NTNU
Norwegian University of
Science and Technology

Computer Vision For Aimpoint Tracking In Biathlon

Bjørn-Ivar Bogfjellmo Haug

Master of Science in Cybernetics and Robotics
Submission date: June 2018
Supervisor: Tor Engebret Onshus, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem description

Related to practice sessions for biathletes it is desired to develop a system that can display the movement of the point of aim (POA) during shooting practice. Together with other systems providing quantifiable data regarding the athletes performance, this will help improve the quality of the practice sessions hence aim to improve the performance of the athletes.

This master thesis aims to use computer vision to track the POA. The developed computer vision system shall work outdoors during a wide range of weather conditions, and successfully track the POA with a high level of precision and robustness. The system shall be developed to run on a small embedded system, and as such processing power considerations will be undertaken.

Abstract

The purpose of this thesis was to look at the development of a computer vision system for aimpoint tracking in biathlon. It's desired to develop a system that tracks the Point-of-Aim of the biathlete in order to give quantifiable feedback during and after shooting practice to accelerate skill acquisition.

This report covers the process of developing a prototype computer vision system, from the initial idea to a working prototype implementation and system evaluation. It was found that the usage of the Circular Hough Transform (CHT) algorithm gave high precision on a wide range of video resolutions and is suitable for future implementations. It came at the cost of high computation demands, and reaching high update frequencies was difficult. 13 FPS was achieved on the desktop implementation when analyzing a 640x424 video, while 50 Hz were obtained on a 216MHz ARM Cortex M7 analyzing a 20x120 video source. Feature tracking based on establishing a region-of-interest within the frame was proved to be successful. A successful aimpoint tracking output was obtained and is presented.

Camera and processing power specifications are identified and presented for future implementations. The usage of Matlab as a development environment is discouraged for further work on the project, and the switch to a OpenCV-based implementation is recommended.

Sammendrag

Målet for denne masteroppgaven var å utvikle et system for følging av siktepunkt i skiskyting. Det er ønskelig å utvikle et system som følger utøverens siktepunkt slik at man kan gi kvantifiserbare tilbakemeldinger under og etter trening.

Denne rapporten omfatter prosessen for utviklingen av datasynsystemet, fra den initielle idéen til en fungerende prototype samt evalueringen av systemet. Resultatene viste at bruken av Hough transformasjon oppnådde høy presisjon på et vidt spekter av ulike oppløsninger og er passende for fremtidige revisjoner av systemet. Presisjonen kom med en bakside bestående av høye krav til regnekraft, og å nå kravet til oppdateringsfrekvens viste seg vanskelig. 13 bilder i sekundet ble oppnådd på PC med en 640x424 videofil som input, mens 50 bilder i sekundet ble nådd med en 216MHz ARM Cortex M7 med en 20x120 videofil som input. Målfølging basert på et interesseområe i bildet viste seg effektivt. En vellykket siktepunktfølger ble implementert og resultatet kan sees i rapporten.

Kamera- og regnekraft-spesifikasjoner ble identifisert og er presentert for bruk i fremtidige iterasjoner av systemet. Bruken av Matlab som utviklingsverktøy i fremtidige iterasjoner er ikke nødvendig, og det er anbefalt å begynne å utvikle og bruke en OpenCV-basert implementasjon i stedet for å nå kravet vedrørende oppdateringsfrekvens.

Preface

Before you lies my thesis, 'Computer Vision for Aimpoint Tracking in Biathlon', a thesis submitted as a fulfillment of the 5 year integrated Master of Science in Engineering Cybernetics, Department of Engineering Cybernetics, Norwegian University of Science and Technology. The initiator of the project for which this thesis is written was Centre for Elite Sports Research (SenTIF) and the work was performed in cooperation between the Department of Engineering Cybernetics and SenTIF.

I had no previous experience of working with computer vision, and as such this has been a journey in unexplored territory for the author. I have however previously had the honor of building two Formula Student race cars through the student project Revolve NTNU and gained valuable experience in system engineering and design which certainly helped when developing prototype systems for this thesis. This project started from scratch and was as such not based on any previous work. It has also been undertaken as an individual exercise with minimal guidance from others.

The results of this thesis were presented to the initiator, SenTIF. The results were very well received and I would like to thank SenTIF, and especially Harry Luchsinger, for their enthusiasm and support along the way. I would also like to thank my supervisor, professor Tor Onshus, for valuable academic guidance. I also extend my gratitude to my parents for not raising a quitter and supporting me in everything I do.

Contents

List of Tables

List of Figures

Abbreviations

.22lr .22 long rifle caliber

BATU Biathlon Aimpoint Tracking Unit

CHT Circular Hough Transform

DSLR digital single-lens reflex camera

HQ High Quality

IMU Inertial Measurement Unit

KLT Kanade-Lucas-Tomasi

LQ Low Quality

NIR Near-Infrared

NTNU Norwegian University of Science and Technology

PoA Point-of-Aim

PoI Point-of-Impact

ROI Region of Interest

SenTIF Centre for Elite Sports Research

VPU Visual Processing Unit

Chapter 1

Introduction

This introductory chapter will briefly provide context for the material/results presented in this report and give motivation and a description of the problem to be solved. A literature review summarizes existing relevant knowledge. Then the contributions of the report are identified, and at the end an overview of the report is presented.

1.1 Motivation

SenTIF is a part of the Norwegian University of Science and Technology (NTNU) dedicated to elite sports research. A major part of this mission is to analyze and identify how top tier athletes perform as they do, and quantify performance. In order to quantify performance various systems are in use, and SenTIF are looking to expand their capabilities within the area. This master thesis is written as part of a SenTIF project aimed at improving data-gathering and analysis capabilities for Biathlon research.

Biathlon is a Nordic winter sport combining cross-country skiing and precision shooting. The athletes both ski and shoot during the same competition. This means athletes shoots with a highly elevated hart rate, fast paced breath and lactic acid build-up in their muscles. These conditions, together with the rules of the sport, makes proper shooting techniques vital in order to succeed. To properly quantify shooting technique is therefore of high interest and a goal for this project, where computer vision will be explored for usage as a aim-point tracking technology in Biathlon.

In addition to this thesis several other master theses and project theses have been and are currently being done for SenTIF:

- Inertial Measurement Unit (IMU) attached to the athletes leg for measuring cross-country skiing technique
- IMU attached to the rifle measuring rifle movement [1]
- Measurement of trigger pull. [2]
- Hardware for optical aimpoint tracking of biathlon rifles [3].
- Central hardware unit for wireless transmission and logging of data from the above-mentioned units to a central stationary unit. [4]

Together these projects forms the foundation for SenTIF's biathlon outdoor-lab vision where the scientific "lab"-activities can be done outdoors in the athletes natural habitat, not in an indoor lab which is widely done today [5]. Alongside introducing new possibilities in data gathering, the goal is to create more user-friendly and less costly products than what's currently on the market, giving Norway a competitive edge in biathlon.

1.2 Literature overview

Fundamental knowledge about computer vision and embedded systems are required. Sources as Forsyth and Ponce [6] covers camera models and camera calibration techniques. Gonzalez and Woods [7] explains basic image process operations, and [7] as well as Prince [8] both covers feature detection. Mattfeldt [9] explains different camera technologies and their usage. Kisačanin et al. [10] merges computer vision with embedded design, where Kolsch [11] covers hardware considerations for embedded vision systems and Saha [12] design methodologies.

Theoretic background on computer vision is not further presented other than the sources listed in this section and assumed known by the reader. This thesis looks at and explores the usage of known computer vision theory and algorithms deployed in a biathlon environment.

1.3 Assumptions

The scope of the thesis has mainly been restricted by the available time. Starting from scratch, foreseeing how much time individual tasks actually would take was a major challenge. Validation of the plausibility of the project idea took more time than anticipated, and as such limited project progress. Especially challenges related to real life testing heavily depending on equipment and weather conditions proved to be quite time-demanding, and surely a valuable experience for future work - both for this and other projects.

1.4 Contributions

The main contribution of this thesis is the foundation work done facilitating further development. The feasibility of the project idea has been verified and the conceptual choices proven to work. Answers to major uncertainties regarding precision and update frequencies have been answered, as well as identifying camera, lens and minimum computing power requirements.

The cooperation with SenTIF was good, and the final results of this thesis were presented for SenTIF in Granåsen on Thursday June 7th. They were satisfied with the results [5], and showing the value of interdisciplinary cooperation with technical departments within NTNU encouraged further cooperation with the Department of Engineering Cybernetics.

1.5 Outline

The report is organized as follows. In Chapter 2 background material is presented. This includes information about biathlon, existing systems and requirements for the system developed in this thesis. Chapter 3 covers conceptual approaches and chosen design concepts for implementation in Chapter 4. Chapter 4 gives a brief overview of the implementation before describing test methodology and the test-setups used. In Chapter 5 results are presented and discussed. Chapter 6 contains the conclusion and provide pointers for future work.

Chapter 2

Background

This chapter gives a description of the system and the given requirements, as well as an overview of existing market solutions.

2.1 Biathlon

Biathlon is a Nordic winter sport combining cross-country skiing with precision shooting. It alternates between skiing a given distance and then entering a stadium with a target range for shooting, and then skiing again. The distance skied and number of stops at the target range varies between different exercises within Biathlon. Distance to target, number and size of the targets is however always the same. The distance is $50m$, there's 5 circular black targets against a white background and the diameter of the target is $115mm$. A picture of a biathlon target can be seen in figure 2.1.

When shooting in the standing position, the target diameter is the entire circle with a diameter of $115mm$. In the prone position, the target is the inner circle with a diameter of $45mm$. The formal specifications for a biathlon target can be seen in figure 2.2. A picture of a biathlete shooting from the prone position can be seen in figure 2.3 [13].

Figure 2.1: Picture of a standard biathlon target taken at SenTIF

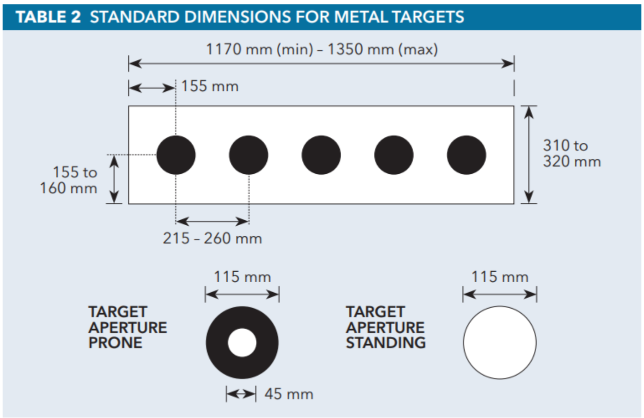


Figure 2.2: Official target specifications

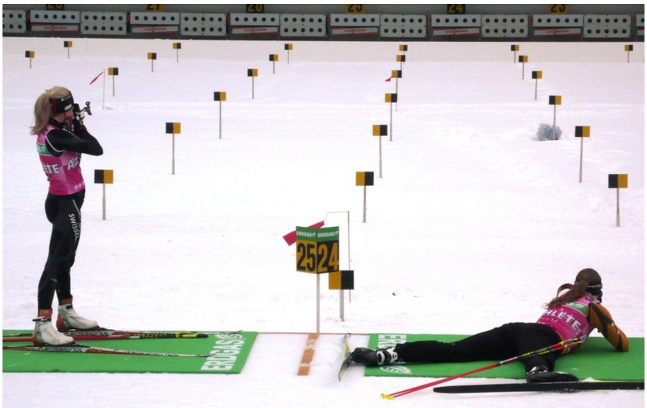


Figure 2.3: Biathletes shooting from the standing and prone positions [14].

2.2 Existing systems

Given the quantity of shooting sports world wide, the market for solutions offering increased shooter feedback is naturally large and there's already similar products on the market. A market research was conducted and the products can be classified and distinguished with the features in Table 2.1.

Table 2.1: Product classifiers

Feature	Explanation
Camera technology	Visible light or other
Environment	Indoors, outdoors, both
Communication	Cabled or wireless
Target compatibility	ie. only dark circles or other types
Distance to target	fixed, dynamic, if dynamic - what range
Precision	Precision of measurements
Resolution	Measurement frequency (Hz)

Given the characteristics of the Biathlon target in Fig. 2.1 two main systems were identified for Biathlon usage: Optel and SCATT.

2.2.1 Noptel

Noptel is Finnish company specialized in Optoelectronic products and SenTIF has a unit of the Noptel Sport II [15]. The Sport II relies on non-visible light and requires a reflector unit to be mounted to the target. It has a range of 3 to 10 meters, although a version with range of 50 meters exists. The unit is connected to a computer with a cable and can be used both indoors and outdoors. Sport II can only track one target at a time.

A Noptel Sport II unit can be seen in the top left corner of figure 2.4.

[redacted]

Figure 2.4: Overview of Noptel Sport II and SCATT systems [5]. Left side: Noptel Sport II unit and debit card for reference. Middle column from the top: SCATT WS-M02(IR), SCATT Biathlon (IR), SCATT USB (Wired). Right side: SCATT trigger measurement unit.

2.2.2 Scatt

The SCATT Shooter Training System (SCATT) is a Russian company established in 1991 and one of the market leaders within computer-training systems for precision shooting. They have two products of particular interest for biathlon: SCATT Biathlon and SCATT Wireless.

SCATT Biathlon

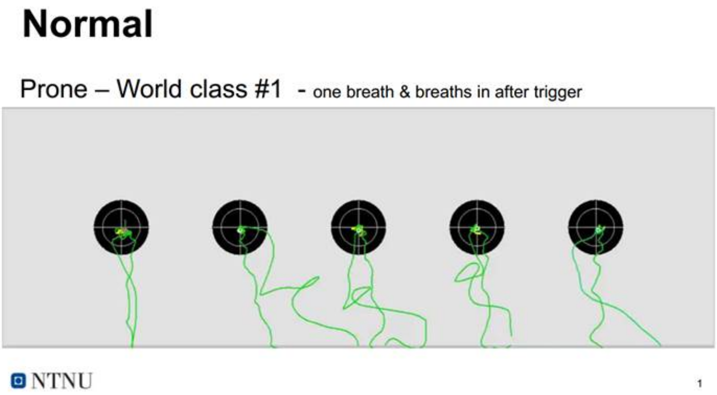
The SCATT Biathlon is developed specifically for indoor biathlon dry-fire training. It uses a special made biathlon target at 5 m range, scaled down to represent a full-size target at 50 m. It can track 5 (dry)shots with high precision and is currently used by SenTIF to measure and evaluate top-tier biathletes on the Norwegian National Team. The output from SCATT Biathlon can be seen in figure 2.5 and figure 2.6.

SCATT Wireless

Whereas SCATT Biathlon is made for specific indoor use, SCATT Wireless is made for general outdoor usage. It's claimed to work at a range of up to 1000 m on a round black target. It does however only support 1 target at a time, and it has been found to be unreliable if several targets are grouped together making it unsuitable for biathlon.

2.2.3 Experience from current systems

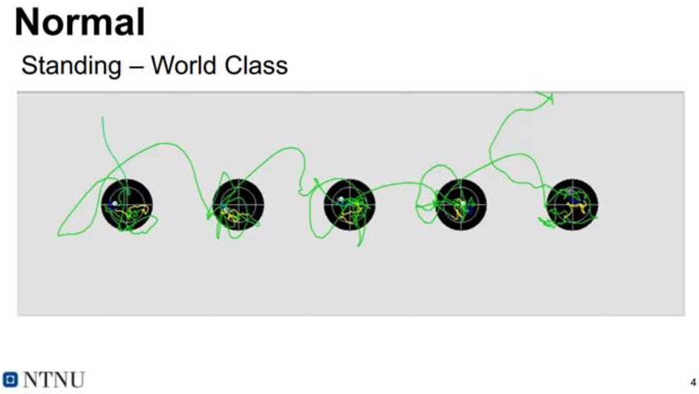
SenTIF has tested both systems. The SCATT Biathlon works as intended indoors and gives precise measurements as seen in figures 2.5 and 2.6. It is however time demanding to use, being wired and requiring connection to a laptop. It's also only available indoors meaning the measurements are not taken in the biathletes natural exercise environment (outdoors, at the range). The same goes for the Noptel system, but as can be observed in figure 2.4 it's large and clumpy and is seldom used in place of SCATT Biathlon. The main problem is the lack of good solutions for outdoor biathlon use, and this capability gap is what this thesis is looking to solve.



Normal

Prone – World class #1 - one breath & breaths in after trigger

Figure 2.5: Data recorded from shooting prone with the SCATT biathlon target [5]



Normal

Standing – World Class

Figure 2.6: Data recorded from shooting standing with the SCATT biathlon target [5]

2.3 System requirements

This thesis focus on the computer vision part of an embedded system for aimpoint tracking. The main overall performance requirements for the system are:

- **Precision** of 1 *mm* at 50 *m*
- **Update frequency** of 100 *Hz*
- **Robustness**: Work outdoors (summer, winter, sun, clouds, rain, snow)
- **Functionality**: *Minimize* required user-input

These requirements are chosen in order to match the performance of existing systems as well as adding capabilities when it comes to robustness and functionality.

Since SenTIF's use-case for this kind of system is scientific, it was also expressed a wish that the output is as close to raw data as possible. I.e. the use of filtering and estimation techniques should be limited for maximum data transparency.

With precision it's meant the ability to output the same result given a stationary input (small variance) and precisely following a moving reference (here: the rifle). Accuracy is not immediately important as some sort of calibration must be undertaken at a later stage.

2.4 Development tools

Matlab and OpenMV were used as software tools in this thesis.

OpenMV is a open source project aimed at making embedded computer vision more accessible to the masses. It's a computer vision library written in microPython and written with embedded applications in mind. A better introduction til OpenMV and it's accompanying hardware can be found in section 4.2.

Matlab and its different toolboxes for computer vision and image analysis are well-known tools. They are well documented online and not explained in further detail in this report.

Chapter 3

Design Concepts

In this chapter different technical solutions and conceptual approaches for fulfilling the required specifications will be explored. Concepts for each area of interest is then chosen and implemented **?** in chapter 4.

3.1 Design approach

Starting from scratch with no previous work done on the project, it was not immediately obvious how to best attack the task at hand. Sources like [10] contains valuable information and insight on how to design computer vision systems for embedded applications, with [11] covering general hardware considerations and [12] design methodologies. In addition there's an abundance of online introduction courses to computer vision. Developing computer vision software for a clearly defined use-case, but with no specific knowledge about either the final hardware nor the sensor to be used, can loosely be thought of as a Chicken or the Egg-dilemma. Should you develop the perfect algorithm first, then to possibly discover it's no good on the embedded platform? What about starting with hardware, choosing all components in order to discover they don't suffice? At the same time the undefined boundaries of the problem gives a lot of design freedom, where the only limitations are given by choices done by oneself.

Given the fact that the application will be embedded, one know from the start that there's not an endless supply of power and computational resources. The overall design goal can then be described as **fulfilling precision and resolution specifications while minimizing power consumption**. From this the following considerations and questions emerge:

- **Precision:** What is needed to obtain the required precision? Depending on the algorithm, what is the *minimum accepted sensor quality* in order to get required precision?
- **Resolution:** Reaching a high resolution/update frequency (*Hz*), depends on available computing power and algorithm effectiveness. How much computing power is actually needed?
- **Power consumption:** Depends on required computing power [1], which is given by the outcome from the *precision* and *resolution* considerations.

The *minimum accepted sensor quality* is a key sentence from the above considerations. It affects both precision and resolution. A higher quality source, here mainly thought of as number of pixels (video resolution), *should* make it easier to obtain the required precision. At the same time a high number of pixels will *increase* the computational load given the increased number of data points the algorithm has access to, slowing down the algorithm. Finding a requirement for sensor quality then seems to be of high importance, and the dilemma of picking a 'good enough' sensor is visualized in figure 3.1.

In order to define sensor requirements, a working feature detection and tracking algorithm had to be implemented. Different camera characteristics could then be tested. The overall design approach can be summed up as:

- Goal: Fulfill precision and resolution specifications while minimizing power consumption

---

[1] It also highly depends on the final hardware, which is outside the scope of this thesis to define. For hardware considerations see [3]

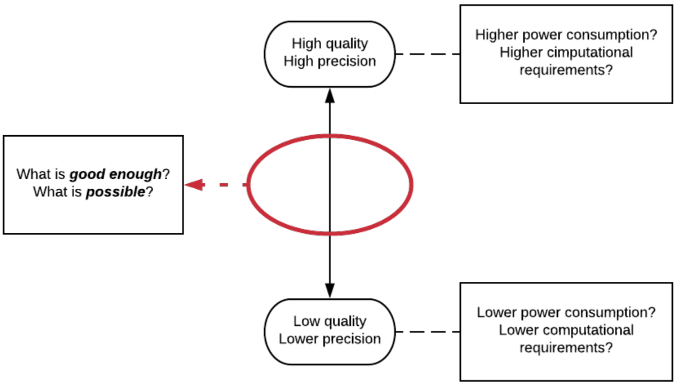


Figure 3.1: Figure visualizing sensor design challenges

1. Implement a working prototype environment
2. Test different camera setups
3. Evaluate precision and update frequency
4. Optimize algorithm for speed when precision is fulfilled
5. Define sensor/camera and computing power requirements
6. Implement the defined hardware and develop software for it (not plausible within the scope of this thesis)

In the following sections technical subsystems are explored.

3.2 Camera

Choosing a camera that is suitable for the task is vital as it creates the data input for the algorithms. When considering camera technologies the operating environment for the system is a defining factor [9]. The operating environment for this system is clearly defined and can be seen in Table 3.1.

Table 3.1: Camera operating environment

Feature	Explanation
Distance to object	50 m
Maximum size of object	1350 mm (horizontally)
Minimum size of interest	Less than 1 mm
Object shape and color	Dark circle against white background
Lighting	Well-lit. Both natural and artificial *visible* light
Environmental variables	Snow and rain

3.2.1 Color vs Monochrome vs NIR camera

Given the biathlon target's black and white appearance, a color camera does not seem to provide additional useful information compared to a monochrome (black and white) camera. In addition, most basic algorithms do only make use of light intensity information and color does not provide any extra information for this.

The presence of rain and snow can occlude the view of the target. Near-Infrared (NIR) cameras utilize the wave-spectrum adjacent to visible light and can be used to 'see through' bad weather, also in low-light conditions [16]. They are however more expensive, and the ability to see the target when a human cannot is not as relevant as the shooting does not happen in zero-vision conditions.

For prototype testing it was chosen to use a monochrome camera and evaluate the performance. If severe snow or rain proves to be difficult to handle with a monochrome source, further investigation into the use of NIR technology might be conducted.

3.3 Feature detection

The first step is to detect the target (feature detection), and the choice of algorithm depends on the target characteristics. As described in section 2.1, the biathlon target is round and dark, surrounded by a white area. The advantage of this is that the transition between dark and white generates strong gradients making it suitable for edge detection, using algorithms such as Canny edge [17] [18]. The challenge is to estimate the target with the required precision from section 2.3.

3.3.1 Algorithm

For detection of circles the CHT is a widely used algorithm making use of edge-detections to estimate circle center and radius [19]. Due to the nature of the algorithm where all candidate points (detected edges) casts votes of possible circles to an accumulator array, memory usage is high and it tends to be computationally expensive [20]. However, the CHT is widely used due to it's robustness. For this application where noise in the presence of snow flakes and rain-drops are present, the advantages were expected to outperform the disadvantages.

3.4 Feature tracking

After detecting and locating the 5 circular targets the next step is to track them. Three conceptual approaches were considered for this specific application.

3.4.1 No tracking

The simplest form of tracking would be to not track anything at all. For each video frame the implemented algorithm could analyze the entire frame each time. Given that the target is stationary and minimal movement during shooting is to be expected (as seen in fig. 2.5 and 2.6), this would not utilize a priori information to limit the analyzed area. It would however make implementation less complex.

Figure 3.2: Visualization of the direct tracking search box

3.4.2 Direct feature tracking

Use a priori information about the target to track the circles. Knowing that the expected movement of the rifle, hence the target, between frames shall be minimal there's no need to analyze the entire frame. With knowledge about the position of the target in the previous frame one can construct a search box centered on the target. The dimensions would be that of the target (in pixels) but expanded with $x_{max-translation}$ and $y_{max-translation}$ in both positive and negative x,y-directions to account for where the target might have moved since last frame. Then CHT can be applied on the datapoints within the search box. See figure 3.2 for a visualization.

3.4.3 Indirect feature tracking

Using other features in the frame to track the movement of the camera, hence the movement of the point of aim. Can detect robust features in the frame and for example use a Kanade-Lucas-Tomasi (KLT) style feature tracker [21]. It is however unclear how

Concept	**No tracking**	**Direct tracking**	**Indirect tracking**
Complexity	*Low*	Low+	Higher
Expected precision	High	High	High?
Computing resources	High	*Less*	*Least?*

Table 3.2: Qualitative comparison of feature tracking concepts.

this would work in total white-out conditions (with the target being the 'only' source of contrast in the picture, the rest covered in white snow), as edge pixels on the target circles could be prone to the aperture problem.

3.4.4 Chosen tracking concept

A comparison of the three conceptual approaches can be seen in table 3.2.

Due to the prototype nature of this project, it was decided to start with *no tracking* to begin with due to ease of implementation. The performance could then be evaluated and the implementation expanded to include *direct tracking* if needed. To further improve from direct tracking *indirect tracking* can be tested, but it's vital to validate its precision compared to tracking the target directly.

3.5 Calibration

The final product needs calibration functionality. It's necessary to sync the estimated Point-of-Aim (PoA) with the real PoA. Together with [3] three calibration concepts were identified and explored:

- **Live fire:** Firing live rounds and analyzing Point-of-Impact (PoI).
- **Laser:** Integral part of the Biathlon Aimpoint Tracking Unit (BATU). Mounted to align with the camera center at 50 m. Adjust until it corresponds with the athlete's PoA.
- **Software:** Translate the output x,y-coordinates to match real-life PoA in visualization software.

Live fire

Standard spread for match-grade .22 long rifle caliber (.22lr) ammunition is 13-19 mm according to SenTIF's own data [5]. Calibrating PoA with PoI observations are therefore infeasible.

Laser

Including a laser in the BATU would give the user the possibility to mechanically adjust the BATU to the point where the laser aligns with the diopter sight of the athlete. It could also be feasible to automatically detect the laser with computer vision software, performing automatic calibration.

Software

It will eventually be necessary to visualize the PoA in dedicated software. The PoA can then be calibrated in the software by the user. The PoA-graph is visualized and the user manually drags it to fit where him/her actually aimed.

3.5.1 Chosen calibration concept

The software method was the chosen calibration concept. First of all it offers most flexibility and is simple to use. Secondly it does not need any work from a computer vision nor a hardware point of view since it's implemented in software at a later stage. Implementing calibration functionality is also not a priority early in the project where actual project plausibility is a lot more important to verify.

3.6 Noise handling

The choice of feature detection algorithm heavily influences noise handling. CHT is already chosen as the main feature detection algorithm. Before identifying edges for the CHT a Gaussian or Median filter will be applied to smooth out process and sensor noise. The expected noise in this application is first and foremost snow and rain, and it’s expected that a majority of it disappears during filtering. No further precautions were taken at this point.

Chapter 4

Implementation & Testing

In this chapter implementation details are presented as well as test methodology and setup. Prototype environments with chosen concepts from chapter 3 were implemented in both Matlab and OpenMV and then tested with respect to the required specifications from section 2.3.

4.1 Matlab

The Matlab system's main goal was to work as a dynamic testing and evaluation environment for different camera and lens configurations, with a focus on algorithm precision validation (given a specific input). The implementation included usage of the Computer Vision Toolbox, Image Processing Toolbox and Camera Calibrator app. These toolboxes are needed if the reader wishes to directly replicate the system. A code example for analyzing a 1080p input source can be seen in A.1, and the corresponding program flow can be seen in figure 4.1.

A short explanation follows: The radius of each round target is estimated as a percentage value of the total height of the input video (i.e. 1080 pixels for a 1080p input source) and results in a estimated range of possible radii (pixel values). The Region of Interest (ROI) is defined according to heuristic experience/a priori information, and the input frame is then cropped to remove unnecessary image content (ref. section 3.4.2). The cropped image along with radius-, threshold- and sensitivity-estimations are sent to the CHT function. Target center X,Y-coordinates along with identified radii are outputted and stored. If the Hough transform doesn't give satisfying results, radii, threshold, and sensitivity parameters must be adjusted. In the end the variance of a detected target is calculated and can be compared to a know reference. Execution-times are also stored for performance evaluations.

The Matlab system was run on a workstation with the following specifications:

- CPU: Intel Core i7-7700 3.60GHz, 4 Cores and 8 threads
- RAM: 32GB 2400MHz
- SSD: PM961 NVMe Samsung 512GB
- GPU: Intel HD Graphics 630

4.2 OpenMV

The OpenMV IDE environment, micro-python programming language [22] and the OpenMV Cam M7 [23] were used for testing concepts on an embedded platform. The same program flow implemented in Matlab applies to the OpenMV as well. A major difference is the real-time video feed of the Open MV CamM7, allowing for real-time testing versus the Matlab implementation which reads from a pre-recorded video file. The code used for the OpenMV Cam M7 can be seen in A.2. Version 1.9.0 of the OpenMV IDE was used together with firmware version 2.9.0.

The Cam M7 is based of a 216MHz ARM Cortex M7 with 512KB RAM.

4.3 Test introduction

A main question concerning the system was whether it was possible to obtain a satisfying precision. Without decent precision the system would not be usable, and

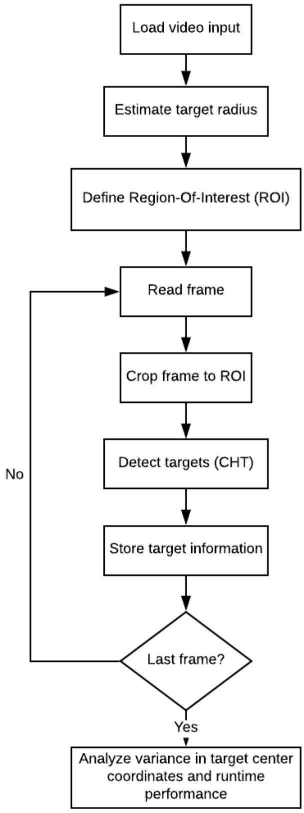


Figure 4.1: Generalized program flow

therefore the precision requirement was more important than the update frequency requirement (section 2.3). Another question related to precision was which video resolution would be sufficient to provide satisfying precision, i.e. how video resolution affected precision. It was decided to use a high-quality video source for capturing test material to be analyzed in Matlab. By doing so one could focus on evaluating algorithm precision given a good data input instead of worrying if the provided data was not of the necessary quality.

The other main question was which update frequencies to expect. It was early understood that reaching update frequencies of up to 100 Hz would pose a challenge (depending on video resolution), even when considering access to fairly powerful hardware[1]. Because of this it was decided to include the OpenMV Cam M7 as an evaluation platform for embedded performance verification as well as representing a Low Quality (LQ) source for algorithm precision testing. This together with the Matlab implementation then gave two different sources for performance testing.

4.4 Test setup

4.4.1 High quality source

A Nikon D7000 digital single-lens reflex camera (DSLR) with a 18-200mm lens were used as the High Quality (HQ) source. Detailed specifications can be seen in table 4.1.

To test system precision real life data was gathered at SenTIF's premises in Granåsen, Trondheim. The camera was mounted on a tripod located at the firing line facing the target. Distance to target were therefore 50m and replicated a biathlete in the standing firing position. Weather conditions were sunny with shadows partly covering the targets. To measure precision, the target was filmed in 20 second clips without moving the camera. Three video resolutions were used: 1080p, 720p and 640x424. In addition, three different focal lengths were used with each resolution: 200mm, 135mm and 70mm. The dataset therefore contains 9 different settings. In addition to the clips with a *non-moving* camera, two clips at each setting were done

---

[1] W10, Intel i7-7700 3.6GHz CPU, 32GB of RAM, 2800MB/s 1600MB/s read/write 512GB SSD

Table 4.1: Camera and lens specifications (HQ)

Feature	Specification
Camera	NIKON D7000
Lens	AF-S DX VR Zoom-NIKKOR 18-200mm f/3.5-5.6G IF-ED
Focal range	18-200mm
Sensor	23.6mm x 15.6mm CMOS
Video options:	1080p at 24 FPS
	720p at 25 FPS
	640x424 at 30 FPS

with a moving (simulating athlete movement) camera for gathering data usable for feature tracking. The camera settings used can be seen in table 4.2. An example snapshot from the data set can be seen in figure 4.2.

Table 4.2: Camera settings used for main data set

Parameter	Specification
Focal lengths	200mm, 135mm and 70mm
Aperture	f5.6
Shutter speed	2000
Focus	manual

4.4.2 Low quality source

The LQ source was the OpenMV Cam M7 [23] with it's OV7725 sensor [24]. Detailed specifications can be seen in table 4.3.

The M7 was used for indoor real-time testing. Given the sensor size and focal length, the M7 was placed at a distance of 25 cm from a printed target to simulate a pixel size of 2 mm corresponding to that of the data captured with the D7000 with a 200 mm focal length and 1080p resolution. A picture of the test-setup can be seen in

Figure 4.2: Example snapshot from HQ test data. 1080p video frame taken with the Nikon D7000. Focal length: 135 mm.

Table 4.3: Camera specifications (LQ)

Feature	Specification
Camera	OpenMV Cam M7
Lens	Custom/unknown
Focal range	2.3mm
Sensor size	3.6mm x 2.7mm
Video options:	VGA(640x480) at 60 FPS QVGA(320x240) at 120 FPS Several others up to VGA

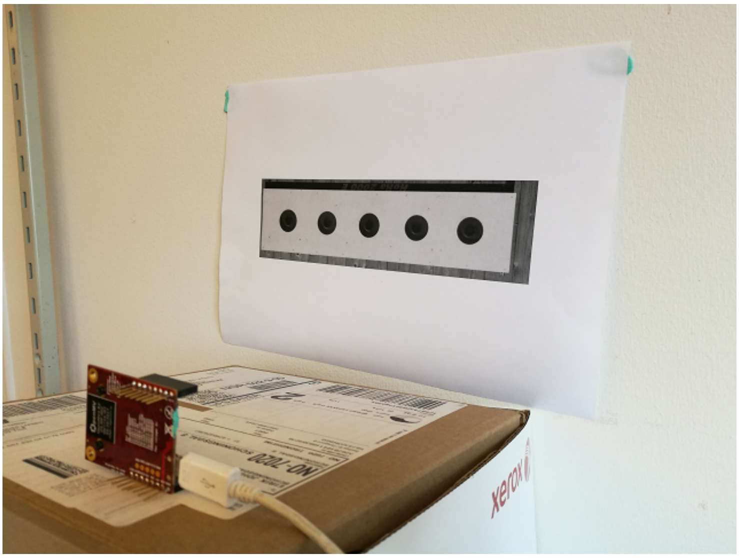


Figure 4.3: Indoor test setup of the OpenMV Cam M7. Distance to target is 25 cm giving a horizontal and vertical pixel size of 2.01 mm.

figure 4.3.

Chapter 5

Results & Discussion

In this chapter results are presented and discussed.

5.1 Results

5.1.1 Precision

The dataset from Granåsen generated by the HQ source were analyzed in Matlab. The results can be seen in table 5.1 and table 5.2. It was seen that given data from a non-moving camera the precision of CHT is good and comparable for all tested resolution. Given that a lower resolution is compensated with a stronger lens (increased focal range) to represent the same pixels/mm-ratio there's no need for a greater resolution than 640x424.

Results from the OpenMV Cam M7 can be seen in table 5.3. The M7 was laying still and stationary during these results. The results differ somewhat from Matlab. It was observed that the precision depends heavily on the magnitude of the detected circle. When tracking a weakly detected circle precision will be poor (Variance of 0.216 pixels) compared to tracking a strong circle (0 variance).

Source	**1080p200mm**	**1080p135mm**	**720p200mm**	**720p135mm**
Var(X)	5.29e-07	1.94e-04	7.52e-04	1.23e-05
Var(Y)	9.69e-08	2.94e-04	9.50e-04	2.02e-04
FPS	0.712	0.705	1.755	2.302

Table 5.1: Matlab results 1080p and 720p. Variance in estimated target center [pixel] and FPS results. 200 mm and 135 mm are focal length numbers.

Source	**640x424 200mm**	**640x424 135mm**	**321x226 200mm**
Var(X)	9.91e-05	3.64e-05	–
Var(Y)	2.88e-05	4.42e-04	–
FPS	6.938	9.575	13.104

Table 5.2: Matlab results 640x424. Variance in estimated target center [pixel] and FPS results. 200 mm and 135 mm are focal length numbers. 321x226 were the size of the tracked ROI when direct tracking was used. Pixels/mm are the same as for 640x424 200mm. An expected substantial increase in FPS can be seen.

Source	**Weak detected circle**	**Strong detected circle**
Var(X)	0.216	0
Var(Y)	0.216	0
FPS w/o ROI	4.56	4.56
FPS w. ROI (20x120)	31.4	31.4

Table 5.3: OpenMV results. 160x140 resolution. Variance in estimated target center [pixel] and FPS results. The *weak detected circle* is a circle that barely passed thresholding while the *strong detected circle* was the strongest/clearest circle. Results from 5 seconds of data. FPS values for both 160x140 and 20x120 are listed.

5.1.2 Update frequency

FPS values from the Matlab implementation are inadequate compared to the wanted specification of 100 Hz. It's quickly concluded that resolutions of 1080p (0.7 Hz) and 720p ($\approx$ 2 Hz) are not feasible for a final implementation. 640x424 can be worked on, but a tenfold increase in FPS is not likely. These values are also run on a desktop computer. More realistic results were obtained from the M7. As can be seen in table 5.3 the M7 managed $\approx$ 4.5 Hz when searching the full frame of 160x140 pixels. When including a region-of-interest search box centered around the detected target limiting the search-space to 20x120 pixels (*The reason for the narrow x-area and full height y-area was due to the M7 laying on the side for practical reasons, making the target vertical in the frame compared to the normal horizontal orientation*) an update frequency of more than 30 Hz was observed. The M7 comes with a fish-eye lens and normally compensates for this by undistorting the image. A fish-eye lens is not needed in the final implementation of this system and the undistortion operation was therefore disabled to simulate having the correct lens. Doing so increased the update frequency further *from 30 Hz to **55 Hz***. When searching the full frame no difference in update frequency were observed when omitting undistortion.

5.1.3 Detection reliability

Problems with false detection and failing to detect actual target circles were encountered and an example can be seen in figure 5.1. The problem with false detection is mostly mitigated when using direct tracking limiting the search area to the target itself. Knowledge about the target circles all being on the same line also works as a filter to filter out false detection. Failing to detect a circle does not need to represent a problem and is discussed in section 5.2. Also, the false detection rate seemed (by observation) to increase as the resolution decreased, but no data was obtained verifying the issue.

5.1.4 Aimpoint tracking

An output from Matlab showing the actual aimpoint tracking can be seen in figure 5.2. The tracking example was done on a 640x424 source with 2 second duration and

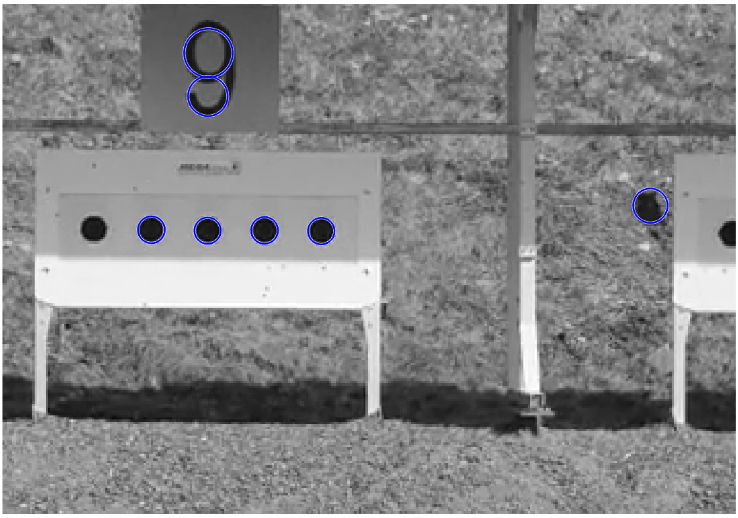


Figure 5.1: Example of CHT giving false detection in the number 9 sign and of the shadow to the right. It also fails to detect the left-most target circle.

200mm focal length. A conservative ROI was used and the figure represents the entire ROI (area outside the ROI is already cropped out and not showing in the figure). The aimpoint coordinates are x,y-values and the tracking graph is drawn onto a frame from the source. The camera was moved from left to right, a bit upwards, to the left then down and to the right again. The tracking was verified by visual comparison of the output and the source video. The uneven part of the tracking is due to camera vibration/resonance because the tripod was not loosened enough before dynamic filming resulting in high friction when trying to pan the camera.

5.1.5 Weather robustness

Due to the time frame of the project and equipment availability the system did not get tested in various weather conditions.

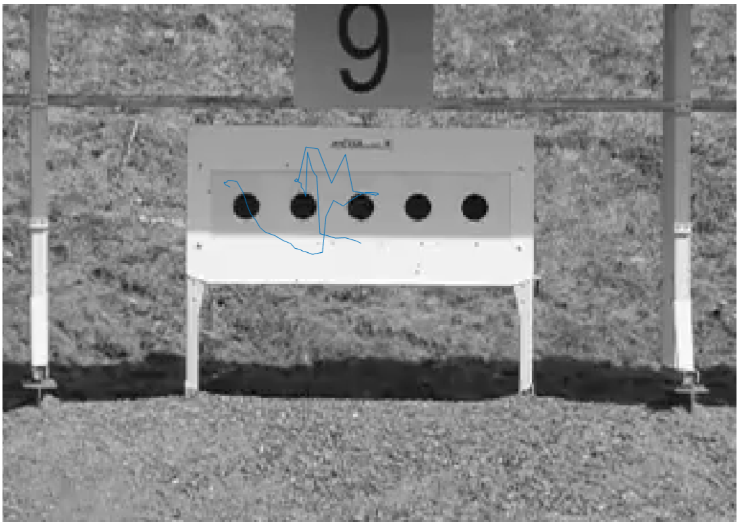


Figure 5.2: Aimpoint tracking output from a 2 second 640x424 source

5.2 Discussion

The goal of this thesis was to look at and start to develop a system for aimpoint tracking in biathlon. By implementing two prototype environments and conducting real world testing, results with respect to (wrt.) precision, update frequency and detection reliability of the proposed design has been presented. A closer look at the implications of the results will now be discussed.

5.2.1 Precision and update frequency

It was found that the precision of the CHT was good even at low resolutions, which means high-resolution, computationally demanding input data is not needed. This gives more freedom when choosing a sensor and puts less strain on the processing capabilities of the hardware. I.e. not only with regards to computing power, but also to buffer sizes etc. On the other hand, to fulfill the requirement of 1 mm precision without using estimation techniques (hardware fulfillment of precision specification) it's necessary that each pixel represents a maximum of 1 mm at 50 m working distance. Using the OV7725 sensor of the Cam M7 as an example sensor the required focal length can be calculated with the following equation

$$Focallength = (Sensorsize * workingdistance)/Fieldofview \tag{5.1}$$

Setting the field of view to twice the width of a biathlon target (2700 mm) we get a focal length of 66.7 mm. Assuming a more standard focal length of 50 mm is used, the 160x120 resolution of the OV7725 results in each pixel representing 22.5 mm at the target. A 22.5 time increase in horizontal resolution to reach 1 mm per pixel is not viable. Using the focal length of 66.7 mm gives a pixel size of 16.87 mm. Decreasing the field of view to that of the target (1350 mm) results in a focal length of 133.3 mm and a pixel size of 8.44 mm. Increasing the resolution to QVGA-standard (320x240) gives 4.22 mm per pixel and a further increase to VGA (640x480) gives 2.11 mm per pixel. These calculations gives a understanding of the difficulty of reaching the precision requirement exclusively through hardware. When using the Cam M7 resolutions above QQVGA (160x120) resulted in memory overflow and system failure. Discussions with SenTIF has resulted in the specification of 1 mm being more of a goal than an absolute necessity. More importantly, the initially required 100 Hz is also a goal and an update frequency of 50 Hz will still give a lot of valuable data [5]. The CHT can in addition be implemented to estimate circle centers in decimal numbers instead of entire pixels (estimating in half pixels will increase the perceived precision correspondingly). On the basis of this, recommending the usage of a OV7725-sized sensor with QVGA resolution and 135 mm focal length can be concluded from test results. It shall give good precision while at the same time offering good enough update frequencies. If the original specification of 1 mm precision and especially 100 Hz is to be fulfilled, the usage of specialized hardware (e.g. FPGA or a Visual Processing Unit (VPU)) is considered to be necessary.

Sensor size	1/4"
Resolution	QVGA (320x240)
Focal length	135mm
Resulting mm/pixel	4.22

Table 5.4: Camera specification recommendations based of a 1/4" sized sensor.

5.2.2 Detection reliability

The issue of failing to detect a target circle was described in 5.1.3. For the majority of the work on this thesis, tracking all 5 circles seemed a requirement. After analyzing results, this no longer seems necessary. As long as one is tracking a robust feature and know its position there's no need to simultaneously track all target circles. The algorithm's unreliability to always track all target circles are therefore not a problem as long as one always know the position of at least one circle. When the target is identified (recognizing several circles are on a line and are separated by given distances), tracking the strongest circle will suffice. For redundancy and minimizing the risk of the chosen circle going out of view, several strong circles can be tracked. This can also further reduce the ROI and increase update frequency.

5.3 Recommendations

Recommendations for future *prototype* development of this system can be made based on the findings in the above sections. For justification of the recommendations chapter 5 must be read in its entirety.

5.3.1 Hardware

Recommended camera sensor specification can be seen in table 5.4. The recommended camera specifications should provide sufficient resolution and field-of-view to obtain the desired precision while minimizing required computing power.

For computing power a chip equivalent to a 400MHz ARM Cortex-M7 with 1MB

RAM is recommended, such as the STM32F765VI [25]. It features twice the computing power of the chip found in the Cam M7.

5.3.2 Software

Matlab was used for proof-of-concept and examining the effect resolution had on precision. With this work being completed, it's recommended to abandon Matlab for future prototyping and instead focus on more specialized software languages and IDE environments. The OpenMV language offers great usability and user friendliness, but one is highly dependable on the firmware released by the OpenMV-team. For full control of the system it's recommended to start using OpenCV right away. OpenCV is a versatile computer vision library designed for efficiency and real-time applications and has both C++, Java and Python interfaces. OpenMV is however still a great way to quickly get introduced to the field of embedded computer vision. In short:

- Abandon Matlab
- Start using OpenCV

Chapter 6

Conclusion

In this report we have gone through the development process from an idea to a working prototype environment. The project was proven to be feasible and further development can be recommended. Design concepts were presented and implemented. The usage of CHT was found to deliver good precision in a wide range of resolutions. Reaching high update frequencies proved a challenge using Matlab, and current results are not satisfactory wrt. the required specification. Satisfactory update frequencies were obtained using the OpenMV Cam M7 but at insufficient video resolutions. Direct tracking of the target using a region-of-interest search box was found to work well while not affecting precision nor introducing a lot of complexity. Approaches for further increasing tracking robustness and reliability were presented. Camera specifications and hardware specifications for future implementations were identified and presented. In addition it was concluded that one should abandon Matlab and change to OpenCV as the main development software.

The main challenge is to reach satisfactory update frequencies, and suggestions for future work on how to approach it is presented next.

6.1 Future work

With CHT proven to deliver high precision but at a high computational prize, three main approaches for increasing update frequencies are suggested:

- Increase available computing power
- Write lower-level software aimed at specific hardware
- Make use of specialized integrated circuits

Increase available computing power

Increasing available computing power is thought of as acquiring a micro-controller with significant more computing power than the current 216MHz Cortex M7 that's part of the Cam M7. See section 5.3.1.

Low-level software

The current implementations utilizes generalized computer vision libraries. Writing the feature detection and feature tracking algorithm from scratch designed to work with specific hardware components could provide an substantial increase in performance.

Make use of specialized integrated circuits

Reaching 100 Hz update frequency at VGA-resolution was not found to be possible even with a strong desktop CPU. The usage of specialized hardware is necessary in order to reach 100 Hz while ensuring 1 mm precision trough the usage of a a high resolution sensor (section 5.2.1). An idea for a future master thesis is to look at the usage of a FPGA or a VPU in order to reach what can be said to be SenTIFs 'dream' specification [5].

Appendix A

Appendix A

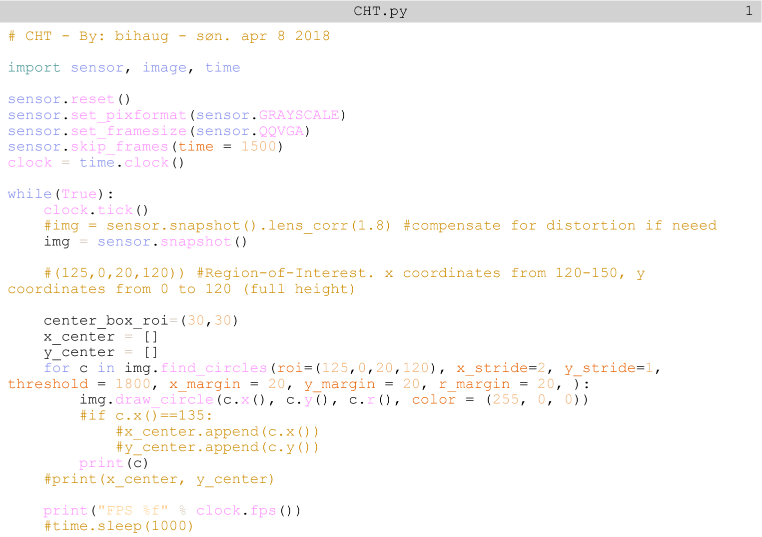
A.1 Matlab

A.2 OpenMV

06.06.18 14:34 C:\Users\bih...\CHT_1080p_input_example.m 1 of 2

```
clear workspace

videoFReader = vision.VideoFileReader('C:\Users\bihaug\OneDrive -↙
NTNU\Masteroppgave\Test material\Grayscales\1080p10fps_200mm_snapshot.mkv');
videoPlayer = vision.VideoPlayer;   %creates video player object

circle_center_tracking_x = [0,0];   %arrays for tracking circle centers
circle_center_tracking_y = [0,0];   %arrays for tracking circle centers

circle_radius_estimation = 2;       %value in percentage of total width of image↙
(200mm)
%circle_radius_estimation = 1.5;    %value in percentage of total width of image↙
(135mm)

video_size = [1920 1080];   %Video sizes for radius and crop estimations
%video_size = [1280 720];
%video_size = [640 450];

circle_radius = int8(video_size(2:2)*circle_radius_estimation/100);

r_min = circle_radius*0.80; %20% error in estimating radius
r_max = circle_radius*1.20; %20% error in estimating radius

x_crop_start = 0.25*video_size(1,1);    %Start crop at 25% of image width
x_crop_end = 0.75*video_size(1,1);      %End crop at 75% of image width
y_crop_start = 0.25*video_size(1,2);    %Start crop at 25% of image height
y_crop_end = 0.75*video_size(1,2);      %End crop at 75% of image height
width = x_crop_end-x_crop_start;        %Width to crop
height = y_crop_end-y_crop_start;       %Height to crop

number_of_circles = 5;
counter=1;
tic;
while ~isDone(videoFReader)
    videoFrame = videoFReader();

    croppedFrame=imcrop(videoFrame,[x_crop_start y_crop_start width height]);   %Crop↙
frame to Region-Of-Interest
    imshow(croppedFrame);

    [centersDark, radiiDark, metricDark] = imfindcircles(croppedFrame,[r_min↙
r_max],'ObjectPolarity','dark', 'Sensitivity', 0.92, 'EdgeThreshold',0.22);

    %centersDarkStrongest=centersDark(1:number_of_circles,:);   %Stores the %5↙
strongest detected circles
    %radiiDarkStrongest=radiiDark(1:number_of_circles,:);
    %metricDarkStrongest=metricDark(1:number_of_circles,:);

    %viscircles(centersDarkStrongest, radiiDarkStrongest,'Color','b');
    viscircles(centersDark, radiiDark,'Color','b');

    circle_center_tracking_y(counter)=centersDark(1,2);
    circle_center_tracking_x(counter)=centersDark(1,1);

    videoPlayer(croppedFrame);
```

Figure A.1: Code for analyzing a 1080p input stream



```
CHT.py                                                                 1
# CHT - By: bihaug - søn. apr 8 2018

import sensor, image, time

sensor.reset()
sensor.set_pixformat(sensor.GRAYSCALE)
sensor.set_framesize(sensor.QQVGA)
sensor.skip_frames(time = 1500)
clock = time.clock()

while(True):
    clock.tick()
    #img = sensor.snapshot().lens_corr(1.8) #compensate for distortion if neeed
    img = sensor.snapshot()

    #(125,0,20,120)) #Region-of-Interest. x coordinates from 120-150, y
coordinates from 0 to 120 (full height)

    center_box_roi=(30,30)
    x_center = []
    y_center = []
    for c in img.find_circles(roi=(125,0,20,120), x_stride=2, y_stride=1,
threshold = 1800, x_margin = 20, y_margin = 20, r_margin = 20, ):
        img.draw_circle(c.x(), c.y(), c.r(), color = (255, 0, 0))
        #if c.x()==135:
            #x_center.append(c.x())
            #y_center.append(c.y())
        print(c)
    #print(x_center, y_center)

    print("FPS %f" % clock.fps())
    #time.sleep(1000)
```

Figure A.2: Code used with the OpenMV Cam M7

Bibliography

[1] Espen Bekkelien. “Validation of using Inertial Measurement Unit to measure the rifle barrel trajectory in biathlon”. Project report. Department of Engineering Cybernetics, Norwegian University of Science and Technology, Norway, 2017.

[2] Brynjar Tufteland Engelsen and Lars Løken Bjørnskau. “Measuring Trigger-Techniques for Biathlon athletes”. Master Thesis. Department of Engineering Cybernetics, Norwegian University of Science and Technology, Norway, 2017.

[3] Eirik Johannes Larsen. “Hardware for optical aimpoint tracking of biathlon rifles”. Project report. Department of Engineering Cybernetics, Norwegian University of Science and Technology, Norway, 2018.

[4] Stine Dombu. “Logging of Sensor Data in Biathlon”. Project report. Department of Engineering Cybernetics, Norwegian University of Science and Technology, Norway, 2018.

[5] Harri Luchsinger - Research Assistant at Olympiatoppen. Consultation regarding biathlon subjects. 2018.

[6] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach.* Prentice Hall Professional Technical Reference, 2002. ISBN: 978-0-13-085198-7.

[7] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN: 978-0-13-168728-8.

[8] Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference.* 1st. New York, NY, USA: Cambridge University Press, 2012. ISBN: 978-1-107-01179-3.

[9] Horst Mattfeldt. “Camera Systems in Machine Vision”. en. In: *Handbook of Machine Vision*. Wiley-Blackwell, 2007, pp. 361–426. ISBN: 978-3-527-61013-6. DOI: `10.1002/9783527610136.ch6`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527610136.ch6` (visited on 03/17/2018).

[10] Branislav Kisačanin, Shuvra S. Bhattacharya, and Sek Chai, eds. *Embedded computer vision*. en. Advances in pattern recognition. OCLC: ocn227032509. London: Springer, 2009. ISBN: 978-1-84800-303-3 978-1-84800-304-0.

[11] Mathias Kölsch and Steven Butner. “Hardware Considerations for Embedded Vision Systems”. en. In: *Embedded Computer Vision*. Advances in Pattern Recognition. Springer, London, 2009, pp. 3–26. ISBN: 978-1-84800-303-3 978-1-84800-304-0. DOI: `10.1007/978-1-84800-304-0_1`. URL: `https://link.springer.com/chapter/10.1007/978-1-84800-304-0_1` (visited on 02/15/2018).

[12] Sankalita Saha and Shuvra S. Bhattacharyya. “Design Methodology for Embedded Computer Vision Systems”. en. In: *Embedded Computer Vision*. Advances in Pattern Recognition. Springer, London, 2009, pp. 27–47. ISBN: 978-1-84800-303-3 978-1-84800-304-0. DOI: `10.1007/978-1-84800-304-0_2`. URL: `https://link.springer.com/chapter/10.1007/978-1-84800-304-0_2` (visited on 02/15/2018).

[13] *International Biathlon Union - IBU, Annexes to the IBU Event and Competition Rules 2016*. URL: `http://www.biathlonworld.com/downloads/` (visited on 02/19/2018).

[14] *Chiemgau_Arena_5.JPG (800×502)*. URL: `https://www.live-production.tv/sites/default/files/styles/original_image_with_copyright/public/Chiemgau_Arena_5.JPG?itok=7cQQ3G4s` (visited on 01/27/2018).

[15] Noptel Oy. *Noptel Oy - Sport Training Products - Sport II Training System*. en. URL: `http://www.noptel.com/eng2/Sport/index.php?doc=2_products/2_sportii` (visited on 02/17/2018).

[16] Nicolas PINCHON, M IBN-KHEDHER, Olivier CASSIGNOL, et al. "All-weather vision for automotive safety: which spectral band?" In: *SIA Vision 2016 - International Conference Night Drive Tests and Exhibition*. Paris, France: Société des Ingénieurs de l'Automobile - SIA, Oct. 2016, 7p. URL: `https://hal.archives-ouvertes.fr/hal-01406023` (visited on 03/10/2018).

[17] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN: 978-0-13-168728-8.

[18] Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference*. 1st. New York, NY, USA: Cambridge University Press, 2012. ISBN: 978-1-107-01179-3.

[19] Mark Haidekker. *Advanced Biomedical Image Analysis*. Hoboken, UNITED STATES: Wiley, 2011. ISBN: 978-0-470-87208-6. URL: `http://ebookcentral.proquest.com/lib/ntnu/detail.action?docID=661692` (visited on 03/20/2018).

[20] Adel A. Sewisy. "Detection of Circular Object with a High Speed Algorithm". In: *Intelligent Problem Solving. Methodologies and Approaches*. Ed. by Rasiah Logananthara, Günther Palm, and Moonis Ali. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 522–534. ISBN: 978-3-540-45049-8.

[21] Carlo Tomasi and Takeo Kanade. *Detection and Tracking of Point Features*. Tech. rep. International Journal of Computer Vision, 1991.

[22] *MicroPython - Python for microcontrollers*. en. URL: `http://micropython.org/` (visited on 06/02/2018).

[23] *OpenMV Cam M7*. URL: `https://openmv.io/products/openmv-cam-m7` (visited on 06/02/2018).

[24] *OmniVision | A leading developer of advanced digital imaging solutions*. en. URL: `http://www.ovt.com/sensors/OV7725` (visited on 06/02/2018).

[25] *STM32H743VI - High-performance and DSP with DP-FPU, ARM Cortex-M7 MCU with 2MBytes Flash, 1MB RAM, 400 MHz CPU, Art Accelerator, L1 cache, external memory interface, large set of peripherals - STMicroelectronics*. URL: `http://www.st.com/en/microcontrollers/stm32h743vi.html` (visited on 06/03/2018).