Software Requirements Specification

for

MotionTracker

Version 1.3

Prepared By:

Humaid Mustajab, Samuel Tsui, Sean Rhee, Logan Kramsky, Johnson Wu, Evan Pugh

Advisor:

Prof. Jeff Salvage

Stakeholder:

Prof. Jeff Salvage

Table of Contents

Table of Contents **2**

1. Document History **3**

2. Introduction **4**
2.1 Purpose 4
2.2 Overview 4
2.3 Scope 4
2.5 Requirements Apportioning 5

3. User Interface and Functional Requirements **6**
3.1 Program Flow 6
R3.1 Requirements of the Program Flow **6**
‘Screen 1’ Requirements 6
‘Screen 2’ Requirements 7
‘Screen 3’ Requirements 8

4. Non-functional Requirements **10**
4.1 Human Factors 10
4.2 Documentation 10
4.3 Hardware 10
4.4 Software 10
4.5 Performance 10
4.6 Error Handling and Reliability 11
4.7 Security 11

5. System Evolution **12**
5.1 Introduction 12
5.2 Context Aware Crop and Zoom 12

1. Document History

Name	Date	Reason	Version
Humaid M, Samuel T, Sean R, Logan Kramsky, Johnson Wu, Evan Pugh	Jan 16, 2023	Initial Draft	1.0
Humaid M, Samuel T, Sean R, Logan Kramsky, Johnson Wu, Evan Pugh	Jan 24, 2023	Revised Draft 1	1.1
Humaid M, Samuel T, Sean R, Logan Kramsky, Johnson Wu, Evan Pugh	Jan 31, 2023	Revised Draft 2	1.2
Humaid M, Samuel T, Sean R, Logan Kramsky, Johnson Wu, Evan Pugh	Apr 18, 2023	Final	1.3

2. Introduction

2.1 Purpose

This document provides the requirement specifications for the application MotionTracker. It specifies user interface attributes, functional and nonfunctional requirements, and long-term ideas for the evolution of the system.

2.2 Overview

Race walking is an internationally recognized Olympic event with strict rules enforced by trained judges. Recorded footage from previous races is used as a tool for training judges to showcase rule-following and rule-breaking actions. The duration of a race walk event ranges anywhere from 6 minutes to 8 hours. Due to the long nature of these events and varying lengths of tracks, there are many hours of dead space in the video footage where no competitor is passing by the camera.

The Motion Capture system deletes dead space from race footage to make training judges a simpler and more effective task. Users upload their race footage, and through motion detection algorithms, the system deletes all dead space and gives the user back a new file which contains a clip containing footage where race walkers are present.

2.3 Scope

This document describes the software requirements for the MotionTracker project. The initial release is tailored to be a proof of concept or a minimum viable product against the requirements. This document is intended for end-users and developers interested in the application’s functionality and future evolution.

2.5 Requirements Apportioning

Priority Level	Description
1	Priority 1 items have the highest level of importance and are necessary for the usability of the product. The items labeled Priority 1 are completed and verified prior to the release of the application.
2	Priority 2 items are not crucial to the usability of the application, but would enhance the user experience and are expected to be implemented in the next release of the application. Priority 2 items not completed prior to release do not inhibit functionality of the release version of the application.
3	Priority 3 items are not crucial to the usability of the application and are not within the current scope of the system. These items are expected to change with future development

3. User Interface and Functional Requirements

3.1 Program Flow

MotionTracker is an application that compresses footage of race walking events to only contain frames where race walking competitors are present. When users launch the application they are prompted to select a video file from their local storage to be processed. After the user makes a selection, they are prompted to input selected parameters including the range of video height and width pixels for the area of interest per frame, the desired frames per second for the output video, and the desired rescaling ratio for the output video. After users submit their parameters, the application processes and outputs a new clip following the user-specified parameters with only frames with race walking competitors from the original clip selected.

R3.1 Requirements of the Program Flow

Screen 1

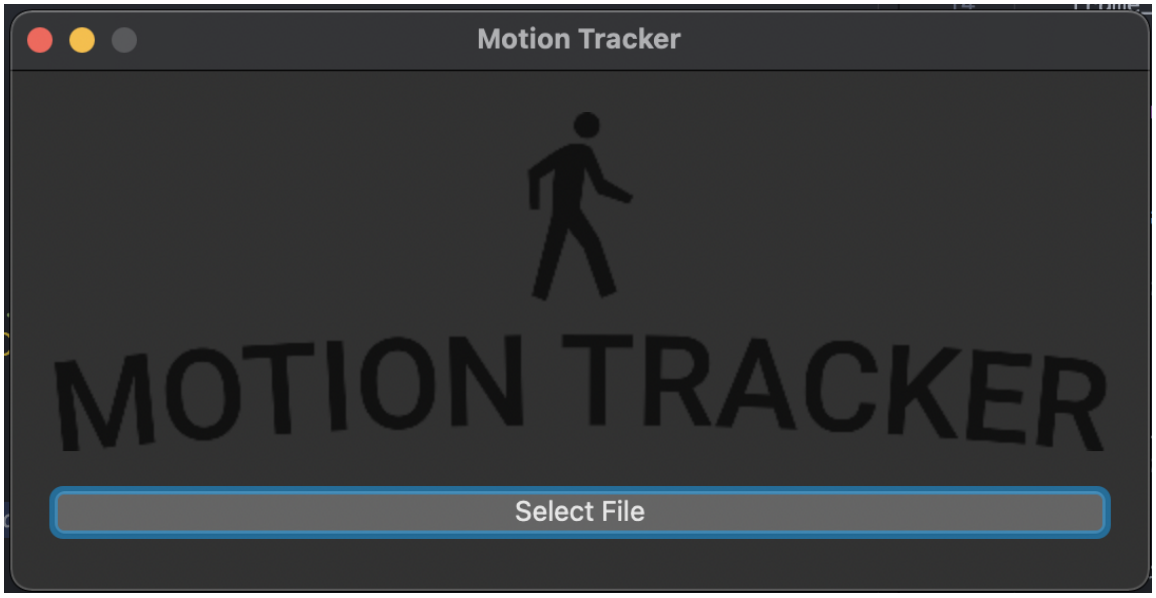


Figure 1. ‘Screen 1’

‘Screen 1’ Requirements

3.1.1 Upon launching the application, the user must choose the ‘Select File Button’ seen in Figure 1 **Priority 1**

3.1.2 The user navigates their local file structure and selects a video file **Priority 1**

Screen 2

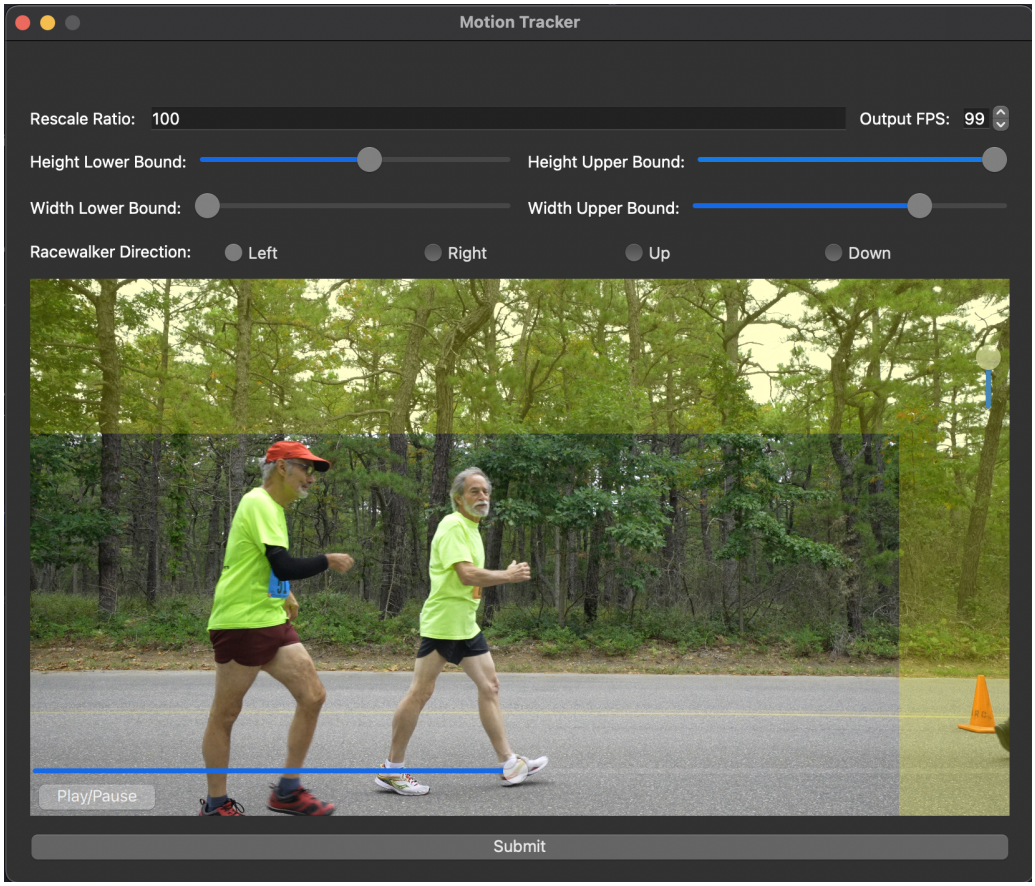


Figure 2. ‘Screen 2’

‘Screen 2’ Requirements

3.1.3 The user is prompted to enter input parameters for a range of video height/width pixels for the area of interest per frame

- **3.1.3.1** The user enters values within the range of the original clip’s height and width **Priority 1**
- **3.1.3.2** User is re-prompted if the value is not an integer or within range or if lower bound is greater than upper bound
- **3.1.3.3** The user is able to see the range affecting the video frame using the UI

3.1.4 The user is prompted to input an integer for the desired frame rate of the output video

- **3.1.4.1** The default value is the native framerate of the selected clip **Priority 1**
- **3.1.4.2** If the user decides to change the default value, their input should be an integer value > 0 **Priority 1**

3.1.5 The user is prompted to input an integer for a rescaling ratio for the output video

- **3.1.5.1** User’s input value should be less than or equal to 100 **Priority 1**

3.1.6 Video player begins to play selected clip

3.1.7 User must be able to click ‘Submit’ to begin video processing **Priority 1**

3.1.8 The user must be able to pick the direction of motion to be filtered using the UI. The directions can be Left to Right, Right to Left, Top to Bottom, or Bottom to Top.

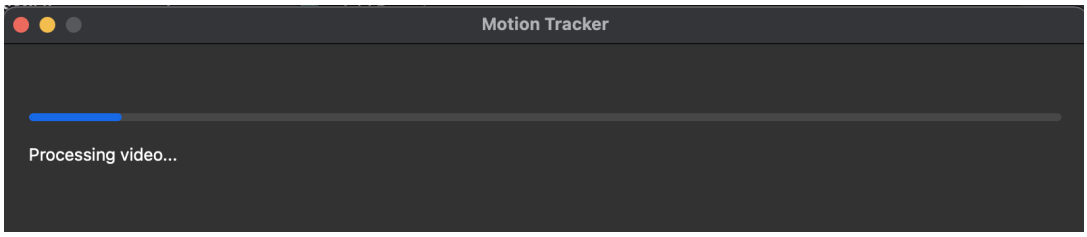
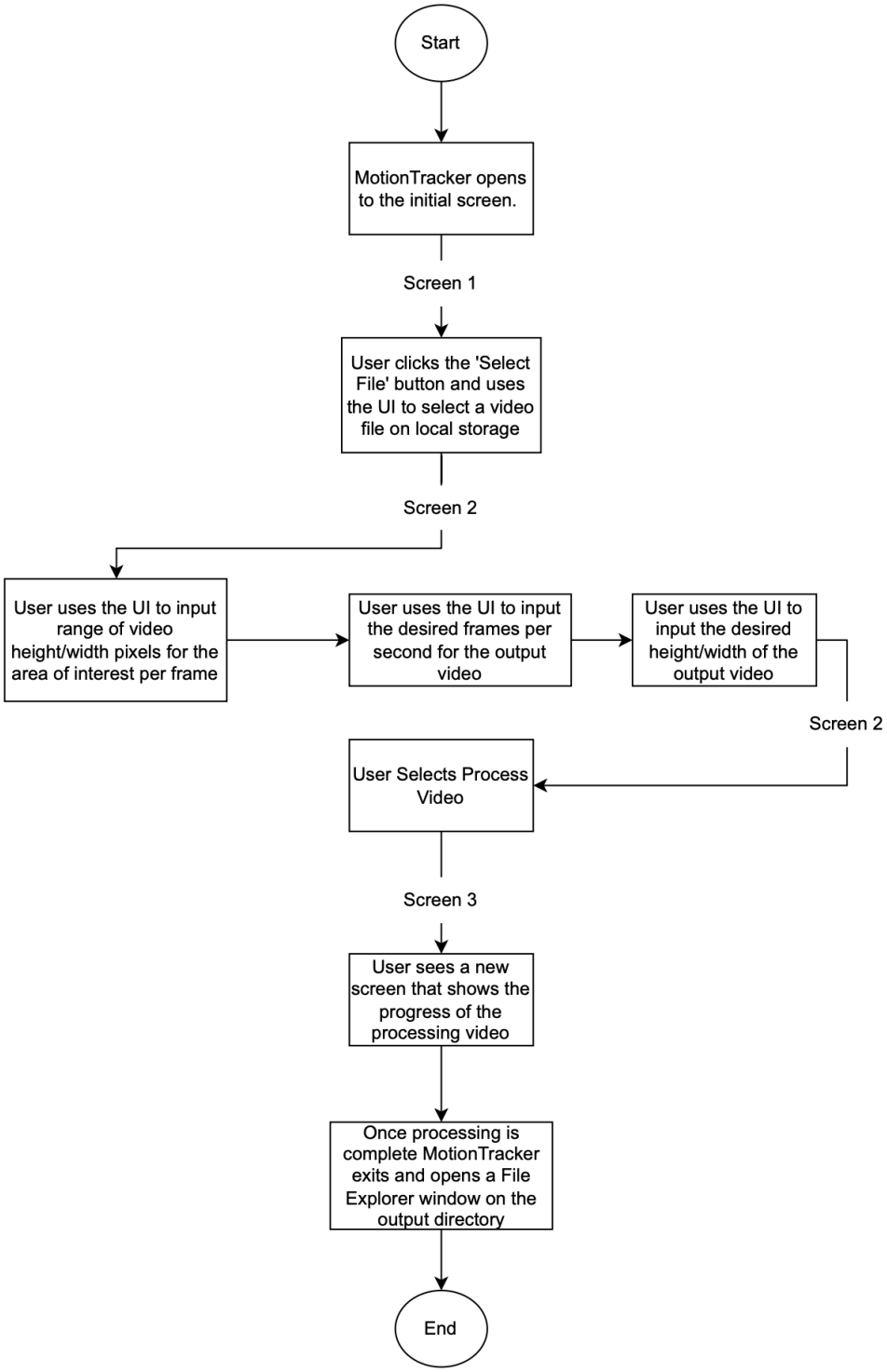


Figure 3. ‘Screen 3’

‘Screen 3’ Requirements

3.1.9 A progress bar is displayed on the UI to represent the progress of the video processing
3.1.10 The user is notified when the processing is completed
3.1.11 MotionTracker exits upon completion of processing
3.1.12 MotionTracker opens a File Explorer window on the output directory


Start
MotionTracker opens to the initial screen.
Screen 1
User clicks the 'Select File' button and uses the UI to select a video file on local storage
Screen 2
User uses the UI to input range of video height/width pixels for the area of interest per frame
User uses the UI to input the desired frames per second for the output video
User uses the UI to input the desired height/width of the output video
Screen 2
User Selects Process Video
Screen 3
User sees a new screen that shows the progress of the processing video
Once processing is complete MotionTracker exits and opens a File Explorer window on the output directory
End

4. Non-functional Requirements

4.1 Human Factors

4.1.1 Types of Users

- **4.1.1.1** Users with no experience with motion-tracking or video editing experience.
- **4.1.1.2** Users with beginner to expert level experience with video editing.

4.2 Documentation

4.2.1 Application Usage Documentation

- **4.2.1.1** Documentation is provided for usage of MotionTracker with explanation for all required parameters.

4.2.2 Application Architecture Documentation

- **4.2.2.1** Documentation explaining the packages used for different subsystems on the MotionTracking application is provided.

4.3 Hardware

4.3.1 The user must have a computer with the following hardware:

- **4.3.1.1** Keyboard
- **4.3.1.2** Mouse
- **4.3.1.3** Monitor

4.3.2 The users system should meet the following minimum requirements

- **4.3.2.1** Processor: 1 gigahertz (GHz) or faster processor or SoC
- **4.3.2.1** RAM: 1 gigabyte (GB) for 32-bit or 2 GB for 64-bit
- **4.3.2.1** Hard disk space: 16 GB for 32-bit OS or 20 GB for 64-bit OS
- **4.3.2.1** Graphics card: DirectX 9 or later with WDDM 1.0 driver
- **4.3.2.1** Display: 800 x 600

4.4 Software

4.4.1 The MotionTracker software must run locally using an distributable executable for Windows and macOS systems without any package requirements.
4.4.2 The MotionTracker application is expected to handle mp4, avi, mov, and mkv file extensions.

4.5 Performance

4.5.1 Time to an output video will be dependent on the resolution of the video, the length of the video, and on the number of frames with detected motion.
4.5.2 A packaged executable is provided for users for Windows 10, 11 and macOS >= 12.
4.5.3 Any OS that can support python >= 3.10 is able to compile and run the application from source code.

4.6 Error Handling and Reliability

4.6.1 Input Errors

4.6.1.1 When invalid inputs are set for the resolution scale or the output fps (ex: characters, unreasonably small/large numbers, etc.), the GUI will not let you proceed.

4.6.1.2 When an invalid file path is set, the GUI will not let you proceed.

4.7 Security

4.7.1 The Motion Tracker software runs locally without the need for an internet connection, and as such network security was not the main objective during development.

5. System Evolution

5.1 Introduction

To focus on the development of the object detection model we decided to limit the time spent on adding UI features and exposing parameters that allow a user to customize the detection model and how the program deals with the input video. Since this project is built using open-source technology, users with sufficient knowledge of opencv and object tracking/detection are welcome to add in new features as they deem possible to augment their workflow. We are including a few examples of feature requests received that can improve the UX of the application.

5.2 Context Aware Crop and Zoom

A feature to crop the video frame and follow a boxed boundary around the detected area through multiple frames would be an improvement over the current feature set of the application. It would work similarly to Apple's Center Stage feature of iMac desktop computers where it zooms into and centers the person within the video frame.