16] ——, “Artificial intelligence (ai) — assessment of the robustness of neural networks,” International Organization for Standardization, Geneva, Switzerland, Standard, 2021, [Online], Available: https://www.iso.org/standard/77609.html [Accessed: Jun 13 2021].

[17] ——, “Information technology — security techniques — information security management systems — requirements,” International Organization for Standardization, Geneva, Switzerland, Standard, 2013, [Online], Available: https://www.iso.org/standard/54534.html [Accessed: Jun 13 2021].

[18] FINA, “Fina facilities rules.” [Online]. Available: https://www.fina.org/sites/default/files/2017_2021_facilities_06102017_full_medium.pdf[Accessed:2020-07-22]

A. Background on Swimming

For the purpose of this work [1], the aspect of swimming we are concerned with is speed and the strokes the swimmers take. The swimmer who can legally swim the specified race distance in the least time possible is the winner of the race. Swimming races are very controlled and thus a competition can take place where many races of the same event can occur in sequence. In order to compare two athletes, the time taken to complete the event is all that is required to rank the two swimmers. In fact swimming is so controlled that swimmers may be ranked across all occurring races at any given time and at any pool around the world and in some cases even when the pool is not the same length. This is in contrast to other sports like marathon events or ski jumping in which the ability of two athletes can only be quantified in a consistent manner when the athletes are at the same competition. Due to the very controlled environment swimming competitions require, many assumptions can be made in terms of pool size and swimmer position.

A.1. Pools

Swim competitions happen in many different pools and environments. Around the world, there are three main competition pool lengths in swimming, long course meters (LCM), short course meters (SCM) and short course yards (SCY). LCM pools are 50 meters long, SCM pools are 25 meters long and SCY pools are 25 yards long. Furthermore, pools can have different numbers of lanes. In each race every lane does not always contain a swimmers. More about pools can found in [18].

A.2. Races

Swim racing has four strokes: butterfly (fly), backstroke (back), breaststroke (breast), and freestyle (free). These strokes all fall under the category of swimming. They are relatively easy to distinguish on a camera and can easily be identified by a human.

Each race starts with a dive. This is when the swimmer is positioned on an elevated starting block and then jumps through the air to enter the pool. This section is generally a very small fraction of a race as it only happens once and takes very little time. This action can be troublesome for automation as the swimmers position on the blocks (bent over) is very different than when they are in the water (stretched out) and due to high speed of the swimmer.

The next part of a race is the underwater sections. In the early 1980s it was shown that swimmers can be much faster underwater than on the surface, especially when pushing off the wall during a turn and after the dive. A considerable portion of a swimming

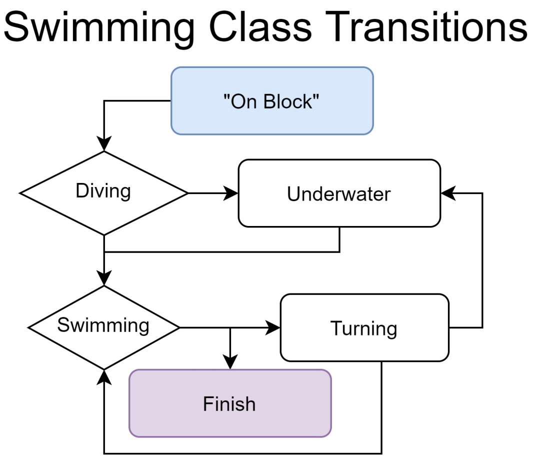

Figure A.1.: Oder of swimming operations in a race

race involves the swimmer underwater. This part of a race is problematic for automated swimming analytics. When swimmers are underwater they can be very difficult to see from an above-water camera. This is due do to the phenomenon of refraction, where light is bent because of the difference in densities between two substrates. This phenomenon can make a swimmer invisible for periods of time, creating what is known as occlusion. Another occluder of swimmers underwater is lane ropes – the plastic floating ropes separating swimmers in a race.

The next portion of the race is the turn action. The goal of a turn is to transition back to swimming when the swimmer reaches the end of a length. Turns can take many forms based on the event being preformed. After a turn, a swimmer usually ducks underwater and starts a new underwater section. The turn is problematic as the white water produced by the swimmers causes occlusions of the boundaries of the swimmer. This can make it difficult to localize the swimmer's position.

Between the underwater portion and turning portions of the race is the swimming. As mentioned before, there are four strokes that can occur, but only once per length. Possibly one of the easiest parts of the race to identify a swimmer is when they are swimming; it is also the most prevalent. There are a few reasons why a swimmer would be hard to recognize in this portion, but it is mainly due to occlusions. These are caused by people or the sets of ropes hanging over the pool designed to warn a swimmer of an approaching wall.

Finally, a race ends with a finish. This simple action transitions the swimmer from swimming to finish and is easy to identify. Once the swimmer is finished they generally stay in the same place until the rest of the swimmers finish, marking the end of a race.

In general, the order of operation in a races is as follows: on-blocks, dive, underwater, swim, turn, underwater, swim and finish. However, there are varying numbers of swim, turn, underwater and finishes depending on the race category (event) – see Figure A.1

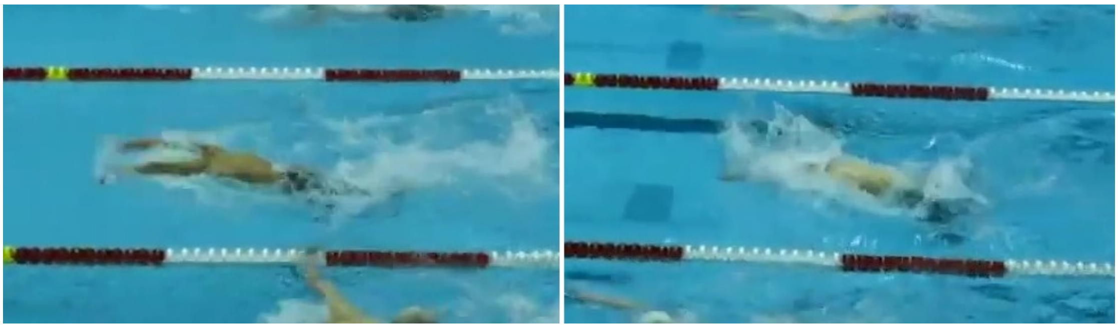

Figure A.2.: An example of Fly and Breast, respectively [3]

for a complete state transition diagram.

In addition to the technicalities of swimming, events can have different distances and styles. Such distances range from lengths of 50 meters to lengths of 1,500 meters. Some styles have swimmers preform all four strokes in one race, such as the individual medley (IM), or only one stroke at a time. Swimming has 18 different styles of individual events and another 5 team events known as relays. All the different positions, transitions, actions, pool lengths, pool environments and pool lane numbers for each event must be taken into account for when creating a fully functioning automated swimming analytics system.

A.3. Stroke Styles Definitions

As mentioned in Section A.2 there are four strokes: fly, back, breast, and free. Each of the four strokes produce propulsion, which moves the swimmer though the water. The propulsion is caused by both the arms and the legs of the swimmer. While the legs produce considerable power and increase the buoyancy of a swimmer in the water, swimming analytics, at this point, is mainly concerned with the arms of a swimmer. This is because arms are more straightforward to correlate to faster swimming and also because the arms are generally responsible for approximately 80% of a strokes propulsion. Furthermore, for each of the four strokes, a stroke is counted as a stroke cycle. Because each stroke has its own definition of a stroke cycle, we shall briefly go over stroke cycle definitions.

In terms of stroke cycles, the four strokes can be split into two categories for the purposes of stroke recognition. The first category is asymmetric strokes, seen in Figure A.3, which are the back and free styles of swimming. They are denoted as asymmetric as the stroke is not symmetric through the human sagittal plane. In terms of a stroke cycle, one arm pulls past the body while the other is recovering above the water allowing for strokes to be taken by each arm individually in quick succession. Thus, stroke cycles for free and back are completed when one of each arm is pulled and recovered past the body.

The second category of strokes is symmetric, seen in Figure A.2, which consists of the

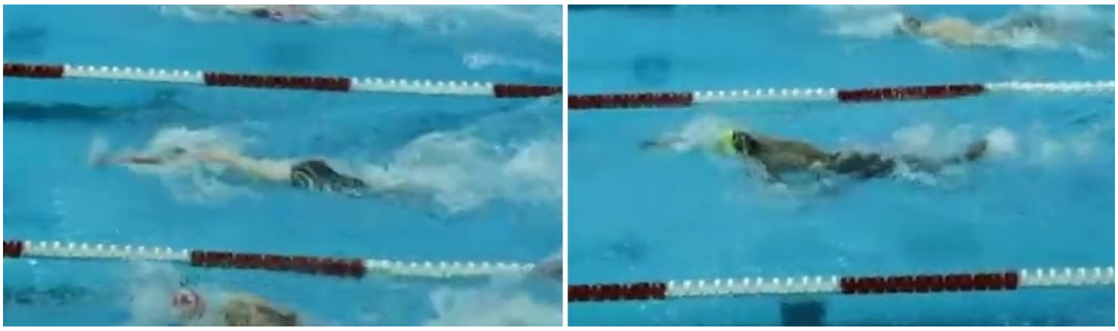

Figure A.3.: An example of Free and Back strokes, respectively [3]

fly and breast styles of swimming. As can be expected, this is because these strokes are symmetric across the human sagittal plane. Stroke cycles are more straightforward in this category because in each stroke, both arms are pulled past the body and recovered in unison. So, stroke cycles for fly and breast are counted for each stroke pull and recovery.

Understanding this subject matter will prove to be useful when considering the issues being solved in this work. Understanding how a race occurs, or how a swimmer swim, allows one to understand the challenges being faced. This section will continuously be referred to throughout this work when exploring challenges and solutions.